



Guide de l'utilisateur

# Amazon CodeCatalyst



# Amazon CodeCatalyst: Guide de l'utilisateur

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

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

---

# Table of Contents

Qu'est-ce qu'Amazon CodeCatalyst ? .....	1
Que puis-je en faire CodeCatalyst ? .....	1
Comment puis-je commencer CodeCatalyst ? .....	2
En savoir plus sur CodeCatalyst .....	2
Concepts .....	3
AWS Espaces Builder ID dans CodeCatalyst .....	4
Espaces qui soutiennent la fédération des identités dans CodeCatalyst .....	4
Projets .....	4
Plans .....	4
Connexions aux comptes .....	5
Connexions VPC .....	5
AWS ID du constructeur .....	5
Profils d'utilisateurs dans CodeCatalyst .....	6
Référentiels de sources .....	6
Validations .....	7
Environnements de développement .....	7
Flux de travail .....	8
Actions .....	8
Problèmes .....	8
Jetons d'accès personnels (PAT) .....	9
Rôles .....	9
Configuration .....	10
Inscription et création de ressources .....	12
Création de votre premier espace et de vos premiers rôles IAM .....	13
Accepter une invitation .....	19
Acceptation d'une invitation et création d'un identifiant AWS Builder .....	20
Connectez-vous avec votre identifiant AWS Builder .....	21
Appareils fiables .....	22
Accepter une invitation par e-mail pour vous connecter avec SSO .....	22
Connectez-vous avec SSO .....	23
Afficher tous les espaces et projets d'un utilisateur .....	23
Afficher et gérer les CodeCatalyst profils .....	24
Consulter votre CodeCatalyst profil .....	25
Consulter le CodeCatalyst profil d'un autre utilisateur .....	25

Mettre à jour votre profil .....	26
Modification de votre CodeCatalyst mot de passe .....	27
Configuration pour utiliser AWS CLI le CodeCatalyst .....	27
Tutoriels de mise en route .....	30
Tutoriel : Création d'un projet avec le plan d'application Web moderne à trois niveaux .....	31
Prérequis .....	33
Étape 1 : Création du projet d'application Web moderne à trois niveaux .....	34
Étape 2 : Invitez quelqu'un à rejoindre votre projet .....	36
Étape 3 : créer des problèmes sur lesquels collaborer et suivre le travail .....	36
Étape 4 : Afficher votre référentiel source .....	37
Étape 5 : Création d'un environnement de développement avec une branche de test et modification rapide du code .....	38
Étape 6 : Afficher le flux de travail qui crée l'application moderne .....	40
Étape 7 : demandez à d'autres personnes de revoir vos modifications .....	43
Étape 8 : Clôturer le problème .....	47
Nettoyage des ressources .....	47
Référence .....	48
Tutoriel : Commencer avec un projet vide .....	50
Prérequis .....	51
Création d'un projet vide .....	51
Création d'un référentiel source .....	51
Création d'un flux de travail pour créer, tester et déployer une modification de code .....	53
Invitez quelqu'un à rejoindre votre projet .....	53
Créez des problèmes sur lesquels collaborer et suivre le travail .....	54
Tutoriel : Utilisation des fonctionnalités génératives de l'IA .....	55
Prérequis .....	56
Ajouter un résumé généré automatiquement lors de la création d'une pull request .....	57
Créez un résumé des commentaires laissés sur les modifications de code dans une pull request .....	60
Créez un problème et attribuez-le à Amazon Q .....	61
Nettoyage des ressources .....	67
Tutoriel : Création d'une application complète avec des plans PDK composables .....	68
Prérequis .....	70
Étape 1 : créer un projet monorepo .....	70
Étape 2 : Ajouter l'API Type Safe au projet .....	71
Étape 3 : ajouter un site Web Cloudscape React pour le projet .....	73

Étape 4 : générer l'infrastructure pour déployer l'application sur le cloud AWS .....	74
Étape 5 : configurer un DevOps flux de travail pour déployer votre projet .....	76
Étape 6 : Confirmez le flux de publication et consultez votre site Web .....	78
Collaborez et itérez sur le projet PDK .....	84
Espaces .....	101
Création d'un espace prenant en charge les utilisateurs de AWS Builder ID .....	103
Modification d'un espace .....	106
Supprimer un espace .....	107
Surveillance de l'activité d'un espace .....	108
Administration Comptes AWS d'un espace .....	108
Ajouter un Compte AWS à un espace .....	109
Ajouter des rôles IAM aux connexions aux comptes .....	113
Ajout de la connexion au compte et des rôles IAM à votre environnement de déploiement ...	115
Afficher les connexions aux comptes .....	116
Supprimer un compte d'un espace (en CodeCatalyst) .....	116
Gestion des rôles IAM pour les comptes connectés .....	118
Rôle CodeCatalystWorkflowDevelopmentRole- <i>spaceName</i> .....	119
Rôle AWSRoleForCodeCatalystSupport .....	120
Création d'un rôle IAM et utilisation de la politique de CodeCatalyst confiance .....	121
Gestion des utilisateurs de l'espace .....	122
Afficher les membres dans un espace .....	122
Inviter un utilisateur directement dans un espace .....	124
Annulation d'une invitation pour un espace .....	125
Modifier le rôle d'un membre de l'espace .....	126
Supprimer un membre de l'espace .....	127
Supprimer ou modifier le rôle d'un utilisateur ayant le rôle d'administrateur de l'espace .....	127
Gérer des équipes .....	129
Création d'une équipe .....	130
Afficher une équipe .....	132
Gérer le rôle spatial d'une équipe .....	132
Gérer un rôle de projet pour une équipe .....	133
Ajouter directement un utilisateur à une équipe .....	134
Supprimer directement un utilisateur d'une équipe .....	135
Ajouter un groupe SSO à une équipe .....	135
Supprimer une équipe .....	136
Gestion des ressources des machines .....	137

Affichage des ressources de la machine .....	137
Désactiver les ressources de la machine .....	137
Activer les ressources de la machine .....	138
Administration des environnements de développement pour un espace .....	139
Visualisation des environnements de développement pour votre espace .....	140
Modification d'un environnement de développement pour votre espace .....	141
Arrêter un environnement de développement pour votre espace .....	141
Supprimer un environnement de développement pour votre espace .....	142
Quotas de places .....	143
Projets .....	145
Création d'un projet .....	146
Création d'un projet à l'aide d'un plan .....	147
Création d'un projet vide .....	148
Création d'un projet avec un GitHub référentiel lié .....	148
Visualisation d'un projet .....	151
Afficher les tâches du projet et les environnements de développement .....	151
Afficher tous les projets .....	152
.....	153
Affichage des paramètres du projet .....	153
Passage à un autre projet dans CodeCatalyst .....	153
Suppression d'un projet .....	153
Gérer les membres du projet .....	154
Afficher les membres d'un projet .....	154
Inviter un utilisateur à rejoindre votre projet .....	156
Annulation d'une invitation .....	157
Supprimer un utilisateur de votre projet .....	157
Accepter ou refuser une invitation pour un projet .....	158
Gestion des équipes pour les projets .....	159
Ajouter une équipe à un projet .....	159
Gérer un rôle de projet pour une équipe .....	160
Supprimer un rôle de projet pour une équipe .....	160
Gestion des ressources de la machine .....	161
Affichage des ressources de la machine .....	161
Désactiver les ressources de la machine .....	161
Activer les ressources de la machine .....	163
Quotas pour les projets .....	163

Utilisation des notifications .....	164
Comment fonctionnent les notifications ? .....	165
Commencer à utiliser les notifications Slack .....	167
Gestion des notifications .....	170
Plans .....	176
Création d'un projet à l'aide d'un plan .....	177
Appliquer et dissocier des plans dans un projet .....	177
Appliquer un plan à un projet .....	178
Dissociation d'un plan d'un projet .....	179
Mettre à jour un plan dans un projet .....	179
Modification de la description d'un plan dans un projet .....	181
Travailler avec la gestion du cycle de vie en tant qu'utilisateur du plan .....	181
Utilisation de la gestion du cycle de vie sur les projets existants .....	182
Utilisation de la gestion du cycle de vie sur plusieurs plans d'un projet .....	182
Gérer les conflits liés au cycle de vie des pull requests .....	182
Se désinscrire des modifications apportées à la gestion du cycle de vie .....	183
Remplacer la gestion du cycle de vie d'un plan dans un projet .....	183
Référence du plan de projet .....	184
Plans disponibles .....	184
Trouver des informations sur le plan de projet .....	189
Travailler avec des plans personnalisés .....	189
Concepts de plans personnalisés .....	190
Commencer à utiliser des plans personnalisés .....	194
Tutoriel : Création et mise à jour d'une application React .....	198
Travailler avec la gestion du cycle de vie en tant qu'auteur de plans .....	206
Élaboration d'un plan personnalisé .....	212
Publication d'un plan personnalisé .....	243
Afficher les détails, les versions et les projets d'un plan personnalisé .....	248
Ajouter et supprimer un plan personnalisé dans un espace .....	249
Gestion des autorisations de publication pour un plan personnalisé .....	250
Gestion des versions pour un plan personnalisé .....	251
Suppression d'un plan ou d'une version personnalisés publiés .....	252
Utilisation des dépendances et de l'outillage .....	253
Contribuer .....	256
Quotas pour les plans .....	256
Référentiels de sources .....	257

Concepts du référentiel source .....	259
Projets .....	4
Référentiels de sources .....	259
Environnements de développement .....	7
Jetons d'accès personnels (PAT) .....	9
Branches .....	261
Branches par défaut .....	261
Validations .....	7
Demandes d'extraction .....	262
Révisions .....	262
Flux de travail .....	8
Configuration .....	263
Installez Git .....	264
Créez un jeton d'accès personnel .....	264
Commencer à utiliser les référentiels de sources .....	265
Création d'un projet à l'aide d'un plan .....	265
Afficher les référentiels d'un projet .....	267
Création d'un environnement de développement .....	268
Création d'une demande d'extraction .....	270
Fusion d'une pull request .....	272
Afficher le code déployé .....	273
Nettoyage des ressources .....	274
Utilisation des référentiels de sources .....	274
Création d'un référentiel source .....	276
Lier un référentiel source .....	277
Affichage d'un référentiel source .....	278
Modification des paramètres d'un référentiel source .....	279
Clonage d'un dépôt source .....	280
Supprimer un référentiel source .....	283
Utilisation de branches .....	284
Création et suppression d'une branche .....	285
Afficher et modifier la branche par défaut d'un référentiel .....	286
Gérer les règles des succursales .....	288
Commandes Git pour les branches .....	290
Afficher les branches et les détails .....	292
Travailler avec des fichiers .....	293



Création ou ajout d'un fichier .....	293
Affichage d'un fichier .....	296
Modification d'un fichier .....	297
Modification du nom ou suppression d'un fichier .....	298
Utilisation des demandes d'extraction .....	298
Création d'une demande d'extraction .....	300
Afficher les pull requests .....	304
Gérer les règles d'approbation .....	307
Révision d'une pull request .....	309
Mettre à jour une pull request .....	312
Fusion d'une pull request .....	314
Clôture d'une pull request .....	318
Travailler avec des commits .....	318
Afficher les validations d'une branche .....	319
Modifier le mode d'affichage des validations (CodeCatalystconsole) .....	320
Quotas pour les référentiels sources .....	321
Environnements de développement .....	326
Création d'un environnement de développement .....	327
Environnements de développement intégrés pris en charge pour les environnements de développement .....	328
Création d'un environnement de développement dans CodeCatalyst .....	328
Création d'un environnement de développement dans un IDE .....	331
Arrêt d'un environnement de développement .....	332
Reprise d'un environnement de développement .....	333
Modification d'un environnement de développement .....	335
Suppression d'un environnement de développement .....	336
Connexion à un environnement de développement via SSH .....	337
Configuration de votre environnement de développement .....	339
Modification d'un fichier de développement d'un référentiel pour un environnement de développement dans CodeCatalyst .....	341
Modification d'un fichier de développement d'un référentiel pour un environnement de développement dans un IDE .....	342
Déplacer un fichier de développement d'un référentiel pour un environnement de développement .....	343
Mode de récupération .....	343
Fonctionnalités de Devfile prises en charge par CodeCatalyst .....	343

Exemple : Configuration du fichier de développement pour votre environnement de développement .....	344
Commandes Devfile .....	345
Événements Devfile .....	346
Composants Devfile .....	347
Images de fichier universelles .....	347
Utilisation d'environnements de développement avec une connexion VPC .....	353
Utilisation d'environnements de développement avec un IDE .....	354
Quotas pour les environnements de développement .....	355
Packages .....	357
Concepts de packages .....	358
Packages .....	358
Espaces de noms de packages .....	358
Versions du package .....	358
Assets .....	359
Référentiels de packages .....	359
Référentiels Gateway .....	359
Référentiels en amont .....	360
Utilisation des référentiels de packages .....	360
Création d'un référentiel de packages .....	360
Connexion à un référentiel de packages .....	361
Modification d'un référentiel de packages .....	361
Suppression d'un dépôt de packages .....	362
Utilisation de référentiels en amont .....	362
Ajouter un référentiel en amont .....	363
Modification de l'ordre de recherche des référentiels en amont .....	364
Demande d'une version de package avec des référentiels en amont .....	365
Supprimer un référentiel en amont .....	368
Connexion à des référentiels externes publics .....	369
Référentiels de packages externes pris en charge et leurs référentiels de passerelle .....	370
Travailler avec des packages .....	370
Publication de packages .....	371
Afficher les détails de la version du package .....	372
Supprimer une version de package .....	372
Mettre à jour le statut d'une version de package .....	373
Modification des contrôles d'origine des packages .....	374

Utilisation de npm .....	380
Configuration et utilisation de npm .....	380
gestion des balises npm .....	390
Quotas pour les packages .....	392
Créez, testez et déployez avec des flux de travail .....	393
À propos du fichier de définition du flux de travail .....	393
Utilisation des éditeurs visuels et YAML de la CodeCatalyst console .....	395
Découvrir les flux de travail .....	397
Afficher les détails de l'exécution du workflow .....	398
Étapes suivantes .....	399
Concepts de workflows .....	399
Flux de travail .....	399
Fichiers de définition du flux de travail .....	400
Actions .....	400
Groupes d'action .....	400
Artefacts .....	401
Calcul .....	401
Environnements .....	401
Rapports .....	401
Exécutions .....	402
Sources .....	402
Variables .....	402
Déclencheurs de .....	402
Commencer à utiliser les flux de travail .....	403
Prérequis .....	403
Étape 1 : créer et configurer votre flux de travail .....	404
Étape 2 : Enregistrez votre flux de travail avec un commit .....	406
Étape 3 : Afficher les résultats de la course .....	406
(Facultatif) Étape 4 : Nettoyer .....	407
Affichage de la qualité du code et de l'état du déploiement par validation .....	407
Construire à l'aide de workflows .....	409
Comment créer une application ? .....	409
Avantages de l'action Build .....	410
Alternatives à l'action de construction .....	411
Ajouter l'action de construction .....	411
Affichage des résultats .....	412

Tutoriel : télécharger des artefacts sur Amazon S3 .....	413
Tests à l'aide de workflows .....	422
Types de rapports de test .....	423
Ajouter l'action de test .....	426
Configuration des rapports .....	426
En utilisant universal-test-runner .....	442
Bonnes pratiques .....	444
Travailler avec des tests .....	447
Déploiement au moyen de workflows .....	450
Comment déployer une application ? .....	450
Liste des actions de déploiement .....	451
Avantages des actions de déploiement .....	452
Alternatives aux actions de déploiement .....	452
Tutoriel : déploiement à l'aide de CloudFormation .....	453
Tutoriel : Déploiement sur Amazon ECS .....	480
Tutoriel : Déploiement sur Amazon EKS .....	517
Ajout de l'action « Deploy AWS CloudFormation stack » .....	550
Ajout de l'action « Déployer sur Amazon ECS » .....	553
Ajout de l'action « Déployer vers le cluster Kubernetes » .....	556
Ajout de l'action « AWS CDK déployer » .....	560
Utilisation de déploiements .....	567
Utilisation des flux de travail .....	579
Création, modification et suppression d'un flux de travail .....	580
Affichage de l'état du flux de travail .....	584
Utilisation des actions .....	586
Travailler avec des artefacts .....	648
Utilisation des images Docker de l'environnement de calcul et d'exécution .....	661
Utilisation des environnements .....	686
Utilisation de la mise en cache de fichiers .....	694
Travailler avec des packages .....	699
Travailler avec des courses .....	704
Travailler avec des secrets .....	716
Travailler avec des sources .....	721
Utilisation de déclencheurs .....	725
Utilisation des variables .....	743
Référence de définition du flux de travail .....	768

Exemple de fichier de définition de flux de travail .....	769
Consignes et conventions de syntaxe .....	770
Propriétés de haut niveau .....	772
Créez et testez une référence d'actions .....	784
Référence d'action « Amazon S3 publish » .....	814
référence d'action « AWS CDK bootstrap » .....	824
référence d'action « AWS CDK déployer » .....	836
référence d'action « AWS Lambda invoquer » .....	853
Référence d'action « Deploy AWS CloudFormation Stack » .....	868
Référence d'action « Déployer sur Amazon ECS » .....	887
Référence d'action « Déployer vers un cluster Kubernetes » .....	900
Référence GitHub d'action « Actions » .....	910
Référence d'action « Définition de la tâche de rendu » .....	931
Quotas pour les workflows .....	939
Problèmes .....	942
Concepts des problèmes .....	943
Problèmes actifs .....	943
Numéros archivés .....	944
Cessionnaire .....	944
Champs personnalisés .....	944
Estimation .....	944
Problème .....	944
Étiquette .....	945
Priorité .....	945
Statut et catégories de statut .....	945
Tâches .....	945
Vues .....	945
Création d'un problème .....	946
Bonnes pratiques pour les problèmes attribués à Amazon Q .....	948
Rédaction et collaboration sur les problèmes .....	950
Modification d'un numéro .....	951
Utilisation de pièces jointes .....	953
Gestion des tâches liées aux problèmes .....	954
Marquer un problème comme bloqué ou débloqué .....	955
Ajouter, modifier ou supprimer des commentaires .....	955
Progression d'un problème .....	957

Archivage d'un problème .....	959
Recherche et affichage des problèmes .....	960
Recherche d'un problème .....	961
Problèmes de tri .....	961
Problèmes de regroupement .....	962
Problèmes de filtrage .....	963
Création d'une vue des problèmes .....	964
Problèmes liés à l'exportation .....	964
Configuration des paramètres des problèmes .....	965
Activation ou désactivation de plusieurs destinataires .....	965
Configuration de l'estimation des efforts liés aux problèmes .....	966
Statutes .....	966
Étiquettes .....	968
Champs personnalisés .....	970
Afficher et gérer les pièces jointes .....	970
Quotas pour les numéros .....	971
Identité, autorisations et accès .....	973
Utilisation des rôles .....	974
Types de rôles .....	974
Autorisations disponibles pour chaque rôle .....	977
Afficher et modifier les rôles des utilisateurs .....	1021
Gestion des jetons d'accès personnels .....	1023
Création de PAT .....	1023
Affichage des PAT .....	1026
Supprimer des PAT .....	1027
Authentification multifactorielle (MFA) sur Amazon CodeCatalyst .....	1028
Comment enregistrer un appareil pour une utilisation avec l'authentification multifactorielle .....	1029
Applications d'authentification .....	1031
Modification de vos appareils MFA .....	1032
Sécurité .....	1033
Protection des données .....	1034
CodeCatalyst et Identity and Access Management .....	1037
Validation de conformité .....	1104
Résilience .....	1105
Sécurité de l'infrastructure .....	1105
Analyse de la configuration et des vulnérabilités .....	1106

Vos données et votre confidentialité sur Amazon CodeCatalyst .....	1106
Bonnes pratiques pour les actions de flux de travail .....	1107
CodeCatalyst modèle de confiance .....	1108
Surveillance sur Amazon CodeCatalyst .....	1110
Enregistrement des appels d' CodeCatalyst API vers les comptes connectés dans AWS	
CloudTrail .....	1113
Accès aux événements enregistrés dans CodeCatalyst .....	1122
Quotas d'identité, d'autorisation et d'accès .....	1125
Résolution des problèmes .....	1126
Problèmes lors de l'inscription .....	1126
Problèmes de connexion .....	1127
Problèmes de déconnexion .....	1128
Je reçois une erreur indiquant qu'un rôle n'existe pas suite à un échec du flux de travail ...	1128
Je reçois une erreur de rôle suite à l'échec d'un flux de travail .....	1128
Je dois mettre à jour le rôle IAM dans le flux de travail d'un projet .....	1129
Comment remplir un formulaire d'assistance ? .....	1129
Extensions .....	1130
Concepts d'extensions .....	1130
Extensions .....	1130
CodeCatalyst catalogue .....	1130
Installation et désinstallation d'extensions .....	1130
Installation d'une extension .....	1131
Désinstaller une extension .....	1132
Utilisation de GitHub référentiels dans CodeCatalyst .....	1132
Démarrage rapide : utilisation de GitHub référentiels dans CodeCatalyst .....	1134
Gestion des GitHub comptes .....	1137
Gestion des GitHub référentiels .....	1140
Affichage des GitHub référentiels liés .....	1145
Utilisation de GitHub référentiels liés dans les flux de travail .....	1145
Utilisation des restrictions d'adresse IP dans GitHub Enterprise Cloud .....	1147
Blocage des fusions de GitHub pull request en cas d'échec des flux de travail .....	1148
Problèmes liés à l'utilisation de Jira dans CodeCatalyst .....	1148
Démarrage rapide : problèmes liés à l'utilisation de Jira dans CodeCatalyst .....	1149
Gestion des sites Jira .....	1153
Gestion des projets Jira .....	1156
Lier les problèmes liés à Jira aux pull requests .....	1158

Afficher CodeCatalyst des événements dans Jira .....	1160
Recherche de problèmes liés à Jira dans CodeCatalyst .....	1161
Search .....	1162
Affiner votre requête de recherche .....	1163
Affinage par type .....	1163
Affinage par domaine .....	1164
Affinage à l'aide d'opérateurs booléens .....	1164
Affinage par projet .....	1165
Considérations relatives à l'utilisation de la recherche .....	1165
Référence aux champs consultables .....	1166
Résolution des problèmes .....	1172
Résolution des problèmes d'accès généraux .....	1172
J'ai oublié mon mot de passe .....	1172
Une partie ou la totalité d'Amazon CodeCatalyst n'est pas disponible .....	1173
Je ne parviens pas à créer un projet dans CodeCatalyst .....	1173
Résolution des problèmes de support .....	1173
Je reçois un message d'erreur lorsque j'AWS Supportaccède à Amazon CodeCatalyst .....	1173
Je ne peux pas créer de dossiers de support technique pour mon espace .....	1174
Mon compte pour les demandes d'assistance n'est plus connecté à mon espace dans CodeCatalyst .....	1174
Je ne parviens pas à ouvrir un dossier d'assistance pour un autre Service AWSAWS Support utilisateur sur Amazon CodeCatalyst .....	1175
Une partie ou la totalité d'Amazon CodeCatalyst n'est pas disponible .....	1173
Je ne parviens pas à créer un projet dans CodeCatalyst .....	1173
Je souhaite envoyer des commentaires dans CodeCatalyst .....	1173
Résolution des problèmes liés aux référentiels sources .....	1176
J'ai atteint la capacité de stockage maximale pour mon espace et je reçois des avertissements ou des erreurs .....	1177
Je reçois un message d'erreur lorsque j'essaie de cloner ou de transférer vers un référentiel CodeCatalyst source Amazon .....	1177
Je reçois un message d'erreur lorsque j'essaie de valider ou de transférer vers un référentiel CodeCatalyst source Amazon .....	1178
J'ai besoin d'un référentiel source pour mon projet .....	1179
Mon dépôt source est tout neuf mais contient un commit .....	1179
Je veux une branche différente comme branche par défaut .....	1179
Je reçois des e-mails concernant l'activité des pull requests .....	1180



J'ai oublié mon jeton d'accès personnel (PAT) .....	1180
Une pull request n'affiche pas les modifications que j'attends .....	1180
Une pull request indique le statut Non fusionnable .....	1181
Projets et plans de résolution des problèmes .....	1181
API Java avec AWS Fargate plan de dépendances manquantes pour apache- maven-3.8.6 .....	1182
Le flux de travail moderne du plan d'application Web à trois niveaux OnPullRequest échoue en raison d'une erreur d'autorisation pour Amazon CodeGuru .....	1183
Vous cherchez toujours à résoudre votre problème ? .....	1187
Dépannage des workflows .....	1187
Comment corriger les messages « Le flux de travail est inactif » ? .....	1188
Comment corriger les erreurs « La définition du flux de travail <i>ne</i> comporte aucune erreur » ? .....	1189
Comment corriger les erreurs « Impossible de localiser les informations d'identification » et ExpiredToken « » ? .....	1191
Comment corriger les erreurs « Impossible de se connecter au serveur » ? .....	1193
Pourquoi les CodeDeploy champs sont-ils absents de l'éditeur visuel ? .....	1193
Comment corriger les erreurs liées aux fonctionnalités IAM ? .....	1194
Comment corriger les erreurs « npm install » ? .....	1196
Pourquoi plusieurs flux de travail portent-ils le même nom ? .....	1199
Puis-je stocker les fichiers de définition de mon flux de travail dans un autre dossier ? .....	1200
Comment ajouter des actions en séquence à mon flux de travail ? .....	1200
Pourquoi mon flux de travail est-il correctement validé mais échoue-t-il lors de l'exécution ? .....	1201
La détection automatique ne détecte aucun rapport concernant mon action .....	1201
Mon action échoue sur les rapports découverts automatiquement une fois que j'ai configuré les critères de réussite .....	1202
La découverte automatique génère des rapports dont je ne veux pas .....	1202
La découverte automatique génère de nombreux petits rapports pour une seule infrastructure de test .....	1202
Les flux de travail répertoriés sous CI/CD ne correspondent pas à ceux du référentiel source .....	1203
Je n'arrive pas à créer ou à mettre à jour des flux de travail .....	1204
Recherche de résolution des problèmes .....	1205
Je ne trouve pas d'utilisateur dans mon projet .....	1205
Je ne vois pas ce que je recherche dans mon projet ou mon espace .....	1205

Le nombre de résultats de recherche ne cesse de changer lorsque je navigue dans les pages .....	1206
Ma requête de recherche n'est pas terminée .....	1206
Résolution des problèmes liés aux comptes .....	1206
Ma demande Compte AWS de connexion reçoit une erreur de jeton non valide .....	1207
CodeCatalyst Le flux de travail de mon projet Amazon échoue avec une erreur concernant le compte, l'environnement ou le rôle IAM configuré .....	1207
J'ai besoin d'un compte, d'un rôle et d'un environnement associés pour créer un projet .....	1209
Je ne parviens pas à accéder à la page Amazon CodeCatalyst Spaces dans AWS Management Console .....	1209
Je souhaite utiliser un autre compte comme compte de facturation .....	1179
Dépannage des environnements de développement .....	1210
La création de mon environnement de développement n'a pas réussi en raison d'un problème de quotas .....	1211
Je ne peux pas transférer les modifications de mon environnement de développement vers une branche spécifique d'un référentiel .....	1212
Mon environnement de développement n'a pas repris .....	1212
Mon environnement de développement est déconnecté .....	1212
Mon environnement de développement connecté au VPC a échoué .....	1212
Je ne trouve pas le répertoire dans lequel se trouve mon projet .....	1213
Je ne parviens pas à me connecter à mon environnement de développement via SSH .....	1213
Je ne parviens pas à me connecter à mon environnement de développement via SSH car ma configuration SSH locale est manquante .....	1213
Je ne parviens pas à me connecter à mon environnement de développement via SSH car je rencontre des problèmes avec mon AWS Config profil codecatalyst .....	1214
Dépannage des IDE .....	1214
Dépannage des fichiers de développement .....	1216
Résolution des problèmes .....	1218
Je ne parviens pas à choisir un destinataire pour mon problème .....	1219
Dépannage AWS CLI et problèmes liés au SDK .....	1219
Je reçois un message d'erreur lorsque j'entre aws codecatalyst sur une ligne de commande ou un terminal indiquant que le choix n'est pas valide .....	1219
Je reçois une erreur d'identification lorsque j'exécute aws codecatalyst des commandes ...	1220
CodeCatalyst rapport de santé .....	1221
CodeCatalyst concepts de rapport de santé .....	1221
Incident .....	1222

---

Statut .....	1222
Fonctionnalités impactées .....	1222
Mis à jour le .....	1222
AWS Support pour Amazon CodeCatalyst .....	1223
Facturation AWS Support pour Amazon CodeCatalyst .....	1223
Configuration de votre espace AWS Support pour Amazon CodeCatalyst .....	1226
Accès au support pour CodeCatalyst AWS Management Console .....	1227
Création d'un dossier de CodeCatalyst support dans CodeCatalyst .....	1228
Résolution d'un dossier d'assistance dans CodeCatalyst .....	1231
Réouverture d'un dossier de support dans CodeCatalyst .....	1232
Quotas .....	1234
Historique de la documentation .....	1236
AWS Glossaire .....	1260
.....	mcclxi

# Qu'est-ce qu'Amazon CodeCatalyst ?

Amazon CodeCatalyst est un service intégré destiné aux équipes de développement de logiciels qui adoptent des pratiques d'intégration et de déploiement continues dans leur processus de développement logiciel. CodeCatalyst réunit tous les outils dont vous avez besoin au même endroit. Vous pouvez planifier le travail, collaborer sur le code et créer, tester et déployer des applications à l'aide d'outils d'intégration continue/de livraison continue (CI/CD). Vous pouvez également intégrer AWS des ressources à vos projets en vous connectant Comptes AWS à votre CodeCatalyst espace. En gérant toutes les étapes et tous les aspects du cycle de vie de vos applications dans un seul outil, vous pouvez fournir des logiciels rapidement et en toute confiance.

Dans CodeCatalyst, vous créez un espace pour représenter votre entreprise, votre département ou votre groupe, puis vous créez des projets contenant les ressources nécessaires pour soutenir vos équipes de développement et leurs tâches. CodeCatalyst les ressources sont structurées au sein de projets qui vivent dans des espaces. Pour aider les équipes à démarrer rapidement, CodeCatalyst fournit des plans de projet basés sur un langage ou des outils. Lorsque vous créez un projet à partir d'un plan de projet, celui-ci est fourni avec des ressources telles qu'un référentiel de sources avec des exemples de code, des scripts de génération, des actions de déploiement, des serveurs virtuels ou des ressources sans serveur, etc.

## Que puis-je en faire CodeCatalyst ?

Vous et votre équipe de développement pouvez CodeCatalyst prendre en charge chaque aspect du développement logiciel, de la planification de votre travail au déploiement de vos applications. Vous pouvez utiliser CodeCatalyst pour :

- Itérez et collaborez sur le code — Travaillez en collaboration avec votre équipe sur le code à l'aide de branches, de fusions, de pull requests et de commentaires dans vos référentiels de code source. Créez des environnements de développement pour travailler rapidement sur le code sans avoir à cloner ou à configurer des connexions aux référentiels.
- Créez, testez et déployez votre application avec des flux de travail : configurez des flux de travail avec des actions de création, de test et de déploiement pour gérer l'intégration et la livraison continues de vos applications. Vous pouvez démarrer les flux de travail manuellement ou les configurer pour qu'ils démarrent automatiquement en fonction d'événements tels que les envois de code ou la création ou la fermeture de pull requests.

- Hiérarchisez le travail de votre équipe grâce au suivi des problèmes : utilisez les problèmes pour créer des arriérés et surveillez l'état des tâches en cours à l'aide de tableaux. La création et le maintien d'un carnet de tâches important sur lequel votre équipe pourra travailler constituent un élément important du développement de logiciels.
- Configuration de la surveillance et des notifications : surveillez l'activité de l'équipe et l'état des ressources, et configurez les notifications pour rester au courant des modifications importantes.

## Comment puis-je commencer CodeCatalyst ?

Si vous n'avez pas d'espace ou si vous souhaitez savoir comment configurer et gérer un espace, nous vous recommandons de commencer par le manuel [Amazon CodeCatalyst Administrator Guide](#).

Si vous débutez dans le domaine d'un projet ou d'un espace, nous vous recommandons de commencer par :

- Révision du [CodeCatalyst concepts](#)
- [Création d'un espace prenant en charge les utilisateurs de AWS Builder ID](#)
- Créez votre premier projet en suivant les étapes décrites dans [Tutoriel : Création d'un projet avec le plan d'application Web moderne à trois niveaux](#)

## En savoir plus sur CodeCatalyst

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

- [AWS DevOps Articles de blog sur Amazon CodeCatalyst](#)
- [Le guide de référence des CodeCatalyst API Amazon](#)
- [Guide du développeur du kit de développement Amazon CodeCatalyst Action](#)
- [CodeCatalyst FAQ](#)
- [Témoignages](#)

# CodeCatalyst concepts

Familiarisez-vous avec les concepts clés pour accélérer votre collaboration et le développement d'applications sur Amazon CodeCatalyst. Ces concepts incluent les termes utilisés dans les domaines du contrôle de source, de l'intégration continue et de la livraison continue (CI/CD), ainsi que de la modélisation et de la configuration des processus de publication automatisés.

Pour des informations conceptuelles supplémentaires, consultez les rubriques suivantes :

- [Concepts du référentiel source](#)
- [Concepts de workflows](#)

## Rubriques

- [AWS Espaces Builder ID dans CodeCatalyst](#)
- [Espaces qui soutiennent la fédération des identités dans CodeCatalyst](#)
- [Projets](#)
- [Plans](#)
- [Connexions aux comptes](#)
- [Connexions VPC](#)
- [AWS ID du constructeur](#)
- [Profils d'utilisateurs dans CodeCatalyst](#)
- [Référentiels de sources](#)
- [Validations](#)
- [Environnements de développement](#)
- [Flux de travail](#)
- [Actions](#)
- [Problèmes](#)
- [Jetons d'accès personnels \(PAT\)](#)
- [Rôles](#)

# AWS Espaces Builder ID dans CodeCatalyst

L'administrateur de l'espace invite les utilisateurs CodeCatalyst en envoyant des e-mails d'invitation individuels depuis la page des membres. Utilisateurs invités ou s'inscrivant pour CodeCatalyst créent leur propre identifiant AWS Builder. Le profil est géré dans AWS Builder ID et s'affiche sous forme de nom d'utilisateur et d'informations de profil dans les paramètres utilisateur de CodeCatalyst.

## Espaces qui soutiennent la fédération des identités dans CodeCatalyst

Utilisateurs qui ont été ajoutés aux utilisateurs et groupes SSO pour l'instance IAM Identity Center, qui sont gérés dans le magasin d'identités et qui sont invités à accéder à votre espace via IAM Identity Center. L'administrateur de Space synchronise la page des CodeCatalyst membres pour les dernières mises à jour. Les utilisateurs se connectent à l'aide du portail de connexion SSO tel que configuré dans l'instance IAM Identity Center de l'entreprise. Les espaces prenant en charge la fédération d'identités sont connectés à l'instance de banque d'identités via l'application Identity Center et son mappage à l'ID de la banque d'identités.

## Projets

Un projet représente un effort de collaboration CodeCatalyst qui soutient les équipes et les tâches de développement. Une fois que vous avez un projet, vous pouvez ajouter, mettre à jour ou supprimer des utilisateurs et des ressources, personnaliser le tableau de bord de votre projet et suivre l'avancement du travail de votre équipe. Vous pouvez avoir plusieurs projets au sein d'un même espace.

Pour plus d'informations sur les projets, consultez [Projets en CodeCatalyst](#).

## Plans

Un plan est un synthétiseur de projet qui génère et étend les fichiers de support des applications et les dépendances pour vous, tout en créant votre CodeCatalyst projet dans la console. Vous choisissez un type de projet parmi une sélection de plans CodeCatalyst, vous visualisez le fichier README et vous prévisualisez le référentiel de projets et les ressources qui seront générés. Votre projet est généré à partir de la configuration de base spécifiée par le plan. Vous effectuez régulièrement une synthèse avec le plan de projet, qui met à jour les fichiers de votre projet, tels que

les dépendances logicielles, et régénère les ressources. Les projets utilisent un outil appelé Projén pour synthétiser les projets en synchronisant les dernières mises à jour des projets et en générant des fichiers de support. Ces fichiers peuvent inclure `package.json`, `Makefile`, `eslint`, et plus encore en fonction du type et de la langue de votre application. Les plans de projet peuvent générer des fichiers supportant AWS des ressources telles que des constructions, des modèles et des AWS CloudFormation modèles CDK. AWS Serverless Application Model

Pour plus d'informations sur les plans de projet, consultez [Référence du plan de projet](#).

## Connexions aux comptes

Une connexion à un compte associe un CodeCatalyst espace à votre Compte AWS. Une fois la connexion de votre compte configurée, elle Compte AWS est mise à la disposition de l'espace. Vous pouvez ensuite ajouter des rôles IAM CodeCatalyst afin qu'il puisse accéder aux ressources de votre Compte AWS. Vous pouvez également utiliser ces rôles pour vos actions de CodeCatalyst flux de travail.

Pour plus d'informations sur les connexions aux comptes, consultez [Administration Comptes AWS d'un espace](#).

## Connexions VPC

Une connexion VPC est une CodeCatalyst ressource qui contient toutes les configurations nécessaires pour que votre flux de travail accède à un VPC. Les administrateurs de l'espace peuvent ajouter leurs propres connexions VPC dans la CodeCatalyst console Amazon au nom des membres de l'espace. En ajoutant une connexion VPC, les membres de l'espace peuvent exécuter des actions de flux de travail et créer des environnements de développement conformes aux règles du réseau et pouvant accéder aux ressources du VPC associé.

Pour plus d'informations sur les connexions VPC, consultez [Managing Amazon Virtual Private Clouds](#) dans le guide de l'CodeCatalyst administrateur.

## AWS ID du constructeur

Un AWS Builder ID est une identité personnelle que vous pouvez utiliser pour vous inscrire et vous connecter aux CodeCatalyst autres applications participantes. Ce n'est pas la même chose qu'un Compte AWS. Votre AWS Builder ID gère les métadonnées telles que l'alias d'utilisateur et l'adresse



e-mail. Votre identifiant AWS Builder est une identité unique qui prend en charge les utilisateurs dans tous les espaces de CodeCatalyst. Pour plus d'informations sur l'accès à votre profil AWS Builder ID, consultez [Mettre à jour votre profil](#). Pour en savoir plus sur AWS Builder ID, consultez [AWS Builder ID](#) dans le Références générales AWS.

Pour plus d'informations sur l'inscription et la connexion, consultez [Configuration CodeCatalyst](#).

## Profils d'utilisateurs dans CodeCatalyst

Vous accédez à votre profil CodeCatalyst utilisateur en choisissant l'option de profil dans le menu déroulant situé sous vos initiales de connexion sur n'importe quelle page de CodeCatalyst. Vous pouvez créer des jetons d'accès personnels (PAT) à partir de votre page de profil, mais vous ne pouvez afficher ou supprimer des PAT qu'à l'aide du. AWS CLI Votre nom d'utilisateur est l'alias que vous avez choisi lors de votre inscription. Vous ne pouvez pas modifier votre nom d'utilisateur. Pour consulter la page de profil CodeCatalyst d'un autre utilisateur, accédez à l'onglet Membres de votre projet et choisissez l'utilisateur approprié.

Vous accédez à votre AWS Builder ID en consultant votre CodeCatalyst profil, puis en choisissant d'accéder à AWS Builder ID. Vous serez redirigé vers votre page de profil AWS Builder ID. Le nom complet, l'adresse e-mail et le mot de passe de votre profil sont gérés par votre AWS Builder ID, et vous pouvez modifier ces informations à l'aide de la page AWS Builder ID. Vous avez saisi ces informations lors de votre inscription. Lorsque vous serez prêt à configurer le MFA pour utiliser une application d'authentification pour vous connecter, vous utiliserez la AWS page Builder ID. Pour plus d'informations sur l'affichage de votre profil AWS Builder ID, consultez [Mettre à jour votre profil](#).

Pour plus d'informations sur l'inscription et la connexion, consultez [Configuration CodeCatalyst](#).

## Référentiels de sources

Un référentiel source est l'endroit où vous stockez en toute sécurité le code et les fichiers de votre projet. Il enregistre également l'historique des versions de vos fichiers. Par défaut, un référentiel source est partagé avec les autres utilisateurs de votre CodeCatalyst projet. Vous pouvez disposer de plusieurs référentiels sources pour un projet. Vous pouvez créer des référentiels sources pour les projets dans CodeCatalyst, ou vous pouvez choisir de lier un référentiel source existant hébergé par un autre service si ce service est pris en charge par une extension installée. Par exemple, vous pouvez lier un GitHub dépôt à un projet après avoir installé l'extension GitHub Repositories. Pour plus d'informations, consultez [Utilisation des référentiels de sources dans CodeCatalyst](#) et [Démarrage rapide : utilisation de GitHub référentiels dans CodeCatalyst](#).

Les référentiels sources sont également l'endroit où sont stockées les informations de configuration de votre CodeCatalyst projet, telles que le fichier de configuration qui définit les attributs et les actions de votre flux de travail CI/CD. Si vous créez votre projet à l'aide d'un plan, un référentiel source contenant les informations de configuration du projet sera créé. Si vous créez un projet vide, vous devez créer un référentiel source avant de pouvoir créer des ressources nécessitant des informations de configuration, telles que des flux de travail.

Pour plus de concepts susceptibles de vous aider à utiliser les référentiels de sources et le contrôle de source, consultez [Concepts du référentiel source](#).

## Validations

Un commit est une modification apportée à un fichier ou à un ensemble de fichiers. Dans la CodeCatalyst console Amazon, un commit enregistre vos modifications et les envoie vers un référentiel source. Le commit inclut des informations sur le changement, notamment l'identité de l'utilisateur qui a effectué le changement, l'heure et la date du changement, le titre du commit et tout message inclus concernant le changement. Pour plus d'informations, consultez [Utilisation des commits sur Amazon CodeCatalyst](#).

Dans le contexte d'un dépôt source en CodeCatalyst, les validations sont des instantanés des modifications apportées au contenu de votre dépôt. Chaque fois qu'un utilisateur valide et introduit une modification, il CodeCatalyst enregistre des informations telles que l'auteur de la modification, la date et l'heure de la validation, ainsi que les modifications apportées dans le cadre de la validation. Vous pouvez également ajouter des balises Git aux validations pour aider à identifier des validations spécifiques.

Pour plus d'informations sur les validations, consultez [Utilisation des commits sur Amazon CodeCatalyst](#).

## Environnements de développement

Un environnement de développement est un environnement de développement basé sur le cloud que vous pouvez utiliser CodeCatalyst pour travailler rapidement sur le code stocké dans les référentiels sources de votre projet. Les outils de projet et les bibliothèques d'applications inclus dans votre environnement de développement sont définis par un fichier de développement dans le référentiel source de votre projet. Si vous n'avez pas de fichier de développement dans votre dépôt source, un fichier de développement par défaut sera automatiquement appliqué. Le fichier de développement par défaut inclut des outils pour les langages de programmation et les frameworks

les plus fréquemment utilisés. Par défaut, un environnement de développement est configuré pour disposer d'un processeur à 2 cœurs, de 4 Go de RAM et de 16 Go de stockage persistant.

## Flux de travail

Un flux de travail est une procédure automatisée qui décrit comment créer, tester et déployer votre code dans le cadre d'un système d'intégration et de livraison continues (CI/CD). Un flux de travail définit une série d'étapes, ou d'actions, à effectuer lors de son exécution. Un flux de travail définit également les événements, ou déclencheurs, qui déclenchent le démarrage du flux de travail. Pour configurer un flux de travail, vous devez créer un fichier de définition de flux de travail à l'aide de [l'éditeur visuel ou YAML](#) de la CodeCatalyst console.

### Tip

Pour un aperçu rapide de la manière dont vous pouvez utiliser les flux de travail dans un projet, [créez un projet avec un plan](#). Chaque plan déploie un flux de travail fonctionnel que vous pouvez examiner, exécuter et tester.

Pour de plus amples informations sur les flux de travail, veuillez consulter [Créez, testez et déployez avec des flux de travail dans CodeCatalyst](#).

## Actions

Une action est le principal élément constitutif d'un flux de travail et définit une unité logique de travail à effectuer lors de l'exécution d'un flux de travail. Généralement, un flux de travail inclut plusieurs actions qui s'exécutent de manière séquentielle ou en parallèle selon la façon dont vous les avez configurées.

Pour plus d'informations sur les actions, consultez [Utilisation des actions](#).

## Problèmes

Un problème est un enregistrement qui permet de suivre le travail lié à votre projet. Vous pouvez créer un problème lié à une fonctionnalité, à une tâche, à un bogue ou à toute autre tâche liée à votre projet. Si vous utilisez le développement agile, un problème peut également décrire une histoire épique ou une histoire utilisateur.

Pour plus d'informations sur les problèmes, consultez [Problèmes liés à CodeCatalyst](#).

## Jetons d'accès personnels (PAT)

Un jeton d'accès personnel (PAT) est similaire à un mot de passe. Il est associé à votre identité d'utilisateur pour être utilisé dans tous les espaces et projets de CodeCatalyst. Vous utilisez les PAT pour accéder à CodeCatalyst des ressources qui incluent des environnements de développement intégrés (IDE) et des référentiels de sources basés sur Git. Les PAT vous représentent CodeCatalyst et vous pouvez les gérer dans vos paramètres utilisateur. Un utilisateur peut avoir plusieurs PAT. Les jetons d'accès personnels ne s'affichent qu'une seule fois. Il est recommandé de les stocker en toute sécurité sur votre ordinateur local. Par défaut, les PAT expirent au bout d'un an.

Pour plus d'informations sur les PAT, consultez [Gestion des jetons d'accès personnels sur Amazon CodeCatalyst](#).

## Rôles

Un rôle définit l'accès d'un utilisateur aux ressources d'un projet ou d'un espace et les actions que cet utilisateur peut effectuer. Vous choisissez le rôle d'un utilisateur lorsque vous l'invitez à participer à un projet. Il existe des rôles au niveau de l'espace et des rôles au niveau du projet dans CodeCatalyst. Un utilisateur doté d'un rôle administratif au niveau approprié peut modifier les rôles assignés. Par exemple, un utilisateur ayant le rôle d'administrateur de projet pour un projet a le contrôle total de ce projet et peut modifier les rôles des utilisateurs dans ce projet. Pour plus d'informations sur les rôles disponibles et les autorisations dont dispose chaque rôle, consultez [Utilisation des rôles sur Amazon CodeCatalyst](#).

Pour plus d'informations sur les rôles, consultez [Utilisation des rôles sur Amazon CodeCatalyst](#).

# Configuration CodeCatalyst

Il existe deux types d'espace que vous pouvez configurer CodeCatalyst : les espaces qui prennent en charge les utilisateurs de AWS Builder ID et la création d'un espace qui prend en charge la fédération d'identité, dans lequel les utilisateurs et les groupes SSO sont gérés dans IAM Identity Center. Les utilisateurs d'un espace AWS Builder ID se connectent à l' aide de leur AWS Builder ID, et les utilisateurs d'un espace Entreprise se connectent à l' aide du portail SSO de l'entreprise associée à l'espace.

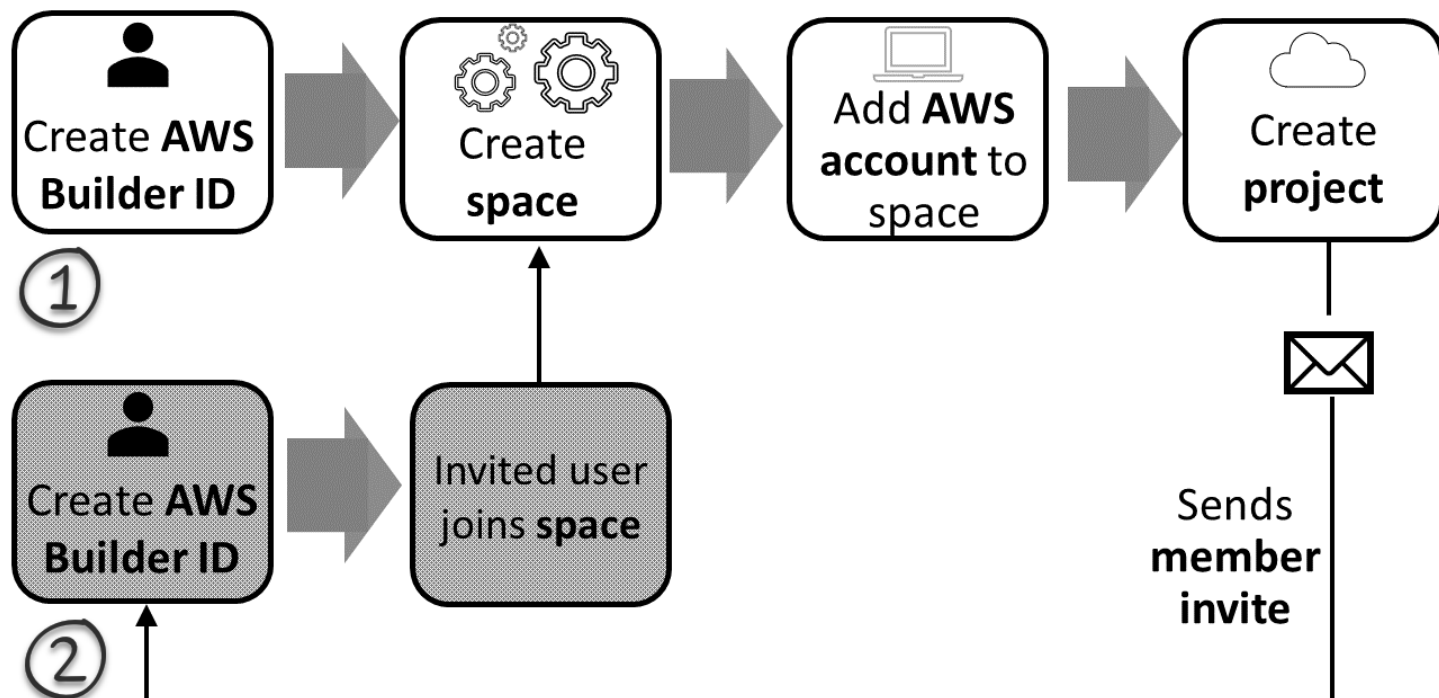
Les étapes de configuration et d'administration d'un espace AWS Builder ID sont décrites dans ce guide. Pour utiliser un espace CodeCatalyst AWS Builder ID, vous devez le configurer à CodeCatalyst l'aide des paramètres utilisateur et du AWS Builder ID auxquels vous vous connectez CodeCatalyst.

Les étapes de configuration et d'administration d'un espace prenant en charge la fédération d'identités sont décrites dans le guide de l'CodeCatalyst administrateur. Pour utiliser des espaces configurés pour la fédération d'identité, consultez la section [Configuration et administration des CodeCatalyst espaces](#) dans le manuel Amazon CodeCatalyst Administrator Guide.

Cette section fournit deux méthodes courantes pour configurer l'utilisation d'un espace AWS Builder ID sur Amazon CodeCatalyst : créer un espace et un projet en tant que premier utilisateur, et accepter une invitation à accéder à un espace ou à un projet existant. Ces flux de travail de configuration sont nécessairement très différents. Le schéma suivant montre les deux processus d'inscription comme suit :

Il existe deux méthodes courantes pour configurer Amazon pour travailler CodeCatalyst : créer un espace et un projet en tant que premier utilisateur, et accepter une invitation à accéder à un espace ou à un projet existant. Ces flux de travail de configuration sont nécessairement très différents. Le schéma suivant montre les deux processus d'inscription comme suit :

1. Dans le premier cas, vous créez et configurez un espace pour votre entreprise, votre équipe ou votre groupe, et vous créez un projet avant d'inviter d'autres personnes à accéder à ces ressources. Un Compte AWS doit être fourni à des fins de facturation, où vous pouvez toujours opter par défaut pour le niveau gratuit.
2. Dans le second cas, si vous vous inscrivez CodeCatalyst en acceptant une invitation à participer à un projet, quelqu'un d'autre a déjà créé un espace et un projet pour vous. Toutefois, vous devez tout de même configurer votre profil afin d'être prêt à commencer à travailler avec d'autres personnes.

**i** Tip

CodeCatalyst utilise des espaces pour regrouper les projets et les ressources. Lorsque vous vous inscrivez pour la première fois CodeCatalyst, il vous sera demandé de créer un espace ainsi qu'un projet.

Que vous vous inscriviez pour créer un espace et un projet ou que vous vous inscriviez pour accepter une invitation, vous créez un AWS Builder ID que vous utiliserez pour vous connecter CodeCatalyst. Pour créer un AWS Builder ID, vous devez fournir le nom complet, le mot de passe et l'adresse e-mail que vous utilisez pour vous connecter aux AWS applications. Vous utiliserez l'e-mail et le mot de passe pour vous connecter CodeCatalyst après ce point. Vous pouvez également utiliser ce AWS Builder ID pour vous connecter à d'autres applications qui utilisent les informations d'identification AWS Builder ID.

Dans CodeCatalyst et dans AWS Builder ID, un profil est généré en fonction de vos informations de connexion. Votre profil contient vos CodeCatalyst préférences en matière de langue et de paramètres de notification pour vos CodeCatalyst projets.

**i** Tip

Si vous rencontrez des problèmes lors de la création de votre CodeCatalyst profil Amazon, suivez les étapes indiquées sur cette page. Si vous avez besoin d'une aide supplémentaire, consultez [Problèmes lors de l'inscription](#).

## Rubriques

- [Inscrivez-vous pour créer votre premier espace et votre rôle de développeur](#)
- [Acceptation d'une invitation et création de votre identifiant AWS Builder](#)
- [Connectez-vous avec votre identifiant AWS Builder](#)
- [Accepter une invitation par e-mail pour vous connecter avec SSO](#)
- [Connectez-vous avec SSO](#)
- [Afficher tous les espaces et projets d'un utilisateur](#)
- [Afficher et gérer les CodeCatalyst profils](#)
- [Configuration pour utiliser AWS CLI le CodeCatalyst](#)

## Inscrivez-vous pour créer votre premier espace et votre rôle de développeur

Vous pouvez vous inscrire à Amazon CodeCatalyst sans être invité à accéder à un espace ou à un projet existant. Lorsque vous le ferez, vous créerez un espace et un projet après avoir créé votre identifiant AWS Builder. Dans le cadre de la création d'un espace, vous devrez en ajouter un à des Compte AWS fins de facturation.

**i** Tip

Si vous rencontrez des problèmes lors de la création de votre CodeCatalyst profil Amazon, suivez les étapes indiquées sur cette page. Si vous avez besoin d'une aide supplémentaire, consultez [Problèmes lors de l'inscription](#).

Voici un flux possible pour un utilisateur débutant CodeCatalyst sans invitation à un projet ou à un espace.

Mary Major est une développeuse intéressée CodeCatalyst et décide de l'essayer. Elle accède à la CodeCatalyst console et choisit l'option de s'inscrire et de créer un identifiant AWS Builder. Mary fournit une adresse e-mail et un mot de passe pour créer son identifiant AWS Builder. Elle pourra utiliser son identifiant AWS Builder pour se connecter à CodeCatalyst d'autres applications. Lorsqu'on lui demande de choisir un alias, elle indique le `MaryMajor` nom CodeCatalyst d'utilisateur qui sera affiché CodeCatalyst et que les autres membres du projet utiliseront pour @mention Mary.

Ensuite, Mary est automatiquement invitée à créer un espace. Dans le cadre de ce flux, Mary est invitée à Compte AWS associer un à l'espace qu'elle crée afin qu'elle puisse voir l'exemple de code lors de la création et du déploiement de son premier projet. Elle ajoute ces informations et crée son espace, où elle choisit l'option de créer un rôle de développement de prévisualisation pouvant être utilisé pour des projets dans son nouvel espace. Mary choisit de créer un projet, puis elle consulte une liste de plans de projets. Après avoir examiné les informations relatives aux plans disponibles, elle décide d'essayer le plan d'application Web moderne à trois niveaux pour son premier projet. Elle remplit les champs obligatoires et crée le projet. Dès que le projet est prêt, elle est redirigée vers une page de résumé du projet qui inclut les activités récentes ainsi que des liens vers le code du projet et le flux de travail qui crée et déploie automatiquement ce code. Elle explore à la fois le code et le flux de travail, notamment en consultant l'exemple d'application Web déployé. Appréciant ce qu'elle voit, elle décide d'inviter certains de ses collègues à participer au projet pour commencer à explorer CodeCatalyst.

Lorsqu'elle a un moment, Mary configure son identifiant AWS Builder pour se connecter à l'CodeCatalyst aide de l'authentification multifactorielle (MFA). Une fois la MFA configurée, Mary peut se connecter à CodeCatalyst l'aide d'une combinaison de son CodeCatalyst mot de passe et d'un code ou d'un jeton provenant d'une application d'authentification tierce approuvée.


## Création de votre premier espace et de vos premiers rôles IAM

Suivez ces étapes pour créer votre CodeCatalyst profil Amazon, créer un espace et ajouter un compte, un rôle de support et un rôle de développeur à votre espace.

La procédure finale crée et ajoute le rôle de développeur. Le rôle de développeur est un rôle AWS IAM qui permet à vos CodeCatalyst flux de travail d'accéder aux AWS ressources. Le rôle de développeur est un rôle de service utilisé pour gérer Services AWS et sera créé dans le compte connecté. Un rôle de service est un [rôle IAM](#) qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Le rôle aura un nom `CodeCatalystWorkflowDevelopmentRole-spaceName`.



Pour plus d'informations sur le rôle et la politique de rôle, consultez [Comprendre le rôle CodeCatalystWorkflowDevelopmentRole-\*spaceName\* du service](#).

 Note

En matière de sécurité, il est recommandé d'attribuer un accès administratif uniquement aux utilisateurs administratifs et aux développeurs qui doivent gérer l'accès aux AWS ressources de l'espace.

Avant de commencer, vous devez être prêt à fournir un Compte AWS identifiant pour un compte sur lequel vous disposez de privilèges administratifs. Ayez votre Compte AWS identifiant à 12 chiffres à portée de main. Pour plus d'informations sur la recherche de votre Compte AWS identifiant, consultez la section [Votre Compte AWS identifiant et son alias](#).

Pour vous inscrire en tant que nouvel utilisateur

1. Avant de commencer dans la CodeCatalyst console, ouvrez le AWS Management Console, puis assurez-vous que vous êtes connecté avec le même Compte AWS que celui que vous souhaitez utiliser pour créer votre espace.
2. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
3. Sur la page d'accueil, choisissez S'inscrire. La page Créez votre identifiant de AWS constructeur s'affiche. Votre AWS Builder ID est une identité que vous créez pour vous connecter. Ce n'est pas la même chose qu'un Compte AWS.
4. Dans Votre adresse e-mail, entrez l'adresse e-mail à laquelle vous souhaitez vous associer CodeCatalyst. Ensuite, sélectionnez Suivant.
5. Dans Votre nom, indiquez le prénom et le nom de famille que vous souhaitez afficher dans les applications où vous utilisez votre AWS Builder ID. Les espaces sont autorisés. Il s'agira du nom de votre profil AWS Builder ID, par exemple Mary Major. Vous pourrez modifier le nom ultérieurement.

Choisissez Suivant. La page de vérification par e-mail s'affiche.

6. Un code de vérification sera envoyé à l'adresse e-mail que vous avez spécifiée. Entrez ce code dans Code de vérification, puis choisissez Vérifier. Si vous ne recevez pas votre code au bout de 5 minutes et que vous ne le trouvez pas dans vos dossiers de spam ou de courrier indésirable, choisissez Renvoyer le code.

- Une fois votre code vérifié, entrez un mot de passe qui répond aux exigences des sections Mot de passe et Confirmation du mot de passe.


Cochez la case confirmant votre accord avec le contrat AWS client et les conditions de AWS service, puis choisissez Create AWS Builder ID.

- Sur la page Créez votre CodeCatalyst alias, entrez un alias que vous souhaitez utiliser pour votre identifiant utilisateur unique CodeCatalyst. Choisissez une version abrégée de votre nom sans espaces, telle que MaryMajor. CodeCatalyst Les autres utilisateurs l'utiliseront pour vous @mention dans les commentaires et les pull requests. Votre CodeCatalyst profil contiendra à la fois votre nom complet issu de votre identifiant AWS Builder et votre CodeCatalyst alias. Vous ne pourrez pas changer CodeCatalyst d'alias ultérieurement.

Votre nom complet et votre alias apparaîtront dans différentes zones de CodeCatalyst. Par exemple, le nom de votre profil s'affiche pour votre activité répertoriée dans le fil d'activités, mais les membres du projet utiliseront votre alias pour vous @mention.

Choisissez Suivant. La page est mise à jour pour afficher la section Créez votre CodeCatalyst espace.

- Dans Donnez un nom à votre espace, entrez le nom de votre espace. Vous ne pourrez pas le modifier ultérieurement.

 Note

Les noms des espaces doivent être uniques d'un côté à l'autre CodeCatalyst. Vous ne pouvez pas réutiliser les noms des espaces supprimés.

- Dans Région AWSle menu déroulant, choisissez la région dans laquelle vous souhaitez stocker votre espace et les données du projet. Vous ne pourrez pas le modifier ultérieurement.
- Choisissez Suivant. La page est mise à jour pour afficher la page permettant d'ajouter unCompte AWS. Ce compte sera utilisé comme compte de facturation pour l'espace.
- Dans Compte AWSID, saisissez l'identifiant à douze chiffres du compte que vous souhaitez connecter à votre espace.

Dans le jeton de vérification du AWS compte, copiez l'identifiant du jeton généré. Le jeton est automatiquement copié pour vous, mais vous souhaitez peut-être le stocker pendant que vous approuvez la demande de AWS connexion.

- Choisissez Accéder à la AWS console pour vérifier.


14. La page Verify Amazon CodeCatalyst Space s'ouvre dans leAWS Management Console. Voici la page Amazon CodeCatalyst Spaces. Vous devrez peut-être vous connecter pour accéder à la page.

Dans leAWS Management Console, assurez-vous de choisir le même Région AWS endroit où vous souhaitez créer votre espace.

Pour accéder directement à la page, connectez-vous aux Amazon CodeCatalyst Spaces AWS Management Console à l'adresse <https://console.aws.amazon.com/codecatalyst/home/>.

Le champ du jeton de vérification dans le AWS Management Console est automatiquement rempli avec le jeton généré dans CodeCatalyst.

15. (Facultatif) Sous Niveaux payants autorisés, choisissez Autoriser les niveaux payants (Standard, Enterprise) pour activer les niveaux payants pour votre compte de facturation.

 Note

Cela ne fait pas passer le niveau de facturation à un niveau payant. Cependant, cela configure le Compte AWS afin que vous puissiez modifier le niveau de facturation de votre espace à tout moment. CodeCatalyst Vous pouvez activer les niveaux payants à tout moment. Sans cette modification, l'espace ne peut utiliser que le niveau gratuit.

16. Choisissez Vérifier l'espace.

Un message de confirmation du compte s'affiche pour indiquer que le compte a été ajouté à l'espace.


17. Restez sur la page Verify Amazon CodeCatalyst Space. Cliquez sur le lien suivant : Pour ajouter des rôles IAM pour cet espace, consultez les détails de l'espace.

La page des connexions avec les détails de CodeCatalyst l'espace s'ouvre dans leAWS Management Console. Voici la page Amazon CodeCatalyst Spaces. Il se peut que vous deviez vous connecter pour accéder à la page.

18. Retournez à la CodeCatalyst page, puis choisissez Next.
19. Un message d'état s'affiche lors de la création de votre espace. Lorsque l'espace est créé, CodeCatalyst le message suivant s'affiche : Votre espace est prêt. La dernière étape consiste à créer un projet. . Vous pouvez effectuer l'une des actions suivantes :

- Choisissez Skip pour le moment.

- Choisissez Créer votre premier projet pour votre espace. Pour un didacticiel qui explique comment créer un projet à l'aide d'un plan, voir [Tutoriel : Création d'un projet avec le plan d'application Web moderne à trois niveaux](#)

 Note


Si une erreur d'autorisation ou une bannière s'affiche, actualisez la page et réessayez de l'afficher.

Pour créer et ajouter CodeCatalyst CodeCatalystWorkflowDevelopmentRole-*spaceName*

1. Avant de commencer dans la CodeCatalyst console, ouvrez leAWS Management Console, puis assurez-vous que vous êtes connecté avec le même identifiant Compte AWS pour votre espace.
2. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
3. Accédez à votre CodeCatalyst espace. Choisissez Settings (Paramètres), puis Comptes AWS.
4. Choisissez le lien correspondant à l'Compte AWSendroit où vous souhaitez créer le rôle. La page Compte AWSde détails s'affiche.
5. Choisissez Gérer les rôles à partir de AWS Management Console.

La page Ajouter un rôle IAM à Amazon CodeCatalyst Space s'ouvre dans leAWS Management Console. Voici la page Amazon CodeCatalyst Spaces. Il se peut que vous deviez vous connecter pour accéder à la page.

6. Choisissez Créer un rôle d'administrateur de CodeCatalyst développement dans IAM. Cette option crée un rôle de service qui contient la politique d'autorisation et la politique de confiance pour le rôle de développement. Le rôle aura un nomCodeCatalystWorkflowDevelopmentRole-*spaceName*. Pour plus d'informations sur le rôle et la politique de rôle, consultez[Comprendre le rôle CodeCatalystWorkflowDevelopmentRole-\*spaceName\*du service](#).

 Note

Ce rôle est uniquement recommandé pour les comptes de développeur et utilise la politique AdministratorAccess AWS gérée, ce qui lui donne un accès complet pour créer de nouvelles politiques et ressources dans ce cadreCompte AWS.

7. Choisissez Créer un rôle de développement.
8. Sur la page des connexions, sous Rôles IAM disponibles pour CodeCatalyst, consultez le `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle dans la liste des rôles IAM ajoutés à votre compte.
9. Pour retourner dans votre espace, choisissez Go to Amazon CodeCatalyst.

#### Pour créer et ajouter CodeCatalyst AWSRoleForCodeCatalystSupport

1. Avant de commencer dans la CodeCatalyst console, ouvrez leAWS Management Console, puis assurez-vous que vous êtes connecté avec le même identifiant Compte AWS pour votre espace.
2. Accédez à votre CodeCatalyst espace. Choisissez Settings (Paramètres), puis Comptes AWS.
3. Choisissez le lien correspondant à l'Compte AWSendroit où vous souhaitez créer le rôle. La page Compte AWSde détails s'affiche.
4. Choisissez Gérer les rôles à partir de AWS Management Console.

La page Ajouter un rôle IAM à Amazon CodeCatalyst Space s'ouvre dans leAWS Management Console. Voici la page Amazon CodeCatalyst Spaces. Vous devrez peut-être vous connecter pour accéder à la page.

5. Sous Détails de CodeCatalyst l'espace, choisissez Ajouter un rôle de CodeCatalyst support. Cette option crée un rôle de service qui contient la politique d'autorisation et la politique de confiance pour le rôle de développement préliminaire. Le rôle portera un nom `AWSRoleForCodeCatalystSupport`avec un identifiant unique ajouté. Pour plus d'informations sur le rôle et la politique de rôle, consultez[Comprendre le rôle AWSRoleForCodeCatalystSupportdu service](#).
6. Sur la page Ajouter un rôle pour le CodeCatalyst support, laissez la valeur par défaut sélectionnée, puis choisissez Créer un rôle.
7. Sous Rôles IAM disponibles pour CodeCatalyst, consultez le `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle dans la liste des rôles IAM ajoutés à votre compte.
8. Pour retourner dans votre espace, choisissez Go to Amazon CodeCatalyst.

Après avoir créé votre identifiant AWS Builder, créé votre premier espace et ajouté un compte, vous pouvez créer un projet. Pour de plus amples informations, veuillez consulter [Création d'un projet sur Amazon CodeCatalyst](#). Si c'est la première fois que vous l'utilisez CodeCatalyst, nous vous

suggérons de commencer par [Tutoriel : Création d'un projet avec le plan d'application Web moderne à trois niveaux](#).

## Acceptation d'une invitation et création de votre identifiant AWS Builder

Vous pouvez vous inscrire à Amazon dans CodeCatalyst le cadre de l'acceptation d'une invitation à un projet ou à un espace. Dans le cadre de l'acceptation de l'invitation, vous serez invité à créer un AWS Builder ID. Vous utiliserez votre identifiant AWS Builder pour accéder aux ressources dans CodeCatalyst.

### Tip

Si vous avez besoin d'une aide supplémentaire, consultez [Problèmes lors de l'inscription](#).

Voici un flux possible pour un utilisateur débutant par une invitation à un projet ou à un espace. CodeCatalyst

Saanvi Sarkar est un développeur qui a reçu une invitation à rejoindre un CodeCatalyst projet en tant qu'administrateur de projet. Saanvi accepte l'invitation, ce qui ouvre la page de connexion pour CodeCatalyst. Elle choisit de s'inscrire et fournit une adresse e-mail et un mot de passe pour créer son identifiant AWS Builder. Saanvi pourra utiliser son identifiant AWS Builder pour se connecter à d'autres CodeCatalyst applications. Plus tard, elle pourra modifier son profil pour changer son adresse e-mail ou son mot de passe de connexion. Lorsqu'on lui demande de choisir un alias, Saanvi indique SaanviSarkar l' CodeCatalyst alias qui sera affiché CodeCatalyst et que les autres membres du projet utiliseront pour @mention Saanvi. Après son inscription, Saanvi pourra également utiliser ses informations de connexion pour d'autres applications utilisant les informations d'identification AWS Builder ID.

Une fois son inscription terminée, Saanvi rejoint automatiquement le CodeCatalyst projet et l'espace spécifiés dans l'invitation. L'invitation fournit également des autorisations prédéterminées pour ses rôles dans le projet et dans l'espace. Dans les paramètres du projet, l'alias de Saanvi apparaît dans la liste des membres avec le rôle qui lui a été attribué dans le projet. Pour travailler avec les référentiels sources dans CodeCatalyst, Saanvi prend un moment pour créer un jeton d'accès personnel (PAT). Le PAT sera utilisé CodeCatalyst pour l'authentification lors de modifications de source ou d'actions nécessitant un jeton d'authentification.

Lorsque Saanvi travaille sur un projet, son alias est répertorié dans le journal des activités professionnelles du projet. Les problèmes et les commentaires de Saanvi afficheront son pseudonyme, ce qui permettra aux autres membres du projet de l'associer à @mention dans leurs réponses. Pour @mention, un autre membre du projet, Saanvi recherche son alias sur son CodeCatalyst profil.

Lorsqu'elle a un moment, Saanvi configure son identifiant AWS Builder pour se connecter à l'CodeCatalyst aide de l'authentification multifactorielle (MFA). Une fois le MFA configuré, Saanvi peut se connecter à CodeCatalyst l'aide d'une combinaison de son CodeCatalyst mot de passe et d'un code ou d'un jeton provenant d'une application d'authentification tierce approuvée.

## Acceptation d'une invitation et création d'un identifiant AWS Builder

Lorsque vous êtes invité à participer à un projet ou à un espace sur Amazon CodeCatalyst, vous recevez un e-mail de [notify@codecatalyst.aws](mailto:notify@codecatalyst.aws) vous demandant d'accepter l'invitation. Si vous possédez déjà un identifiant AWS Builder et que vous êtes connecté CodeCatalyst, le fait de choisir Accepter l'invitation ouvrira automatiquement le projet ou l'espace dans un onglet du navigateur. Si vous n'êtes pas connecté à la console mais que vous possédez un identifiant AWS Builder, vous serez redirigé vers la page de connexion. Pour en savoir plus, consultez [Connectez-vous avec votre identifiant AWS Builder](#).

Si vous n'avez pas d'identifiant de AWS constructeur, cliquez sur Accepter l'invitation pour accéder à la page de connexion, où vous devez choisir l'option de création de votre identifiant de AWS constructeur.

Pour accepter une invitation et créer un identifiant AWS Builder

1. Dans l'e-mail d'invitation, choisissez Accepter l'invitation.
2. Sur la page de connexion, sélectionnez Non inscrit ? Créez votre identifiant AWS Builder.

### Tip

Votre AWS Builder ID est une identité que vous créez pour vous connecter. Ce n'est pas la même chose qu'un Compte AWS.

3. Sur la page Créez votre identifiant de AWS constructeur, dans Adresse e-mail, entrez l'adresse e-mail que vous souhaitez utiliser pour votre identifiant de AWS constructeur.

Dans Votre nom, indiquez le prénom et le nom de famille que vous souhaitez afficher dans les applications où vous utilisez votre AWS Builder ID. Les espaces sont autorisés. Il s'agira du nom de votre profil AWS Builder ID, par exemple Mary Major. Vous pourrez modifier le nom ultérieurement.

Choisissez Suivant.

Un code de vérification sera envoyé à l'adresse e-mail que vous avez spécifiée. Entrez ce code dans Code de vérification, puis choisissez Vérifier. Si vous ne recevez pas votre code au bout de 5 minutes et que vous ne le trouvez pas dans vos dossiers de spam ou de courrier indésirable, choisissez Renvoyer le code.

4. Une fois votre code vérifié, entrez un mot de passe qui répond aux exigences des sections Mot de passe et Confirmation du mot de passe.
5. Choisissez Create AWS Builder ID.
6. Sur la page Créez votre alias, entrez un alias que vous souhaitez utiliser pour votre identifiant utilisateur unique CodeCatalyst. Choisissez une version abrégée de votre nom sans espaces, telle que MaryMajor. CodeCatalyst Les autres utilisateurs l'utiliseront pour vous @mention dans les commentaires et les pull requests. Votre CodeCatalyst profil contiendra à la fois votre nom complet issu de votre identifiant AWS Builder et votre CodeCatalyst alias. Vous ne pouvez pas changer CodeCatalyst d'alias.

Votre nom complet et votre alias s'afficheront dans différentes zones de CodeCatalyst. Par exemple, le nom de votre profil s'affiche pour votre activité répertoriée dans le fil d'activités, mais les membres du projet utiliseront votre alias pour vous @mention.

Choisissez Créer un alias. Vous serez redirigé vers le projet ou l'espace auquel vous avez été invité.

## Connectez-vous avec votre identifiant AWS Builder

Suivez ces étapes pour vous connecter à votre CodeCatalyst profil Amazon.

### Note

Avez-vous déjà enregistré un appareil pour l'authentification multifactorielle (MFA) ? Nous vous recommandons vivement de configurer le MFA sur Amazon CodeCatalyst pour



renforcer votre sécurité. Pour en savoir plus, consultez [Comment enregistrer un appareil pour une utilisation avec l'authentification multifactorielle](#).

Pour vous connecter avec votre identifiant AWS Builder

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Entrez votre adresse e-mail. Vous pouvez éventuellement choisir Enregistrer mon adresse e-mail si vous souhaitez enregistrer votre adresse e-mail pour de futures connexions. Choisissez Continuer.
3. Entrez votre mot de passe. Choisissez Sign in (Connexion). Si vous ne vous souvenez pas de votre mot de passe, suivez les étapes décrites dans [J'ai oublié mon mot de passe](#).

## Appareils fiables

Une fois que vous avez sélectionné l'option Ceci est un appareil fiable sur la page de connexion, Amazon CodeCatalyst considère toutes les futures connexions à partir de cet appareil comme autorisées. Amazon ne CodeCatalyst proposera pas d'option permettant de saisir un code MFA tant que vous utiliserez cet appareil fiable. Certaines exceptions incluent la connexion depuis un nouveau navigateur ou lorsqu'une adresse IP inconnue a été attribuée à votre appareil.

## Accepter une invitation par e-mail pour vous connecter avec SSO

Les utilisateurs ajoutés à un espace prenant en charge la fédération d'identité reçoivent un e-mail contenant le lien du portail de connexion et les informations de configuration. Suivez ces étapes pour accepter une invitation et vous connecter avec SSO.

Pour vous connecter avec votre identifiant AWS Builder à la place, consultez [Connectez-vous avec votre identifiant AWS Builder](#).

Pour accepter et se connecter avec SSO

1. Cliquez sur le bouton dans l'e-mail qui accepte la demande. Utilisez le lien fourni dans l'e-mail pour accéder au portail de connexion de l'espace associé à votre entreprise.
2. Dans Nom d'utilisateur et mot de passe, entrez vos informations d'identification.
3. Choisissez Sign in (Connexion).

## Connectez-vous avec SSO

Suivez ces étapes pour utiliser l'authentification unique pour vous connecter à Amazon CodeCatalyst.

Pour vous connecter avec votre identifiant AWS Builder à la place, consultez [Connectez-vous avec votre identifiant AWS Builder](#).

Pour vous connecter avec SSO

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Sous Choisir une option de connexion, choisissez Utiliser l'authentification unique (SSO).
3. Dans Nom de l'application AWS Identity Center, entrez le nom de l'application fourni par votre administrateur de fédération d'identités.
4. Choisissez Continuer vers IAM Identity Center.

## Afficher tous les espaces et projets d'un utilisateur

Vous pouvez consulter la liste de vos espaces et projets sur la page d'accueil de l'utilisateur. La page d'accueil de l'utilisateur affiche une liste de chaque espace auquel appartient l'utilisateur, le rôle de l'utilisateur dans cet espace, tel que l'administrateur de l'espace, et les projets dans chaque espace auquel l'utilisateur est membre.

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans le navigateur, entrez l'adresse suivante : <https://codecatalyst.aws/home>

The screenshot displays the Amazon CodeCatalyst interface. At the top, there is a header with the text "s spaces (9)" and two buttons: "Manage AWS Builder ID" and "Create space". Below the header is a search bar labeled "Filter spaces". The main content area is divided into three sections, each representing a different space:

- EnchantedForest**: Space administrator. A "Create project" button is visible. Below this, there are two project cards: "WildWaves" and "FracturedFairyTales". Each card lists "Pull requests", "Workflows", "Source repositories", and "Environments".
- org**: Space administrator. A "Create project" button is visible. Below this, there are four project cards: "migration", "test", and "12597". Each card lists "Pull requests", "Workflows", "Source repositories", and "Environments".
- AnyCompany**: Space member. A "Create project" button is visible. Below this, there is one project card: "newproject", which lists "Pull requests", "Workflows", "Source repositories", and "Environments".

3. Choisissez l'espace ou le projet que vous souhaitez ouvrir. Si vous ne voyez pas l'espace ou le projet que vous vous attendiez à voir, vous devrez peut-être vous connecter en tant qu'utilisateur différent.

## Afficher et gérer les CodeCatalyst profils

Vous pouvez consulter les profils des utilisateurs sur Amazon CodeCatalyst pour obtenir des informations telles que les adresses e-mail et les CodeCatalyst alias. Vous pouvez également mettre

à jour votre profil et votre identifiant AWS Builder. Si vous oubliez votre mot de passe, vous pouvez demander une réinitialisation du mot de passe.

## Consulter votre CodeCatalyst profil

Lors de l'inscription, vous fournissez des informations qui seront utilisées comme informations d'identification pour vous connecter à Amazon CodeCatalyst et qui seront gérées dans votre profil. Cela inclut votre nom, votre surnom et l'adresse e-mail que vous utilisez pour vous connecter CodeCatalyst.

### Note

Le surnom AWS Builder ID n'est pas votre CodeCatalyst alias. Vous avez sélectionné votre CodeCatalyst alias lors de votre inscription.

Pour consulter votre CodeCatalyst profil

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. En haut à droite, cliquez sur la flèche à côté de l'icône portant votre première initiale, puis sélectionnez Mes paramètres. La page CodeCatalyst Mes paramètres s'ouvre.
3. Pour mettre à jour votre adresse e-mail ou votre mot de passe AWS Builder ID, ou pour configurer le MFA, choisissez Manage AWS Builder ID. La page AWS Builder ID s'ouvre.

## Consulter le CodeCatalyst profil d'un autre utilisateur

Pour consulter le CodeCatalyst profil d'un autre utilisateur

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la barre de navigation latérale, choisissez Paramètres du projet. Choisissez l'onglet Membres. Consultez la liste des membres de votre CodeCatalyst projet.
3. Choisissez le nom du membre que vous souhaitez rechercher ou @mention. La page Mes paramètres indique l'alias, l'adresse e-mail et le nom complet de l'utilisateur. Utilisez l'CodeCatalyst alias to @mention pour les membres du projet.

**Note**

Le surnom AWS Builder ID d'un utilisateur n'est pas son CodeCatalyst alias. Ils ont sélectionné leur CodeCatalyst alias lors de leur inscription.

Pour consulter le profil d'un autre utilisateur dans votre projet, choisissez son nom dans la liste.

## Mettre à jour votre profil

Dans CodeCatalyst, votre profil comprend des informations personnelles gérées par AWSBuilder ID et des paramètres gérés dans CodeCatalyst.

- Le nom complet, l'adresse e-mail et le mot de passe de votre profil sont gérés par AWSBuilder ID. Vous avez saisi ces informations lors de votre inscription. Lorsque vous configurez le MFA pour utiliser une application d'authentification pour la connexion aux applications, vous CodeCatalyst accédez à la page Builder ID. AWS
- CodeCatalyst les paramètres de votre jeton d'accès personnel (PAT), CodeCatalyst des notifications et des préférences linguistiques sont gérés sur la page Mes paramètres de CodeCatalyst. Pour en savoir plus, consultez [Gestion des jetons d'accès personnels sur Amazon CodeCatalyst](#).

**Note**


Vous pouvez mettre à jour le nom complet (nom CodeCatalyst d'affichage) et le prénom de votre AWS Builder ID. Cependant, vous ne pouvez pas modifier votre CodeCatalyst alias.

## Mettre à jour votre identifiant AWS Builder ou votre adresse e-mail

Pour mettre à jour votre adresse ID de constructeur AWS ou votre adresse e-mail


1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. En haut à droite, cliquez sur la flèche à côté de l'icône portant votre première initiale, puis sélectionnez Mes paramètres. La page CodeCatalyst Mes paramètres s'ouvre.
3. Sur la page de profil, choisissez Manage AWS Builder ID. La page AWSBuilder ID s'ouvre.

4. Sur le côté gauche de la page, sélectionnez Mes informations.
5. Sous Informations de profil, choisissez Modifier pour mettre à jour votre nom ou surnom. Si vous n'avez pas indiqué de surnom, le champ Surnom indique le prénom figurant dans le nom complet. Ce n'est pas ton CodeCatalyst pseudonyme.

 Note

Cela met à jour le nom complet et le prénom du AWS Builder ID. Cela ne met pas à jour votre CodeCatalyst alias.

Sous Informations de contact, choisissez Modifier pour mettre à jour votre adresse e-mail.

 Note

Cela met à jour l'adresse e-mail que vous utiliserez pour vous connecter CodeCatalyst.

## Modification de votre CodeCatalyst mot de passe

Pour modifier votre CodeCatalyst mot de passe

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. En haut à droite, cliquez sur la flèche à côté de l'icône portant votre première initiale, puis sélectionnez Profil utilisateur. La page CodeCatalyst Mes paramètres s'ouvre.
3. Sur la page de profil, choisissez Manage AWS Builder ID. La page AWS Builder ID s'ouvre.
4. Sur le côté gauche de la page, sélectionnez Sécurité.
5. Choisissez Changer le mot de passe et suivez les instructions.

## Configuration pour utiliser AWS CLI le CodeCatalyst

La CodeCatalyst console Amazon est l'endroit où vous travaillerez sur la plupart de vos tâches quotidiennes. Cependant, vous souhaitez peut-être configurer le AWS CLI lorsque vous travaillez avec des environnements de développement, des jetons d'accès personnels ou des journaux d'événements CodeCatalyst. Vous devez installer AWS CLI et configurer un profil avant de pouvoir l'utiliser avec CodeCatalyst.

## Pour configurer le AWS CLI pour CodeCatalyst

1. Installez la dernière version du AWS CLI. Si une version est déjà AWS CLI installée, assurez-vous qu'elle est récente et inclut des commandes pour CodeCatalyst, et mettez-la à jour si nécessaire. Pour vérifier que vous disposez d'une version qui inclut des CodeCatalyst commandes, ouvrez une invite de commande et exécutez la commande suivante :

```
aws codecatalyst help
```

Si vous voyez une liste de CodeCatalyst commandes, c'est que vous disposez d'une version qui les prend en charge CodeCatalyst. Si la commande n'est pas reconnue, mettez AWS CLI à jour votre version de. Pour plus d'informations, voir [Installation ou mise à jour de la dernière version du AWS CLI dans le](#) guide de AWS Command Line Interface l'utilisateur.

2. Exécutez la `aws configure` commande pour créer un profil si vous n'en avez pas ou si vous souhaitez utiliser un profil nommé spécifiquement pour CodeCatalyst. Nous vous recommandons de créer un profil nommé à utiliser spécifiquement CodeCatalyst, mais vous pouvez également utiliser le profil par défaut. Pour plus d'informations, consultez la section [Principes de base de la configuration](#).
3. Modifiez le config fichier du profil pour ajouter une section à laquelle vous pouvez vous connecter CodeCatalyst , comme suit. Le config fichier se trouve sous `~/.aws/config` Linux ou macOS, ou `C:\Users\USERNAME\.aws\config` sous Windows.

```
[profile codecatalyst]
region = us-west-2
sso_session = codecatalyst

[sso-session codecatalyst]
sso_region = us-east-1
sso_start_url = https://view.awsapps.com/start
sso_registration_scopes = codecatalyst:read_write
```

4. Sauvegardez le fichier.
5. Avant de tenter d'exécuter des CodeCatalyst commandes, ouvrez un nouveau terminal ou une nouvelle invite de commande et exécutez la commande suivante pour demander et récupérer les informations d'identification nécessaires à l'exécution `aws codecatalyst` des commandes. `codecatalyst` Remplacez-le par le nom de votre profil si nécessaire.

```
aws sso login --profile codecatalyst
```

---

Pour consulter des exemples de codecatalyst commandes, consultez les rubriques suivantes :

- [Gestion des jetons d'accès personnels sur Amazon CodeCatalyst](#)
- [Accès aux événements enregistrés dans CodeCatalyst](#)



# Tutoriels de mise en route

Amazon CodeCatalyst propose différents modèles pour vous aider à démarrer vos projets. Vous pouvez également choisir de commencer par un projet vide et d'y ajouter des ressources. Suivez les étapes décrites dans ces didacticiels pour découvrir certaines des manières dont vous pouvez travailler CodeCatalyst.

Si c'est la première fois que vous l'utilisez CodeCatalyst, nous vous suggérons de commencer par [Tutoriel : Création d'un projet avec le plan d'application Web moderne à trois niveaux](#).

## Note

Pour suivre ces didacticiels, vous devez d'abord terminer la configuration. Pour plus d'informations, consultez [Configuration CodeCatalyst](#).

## Rubriques

- [Tutoriel : Création d'un projet avec le plan d'application Web moderne à trois niveaux](#)
- [Tutoriel : Commencer avec un projet vide et ajouter manuellement des ressources](#)
- [Tutoriel : Utiliser les fonctionnalités CodeCatalyst génératives de l'IA pour accélérer votre travail de développement](#)
- [Tutoriel : Création d'une application complète avec des plans PDK composables](#)

Pour des didacticiels supplémentaires axés sur des domaines fonctionnels spécifiques dans CodeCatalyst, voir :

- [Commencer à utiliser les notifications Slack](#)
- [Commencer à utiliser les référentiels de CodeCatalyst sources et le plan d'application d'une seule page](#)
- [Commencer à utiliser les flux de travail dans CodeCatalyst](#)
- [Commencer à utiliser des plans personnalisés](#)
- [Commencez avec le guide du développeur CodeCatalyst d'Amazon Action](#)

Pour des didacticiels approfondis, voir :

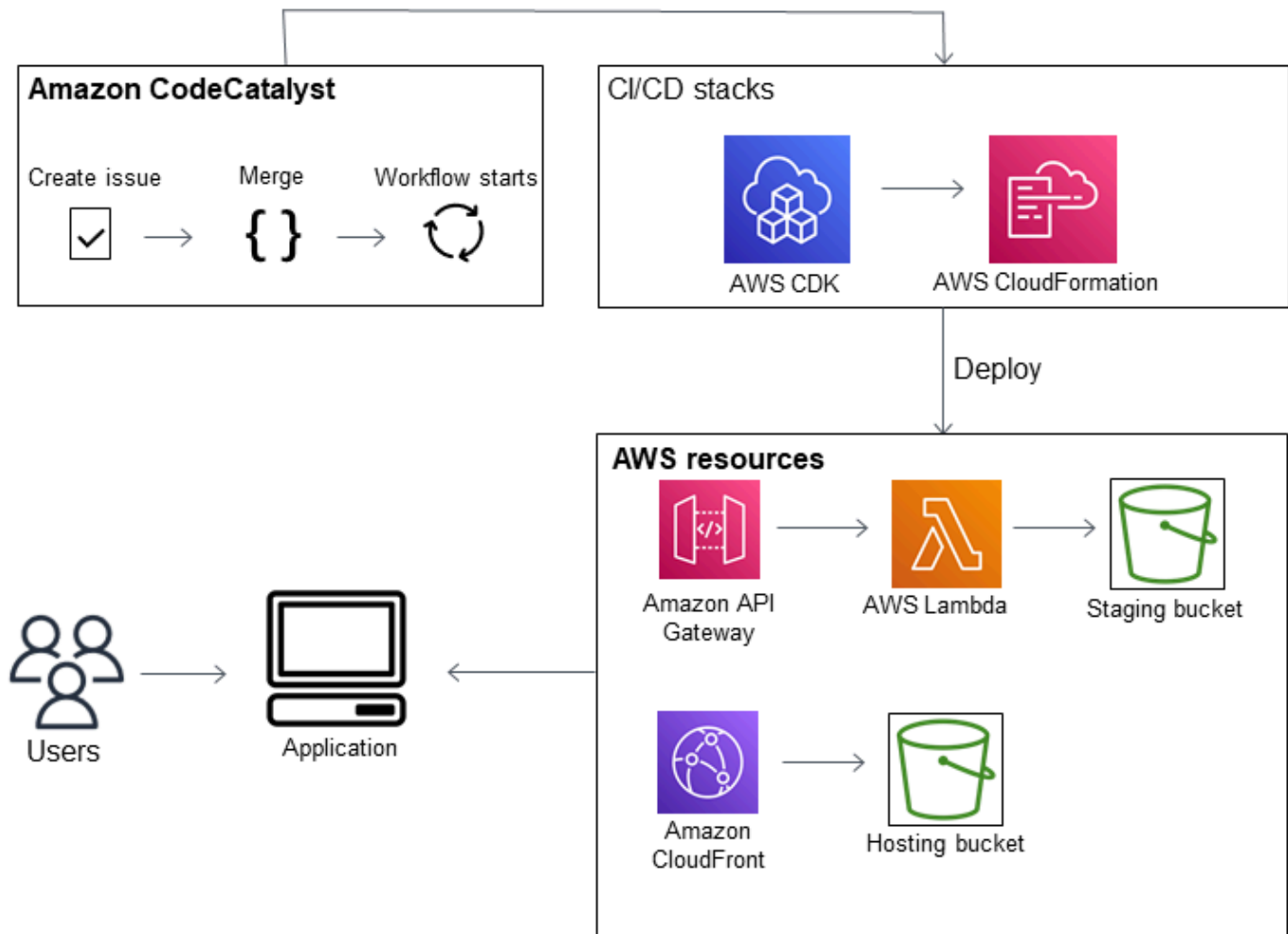
- [Tutoriel : télécharger des artefacts sur Amazon S3](#)
- [Tutoriel : Déployer une application sans serveur à l'aide de AWS CloudFormation](#)
- [Tutoriel : Déploiement d'une application sur Amazon ECS](#)
- [Tutoriel : Déploiement d'une application sur Amazon EKS](#)
- [Tutoriel : code Lint à l'aide d'une action GitHub](#)
- [Tutoriel : Création et mise à jour d'une application React](#)

## Tutoriel : Création d'un projet avec le plan d'application Web moderne à trois niveaux

Vous pouvez vous lancer plus rapidement dans le développement de logiciels en créant un projet à l'aide d'un plan. Un projet créé à l'aide d'un plan inclut les ressources dont vous avez besoin, notamment un référentiel de sources pour gérer votre code et un flux de travail pour créer et déployer l'application. Dans ce didacticiel, nous allons vous expliquer comment utiliser le modèle d'application Web moderne à trois niveaux pour créer un projet sur Amazon. CodeCatalyst Le didacticiel inclut également la visualisation de l'exemple déployé, l'invitation d'autres utilisateurs à travailler dessus et la modification du code avec des pull requests qui sont automatiquement créées et déployées sur les ressources du connecté Compte AWS lorsque la pull request est fusionnée. Lorsque CodeCatalyst crée votre projet à l'aide de rapports, de flux d'activités et d'autres outils, votre plan crée AWS des ressources Compte AWS associées à votre projet. Vos fichiers de plan vous permettent de créer et de tester un exemple d'application moderne et de le déployer sur l'infrastructure du AWS Cloud.

L'illustration suivante montre comment les outils intégrés CodeCatalyst sont utilisés pour créer un problème afin de suivre, de fusionner et de créer automatiquement le changement, puis de démarrer un flux de travail dans le CodeCatalyst projet qui exécute des actions pour autoriser AWS CDK et AWS CloudFormation approvisionner votre infrastructure.

Les actions génèrent des ressources dans le système associé Compte AWS et déploient votre application vers une AWS Lambda fonction sans serveur dotée d'un point de terminaison API Gateway. L' AWS Cloud Development Kit (AWS CDK) action convertit une ou plusieurs AWS CDK piles en AWS CloudFormation modèles et déploie des piles sur votre. Compte AWS Les ressources de vos piles incluent les CloudFront ressources Amazon pour distribuer du contenu Web dynamique, une instance Amazon DynamoDB pour les données de votre application, ainsi que les rôles et les politiques qui prennent en charge l'application déployée.



Lorsque vous créez un projet avec le plan d'application Web moderne à trois niveaux, votre projet est créé avec les ressources suivantes :

Dans le CodeCatalyst projet :

- Un [référentiel source](#) avec un exemple de code et un flux de travail YAML
- Un [flux de travail](#) qui crée et déploie l'exemple de code chaque fois qu'une modification est apportée à la branche par défaut
- Un tableau des problèmes et un carnet de dossiers que vous pouvez utiliser pour planifier et suivre le travail
- Une suite de rapports de test avec des rapports automatisés inclus dans l'exemple de code

Dans le document associé Compte AWS :

- Trois AWS CloudFormation piles qui créent les ressources nécessaires à l'application.

Pour plus de détails sur les ressources qui seront créées dans AWS et dans CodeCatalyst le cadre de ce didacticiel, voir [Référence](#).

#### Note

Les ressources et les exemples inclus dans un projet dépendent du plan que vous sélectionnez. Amazon CodeCatalyst propose plusieurs plans de projet qui définissent les ressources liées au langage ou au cadre défini. Pour en savoir plus sur les plans, voir [Référence du plan de projet](#).

## Rubriques

- [Prérequis](#)
- [Étape 1 : Création du projet d'application Web moderne à trois niveaux](#)
- [Étape 2 : Invitez quelqu'un à rejoindre votre projet](#)
- [Étape 3 : créer des problèmes sur lesquels collaborer et suivre le travail](#)
- [Étape 4 : Afficher votre référentiel source](#)
- [Étape 5 : Création d'un environnement de développement avec une branche de test et modification rapide du code](#)
- [Étape 6 : Afficher le flux de travail qui crée l'application moderne](#)
- [Étape 7 : demandez à d'autres personnes de revoir vos modifications](#)
- [Étape 8 : Clôturer le problème](#)
- [Nettoyage des ressources](#)
- [Référence](#)

## Prérequis

Pour créer un projet d'application moderne dans ce didacticiel, vous devez avoir effectué les tâches [Configuration CodeCatalyst](#) suivantes :

- Disposez d'un identifiant AWS Builder pour vous connecter à CodeCatalyst.

- Appartenez à un espace et le rôle d'administrateur de l'espace ou d'utilisateur avancé vous est attribué dans cet espace. Pour plus d'informations, consultez [Création d'un espace prenant en charge les utilisateurs de AWS Builder ID](#), [Gestion des utilisateurs de l'espace](#) et [Rôle d'administrateur de l'espace](#).
- Compte AWS Associez-vous à votre espace et détenez le rôle IAM que vous avez créé lors de votre inscription. Par exemple, lors de l'inscription, vous avez la possibilité de choisir de créer un rôle de service avec une politique de rôle appelée politique de CodeCatalystWorkflowDevelopmentRole-*spaceName*rôle. Le rôle portera un nom CodeCatalystWorkflowDevelopmentRole-*spaceName* avec un identifiant unique ajouté. Pour plus d'informations sur le rôle et la politique de rôle, consultez [Comprendre le rôle CodeCatalystWorkflowDevelopmentRole-\*spaceName\*du service](#). Pour connaître les étapes de création du rôle, consultez [Création du CodeCatalystWorkflowDevelopmentRole-\*spaceName\*rôle pour votre compte et votre espace](#).

## Étape 1 : Création du projet d'application Web moderne à trois niveaux

Une fois que vous l'avez créé, c'est dans votre projet que vous allez développer et tester le code, coordonner les tâches de développement et consulter les indicateurs du projet. Votre projet contient également vos outils et ressources de développement.

Dans ce didacticiel, vous allez utiliser le modèle d'application Web moderne à trois niveaux pour créer une application interactive. Le flux de travail créé et exécuté automatiquement dans le cadre de votre projet créera et déploiera l'application. Le flux de travail ne s'exécute correctement qu'une fois que tous les rôles et informations de compte ont été configurés pour votre espace. Une fois le flux de travail exécuté avec succès, vous pouvez accéder à l'URL du point de terminaison pour voir l'application.

Pour créer un projet à l'aide d'un plan

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la CodeCatalyst console, accédez à l'espace dans lequel vous souhaitez créer un projet.
3. Sélectionnez Create a project (Créer un projet).
4. Choisissez Commencer par un plan.
5. Dans la barre de recherche, saisissez **modern**.
6. Sélectionnez le plan d'application Web moderne à trois niveaux, puis cliquez sur Suivant.
7. Dans Nommer votre projet, entrez un nom de projet. Par exemple :

## MyExampleProject.

### Note

Le nom doit être unique dans votre espace.

8. Dans Compte, choisissez celui Compte AWS que vous avez ajouté lors de votre inscription. Le plan installera des ressources dans ce compte.
9. Dans Rôle de déploiement, choisissez le rôle que vous avez ajouté lors de votre inscription. Par exemple, sélectionnez `CodeCatalystWorkflowDevelopmentRole-spaceName`.  
  
Si aucun rôle n'est répertorié, ajoutez-en un. Pour ajouter un rôle, choisissez Ajouter un rôle IAM et ajoutez le rôle à votre Compte AWS. Pour plus d'informations, consultez [Administration Comptes AWS d'un espace](#).
10. Dans Compute platform, choisissez Lambda.
11. Dans l'option d'hébergement frontal, choisissez Amplify Hosting. Pour plus d'informations AWS Amplify, voir [Qu'est-ce que l' AWS Amplify hébergement ?](#) dans le guide de AWS Amplify l'utilisateur.
12. Dans Région de déploiement, entrez le code de région dans Région AWS lequel vous souhaitez que le plan pilote déploie l'application Mysfits et les ressources de support. Pour obtenir la liste des codes de région, consultez la section [Points de terminaison régionaux](#) dans le Références générales AWS.
13. Dans Nom de l'application, laissez la valeur par défaut `demysfitsstring`.
14. (Facultatif) Sous Générer un aperçu du projet, choisissez Afficher le code pour prévisualiser les fichiers source que le plan va installer. Choisissez Afficher le flux de travail pour prévisualiser les fichiers de définition du flux de travail CI/CD que le plan va installer. L'aperçu est mis à jour de manière dynamique en fonction de vos sélections.
15. Sélectionnez Create a project (Créer un projet).

Le flux de travail du projet démarre dès que vous créez le projet. Il faudra un peu de temps pour terminer la création et le déploiement du code. En attendant, allez-y et invitez quelqu'un d'autre à participer à votre projet.

## Étape 2 : Invitez quelqu'un à rejoindre votre projet

Maintenant que vous avez configuré votre projet, invitez d'autres personnes à travailler avec vous.

Pour inviter quelqu'un à rejoindre votre projet

1. Accédez au projet auquel vous souhaitez inviter des utilisateurs.
2. Dans le volet de navigation, sélectionnez Paramètres du projet.
3. Dans l'onglet Membres, choisissez Inviter.
4. Entrez les adresses e-mail des personnes que vous souhaitez inviter en tant qu'utilisateurs de votre projet. Vous pouvez saisir plusieurs adresses e-mail séparées par un espace ou une virgule. Vous pouvez également choisir parmi les membres de votre espace qui ne sont pas membres du projet.
5. Choisissez le rôle de l'utilisateur.

Lorsque vous avez terminé d'ajouter des utilisateurs, choisissez Inviter.

## Étape 3 : créer des problèmes sur lesquels collaborer et suivre le travail

CodeCatalyst vous aide à suivre les fonctionnalités, les tâches, les bogues et tout autre travail impliqué dans votre projet présentant des problèmes. Vous pouvez créer des problèmes pour suivre le travail et les idées nécessaires. Par défaut, lorsque vous créez un problème, celui-ci est ajouté à votre backlog. Vous pouvez déplacer les problèmes vers un tableau où vous pouvez suivre le travail en cours. Vous pouvez également attribuer un problème à un membre du projet en particulier.

Pour créer un problème pour un projet

1. Dans le volet de navigation, sélectionnez Problèmes.
2. Choisissez Créer un problème.
3. Dans Titre du numéro, saisissez le nom du problème. Fournissez éventuellement une description du problème. Dans cet exemple, utilisez **make a change in the src/mysfit\_data.json file**.
4. Choisissez la priorité, l'estimation, le statut et les étiquettes. Sous Personne assignée, choisissez +Ajouter moi pour vous attribuer le problème.
5. Choisissez Créer un problème. Le problème est désormais visible sur le tableau. Choisissez la carte pour déplacer le problème vers la colonne En cours.

Pour plus d'informations, consultez [Problèmes liés à CodeCatalyst](#).

## Étape 4 : Afficher votre référentiel source

Votre plan installe un référentiel source qui contient les fichiers permettant de définir et de prendre en charge votre application ou votre service. Voici quelques répertoires et fichiers remarquables du référentiel source :

- Répertoire `.cloud9` : contient les fichiers de support pour l'environnement de AWS Cloud9 développement.
- Répertoire `.codecatalyst` : contient le fichier de définition du YAML flux de travail pour chaque flux de travail inclus dans le plan.
- Répertoire `.idea` — Contient les fichiers de support pour les environnements de JetBrains développement.
- Répertoire `.vscode` : contient les fichiers de support pour l'environnement de développement de code Visual Studio.
- Répertoire `CDKStacks` — Contient les fichiers de AWS CDK pile qui définissent l'infrastructure dans le. AWS Cloud
- répertoire `src` — Contient le code source de l'application.
- répertoire `tests` : contient des fichiers pour les tests entiers et unitaires exécutés dans le cadre du flux de travail CI/CD automatisé qui s'exécute lorsque vous créez et testez votre application.
- répertoire `Web` — Contient le code source du frontend. Les autres fichiers incluent des fichiers de projet tels que le package `.json` fichier contenant des métadonnées importantes concernant votre projet, la `index.html` page du site Web, le `.eslintrc.cjs` fichier contenant le code linting et le `tsconfig.json` fichier permettant de spécifier les fichiers racines et les options du compilateur.
- `Dockerfile` — Décrit le conteneur de l'application.
- `README.md` fichier — Contient les informations de configuration du projet.

Pour accéder aux référentiels sources d'un projet

1. Accédez à votre projet, puis effectuez l'une des opérations suivantes :
  - Sur la page de résumé de votre projet, choisissez le référentiel souhaité dans la liste, puis choisissez Afficher le référentiel.



- Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources. Dans Référentiels sources, choisissez le nom du référentiel dans la liste. Vous pouvez filtrer la liste des référentiels en saisissant une partie du nom du référentiel dans la barre de filtre.
2. Sur la page d'accueil du référentiel, consultez le contenu du référentiel et les informations sur les ressources associées, telles que le nombre de pull requests et les flux de travail. Par défaut, le contenu de la branche par défaut est affiché. Vous pouvez modifier l'affichage en choisissant une autre branche dans la liste déroulante.

## Étape 5 : Création d'un environnement de développement avec une branche de test et modification rapide du code

Vous pouvez rapidement travailler sur le code de votre référentiel source en créant un environnement de développement. Pour ce didacticiel, nous partons du principe que vous allez :

- Créez un environnement AWS Cloud9 de développement.
- Choisissez l'option permettant de travailler dans une nouvelle branche de la branche principale lors de la création de l'environnement de développement.
- Utilisez le nom `test` de cette nouvelle branche.

Dans une étape ultérieure, vous utiliserez l'environnement de développement pour modifier le code et créer une pull request.

Pour créer un environnement de développement avec une nouvelle branche

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez au projet dans lequel vous souhaitez créer un environnement de développement.
3. Choisissez le nom du référentiel dans la liste des référentiels sources du projet. Dans le volet de navigation, vous pouvez également choisir Code, choisir Référentiels sources et choisir le référentiel pour lequel vous souhaitez créer un environnement de développement.
4. Sur la page d'accueil du référentiel, choisissez Create Dev Environment.
5. Choisissez un IDE compatible dans le menu déroulant. Pour plus d'informations, consultez [Environnements de développement intégrés pris en charge pour les environnements de développement](#).

6. Choisissez le référentiel à cloner, choisissez Travailler dans une nouvelle branche, entrez le nom de la branche dans le champ Nom de la branche et choisissez une branche à partir de laquelle créer la nouvelle branche dans le menu déroulant Créer une branche depuis.
7. Ajoutez éventuellement un alias pour l'environnement de développement.
8. Vous pouvez éventuellement cliquer sur le bouton d'édition de la configuration de l'environnement de développement pour modifier la configuration de calcul, de stockage ou de temporisation de l'environnement de développement.
9. Choisissez Créer. Pendant la création de votre environnement de développement, la colonne d'état de l'environnement de développement affichera Démarrage, et la colonne d'état indiquera En cours d'exécution une fois l'environnement de développement créé. Un nouvel onglet s'ouvre avec votre environnement de développement dans l'IDE de votre choix. Vous pouvez modifier le code, valider et appliquer vos modifications.

Dans cette section, vous allez travailler avec votre exemple d'application généré en CodeCatalyst apportant des modifications au code à l'aide de requêtes d'extraction qui sont automatiquement créées et déployées sur les ressources de la connexion Compte AWS lorsque la demande d'extraction est fusionnée.

Pour apporter une modification à votre `src/mysfit_data.json` fichier

1. Accédez à l'environnement de développement de votre projet. Dans AWS Cloud9, développez le menu de navigation latéral pour parcourir les fichiers. `mysfits`Développez `src` et ouvrez `src/mysfit_data.json`.
2. Dans le fichier, modifiez la valeur du "Age" : champ de 6 à 12. Votre ligne doit ressembler à ce qui suit :

```
{
  "Age": 12,
  "Description": "Twilight's personality sparkles like the night sky and is looking for a forever home with a Greek hero or God. While on the smaller side at 14 hands, he is quite adept at accepting riders and can fly to 15,000 feet. Twilight needs a large area to run around in and will need to be registered with the FAA if you plan to fly him above 500 feet. His favorite activities include playing with chimeras, going on epic adventures into battle, and playing with a large inflatable ball around the paddock. If you bring him home, he'll quickly become your favorite little Pegasus.",
  "GoodEvil": "Good",
  "LawChaos": "Lawful",
```

```
"Name": "Twilight Glitter",
  "ProfileImageUri": "https://www.mythicalmysfits.com/images/
pegasus_hover.png",
  "Species": "Pegasus",
  "ThumbImageUri": "https://www.mythicalmysfits.com/images/pegasus_thumb.png"
},
```

3. Enregistrez le fichier.
4. Accédez au référentiel mysfits à l'aide de la **cd /projects/mysfits** commande.
5. Ajoutez, validez et publiez vos modifications à l'aide des commandes git add, git commit et git push.

```
git add .
git commit -m "make an example change"
git push
```

## Étape 6 : Afficher le flux de travail qui crée l'application moderne

Après avoir créé le projet d'application moderne, CodeCatalyst génère plusieurs ressources en votre nom, y compris un flux de travail. Un flux de travail est une procédure automatisée définie dans un fichier .yaml qui décrit comment créer, tester et déployer votre code.

Dans ce didacticiel, vous avez CodeCatalyst créé un flux de travail et l'avez démarré automatiquement lorsque vous avez créé votre projet. (Le flux de travail est peut-être toujours en cours d'exécution en fonction de la date à laquelle vous avez créé votre projet.) Utilisez les procédures suivantes pour vérifier la progression du flux de travail, consulter les journaux et les rapports de test générés, puis accéder à l'URL de l'application déployée.

Pour vérifier la progression du flux de travail

1. Dans le volet de navigation de la CodeCatalyst console, choisissez CI/CD, puis Workflows.

La liste des flux de travail s'affiche. Il s'agit des flux de travail générés et lancés par le CodeCatalyst plan lorsque vous avez créé votre projet.

2. Observez la liste des flux de travail. Vous devriez en voir quatre :

- Les deux flux de travail en haut correspondent à la **test** branche que vous avez créée précédemment [Étape 5 : Création d'un environnement de développement avec une branche de test et modification rapide du code](#). Ces flux de travail sont des clones, les flux de travail de la

main branche. Le n'ApplicationDeploymentPipelineest pas actif car il est configuré pour être utilisé avec la main branche. Le OnPullRequestflux de travail n'a pas été exécuté car aucune pull request n'a été effectuée.

- Les deux flux de travail en bas correspondent à la main branche créée lorsque vous avez exécuté le plan plus tôt. Le ApplicationDeploymentPipelineflux de travail est actif et son exécution est en cours (ou terminée).

#### Note

Si l'ApplicationDeploymentPipelineexécution échoue avec une DeployBackenderreur ou une erreur Build @cdk\_bootstrap, cela peut être dû au fait que vous avez déjà exécuté l'application Web moderne à trois niveaux et qu'elle a laissé d'anciennes ressources en conflit avec le plan actuel. Vous devrez supprimer ces anciennes ressources, puis réexécuter le flux de travail. Pour plus d'informations, consultez [Nettoyage des ressources](#).

3. Choisissez le ApplicationDeploymentPipelineflux de travail associé à la main branche, en bas. Ce flux de travail a été exécuté à l'aide du code source de la main branche.



Un diagramme de flux de travail apparaît. Le diagramme montre plusieurs blocs, chacun représentant une tâche ou une action. La plupart des actions sont organisées verticalement, les actions situées en haut étant exécutées avant celles situées en dessous. Les actions agencées côte à côte s'exécutent en parallèle. Les actions regroupées doivent toutes s'exécuter correctement avant que l'action située en dessous puisse démarrer.

Les principaux blocs sont les suivants :

- WorkflowSource— Ce bloc représente votre dépôt source. Il indique, entre autres informations, le nom du référentiel source (mysfits) et le commit qui a automatiquement lancé l'exécution du flux de travail. CodeCatalyst a généré ce commit lorsque vous avez créé votre projet.
- Construire — Ce bloc représente un regroupement de deux actions qui doivent toutes deux être terminées avec succès pour que l'action suivante commence.
- DeployBackend— Ce bloc représente une action qui déploie les composants principaux de l'application dans le AWS cloud.
- Tests — Ce bloc représente un groupe de deux actions de test qui doivent toutes deux être terminées avec succès pour que l'action suivante démarre.

- `DeployFrontend`— Ce bloc représente une action qui déploie les composants frontaux de l'application dans le AWS cloud.
4. Choisissez l'onglet Définition (en haut). Le [fichier de définition du flux](#) de travail apparaît sur la droite. Le fichier contient les sections importantes suivantes :
    - Une `Triggers` section, en haut. Cette section indique que le flux de travail doit démarrer chaque fois que le code est transféré vers la `main` branche du référentiel source. Les push vers d'autres branches (par exemple `test`) ne démarreront pas ce flux de travail. Le flux de travail s'exécute à l'aide des fichiers de la `main` branche.
    - Une `Actions` section, sous `Triggers`. Cette section définit les actions que vous voyez dans le diagramme de flux de travail.
  5. Choisissez l'onglet État le plus récent (en haut), puis choisissez une action dans le diagramme de flux de travail.
  6. Sur la droite, choisissez l'onglet Configuration pour voir les paramètres de configuration utilisés par l'action lors de la dernière exécution. Chaque paramètre de configuration possède une propriété correspondante dans le fichier de définition du flux de travail.
  7. Laissez la console ouverte et passez à la procédure suivante.

Pour consulter les journaux de construction et les rapports de test

1. Choisissez l'onglet État le plus récent.
2. Dans le diagramme du flux de travail, choisissez l'`DeployFrontend` action.
3. Attendez que l'action soit terminée. Surveillez que l'icône « En cours » () soit remplacée par une icône « Succès » ()
4. Choisissez l'action `build_backend`.
5. Choisissez l'onglet Journaux et développez quelques sections pour afficher les messages du journal relatifs à ces étapes. Vous pouvez voir les messages relatifs à la configuration du backend.
6. Choisissez l'onglet Rapports, puis sélectionnez le `backend-coverage.xml` rapport. CodeCatalyst affiche le rapport associé. Le rapport indique les tests de couverture de code qui ont été exécutés et indique la proportion de lignes de code validées avec succès par les tests, par exemple 80 %.

Pour plus d'informations sur les rapports de test, consultez [Tests à l'aide de workflows dans CodeCatalyst](#).


 Tip

Vous pouvez également consulter vos rapports de test en choisissant Rapports dans le volet de navigation.

7. Laissez la CodeCatalyst console ouverte et passez à la procédure suivante.

Pour confirmer que l'application moderne a été déployée avec succès

1. Retournez au ApplicationDeploymentPipelineflux de travail et choisissez le lien Exécuter - **chaîne** de caractères de la dernière exécution.
2. Dans le diagramme du flux de travail, recherchez l'DeployFrontendaction et cliquez sur le lien Afficher l'application. Le site Web de Mysfit apparaît.

 Note

Si le lien de l'application View n'apparaît pas dans l'DeployFrontendaction, assurez-vous d'avoir choisi le lien Run ID.

3. Recherchez le pégase Mysfit nommé Twilight Glitter. Notez la valeur correspondant à l'âge. C'est le cas6. Vous allez modifier le code pour mettre à jour l'âge.

## Étape 7 : demandez à d'autres personnes de revoir vos modifications

Maintenant que des modifications ont été apportées à une branche nommée `test`, vous pouvez demander à d'autres personnes de les examiner en créant une pull request. Procédez comme suit pour créer une pull request afin de fusionner les modifications de la `test` branche dans la `main` branche.

Pour créer une pull request

1. Accédez à votre projet.
2. Effectuez l'une des actions suivantes :


- Dans le volet de navigation, choisissez Code, choisissez Pull requests, puis Create pull request.
  - Sur la page d'accueil du référentiel, choisissez Plus, puis Create pull request.
  - Sur la page du projet, choisissez Create pull request.
3. Dans le référentiel source, assurez-vous que le référentiel source spécifié est celui qui contient le code validé. Cette option n'apparaît que si vous n'avez pas créé la pull request depuis la page principale du dépôt.
  4. Dans Branche de destination, choisissez la branche dans laquelle vous souhaitez fusionner le code une fois celui-ci révisé.
  5. Dans Branche source, choisissez la branche qui contient le code validé.
  6. Dans Titre de la demande d'extraction, entrez un titre qui aide les autres utilisateurs à comprendre ce qui doit être revu et pourquoi.
  7. (Facultatif) Dans la description de la Pull request, fournissez des informations telles qu'un lien vers les problèmes ou une description de vos modifications.

 Tip

Vous pouvez choisir Write description for me afin de générer CodeCatalyst automatiquement une description des modifications contenues dans la pull request. Vous pouvez apporter des modifications à la description générée automatiquement après l'avoir ajoutée à la pull request.

Cette fonctionnalité nécessite que les fonctionnalités d'IA génératives soient activées pour l'espace. Pour plus d'informations, consultez [la section Gestion des fonctionnalités d'IA générative](#).

8. (Facultatif) Dans Problèmes, choisissez Lier les problèmes, puis choisissez un problème dans la liste ou entrez son ID. Pour dissocier un problème, cliquez sur l'icône de dissociation.
9. (Facultatif) Dans Réviseurs obligatoires, sélectionnez Ajouter les réviseurs requis. Choisissez parmi la liste des membres du projet pour les ajouter. Les réviseurs requis doivent approuver les modifications avant que la pull request puisse être fusionnée dans la branche de destination.

 Note

Vous ne pouvez pas ajouter un réviseur à la fois en tant que réviseur obligatoire et en tant que réviseur facultatif. Vous ne pouvez pas vous ajouter en tant que réviseur.

10. (Facultatif) Dans Réviseurs facultatifs, sélectionnez Ajouter des réviseurs facultatifs. Choisissez parmi la liste des membres du projet pour les ajouter. Les réviseurs facultatifs n'ont pas à approuver les modifications comme condition préalable pour que la pull request puisse être fusionnée dans la branche de destination.
11. Passez en revue les différences entre les branches. La différence affichée dans une pull request réside dans les modifications entre la révision dans la branche source et la base de fusion, qui est le commit principal de la branche de destination au moment de la création de la pull request. Si aucune modification ne s'affiche, il se peut que les branches soient identiques ou que vous ayez choisi la même branche pour la source et pour la destination.
12. Lorsque vous êtes certain que la pull request contient le code et les modifications que vous souhaitez vérifier, choisissez Create.

#### Note

Après avoir créé la pull request, vous pouvez ajouter des commentaires. Des commentaires peuvent être ajoutés à la pull request ou à des lignes individuelles des fichiers, ainsi qu'à la pull request globale. Vous pouvez ajouter des liens vers des ressources, telles que des fichiers, en utilisant le signe @ suivi du nom du fichier.


Lorsque vous créez la pull request, le OnPullRequestflux de travail commence à utiliser les fichiers source de la test branche. Pendant que vos réviseurs approuvent votre modification de code, vous pouvez observer les résultats en choisissant le flux de travail et en consultant le résultat du test.

Après avoir passé en revue la modification, vous pouvez fusionner le code. La fusion du code avec la branche par défaut lancera automatiquement le flux de travail qui créera et déploiera vos modifications.

Pour fusionner une pull request depuis la CodeCatalyst console

1. Accédez à votre projet d'application moderne.
2. Sur la page du projet, sous Ouvrir les pull requests, choisissez la pull request que vous souhaitez fusionner. Si vous ne voyez pas la pull request, choisissez Afficher tout, puis choisissez-la dans la liste. Choisissez Merge (Fusionner).
3. Choisissez parmi les stratégies de fusion disponibles pour la pull request. Sélectionnez ou désélectionnez éventuellement l'option permettant de supprimer la branche source après avoir fusionné la pull request, puis choisissez Merge.



 Note

Si le bouton Fusionner n'est pas actif ou si le libellé Non fusionnable s'affiche, cela signifie qu'un ou plusieurs réviseurs requis n'ont pas encore approuvé la pull request ou que la pull request ne peut pas être fusionnée dans la CodeCatalyst console. Un réviseur qui n'a pas approuvé une pull request est indiqué par une icône en forme d'horloge dans la section Vue d'ensemble, dans la zone des détails de la pull request. Si tous les réviseurs requis ont approuvé la pull request mais que le bouton Fusionner n'est toujours pas actif, il se peut que vous ayez un conflit de fusion. Vous pouvez résoudre les conflits de fusion pour la branche de destination dans la CodeCatalyst console, puis fusionner la pull request, ou vous pouvez résoudre les conflits et fusionner localement, puis transférer le commit contenant la fusion vers CodeCatalyst. Pour plus d'informations, consultez [Fusion d'une pull request \(Git\)](#) votre documentation Git.

Une fois que vous avez fusionné les modifications de la `test` branche dans la `main` branche, les modifications démarrent automatiquement le `ApplicationDeploymentPipeline` flux de travail qui crée et déploie votre modification.

Pour voir le commit fusionné s'exécuter dans le `ApplicationDeploymentPipeline` flux de travail

1. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
2. Dans Workflows, dans `ApplicationDeploymentPipeline`, développez Recent runs. Vous pouvez voir le flux de travail lancé par le commit de fusion. Sélectionnez-le éventuellement pour suivre la progression de la course.
3. Une fois l'exécution terminée, rechargez l'URL que vous avez consultée plus tôt. Regardez le pégase pour vérifier que l'âge a changé.



## Étape 8 : Clôturer le problème

Lorsqu'un problème est résolu, il peut être fermé sur la CodeCatalyst console.

Pour résoudre un problème lié à un projet

1. Accédez à votre projet.
2. Dans le volet de navigation, sélectionnez Problèmes.
3. Ajoutez rag-and-drop le problème à la colonne Terminé.

Pour plus d'informations, consultez [Problèmes liés à CodeCatalyst](#).

## Nettoyage des ressources

Nettoyez CodeCatalyst et supprimez AWS les traces de ce didacticiel dans votre environnement.

Vous pouvez choisir de continuer à utiliser le projet que vous avez utilisé pour ce didacticiel ou de supprimer le projet et les ressources associées.

**Note**

La suppression de ce projet supprimera tous les référentiels, problèmes et artefacts du projet pour tous les membres.

Pour supprimer un projet

1. Accédez à votre projet, puis sélectionnez Paramètres du projet.
2. Choisissez l'onglet Général.
3. Sous le nom du projet, choisissez Supprimer le projet.

Pour supprimer des ressources dans AWS CloudFormation et Amazon S3

1. Connectez-vous AWS Management Console avec le même compte que celui que vous avez ajouté à votre CodeCatalyst espace.
2. Va au AWS CloudFormationservice.
3. Supprimez la pile de **chaînes** mysfits.
4. **Supprimez la pile de chaînes development-mysfits.**
5. Choisissez (mais ne supprimez pas) la pile CDKToolkit. Sélectionnez l'onglet Ressources. Choisissez le StagingBucketlien, puis supprimez le compartiment et son contenu dans Amazon S3.

**Note**

Si vous ne supprimez pas ce compartiment manuellement, une erreur peut s'afficher lors de la réexécution du plan d'application Web moderne à trois niveaux.


6. (Facultatif) Supprimez la pile CDKToolkit.

## Référence

Le plan d'application Web moderne à trois niveaux déploie des ressources dans votre CodeCatalyst espace et votre AWS compte dans le cloud. AWS Ces ressources sont les suivantes :

- Dans votre CodeCatalyst espace :

- Un CodeCatalyst projet qui inclut les ressources suivantes :
  - Un [référentiel source](#) — Ce référentiel contient un exemple de code pour une application Web « Mysfits ».
  - Un [flux de travail](#) — Ce flux de travail crée et déploie le code de l'application Mysfits chaque fois qu'une modification est apportée à la branche par défaut
  - Un [tableau des problèmes](#) et un carnet de dossiers — Ce tableau et ce carnet peuvent être utilisés pour planifier et suivre le travail.
  - Une [suite de rapports de test](#) — Cette suite inclut des rapports automatisés inclus dans l'exemple de code.
- Dans le document associé Compte AWS :
  - Une pile CDKToolkit — Cette pile déploie les ressources suivantes :
    - Un compartiment intermédiaire Amazon S3, une politique de compartiment et la AWS KMS clé utilisée pour chiffrer le compartiment.
    - Rôle de déploiement IAM pour l'action de déploiement.
    - AWS Rôles et politiques IAM à l'appui des ressources de la pile.

 Note

Le CDKToolkit n'est pas démonté et recréé pour chaque déploiement. Il s'agit d'une pile qui est initiée dans chaque compte pour prendre en charge le AWS CDK.

- Une pile de **chaînes** development-mysfits — Cette BackEnd pile déploie les ressources de backend suivantes :
  - Un point de terminaison Amazon API Gateway.
  - AWS Rôles et politiques IAM à l'appui des ressources de la pile.
  - Une AWS Lambda fonction et une couche fournissent la plate-forme de calcul sans serveur pour les applications modernes.
  - Une politique et un rôle IAM pour le déploiement du bucket et la fonction Lambda.
- Une pile de **chaînes** mysfits — Cette pile déploie l' AWS Amplify application frontale.

## Consultez aussi

Pour plus d'informations sur les AWS services dans lesquels les ressources sont créées dans le cadre de ce didacticiel, consultez les rubriques suivantes .

- Amazon S3 : un service permettant de stocker vos ressources frontales sur un service de stockage d'objets offrant une évolutivité, une haute disponibilité des données, une sécurité et des performances de pointe. Pour plus d'informations, consultez le [guide de l'utilisateur Amazon S3](#).
- Amazon API Gateway : un service de création, de publication, de maintenance, de surveillance et de sécurisation des API REST, HTTP et des WebSocket API à n'importe quelle échelle. Pour plus d'informations, consultez le [guide du développeur d'API Gateway](#).
- Amplify — Un service d'hébergement de votre application frontale. Pour plus d'informations, consultez le [Guide de l'utilisateur de l'AWS Amplify hébergement](#).
- AWS Cloud Development Kit (AWS CDK)— Un cadre permettant de définir l'infrastructure cloud dans le code et de la provisionner via AWS CloudFormation ce dernier. AWS CDK Cela inclut le AWS CDK Toolkit, qui est un outil de ligne de commande permettant d'interagir avec des AWS CDK applications et des piles. Pour de plus amples informations, consultez le [Guide du développeur AWS Cloud Development Kit \(AWS CDK\)](#).
- Amazon DynamoDB : service de base de données NoSQL entièrement géré pour le stockage de données. Pour plus d'informations, consultez le guide du [développeur Amazon DynamoDB](#).
- AWS Lambda— Un service permettant d'invoquer votre code sur une infrastructure informatique à haute disponibilité sans provisionner ni gérer de serveurs. Pour de plus amples informations, consultez le [Guide du développeur AWS Lambda](#).
- AWS IAM — Service permettant de contrôler en toute sécurité l'accès à AWS ses ressources. Pour plus d'informations, consultez le [Guide de l'utilisateur IAM](#).

## Tutoriel : Commencer avec un projet vide et ajouter manuellement des ressources

Vous pouvez créer un projet vide sans aucune ressource prédéfinie en choisissant le plan de projet vide lorsque vous créez le projet. Après avoir créé un projet vide, vous pouvez créer et y ajouter des ressources en fonction des besoins de votre projet. Étant donné que les projets créés sans plan sont vides lors de leur création, cette option nécessite des connaissances supplémentaires en matière de création et de configuration de CodeCatalyst ressources pour démarrer.

### Rubriques

- [Prérequis](#)
- [Création d'un projet vide](#)
- [Création d'un référentiel source](#)

- [Création d'un flux de travail pour créer, tester et déployer une modification de code](#)
- [Invitez quelqu'un à rejoindre votre projet](#)
- [Créez des problèmes sur lesquels collaborer et suivre le travail](#)

## Prérequis

Pour créer un projet vide, le rôle d'administrateur de l'espace ou d'utilisateur avancé doit vous être attribué. Si c'est la première fois que vous vous connectez à CodeCatalyst, consultez [Configuration CodeCatalyst](#).

## Création d'un projet vide

La création d'un projet est la première étape pour pouvoir travailler ensemble. Si vous souhaitez créer vos propres ressources, telles que des référentiels sources et des flux de travail, vous pouvez commencer par un projet vide.

Pour créer un projet vide

1. Accédez à l'espace dans lequel vous souhaitez créer un projet.
2. Sur le tableau de bord de l'espace, choisissez Créer un projet.
3. Choisissez Commencer à zéro.
4. Sous Donnez un nom à votre projet, entrez le nom que vous souhaitez attribuer à votre projet. Le nom doit être unique dans votre espace.
5. Sélectionnez Create a project (Créer un projet).

Maintenant que vous avez un projet vide, l'étape suivante consiste à créer un référentiel source.


## Création d'un référentiel source

Créez un référentiel source pour stocker et collaborer sur le code de votre projet. Les membres du projet peuvent cloner ce dépôt sur leur ordinateur local pour travailler sur le code. Vous pouvez également choisir de lier un dépôt hébergé dans un service pris en charge, mais cela n'est pas abordé dans ce didacticiel. Pour plus d'informations, consultez [Lier un référentiel source](#).

Pour créer un référentiel source

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).

2. Accédez à votre projet.
3. Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources.
4. Choisissez Ajouter un référentiel, puis sélectionnez Créer un référentiel.
5. Dans Nom du référentiel, saisissez un nom pour le référentiel. Dans ce guide, nous utilisons un autre nom *codecatalyst-source-repository*, mais vous pouvez en choisir un autre. Les noms des référentiels doivent être uniques au sein d'un projet. Pour plus d'informations sur les exigences relatives aux noms de référentiels, consultez [Quotas pour les référentiels sources dans CodeCatalyst](#).
6. (Facultatif) Dans Description, ajoutez une description du référentiel afin d'aider les autres utilisateurs du projet à comprendre à quoi sert le référentiel.
7. (Facultatif) Ajoutez un `.gitignore` fichier correspondant au type de code que vous souhaitez envoyer.
8. Choisissez Créer.

 Note

CodeCatalyst ajoute un README .md fichier à votre dépôt lorsque vous le créez. CodeCatalyst crée également un commit initial pour le dépôt dans une branche par défaut nommée main. Vous pouvez modifier ou supprimer le fichier README.md, mais vous ne pouvez pas modifier ou supprimer la branche par défaut.

Vous pouvez rapidement ajouter du code dans votre dépôt en créant un environnement de développement. Pour ce didacticiel, nous vous recommandons de créer un environnement de développement en utilisant AWS Cloud9 et de choisir l'option permettant de créer une branche à partir de la branche principale lors de la création de l'environnement de développement. Nous utilisons le nom **test** de cette succursale, mais vous pouvez saisir un autre nom de succursale si vous préférez.

Pour créer un environnement de développement avec une nouvelle branche

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez au projet dans lequel vous souhaitez créer un environnement de développement.
3. Choisissez le nom du référentiel dans la liste des référentiels sources du projet. Dans le volet de navigation, vous pouvez également choisir Code, choisir Référentiels sources et choisir le référentiel pour lequel vous souhaitez créer un environnement de développement.

4. Sur la page d'accueil du référentiel, choisissez Create Dev Environment.
5. Choisissez un IDE compatible dans le menu déroulant. Pour plus d'informations, consultez [Environnements de développement intégrés pris en charge pour les environnements de développement](#).
6. Choisissez le référentiel à cloner, choisissez Travailler dans une nouvelle branche, entrez le nom de la branche dans le champ Nom de la branche et choisissez une branche à partir de laquelle créer la nouvelle branche dans le menu déroulant Créer une branche depuis.
7. Ajoutez éventuellement un alias pour l'environnement de développement.
8. Vous pouvez éventuellement cliquer sur le bouton d'édition de la configuration de l'environnement de développement pour modifier la configuration de calcul, de stockage ou de temporisation de l'environnement de développement.
9. Choisissez Créer. Pendant la création de votre environnement de développement, la colonne d'état de l'environnement de développement affichera Démarrage, et la colonne d'état affichera En cours d'exécution une fois l'environnement de développement créé. Un nouvel onglet s'ouvre avec votre environnement de développement dans l'IDE de votre choix. Vous pouvez modifier le code, valider et appliquer vos modifications.

## Création d'un flux de travail pour créer, tester et déployer une modification de code

Dans CodeCatalyst, vous organisez la création, le test et le déploiement de vos applications ou services dans des flux de travail. Les flux de travail se composent d'actions et peuvent être configurés pour s'exécuter automatiquement après la survenue d'événements spécifiques du référentiel source, tels que des transferts de code ou l'ouverture ou la mise à jour d'une pull request. Pour de plus amples informations sur les flux de travail, veuillez consulter [Créez, testez et déployez avec des flux de travail dans CodeCatalyst](#).

Suivez les instructions [Commencer à utiliser les flux de travail dans CodeCatalyst](#) pour créer votre premier flux de travail.

## Invitez quelqu'un à rejoindre votre projet

Maintenant que vous avez configuré votre projet personnalisé, invitez d'autres personnes à travailler avec vous.



## Pour inviter quelqu'un à rejoindre votre projet

1. Accédez au projet auquel vous souhaitez inviter des utilisateurs.
2. Dans le volet de navigation, sélectionnez Paramètres du projet.
3. Dans l'onglet Membres, choisissez Inviter.
4. Entrez les adresses e-mail des personnes que vous souhaitez inviter en tant qu'utilisateurs de votre projet. Vous pouvez saisir plusieurs adresses e-mail séparées par un espace ou une virgule. Vous pouvez également choisir parmi les membres de votre espace qui ne sont pas membres du projet.
5. Choisissez le rôle de l'utilisateur.

Lorsque vous avez terminé d'ajouter des utilisateurs, choisissez Inviter.

## Créez des problèmes sur lesquels collaborer et suivre le travail

CodeCatalyst vous aide à suivre les fonctionnalités, les tâches, les bogues et tout autre travail impliqué dans votre projet présentant des problèmes. Vous pouvez créer des problèmes pour suivre le travail et les idées nécessaires. Par défaut, lorsque vous créez un problème, celui-ci est ajouté à votre backlog. Vous pouvez déplacer les problèmes vers un tableau où vous pouvez suivre le travail en cours. Vous pouvez également attribuer un problème à un membre du projet en particulier.

### Pour créer un problème pour un projet

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).

Assurez-vous de naviguer dans le projet dans lequel vous souhaitez créer des problèmes. Pour afficher tous les projets, dans le volet de navigation, choisissez Amazon CodeCatalyst, et si nécessaire, choisissez Afficher tous les projets. Choisissez le projet dans lequel vous souhaitez créer ou traiter les problèmes.

2. Dans le volet de navigation, choisissez Track, puis Backlog.
3. Choisissez Créer un problème.
4. Dans Titre du numéro, saisissez le nom du problème. Fournissez éventuellement une description du problème. Choisissez le statut, la priorité et l'estimation du problème si vous le souhaitez. Vous pouvez également attribuer le problème à un membre du projet depuis la liste des membres du projet.

**i** Tip

Vous pouvez choisir d'attribuer un problème à Amazon Q pour qu'Amazon Q essaie de le résoudre. Si la tentative aboutit, une pull request sera créée et le statut du problème passera à En révision afin que vous puissiez revoir et tester le code. Pour plus d'informations, consultez [Tutoriel : Utiliser les fonctionnalités CodeCatalyst génératives de l'IA pour accélérer votre travail de développement](#).

Cette fonctionnalité nécessite que les fonctionnalités d'IA génératives soient activées pour l'espace. Pour plus d'informations, consultez [la section Gestion des fonctionnalités d'IA générative](#).

## 5. Choisissez Enregistrer.

Après avoir créé des problèmes, vous pouvez les attribuer aux membres du projet, les estimer et les suivre sur un tableau Kanban. Pour plus d'informations, voir [Problèmes liés à CodeCatalyst](#).

# Tutoriel : Utiliser les fonctionnalités CodeCatalyst génératives de l'IA pour accélérer votre travail de développement

Les fonctionnalités d'intelligence artificielle génératives d'Amazon CodeCatalyst sont en version préliminaire et sont susceptibles d'être modifiées. Ils ne sont disponibles que dans la région ouest des États-Unis (Oregon). L'accès aux fonctionnalités génératives de l'IA varie selon le niveau. Pour plus d'informations, consultez [Tarification d'](#).

Si vous avez un projet et un référentiel de sources sur Amazon CodeCatalyst dans un espace où les fonctionnalités d'IA générative sont activées, vous pouvez utiliser ces fonctionnalités pour accélérer le développement de logiciels. Les développeurs ont souvent plus de tâches à accomplir qu'ils n'ont le temps de les accomplir. Ils ne prennent souvent pas le temps d'expliquer leurs modifications de code à leurs collègues lorsqu'ils créent des pull requests pour examiner ces modifications, s'attendant à ce que les autres utilisateurs trouvent ces modifications explicites. Les créateurs et réviseurs de pull request n'ont pas non plus le temps de rechercher et de lire attentivement tous les commentaires sur une pull request, en particulier si la pull request comporte plusieurs révisions. CodeCatalyst inclut des fonctionnalités d'intelligence artificielle génératives qui peuvent à la fois aider les membres de

l'équipe à accomplir leurs tâches plus rapidement et augmenter le temps dont ils disposent pour se concentrer sur les aspects les plus importants de leur travail.

### Note

Propulsé par Amazon Bedrock : AWS implémente la [détection automatique des abus](#). Dans la mesure où les fonctionnalités de développement des fonctionnalités Write description for me, Create content summary et Assign issues to Amazon Q sont basées sur Amazon Bedrock, les utilisateurs peuvent tirer pleinement parti des contrôles mis en œuvre dans Amazon Bedrock pour renforcer la sûreté, la sécurité et l'utilisation responsable de l'intelligence artificielle (IA).

Dans ce didacticiel, vous apprendrez à utiliser les fonctionnalités de l'IA générative pour vous aider CodeCatalyst à résumer les modifications entre les branches lors de la création de pull requests et pour résumer les commentaires laissés sur une pull request. Vous apprendrez également à créer des problèmes avec vos idées de modifications ou d'améliorations de code simples et à les attribuer à Amazon Q.

## Prérequis

Pour utiliser les CodeCatalyst fonctionnalités de ce didacticiel, vous devez d'abord avoir terminé les ressources suivantes et y avoir accès :

- Vous disposez d'un AWS Builder ID ou d'une identité d'authentification unique (SSO) à laquelle vous pouvez vous connecter. CodeCatalyst
- Votre projet se trouve dans un espace doté de fonctionnalités d'IA génératives activées. Pour plus d'informations, consultez [la section Gestion des fonctionnalités d'IA générative](#).
- Vous avez le rôle de contributeur ou d'administrateur de projet dans un projet dans cet espace.
- Au moins un référentiel source est configuré pour le projet. Les référentiels liés ne sont pas pris en charge.
- Lorsque vous attribuez des problèmes à une solution initiale créée par l'IA générative, le projet ne peut pas être configuré avec l'extension Jira Software. L'extension n'est pas prise en charge pour cette fonctionnalité.

Pour plus d'informations, consultez [Création d'un espace prenant en charge les utilisateurs de AWS Builder ID](#), [Problèmes liés à CodeCatalyst](#), [Extensions dans CodeCatalyst](#) et [Utilisation des rôles sur Amazon CodeCatalyst](#).

Ce didacticiel est basé sur un projet créé à l'aide du modèle d'application Web moderne à trois niveaux avec Python. Si vous utilisez un projet créé avec un plan différent, vous pouvez toujours suivre les étapes, mais certaines informations peuvent varier, telles que l'exemple de code et la langue.

## Créez un résumé des modifications de code entre les branches lors de la création d'une pull request

Une pull request est le principal moyen pour vous et les autres membres du projet de consulter, commenter et fusionner les modifications de code d'une branche à l'autre. Vous pouvez utiliser des pull requests pour examiner les modifications de code de manière collaborative afin de détecter des modifications ou des correctifs mineurs, des ajouts de fonctionnalités majeures ou de nouvelles versions de vos logiciels publiés. Le résumé des modifications du code et de l'intention qui sous-tend ces modifications dans le cadre de la description de la pull request est utile aux personnes qui examineront le code, et permet également de comprendre l'historique des modifications apportées au code au fil du temps. Cependant, les développeurs s'appuient souvent sur leur code pour s'expliquer ou pour fournir des détails ambigus plutôt que de décrire leurs modifications avec suffisamment de détails pour que les réviseurs puissent comprendre ce qu'ils examinent ou quelle était l'intention qui sous-tend les modifications apportées au code.

Vous pouvez utiliser la fonctionnalité Write description for me lorsque vous créez des pull requests pour qu'Amazon Q crée une description des modifications contenues dans une pull request. Lorsque vous choisissez cette option, Amazon Q analyse les différences entre la branche source contenant les modifications de code et la branche de destination dans laquelle vous souhaitez fusionner ces modifications. Il crée ensuite un résumé de ces modifications, ainsi que sa meilleure interprétation de l'intention et de l'effet de ces modifications.

Vous pouvez essayer cette fonctionnalité avec n'importe quelle pull request que vous créez, mais dans ce didacticiel, nous allons la tester en apportant quelques modifications simples au code contenu dans un projet créé dans un plan d'application Web moderne à trois niveaux basé sur Python.

**i** Tip

Si vous utilisez un projet créé avec un plan différent ou votre propre code, vous pouvez toujours suivre ce didacticiel, mais les exemples de ce didacticiel ne correspondront pas au code de votre projet. Au lieu de suivre l'exemple suggéré ci-dessous, apportez de simples modifications au code de votre projet dans une branche, puis créez une pull request pour tester la fonctionnalité, comme indiqué dans les étapes suivantes.

Tout d'abord, vous allez créer une branche dans le référentiel source. Vous allez ensuite modifier rapidement le code d'un fichier de cette branche à l'aide de l'éditeur de texte de la console. Vous allez ensuite créer une pull request et utiliser la fonctionnalité Rédiger une description pour moi pour résumer les modifications que vous avez apportées.

Pour créer une branche (console)

1. Dans la CodeCatalyst console, accédez au projet dans lequel se trouve votre référentiel source.
2. Choisissez le nom du référentiel dans la liste des référentiels sources du projet. Dans le volet de navigation, vous pouvez également choisir Code, puis Référentiels sources.
3. Choisissez le référentiel dans lequel vous souhaitez créer une branche.
4. Sur la page d'aperçu du référentiel, choisissez Plus, puis sélectionnez Créer une branche.
5. Entrez le nom de la branche.
6. Choisissez une branche à partir de laquelle créer la branche, puis choisissez Create.

Une fois que vous avez une branche, modifiez un fichier dans cette branche par une simple modification. Dans cet exemple, vous allez modifier le `test_endpoint.py` fichier pour faire passer le nombre de tentatives de test **3** à **5**.

**i** Tip

Vous pouvez également choisir de créer ou d'utiliser un environnement de développement pour effectuer cette modification de code. Pour de plus amples informations, veuillez consulter [Création d'un environnement de développement](#).

## Pour modifier le `test_endpoint.py` fichier dans la console

1. Sur la page d'aperçu du référentiel **mysfits** source, choisissez la liste déroulante des branches et choisissez la branche que vous avez créée lors de la procédure précédente.
2. Dans Fichiers, accédez au fichier que vous souhaitez modifier. Par exemple, pour modifier le `test_endpoint.py` fichier, développez les tests, développez l'entier, puis choisissez `test_endpoint.py`.
3. Choisissez Modifier.
4. À la ligne 7, modifiez le nombre de fois où tous les tests seront réessayés :

```
def test_list_all(retry=3):
```

par :

```
def test_list_all(retry=5):
```

5. Choisissez Commit et validez vos modifications dans votre branche.

Maintenant que vous avez une branche modifiée, vous pouvez créer une pull request.

### Créez une pull request avec un résumé des modifications

1. Sur la page de présentation du référentiel, choisissez Plus, puis Create pull request.
2. Dans Branche de destination, choisissez la branche dans laquelle vous souhaitez fusionner le code une fois celui-ci révisé.

#### Tip

Choisissez la branche à partir de laquelle vous avez créé votre branche dans la procédure précédente pour une démonstration la plus simple de cette fonctionnalité. Par exemple, si vous avez créé votre branche à partir de la branche par défaut du dépôt, choisissez cette branche comme branche de destination pour votre pull request.

3. Dans Branche source, choisissez la branche qui contient les modifications que vous venez de valider dans le `test_endpoint.py` fichier.
4. Dans Titre de la demande d'extraction, entrez un titre qui aide les autres utilisateurs à comprendre ce qui doit être revu et pourquoi.

5. Dans Description de la pull request, choisissez Write description for me afin de CodeCatalyst créer une description des modifications contenues dans la pull request.
6. Un résumé des modifications s'affiche. Passez en revue le texte suggéré, puis choisissez Accepter et ajouter à la description.
7. Modifiez éventuellement le résumé pour mieux refléter les modifications que vous avez apportées au code. Vous pouvez également choisir d'ajouter des réviseurs ou de lier des problèmes à cette pull request. Lorsque vous avez terminé d'apporter les modifications supplémentaires souhaitées, choisissez Créer.

## Créez un résumé des commentaires laissés sur les modifications de code dans une pull request

Lorsque les utilisateurs examinent une pull request, ils laissent souvent plusieurs commentaires sur les modifications apportées à cette pull request. S'il y a beaucoup de commentaires de la part de nombreux réviseurs, il peut être difficile de sélectionner des thèmes communs dans les commentaires, ou même de s'assurer que vous avez examiné tous les commentaires dans toutes les révisions. Vous pouvez utiliser la fonctionnalité Créer un résumé des commentaires pour qu'Amazon Q analyse tous les commentaires laissés sur les modifications de code dans une pull request et crée un résumé de ces commentaires.

### Note

Les résumés des commentaires sont éphémères. Si vous actualisez une pull request, le résumé disparaît. Les résumés de contenu n'incluent pas de commentaires sur l'ensemble de la pull request, mais uniquement des commentaires laissés sur les différences de code dans les révisions de la pull request.

Pour créer un résumé des commentaires dans une pull request

1. Accédez à la pull request que vous avez créée lors de la procédure précédente.

**i** Tip

Si vous préférez, vous pouvez utiliser n'importe quelle pull request ouverte dans votre projet. Dans la barre de navigation, choisissez Code, choisissez Pull requests et choisissez n'importe quelle pull request ouverte.

2. Ajoutez quelques commentaires à la pull request dans Changes si la pull request ne contient pas déjà de commentaires.
3. Dans Vue d'ensemble, choisissez Créer un résumé des commentaires. Une fois terminée, la section Récapitulatif des commentaires s'agrandit.
4. Consultez le résumé des commentaires laissés sur les modifications du code dans les révisions de la pull request et comparez-le aux commentaires de la pull request.

## Créez un problème et attribuez-le à Amazon Q

Les équipes de développement créent des problèmes pour suivre et gérer leur travail, mais il arrive qu'un problème persiste, soit parce que l'on ne sait pas exactement qui doit y travailler, soit parce qu'il nécessite des recherches sur une partie particulière de la base de code, soit parce qu'il faut d'abord s'occuper d'autres tâches urgentes. CodeCatalyst inclut l'intégration avec un assistant d'intelligence artificielle génératif appelé Amazon Q qui peut analyser un problème en fonction de son titre et de sa description. Si vous attribuez le problème à Amazon Q, celui-ci tentera de créer un projet de solution que vous pourrez évaluer. Cela peut vous aider, vous et votre équipe, à vous concentrer et à optimiser votre travail sur les problèmes qui nécessitent votre attention, tandis qu'Amazon Q travaille sur une solution aux problèmes que vous ne disposez pas des ressources nécessaires pour résoudre immédiatement.

**i** Tip

Amazon Q est le plus performant pour résoudre les problèmes simples et directs. Pour de meilleurs résultats, utilisez un langage simple pour expliquer clairement ce que vous voulez faire.

Lorsque vous attribuez un problème à Amazon Q, CodeCatalyst celui-ci est marqué comme bloqué jusqu'à ce que vous confirmiez la manière dont vous souhaitez qu'Amazon Q le traite. Vous devez répondre à trois questions avant de pouvoir continuer :



- Que vous souhaitiez confirmer chaque étape ou que vous souhaitiez qu'elle se poursuive sans commentaires. Si vous choisissez de confirmer chaque étape, vous pouvez répondre à Amazon Q avec des commentaires sur l'approche créée afin qu'il puisse répéter son approche si nécessaire. Amazon Q peut également consulter les commentaires des utilisateurs sur toute pull request qu'il crée si vous choisissez cette option. Si vous choisissez de ne pas confirmer chaque étape, Amazon Q peut terminer son travail plus rapidement, mais il ne prendra pas en compte les commentaires que vous lui ferez dans le cadre du problème ou des pull request qu'il crée.
- Si vous souhaitez l'autoriser à mettre à jour les fichiers de flux de travail dans le cadre de son travail. Les flux de travail de votre projet sont peut-être configurés pour démarrer des exécutions lors d'événements de pull request. Dans ce cas, toute pull request créée par Amazon Q qui inclut la création ou la mise à jour du flux de travail YAML peut démarrer une exécution des flux de travail inclus dans la pull request. Il est recommandé de ne pas autoriser Amazon Q à travailler sur des fichiers de flux de travail, sauf si vous êtes certain qu'aucun flux de travail de votre projet n'exécutera automatiquement ces flux de travail avant de consulter et d'approuver la pull request qu'il crée.
- Dans quel référentiel source souhaitez-vous qu'il fonctionne. Même si votre projet possède plusieurs référentiels de sources, Amazon Q ne peut travailler que sur le code d'un seul référentiel de sources. Les référentiels liés ne sont pas pris en charge.

Une fois que vous avez fait et confirmé vos choix, Amazon Q place le problème dans l'état En cours pendant qu'il tente de déterminer la nature de la demande en fonction du titre du problème et de sa description, ainsi que du code contenu dans le référentiel spécifié. Il créera un commentaire épinglé dans lequel il fournira des mises à jour sur l'état de ses travaux. Après avoir examiné les données, Amazon Q formulera une approche potentielle pour trouver une solution. Amazon Q enregistre ses actions en mettant à jour son commentaire épinglé et en commentant l'évolution du problème à chaque étape. Contrairement aux commentaires et réponses épinglés, il ne conserve pas une trace strictement chronologique de ses travaux. Il place plutôt les informations les plus pertinentes sur son travail au niveau supérieur du commentaire épinglé. Il tentera de créer du code en fonction de son approche et de son analyse du code déjà présent dans le référentiel. S'il parvient à générer une solution potentielle, il créera une branche et validera le code à cette branche. Il crée ensuite une pull request qui fusionnera cette branche avec la branche par défaut. Lorsque Amazon Q a terminé son travail, il place le problème dans En cours de révision afin que vous et votre équipe sachiez que du code est prêt à être évalué.

**Note**

Cette fonctionnalité n'est disponible que via Issues. Il n'est pas disponible si vous avez configuré votre projet pour utiliser Jira avec l'extension Jira Software. De plus, si vous avez personnalisé la mise en page de votre tableau, il est possible que le problème ne change pas d'état. Pour de meilleurs résultats, n'utilisez cette fonctionnalité qu'avec des projets dotés d'une disposition de tableau standard.

Une fois que vous avez attribué un problème à Amazon Q, vous ne pouvez pas modifier le titre ou la description du problème, ni l'attribuer à quelqu'un d'autre. Si vous annulez l'attribution du problème à Amazon Q, celui-ci terminera l'étape en cours, puis cessera de fonctionner. Il ne peut pas reprendre le travail ni être réaffecté au problème une fois qu'il a été annulé.

Dans cette partie du didacticiel, vous allez créer trois problèmes basés sur les fonctionnalités potentielles du code inclus dans les projets créés avec le modèle d'application Web moderne à trois niveaux : un pour ajouter un pour créer une nouvelle créature Mysfit, un pour ajouter une fonctionnalité de tri et un pour mettre à jour un flux de travail afin d'inclure une branche nommée.

**test****Note**

Si vous travaillez sur un projet avec un code différent, créez des problèmes avec les titres et les descriptions liés à cette base de code.

Pour créer un problème et obtenir une solution que vous pourrez évaluer

1. Dans le volet de navigation, choisissez Problèmes et assurez-vous que vous êtes dans la vue Tableau.
2. Choisissez Créer un problème.
3. Donnez au numéro un titre qui explique ce que vous voulez faire dans un langage simple. Par exemple, pour ce numéro, entrez le titre de **Create another mysfit named Quokkapus**. Dans Description, fournissez les informations suivantes :

```
Expand the table of mysfits to 13, and give the new mysfit the following characteristics:
```

Name: Quokkapus

Species: Quokka-Octopus hybrid

Good/Evil: Good

Lawful/Chaotic: Chaotic

Age: 216

Description: Australia is full of amazing marsupials, but there's nothing there quite like the Quokkapus.

She's always got a friendly smile on her face, especially when she's using her eight limbs to wrap you up in a great big hug. She exists on a diet of code bugs and caffeine. If you've got some gnarly code that needs assistance, adopt Quokkapus and put her to work - she'll love it! Just make sure you leave enough room for her to grow, and keep that coffee coming.

4. (Facultatif) Joignez une image à utiliser comme miniature et photo de profil pour le MysFit to the Issue. Dans ce cas, mettez à jour la description pour inclure des détails sur les images que vous souhaitez utiliser et pourquoi. Par exemple, vous pouvez ajouter ce qui suit à la description :  
« Le Mysfit nécessite le déploiement de fichiers image sur le site Web. Ajoutez ces images au référentiel source. »

#### Note

Les images jointes ne seront pas déployées sur le site Web dans ce didacticiel. Vous pouvez ajouter vous-même les images au site Web, puis laisser des commentaires pour qu'Amazon Q mette à jour son code afin qu'il pointe vers les images que vous souhaitez qu'il utilise une fois qu'il a créé une pull request.

Passez en revue la description et assurez-vous qu'elle contient tous les détails nécessaires avant de passer à l'étape suivante.

5. Dans Destinataires, choisissez Attribuer à Amazon Q.
6. Dans Référentiel source, choisissez le référentiel source qui contient le code du projet.
7. Faites glisser le bouton Require Amazon Q pour qu'il s'arrête après chaque étape et attendez que son sélecteur de travail passe à l'état actif.

**Note**

Choisir l'option d'arrêt d'Amazon Q après chaque étape vous permet de commenter le problème et de demander à Amazon Q de modifier son approche jusqu'à trois fois en fonction de vos commentaires. Si vous choisissez de ne pas arrêter Amazon Q après chaque étape afin de pouvoir passer en revue son travail, le travail progressera peut-être plus rapidement car Amazon Q n'attend pas vos commentaires, mais vous ne serez pas en mesure d'influencer l'orientation d'Amazon Q en laissant des commentaires. Amazon Q ne répondra pas non plus aux commentaires laissés dans une pull request si vous choisissez cette option.

8. Laissez le sélecteur Autoriser Amazon Q à modifier les fichiers de flux de travail à l'état inactif.
9. Choisissez Créer un problème. Votre point de vue change sur le tableau des problèmes.
10. Choisissez Créer un numéro pour créer un autre numéro, cette fois avec le titre **Change the `get_all_mysfits()` API to return mysfits sorted by the Age attribute**. Attribuez ce problème à Amazon Q et créez-le.
11. Choisissez Créer un numéro pour créer un autre numéro, cette fois avec le titre **Update the `OnPullRequest` workflow to include a branch named test in its triggers**. Vous pouvez éventuellement créer un lien vers le flux de travail dans la description. Attribuez ce problème à Amazon Q, mais assurez-vous cette fois que le sélecteur Autoriser Amazon Q à modifier les fichiers de flux de travail est défini sur l'état actif. Créez le problème pour revenir au tableau des problèmes.

**Tip**

Vous pouvez rechercher des fichiers, y compris des fichiers de flux de travail, en saisissant le symbole arobase (@) et en saisissant le nom du fichier.

Une fois que vous avez créé et attribué les problèmes, ils seront placés dans En cours. Amazon Q ajoutera des commentaires permettant de suivre l'évolution du problème dans un commentaire épinglé. S'il est en mesure de définir une approche pour une solution, il mettra à jour la description du problème avec une section Contexte contenant son analyse de la base de code et une section Approche détaillant l'approche proposée pour créer une solution. Si Amazon Q parvient à trouver une solution au problème décrit dans le problème, il créera une branche et modifiera le code dans cette branche afin de mettre en œuvre la solution proposée. Une fois que le code est prêt, il crée une pull

request afin que vous puissiez consulter les modifications de code suggérées, ajoutez un lien vers cette pull request au problème et placez le problème dans En révision.

### Important

Vous devez toujours examiner les modifications de code dans une pull request avant de la fusionner. Les modifications de code de fusion effectuées par Amazon Q, comme toute autre modification de code, peuvent avoir un impact négatif sur votre code de base et votre code d'infrastructure si le code fusionné n'est pas correctement examiné et contient des erreurs lors de la fusion.

Pour examiner un problème et une pull request associée contenant des modifications apportées par Amazon Q

1. Dans Problèmes, choisissez un problème en cours attribué à Amazon Q. Passez en revue les commentaires pour suivre la progression du bot. Le cas échéant, passez en revue le contexte et abordez-le dans la description du problème, puis choisissez X pour fermer le volet des problèmes.
2. Choisissez maintenant un problème attribué à Amazon Q qui est en cours de révision. Passez en revue le contexte et abordez l'approche qu'il enregistre dans la description du problème. Passez en revue les commentaires pour comprendre les actions qu'il a effectuées. Dans Pull requests, choisissez le lien vers la pull request à côté de l'étiquette Open pour vérifier le code.
3. Dans la pull request, passez en revue les modifications apportées au code. Pour de plus amples informations, veuillez consulter [Révision d'une pull request](#). Laissez des commentaires sur la pull request si vous souhaitez qu'Amazon Q modifie l'un des codes suggérés. Soyez précis lorsque vous laissez des commentaires pour Amazon Q pour de meilleurs résultats.

Par exemple, lorsque vous examinez la pull request créée pour **Create another mysfit named Quokkapus**, vous remarquerez peut-être une faute de frappe dans la description. Vous pouvez laisser un commentaire pour Amazon Q indiquant « Modifiez la description pour corriger la faute de frappe « needsa » en ajoutant un espace entre « needs » et « a » ». Vous pouvez également laisser un commentaire demandant à Amazon Q de mettre à jour la description et de fournir l'intégralité de la description révisée afin qu'elle soit incorporée.

Si vous avez chargé des images du nouveau Mysfit sur le site Web, vous pouvez laisser un commentaire pour qu'Amazon Q mette à jour le Mysfit avec des pointeurs vers l'image et la miniature à utiliser pour le nouveau Mysfit.

**Note**

Amazon Q ne répondra pas aux commentaires individuels. Amazon Q n'intégrera les commentaires laissés dans les commentaires des pull requests que si vous avez choisi l'option par défaut qui consiste à arrêter après chaque étape d'approbation lors de la création du problème.

- (Facultatif) Une fois que vous et les autres utilisateurs du projet avez laissé tous les commentaires souhaités concernant les modifications du code, choisissez **Create revision** pour qu'Amazon Q crée une révision de la pull request qui intègre les modifications que vous avez demandées dans les commentaires. La progression de la création de révisions sera signalée par Amazon Q dans la section **Vue d'ensemble**, et non dans la section **Modifications**. Assurez-vous d'actualiser votre navigateur pour consulter les dernières mises à jour d'Amazon Q concernant la création de la révision.

**Note**

Seul l'utilisateur qui a créé le problème peut créer une révision de la pull request. Vous ne pouvez demander qu'une seule révision d'une pull request. Assurez-vous d'avoir résolu tous les problèmes liés aux commentaires et d'être satisfait du contenu des commentaires avant de choisir **Créer une révision**.

- Un flux de travail est exécuté pour chaque pull request dans cet exemple de projet. Assurez-vous qu'un flux de travail s'exécute correctement avant de fusionner la pull request. Vous pouvez également choisir de créer des flux de travail et des environnements supplémentaires pour tester le code avant de le fusionner. Pour de plus amples informations, veuillez consulter [Commencer à utiliser les flux de travail dans CodeCatalyst](#).
- Lorsque vous êtes satisfait de la dernière révision de la pull request, choisissez **Merge**.

## Nettoyage des ressources

Une fois que vous aurez terminé ce didacticiel, envisagez de prendre les mesures suivantes pour nettoyer les ressources que vous avez créées au cours de ce didacticiel et dont vous n'avez plus besoin.

- Annulez l'attribution à Amazon Q de tous les problèmes qui ne sont plus résolus. Si Amazon Q a terminé de résoudre un problème ou n'a pas trouvé de solution, veillez à annuler l'attribution d'Amazon Q afin d'éviter d'atteindre le quota maximum pour les fonctionnalités d'IA génératives. Pour plus d'informations, consultez [Gestion des fonctionnalités d'IA générative](#) et [tarification](#).
- Déplacez tous les problèmes pour lesquels le travail est terminé vers Terminé.
- Si le projet n'est plus nécessaire, supprimez-le.

## Tutoriel : Création d'une application complète avec des plans PDK composables

Amazon CodeCatalyst fournit un certain nombre de plans différents pour vous aider à démarrer rapidement vos projets. Un projet créé à l'aide d'un plan inclut les ressources dont vous avez besoin, notamment un référentiel de sources, des exemples de code source, des flux de travail CI/CD, des rapports de construction et de test, ainsi que des outils intégrés de suivi des problèmes. Cependant, il peut arriver que vous souhaitiez développer progressivement un projet ou ajouter des fonctionnalités à un projet existant créé par un plan. Vous pouvez également le faire avec des plans. Ce didacticiel explique comment démarrer avec un plan unique qui pose les bases et vous permet de stocker tout le code de votre projet dans un référentiel unique. À partir de là, vous avez la flexibilité d'intégrer des ressources et une infrastructure supplémentaires en appliquant d'autres plans en plus du plan initial à votre convenance. Grâce à cette méthode modulaire, vous pouvez répondre à des exigences spécifiques dans le cadre de plusieurs projets.

Ce didacticiel explique comment composer ensemble plusieurs plans du kit de développement de projet AWS (AWS PDK) afin de créer une application composée d'un site Web React, d'une API Smithy et de l'infrastructure CDK associée pour la déployer sur AWS. Le kit AWS PDK fournit des éléments de base pour les modèles courants ainsi que des outils de développement pour gérer et développer vos projets. Pour plus d'informations, consultez le [référentiel GitHub source du kit AWS PDK](#).

Les plans PDK suivants sont conçus pour être utilisés les uns avec les autres afin de créer une application de manière composable :

- [Monorepo](#) - Crée un projet au niveau racine qui gère les interdépendances entre les projets au sein du monorepo. Le projet fournit également une mise en cache des builds et une visualisation des dépendances.

- [API sécurisée de type](#) : crée une API qui peut être définie dans [Smithy](#) ou [OpenAPI v3](#), et gère la génération de code au moment de la construction pour vous permettre d'implémenter et d'interagir avec votre API de manière sécurisée. Vend une construction CDK qui gère le déploiement de votre API sur API Gateway et configure la validation automatique des entrées.
- Site Web [Cloudscape React : crée un site Web](#) basé sur React créé à l'aide de [Cloudscape qui](#) est préintégré à Cognito Auth et (éventuellement) à l'API que vous avez créée, ce qui vous permet d'appeler votre API en toute sécurité.
- [Infrastructure](#) : crée un projet qui met en place toute l'infrastructure liée au CDK nécessaire au déploiement de votre application. Il est également préconfiguré pour générer un diagramme basé sur votre code CDK à chaque fois que vous créez.
- [DevOps](#)- Crée DevOps des flux de travail compatibles avec les structures du kit de développement de projets AWS (AWS PDK).

Le didacticiel explique également comment afficher l'application déployée, inviter d'autres utilisateurs à travailler dessus et apporter des modifications au code à l'aide de demandes d'extraction qui sont automatiquement créées et déployées sur les ressources du compte AWS connecté lorsque la demande d'extraction est fusionnée.

Lorsque vous créez un projet composé de plans PDK, votre projet est créé avec les ressources suivantes dans un CodeCatalyst projet :

- Un référentiel source configuré en tant que monorepo.
- Un flux de travail qui exécute une analyse statique du code et une vérification des licences, ainsi que qui crée et déploie l'exemple de code chaque fois qu'une modification est apportée à la branche par défaut. Un schéma d'architecture est généré chaque fois que vous modifiez le code.
- Un tableau des problèmes et un carnet de dossiers que vous pouvez utiliser pour planifier et suivre le travail.
- Une suite de rapports de test avec des rapports automatisés.

## Rubriques

- [Prérequis](#)
- [Étape 1 : créer un projet monorepo](#)
- [Étape 2 : Ajouter l'API Type Safe au projet](#)
- [Étape 3 : ajouter un site Web Cloudscape React pour le projet](#)



- [Étape 4 : générer l'infrastructure pour déployer l'application sur le cloud AWS](#)
- [Étape 5 : configurer un DevOps flux de travail pour déployer votre projet](#)
- [Étape 6 : Confirmez le flux de publication et consultez votre site Web](#)
- [Collaborez et itérez sur le projet PDK](#)

## Prérequis

Pour créer et mettre à jour un projet, vous devez avoir effectué les tâches [Configuration CodeCatalyst](#) suivantes :

- Disposez d'un identifiant AWS Builder pour vous connecter à CodeCatalyst.
- Appartenez à un espace et le rôle d'administrateur de l'espace ou d'utilisateur avancé vous est attribué dans cet espace. Pour plus d'informations, consultez [Création d'un espace prenant en charge les utilisateurs de AWS Builder ID](#), [Gestion des utilisateurs de l'espace](#) et [Rôle d'administrateur de l'espace](#).
- Associez un compte AWS à votre espace et détenez le rôle IAM que vous avez créé lors de votre inscription. Par exemple, lors de l'inscription, vous avez la possibilité de choisir de créer un rôle de service avec une politique de rôle appelée politique de rôle CodeCatalystWorkflowDevelopmentRole- *SpaceName*. Le rôle portera un nom CodeCatalystWorkflowDevelopmentRole-*spaceName* avec un identifiant unique ajouté. Pour plus d'informations sur le rôle et la politique de rôle, consultez [Comprendre le rôle CodeCatalystWorkflowDevelopmentRole-\*spaceName\* du service](#). Pour connaître les étapes de création du rôle, consultez [Création du CodeCatalystWorkflowDevelopmentRole-\*spaceName\* rôle pour votre compte et votre espace](#).

## Étape 1 : créer un projet monorepo

Commencez par le plan PDK - Monorepo pour créer votre base de code monorepo qui servira de base, vous permettant d'ajouter des plans PDK supplémentaires.

Pour créer un projet à l'aide du plan PDK - Monorepo

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la CodeCatalyst console, accédez à l'espace dans lequel vous souhaitez créer un projet.
3. Sur le tableau de bord de l'espace, choisissez Créer un projet.
4. Choisissez Commencer par un plan.

5. Choisissez le plan PDK - Monorepo, puis choisissez Next.
6. Sous Nommer votre projet, entrez le nom que vous souhaitez attribuer à votre projet et les noms des ressources qui lui sont associés. Le nom doit être unique dans votre espace.
7. Sous Ressources du projet, procédez comme suit :
  - a. Sous Langage de programmation principal, choisissez le langage dans lequel vous souhaitez développer le code de votre projet. Vous pouvez choisir entre TypeScript Java ou Python.
  - b. Choisissez la configuration du code
  - c. Dans le champ de saisie de texte Référentiel source, entrez le nom d'un référentiel source, qui créera un nouveau référentiel, ou sélectionnez-le parmi un référentiel lié existant. Le référentiel existant doit être vide. Pour plus d'informations, consultez [Lier un référentiel source](#).
  - d. (Facultatif) Dans le menu déroulant Gestionnaire de packages, choisissez un gestionnaire de packages. Cela n'est nécessaire que si vous avez sélectionné TypeScript comme langage de programmation principal.
8. (Facultatif) Pour prévisualiser le code qui sera généré en fonction des paramètres de projet que vous avez sélectionnés, choisissez Afficher le code dans Générer un aperçu du projet.
9. (Facultatif) Choisissez Afficher les détails dans la fiche du plan pour afficher des détails spécifiques sur le plan, tels qu'un aperçu de l'architecture du plan, les connexions et autorisations requises, ainsi que le type de ressources créé par le plan.
10. Choisissez Create project pour créer votre projet monorepo. Le projet créé au niveau racine gère les interdépendances entre les projets au sein du monorepo, ainsi que la mise en cache des builds et la gestion des dépendances.

Pour plus d'informations sur les plans de projet, consultez [Référence du plan de projet](#).

Le plan PDK - Monorepo ne fait que créer les bases du projet. Pour créer une application fonctionnelle à l'aide du plan, vous devez ajouter d'autres plans PDK tels que l'API Type Safe, le site Web Cloudscape React, l'infrastructure ou. DevOps À l'étape suivante, vous allez appliquer une API Type Safe au projet.

## Étape 2 : Ajouter l'API Type Safe au projet

Le plan d'API PDK - Type Safe vous permet de définir une API à l'aide de Smithy ou d'OpenAI v3. Il génère des packages d'exécution à partir de la définition de votre API, qui incluent des clients

pour interagir avec votre API et du code côté serveur pour implémenter votre API. Le plan génère également une construction CDK avec une sécurité de type pour chaque opération d'API. Vous pouvez appliquer le plan à un projet monorepo PDK existant pour ajouter des fonctionnalités d'API au projet.

### Pour appliquer le modèle d'API PDK - Type Safe

1. Dans le volet de navigation de votre projet monorepo, choisissez Blueprints, puis choisissez Appliquer le blueprint.
2. Choisissez le plan d'API PDK - Type Safe, puis choisissez Next.
3. Sous Configurer le plan, configurez les paramètres du plan :
  - Sous Langage du modèle, choisissez la langue dans laquelle le modèle d'API est défini.
  - Dans le champ de saisie de texte Namespace, entrez un espace de noms pour votre API.
  - Dans le champ de saisie du nom de l'API, entrez le nom de votre API.
  - Dans Langage CDK, choisissez votre langue préférée pour écrire l'infrastructure CDK dans laquelle déployer l'API.
  - Choisissez le menu déroulant Langue (s) du gestionnaire, puis choisissez les langues dans lesquelles vous souhaitez implémenter les gestionnaires pour les opérations d'API.
  - Choisissez le menu déroulant Format (s) de documentation, puis choisissez les formats que vous souhaitez pour générer la documentation de l'API.
4. Dans l'onglet Modifications du code, passez en revue les modifications proposées. La différence affichée dans une pull request indique les modifications apportées à votre projet au moment de la création de la pull request.
5. Lorsque vous êtes satisfait des modifications proposées qui seront apportées lors de l'application du plan, choisissez Appliquer le plan.

Après avoir créé une pull request, vous pouvez ajouter des commentaires. Des commentaires peuvent être ajoutés à la pull request ou à des lignes individuelles des fichiers, ainsi qu'à la pull request globale. Vous pouvez ajouter des liens vers des ressources telles que des fichiers en utilisant le @ signe suivi du nom du fichier.

**Note**

Le plan ne sera pas appliqué tant que la pull request ne sera pas approuvée et fusionnée. Pour plus d'informations, consultez [Révision d'une pull request](#) et [Fusion d'une pull request](#).

6. Dans la colonne Status, choisissez Pending pull request for the PDK - Type Safe API blueprint row, puis choisissez le lien de la pull request ouverte.
7. Choisissez Fusionner, choisissez votre stratégie de fusion préférée, puis choisissez Fusionner pour intégrer les modifications du plan appliqué.

Une fois la pull request fusionnée, un nouveau `packages/apis/myjdkapi` dossier est généré dans votre projet monorepo, qui contient tout le code source lié à l'API Type Safe configurée.

8. Dans le volet de navigation, choisissez Blueprints pour confirmer que le statut de l'API PDK - Type Safe indique À jour.

## Étape 3 : ajouter un site Web Cloudscape React pour le projet

Le plan de site Web PDK - Cloudscape React génère un site Web. Vous pouvez associer un paramètre facultatif (API Type Safe) pour configurer automatiquement votre site Web afin de configurer des clients sécurisés authentifiés, ainsi qu'un explorateur d'API interactif pour tester vos différentes API.

Pour appliquer le plan du site Web PDK - Cloudscape React

1. Dans le volet de navigation de votre projet monorepo, choisissez Blueprints, puis choisissez Appliquer le blueprint.
2. Choisissez le plan du site Web PDK - Cloudscape React, puis choisissez Next.
3. Sous Configurer le plan, configurez les paramètres du plan :
  - Dans le champ de saisie du nom du site Web, entrez le nom de votre site Web.
  - Choisissez le menu déroulant Type Safe APIs, puis choisissez les plans d'API que vous souhaitez intégrer au site Web. La transmission d'une API configure les clients authentifiés et ajoute les dépendances requises, l'explorateur d'API et d'autres fonctionnalités.

4. Dans l'onglet Modifications du code, passez en revue les modifications proposées. La différence affichée dans une pull request indique les modifications apportées à votre projet au moment de la création de la pull request.
5. Lorsque vous êtes satisfait des modifications proposées qui seront apportées lors de l'application du plan, choisissez Appliquer le plan.

Après avoir créé une pull request, vous pouvez ajouter des commentaires. Des commentaires peuvent être ajoutés à la pull request ou à des lignes individuelles des fichiers, ainsi qu'à la pull request globale. Vous pouvez ajouter des liens vers des ressources telles que des fichiers en utilisant le @ signe suivi du nom du fichier.

#### Note

Le plan ne sera pas appliqué tant que la pull request ne sera pas approuvée et fusionnée. Pour plus d'informations, consultez [Révision d'une pull request](#) et [Fusion d'une pull request](#).

6. Dans la colonne État, choisissez Pending pull request for the PDK - Cloudscape React Website blueprint row, puis choisissez le lien de la pull request ouverte.
7. Choisissez Fusionner, choisissez votre stratégie de fusion préférée, puis choisissez Fusionner pour intégrer les modifications du plan appliqué.

Une fois la pull request fusionnée, un nouveau packages/websites/*my-website-name* dossier est généré dans votre projet monorepo, qui contient tout le code source de votre nouveau site Web.

8. Dans le volet de navigation, choisissez Blueprints pour confirmer que l'état du site Web PDK - Cloudscape React est affiché à jour.

Vous allez ensuite appliquer le PDK - Plan d'infrastructure pour générer l'infrastructure nécessaire au déploiement de votre site Web dans le cloud AWS.

## Étape 4 : générer l'infrastructure pour déployer l'application sur le cloud AWS

Le PDK - Infrastructure Blueprint met en place un package contenant l'ensemble de votre code CDK pour déployer votre site Web et votre API. Il fournit également la génération de diagrammes et la conformité au pack de prototypage par défaut.

## Pour appliquer le PDK - Plan d'infrastructure

1. Dans le volet de navigation de votre projet monorepo, choisissez Blueprints, puis choisissez Appliquer le blueprint.
2. Choisissez le PDK - Plan d'infrastructure, puis cliquez sur Next.
3. Sous Configurer le plan, configurez les paramètres du plan :
  - Dans Langage CDK, choisissez le langage avec lequel vous souhaitez développer votre infrastructure.
  - Dans le champ de saisie de texte Nom de la pile, entrez le nom de la CloudFormation pile générée pour votre plan.

### Note

Notez le nom de cette pile pour l'étape suivante, au cours de laquelle vous allez définir un DevOps flux de travail.

- Choisissez le menu déroulant Type Safe APIs, puis choisissez les plans d'API que vous souhaitez intégrer au site Web.
  - Choisissez le menu déroulant des sites Web de Cloudscape React TS, puis choisissez les plans de sites Web que vous souhaitez déployer au sein de votre infrastructure (par exemple, PDK - Cloudscape React Website).
4. Dans l'onglet Modifications du code, passez en revue les modifications proposées. La différence affichée dans une pull request indique les modifications apportées à votre projet au moment de la création de la pull request.
  5. Lorsque vous êtes satisfait des modifications proposées qui seront apportées lors de l'application du plan, choisissez Appliquer le plan.

Après avoir créé une pull request, vous pouvez ajouter des commentaires. Des commentaires peuvent être ajoutés à la pull request ou à des lignes individuelles des fichiers, ainsi qu'à la pull request globale. Vous pouvez ajouter des liens vers des ressources telles que des fichiers en utilisant le @ signe suivi du nom du fichier.

**Note**

Le plan ne sera pas appliqué tant que la pull request ne sera pas approuvée et fusionnée. Pour plus d'informations, consultez [Révision d'une pull request](#) et [Fusion d'une pull request](#).

6. Dans la colonne Status, choisissez Pending pull request for the PDK - Infrastructure blueprint row, puis choisissez le lien de la pull request ouverte.
7. Choisissez Fusionner, choisissez votre stratégie de fusion préférée, puis choisissez Fusionner pour intégrer les modifications du plan appliqué.

Une fois la pull request fusionnée, un nouveau packages/infra dossier est généré dans votre projet monorepo, qui contient l'infrastructure qui déploiera votre projet dans le cloud AWS.

8. Dans le volet de navigation, choisissez Blueprints pour confirmer que l'état du PDK - Infrastructure indique À jour.

Vous allez ensuite appliquer le DevOps plan PDK pour déployer votre application.


## Étape 5 : configurer un DevOps flux de travail pour déployer votre projet

Le PDK - DevOps Blueprint génère les DevOps flux de travail nécessaires pour créer et déployer votre projet à l'aide du compte et du rôle AWS spécifiés dans la configuration.

Pour appliquer le PDK - plan DevOps

1. Dans le volet de navigation de votre projet monorepo, choisissez Blueprints, puis choisissez Appliquer le blueprint.
2. Choisissez le PDK - DevOps Blueprint, puis cliquez sur Next.
3. Sous Configurer le plan, configurez les paramètres du plan :
  - Choisissez Bootstrap CDK dans l'environnement actuel.
  - Dans le champ de saisie du nom de la pile, entrez le nom de la CloudFormation pile que vous souhaitez déployer. Cela doit correspondre au nom de pile configuré dans [Étape 4 : générer l'infrastructure pour déployer l'application sur le cloud AWS](#) le PDK - Infrastructure Blueprint.


- Choisissez le menu déroulant de connexion au compte AWS, puis choisissez le compte AWS que vous souhaitez utiliser pour les ressources. Pour plus d'informations, consultez [Ajouter un Compte AWS à un espace](#).
- Choisissez le menu déroulant Le rôle à utiliser pour déployer votre application, puis choisissez le rôle IAM que vous souhaitez utiliser pour déployer l'application de votre projet.

 Note

Lorsque vous créez un rôle IAM, limitez-le SourceArn à ce qui ProjectID se trouve actuellement dans les paramètres du projet. Pour plus d'informations, consultez [Comprendre le rôle CodeCatalystWorkflowDevelopmentRole-\*spaceName\*du service](#).

- Choisissez le menu déroulant Région, puis choisissez la région dans laquelle vous souhaitez déployer votre projet monorepo. Le déploiement ne fonctionne que dans les régions où les services AWS requis existent. Pour plus d'informations, consultez la section [Services AWS par région](#).
4. Dans l'onglet Modifications du code, passez en revue les modifications proposées. La différence affichée dans une pull request indique les modifications apportées à votre projet au moment de la création de la pull request.
  5. Lorsque vous êtes satisfait des modifications proposées qui seront apportées lors de l'application du plan, choisissez Appliquer le plan.

Après avoir créé une pull request, vous pouvez ajouter des commentaires. Des commentaires peuvent être ajoutés à la pull request ou à des lignes individuelles des fichiers, ainsi qu'à la pull request globale. Vous pouvez ajouter des liens vers des ressources telles que des fichiers en utilisant le @ signe suivi du nom du fichier.

 Note

Le plan ne sera pas appliqué tant que la pull request ne sera pas approuvée et fusionnée. Pour plus d'informations, consultez [Révision d'une pull request](#) et [Fusion d'une pull request](#).

6. Dans la colonne Status, choisissez Pending pull request for the PDK - Infrastructure blueprint row, puis choisissez le lien de la pull request ouverte.
7. Choisissez Fusionner, choisissez votre stratégie de fusion préférée, puis choisissez Fusionner pour intégrer les modifications du plan appliqué.



Une fois la pull request fusionnée, un nouveau `.codecatalyst/workflows` dossier est généré dans votre projet monorepo.

8. Dans le volet de navigation, choisissez Blueprints pour confirmer que l'état du PDK DevOps indique « À jour ».

#### Note

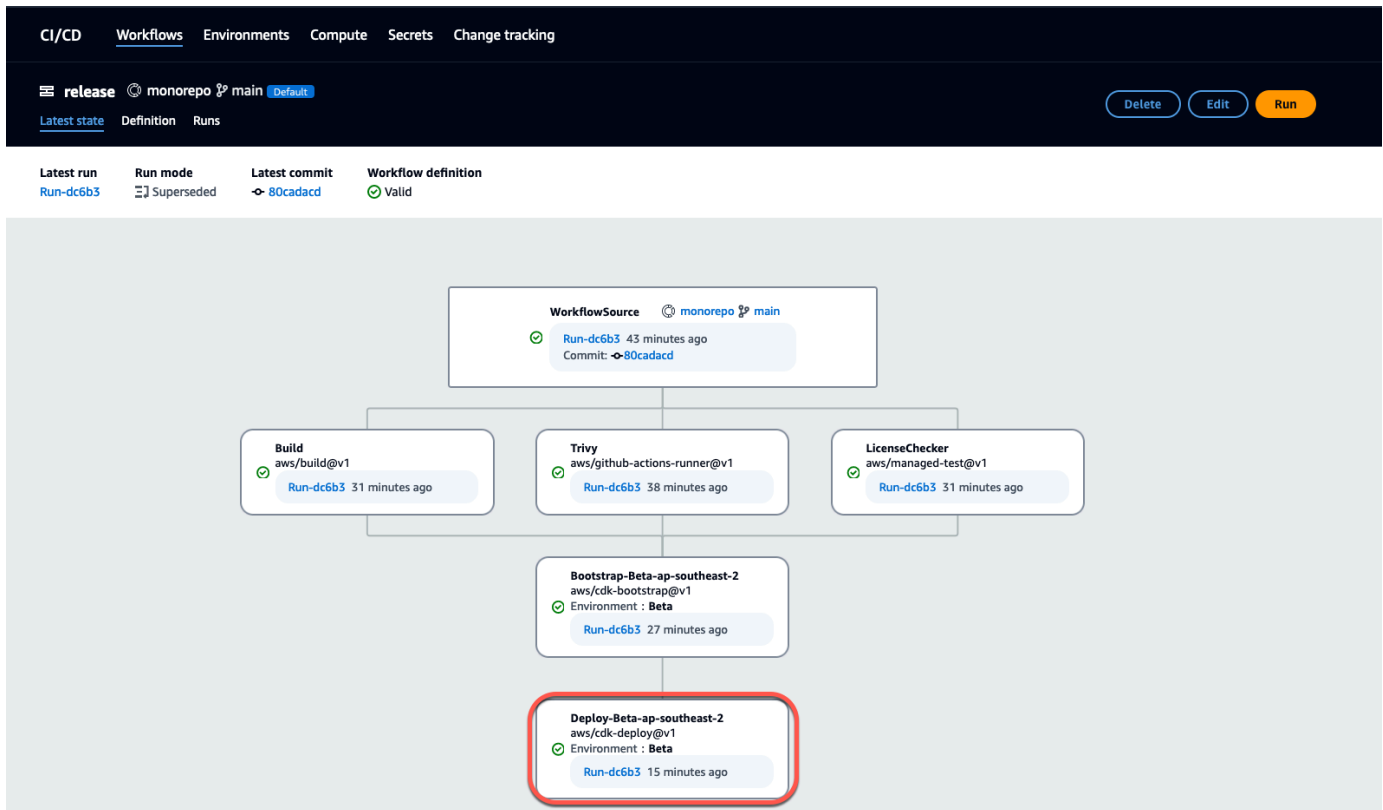
Le DevOps plan PDK et toutes les modifications ultérieures apportées aux plans PDK seront nettement plus lents à partir de maintenant, car des fichiers de verrouillage sont générés en arrière-plan pour garantir la reproductibilité des builds et des déploiements à l'avenir. Cela générera des fichiers de verrouillage pour tous les packages dans toutes les langues prises en charge.

## Étape 6 : Confirmez le flux de publication et consultez votre site Web

Une fois que vous avez terminé les étapes précédentes, vous pouvez confirmer le flux de travail de publication pour vous assurer que le projet est en cours de création.

Pour confirmer le flux de publication et consulter votre site Web

1. Dans le volet de navigation de votre projet monorepo, choisissez CI/CD, puis Workflows.
2. Pour le flux de travail de publication, choisissez le dernier flux de travail exécuté pour afficher les détails. Pour plus d'informations, consultez [Afficher le statut et les détails d'une seule exécution](#).
3. Une fois le flux de travail exécuté avec succès, choisissez la dernière action du flux de travail (par exemple, Déploy-B eta-ap-southeast -2), puis choisissez Variables.



- Affichez le site Web déployé en copiant et en collant le lien trouvé dans le tableau des variables (par exemple, **MyPDKAPI** websiteDistributionDomain NameXXXX) dans une nouvelle fenêtre de navigateur.

## Deploy-Beta-ap-southeast-2



✔ Succeeded Start time: about 13 hours ago | Duration: 9 minutes 51 seconds


Logs

Summary

Configuration

Variables

## Output variables (7)

Name ▲	Value ▼
CalculateApiEndpoint1B9112D8	https://iczdb3kx34.execute-api.ap-southeast-2.amazonaws.com/prod/
CalculatewebsiteDistributionDomainName5F8EAA19	d1c3j5sbejrjio.cloudfront.net
deployment-platform	AWS:CloudFormation
infracalculatebetaUserIdentityinfracalculatebetaUserIdentityIdentityPoolIdB56E5D31	ap-southeast-2:719e759a-8dcb-4113-a9eb-687cb0b65f0d
infracalculatebetaUserIdentityinfracalculatebetaUserIdentityUserPoolId380E2DD7	ap-southeast-2_aDUKfIH4p
region	ap-southeast-2
stack-id	 arn:aws:cloudformation:ap-southeast-2:78062387952:1:stack/infra-calculate-beta/f0220560-f470-11ee-940e-065f17dab4c7

Vous avez besoin d'un compte Amazon Cognito pour vous connecter à votre site Web. Par défaut, le groupe d'utilisateurs n'est pas configuré pour autoriser l'auto-enregistrement.

- a. Accédez à la console [AWS Cognito](#).
- b. *Dans le tableau **Groupes d'utilisateurs**, choisissez le nom du groupe d'utilisateurs qui correspond au groupe d'utilisateurs créé par le PDK - DevOps blueprint, qui se trouve dans le tableau Variables (par exemple, **infra**, **calculer**, **calculerbetaUserIdentityinfra**, **XXXXX**).*  
betaUserIdentity IdentityPoolId Pour plus d'informations, voir [Commencer à utiliser les groupes d'utilisateurs](#).

## Deploy-Beta-ap-southeast-2



✔ Succeeded Start time: about 13 hours ago | Duration: 9 minutes 51 seconds

Logs

Summary

Configuration

Variables

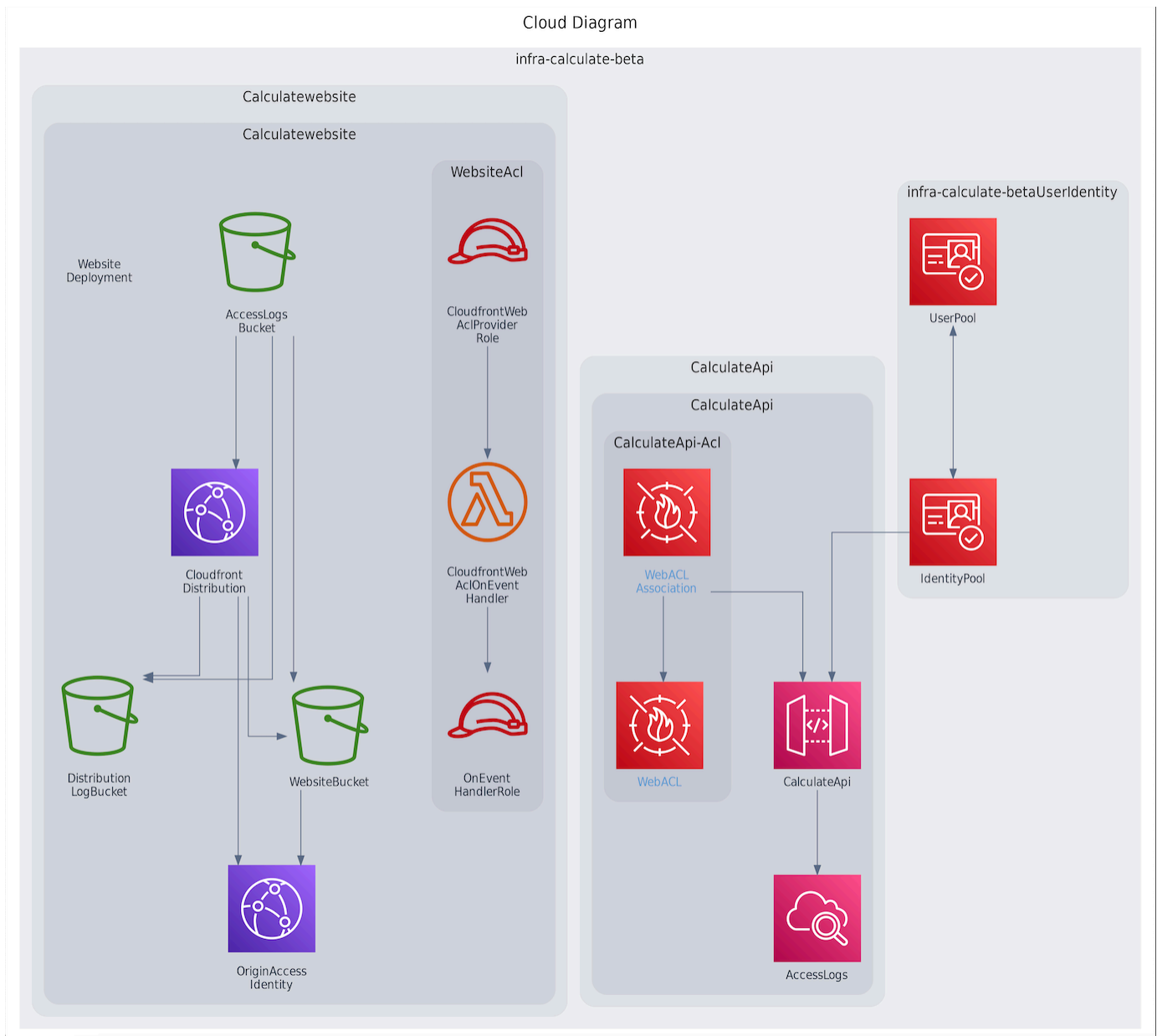
## Output variables (7)

Name ▲	Value ▼
CalculateApiEndpoint1B9112D8	https://icfdb3kx34.execute-api.ap-southeast-2.amazonaws.com/prod/
CalculatewebsiteDistributionDomainName5F8EAA19	d1c3j5sbejrjio.cloudfront.net
deployment-platform	AWS:CloudFormation
infracalculatebetaUserIdentityUserPoolId3dB56E5D31	ap-southeast-2:719e759a-8dcb-4113-a9eb-687cb0b65f0d
infracalculatebetaUserIdentityUserPoolId380E2DD7	ap-southeast-2_aDUKfIH4p
region	ap-southeast-2
stack-id	arn:aws:cloudformation:ap-southeast-2:78062387952:1:stack/infra-calculate-beta/f0220560-f470-11ee-940e-065f17dab4c7

- c. Choisissez Create user (Créer un utilisateur).
  - d. Configurez les paramètres des informations utilisateur :
    - Sous Message d'invitation, choisissez Envoyer une invitation par e-mail.
    - Dans le champ de saisie du nom d'utilisateur, entrez un nom d'utilisateur.
    - Dans le champ de saisie de texte de l'adresse e-mail, entrez un nom d'utilisateur.
    - Sous Mot de passe temporaire, choisissez Générer un mot de passe.
  - e. Choisissez Create user (Créer un utilisateur).
  - f. Accédez au compte e-mail que vous avez saisi pour les paramètres d'informations utilisateur, ouvrez un e-mail avec un mot de passe temporaire. Notez le mot de passe.
  - g. Revenez au site Web déployé, entrez le nom d'utilisateur que vous avez créé et le mot de passe temporaire que vous avez reçu, puis choisissez Se connecter.
5. (Facultatif) Une fois le flux de travail exécuté avec succès, vous pouvez également afficher le diagramme généré. Choisissez l'onglet Artefacts dans CodeCatalyst, choisissez Télécharger pour la ligne du diagramme, puis ouvrez le ou les fichiers téléchargés.

The screenshot shows the Amazon CodeCatalyst interface for a workflow named 'Run-ef953'. The 'Artifacts' tab is selected, displaying a table of artifacts. The 'Diagram' artifact is highlighted with a red circle. The table has columns for Artifact name, Files, Produced by, and Consumed by. The 'Diagram' artifact has a 'Download' link next to it.

Artifact name	Files	Produced by	Consumed by
Built	<a href="#">Download</a>	Build	Bootstrap-Beta-ap-southeast-2 Deploy-Beta-ap-southeast-2
Diagram	<a href="#">Download</a>	Build	-
bdd254b65baac169f6ac50e8175ce6d930c1fcb086dec59808d3e0170ae2291d_report	<a href="#">Download</a>	Build	-
a093422585a8a4cb763d89a0fa8e76744a80830fe24724c7e7943a50ec479240_report	<a href="#">Download</a>	Trivy	-



## Collaborez et itérez sur le projet PDK

Une fois votre projet configuré, vous pouvez apporter des modifications au code source. Vous pouvez également inviter d'autres membres de l'espace à travailler sur le projet. Les plans PDK vous permettent de créer votre application de manière itérative, en ajoutant uniquement ce dont vous avez besoin, quand vous en avez besoin, tout en gardant le contrôle total de la configuration de chaque plan.

### Rubriques

- [Étape 1 : Invitez des membres à rejoindre votre projet](#)
- [Étape 2 : créer des problèmes pour collaborer et suivre le travail](#)
- [Étape 3 : Afficher votre référentiel source](#)
- [Étape 4 : créer un environnement de développement et apporter des modifications au code](#)
- [Étape 5 : Transférez et fusionnez les modifications de code](#)

## Étape 1 : Invitez des membres à rejoindre votre projet

Vous pouvez utiliser la console pour inviter des utilisateurs à rejoindre votre projet. Vous pouvez inviter des membres de votre espace ou ajouter des noms provenant de l'extérieur de votre espace.

Pour inviter des utilisateurs à rejoindre votre projet, vous devez être connecté avec le rôle d'administrateur du projet ou d'administrateur de l'espace.

Il n'est pas nécessaire d'inviter un utilisateur ayant le rôle d'administrateur de l'espace à rejoindre votre projet, car il dispose déjà d'un accès implicite à tous les projets de l'espace.

Lorsque vous invitez un utilisateur à rejoindre votre projet (sans lui attribuer le rôle d'administrateur de l'espace), l'utilisateur apparaît dans le tableau des membres du projet sous les projets et dans le tableau des membres du projet sous les espaces.

Pour inviter un membre à rejoindre votre projet depuis l'onglet Paramètres du projet

1. Accédez à votre projet.

### Tip

Vous pouvez choisir le projet à afficher dans la barre de navigation supérieure.

2. Dans le volet de navigation, sélectionnez Paramètres du projet.
3. Choisissez l'onglet Membres.
4. Dans Membres du projet, sélectionnez Inviter un nouveau membre.
5. Entrez l'adresse e-mail du nouveau membre, choisissez le rôle de ce membre, puis choisissez Inviter. Pour plus d'informations sur les rôles, consultez [Utilisation des rôles sur Amazon CodeCatalyst](#).



Pour inviter un membre à rejoindre votre projet depuis la page d'aperçu du projet

1. Accédez à votre projet.

 Tip

Vous pouvez choisir le projet à afficher dans la barre de navigation supérieure.


2. Cliquez sur le bouton Membres +.
3. Entrez l'adresse e-mail du nouveau membre, choisissez le rôle de ce membre, puis choisissez Inviter. Pour plus d'informations sur les rôles, consultez [Utilisation des rôles sur Amazon CodeCatalyst](#).

## Étape 2 : créer des problèmes pour collaborer et suivre le travail

CodeCatalyst vous aide à suivre les fonctionnalités, les tâches, les bogues et tout autre travail impliqué dans votre projet présentant des problèmes. Vous pouvez créer des problèmes pour suivre le travail et les idées nécessaires. Par défaut, lorsque vous créez un problème, celui-ci est ajouté à votre backlog. Vous pouvez déplacer les problèmes vers un tableau où vous pouvez suivre le travail en cours. Vous pouvez également attribuer un problème à un membre du projet en particulier. Au cours de cette étape, créez un problème pour apporter des modifications à votre projet PDK.

Pour créer un problème

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre projet monorepo dans lequel vous souhaitez créer un problème.
3. Sur la page d'accueil du projet, choisissez Créer un problème. Sinon, dans le volet de navigation, sélectionnez Problèmes.
4. Choisissez Créer un problème.

 Note

Vous pouvez également ajouter des problèmes en ligne lorsque vous utilisez une vue en grille.

5. Entrez le titre du numéro.

6. (Facultatif) Entrez une description. Pour ce problème, entrez la description suivante : `a change in the src/mysfit_data.json file`. Vous pouvez utiliser Markdown pour ajouter du formatage.
7. (Facultatif) Choisissez un statut, une priorité ou une estimation pour le problème.
8. (Facultatif) Ajoutez une étiquette existante ou créez-en une nouvelle et ajoutez-la en choisissant + Ajouter une étiquette.
  - a. Pour ajouter une étiquette existante, choisissez-la dans la liste. Vous pouvez saisir un terme de recherche dans le champ pour rechercher toutes les étiquettes contenant ce terme dans le projet.
  - b. Pour créer une nouvelle étiquette et l'ajouter, entrez le nom de l'étiquette que vous souhaitez créer dans le champ de recherche et appuyez sur Entrée.
9. (Facultatif) Ajoutez un destinataire en choisissant + Ajouter un destinataire. Vous pouvez rapidement vous ajouter en tant que cessionnaire en choisissant + M'ajouter.

 Tip

Vous pouvez choisir d'attribuer un problème à Amazon Q pour qu'Amazon Q essaie de le résoudre. Pour plus d'informations, consultez [Tutoriel : Utiliser les fonctionnalités CodeCatalyst génératives de l'IA pour accélérer votre travail de développement](#). Cette fonctionnalité nécessite que les fonctionnalités d'IA génératives soient activées pour l'espace. Pour plus d'informations, consultez [la section Gestion des fonctionnalités d'IA générative](#).

10. (Facultatif) Ajoutez un champ personnalisé existant ou créez-en un nouveau. Les problèmes peuvent comporter plusieurs champs personnalisés.
  - a. Pour ajouter un champ personnalisé existant, choisissez-le dans la liste. Vous pouvez saisir un terme de recherche dans le champ pour rechercher tous les champs personnalisés contenant ce terme dans le projet.
  - b. Pour créer un nouveau champ personnalisé et l'ajouter, entrez le nom du champ personnalisé que vous souhaitez créer dans le champ de recherche et appuyez sur Entrée. Choisissez ensuite le type de champ personnalisé que vous souhaitez créer et définissez une valeur.
11. Choisissez Créer un problème. Une notification apparaît dans le coin inférieur droit : si le problème a été créé avec succès, un message de confirmation apparaît indiquant que le

problème a été créé avec succès. Si le problème n'a pas été créé correctement, un message d'erreur indiquant la raison de l'échec apparaît. Vous pouvez ensuite choisir Réessayer pour modifier et réessayer de créer le problème, ou choisir Ignorer pour ignorer le problème. Les deux options annuleront la notification.

#### Note

Vous ne pouvez pas lier une pull request à un problème lorsque vous la créez. Cependant, vous pouvez le [modifier après l'avoir créé](#) pour ajouter des liens vers des pull requests.

Pour plus d'informations, consultez [Problèmes liés à CodeCatalyst](#).

## Étape 3 : Afficher votre référentiel source

Vous pouvez consulter les référentiels de sources associés à un projet sur Amazon CodeCatalyst. Pour les référentiels sources dans CodeCatalyst, la page de présentation d'un référentiel fournit un aperçu rapide des informations et des activités dans ce référentiel, notamment :

- Description du référentiel, le cas échéant
- Le nombre de branches dans le référentiel
- Le nombre de pull requests ouvertes pour le référentiel
- Le nombre de flux de travail associés pour le référentiel
- Les fichiers et dossiers de la branche par défaut ou de la branche que vous choisissez
- Le titre, l'auteur et la date du dernier commit dans la branche affichée
- Le contenu du fichier README.md affiché dans Markdown, si un fichier README.md est inclus

Cette page fournit également des liens vers les commits, les branches et les pull requests du référentiel, ainsi qu'un moyen rapide d'ouvrir, de visualiser et de modifier des fichiers individuels.

#### Note

Vous ne pouvez pas afficher ces informations sur les référentiels liés dans la CodeCatalyst console. Pour afficher des informations sur les référentiels liés, cliquez sur le lien dans la liste des référentiels pour ouvrir ce référentiel dans le service qui l'héberge.

## Pour accéder aux référentiels sources d'un projet

1. Accédez à votre projet, puis effectuez l'une des opérations suivantes :
  - Sur la page de résumé de votre projet, choisissez le référentiel souhaité dans la liste, puis choisissez Afficher le référentiel.
  - Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources. Dans Référentiels sources, choisissez le nom du référentiel dans la liste. Vous pouvez filtrer la liste des référentiels en saisissant une partie du nom du référentiel dans la barre de filtre.
2. Sur la page d'accueil du référentiel, consultez le contenu du référentiel et les informations sur les ressources associées, telles que le nombre de pull requests et les flux de travail. Par défaut, le contenu de la branche par défaut est affiché. Vous pouvez modifier l'affichage en choisissant une autre branche dans la liste déroulante.

### Tip

Vous pouvez également accéder rapidement aux référentiels de votre projet en choisissant Voir le code du projet sur la page de résumé du projet.


## Étape 4 : créer un environnement de développement et apporter des modifications au code

Au cours de cette étape, créez un environnement de développement et apportez des modifications au code qui sont ensuite fusionnées dans la branche principale. Ce didacticiel vous présente un projet AWS PDK simple, mais vous pouvez également suivre un exemple plus complexe fourni dans le référentiel [AWS PDK GitHub](#).

Pour créer un environnement de développement avec une nouvelle branche

1. Dans le volet de navigation de votre projet monorepo, effectuez l'une des opérations suivantes :
  - Choisissez Overview, puis accédez à la section Mes environnements de développement.
  - Choisissez Code, puis Dev Environments.
  - Choisissez Code, choisissez Référentiels sources, puis choisissez le référentiel monorepo pour lequel vous souhaitez créer un environnement de développement.


2. Choisissez un IDE compatible dans le menu déroulant. Pour plus d'informations, consultez [Environnements de développement intégrés pris en charge pour les environnements de développement](#).
3. Choisissez Cloner un dépôt.
4. Choisissez le référentiel à cloner, choisissez Travailler dans une nouvelle branche, entrez le nom de la branche dans le champ Nom de la branche et choisissez une branche à partir de laquelle créer la nouvelle branche dans le menu déroulant Créer une branche depuis.

 Note

Si vous créez un environnement de développement à partir de la page Référentiels sources ou à partir d'un référentiel source spécifique, vous n'avez pas besoin de choisir un référentiel. L'environnement de développement sera créé à partir du référentiel source que vous avez choisi sur la page Référentiels source.

5. (Facultatif) Dans Alias - facultatif, entrez un alias pour l'environnement de développement.
6. (Facultatif) Cliquez sur le bouton d'édition de la configuration de l'environnement de développement pour modifier la configuration de calcul, de stockage ou de temporisation de l'environnement de développement.
7. (Facultatif) Dans Amazon Virtual Private Cloud (Amazon VPC), sélectionnez une connexion VPC que vous souhaitez associer à votre environnement de développement dans le menu déroulant.

Si un VPC par défaut est défini pour votre espace, vos environnements de développement s'exécuteront connectés à ce VPC. Vous pouvez contourner cela en associant une autre connexion VPC. Notez également que les environnements de développement connectés au VPC ne sont pas compatibles avec AWS Toolkit.

 Note

Lorsque vous créez un environnement de développement avec une connexion VPC, une nouvelle interface réseau est créée au sein du VPC. CodeCatalyst interagit avec cette interface en utilisant le rôle VPC associé. Assurez-vous également que votre bloc d'adresse CIDR IPv4 n'est pas configuré pour la plage d'adresses 172.16.0.0/12 IP.

8. Choisissez Créer. Pendant la création de votre environnement de développement, la colonne d'état de l'environnement de développement affichera Démarrage, et la colonne d'état affichera En cours d'exécution une fois l'environnement de développement créé.

Une fois que votre environnement de développement est en cours d'exécution, vous pouvez utiliser l'exemple d'application que vous avez généré en CodeCatalyst modifiant le code avec des pull requests qui sont automatiquement créées et déployées sur les ressources du compte AWS connecté lorsque la pull request est fusionnée. Le monorepo vend un fichier de développement afin que toutes les dépendances et tous les environnements d'exécution globaux requis soient automatiquement présents.

Pour modifier le code de votre projet

1. Dans un terminal de travail de votre environnement de développement, accédez à votre projet monorepo, puis installez les dépendances de votre projet en exécutant la commande suivante :

```
npx projen install
```

2. Accédez à `packages/apis/myjdkapi/model/src/main/smithy/operations/say-hello.smithy`, qui définit un exemple d'opération d'API. Dans ce didacticiel, vous allez créer une `Calculate` opération simple qui consiste à additionner deux nombres. Modifiez le code pour définir cette opération, y compris son entrée et sa sortie.

Exemple :

```
$version: "2"
namespace com.aws

@http(method: "POST", uri: "/calculate")
@handler(language: "typescript")
operation Calculate {
  input := {
    @required
    numberA: Integer
    @required
    numberB: Integer
  }
  output := {
    @required
    result: Integer
  }
}
```

Le `@handler` trait indique à l'API Type Safe que vous allez implémenter cette opération en tant que gestionnaire AWS Lambda écrit en TypeScript. L'API Type Safe générera un stub pour cette

opération dans TypeScript lequel vous pourrez l'implémenter. Le `@required` trait est ajouté, ce qui signifie qu'il sera appliqué au moment de l'exécution par la passerelle d'API déployée. Pour plus d'informations, consultez la [documentation de Smithy](#).

3. Renommez le `/say-hello.smithy` nom du fichier avec un nom qui correspond aux modifications de votre code (par exemple, `calculate.smithy`).
4. Accédez à `packages/apis/myjdkapi/model/src/main/smithy/main.smithy`, et modifiez le code pour connecter l'opération. Vous pouvez exposer l'`Calculate` opération définie dans le `/calculate.smithy` en la listant dans le `operations` champ de ce fichier.

Exemple :

```
$version: "2"
namespace com.aws

use aws.protocols#restJson1

/// A sample smithy api
@restJson1
service MyPDKApi {
  version: "1.0"
  operations: [Calculate]
  errors: [
    BadRequestError
    NotAuthorizedError
    InternalFailureError
  ]
}
```

5. Créez les modifications en exécutant la commande suivante :

```
npx projen build
```

#### Note

Vous pouvez éventuellement transmettre un `--parallel X` drapeau, qui répartira le build entre les X cœurs.

Depuis l'ajout du `@handler` trait, les fichiers suivants sont générés une fois la construction terminée :

- `/packages/apis/myjdkapi/handlers/typescript/src/calculate.ts`
- `/packages/apis/myjdkapi/handlers/typescript/test/calculate.test.ts`

6. Accédez au `packages/apis/myjdkapi/handlers/typescript/src/calculate.ts` code et modifiez-le. Ce fichier est le gestionnaire de serveur invoqué pour l'API.

```
import {
  calculateHandler,
  CalculateChainedHandlerFunction,
  INTERCEPTORS,
  Response,
  LoggingInterceptor,
} from 'myjdkapi-typescript-runtime';

/**
 * Type-safe handler for the Calculate operation
 */
export const calculate: CalculateChainedHandlerFunction = async (request) => {
  LoggingInterceptor.getLogger(request).info('Start Calculate Operation');

  const { input } = request;

  return Response.success({
    result: input.body.numberA + input.body.numberB,
  });
};

/**
 * Entry point for the AWS Lambda handler for the Calculate operation.
 * The calculateHandler method wraps the type-safe handler and manages marshalling
 * inputs and outputs
 */
export const handler = calculateHandler(...INTERCEPTORS, calculate);
```

7. Accédez au `/packages/apis/myjdkapi/handlers/typescript/test/calculate.test.ts` fichier et modifiez le code pour mettre à jour les tests unitaires.

Exemple :



```
import {
  CalculateChainedRequestInput,
  CalculateResponseContent,
} from 'mypdkapi-typescript-runtime';
import {
  calculate,
} from '../src/calculate';

// Common request arguments
const requestArguments = {
  chain: undefined as never,
  event: {} as any,
  context: {} as any,
  interceptorContext: {
    logger: {
      info: jest.fn(),
    },
  },
} satisfies Omit<CalculateChainedRequestInput, 'input'>;

describe('Calculate', () => {

  it('should return correct sum', async () => {
    const response = await calculate({
      ...requestArguments,
      input: {
        requestParameters: {},
        body: {
          numberA: 1,
          numberB: 2
        }
      },
    });

    expect(response.statusCode).toBe(200);
    expect((response.body as CalculateResponseContent).result).toEqual(3);
  });
});
```

8. Accédez au `/packages/infra/main/src/constructs/apis/mypdkapi.ts` fichier et modifiez le code pour ajouter une intégration pour l'Calculate opération dans votre

infrastructure CDK. La construction de l'API possède une propriété d'intégration, dans laquelle vous pouvez transmettre l'implémentation que vous avez ajoutée précédemment. Puisque vous utilisez le `@handler` trait dans votre modèle Smithy pour l'opération `Calculate`, vous pouvez utiliser la construction `CalculateFunction` CDK générée, qui est préconfigurée, pour pointer vers l'implémentation de votre gestionnaire.

Exemple :

```
import { UserIdentity } from "@aws/pdk/identity";
import { Authorizers, Integrations } from "@aws/pdk/type-safe-api";
import { Stack } from "aws-cdk-lib";
import { Cors } from "aws-cdk-lib/aws-apigateway";
import {
  AccountPrincipal,
  AnyPrincipal,
  Effect,
  PolicyDocument,
  PolicyStatement,
} from "aws-cdk-lib/aws-iam";
import { Construct } from "constructs";
import { Api, CalculateFunction } from "calculateapi-typescript-infra";

/**
 * Api construct props.
 */
export interface CalculateApiProps {
  /**
   * Instance of the UserIdentity.
   */
  readonly userIdentity: UserIdentity;
}

/**
 * Infrastructure construct to deploy a Type Safe API.
 */
export class CalculateApi extends Construct {
  /**
   * API instance
   */
  public readonly api: Api;

  constructor(scope: Construct, id: string, props?: CalculateApiProps) {
    super(scope, id);
  }
}
```

```
this.api = new Api(this, id, {
  defaultAuthorizer: Authorizers.iam(),
  corsOptions: {
    allowOrigins: Cors.ALL_ORIGINS,
    allowMethods: Cors.ALL_METHODS,
  },
  integrations: {
    calculate: {
      integration: Integrations.lambda(new CalculateFunction(this,
"CalculateFunction"))
    }
  },
  policy: new PolicyDocument({
    statements: [
      // Here we grant any AWS credentials from the account that the prototype
is deployed in to call the api.
      // Machine to machine fine-grained access can be defined here using more
specific principals (eg roles or
      // users) and resources (ie which api paths may be invoked by which
principal) if required.
      // If doing so, the cognito identity pool authenticated role must still
be granted access for cognito users to
      // still be granted access to the API.
      new PolicyStatement({
        effect: Effect.ALLOW,
        principals: [new AccountPrincipal(Stack.of(this).account)],
        actions: ["execute-api:Invoke"],
        resources: ["execute-api:/*"],
      }),
      // Open up OPTIONS to allow browsers to make unauthenticated preflight
requests
      new PolicyStatement({
        effect: Effect.ALLOW,
        principals: [new AnyPrincipal()],
        actions: ["execute-api:Invoke"],
        resources: ["execute-api:/*/OPTIONS/*"],
      }),
    ],
  }),
});

// Grant authenticated users access to invoke the api
props?.userIdentity.identityPool.authenticatedRole.addToPrincipalPolicy(
```

```
    new PolicyStatement({
      effect: Effect.ALLOW,
      actions: ["execute-api:Invoke"],
      resources: [this.api.api.arnForExecuteApi("*", "/*", "*")],
    }),
  );
}
```

9. Créez les modifications en exécutant la commande suivante :

```
npx projen build
```

Une fois la construction de votre projet terminée, vous pouvez consulter le schéma généré mis à jour, qui se trouve dans `le/packages/infra/main/cdk.out/cdkgraph/diagram.png`. Le schéma montre comment la fonction est ajoutée et connectée à l'API créée. Au fur et à mesure que le code CDK est modifié, ce diagramme est également mis à jour.

Vous pouvez désormais déployer vos modifications en les transférant et en les fusionnant dans la branche principale de votre dépôt.

## Étape 5 : Transférez et fusionnez les modifications de code

Validez et publiez vos modifications de code, qui peuvent ensuite être fusionnées dans la branche principale de votre dépôt source.

Pour appliquer les modifications à votre branche de fonctionnalités

- Validez et envoyez les modifications à votre branche de fonctionnalités en exécutant les commandes suivantes :

```
git add .
```

```
git commit -m "my commit message"
```

```
git push
```

Le fait d'apporter des modifications déclenche l'exécution d'un nouveau flux de travail pour votre branche de fonctionnalités, que vous pouvez consulter dans la CodeCatalyst console. Vous pouvez ensuite créer une pull request pour fusionner les modifications apportées à la branche principale de votre dépôt source. La fusion de la branche des fonctionnalités avec votre branche principale déclenche le flux de production. Vous pouvez également associer la pull request à votre problème.

Pour créer une pull request et l'associer à votre problème


1. Dans votre projet monorepo, effectuez l'une des opérations suivantes :
  - Dans le volet de navigation, choisissez Code, choisissez Pull requests, puis Create pull request.
  - Sur la page d'accueil du référentiel, choisissez Plus, puis Create pull request.
  - Sur la page du projet, choisissez Create pull request.
2. Dans le référentiel source, assurez-vous que le référentiel source spécifié est celui qui contient le code validé. Cette option n'apparaît que si vous n'avez pas créé la pull request depuis la page principale du dépôt.
3. Dans Branche de destination, choisissez la branche principale dans laquelle vous souhaitez fusionner le code une fois celui-ci révisé.
4. Dans Branche source, choisissez la branche de fonctionnalité qui contient le code validé.
5. Dans Titre de la demande d'extraction, entrez un titre qui aide les autres utilisateurs à comprendre ce qui doit être revu et pourquoi.
6. (Facultatif) Dans la description de la Pull request, fournissez des informations telles qu'un lien vers les problèmes ou une description de vos modifications.

 Tip

Vous pouvez choisir Write description for me afin de générer CodeCatalyst automatiquement une description des modifications contenues dans la pull request. Vous pouvez apporter des modifications à la description générée automatiquement après l'avoir ajoutée à la pull request.

Cette fonctionnalité nécessite que les fonctionnalités d'IA génératives soient activées pour l'espace. Pour plus d'informations, consultez [la section Gestion des fonctionnalités d'IA générative sur Amazon CodeCatalyst](#).

7. Dans Problèmes, choisissez Lier les problèmes, puis choisissez le problème que vous avez créé dans [Étape 2 : créer des problèmes pour collaborer et suivre le travail](#). Pour dissocier un problème, cliquez sur l'icône de dissociation.
8. (Facultatif) Dans Réviseurs obligatoires, sélectionnez Ajouter les réviseurs requis. Choisissez parmi la liste des membres du projet pour les ajouter. Les réviseurs requis doivent approuver les modifications avant que la pull request puisse être fusionnée dans la branche de destination.

 Note

Vous ne pouvez pas ajouter un réviseur à la fois comme réviseur obligatoire et comme réviseur facultatif. Vous ne pouvez pas vous ajouter en tant que réviseur.

9. (Facultatif) Dans Réviseurs facultatifs, sélectionnez Ajouter des réviseurs facultatifs. Choisissez parmi la liste des membres du projet pour les ajouter. Les réviseurs facultatifs n'ont pas à approuver les modifications comme condition préalable pour que la pull request puisse être fusionnée dans la branche de destination.
10. Votre pull request doit être revue et fusionnée dans la branche principale par les réviseurs ou par vous-même. Pour plus d'informations, consultez [Fusion d'une pull request](#).

Lorsque vos modifications sont fusionnées dans la branche principale de votre référentiel source, un nouveau flux de travail est automatiquement déclenché.

11. Une fois la fusion terminée, vous pouvez déplacer votre problème vers Terminé.
  - a. Dans le volet de navigation, sélectionnez Problèmes.
  - b. Choisissez le problème créé dans [Étape 2 : créer des problèmes pour collaborer et suivre le travail](#), choisissez la liste déroulante État, puis cliquez sur Terminé.

Le flux de production déploie votre application après une exécution réussie, afin que vous puissiez visualiser les modifications.

Pour confirmer le flux de publication et consulter votre site Web

1. Dans le volet de navigation de votre projet monorepo, choisissez CI/CD, puis Workflows.
2. Pour le flux de travail de publication, choisissez le dernier flux de travail exécuté pour afficher les détails. Pour plus d'informations, consultez [Afficher le statut et les détails d'une seule exécution](#).
3. Une fois le flux de travail exécuté avec succès, choisissez la dernière action du flux de travail (Déploy-B eta-ap-souteast -2), puis choisissez Variables.

4. Affichez le site Web déployé en copiant et en collant le lien de la ligne ***MyPDKAPI*** `websiteDistributionDomain NameXXXXX` dans une nouvelle fenêtre de navigateur.
5. Entrez le nom d'utilisateur et le mot de passe que vous avez créés [Étape 6 : Confirmez le flux de publication et consultez votre site Web](#), puis choisissez Se connecter.
6. (Facultatif) Testez les modifications dans votre application.
  - a. Choisissez le menu déroulant POST.
  - b. Entrez deux valeurs pour `numberA` et `number B`, puis choisissez Exécute.
  - c. Confirmez les résultats dans le corps de la réponse.

Au fil du temps, les versions du catalogue des plans PDK peuvent changer. Vous pouvez modifier les plans de votre projet en fonction des versions du catalogue pour rester au courant des dernières modifications. Vous pouvez consulter les modifications du code et les environnements concernés avant de modifier les versions du plan de votre projet. Pour plus d'informations, voir [Mettre à jour un plan dans un projet](#).

# Espaces dans CodeCatalyst

Vous créez un espace qui vous représente, ainsi que votre entreprise, votre département ou votre groupe, et qui fournit un endroit où vos équipes de développement peuvent gérer des projets. Vous devez créer un espace pour ajouter des projets, des membres et les ressources cloud associées que vous créez sur Amazon CodeCatalyst.

## Note

Les noms des espaces doivent être uniques d'un côté à l'autre CodeCatalyst. Vous ne pouvez pas réutiliser les noms des espaces supprimés.

Lorsque vous créez un espace, le rôle d'administrateur de l'espace vous est automatiquement attribué. Vous pouvez ajouter ce rôle aux autres utilisateurs de l'espace.

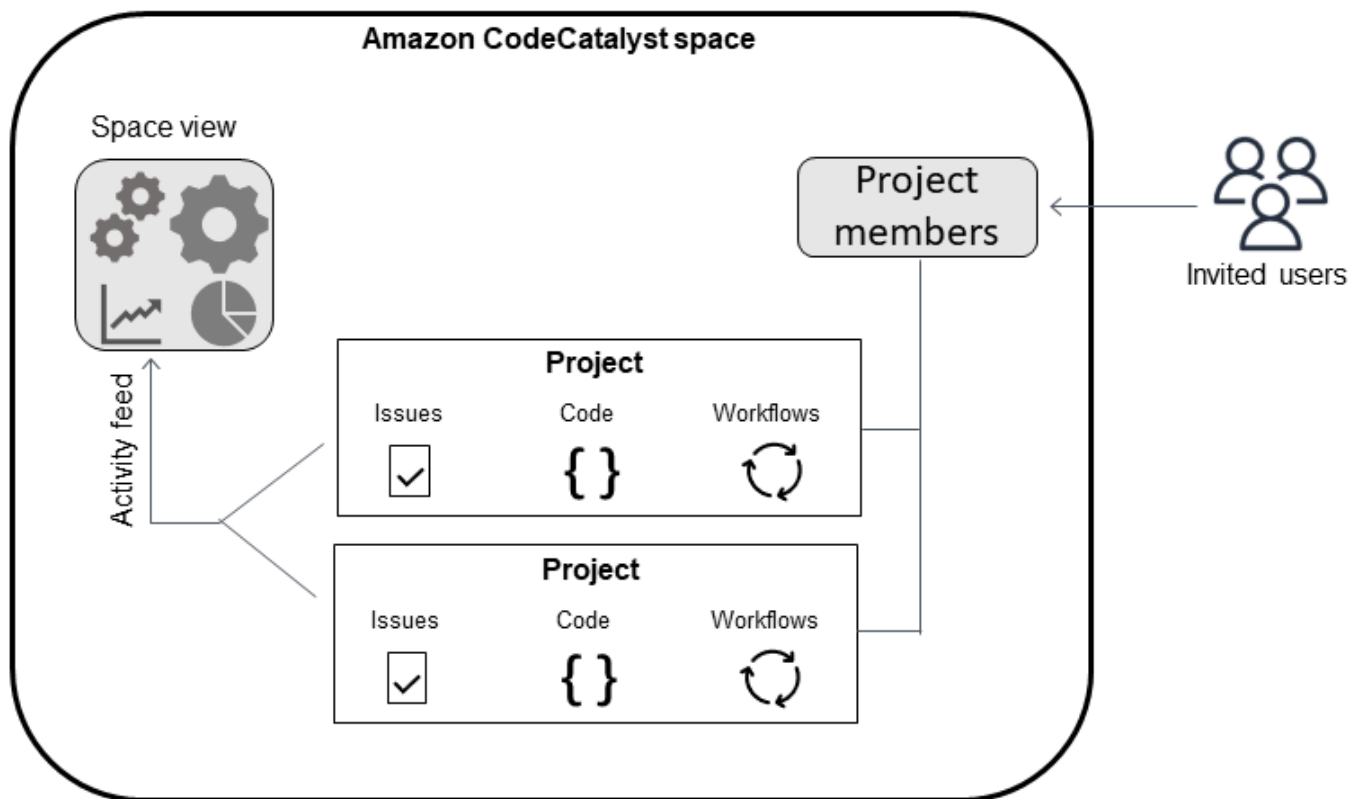
Avec le rôle d'administrateur de l'espace, vous pouvez gérer l'espace comme suit :

- Ajouter d'autres administrateurs de l'espace à l'espace
- Modifier les rôles et les autorisations des membres
- Modifier ou supprimer l'espace
- Créez des projets et invitez des membres à participer au projet
- Afficher la liste de tous les projets de l'espace
- Afficher le flux d'activités de tous les projets de l'espace

Lorsque vous créez un espace, vous y êtes automatiquement ajouté avec deux rôles : le rôle d'administrateur de l'espace et le rôle d'administrateur de projet pour le projet que vous avez créé dans le cadre de la création de l'espace. Les utilisateurs supplémentaires sont automatiquement ajoutés en tant que membres à l'espace lorsqu'ils acceptent des invitations à des projets. Cette appartenance à l'espace n'octroie aucune autorisation dans l'espace. Ce que les utilisateurs peuvent faire dans un espace est déterminé par le rôle de l'utilisateur dans un projet spécifique.

Pour plus d'informations sur les rôles , consultez [Utilisation des rôles sur Amazon CodeCatalyst](#).





Voici d'autres considérations relatives à l'ajout de comptes :

- Il existe un one-to-one mappage de la connexion au compte Compte AWS pour un espace. Un seul Compte AWS peut être ajouté à plusieurs espaces différents. Les comptes AWS sur lesquels vous déployez n'ont pas besoin d'être uniques et peuvent être utilisés par plusieurs espaces.
- Comptes AWS ajouté à un CodeCatalyst espace peut être utilisé dans n'importe quel projet dans cet espace.
- Bien que chaque environnement puisse en prendre en charge plusieurs Comptes AWS, vous ne pouvez utiliser qu'un seul compte par environnement dans une action.
- La facturation est configurée au niveau de l'espace. Plusieurs comptes peuvent être configurés pour la facturation, mais un seul peut être actif dans un CodeCatalyst espace. Un seul Compte AWS peut être utilisé comme compte de facturation pour un espace dans CodeCatalyst. Si un compte est déjà utilisé pour un espace, vous devez utiliser un autre compte de facturation pour l'espace supplémentaire.
- Après avoir créé une connexion, vous devez ajouter des rôles AWS IAM à votre connexion si votre flux de travail doit accéder à ces rôles IAM dans votre CodeCatalyst environnement. Pour

plus d'informations sur la façon dont les environnements sont utilisés, consultez [Utilisation des environnements](#).

## Rubriques

- [Création d'un espace prenant en charge les utilisateurs de AWS Builder ID](#)
- [Modification d'un espace](#)
- [Supprimer un espace dans CodeCatalyst](#)
- [Surveillance de l'activité d'un espace](#)
- [Administration Comptes AWS d'un espace](#)
- [Gestion des rôles IAM pour les comptes connectés](#)
- [Gestion des utilisateurs de l'espace](#)
- [Gérer des équipes](#)
- [Gestion des ressources des machines](#)
- [Administration des environnements de développement pour un espace](#)
- [Quotas de places dans CodeCatalyst](#)

## Création d'un espace prenant en charge les utilisateurs de AWS Builder ID

Lorsque vous vous inscrivez pour la première fois sur Amazon CodeCatalyst avec votre identifiant AWS Builder, vous devez créer un espace. Pour plus d'informations, consultez [Configuration CodeCatalyst](#). Vous pouvez choisir de créer des espaces supplémentaires pour répondre aux besoins de votre entreprise.

### Note

Les noms des espaces doivent être uniques d'un côté à l'autre CodeCatalyst. Vous ne pouvez pas réutiliser les noms des espaces supprimés.

Les informations contenues dans ce guide sont fournies pour créer des espaces destinés CodeCatalyst aux utilisateurs de AWS Builder ID. Les étapes de configuration et d'administration

d'un espace prenant en charge la fédération d'identités sont décrites dans le guide de l'CodeCatalyst administrateur. Pour utiliser des espaces configurés pour la fédération d'identité, consultez la section [Configuration et administration des CodeCatalyst espaces](#) dans le manuel Amazon CodeCatalyst Administrator Guide.

Pour créer des espaces supplémentaires qui prennent en charge les utilisateurs du AWS Builder ID, le rôle d'administrateur de l'espace doit vous être attribué.

#### Note

Lorsque vous créez un espace supplémentaire, vous n'êtes pas invité à créer un projet. Pour savoir comment créer des projets dans un espace, voir [Création d'un projet sur Amazon CodeCatalyst](#).

Pour créer un autre espace

1. Dans le AWS Management Console, assurez-vous d'être connecté avec le même nom Compte AWS que celui que vous souhaitez associer à votre CodeCatalyst espace.
2. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
3. Accédez à votre espace.

#### Tip

Si vous appartenez à plusieurs espaces, choisissez-en un dans la barre de navigation supérieure.

4. Choisissez Créer un espace.
5. Sur la page Créer un espace, dans Nom de l'espace, entrez le nom de l'espace. Vous ne pourrez pas le modifier ultérieurement.

#### Note

Les noms des espaces doivent être uniques d'un côté à l'autre CodeCatalyst. Vous ne pouvez pas réutiliser les noms des espaces supprimés.

6. Dans Région AWS, choisissez la région dans laquelle vous souhaitez stocker votre espace et les données du projet. Vous ne pourrez pas le modifier ultérieurement.

7. Dans Compte AWS ID, saisissez l'identifiant à douze chiffres du compte que vous souhaitez connecter à votre espace.

Dans le jeton de vérification du AWS compte, copiez l'identifiant du jeton généré. Le jeton est automatiquement copié pour vous, mais vous souhaitez peut-être le stocker pendant que vous approuvez la demande de AWS connexion.

8. Choisissez Verify in AWS.
9. La page Verify Amazon CodeCatalyst Space s'ouvre dans le AWS Management Console. Voici la page Amazon CodeCatalyst Spaces. Il se peut que vous deviez vous connecter pour accéder à la page.

Dans le AWS Management Console, assurez-vous de choisir le même Région AWS endroit où vous souhaitez créer votre espace.

Pour accéder directement à la page, connectez-vous aux Amazon CodeCatalyst Spaces AWS Management Console à l'adresse <https://console.aws.amazon.com/codecatalyst/home/>.

Le jeton de vérification est automatiquement saisi dans le jeton de vérification. Une bannière de réussite affiche un message indiquant que le jeton est un jeton valide.

10. Choisissez Vérifier l'espace.

Un message de confirmation du compte s'affiche pour indiquer que le compte a été ajouté à l'espace.

11. Restez sur la page Verify Amazon CodeCatalyst Space. Cliquez sur le lien suivant : Pour ajouter des rôles IAM pour cet espace, consultez les détails de l'espace.

La page de détails de CodeCatalyst l'espace s'ouvre dans le AWS Management Console. Voici la page Amazon CodeCatalyst Spaces. Il se peut que vous deviez vous connecter pour accéder à la page.

12. Dans la section Rôles IAM disponibles pour CodeCatalyst, choisissez Ajouter un rôle IAM.

La CodeCatalyst page Ajouter des rôles IAM disponibles s'affiche.

13. Choisissez Créer un rôle d'administrateur de CodeCatalyst développement dans IAM. Cette option crée un rôle de service qui contient la politique d'autorisation et la politique de confiance pour le rôle de développement.

Le rôle de développeur est un rôle AWS IAM qui permet à vos CodeCatalyst flux de travail d'accéder à AWS des ressources telles qu'Amazon S3, Lambda et. AWS CloudFormation Le rôle

portera un nom `CodeCatalystWorkflowDevelopmentRole-spaceName` avec un identifiant unique ajouté. Pour plus d'informations sur le rôle et la politique de rôle, consultez [Comprendre le rôle CodeCatalystWorkflowDevelopmentRole-\*spaceName\* du service](#).

14. Choisissez Créer un rôle de développement.
15. Sur la page de connexion, sous Rôles IAM disponibles pour CodeCatalyst, consultez le rôle de développeur dans la liste des rôles IAM ajoutés à votre compte.
16. Choisissez Go to Amazon CodeCatalyst.
17. Sur la page de création de CodeCatalyst, choisissez Créer un espace.

## Modification d'un espace

Vous pouvez modifier la description d'un espace pour aider les utilisateurs à mieux comprendre à quoi il sert.

Vous devez avoir le rôle d'administrateur de l'espace pour modifier les détails de l'espace.

Les informations contenues dans ce guide sont fournies pour modifier les espaces destinés CodeCatalyst aux utilisateurs de AWS Builder ID. Pour en savoir plus sur les étapes de configuration et d'administration d'un espace prenant en charge la fédération d'identités, consultez la section [Configuration et administration des CodeCatalyst espaces](#) dans le manuel Amazon CodeCatalyst Administrator Guide.

Pour modifier la description d'un espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace.

### Tip

Si vous appartenez à plusieurs espaces, choisissez-en un dans la barre de navigation supérieure.

3. Dans l'onglet Paramètres de l'espace, choisissez Modifier. Apportez les modifications souhaitées à la description de l'espace, puis choisissez Enregistrer.

## Supprimer un espace dans CodeCatalyst

Vous pouvez supprimer un espace pour supprimer l'accès à toutes les ressources de l'espace. Vous devez avoir le rôle d'administrateur de l'espace pour supprimer un espace.

### Note

Vous ne pouvez pas annuler la suppression d'un espace.

Une fois que vous avez supprimé un espace, tous les membres de l'espace ne pourront pas accéder aux ressources de l'espace. La facturation des ressources spatiales sera également interrompue, et tous les flux de travail demandés par des référentiels sources tiers seront interrompus.

### Note

Les noms des espaces doivent être uniques d'un côté à l'autre CodeCatalyst. Vous ne pouvez pas réutiliser les noms des espaces supprimés.

Les informations contenues dans ce guide sont fournies pour supprimer les espaces destinés CodeCatalyst aux utilisateurs de AWS Builder ID. Pour en savoir plus sur les étapes de configuration et d'administration d'un espace prenant en charge la fédération d'identités, consultez la section [Configuration et administration des CodeCatalyst espaces](#) dans le manuel Amazon CodeCatalyst Administrator Guide.

Pour supprimer un espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace.

### Tip

Si vous appartenez à plusieurs espaces, choisissez-en un dans la barre de navigation supérieure.

3. Choisissez Paramètres, puis sélectionnez Supprimer.
4. Tapez **delete** pour confirmer la suppression.

## 5. Sélectionnez Delete (Supprimer).

### Note

Si vous appartenez à plusieurs espaces, vous êtes redirigé vers la page de présentation des espaces. Si vous appartenez à un espace, vous êtes redirigé vers la page de création de l'espace.

## Surveillance de l'activité d'un espace

Pour consulter les projets récemment créés et les mises à jour de statut, vous pouvez utiliser la CodeCatalyst console pour consulter un flux d'activités présentant les mises à jour des ressources spatiales.

Dans le flux d'activités, vous pouvez consulter des indicateurs tels que les échecs d'exécution des flux de travail et les projets créés.

Pour consulter l'activité dans votre espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.

### Tip

Si vous appartenez à plusieurs espaces, choisissez-en un dans la barre de navigation supérieure.

3. Choisissez Activity (Activité).
4. Consultez les informations dans Activité.
5. Pour filtrer par activité, choisissez le sélecteur en haut à droite.
6. Pour afficher toutes les activités de votre espace, choisissez N'importe quel type d'activité.

## Administration Comptes AWS d'un espace

Vous pouvez utiliser les ressources Comptes AWS de votre CodeCatalyst espace Amazon. Pour ce faire, vous devez établir une connexion entre le Comptes AWS et votre espace dans CodeCatalyst.

La création d'une telle connexion signifie que les projets et les flux de travail de votre CodeCatalyst espace peuvent interagir avec les ressources de votre espace Comptes AWS. Vous devez créer une connexion pour chacune des connexions que Compte AWS vous souhaitez utiliser avec votre CodeCatalyst espace.

Après avoir créé une connexion, vous pouvez choisir d'y associer des rôles AWS IAM.

## Rubriques

- [Ajouter un Compte AWS à un espace](#)
- [Ajouter des rôles IAM aux connexions aux comptes](#)
- [Ajout de la connexion au compte et des rôles IAM à votre environnement de déploiement](#)
- [Afficher les connexions aux comptes](#)
- [Supprimer un compte d'un espace \(en CodeCatalyst\)](#)

Vous pouvez configurer CodeCatalyst l'utilisation autorisée Comptes AWS en ajoutant les comptes à votre espace. En Comptes AWS agrandissant votre CodeCatalyst espace, vous pouvez donner aux flux de travail de votre projet l'accès aux Compte AWS ressources et à votre configuration de facturation.

L'ajout d'un Compte AWS crée une connexion qui autorise l'utilisation CodeCatalyst de ce compte. Vous pouvez utiliser Added Comptes AWS pour effectuer les opérations suivantes :

- Configurez la facturation d'un CodeCatalyst espace. Consultez [la section Gestion de la facturation](#) dans le guide de CodeCatalyst l'administrateur Amazon.
- Autoriser CodeCatalyst à assumer des rôles IAM pour accéder aux AWS ressources et les déployer Services AWS dans le compte. veuillez consulter [Administration Comptes AWS d'un espace](#).

Les connexions aux comptes sont créées en complétant l'autorisation avec le Compte AWS. Une fois la connexion créée, vous pouvez configurer la connexion pour les flux de travail et les projets à utiliser en ajoutant des rôles IAM.

## Ajouter un Compte AWS à un espace

Vous utilisez la CodeCatalyst console et le AWS Management Console pour connecter votre espace à un Compte AWS.



Avant d'ajouter un Compte AWS à un espace dans CodeCatalyst, remplissez les conditions préalables suivantes :

- Créez Compte AWS et obtenez les autorisations nécessaires pour créer des rôles AWS IAM dans le compte que vous souhaitez connecter.
- Créez le ou les rôles IAM que vous souhaitez associer à la connexion de votre compte, y compris les politiques IAM avec les autorisations pour les rôles.
- Acquérez le rôle d'administrateur de l' CodeCatalystespace dans lequel vous souhaitez créer la connexion.

## Rubriques

- [Étape 1 : Création d'une demande de connexion](#)
- [Étape 2 : Acceptation d'une demande de connexion au compte](#)
- [Étape 3 : vérifier une connexion approuvée](#)
- [Étape 4 : ajouter des rôles IAM à votre connexion](#)
- [Étapes suivantes : créer des rôles IAM supplémentaires pour la connexion à votre compte](#)

## Étape 1 : Création d'une demande de connexion

La création d'une demande de connexion dans la CodeCatalyst console génère un jeton de connexion que vous pouvez utiliser pour finaliser l'autorisation.

Vous devez avoir le rôle d'administrateur de l'espace ou d'utilisateur avancé dans l' CodeCatalyst espace dans lequel vous souhaitez créer la connexion. Vous devez également disposer d'autorisations administratives pour les Compte AWS éléments que vous souhaitez ajouter.

### Pour créer une connexion

1. Dans le AWS Management Console, assurez-vous que vous êtes connecté avec le même compte que celui avec lequel vous souhaitez créer une connexion.
2. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
3. Accédez à votre CodeCatalyst espace. Choisissez Settings (Paramètres), puis Comptes AWS.
4. Choisissez Ajouter un Compte AWS.
5. Sur la CodeCatalyst page Compte AWS Associer à Amazon, dans Compte AWS ID, saisissez l'identifiant à douze chiffres du compte que vous souhaitez connecter à votre espace. Pour plus

d'informations sur la recherche de votre Compte AWS identifiant, consultez la section [Votre Compte AWS identifiant et son alias](#).

6. Dans Nom CodeCatalyst d'affichage Amazon, entrez un nom de référence pour le compte.
7. (Facultatif) Dans Description de la connexion, entrez une description du compte qui vous aidera à choisir les projets auxquels le compte et le ou les rôles s'appliqueront.
8. Choisissez Associer Compte AWS.
9. La page revient à la page de Compte AWS détails où une bannière de réussite s'affiche.

## Étape 2 : Acceptation d'une demande de connexion au compte

Après avoir soumis une demande de connexion à votre dans la CodeCatalyst console Compte AWS, vous devez contacter votre AWS administrateur pour accepter la demande de connexion en la soumettant avec le jeton de connexion fourni.

Assurez-vous que vous disposez des autorisations d'administrateur pour votre compte et que vous êtes connecté AWS Management Console au même compte que celui Compte AWS pour lequel vous créez la connexion.


Pour approuver une demande de connexion (console)

1. Dans le AWS Management Console, assurez-vous que vous êtes connecté avec le même compte que celui avec lequel vous souhaitez créer une connexion.
2. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
3. Accédez à votre CodeCatalyst espace. Choisissez Settings (Paramètres), puis Comptes AWS.
4. Sur la page de Compte AWS détails, sélectionnez Terminer la configuration dans le AWS Management Console.
5. La page Verify Amazon CodeCatalyst Space s'ouvre dans le AWS Management Console. Voici la page Amazon CodeCatalyst Spaces. Il se peut que vous deviez vous connecter pour accéder à la page.

Pour accéder directement à la page, connectez-vous aux Amazon CodeCatalyst Spaces AWS Management Console à l'adresse <https://console.aws.amazon.com/codecatalyst/home/>.

Le jeton de vérification est automatiquement saisi dans le jeton de vérification. Un message de réussite indique que le jeton est un jeton valide.

6. (Facultatif) Sous Niveaux payants autorisés, choisissez Autoriser les niveaux payants (Standard, Enterprise) pour activer les niveaux payants pour votre compte de facturation.

 Note

Cela ne fait pas passer le niveau de facturation à un niveau payant. Cependant, cela configure le Compte AWS afin que vous puissiez modifier le niveau de facturation de votre espace à tout moment. CodeCatalyst Vous pouvez activer les niveaux payants à tout moment. Sans cette modification, l'espace ne peut utiliser que le niveau gratuit.

7. Choisissez Vérifier l'espace.

Un message de confirmation du compte s'affiche pour indiquer que le compte a été ajouté à l'espace.

### Étape 3 : vérifier une connexion approuvée

Une fois qu'une connexion a été approuvée, vous pouvez consulter la connexion dans la console, ainsi que les rôles IAM que vous y avez ajoutés.

Pour vérifier une connexion approuvée

1. Accédez à votre CodeCatalyst espace. Choisissez Settings (Paramètres), puis Comptes AWS.
2. La connexion au compte est répertoriée avec la date à laquelle elle a été créée.
3. Choisissez le nom d'affichage du compte. La page Compte AWS de détails s'affiche.

### Étape 4 : ajouter des rôles IAM à votre connexion

Si vous utilisez un rôle IAM configuré pour une action de CodeCatalyst déploiement, ajoutez-le à votre environnement de déploiement. Pour plus d'informations, consultez [Ajouter des rôles IAM aux connexions aux comptes](#).

Étapes suivantes : créer des rôles IAM supplémentaires pour la connexion à votre compte

Après avoir créé une connexion, vous pouvez créer des rôles IAM supplémentaires à ajouter à celle-ci. Les rôles IAM que vous ajoutez dépendent de vos flux de travail. Par exemple, une action de CodeCatalyst construction nécessite le rôle de CodeCatalyst construction.

Pour connecter votre compte, vous aurez besoin du nom de ressource Amazon (ARN) correspondant aux rôles que vous avez créés. Copiez l'ARN de votre ou de vos rôles, comme indiqué ici. Pour plus d'informations sur l'utilisation des ARN pour les rôles IAM, consultez [Amazon Resource Name \(ARN\)](#).

Pour accéder à l'ARN de votre rôle IAM

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le panneau de navigation, sélectionnez Rôles.
3. Dans le champ de recherche, entrez le nom du rôle que vous souhaitez ajouter.
4. Choisissez le rôle dans la liste.

La page Résumé du rôle apparaît.

5. En haut, copiez la valeur ARN du rôle.

## Ajouter des rôles IAM aux connexions aux comptes

Lors de la création de la connexion à votre compte, vous devez notamment ajouter le ou les rôles IAM que vous souhaitez utiliser dans les projets de votre CodeCatalyst espace.

### Note

Pour utiliser des rôles IAM avec une connexion à un compte, assurez-vous que la politique de confiance est mise à jour pour utiliser le principal de CodeCatalyst service.

Ajouter des rôles IAM à une connexion de compte (console)

1. Dans le AWS Management Console, assurez-vous que vous êtes connecté avec le même compte que celui que vous souhaitez gérer.
2. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
3. Accédez à votre CodeCatalyst espace. Choisissez Settings (Paramètres), puis Comptes AWS.
4. Choisissez le nom CodeCatalyst d'affichage Amazon de la connexion à votre compte, puis sélectionnez Gérer les rôles depuis AWS Management Console.

La page Ajouter un rôle IAM à Amazon CodeCatalyst Space s'affiche.

5. Effectuez l'une des actions suivantes :

- Pour créer un rôle de service contenant la politique d'autorisation et la politique de confiance pour le rôle de développeur, choisissez Créer un rôle d'administrateur de CodeCatalyst développement dans IAM. Le rôle portera un nom CodeCatalystWorkflowDevelopmentRole-*spaceName* avec un identifiant unique ajouté. Pour plus d'informations sur le rôle et la politique de rôle, consultez [Comprendre le rôle CodeCatalystWorkflowDevelopmentRole-\*spaceName\* du service](#).

Choisissez Créer un rôle de développement.

- Pour ajouter un rôle que vous avez déjà créé dans IAM, choisissez Ajouter un rôle IAM existant. Dans Sélectionner un rôle IAM existant, choisissez le rôle dans la liste déroulante.

Choisissez Ajouter un rôle.

La page s'ouvre dans le AWS Management Console. Vous devrez peut-être vous connecter pour accéder à la page.


6. Dans le volet de navigation de la page Amazon CodeCatalyst Spaces, sélectionnez Spaces.

Pour accéder directement à la page, connectez-vous aux Amazon CodeCatalyst Spaces AWS Management Console à l'adresse <https://console.aws.amazon.com/codecatalyst/home/>.

7. Choisissez le compte ajouté à votre CodeCatalyst espace. La page de connexion s'affiche.
8. Sur la page de connexion, sous Rôles IAM disponibles pour CodeCatalyst, consultez la liste des rôles IAM ajoutés à votre compte. Choisissez Associer le rôle IAM à CodeCatalyst
9. Dans la fenêtre contextuelle Associer un rôle IAM, dans Rôle ARN, entrez le nom de ressource Amazon (ARN) du rôle IAM que vous souhaitez associer à votre CodeCatalyst espace.

Sous Objectif, choisissez un objectif de rôle qui décrit la manière dont vous souhaitez utiliser le rôle dans le cadre de la connexion à votre compte. Spécifiez RUNNER les rôles que vous utilisez pour exécuter des actions dans les flux de travail. Spécifiez SERVICE les rôles que vous utilisez pour accéder à un autre service.

Vous pouvez définir plusieurs objectifs.

 Note

Il est nécessaire de choisir un objectif pour le rôle ARN.

10. Choisissez Associer un rôle IAM. Répétez ces étapes pour les rôles IAM supplémentaires.

## Ajout de la connexion au compte et des rôles IAM à votre environnement de déploiement

Pour accéder à AWS des ressources, telles qu'Amazon ECS ou à AWS Lambda des ressources pour les déploiements, les actions de CodeCatalyst création et de déploiement nécessitent des rôles IAM autorisés à accéder à ces ressources. Avec le rôle d'administrateur de l'espace ou d'utilisateur avancé, vous pouvez connecter votre CodeCatalyst compte à l' Compte AWS endroit où vos ressources sont créées. Vous ajoutez ensuite le rôle IAM à la connexion de votre compte. Pour les actions de déploiement, vous devez ensuite ajouter le rôle IAM à un CodeCatalyst environnement.

Vous devez ajouter les rôles IAM que vous souhaitez utiliser dans les environnements de déploiement de vos projets. L'ajout des rôles à la connexion au compte n'ajoute pas les rôles et la connexion aux environnements de déploiement du projet. Pour ajouter votre connexion au compte et les rôles IAM à votre environnement de déploiement, assurez-vous que la connexion au compte et les rôles sont créés comme indiqué dans [Étape 4 : ajouter des rôles IAM à votre connexion](#).

Utilisez ensuite la page Environnements de la CodeCatalyst console pour ajouter votre connexion au compte et votre rôle IAM à un environnement de déploiement dans un projet.

### Note

Vous n'ajoutez un rôle IAM à un environnement que si le rôle IAM est utilisé pour une CodeCatalyst action nécessitant un rôle IAM. Toutes les actions de flux de travail qui nécessitent des rôles IAM, y compris les actions de génération, doivent utiliser un environnement. CodeCatalyst

Pour ajouter votre connexion au compte et vos rôles IAM à votre environnement de déploiement

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez au projet contenant l'environnement de déploiement dans lequel vous souhaitez ajouter la connexion au compte et les rôles IAM.
3. Développez CI/CD, puis choisissez Environments.
4. Choisissez votre environnement, puis les onglets supplémentaires s'affichent.
5. Choisissez l'onglet Compte AWS Connexions. Sous Nom de la connexion, les comptes qui ont été ajoutés à l'environnement, le cas échéant, sont répertoriés.

6. Choisissez Associer Compte AWS. La <environment\_name>page Compte AWS Associer à s'affiche.
7. Sous Connexion, choisissez le nom de la connexion au compte avec les rôles IAM que vous souhaitez ajouter. Choisissez Associer.

## Afficher les connexions aux comptes

Vous pouvez consulter la liste de vos connexions et les détails de chaque connexion.

Vous devez avoir le rôle d'administrateur de l'espace ou d'utilisateur avancé pour gérer les connexions à votre espace.

Pour afficher toutes les connexions d'un CodeCatalyst espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à l'espace avec la connexion au compte que vous souhaitez consulter.
3. Choisissez l'onglet AWS Comptes.
4. Sous AWS comptes, consultez la liste des connexions aux comptes pour l'espace, y compris l'ID du compte et le statut de chaque connexion.

Pour consulter les détails de connexion au compte

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace. Choisissez Settings (Paramètres), puis Comptes AWS.
3. Dans Nom CodeCatalyst d'affichage Amazon, choisissez le nom de la connexion. Sur la page Détails, consultez la liste des rôles IAM associés à la connexion ainsi que d'autres informations.

## Supprimer un compte d'un espace (en CodeCatalyst)

Vous pouvez supprimer une connexion au compte dont vous n'avez plus besoin. Pour cette procédure, vous allez utiliser CodeCatalyst pour supprimer une connexion au compte que vous avez précédemment ajoutée à votre espace. Cela supprime la connexion au compte de votre espace, à condition que le compte ne soit pas le compte de facturation de l'espace.

**⚠ Important**

Une fois la connexion à un compte supprimée, vous ne pouvez pas la reconnecter. Vous devez créer une nouvelle connexion au compte, puis associer des rôles et des environnements IAM, ou configurer la facturation, selon les besoins.

Un compte de facturation doit être désigné pour votre CodeCatalyst espace, même si l'utilisation de l'espace ne dépassera pas le niveau gratuit. Avant de pouvoir supprimer un espace pour un compte qui est un compte de facturation désigné, vous devez ajouter un autre compte pour votre espace. Consultez [la section Gestion de la facturation](#) dans le guide de CodeCatalyst l'administrateur Amazon.

**⚠ Important**

Bien que vous puissiez utiliser ces étapes pour supprimer un compte, cela n'est pas recommandé. Le compte peut également être configuré pour prendre en charge les flux de travail dans CodeCatalyst.

Pour gérer les connexions aux comptes de votre espace, vous devez avoir le rôle d'administrateur de l'espace ou d'utilisateur avancé.

Pour supprimer une connexion à un compte

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace. Choisissez Settings (Paramètres), puis Comptes AWS.
3. Sous le nom CodeCatalyst d'affichage Amazon, choisissez le sélecteur à côté de la connexion au compte que vous souhaitez supprimer.
4. Sélectionnez Remove (Supprimer) Compte AWS. Confirmez la suppression en saisissant le nom dans le champ, puis en choisissant Supprimer.

Une bannière de réussite s'affiche et la connexion au compte est supprimée de la liste des connexions.



## Gestion des rôles IAM pour les comptes connectés

Vous créez des rôles dans AWS Identity and Access Management (IAM) pour le compte que vous souhaitez ajouter CodeCatalyst. Si vous ajoutez un compte de facturation, il n'est pas nécessaire de créer des rôles.

Dans votre Compte AWS, vous devez être autorisé à créer des rôles pour les Compte AWS personnes que vous souhaitez ajouter à votre espace. Pour plus d'informations sur les rôles et les politiques IAM, y compris les références IAM et les exemples de politiques, consultez [Identity and Access Management et Amazon CodeCatalyst](#) Pour plus d'informations sur la politique de confiance et les principes de service utilisés dans CodeCatalyst, consultez [CodeCatalyst modèle de confiance](#).

Vous devez être connecté avec le rôle d'administrateur de l'espace pour effectuer les étapes d'ajout de comptes (et de rôles, le cas échéant) à votre espace. CodeCatalyst

Vous pouvez ajouter des rôles aux connexions de votre compte en utilisant l'une des méthodes suivantes.

- Pour créer un rôle de service contenant la politique d'autorisation et la politique de confiance associées au CodeCatalystWorkflowDevelopmentRole-*spaceName*rôle, consultez [Rôle CodeCatalystWorkflowDevelopmentRole-\*spaceName\*](#).
- Pour un exemple de création d'un rôle et d'ajout d'une politique pour créer un projet à partir d'un plan, consultez [Création d'un rôle IAM et utilisation de la politique de CodeCatalyst confiance](#).
- Pour obtenir une liste d'exemples de politiques de rôle à utiliser lors de la création de vos rôles IAM, consultez [Rôles IAM pour l' CodeCatalyst accès d'Amazon aux ressources AWS](#).
- Pour connaître les étapes détaillées de création de rôles pour les actions de flux de travail, consultez le didacticiel de flux de travail correspondant à cette action, comme suit :
  - [Tutoriel : télécharger des artefacts sur Amazon S3](#)
  - [Tutoriel : Déployer une application sans serveur à l'aide de AWS CloudFormation](#)
  - [Tutoriel : Déploiement d'une application sur Amazon ECS](#)
  - [Tutoriel : code Lint à l'aide d'une action GitHub](#)

### Rubriques

- [Rôle CodeCatalystWorkflowDevelopmentRole-spaceName](#)
- [Rôle AWSRoleForCodeCatalystSupport](#)
- [Création d'un rôle IAM et utilisation de la politique de CodeCatalyst confiance](#)

## Rôle CodeCatalystWorkflowDevelopmentRole-*spaceName*

Vous créez le rôle de développeur en un clic dans IAM. Vous devez avoir le rôle d'administrateur de l'espace ou d'utilisateur avancé dans l'espace où vous souhaitez ajouter le compte. Vous devez également disposer d'autorisations administratives pour les Compte AWS éléments que vous souhaitez ajouter.

Avant de commencer la procédure ci-dessous, vous devez vous connecter AWS Management Console avec le même compte que celui que vous souhaitez ajouter à votre CodeCatalyst espace. Dans le cas contraire, la console renverra une erreur de compte inconnue.

Pour créer et ajouter CodeCatalyst CodeCatalystWorkflowDevelopmentRole-*spaceName*

1. Avant de commencer dans la CodeCatalyst console, ouvrez le AWS Management Console, puis assurez-vous que vous êtes connecté avec le même identifiant Compte AWS pour votre espace.
2. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
3. Accédez à votre CodeCatalyst espace. Choisissez Settings (Paramètres), puis Comptes AWS.
4. Choisissez le lien correspondant à l' Compte AWS endroit où vous souhaitez créer le rôle. La page Compte AWS de détails s'affiche.
5. Choisissez Gérer les rôles à partir de AWS Management Console.

La page Ajouter un rôle IAM à Amazon CodeCatalyst Space s'ouvre dans le AWS Management Console. Voici la page Amazon CodeCatalyst Spaces. Il se peut que vous deviez vous connecter pour accéder à la page.

6. Choisissez Créer un rôle d'administrateur de CodeCatalyst développement dans IAM. Cette option crée un rôle de service qui contient la politique d'autorisation et la politique de confiance pour le rôle de développement. Le rôle aura un nomCodeCatalystWorkflowDevelopmentRole-*spaceName*. Pour plus d'informations sur le rôle et la politique de rôle, consultez[Comprendre le rôle CodeCatalystWorkflowDevelopmentRole-\*spaceName\*du service](#).

### Note

Ce rôle est uniquement recommandé pour les comptes de développeur et utilise la politique AdministratorAccess AWS gérée, ce qui lui donne un accès complet pour créer de nouvelles politiques et ressources dans ce cadre Compte AWS.

7. Choisissez Créer un rôle de développement.
8. Sur la page des connexions, sous Rôles IAM disponibles pour CodeCatalyst, consultez le CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle dans la liste des rôles IAM ajoutés à votre compte.
9. Pour retourner dans votre espace, choisissez Go to Amazon CodeCatalyst.

## Rôle AWSRoleForCodeCatalystSupport

Vous créez le rôle de support sous la forme d'un rôle en un clic dans IAM. Vous devez avoir le rôle d'administrateur de l'espace ou d'utilisateur avancé dans l'espace où vous souhaitez ajouter le compte. Vous devez également disposer d'autorisations administratives pour les Compte AWS éléments que vous souhaitez ajouter.

Avant de commencer la procédure ci-dessous, vous devez vous connecter AWS Management Console avec le même compte que celui que vous souhaitez ajouter à votre CodeCatalyst espace. Dans le cas contraire, la console renverra une erreur de compte inconnue.

Pour créer et ajouter CodeCatalyst AWSRoleForCodeCatalystSupport

1. Avant de commencer dans la CodeCatalyst console, ouvrez le AWS Management Console, puis assurez-vous que vous êtes connecté avec le même identifiant Compte AWS pour votre espace.
2. Accédez à votre CodeCatalyst espace. Choisissez Settings (Paramètres), puis Comptes AWS.
3. Choisissez le lien correspondant à l' Compte AWS endroit où vous souhaitez créer le rôle. La page Compte AWS de détails s'affiche.
4. Choisissez Gérer les rôles à partir de AWS Management Console.

La page Ajouter un rôle IAM à Amazon CodeCatalyst Space s'ouvre dans le AWS Management Console. Voici la page Amazon CodeCatalyst Spaces. Vous devrez peut-être vous connecter pour accéder à la page.

5. Sous Détails de CodeCatalyst l'espace, choisissez Ajouter un rôle de CodeCatalyst support. Cette option crée un rôle de service qui contient la politique d'autorisation et la politique de confiance pour le rôle de développement préliminaire. Le rôle portera un nom AWSRoleForCodeCatalystSupportavec un identifiant unique ajouté. Pour plus d'informations sur le rôle et la politique de rôle, consultez [Comprendre le rôle AWSRoleForCodeCatalystSupportdu service](#).

6. Sur la page Ajouter un rôle pour le CodeCatalyst support, laissez la valeur par défaut sélectionnée, puis choisissez Créer un rôle.
7. Sous Rôles IAM disponibles pour CodeCatalyst, consultez le CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle dans la liste des rôles IAM ajoutés à votre compte.
8. Pour retourner dans votre espace, choisissez Go to Amazon CodeCatalyst.

## Création d'un rôle IAM et utilisation de la politique de CodeCatalyst confiance

Les rôles IAM à utiliser CodeCatalyst avec Compte AWS les connexions doivent être configurés pour utiliser la politique de confiance fournie ici. Suivez ces étapes pour créer un rôle IAM et associer une politique qui vous permet de créer des projets à partir de plans intégrés. CodeCatalyst

Vous pouvez également créer un rôle de service contenant la politique d'autorisation et la politique de confiance associées au CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle. Pour plus d'informations, consultez [Ajouter des rôles IAM aux connexions aux comptes](#).

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse <https://console.aws.amazon.com/iam/>.
2. Cliquez sur Rôles, puis sur Créer un rôle.
3. Choisissez Politique de confiance personnalisée.
4. Dans le formulaire de politique de confiance personnalisée, collez la politique de confiance suivante.

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

```
        "ArnLike": {
            "aws:SourceArn": "arn:aws:codecatalyst:::space/spaceId/project/
* "
        }
    }
}
```

5. Choisissez Suivant.
6. Sous Ajouter des autorisations, recherchez et sélectionnez une politique personnalisée que vous avez déjà créée dans IAM.
7. Choisissez Suivant.
8. Dans Nom du rôle, entrez un nom pour le rôle, par exemple : `codecatalyst-project-role`
9. Sélectionnez Créer un rôle.
10. Copiez le rôle Amazon Resource Name (ARN). Vous devrez fournir ces informations lors de l'ajout du rôle à la connexion ou à l'environnement de votre compte.

## Gestion des utilisateurs de l'espace

Vous pouvez gérer les membres d'un espace en consultant, en ajoutant, en supprimant ou en modifiant les rôles des utilisateurs qui rejoignent l'espace.

Les informations contenues dans ce guide sont fournies pour inviter et gérer les utilisateurs dans les espaces CodeCatalyst qui prennent en charge les utilisateurs de AWS Builder ID. Pour en savoir plus sur les étapes de configuration et d'administration d'un espace prenant en charge la fédération d'identités, consultez la section [Configuration et administration des CodeCatalyst espaces](#) dans le manuel Amazon CodeCatalyst Administrator Guide.

### Afficher les membres dans un espace

Vous pouvez afficher les utilisateurs de votre espace, y compris les informations concernant leurs noms d'affichage, leurs alias et le rôle qu'ils jouent dans l'espace. Les membres d'un espace ont trois rôles :

- **Administrateur de l'espace** : ce rôle dispose de toutes les autorisations nécessaires CodeCatalyst, y compris de création de projets. Attribuez ce rôle uniquement aux utilisateurs qui doivent administrer tous les aspects d'un espace, tels que l'accès à tous les projets de l'espace.

Vous ne pouvez pas modifier ce rôle ultérieurement sans supprimer d'abord l'utilisateur. Pour plus d'informations, consultez [Rôle d'administrateur de l'espace](#).

- **Utilisateur expérimenté** : ce rôle est le deuxième rôle le plus puissant dans Amazon CodeCatalyst Spaces, mais il n'a aucun accès aux projets d'un espace. Il est conçu pour les utilisateurs qui doivent être en mesure de créer des projets dans un espace et d'aider à gérer les utilisateurs et les ressources de l'espace. Pour plus d'informations, consultez [Rôle d'utilisateur avancé](#).
- **Accès limité** : ce rôle est attribué par défaut aux utilisateurs qui rejoignent l'espace en acceptant des invitations à participer à des projets dans l'espace. Les membres du projet se voient attribuer un rôle dans un projet. Pour plus d'informations sur la gestion des membres du projet, consultez [Gérer les membres du projet](#).

Le tableau des administrateurs de l'espace indique les utilisateurs ayant le rôle d'administrateur de l'espace. Ces utilisateurs ne figurent pas dans les membres de l'espace car ils sont automatiquement (implicitement) affectés à tous les projets de l'espace et n'ont aucun rôle dans un projet.

Le tableau des membres de l'espace indique tous les membres de l'espace qui jouent un rôle dans un projet sans avoir le rôle d'administrateur de l'espace.

Les utilisateurs sont affichés selon qu'ils ont ou non le rôle d'administrateur de l'espace CodeCatalyst, comme suit :

- Un utilisateur ayant le rôle d'administrateur de l'espace qui accepte ultérieurement une invitation et un rôle de projet n'apparaîtra pas dans le tableau des membres de l'espace sous les espaces ni dans le tableau des membres du projet sous les projets. Ils continueront d'être affichés dans le tableau des administrateurs de l'espace aux deux endroits. Dans chaque projet, tous les utilisateurs ayant le rôle d'administrateur de l'espace sont indiqués dans le tableau des administrateurs de l'espace du projet correspondant.
- Un utilisateur qui accepte une invitation à participer à un projet avec un rôle de projet est ajouté à l'espace avec le rôle à accès limité. Si le rôle de l'utilisateur passe ultérieurement au rôle d'administrateur de l'espace, mais qu'il passe également du tableau des membres de l'espace au tableau des administrateurs de l'espace. Dans le cadre du projet, l'utilisateur passera du tableau des membres du projet au tableau des administrateurs de l'espace.

Pour afficher les utilisateurs et les rôles de votre espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).

## 2. Accédez à votre espace.

### Tip

Si vous appartenez à plusieurs espaces, choisissez-en un dans la barre de navigation supérieure.

## 3. Choisissez Paramètres, puis Membres.

Les utilisateurs membres de l'espace sont indiqués dans le tableau des membres de l'espace.

### Tip

Si vous avez le rôle d'administrateur de l'espace, vous pouvez voir les projets auxquels vous avez été directement invité. Accédez aux paramètres du projet, puis sélectionnez Mes projets.

Dans la colonne État, les valeurs suivantes sont valides :

- Invité : CodeCatalyst a envoyé l'invitation mais l'utilisateur n'a pas encore accepté ou refusé.
- Membre — L'utilisateur a accepté l'invitation.

## Inviter un utilisateur directement dans un espace

Vous pouvez inviter des utilisateurs directement dans votre CodeCatalyst espace. Cela est utile lorsque vous souhaitez inviter cet utilisateur à vous aider à gérer l'espace en lui attribuant le rôle d'administrateur de l'espace ou d'utilisateur avancé. L'attribution de l'un de ces rôles à d'autres utilisateurs peut vous aider à répartir les responsabilités liées à la gestion de l'espace entre un plus grand nombre de personnes sans avoir à inviter ces utilisateurs à participer à des projets.

### Note

Vous devez avoir le rôle d'administrateur de l'espace ou d'utilisateur avancé pour inviter des membres.

Le tableau des administrateurs de l'espace indique les utilisateurs ayant le rôle d'administrateur de l'espace. Ces utilisateurs ne figurent pas dans le tableau des membres de l'espace car ils sont automatiquement (implicitement) affectés à tous les projets de l'espace et n'ont aucun rôle dans un projet.

Les membres qui acceptent une invitation à un projet sont ajoutés à l'espace par défaut. Le tableau des membres du projet indique tous les membres de l'espace qui jouent un rôle dans un projet.

Pour plus d'informations sur la façon d'accepter une invitation et de se connecter pour la première fois, consultez [Configuration CodeCatalyst](#).

Pour inviter un utilisateur dans votre espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace.
3. Choisissez Paramètres, puis Membres.
4. Choisissez Inviter.
5. Entrez l'e-mail de la personne que vous souhaitez inviter à rejoindre votre espace. Dans Rôle, choisissez le rôle que vous souhaitez attribuer à cet utilisateur dans l'espace.
6. Choisissez Inviter

## Annulation d'une invitation pour un espace

Si vous souhaitez annuler une invitation à rejoindre un espace que vous avez envoyée récemment et qu'elle n'a pas encore été acceptée, vous pouvez l'annuler.

Pour gérer les invitations à des espaces, vous devez avoir le rôle d'administrateur de l'espace ou d'utilisateur avancé.

Pour annuler l'invitation d'un membre de l'espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace.



**i** Tip

Si vous appartenez à plusieurs espaces, choisissez-en un dans la barre de navigation supérieure.

3. Choisissez Paramètres, puis Membres.
4. Vérifiez que le membre a le statut Invité.

**i** Note

Vous ne pouvez annuler qu'une invitation qui n'a pas encore été acceptée.

5. Choisissez l'option à côté de la ligne avec le membre invité, puis cliquez sur Annuler l'invitation.
6. Une fenêtre de confirmation s'affiche. Choisissez Annuler l'invitation pour confirmer.

## Modifier le rôle d'un membre de l'espace

Vous pouvez modifier le rôle attribué à un membre de votre espace. Vous devez avoir le rôle d'administrateur de l'espace pour modifier le rôle d'un utilisateur dans l'espace.

Le tableau des administrateurs de l'espace indique les utilisateurs ayant le rôle d'administrateur de l'espace. Ces utilisateurs ne figurent pas dans le tableau des membres de l'espace car ils sont automatiquement (implicitement) affectés à tous les projets de l'espace.

Pour modifier le rôle d'un utilisateur dans votre espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace.

**i** Tip

Si vous appartenez à plusieurs espaces, choisissez-en un dans la barre de navigation supérieure.

3. Choisissez Paramètres, puis Membres.
4. Dans le tableau des membres de l'espace, choisissez l'utilisateur dont vous souhaitez modifier le rôle. Choisissez Changer de rôle.

## Supprimer un membre de l'espace

Vous pouvez supprimer un membre de votre espace lorsqu'il n'a pas besoin d'accéder aux ressources de l'espace. Vous devez avoir le rôle d'administrateur de l'espace pour supprimer un membre d'un espace.

Le tableau des administrateurs de l'espace indique les utilisateurs ayant le rôle d'administrateur de l'espace. Ces utilisateurs ne figurent pas dans le tableau des membres de l'espace car ils sont automatiquement (implicitement) affectés à tous les projets de l'espace et n'ont aucun rôle dans un projet. Vous ne pouvez supprimer directement qu'un membre de votre espace dans ce tableau.

Pour supprimer un utilisateur du tableau des membres du projet

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace.

### Tip

Si vous appartenez à plusieurs espaces, choisissez-en un dans la barre de navigation supérieure.

3. Choisissez Paramètres, puis Membres.
4. Choisissez l'utilisateur dans le tableau des membres du projet. Sélectionnez Remove (Supprimer).


### Note

La suppression d'un membre de l'espace supprimera l'utilisateur de tous les projets de l'espace, ainsi que les autorisations associées aux ressources de ces projets.

## Supprimer ou modifier le rôle d'un utilisateur ayant le rôle d'administrateur de l'espace


Vous pouvez supprimer ou modifier le rôle d'un utilisateur avec le rôle d'administrateur de l'espace pour votre espace.

Vous devez avoir le rôle d'administrateur de l'espace pour supprimer d'un espace un utilisateur ayant le rôle d'administrateur de l'espace. La modification du rôle d'un utilisateur doté du rôle d'administrateur de l'espace supprime essentiellement l'utilisateur du tableau des administrateurs de l'espace. Si cet utilisateur n'a aucun rôle de projet dans les projets de l'espace, le fait de le supprimer du rôle d'administrateur de l'espace entraîne le retrait de l'utilisateur de l'espace.

 Note

En tant qu'utilisateur doté du rôle d'administrateur de l'espace, vous ne pouvez pas vous désinscrire. Contactez un autre utilisateur ayant le rôle d'administrateur de l'espace.

Pour supprimer un utilisateur ayant le rôle d'administrateur de l'espace dans le tableau des membres de l'espace

 Note

Pour un utilisateur qui n'a pas été ajouté explicitement à un projet, il n'a aucun rôle dans le projet (administrateur de projet ou contributeur). Si le rôle d'administrateur de l'espace est le seul rôle de l'utilisateur, celui-ci est complètement supprimé de l'espace.

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à l'espace dans lequel vous souhaitez supprimer ou modifier le rôle d'un utilisateur ayant le rôle d'administrateur de l'espace.
3. Choisissez Paramètres, puis Membres.
4. Consultez le statut de l'invitation pour la liste des membres et assurez-vous qu'elle ne contient aucune invitation en attente non autorisée à accéder à l'espace (statut Invité).

 Important

Avant de supprimer un utilisateur ayant le rôle d'administrateur de l'espace, vous devez vérifier qu'aucune invitation en attente n'a été lancée.

5. Choisissez l'onglet Membres. Dans le tableau des administrateurs de l'espace, choisissez l'utilisateur, puis sélectionnez Supprimer.

Dans la boîte de dialogue Supprimer un membre, effectuez l'une des opérations suivantes.

- Choisissez l'option permettant de supprimer uniquement le rôle d'administrateur de l'espace de l'utilisateur. Sélectionnez Remove (Supprimer).

#### Important

Si aucun autre rôle n'est attribué à l'utilisateur, le changement de rôle depuis Administrateur de l'espace entraîne le retrait de l'utilisateur de l'espace.

- Choisissez l'option permettant de supprimer un utilisateur ayant le rôle d'administrateur de l'espace et de tous ses projets. Sélectionnez Remove (Supprimer).
6. Actualisez l'onglet Membres. L'utilisateur est automatiquement ajouté à la liste des membres du projet dans tous les projets auxquels il était membre par le biais de rôles dans le projet. Si le rôle d'administrateur de l'espace était le seul rôle de l'utilisateur, celui-ci est complètement supprimé de l'espace.

## Gérer des équipes

Après avoir créé un espace, vous pouvez ajouter des équipes. Les équipes vous permettent de regrouper les utilisateurs afin qu'ils puissent partager les autorisations et gérer les projets, le suivi des problèmes, les rôles et les ressources dans CodeCatalyst.

Vous devez avoir le rôle d'administrateur de l'espace pour gérer les équipes.

### Rubriques

- [Création d'une équipe](#)
- [Afficher une équipe](#)
- [Gérer le rôle spatial d'une équipe](#)
- [Gérer un rôle de projet pour une équipe](#)
- [Ajouter directement un utilisateur à une équipe](#)
- [Supprimer directement un utilisateur d'une équipe](#)
- [Ajouter un groupe SSO à une équipe](#)
- [Supprimer une équipe](#)

## Création d'une équipe

Une équipe peut disposer d'autorisations de rôle, telles que Power User, dans un espace. Une équipe peut également disposer d'autorisations de projet, telles que celle d'administrateur de projet, dans un projet. Les équipes peuvent être associées à de nombreux projets avec des rôles différents pour chaque projet. Vous pouvez gérer des équipes dont les membres sont soit des utilisateurs individuels pour un espace AWS Builder ID, soit des groupes SSO pour un espace prenant en charge la fédération d'identité.

Sur la page des membres réservée aux utilisateurs de l'espace et du projet, les utilisateurs peuvent avoir plusieurs rôles. Les utilisateurs ayant plusieurs rôles afficheront un indicateur lorsqu'ils ont plusieurs rôles, et le rôle ayant le plus d'autorisations sera affiché en premier.

### Note

Si votre espace prend en charge la fédération d'identité, vos utilisateurs ou groupes SSO doivent déjà être configurés dans IAM Identity Center.

La façon dont vous gérez les membres de l'équipe dépend de la manière dont vous ajouterez et supprimerez des utilisateurs. Il existe deux options pour gérer les membres de l'équipe :


- Ajouter des utilisateurs directement : vous ajoutez ou supprimez des utilisateurs individuellement. Par exemple, vous ajoutez des utilisateurs à une équipe en choisissant des utilisateurs AWS Builder ID ou des utilisateurs SSO déjà configurés dans IAM Identity Center. Lorsque vous choisissez de gérer les membres de l'équipe en ajoutant directement des utilisateurs AWS Builder ID ou des utilisateurs SSO, l'option permettant d'utiliser des groupes SSO ne sera plus disponible.
- Utiliser des groupes SSO : vous gérez les membres de l'équipe par le biais de groupes SSO déjà configurés dans IAM Identity Center. Lorsque vous choisissez de gérer les membres de l'équipe à l'aide de groupes SSO, l'option permettant d'ajouter directement des utilisateurs ne sera plus disponible.

Vous devez avoir le rôle d'administrateur de l'espace pour gérer les équipes.

Pour créer une équipe

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace. Choisissez Paramètres, puis Teams.

3. Choisissez Créer une équipe.
4. Dans Nom de l'équipe, entrez un nom descriptif pour votre équipe.


 Note

Le nom de l'équipe doit être unique dans votre espace.

(Facultatif) Dans Description de l'équipe, entrez une description pour votre équipe.

5. Sous Rôle spatial, choisissez un rôle dans la liste des rôles spatiaux disponibles CodeCatalyst que vous souhaitez attribuer à l'équipe. Le rôle sera hérité par tous les membres de l'équipe.
  - Administrateur de l'espace - Pour plus de détails, voir [Rôle d'administrateur de l'espace](#).
  - Accès limité - Pour plus de détails, voir [Rôle à accès limité](#).
  - Utilisateur expérimenté - Pour plus de détails, voir [Rôle d'utilisateur avancé](#).
6. Dans Appartenance à l'équipe, choisissez l'une des options suivantes pour choisir la méthode d'ajout de membres à l'équipe.
  - Choisissez Ajouter des membres directement pour gérer les utilisateurs individuellement. Cela inclut l'ajout d'utilisateurs AWS Builder ID pour un espace ou l'ajout d'utilisateurs SSO pour un espace prenant en charge la fédération d'identité.
  - Choisissez Utiliser les groupes SSO pour sélectionner les groupes SSO que vous avez déjà configurés dans IAM Identity Center.

Dans Groupes SSO, cochez la case à côté des groupes que vous souhaitez ajouter. Vous pouvez ajouter jusqu'à cinq groupes SSO.

 Note

Vous ne pourrez pas le modifier ultérieurement. Lorsque vous choisissez de gérer les membres de l'équipe en ajoutant directement des utilisateurs AWS Builder ID ou des utilisateurs SSO, l'option permettant d'utiliser des groupes SSO ne sera plus disponible. Lorsque vous choisissez de gérer les membres de l'équipe à l'aide de groupes SSO, l'option permettant d'ajouter directement des utilisateurs ne sera plus disponible.

7. Choisissez Créer.

**Note**

Lorsque vous choisissez d'utiliser des groupes SSO, notez que les utilisateurs du groupe SSO ne sont pas sélectionnés lors de la création de l'équipe. Les utilisateurs devront s'être connectés pour CodeCatalyst être visibles dans la liste.

## Afficher une équipe

Dans CodeCatalyst, vous pouvez consulter les projets et les rôles de votre équipe. Sur la page des membres, vous pouvez consulter les rôles du projet et la liste des utilisateurs. Pour les équipes de type groupe SSO, vous pourrez également voir une liste des groupes SSO associés à l'équipe.

Pour consulter une équipe

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace. Choisissez Paramètres, puis Teams.
3. Dans Rôle spatial, consultez le rôle attribué à l'équipe pour cet espace.
4. Dans l'onglet Rôles du projet, consultez le projet et le rôle de projet attribués à l'équipe pour chaque CodeCatalyst projet dans l'espace où l'équipe a été ajoutée en tant que membre (pour un espace AWS Builder ID uniquement).
5. Dans l'onglet Membres, consultez la liste des membres affectés à l'équipe.
6. Dans l'onglet Groupes SSO, consultez la liste des groupes SSO affectés à l'équipe (pour un espace prenant en charge la fédération d'identité uniquement).

## Gérer le rôle spatial d'une équipe

Une équipe peut disposer d'autorisations de rôle, telles que Power User, dans un espace. Vous pouvez modifier le rôle spatial d'une équipe, mais notez que tous les membres de l'équipe hériteront de ces autorisations.

Vous devez avoir le rôle d'administrateur de l'espace pour gérer les équipes.

Modifier le rôle spatial d'une équipe

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).

2. Accédez à votre espace. Choisissez Paramètres, puis Teams.
3. Dans Actions, sélectionnez Modifier le rôle spatial. Vous pouvez remplacer le rôle spatial par l'un des rôles suivants. Cela modifie le rôle de tous les membres de l'équipe.
  - Administrateur de l'espace - Pour plus de détails, voir [Rôle d'administrateur de l'espace](#).
  - Accès limité - Pour plus de détails, voir [Rôle à accès limité](#).
  - Utilisateur expérimenté - Pour plus de détails, voir [Rôle d'utilisateur avancé](#).
4. Choisissez Enregistrer.

## Gérer un rôle de projet pour une équipe

Une équipe dans CodeCatalyst est similaire à un utilisateur dans la mesure où les membres de l'équipe peuvent avoir des autorisations de rôle, telles que celle d'administrateur de projet, dans un projet. Un changement de rôle sera appliqué à l'équipe, et tous les membres de l'équipe hériteront de ces autorisations. Vous pouvez choisir un rôle pour chaque projet qui sera automatiquement attribué à l'équipe.

Vous devez avoir le rôle d'administrateur de l'espace pour gérer les équipes.

Pour ajouter ou modifier un rôle dans un projet

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace. Choisissez Paramètres, puis Teams.
3. Choisissez l'onglet Rôles du projet.
4. Pour modifier un rôle, choisissez le sélecteur situé à côté du projet dans cette liste, puis choisissez Changer de rôle. Pour ajouter un rôle, choisissez Ajouter un rôle dans le projet. Dans Projet, choisissez le projet que vous souhaitez ajouter et dans Rôle, choisissez le rôle. Choisissez l'un des rôles de projet disponibles :
  - Administrateur de projet - Pour plus de détails, voir [Rôle d'administrateur de projet](#).
  - Contributeur - Pour plus de détails, voir [Rôle du contributeur](#).
  - Réviseur - Pour plus de détails, voir [Rôle du réviseur](#).
  - Lecture seule - Pour plus de détails, voir [Rôle en lecture seule](#).
5. Choisissez Enregistrer.



## Pour supprimer un rôle dans le projet

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace. Choisissez Paramètres, puis Teams.
3. Choisissez l'onglet Rôles du projet.
4. Choisissez le rôle que vous souhaitez supprimer.

### Important

La suppression d'un rôle dans une équipe entraîne la suppression des autorisations associées pour tous les utilisateurs de l'équipe.

5. Choisissez Enregistrer.

## Ajouter directement un utilisateur à une équipe

Vous pouvez ajouter des membres à votre équipe. Lorsque vous ajoutez un utilisateur, le nouvel utilisateur hérite des autorisations de tous les rôles existants au sein de l'équipe.

Que votre espace soit configuré pour l'assistance aux utilisateurs de AWS Builder ID ou pour la fédération d'identité, vous pouvez configurer votre espace pour ajouter directement des utilisateurs.

### Note

Lorsque votre espace est configuré pour gérer les membres de l'équipe à l'aide de groupes SSO, l'option permettant d'utiliser directement l'option Ajouter des utilisateurs n'est pas disponible. Pour utiliser les groupes SSO, voir [Ajouter un groupe SSO à une équipe](#).

Vous devez avoir le rôle d'administrateur de l'espace pour gérer les équipes.

## Pour ajouter directement un utilisateur

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace. Choisissez Paramètres, puis Teams.
3. Choisissez l'onglet Membres.
4. Choisissez Ajouter un membre.

**Note**

Les utilisateurs ajoutés à une équipe doivent déjà être membres d'un espace. Vous ne pouvez pas ajouter ou inviter un membre de l'équipe qui n'est pas membre de l'espace.

5. Choisissez un utilisateur dans le champ déroulant, puis cliquez sur Enregistrer. Choisissez des utilisateurs AWS Builder ID ou des utilisateurs SSO déjà configurés dans IAM Identity Center.

## Supprimer directement un utilisateur d'une équipe

Vous pouvez supprimer des membres de votre équipe. Toutes les autorisations ne seront plus héritées par l'utilisateur. Vous pourrez réintégrer l'utilisateur dans l'équipe ultérieurement.

**Note**

Lorsque vous supprimez un membre de l'équipe, les autorisations associées sont supprimées pour l'utilisateur pour tous les projets et ressources de l'espace.

Vous devez avoir le rôle d'administrateur de l'espace pour gérer les équipes.

Pour supprimer un membre de l'équipe

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace. Choisissez Paramètres, puis Teams.
3. Choisissez l'onglet Membres.
4. Choisissez le sélecteur situé à côté de l'utilisateur que vous souhaitez supprimer, puis sélectionnez Supprimer.
5. Entrez Supprimer dans le champ de saisie, puis choisissez Supprimer.

## Ajouter un groupe SSO à une équipe

Si votre espace est configuré comme un espace avec des utilisateurs et des groupes SSO gérés dans IAM Identity Center, vous pouvez ajouter un groupe SSO qui rejoindra l'espace en tant qu'équipe distincte.

**Note**

Lorsque vous choisissez de gérer les membres de l'équipe en ajoutant directement des utilisateurs AWS Builder ID ou des utilisateurs SSO, l'option permettant d'utiliser des groupes SSO n'est pas disponible. Pour ajouter des utilisateurs directement, consultez [Ajouter directement un utilisateur à une équipe](#).

Vous devez avoir le rôle d'administrateur de l'espace pour gérer les équipes.

Pour ajouter un groupe SSO en tant qu'équipe

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Sur la page de votre espace, sélectionnez Teams. Choisissez l'onglet SSO groups.
3. Choisissez les groupes SSO que vous souhaitez ajouter. Vous pouvez ajouter jusqu'à cinq groupes SSO.

## Supprimer une équipe

Vous pouvez supprimer une équipe dont vous n'avez plus besoin.

**Note**

Lorsque vous supprimez une équipe, les autorisations associées sont supprimées pour tous les membres de l'équipe, quels que soient les projets et les ressources de l'espace.

Vous devez avoir le rôle d'administrateur de l'espace pour gérer les équipes.

Supprimer une équipe

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace. Choisissez Paramètres, puis Teams.
3. Dans Actions, sélectionnez Supprimer l'équipe. Cela change le rôle de l'ensemble de l'équipe.
4. Sélectionnez Supprimer.

# Gestion des ressources des machines

Une ressource machine représente votre identité à partir de votre ressource autorisée lors de l'accès CodeCatalyst via SSO. Les ressources de la machine sont utilisées pour accorder des autorisations aux ressources de l'espace, telles que les plans et les flux de travail. Vous pouvez consulter les ressources de la machine dans votre espace, et vous pouvez choisir d'activer ou de désactiver les ressources de la machine pour votre espace. Par exemple, vous souhaitez peut-être désactiver une ressource de machine pour gérer l'accès, puis la réactiver ultérieurement.

Ces opérations sont disponibles pour les ressources de la machine dans les cas où une ressource de la machine doit être révoquée ou désactivée. Par exemple, si vous pensez que les informations d'identification ont été compromises, vous pouvez désactiver la ressource machine. En général, il n'est pas nécessaire d'utiliser ces opérations.

Vous devez avoir le rôle d'administrateur de l'espace pour afficher cette page et gérer les ressources de la machine au niveau de l'espace.

## Rubriques

- [Affichage des ressources de la machine](#)
- [Désactiver les ressources de la machine](#)
- [Activer les ressources de la machine](#)

## Affichage des ressources de la machine

Vous pouvez consulter la liste des ressources informatiques utilisées dans votre espace.

Vous devez avoir le rôle d'administrateur de l'espace pour gérer les ressources de la machine.

Pour afficher les ressources de la machine

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace, puis sélectionnez Paramètres. Choisissez Ressources de la machine.
3. Dans le menu déroulant, sélectionnez Action de flux de travail pour afficher uniquement les ressources de machine associées aux flux de travail. Choisissez Blueprint pour afficher uniquement les ressources de la machine pour les plans.

Vous pouvez également filtrer par nom à l'aide du champ Filtre.

## Désactiver les ressources de la machine

Vous pouvez choisir de désactiver les ressources de la machine utilisées dans votre espace.

### Important

La désactivation des ressources de la machine supprimera toutes les autorisations relatives à tous les plans ou flux de travail associés dans l'espace.

Vous devez avoir le rôle d'administrateur de l'espace pour gérer les ressources de la machine.

Pour désactiver les ressources de la machine

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace, puis sélectionnez Paramètres. Choisissez Ressources de la machine.
3. Choisissez l'une des options suivantes.

### Important

La désactivation des ressources de la machine supprimera toutes les autorisations relatives à tous les plans ou flux de travail associés dans l'espace.

- Pour désactiver individuellement, choisissez le sélecteur situé à côté d'une ou de plusieurs ressources de machine que vous souhaitez désactiver. Choisissez Désactiver, puis choisissez Cette ressource.
- Pour désactiver toutes les ressources, choisissez Désactiver, puis Toutes les ressources.
- Pour désactiver toutes les actions du flux de travail, choisissez Désactiver, puis sélectionnez Toutes les actions du flux de travail.
- Pour désactiver tous les plans, choisissez Désactiver, puis Tous les plans.

## Activer les ressources de la machine

Vous pouvez choisir d'activer les ressources de la machine qui sont utilisées dans votre espace et qui ont été désactivées.

Vous devez avoir le rôle d'administrateur de l'espace pour gérer les ressources de la machine.

Pour activer les ressources de la machine

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace, puis sélectionnez Paramètres. Choisissez Ressources de la machine.
3. Choisissez l'une des options suivantes.
  - Pour une activation individuelle, choisissez le sélecteur situé à côté d'une ou de plusieurs ressources de machine que vous souhaitez activer. Choisissez Activer, puis choisissez Cette ressource.
  - Pour activer toutes les ressources, choisissez Activer, puis Toutes les ressources.
  - Pour activer toutes les actions du flux de travail, choisissez Activer, puis sélectionnez Toutes les actions du flux de travail.
  - Pour activer tous les plans, choisissez Activer, puis Tous les plans.

## Administration des environnements de développement pour un espace

Tous les environnements de développement sont créés dans le cadre d'un projet au sein d'un espace. Les membres de l'espace peuvent créer leurs propres environnements de développement au sein d'un projet au niveau du référentiel source. Les administrateurs de l'espace peuvent ensuite utiliser la CodeCatalyst console Amazon pour afficher, modifier, supprimer et arrêter les environnements de développement au nom des membres de l'espace. En bref, les administrateurs de l'espace gèrent les environnements de développement au niveau de l'espace.

Considérations relatives à l'administration des environnements de développement

- Vous devez avoir le rôle d'administrateur de l'espace pour afficher la page Environnements de développement sous Paramètres et pour gérer les environnements de développement au niveau de l'espace.
- Les membres de l'espace gèrent les environnements de développement qu'ils créent dans les projets via leurs CodeCatalyst comptes. Lorsque vous administrez des environnements de développement en tant qu'administrateur d'espace, vous gérez ces ressources pour le compte des membres de l'espace.

- Les environnements de développement utilisent par défaut une configuration de calcul et de stockage spécifique. Pour plus d'informations sur la facturation et les tarifs de mise à niveau de votre configuration, consultez la [page de CodeCatalyst tarification d'Amazon](#).

Pour d'autres considérations relatives aux environnements de développement, notamment l'arrêt des instances en cours d'exécution, la configuration de calcul par défaut, la mise à niveau de votre capacité de calcul, les coûts engagés et la configuration des délais d'attente, consultez [Environnements de développement dans CodeCatalyst](#)

## Rubriques

- [Visualisation des environnements de développement pour votre espace](#)
- [Modification d'un environnement de développement pour votre espace](#)
- [Arrêter un environnement de développement pour votre espace](#)
- [Supprimer un environnement de développement pour votre espace](#)

## Visualisation des environnements de développement pour votre espace

Vous pouvez consulter le type, le statut et les détails de tous les environnements de développement de votre espace. Pour plus d'informations sur la création et l'exécution d'environnements de développement, consultez [Création d'un environnement de développement](#).

Vous devez avoir le rôle d'administrateur de l'espace pour afficher cette page et pour gérer les environnements de développement au niveau de l'espace.

Pour afficher les environnements de développement de votre espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.

### Tip

Si vous appartenez à plusieurs espaces, choisissez-en un dans la barre de navigation supérieure.

3. Choisissez Paramètres, puis Dev Environments.

La page répertorie tous les environnements de développement de votre espace. Vous pouvez afficher le nom de la ressource, l'alias de ressource le cas échéant, le type d'IDE, le calcul et le stockage par défaut ou configurés, ainsi que le délai d'expiration configuré pour chaque environnement de développement.

## Modification d'un environnement de développement pour votre espace

Vous pouvez modifier la configuration d'un environnement de développement, par exemple le délai d'expiration configuré, le cas échéant, pour qu'un environnement de développement inactif cesse de fonctionner. Pour plus d'informations sur la modification d'un environnement de développement, consultez [Modification d'un environnement de développement](#).

Vous devez avoir le rôle d'administrateur de l'espace pour afficher cette page et pour gérer les environnements de développement au niveau de l'espace.

Pour modifier les environnements de développement de votre espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.

### Tip

Si vous appartenez à plusieurs espaces, choisissez-en un dans la barre de navigation supérieure.

3. Choisissez Paramètres, puis Dev Environments.
4. Choisissez le sélecteur situé à côté de l'environnement de développement que vous souhaitez gérer. Choisissez Modifier.
5. Apportez les modifications souhaitées au délai de calcul ou d'inactivité de l'environnement de développement.
6. Choisissez Enregistrer.

## Arrêter un environnement de développement pour votre espace

Vous pouvez arrêter un environnement de développement en cours d'exécution avant qu'il ne devienne inactif si l'environnement de développement est configuré pour avoir un délai d'expiration.



Sinon, un environnement de développement dont le délai d'expiration est expiré sera déjà arrêté. Pour plus d'informations sur l'arrêt d'un environnement de développement, consultez [Arrêt d'un environnement de développement](#).

Vous devez avoir le rôle d'administrateur de l'espace pour afficher cette page et pour gérer les environnements de développement au niveau de l'espace.

Pour arrêter un environnement de développement dans votre espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.

 Tip

Si vous appartenez à plusieurs espaces, choisissez-en un dans la barre de navigation supérieure.

3. Choisissez Paramètres, puis Dev Environments.
4. Choisissez le sélecteur situé à côté de l'environnement de développement que vous souhaitez gérer. Choisissez Arrêter.

## Supprimer un environnement de développement pour votre espace

Vous pouvez supprimer un environnement de développement dont vous n'avez plus besoin ou qui n'a plus de propriétaire. Pour plus d'informations sur les considérations relatives à la suppression d'un environnement de développement, consultez [Suppression d'un environnement de développement](#).

Vous devez avoir le rôle d'administrateur de l'espace pour afficher cette page et pour gérer les environnements de développement au niveau de l'espace.

Pour supprimer des environnements de développement dans votre espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.

**i** Tip

Si vous appartenez à plusieurs espaces, choisissez-en un dans la barre de navigation supérieure.

3. Choisissez Paramètres, puis Dev Environments.
4. Choisissez le sélecteur à côté de l'environnement de développement que vous souhaitez gérer. Sélectionnez Delete (Supprimer). Pour confirmer, tapez `delete`, puis choisissez Supprimer.

## Quotas de places dans CodeCatalyst

Le tableau suivant décrit les quotas et les limites des espaces sur Amazon CodeCatalyst. Pour plus d'informations sur les quotas sur Amazon CodeCatalyst, consultez [Quotas pour CodeCatalyst](#).

Nombre d'espaces actifs par utilisateur Région AWS	Cinq
Nombre de créations d'espace par région, par mois et par utilisateur	Cinq
Descriptions des espaces	<p>Les descriptions des espaces sont facultatives. S'ils sont spécifiés, ils doivent comporter entre 0 et 200 caractères. Ils peuvent contenir n'importe quelle combinaison de lettres, de chiffres, d'espaces, de points, de traits de soulignement, de virgules, de tirets et les caractères spéciaux suivants :</p> <p>? &amp; \$ % + = / \ ; : \n \t \r</p>
Noms des espaces	<p>Les noms des espaces doivent être uniques d'un côté à l'autre CodeCatalyst. Vous ne pouvez pas réutiliser les noms des espaces supprimés.</p> <p>Les noms d'espaces doivent comporter entre 3 et 63 caractères. Ils doivent également</p>

commencer par un caractère alphanumérique. Les noms d'espaces peuvent contenir n'importe quelle combinaison de lettres, de chiffres, de points, de traits de soulignement et de tirets. Ils ne peuvent contenir aucun des caractères suivants :

! ? @ # \$ % ^ & \* ( ) + = { } [ ]  
| \ > < ~ ` ' " ; :

# Projets en CodeCatalyst

Vous utilisez des projets sur Amazon CodeCatalyst pour créer un espace de collaboration dans lequel les équipes de développement peuvent effectuer des tâches de développement avec des flux de travail et des référentiels d'intégration continue/livraison continue (CI/CD) partagés. Lorsque vous créez un projet, vous pouvez ajouter, mettre à jour ou supprimer des ressources. Vous pouvez également suivre l'avancement du travail de votre équipe. Vous pouvez avoir plusieurs projets au sein d'un même espace.

Les espaces CodeCatalyst sont composés de projets. Vous pouvez voir tous les projets de votre espace, mais vous ne pouvez utiliser que les projets dont vous êtes membre. Lorsque vous créez un projet, des rôles par défaut sont générés pour votre projet, que vous attribuez aux utilisateurs que vous invitez à rejoindre votre projet.

- Toute personne affectée au projet avec un rôle de projet, tel que le rôle de contributeur, peut accéder aux ressources du projet, telles qu'un référentiel de sources.
- Toute personne ayant le rôle d'administrateur de l'espace Administrateur de projet ou d'administrateur de projet peut envoyer des invitations à rejoindre un projet.
- Les utilisateurs dotés du rôle d'administrateur de projet peuvent suivre l'activité, le statut et les autres paramètres des ressources partagées.
- Les utilisateurs dotés du rôle d'accès limité peuvent gérer les attributions de projet relatives aux fonctionnalités, aux corrections de code et aux tests dans le cadre des flux de travail CI/CD.

Les flux de travail sont utilisés pour créer, tester et publier ou mettre à jour des applications sous forme de pipeline CI/CD. Vous pouvez assembler des flux de travail en ajoutant des actions qui transfèrent et agissent sur vos artefacts sources. Lorsque vous exécutez des actions, les ressources cloud de votre projet sont utilisées pour fournir une capacité de calcul à la demande pour les actions de votre flux de travail. Vous pouvez configurer davantage de flux de travail CI/CD en fonction de l'activité et du résultat que vous souhaitez configurer. Par exemple, vous pouvez créer un flux de travail pour les actions de génération et de test uniquement, dans lequel vous pouvez consulter les résultats des tests et terminer le flux de travail sans déploiement pendant que vous corrigez les bogues. Vous pouvez ensuite créer un autre flux de travail pour créer et déployer votre application dans un environnement intermédiaire.

Lorsque vous créez un projet, vous pouvez utiliser un plan pour créer un projet qui contient un exemple de code et crée des ressources, ou vous pouvez commencer par un projet vide. Si vous

créez un projet à l'aide d'un plan, le plan que vous choisissez détermine les ressources qui sont ajoutées à votre projet et les outils que CodeCatalyst crée ou configure afin que vous puissiez suivre et utiliser les ressources de votre projet. Vous pouvez ajouter ou supprimer des ressources manuellement après avoir créé un projet. Les ressources suivantes peuvent être créées ou configurées par CodeCatalyst :

- **Problèmes** — Vous suivez le travail lié à votre projet dans un enregistrement distinct appelé problèmes. Vous pouvez créer des problèmes liés aux fonctionnalités, aux tâches, aux bogues et à toute autre tâche liée à vos projets.
- **Notifications** — Vous configurez les notifications en choisissant la ressource que vous souhaitez surveiller, l'événement que vous souhaitez surveiller et le client ou le courrier électronique de destination où vous souhaitez recevoir les notifications.
- **Recherche** : vous pouvez rechercher du code, des problèmes, des utilisateurs, des pull requests et des packages dans votre projet. Vous pouvez effectuer une recherche dans un seul projet ou dans tous les projets.
- **Référentiel source** — Vous travaillez avec votre code source dans un référentiel pour votre projet. Lorsque vous validez une modification du code source ou que vous fusionnez une pull request dans une branche spécifiée, votre source est CodeCatalyst mise à jour.

Chaque projet suit l'activité du projet sous la forme d'une liste d'événements par utilisateur, par exemple lorsqu'un projet est créé ou qu'une ressource est modifiée. L'activité du projet est surveillée et agrégée au niveau de l'espace. Pour plus d'informations sur l'utilisation des données d'activité, consultez [Afficher tous les projets](#).

Si votre projet utilise des AWS ressources, vous pouvez connecter votre CodeCatalyst compte à un AWS compte dans lequel vous disposez des autorisations administratives nécessaires pour intégrer des ressources à votre projet.

Vous pouvez ajouter des référentiels sources, des problèmes et d'autres ressources à votre projet après l'avoir créé. Vous devez avoir le rôle d'administrateur de l'espace pour créer des projets.

## Création d'un projet sur Amazon CodeCatalyst

Avec les CodeCatalyst projets, vous pouvez effectuer des tâches de développement avec des flux de travail et des référentiels d'intégration continue/livraison continue (CI/CD) partagés, gérer les ressources, suivre les problèmes et ajouter des utilisateurs.

Avant de créer un projet, vous devez avoir le rôle d'administrateur de l'espace ou d'utilisateur avancé.

## Rubriques

- [Création d'un projet à l'aide d'un plan](#)
- [Création d'un projet vide dans Amazon CodeCatalyst](#)
- [Création d'un projet avec un GitHub référentiel lié](#)

## Création d'un projet à l'aide d'un plan

Vous pouvez fournir toutes les ressources de votre projet et un exemple de code avec un plan de projet. Pour plus d'informations sur les plans, consultez la référence du plan à l'adresse. [Référence du plan de projet](#)

Pour créer un projet à l'aide d'un plan

1. Dans la CodeCatalyst console, accédez à l'espace dans lequel vous souhaitez créer un projet.
2. Sur le tableau de bord de l'espace, choisissez Créer un projet.
3. Choisissez Commencer par un plan.
4. Choisissez un plan, puis cliquez sur Suivant.
5. Sous Donnez un nom à votre projet, entrez le nom que vous souhaitez attribuer à votre projet et les noms de ressources associés. Le nom doit être unique dans votre espace.
6. Sous Ressources du projet, configurez les paramètres communs du projet.
7. (Facultatif) Pour afficher les fichiers de définition mis à jour en fonction des paramètres de projet que vous avez sélectionnés, choisissez Afficher le code ou Afficher le flux de travail dans Générer un aperçu du projet.
8. (Facultatif) Choisissez Afficher les détails dans la fiche du plan pour afficher des détails spécifiques sur le plan, tels qu'un aperçu de l'architecture du plan, les connexions et autorisations requises, ainsi que le type de ressources créé par le plan.
9. Sélectionnez Create a project (Créer un projet).

Pour plus d'informations sur les plans de projet, consultez [Référence du plan de projet](#).

## Création d'un projet vide dans Amazon CodeCatalyst

Vous pouvez créer un projet vide sans ressources et ajouter manuellement les ressources souhaitées ultérieurement.

Avant de créer un projet, vous devez avoir le rôle d'administrateur de l'espace ou d'utilisateur avancé.

Pour créer un projet vide

1. Accédez à l'espace dans lequel vous souhaitez créer un projet.
2. Sur le tableau de bord de l'espace, choisissez Créer un projet.
3. Choisissez Commencer à zéro.
4. Sous Donnez un nom à votre projet, entrez le nom que vous souhaitez attribuer à votre projet. Le nom doit être unique dans votre espace.
5. Sélectionnez Create a project (Créer un projet).

## Création d'un projet avec un GitHub référentiel lié

Vous pouvez créer un nouveau CodeCatalyst projet lié à un référentiel GitHub source. Vous pouvez ensuite utiliser votre référentiel GitHub source lié dans votre CodeCatalyst projet.

Avant de créer un CodeCatalyst projet, vous devez avoir le rôle d'administrateur de l'espace ou d'utilisateur avancé. Pour plus d'informations, consultez [Création d'un espace prenant en charge les utilisateurs de AWS Builder ID](#) et [Inviter un utilisateur directement dans un espace](#).

Vous devez déjà avoir un GitHub compte, et vous devez avoir déjà créé le référentiel vers lequel vous souhaitez créer le lien.

Pour créer un projet CodeCatalyst lié à un référentiel source de votre GitHub compte, vous devez effectuer les trois tâches suivantes :

1. Installez l'extension GitHub Repositories.
2. Connectez votre GitHub compte à CodeCatalyst.
3. Créez un CodeCatalyst projet lié à votre GitHub compte.

Pour installer l'extension GitHub Repositories

1. Accédez à l'espace dans lequel vous souhaitez créer un projet.

2. Sur le tableau de bord de l'espace, choisissez Créer un projet.
3. Choisissez Apportez votre propre code.

Si l'extension GitHub des référentiels n'est pas déjà installée, une invite d'installation s'affiche.

4. Choisissez Installer. Vérifiez les autorisations requises par l'extension, et si vous souhaitez continuer, sélectionnez à nouveau Installer.

Après avoir installé l'extension GitHub Repositories, l'étape suivante consiste à connecter votre GitHub compte à votre CodeCatalyst espace.

#### Note

Vous ne pouvez pas utiliser de GitHub référentiels vides ou archivés avec des CodeCatalyst projets. L'extension GitHub Repositories n'est pas compatible avec les référentiels GitHub Enterprise Server.

Pour connecter votre GitHub compte à CodeCatalyst

1. Sur la page Créer un projet dans CodeCatalyst, si aucun GitHub compte n'est connecté, un message s'affiche. Choisissez Connect GitHub account pour accéder au site externe pour GitHub.
2. Connectez-vous à votre GitHub compte à l'aide de vos GitHub informations d'identification, puis choisissez le compte sur lequel vous souhaitez installer Amazon CodeCatalyst.

#### Tip

Si vous avez déjà connecté un GitHub compte à l'espace, il ne vous sera pas demandé de procéder à une nouvelle autorisation. Vous verrez plutôt une boîte de dialogue vous demandant où vous souhaitez installer l'extension si vous êtes membre ou collaborateur de plusieurs GitHub espaces, ou la page de configuration de l' CodeCatalyst application Amazon si vous n'appartenez qu'à un seul GitHub espace. Configurez l'application pour l'accès au référentiel que vous souhaitez autoriser, puis choisissez Enregistrer. Si le bouton Enregistrer n'est pas actif, modifiez la configuration, puis réessayez.

3. Choisissez si vous souhaitez autoriser l'accès CodeCatalyst à tous les référentiels actuels et futurs, ou choisissez les GitHub référentiels spécifiques dans lesquels vous souhaitez les utiliser.



CodeCatalyst L'option par défaut consiste à inclure tous les GitHub référentiels du GitHub compte, y compris les futurs référentiels auxquels le compte aura accès. CodeCatalyst

4. Vérifiez les autorisations accordées à CodeCatalyst, puis choisissez Installer.

Après avoir connecté votre GitHub compte à CodeCatalyst, vous pourrez lier les GitHub référentiels de ce compte à vos CodeCatalyst projets.

Pour créer votre projet

1. Sur la page Créer un projet, dans le menu déroulant du GitHub compte, effectuez l'une des opérations suivantes :
  - Choisissez un GitHub compte auquel vous êtes déjà connecté CodeCatalyst.
  - (Facultatif) Si vous ne voyez pas le GitHub compte que vous souhaitez utiliser, choisissez Connect a GitHub account pour accéder à la page des extensions CodeCatalyst. Pour plus d'informations, consultez [Utilisation de GitHub référentiels dans CodeCatalyst](#).
2. Dans le menu déroulant du GitHub référentiel, les GitHub référentiels de votre GitHub compte connecté sont affichés dans le menu déroulant. Choisissez le GitHub référentiel que vous souhaitez lier à votre projet.
3. Dans le champ de saisie de texte Nommez votre projet, entrez le nom que vous souhaitez attribuer à votre projet. Le nom doit être unique dans votre espace.
4. Sélectionnez Create a project (Créer un projet).

Une fois que votre projet est prêt, vous pouvez ajouter des ressources et des tâches.

- Pour en savoir plus sur les flux de travail CI/CD créés avec votre projet, consultez. [Commencer à utiliser les flux de travail dans CodeCatalyst](#)
- Pour utiliser des actions de génération similaires à celles de votre nouveau projet qui déploient des artefacts de génération dans un compartiment Amazon S3, consultez [Création à l'aide de flux de travail dans CodeCatalyst](#) et [Tutoriel : télécharger des artefacts sur Amazon S3](#).
- Pour commencer avec un projet vide et travailler sur le déploiement d'une application sans serveur similaire avec un déploiement en AWS CloudFormation pile, voir [Tutoriel : Déployer une application sans serveur à l'aide de AWS CloudFormation](#).
- Pour ajouter un tableau de planification des problèmes, voir [Problèmes liés à CodeCatalyst](#).

- Pour consulter l'aperçu du projet, l'état du projet, les activités récentes de l'équipe et le travail assigné, voir [Visualisation d'un projet](#).
- Pour afficher le code source ou créer une pull request, consultez [Référentiels de sources dans CodeCatalyst](#).
- Pour configurer des notifications qui envoient des alertes d'état en cas de réussite ou d'échec de l'exécution du flux de travail, voir [Gestion des notifications sur Amazon CodeCatalyst](#).
- Pour inviter des membres à rejoindre votre projet, consultez [Gérer les membres du projet](#).
- Pour configurer des environnements de développement, voir [Environnements de développement dans CodeCatalyst](#).

## Visualisation d'un projet

Depuis votre CodeCatalyst espace, vous pouvez consulter les détails de chaque projet pour lequel vous avez des autorisations de projet.

Pour consulter un projet, vous devez être membre du projet ou avoir le rôle d'administrateur de l'espace pour cet espace.

Si vous n'avez pas encore créé de projet, consultez [Création d'un projet sur Amazon CodeCatalyst](#). Vous devez avoir le rôle d'administrateur de l'espace dans lequel vous souhaitez créer un projet.

- Dans l'aperçu du projet, vous pouvez consulter les membres du projet, les référentiels sources, les exécutions de flux de travail, les pull requests ouvertes, les environnements de développement du projet et les problèmes.
- Dans les paramètres du projet, vous pouvez afficher et gérer les détails du projet, supprimer le projet, inviter de nouveaux membres au projet, gérer les membres du projet et configurer les notifications.

## Afficher les tâches du projet et les environnements de développement

Pour afficher un résumé des tâches du projet, telles que les problèmes en suspens et les pull requests qui vous sont assignés ou créés par vous, ainsi que les environnements de développement associés au projet, utilisez la console.

Pour consulter un projet, vous devez être membre du projet ou avoir le rôle d'administrateur de l'espace pour cet espace.

Pour consulter vos référentiels sources, les exécutions de flux de travail, les problèmes, les pull requests, les environnements de développement et les problèmes

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à l'espace contenant le projet que vous souhaitez consulter. Sous Projets, sélectionnez votre projet.
3. Dans le panneau de navigation, sélectionnez Présentation.
4. Consultez les tâches de projet qui vous ont été assignées et que vous avez créées.
  - Afficher la liste des membres + Afficher tout pour voir la liste des membres du projet.
  - Consultez la fiche Référentiels pour afficher les référentiels sources associés au projet.
  - Consultez la carte Workflow runs pour afficher les flux de travail associés au projet.
  - Consultez la carte Open pull requests pour consulter un résumé de l'état du référentiel de code, en plus des pull requests qui vous ont été attribuées et que vous avez créées.
  - Consultez la fiche Mes environnements de développement pour afficher un résumé des environnements de développement associés au projet.
  - Consultez la fiche Problèmes pour afficher un résumé des tâches qui vous sont assignées ou des tâches que vous avez créées.

## Afficher tous les projets

Dans la liste des projets de votre espace, vous pouvez voir tous les projets pour lesquels vous disposez d'autorisations.

Pour afficher un résumé des tâches du projet, telles que les problèmes en suspens et les pull requests qui vous sont assignés ou créés par vous, ainsi que les environnements de développement associés au projet, utilisez la console.

Pour consulter un projet, vous devez être membre du projet ou avoir le rôle d'administrateur de l'espace pour cet espace.

Pour consulter vos référentiels sources, les exécutions de flux de travail, les problèmes, les pull requests, les environnements de développement et les problèmes

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à l'espace contenant le projet que vous souhaitez consulter. Sous Projets, sélectionnez votre projet.

3. Dans le volet de navigation, sélectionnez Paramètres du projet.
4. Affichez le nom, le chemin, l'identifiant et la description du projet.

## Affichage des paramètres du projet

Dans les paramètres du projet, vous pouvez afficher les membres du projet, les référentiels sources, les exécutions de flux de travail, les pull requests ouvertes, les environnements de développement du projet et les problèmes.

Pour afficher un résumé des tâches du projet, telles que les problèmes en suspens et les pull requests qui vous sont assignés ou créés par vous, ainsi que les environnements de développement associés au projet, utilisez la console.

Pour consulter vos référentiels sources, les exécutions de flux de travail, les problèmes, les pull requests, les environnements de développement et les problèmes

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à l'espace contenant le projet que vous souhaitez consulter. Sous Projets, sélectionnez votre projet.
3. Dans le volet de navigation, sélectionnez Paramètres du projet.
4. Affichez le nom, le chemin, l'identifiant et la description du projet.

## Passage à un autre projet dans CodeCatalyst

Pour passer à un autre projet, utilisez la console pour choisir parmi une liste de projets auxquels vous avez accès.

Pour passer à un autre projet

1. Dans la CodeCatalyst console, choisissez le sélecteur de projet en haut.
2. Développez le menu déroulant et choisissez le projet vers lequel vous souhaitez accéder.

## Supprimer un projet sur Amazon CodeCatalyst

Vous pouvez supprimer un projet pour supprimer tout accès aux ressources du projet. Vous devez avoir le rôle d'administrateur de l'espace ou d'administrateur de projet pour supprimer un projet.

Une fois que vous avez supprimé un projet, les membres du projet ne pourront pas accéder aux ressources du projet et tous les flux de travail demandés par des référentiels sources tiers seront interrompus.

Pour supprimer votre projet

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à l'espace contenant le projet que vous souhaitez consulter. Sous Projets, sélectionnez votre projet.
3. Dans le volet de navigation, sélectionnez Paramètres du projet.
4. Choisissez Supprimer le projet.
5. Entrez **delete** pour confirmer la suppression.
6. Choisissez Supprimer le projet.

## Gérer les membres du projet

Vous pouvez gérer les membres de vos projets à l'aide de la CodeCatalyst console Amazon. Vous pouvez ajouter ou supprimer des utilisateurs, gérer les rôles des membres actuels, envoyer des invitations à rejoindre votre projet et annuler des invitations qui n'ont pas encore été acceptées.

Sur la page des membres réservée aux utilisateurs de l'espace et du projet, les utilisateurs peuvent avoir plusieurs rôles. Les utilisateurs ayant plusieurs rôles afficheront un indicateur lorsqu'ils ont plusieurs rôles, et le rôle ayant le plus d'autorisations sera affiché en premier.

## Afficher les membres d'un projet

Lorsque vous ajoutez un utilisateur à votre projet, vous attribuez un rôle qui accorde des autorisations au projet comme suit :

- Le rôle d'administrateur de projet dispose de toutes les autorisations dans un projet. Attribuez ce rôle uniquement aux utilisateurs qui doivent administrer tous les aspects d'un projet, y compris la modification des paramètres du projet, la gestion des autorisations du projet et la suppression du projet. Pour plus d'informations, consultez [Rôle d'administrateur de projet](#).
- Le rôle de contributeur dispose des autorisations requises pour travailler dans un projet. Attribuez ce rôle aux utilisateurs qui doivent travailler avec du code, des flux de travail, des problèmes et des actions dans le cadre d'un projet. Pour plus d'informations, consultez [Rôle du contributeur](#).

- Le rôle de réviseur dispose d'autorisations de révision. Pour plus de détails, consultez [Rôle du réviseur](#).
- Le rôle Lecture seule dispose d'autorisations de lecture. Pour plus de détails, consultez [Rôle en lecture seule](#).


Il n'est pas nécessaire d'inviter un utilisateur ayant le rôle d'administrateur de l'espace à rejoindre votre projet, car il dispose déjà d'un accès implicite à tous les projets de l'espace.

Lorsque vous invitez un utilisateur à rejoindre votre projet (sans lui attribuer le rôle d'administrateur de l'espace), l'utilisateur apparaît dans le tableau des membres du projet sous les projets et dans le tableau des membres du projet sous les espaces.

Pour afficher les utilisateurs et les rôles de votre espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à l'espace contenant le projet que vous souhaitez consulter. Sous Projets, sélectionnez votre projet.
3. Dans le volet de navigation, sélectionnez Paramètres du projet.
4. Choisissez l'onglet Membres.

Le tableau des membres du projet indique tous les membres qui jouent un rôle dans un projet.

 Tip

Si vous avez le rôle d'administrateur de l'espace, vous pouvez voir les projets auxquels vous avez été directement invité. Accédez aux paramètres du projet, puis sélectionnez Mes projets.

Le tableau des administrateurs de l'espace indique les utilisateurs ayant le rôle d'administrateur de l'espace. Ces utilisateurs sont automatiquement (implicitement) affectés à tous les projets de l'espace et n'ont aucun rôle dans un projet.

Dans la colonne État, les valeurs suivantes sont valides :

- Invité : CodeCatalyst a envoyé l'invitation mais l'utilisateur n'a pas encore accepté ou refusé.
- Membre — L'utilisateur a accepté l'invitation.

## Rubriques

- [Inviter un utilisateur à rejoindre votre projet](#)
- [Annulation d'une invitation](#)
- [Supprimer un utilisateur de votre projet](#)
- [Accepter ou refuser une invitation pour un projet](#)

## Inviter un utilisateur à rejoindre votre projet

Vous pouvez utiliser la console pour inviter des utilisateurs à rejoindre votre projet. Vous pouvez inviter des membres de votre espace ou ajouter des noms provenant de l'extérieur de votre espace.

Pour inviter des utilisateurs à rejoindre votre projet, vous devez être connecté avec le rôle d'administrateur du projet ou d'administrateur de l'espace.

Il n'est pas nécessaire d'inviter un utilisateur ayant le rôle d'administrateur de l'espace à rejoindre votre projet, car il dispose déjà d'un accès implicite à tous les projets de l'espace.

Lorsque vous invitez un utilisateur à rejoindre votre projet (sans lui attribuer le rôle d'administrateur de l'espace), l'utilisateur apparaît dans le tableau des membres du projet sous les projets et dans le tableau des membres du projet sous les espaces.

Pour inviter un membre à rejoindre votre projet depuis l'onglet Paramètres du projet

1. Accédez à votre projet.

### Tip

Vous pouvez choisir le projet à afficher dans la barre de navigation supérieure.

2. Dans le volet de navigation, sélectionnez Paramètres du projet.
3. Choisissez l'onglet Membres.
4. Dans Membres du projet, sélectionnez Inviter un nouveau membre.
5. Entrez l'adresse e-mail du nouveau membre, choisissez le rôle de ce membre, puis choisissez Inviter. Pour plus d'informations sur les rôles, consultez [Utilisation des rôles sur Amazon CodeCatalyst](#).

Pour inviter un membre à rejoindre votre projet depuis la page d'aperçu du projet

1. Accédez à votre projet.

 Tip

Vous pouvez choisir le projet à afficher dans la barre de navigation supérieure.

2. Cliquez sur le bouton Membres +.
3. Entrez l'adresse e-mail du nouveau membre, choisissez le rôle de ce membre, puis choisissez Inviter. Pour plus d'informations sur les rôles , consultez [Utilisation des rôles sur Amazon CodeCatalyst](#).


## Annulation d'une invitation

Si vous avez récemment envoyé une invitation, vous pouvez l'annuler tant que l'invitation n'a pas encore été acceptée.

Pour gérer les invitations à des projets, vous devez avoir le rôle d'administrateur de projet ou d'administrateur de l'espace.

Pour annuler l'invitation d'un membre du projet

1. Accédez au projet pour lequel vous avez envoyé une invitation que vous souhaitez annuler.
2. Dans le volet de navigation, sélectionnez Paramètres du projet.
3. Consultez l'onglet Membres et vérifiez que le membre a le statut Invité.

 Note

Vous ne pouvez annuler qu'une invitation qui n'a pas encore été acceptée.

4. Choisissez l'option à côté de la ligne avec le membre invité, puis cliquez sur Annuler l'invitation.
5. Une fenêtre de confirmation s'affiche. Choisissez Annuler l'invitation pour confirmer.

## Supprimer un utilisateur de votre projet

Vous pouvez utiliser la console pour supprimer un utilisateur de votre projet.



Pour supprimer un utilisateur de votre projet, vous devez être connecté avec le rôle d'administrateur du projet ou d'administrateur de l'espace.

#### Note

La suppression d'un utilisateur de tous les projets d'un espace entraîne automatiquement le retrait de l'utilisateur de cet espace.

Pour supprimer un utilisateur d'un projet

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à l'espace contenant le projet que vous souhaitez consulter. Sous Projets, sélectionnez votre projet.
3. Dans le volet de navigation, sélectionnez Paramètres du projet.
4. Choisissez l'onglet Membres.
5. Choisissez le sélecteur situé à côté du profil que vous souhaitez supprimer, puis sélectionnez Supprimer.
6. Confirmez que vous souhaitez supprimer l'utilisateur, puis choisissez Supprimer.

## Accepter ou refuser une invitation pour un projet

Il se peut que vous receviez une invitation par e-mail pour rejoindre un CodeCatalyst projet Amazon. Vous pouvez accepter ou refuser l'invitation.

Pour accepter ou refuser une invitation

1. Ouvrez l'e-mail d'invitation.
2. Choisissez le lien du projet dans l'e-mail.
3. Choisissez Accepter ou Refuser.

Si vous choisissez Refuser, un e-mail est envoyé au compte de gestion du projet pour l'informer que vous avez refusé l'invitation.

# Gestion des équipes pour les projets

Après avoir créé un projet, vous pouvez ajouter des équipes. Les équipes vous permettent de regrouper les utilisateurs afin qu'ils puissent partager les autorisations et gérer les projets, le suivi des problèmes, les rôles et les ressources en CodeCatalyst tant que membres du projet et de l'espace.

Vous devez avoir le rôle d'administrateur de projet pour gérer les équipes de votre projet.

## Rubriques

- [Ajouter une équipe à un projet](#)
- [Gérer un rôle de projet pour une équipe](#)
- [Supprimer un rôle de projet pour une équipe](#)

## Ajouter une équipe à un projet

Vous pouvez gérer des équipes où les membres de l'équipe peuvent accéder aux ressources de votre projet.

Sur la page des membres réservée aux utilisateurs de l'espace et du projet, les utilisateurs peuvent avoir plusieurs rôles. Les utilisateurs ayant plusieurs rôles afficheront un indicateur lorsqu'ils ont plusieurs rôles, et le rôle ayant le plus d'autorisations sera affiché en premier.

### Pour ajouter une équipe

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre projet. Choisissez Paramètres du projet, puis Teams.
3. Choisissez Ajouter une équipe.
4. Dans Équipe, choisissez une équipe dans la liste des équipes disponibles.
5. Sous Rôle dans le projet, choisissez un rôle dans la liste des rôles de projet disponibles dans CodeCatalyst.
  - Administrateur de projet — Pour plus de détails, voir [Rôle d'administrateur de projet](#).
  - Contributeur — Pour plus de détails, voir [Rôle du contributeur](#).
  - Réviseur — Pour plus de détails, voir [Rôle du réviseur](#).
  - Lecture seule — Pour plus de détails, voir [Rôle en lecture seule](#).
6. Choisissez Ajouter une équipe.

## Gérer un rôle de projet pour une équipe

Une équipe peut disposer d'autorisations de rôle, telles que Power User, dans un espace. Vous pouvez modifier le rôle spatial d'une équipe, mais notez que tous les membres de l'équipe hériteront de ces autorisations.

Pour ajouter ou modifier un rôle dans un projet

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace. Choisissez Paramètres du projet, puis Teams.
3. Pour modifier un rôle, choisissez le sélecteur situé à côté de l'équipe dans cette liste, puis choisissez Changer de rôle. Pour ajouter un rôle, choisissez Ajouter un rôle dans le projet. Dans Projet, choisissez le projet que vous souhaitez ajouter et dans Rôle, choisissez le rôle. Choisissez l'un des rôles de projet disponibles :
  - Administrateur de projet - Pour plus de détails, voir [Rôle d'administrateur de projet](#).
  - Contributeur - Pour plus de détails, voir [Rôle du contributeur](#).
  - Réviseur - Pour plus de détails, voir [Rôle du réviseur](#).
  - Lecture seule - Pour plus de détails, voir [Rôle en lecture seule](#).
4. Choisissez Enregistrer.

## Supprimer un rôle de projet pour une équipe

Dans CodeCatalyst, vous pouvez consulter les rôles de votre équipe dans le projet. Vous pouvez également consulter les membres d'une équipe. Vous pouvez supprimer le rôle d'une équipe dans le projet.

Pour supprimer un rôle dans le projet

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre espace. Choisissez Paramètres du projet, puis Teams.
3. Choisissez l'onglet Rôles du projet.
4. Choisissez le rôle que vous souhaitez supprimer.

**⚠ Important**

La suppression d'un rôle dans une équipe entraîne la suppression des autorisations associées pour tous les utilisateurs de l'équipe.

5. Choisissez Enregistrer.

## Gestion des ressources de la machine

Une ressource machine représente votre identité à partir de votre ressource autorisée lors de l'accès CodeCatalyst via SSO. Les ressources de la machine sont utilisées pour accorder des autorisations aux ressources de votre projet, telles que les plans et les flux de travail. Vous pouvez consulter les ressources de machine de votre projet, et vous pouvez choisir d'activer ou de désactiver les ressources de machine pour votre projet. Par exemple, vous souhaitez peut-être désactiver une ressource de machine pour gérer l'accès, puis la réactiver ultérieurement.

Ces opérations sont disponibles pour les ressources de la machine dans les cas où une ressource de la machine doit être révoquée ou désactivée. Par exemple, si vous pensez que les informations d'identification ont été compromises, vous pouvez désactiver la ressource machine. En général, il n'est pas nécessaire d'utiliser ces opérations.

Vous devez avoir le rôle d'administrateur de l'espace ou le rôle d'administrateur du projet pour afficher cette page et gérer les ressources de la machine au niveau du projet.

### Rubriques

- [Affichage des ressources de la machine](#)
- [Désactiver les ressources de la machine](#)
- [Activer les ressources de la machine](#)

## Affichage des ressources de la machine

Vous pouvez consulter la liste des ressources informatiques utilisées dans votre projet.

Vous devez avoir le rôle d'administrateur de l'espace ou le rôle d'administrateur du projet.

## Pour afficher les ressources de la machine

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre projet, puis sélectionnez Paramètres du projet. Choisissez Ressources de la machine.
3. Dans le menu déroulant, sélectionnez Action de flux de travail pour afficher uniquement les ressources de machine associées aux flux de travail. Choisissez Blueprint pour afficher uniquement les ressources de la machine associées aux Blueprints.

Vous pouvez également filtrer par nom à l'aide du champ Filtre.

## Désactiver les ressources de la machine

Vous pouvez choisir de désactiver les ressources de machine utilisées dans votre projet.

### Important

La désactivation des ressources de la machine supprimera toutes les autorisations relatives à tous les plans ou flux de travail associés dans l'espace.

Vous devez avoir le rôle d'administrateur de l'espace ou le rôle d'administrateur du projet.

### Pour désactiver les ressources de la machine

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre projet, puis sélectionnez Paramètres du projet. Choisissez Ressources de la machine.
3. Choisissez l'une des options suivantes.

### Important

La désactivation des ressources de la machine supprimera toutes les autorisations relatives à tous les plans ou flux de travail associés dans l'espace.

- Pour désactiver individuellement, choisissez le sélecteur situé à côté d'une ou de plusieurs ressources de machine que vous souhaitez désactiver. Choisissez Désactiver, puis choisissez Cette ressource.
- Pour désactiver toutes les ressources, choisissez Désactiver, puis Toutes les ressources.
- Pour désactiver toutes les actions du flux de travail, choisissez Désactiver, puis sélectionnez Toutes les actions du flux de travail.
- Pour désactiver tous les plans, choisissez Désactiver, puis Tous les plans.

## Activer les ressources de la machine

Vous pouvez choisir d'activer les ressources de machine qui sont utilisées dans votre projet et qui ont été désactivées.

Vous devez avoir le rôle d'administrateur de l'espace ou le rôle d'administrateur du projet.

Pour activer les ressources de la machine

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre projet, puis sélectionnez Paramètres du projet. Choisissez Ressources de la machine.
3. Choisissez l'une des options suivantes.
  - Pour une activation individuelle, choisissez le sélecteur situé à côté d'une ou de plusieurs ressources de machine que vous souhaitez activer. Choisissez Activer, puis choisissez Cette ressource.
  - Pour activer toutes les ressources, choisissez Activer, puis Toutes les ressources.
  - Pour activer toutes les actions du flux de travail, choisissez Activer, puis sélectionnez Toutes les actions du flux de travail.
  - Pour activer tous les plans, choisissez Activer, puis Tous les plans.

## Quotas pour les projets dans CodeCatalyst

Le tableau suivant décrit les quotas et les limites pour les projets sur Amazon CodeCatalyst. Pour plus d'informations sur les quotas sur Amazon CodeCatalyst, consultez [Quotas pour CodeCatalyst](#).

Nombre maximum de projets par espace	100
Nombre maximum de projets auxquels un utilisateur peut appartenir	1 000
Nombre maximum de membres pouvant appartenir à un projet.	10 000
Noms de projet	<p>Les noms de projets doivent être uniques dans un espace. Les noms doivent comporter entre 3 et 63 caractères. Les noms sont sensibles à la casse. Les noms des projets doivent commencer par un caractère alphanumérique. Caractères valides : A-Z, a-z, 0-9, espaces et., _ (trait de soulignement) - (tiret)</p> <p>Les noms de projets ne peuvent contenir aucun des caractères suivants : ! ? @ # \$ % ^ &amp; * ( ) + = { } [ ]   \ / &gt; &lt; ~ ` ' " ; :</p>
Descriptions de projet	<p>Les descriptions de projet peuvent comporter jusqu'à 200 caractères. Caractères valides : A-Z, a-z, 0-9, espaces et., _ (trait de soulignement) - (tiret). Les descriptions de projet sont facultatives.</p>

## Utilisation des notifications dans CodeCatalyst

Vous pouvez configurer des notifications pour surveiller vos projets et vos ressources dans CodeCatalyst. Les utilisateurs peuvent choisir les événements du projet au sujet desquels ils souhaitent recevoir des e-mails dans tous les projets dont ils sont membres. Vous pouvez également choisir de configurer les notifications envoyées à toute une équipe dans une application de messagerie d'équipe, telle que Slack, en configurant l'accès entre un CodeCatalyst espace et un espace de travail Slack, puis en configurant les notifications pour un projet à envoyer à une ou plusieurs chaînes de cet espace de travail Slack. Une fois que vous avez configuré l'accès entre un CodeCatalyst espace et un espace de travail Slack, les membres du projet auront également la

possibilité d'ajouter leurs propres identifiants de membre Slack afin d'être informés directement des CodeCatalyst événements survenus dans les espaces de travail et les chaînes Slack connectés.

#### Note

L'ensemble des événements du projet qui peuvent être envoyés à Slack n'est pas le même que celui pour lequel les utilisateurs peuvent choisir d'être informés par e-mail.

## Rubriques

- [Comment fonctionnent les notifications ?](#)
- [Commencer à utiliser les notifications Slack](#)
- [Gestion des notifications sur Amazon CodeCatalyst](#)

## Comment fonctionnent les notifications ?

Vous pouvez configurer votre projet pour envoyer des notifications à l'application de messagerie de votre équipe, telle que Slack.

## Quelles sont les autorisations nécessaires pour les notifications ?

Tout membre du projet peut configurer, afficher, mettre à jour ou supprimer les paramètres de notification d'une chaîne dans CodeCatalyst. Toutefois, seuls les utilisateurs dotés du rôle d'administrateur d'espace peuvent ajouter ou supprimer des espaces de travail Slack. Tous les utilisateurs peuvent configurer les événements du projet au sujet desquels ils souhaitent recevoir des e-mails pour les projets auxquels ils appartiennent CodeCatalyst.

## À propos de quels CodeCatalyst événements puis-je configurer les notifications ?

Vous pouvez configurer CodeCatalyst pour envoyer des notifications à une ou plusieurs chaînes Slack concernant les événements du flux de travail. Une fois les notifications configurées entre un CodeCatalyst projet et Slack, les utilisateurs du projet peuvent choisir d'ajouter leur propre identifiant de membre Slack afin de recevoir des messages directs sur les chaînes Slack concernant les événements. CodeCatalyst Les utilisateurs qui ajoutent leur identifiant de membre Slack recevront des mentions directes sur leur identifiant dans les chaînes Slack configurées pour leurs projets, afin de mieux faire connaître les événements qui leur tiennent à cœur.



Vous pouvez également choisir les événements au sujet desquels vous souhaitez recevoir des e-mails. Ces e-mails sont envoyés à l'adresse e-mail configurée pour votre AWS Builder ID.

## Comment apparaissent les notifications ?

Vous pouvez configurer CodeCatalyst pour envoyer des notifications à une ou plusieurs chaînes Slack. Vous devez être autorisé CodeCatalyst pour autoriser l'accès à votre espace de travail Slack. Une fois l'autorisation fournie, CodeCatalyst vous pouvez envoyer des notifications aux chaînes Slack que vous configurez. Si un membre du projet choisit d'ajouter son identifiant de membre Slack, il peut recevoir des mentions d'événements dans les chaînes Slack configurées pour ce projet.

## Comment configurer les notifications ?

Les notifications par e-mail sont configurées dans le cadre de CodeCatalyst. Les utilisateurs du projet peuvent choisir les événements au sujet desquels ils souhaitent recevoir des e-mails sur leur page Mes paramètres.

Pour configurer les notifications Slack pour les ressources de votre projet, vous devez effectuer les tâches de haut niveau suivantes.

Pour configurer les notifications (tâches de haut niveau)

1. Dans CodeCatalyst, vous configurez une connexion entre CodeCatalyst et un client de messagerie, tel que Slack. Une fois qu'un espace de travail Slack est connecté, il sera accessible à tous les projets de l'espace.

### Note

Seuls les utilisateurs ayant un rôle d'administrateur d'espace peuvent ajouter ou supprimer un espace de travail Slack.

2. Dans votre projet CodeCatalyst, ajoutez le canal sur lequel vous souhaitez que votre équipe reçoive des notifications.
3. Dans CodeCatalyst, vous activez les notifications pour divers événements, tels que l'échec de l'exécution du flux de travail, et vous spécifiez le canal sur lequel vous souhaitez qu'elles soient envoyées.

Pour obtenir des instructions complètes, consultez [Commencer à utiliser les notifications Slack](#).

Une fois que les notifications ont été configurées entre un CodeCatalyst espace et Slack, les utilisateurs peuvent choisir d'ajouter leurs propres identifiants de membre Slack pour recevoir des messages directs concernant les CodeCatalyst événements survenus dans les chaînes Slack configurées pour leurs projets,

## Commencer à utiliser les notifications Slack

Après avoir créé un projet, vous pouvez configurer des notifications Slack qui aident votre équipe à surveiller les ressources du projet.

Ces étapes vous guident dans la configuration des notifications Slack pour la première fois depuis CodeCatalyst. Si vous avez déjà configuré les notifications, consultez [Gestion des notifications sur Amazon CodeCatalyst](#).

### Note

L'ensemble des événements du projet qui peuvent être envoyés aux canaux de notification n'est pas le même que celui pour lequel les utilisateurs peuvent choisir d'être informés par e-mail. Pour en savoir plus, consultez [Gestion des notifications sur Amazon CodeCatalyst](#).

## Rubriques

- [Prérequis](#)
- [Étape 1 : Connectez-vous CodeCatalyst à votre espace de travail Slack](#)
- [Étape 2 : ajoutez votre chaîne Slack à CodeCatalyst](#)
- [Étape 3 : Tester les notifications depuis CodeCatalyst Slack](#)
- [Étape 4 : étapes suivantes](#)

## Prérequis

Avant de commencer, vous avez besoin de ce qui suit :

- Un CodeCatalyst espace. Pour plus d'informations sur la création d'un CodeCatalyst espace et la première connexion, consultez [Configuration CodeCatalyst](#).
- Un CodeCatalyst projet. Pour en savoir plus, consultez [Création d'un projet sur Amazon CodeCatalyst](#).

- Un CodeCatalyst compte doté du rôle d'administrateur de projet ou d'administrateur de l'espace. Pour en savoir plus, consultez [Utilisation des rôles sur Amazon CodeCatalyst](#).
- Un compte Slack et un espace de travail Slack accessibles par CodeCatalyst
- Une chaîne Slack où CodeCatalyst vous enverrez des notifications. La chaîne peut être publique ou privée.

## Étape 1 : Connectez-vous CodeCatalyst à votre espace de travail Slack

Seuls les utilisateurs dotés du rôle d'administrateur de l'espace peuvent ajouter ou supprimer des espaces de travail Slack. L'ajout ou la suppression d'un espace de travail Slack affecte tous les projets de cet espace. Pour établir la connexion entre Slack CodeCatalyst et Slack, effectuez une connexion CodeCatalyst d'authentification OAuth sécurisée avec votre espace de travail Slack.

Suivez les instructions ci-dessous pour vous connecter CodeCatalyst à votre espace de travail Slack.

### Note

Cela ne doit être effectué qu'une seule fois pour chaque espace de travail Slack. Vous pouvez ensuite configurer les notifications par canal Slack.

Pour vous connecter CodeCatalyst à votre espace de travail Slack

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre projet.
3. Dans le volet de navigation, sélectionnez Paramètres du projet.
4. Choisissez l'onglet Notifications.
5. Choisissez Configurer les notifications.
6. Choisissez Se connecter à l'espace de travail Slack.
7. Lisez le contenu de la boîte de dialogue, puis choisissez Se connecter à l'espace de travail Slack.
8. Sur le message du AWSChatbot :
  - a. Dans le coin supérieur droit, choisissez l'espace de travail Slack qui contient votre chaîne.
  - b. Sélectionnez Allow (Autoriser).

Vous revenez à la CodeCatalyst console.

9. Passez au [Étape 2 : ajoutez votre chaîne Slack à CodeCatalyst](#).

## Étape 2 : ajoutez votre chaîne Slack à CodeCatalyst

Vous avez besoin de l'identifiant de chaîne Slack pour y ajouter votre chaîne. CodeCatalyst

Pour obtenir l'identifiant de votre chaîne Slack

1. Connectez-vous à Slack. Pour plus d'informations, voir [Se connecter à Slack](#).
2. Accédez à l'espace de travail Slack qui contient la chaîne sur laquelle vous souhaitez envoyer les notifications. Pour plus d'informations, consultez [Basculer entre les espaces de travail Slack ou Se connecter à d'autres espaces de travail Slack](#).
3. Dans le volet de navigation, ouvrez le menu contextuel (clic droit) de la chaîne vers laquelle vous souhaitez envoyer les notifications, puis choisissez Ouvrir les détails de la chaîne.

L'ID du canal est affiché en bas de la boîte de dialogue.

4. Copiez la valeur de l'ID de canal. Vous en aurez besoin à l'étape suivante.

À l'aide de l'identifiant de chaîne que vous venez de copier, vous pouvez désormais connecter votre chaîne Slack à CodeCatalyst.

Pour ajouter votre chaîne Slack à CodeCatalyst

1. Avant de commencer, si votre chaîne Slack est privée, ajoutez l'application AWS Chatbot à la chaîne comme suit :
  - a. Dans la boîte de message de votre chaîne Slack, entrez **@aws** et choisissez l'application aws dans la boîte de dialogue.
  - b. Appuyez sur Entrée.

Un message Slackbot apparaît, indiquant que le AWS Chatbot n'est pas dans le canal privé.

- c. Choisissez Invite Them pour inviter AWS Chatbot à rejoindre la chaîne.
2. Dans la CodeCatalyst console, choisissez Next.
  3. Dans ID de chaîne, collez l'identifiant de chaîne Slack que vous avez obtenu précédemment.

4. Dans Nom de la chaîne, entrez un nom. Nous vous recommandons d'utiliser le nom de la chaîne Slack.
5. Choisissez Suivant.
6. Dans Sélectionner les événements de notification, choisissez le type d'événement pour lequel vous souhaitez recevoir des notifications.
7. Choisissez Finish (Terminer).

### Étape 3 : Tester les notifications depuis CodeCatalyst Slack

Une fois que votre projet est configuré pour envoyer des notifications concernant l'état du flux de travail, vous pouvez consulter vos notifications dans Slack.

Pour consulter vos notifications dans Slack

1. Dans votre CodeCatalyst projet, [lancez un flux de travail manuellement](#) afin de terminer une exécution de flux de travail et de recevoir une notification d'état à la fin de l'exécution.
2. Dans Slack, consultez la chaîne que vous avez configurée pour les notifications. Vos notifications indiquent l'état le plus récent de chaque exécution du flux de travail et indiquent s'il a échoué ou réussi.

### Étape 4 : étapes suivantes

Une fois qu'un espace de travail Slack est configuré pour votre CodeCatalyst espace, vous pouvez ajouter d'autres chaînes Slack à des CodeCatalyst projets existants, et les ajouter à de nouveaux projets une fois que vous les avez créés. Vous pouvez également indiquer aux utilisateurs du projet qu'ils peuvent configurer des notifications Slack personnelles pour leurs identifiants de membre Slack et configurer les événements pour lesquels ils recevront des e-mails. Pour plus d'informations, consultez [Gestion des notifications sur Amazon CodeCatalyst](#).

## Gestion des notifications sur Amazon CodeCatalyst

Vous pouvez configurer CodeCatalyst pour envoyer des notifications concernant les événements de votre projet. Vous pouvez envoyer des notifications à des clients de messagerie tels que les chaînes Slack. Les utilisateurs du projet peuvent choisir les événements du projet dont ils seront informés par des e-mails envoyés à l'adresse e-mail configurée pour leur profil.

**Note**

L'ensemble des événements du projet qui peuvent être envoyés aux canaux de notification n'est pas le même que celui pour lequel les utilisateurs peuvent choisir d'être informés par e-mail.

**Rubriques**

- [Gérer les notifications qui vous sont envoyées directement](#)
- [Gestion des notifications envoyées aux chaînes](#)

**Gérer les notifications qui vous sont envoyées directement**

Vous pouvez choisir de recevoir des notifications par e-mail concernant les événements relatifs à tout projet dont vous êtes membre. Ces e-mails seront envoyés à l'adresse e-mail configurée dans votre identifiant AWS Builder. Par défaut, vous recevrez des e-mails concernant tous les événements du projet pour lesquels des e-mails peuvent être envoyés.

Si un projet a été configuré pour envoyer des notifications à une chaîne Slack, vous pouvez ajouter votre identifiant de membre Slack pour recevoir des mentions directes sur les CodeCatalyst événements de cette chaîne Slack. Cela peut vous aider à mieux connaître les événements qui se déroulent dans le cadre des projets dans lesquels vous avez un rôle à jouer.

Pour configurer les notifications par e-mail pour les événements du projet

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la barre de menu supérieure, choisissez votre badge de profil, puis sélectionnez Mes paramètres. La page CodeCatalyst Mes paramètres s'ouvre.

**Tip**

Vous pouvez également accéder à votre profil utilisateur en vous rendant sur la page des membres d'un projet ou d'un espace et en choisissant votre nom dans la liste des membres.

3. Dans Notifications par e-mail, recherchez le projet dans la liste pour lequel vous souhaitez configurer les notifications par e-mail, puis choisissez Modifier.

4. Sélectionnez les événements pour lesquels vous souhaitez recevoir des e-mails, puis choisissez Enregistrer.


Pour configurer les notifications personnelles de Slack

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la barre de menu supérieure, choisissez votre badge de profil, puis sélectionnez Mes paramètres. La page CodeCatalyst Mes paramètres s'ouvre.

 Tip

Vous pouvez également accéder à votre profil utilisateur en vous rendant sur la page des membres d'un projet ou d'un espace et en choisissant votre nom dans la liste des membres.

3. Dans les notifications personnelles de Slack, choisissez Connect Slack ID, puis sélectionnez Connect to Slack workspace. Une fenêtre séparée s'ouvrira.

 Tip

Cette option n'est pas configurable sauf si un utilisateur ayant le rôle d'administrateur de l'espace a ajouté un espace de travail Slack à votre CodeCatalyst espace. Pour plus d'informations, veuillez consulter [Commencer à utiliser les notifications Slack](#) et [Gestion des notifications envoyées aux chaînes](#).

4. Dans la fenêtre de demande d'autorisations, assurez-vous que le nom de l'espace de travail correspond à celui de l'espace de travail Slack configuré pour CodeCatalyst cet espace. Choisissez Autoriser pour autoriser AWS Chatbot l'accès à l'espace de travail. La fenêtre se fermera et l'espace de travail Slack indiquera que l'état de la connexion est Connecté.

 Tip

Si l'état de connexion ne change pas, vérifiez si une erreur s'est produite lors de la connexion à l'espace de travail Slack. Vous devrez peut-être faire défiler l'écran vers le haut pour voir le message d'erreur.

5. Pour ne plus recevoir de notifications Slack personnelles, choisissez l'espace de travail Slack connecté, puis choisissez Déconnecter l'identifiant Slack.

## Gestion des notifications envoyées aux chaînes

Vous pouvez choisir d'ajouter et de gérer les notifications relatives aux événements du projet CodeCatalyst envoyées aux ressources de l'équipe, telles qu'un canal Slack d'équipe. Vous pouvez ainsi vous assurer que l'ensemble de votre équipe est au courant des événements importants, tels que l'échec d'un flux de travail.

### Note

Tout membre d'un projet peut gérer les notifications envoyées aux chaînes associées à ce projet. Toutefois, seuls les utilisateurs dotés du rôle d'administrateur de l'espace peuvent ajouter ou supprimer des espaces de travail Slack.

### Ajouter un canal de notification pour un projet

Vous pouvez ajouter une chaîne sur laquelle vous souhaitez recevoir des notifications, telle que la chaîne Slack de votre équipe.

Pour ajouter une chaîne Slack pour les notifications

1. Si vous ajoutez votre première chaîne Slack, consultez plutôt [Commencer à utiliser les notifications Slack](#).

Après avoir configuré votre première chaîne, revenez à cette procédure pour configurer des chaînes supplémentaires.

2. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
3. Accédez à votre projet.
4. Dans le volet de navigation, sélectionnez Paramètres du projet.
5. Choisissez l'onglet Notifications.
6. Choisissez Add channel (Ajouter un canal).
7. Choisissez Choisir un espace de travail, puis sélectionnez l'espace de travail Slack contenant la chaîne sur laquelle vous souhaitez envoyer des notifications.

Si votre espace de travail Slack ne figure pas dans la liste, vous pouvez l'ajouter en suivant les instructions figurant dans [Commencer à utiliser les notifications Slack](#).

8. Avant de saisir un identifiant de chaîne, si la chaîne Slack que vous souhaitez ajouter est privée, procédez comme suit :



- a. Dans la boîte de message de votre chaîne Slack, entrez **@aws** et choisissez l'application aws dans la fenêtre contextuelle.
  - b. Appuyez sur Entrée.  
  
Un message Slackbot apparaît, indiquant que le AWS Chatbot n'est pas dans le canal privé.
  - c. Choisissez Invite Them pour inviter AWS Chatbot à rejoindre la chaîne.
9. Dans CodeCatalyst le champ ID de chaîne, saisissez l'identifiant de chaîne Slack. Pour trouver l'identifiant, accédez à Slack, puis dans le volet de navigation, cliquez avec le bouton droit sur la chaîne et choisissez Ouvrir les détails de la chaîne.
- L'ID du canal est affiché en bas de la boîte de dialogue.
10. Dans Nom de la chaîne, entrez un nom. Nous vous recommandons d'utiliser le nom de la chaîne Slack.
  11. Dans Sélectionner les événements de notification, choisissez le type d'événement pour lequel vous souhaitez recevoir des notifications.
  12. Choisissez Ajouter.

### Modification des notifications pour un canal de notification

Vous pouvez modifier les canaux auxquels les notifications sont destinées, et vous pouvez désactiver complètement certaines notifications.

#### Pour modifier les notifications

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre projet.
3. Dans le volet de navigation, sélectionnez Paramètres du projet.
4. Choisissez l'onglet Notifications.
5. Choisissez Modifier les notifications.
6. Effectuez l'une des actions suivantes :
  - Pour envoyer une notification à un canal spécifique, choisissez-le dans la liste déroulante.
  - Pour désactiver une notification de manière globale, cliquez sur le bouton situé à côté de la notification.
  - Pour arrêter d'envoyer une notification à une chaîne spécifique, choisissez le X sur la chaîne.

## 7. Choisissez Save (Enregistrer).

### Supprimer une chaîne

#### Pour supprimer une chaîne

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre projet. Dans le volet de navigation, sélectionnez Paramètres du projet.
3. Sur la page des paramètres du projet, choisissez l'onglet Notifications.
4. Choisissez l'indicateur situé à côté de la chaîne que vous souhaitez supprimer, puis choisissez Supprimer la chaîne. Lorsque vous y êtes invité, choisissez OK dans la fenêtre de confirmation.

# Des plans dans CodeCatalyst

Les plans sont des générateurs de code arbitraires qui représentent un composant architectural d'un CodeCatalyst projet. Le composant peut consister en n'importe quoi, qu'il s'agisse d'un flux de travail dans un seul fichier ou de l'ensemble du projet avec un exemple de code. Les plans prennent un ensemble arbitraire d'options et les utilisent pour générer un ensemble arbitraire de code de sortie qui est transféré dans un projet. Au fur et à mesure que le plan est mis à jour avec les dernières meilleures pratiques ou de nouvelles options, il peut régénérer les parties pertinentes de votre base de code dans les projets contenant ce plan.

Vous pouvez utiliser un CodeCatalyst plan Amazon pour créer un projet complet avec un référentiel de sources, des exemples de code source, des flux de travail CI/CD, des rapports de génération et de test, ainsi que des outils de suivi des problèmes intégrés. Un CodeCatalyst plan génère des ressources et du code source en fonction des paramètres de configuration définis. Lorsque vous utilisez un plan CodeCatalyst géré, le plan que vous choisissez détermine les ressources ajoutées à votre projet, ainsi que les outils que CodeCatalyst crée ou configure, afin que vous puissiez suivre et utiliser les ressources de votre projet. En tant qu'utilisateur d'un plan, vous pouvez créer un projet à l'aide d'un plan ou l'appliquer à un projet existant CodeCatalyst . Vous pouvez appliquer plusieurs plans à votre projet, et chacun peut être appliqué en tant que composant indépendant. Par exemple, vous pouvez avoir un projet créé à l'aide d'un plan d'application Web, puis appliquer un plan de sécurité ultérieurement. Lorsque l'un des plans est mis à jour, vous pouvez intégrer les modifications ou les correctifs dans votre projet grâce à la gestion du cycle de vie. Pour plus d'informations, consultez [Référence du plan de projet](#) et [Travailler avec la gestion du cycle de vie en tant qu'utilisateur du plan](#).

En tant qu'auteur de plans, vous pouvez également créer et publier des plans personnalisés pour que les membres de votre CodeCatalyst espace puissent utiliser les ressources de votre projet. Les plans personnalisés peuvent être développés pour répondre aux besoins spécifiques des projets de votre espace. Après avoir ajouté un plan personnalisé au catalogue de plans de votre espace, vous pouvez gérer le plan et continuer à le mettre à jour afin que les projets de votre espace restent à jour conformément aux meilleures pratiques les plus récentes. Pour plus d'informations, consultez [Travailler avec des plans personnalisés dans CodeCatalyst](#). [Pour consulter le SDK de plans et des exemples de plans, consultez le référentiel open source. GitHub](#)

## Rubriques

- [Création d'un projet à l'aide d'un plan](#)
- [Appliquer et dissocier des plans dans un projet](#)

- [Mettre à jour un plan dans un projet](#)
- [Modification de la description d'un plan dans un projet](#)
- [Travailler avec la gestion du cycle de vie en tant qu'utilisateur du plan](#)
- [Référence du plan de projet](#)
- [Travailler avec des plans personnalisés dans CodeCatalyst](#)
- [Quotas pour les plans dans CodeCatalyst](#)

## Création d'un projet à l'aide d'un plan

Vous pouvez créer rapidement un projet à l'aide d'un plan issu du CodeCatalyst catalogue Amazon ou du catalogue spatial de votre équipe avec des plans personnalisés. En fonction du plan, votre projet est créé avec des ressources spécifiques. Pour plus d'informations, consultez [Création d'un projet à l'aide d'un plan](#) et [Référence du plan de projet](#).

Après avoir créé un projet, vous pouvez appliquer des plans supplémentaires à votre CodeCatalyst projet à partir du CodeCatalyst catalogue ou du catalogue de votre espace avec des plans personnalisés. Les plans représentent des composants architecturaux, de sorte que plusieurs plans peuvent être utilisés ensemble dans votre projet afin d'intégrer les meilleures pratiques de votre équipe. Cela vous permet également de vous assurer que votre projet est à jour avec les dernières modifications apportées aux composants en évolution. Pour en savoir plus sur l'utilisation des plans dans votre projet, consultez [Travailler avec la gestion du cycle de vie en tant qu'utilisateur du plan](#).

## Appliquer et dissocier des plans dans un projet

Vous pouvez appliquer plusieurs plans dans un projet afin d'intégrer des composants fonctionnels, des ressources et une gouvernance. Vos projets peuvent prendre en charge différents éléments gérés indépendamment dans des plans distincts. L'application de plans à un projet réduit le besoin de créer manuellement des ressources et de rendre les composants logiciels fonctionnels. Vos projets peuvent également rester à jour au fur et à mesure de l'évolution des exigences. Si votre projet ne nécessite plus les ressources d'un plan, vous pouvez dissocier le plan de votre projet. Pour en savoir plus sur l'application de plans dans votre projet, consultez [Travailler avec la gestion du cycle de vie en tant qu'utilisateur du plan](#).

### Rubriques

- [Appliquer un plan à un projet](#)

- [Dissociation d'un plan d'un projet](#)

## Appliquer un plan à un projet

Pour appliquer un plan à votre projet

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la CodeCatalyst console, accédez à l'espace, puis choisissez le projet dans lequel vous souhaitez appliquer un plan.
3. Dans le volet de navigation, choisissez Blueprints, puis choisissez Appliquer le Blueprint.
4. Choisissez un plan dans l'onglet des CodeCatalyst plans ou un plan personnalisé dans l'onglet des plans spatiaux, puis choisissez Suivant.
5. Sous Détails du plan, choisissez une version du plan dans le menu déroulant Version cible. La dernière version est automatiquement sélectionnée.
6. Sous Configurer le plan, configurez les paramètres du plan.
7. Passez en revue les différences entre la version actuelle du plan directeur et votre version mise à jour. La différence affichée dans une pull request indique les modifications entre la version actuelle et la dernière version, qui est la version souhaitée au moment de la création de la pull request. Si aucune modification ne s'affiche, les versions sont peut-être identiques ou vous avez peut-être choisi la même version pour la version actuelle et pour la version souhaitée.
8. Lorsque vous êtes certain que la pull request contient le code et les modifications que vous souhaitez vérifier, choisissez Appliquer le plan. Après avoir créé la pull request, vous pouvez ajouter des commentaires. Des commentaires peuvent être ajoutés à la pull request ou à des lignes individuelles des fichiers, ainsi qu'à la pull request globale. Vous pouvez ajouter des liens vers des ressources telles que des fichiers en utilisant le @ signe suivi du nom du fichier.

### Note

Le plan ne sera pas appliqué tant que la pull request ne sera pas approuvée et fusionnée. Pour plus d'informations, consultez [Révision d'une pull request](#) et [Fusion d'une pull request](#).

Les auteurs de plans peuvent également appliquer un plan personnalisé à des projets situés dans des espaces spécifiques où le plan n'est pas disponible pour créer de nouveaux projets ou l'appliquer

à des projets existants. Pour plus d'informations, consultez [Publication et application d'un plan personnalisé dans des espaces et des projets spécifiques](#).

## Dissociation d'un plan d'un projet

Si vous ne souhaitez pas de nouvelles mises à jour depuis un plan, vous pouvez dissocier le plan de votre projet. Les ressources et les composants logiciels fonctionnels ajoutés à votre projet à partir du plan seront au cœur de votre projet.

Pour dissocier un plan de votre projet

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la CodeCatalyst console, accédez à l'espace, puis choisissez le projet dans lequel vous souhaitez dissocier un plan.
3. Dans le panneau de navigation, sélectionnez Blueprints (Modèles).
4. Choisissez le plan contenant les ressources que vous souhaitez dissocier, choisissez le menu déroulant Actions, puis choisissez Dissocier le plan directeur.
5. Entrez `confirm` pour confirmer la dissociation.
6. Choisissez Confirmer.


## Mettre à jour un plan dans un projet

Si vous avez créé un projet avec un plan ou si vous avez appliqué un plan à un projet existant, vous êtes informé des nouvelles versions du plan. Avant que la version du plan ne soit mise à jour par le biais d'une pull request approuvée, vous pouvez consulter les modifications du code et les environnements concernés. La gestion du cycle de vie vous permet de mettre à jour un ou plusieurs plans appliqués à votre projet, afin que chaque plan puisse être mis à jour sans affecter les autres domaines de votre projet. Vous pouvez également annuler les mises à jour du plan. Pour plus d'informations, consultez [Travailler avec la gestion du cycle de vie en tant qu'utilisateur du plan](#).


Pour mettre à jour un plan avec la dernière version

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la CodeCatalyst console, accédez à l'espace dans lequel vous souhaitez mettre à jour la version d'un plan.
3. Sur le tableau de bord de l'espace, choisissez le projet dont vous souhaitez mettre à jour le plan.

4. Dans le volet de navigation, choisissez Blueprints, puis cliquez sur le bouton radio correspondant au plan que vous souhaitez mettre à jour.
5. Choisissez le menu déroulant Actions, puis sélectionnez Mettre à jour la version.
6. Dans le menu déroulant Version cible, choisissez la version que vous souhaitez mettre à jour. La dernière version est automatiquement sélectionnée.
7. Sous Configurer le plan, configurez les paramètres du plan.
8. Passez en revue les différences entre la version actuelle du plan directeur et la version mise à jour. La différence affichée dans une pull request réside dans les modifications entre la version actuelle et la dernière version, qui est la version souhaitée au moment de la création de la pull request. Si aucune modification ne s'affiche, les versions sont peut-être identiques ou vous avez peut-être choisi la même version pour la version actuelle et pour la version souhaitée.
9. Lorsque vous êtes certain que la pull request contient le code et les modifications que vous souhaitez vérifier, choisissez Appliquer la mise à jour. Après avoir créé la pull request, vous pouvez ajouter des commentaires. Des commentaires peuvent être ajoutés à la pull request ou à des lignes individuelles des fichiers et à la pull request globale. Vous pouvez ajouter des liens vers des ressources telles que des fichiers en utilisant le @ signe suivi du nom du fichier.

 Note

Le plan ne sera pas mis à jour tant que la pull request ne sera pas approuvée et fusionnée. Pour plus d'informations, consultez [Révision d'une pull request](#) et [Fusion d'une pull request](#).

 Note

Si des pull requests sont déjà ouvertes pour mettre à jour une version du plan, fermez les pull requests précédentes avant d'en créer une nouvelle. Lorsque vous choisissez Mettre à jour la version, vous êtes redirigé vers la liste des pull requests en attente pour le plan. Vous pouvez également consulter les pull requests en attente depuis l'onglet Blueprints des paramètres du projet et de la page de résumé du projet. Pour plus d'informations, voir [Afficher les pull requests](#).

## Modification de la description d'un plan dans un projet

Vous pouvez modifier la description d'un plan que vous avez utilisé pour créer un projet ou que vous avez appliqué après la création d'un projet. Un plan peut être utilisé plusieurs fois dans un projet. Pour différencier l'objectif des plans dans votre projet, vous pouvez utiliser des descriptions pour ces plans. Les descriptions peuvent également être utilisées pour identifier les composants que vous appliquez à partir d'un plan spécifique.

Pour modifier la description d'un plan dans votre projet

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la CodeCatalyst console, accédez à votre espace, puis choisissez le projet avec les paramètres du plan que vous souhaitez mettre à jour.
3. Dans le panneau de navigation, sélectionnez Blueprints (Modèles).
4. Choisissez le plan contenant la description que vous souhaitez mettre à jour, choisissez le menu déroulant Actions, puis sélectionnez Modifier la description.
5. Dans le champ de saisie du texte de description du plan, entrez une description pour identifier le plan de votre projet.
6. Choisissez Enregistrer.

## Travailler avec la gestion du cycle de vie en tant qu'utilisateur du plan

La gestion du cycle de vie est la capacité de régénérer une base de code à partir d'options ou de versions mises à jour d'un plan. Cela permet à l'auteur d'un plan de gérer de manière centralisée le cycle de développement logiciel de chaque projet contenant un plan spécifique. Par exemple, l'ajout d'un correctif de sécurité à un plan d'application Web permettra à chaque projet contenant ou créé à partir du plan d'application Web de récupérer automatiquement ce correctif. Ce même cadre de gestion vous permet également, en tant qu'utilisateur du plan, de modifier les options du plan une fois celles-ci sélectionnées.

### Rubriques

- [Utilisation de la gestion du cycle de vie sur les projets existants](#)
- [Utilisation de la gestion du cycle de vie sur plusieurs plans d'un projet](#)
- [Gérer les conflits liés au cycle de vie des pull requests](#)



- [Se désinscrire des modifications apportées à la gestion du cycle de vie](#)
- [Remplacer la gestion du cycle de vie d'un plan dans un projet](#)

## Utilisation de la gestion du cycle de vie sur les projets existants

Vous pouvez utiliser la gestion du cycle de vie pour les projets créés à partir de plans ou pour des projets existants qui ne sont associés à aucun plan. Par exemple, vous pouvez ajouter un plan de pratiques de sécurité standard dans une application five-year-old Java qui n'a jamais été créée à partir d'un plan. Le plan génère un flux de travail d'analyse de sécurité et d'autres codes associés. Cette partie de la base de code de l'application Java sera désormais automatiquement mise à jour conformément aux meilleures pratiques de votre équipe chaque fois que des modifications sont apportées au plan.

## Utilisation de la gestion du cycle de vie sur plusieurs plans d'un projet

Comme les plans représentent des composants architecturaux, plusieurs plans peuvent souvent être utilisés ensemble dans le même projet. Par exemple, un projet peut être composé d'un plan d'API Web central élaboré par un ingénieur de plate-forme de l'entreprise, ainsi que d'un plan de vérification des versions élaboré par l'équipe chargée de la sécurité des applications. Chacun de ces plans peut être mis à jour indépendamment et mémoriser les résolutions de fusion qui lui ont été appliquées dans le passé.

### Note

En tant que composants architecturaux arbitraires, tous les plans n'ont pas de sens ensemble ou ne fonctionneront pas logiquement ensemble, même s'ils essaieront tout de même de se fondre les uns dans les autres.

## Gérer les conflits liés au cycle de vie des pull requests

Parfois, les pull requests relatives au cycle de vie peuvent générer des conflits de fusion. Ces problèmes peuvent être résolus manuellement. Les résolutions sont mémorisées lors des mises à jour ultérieures du plan.

## Se désinscrire des modifications apportées à la gestion du cycle de vie

Les utilisateurs peuvent supprimer un plan d'un projet pour dissocier toutes les références au plan et refuser les mises à jour du cycle de vie. Pour des raisons de sécurité, cela ne supprime ni n'affecte le code ou les ressources du projet, y compris ce qui a été ajouté à partir du plan. Pour plus d'informations, consultez [Dissociation d'un plan d'un projet](#).

## Remplacer la gestion du cycle de vie d'un plan dans un projet

Si vous souhaitez annuler les mises à jour d'un plan pour des fichiers spécifiques de votre projet, vous pouvez inclure un fichier de propriété dans votre référentiel. [GitLabLa spécification Code Owners](#) est la directive recommandée. Le plan respecte toujours le fichier du propriétaire du code par rapport à tout le reste et peut en générer un exemple comme celui-ci :

```
new BlueprintOwnershipFile(sourceRepo, {
  resynthesis: {
    strategies: [
      {
        identifiant: 'dont-override-sample-code',
        description: 'This strategy is applied accross all sample code. The
blueprint will create sample code, but skip attempting to update it.',
        strategy: MergeStrategies.neverUpdate,
        globs: [
          '**/src/**',
          '**/css/**',
        ],
      },
    ],
  },
});
```

Cela génère un `.ownership-file` avec le contenu suivant :

```
[dont-override-sample-code] @amazon-codecatalyst/blueprints.import-from-git
# This strategy is applied accross all sample code. The blueprint will create sample
code, but skip attempting to update it.
# Internal merge strategy: neverUpdate
**/src/**
**/css/**
```

## Référence du plan de projet

Lorsque vous créez un projet à l'aide d'un plan, vous CodeCatalyst créez un projet complet avec un référentiel source, des exemples de code source, des flux de travail CI/CD, des rapports de construction et de test, ainsi que des outils de suivi des problèmes intégrés. Un plan de projet utilise du code pour fournir une infrastructure cloud, des ressources et des exemples d'artefacts de source pour différents types d'applications et de frameworks.

Pour plus d'informations, consultez [Création d'un projet sur Amazon CodeCatalyst](#). Vous devez être l'administrateur de l'espace pour créer un projet.

### Rubriques

- [Plans disponibles](#)
- [Trouver des informations sur le plan de projet](#)

## Plans disponibles

Nom du plan	Description du plan
API Web ASP.NET Core	Ce plan crée une application d'API Web .NET 6 ASP.NET Core. Le plan utilise l'outil de AWS déploiement pour .NET et fournit une option permettant de configurer Amazon Elastic Container Service ou AWS Elastic Beanstalk en tant que cible de déploiement. AWS App Runner
AWS Glue ETL	Ce plan crée un exemple d'implémentation de référence ETL (Extract Transform Load) à l'aide de AWS CDK, AWS Glue, AWS Lambda et Amazon Athena pour convertir des valeurs séparées par des virgules (CSV) en Apache Parquet.
DevOps pipeline de déploiement	Ce plan crée un pipeline de déploiement à l'aide de l'architecture de référence du pipeline

Nom du plan	Description du plan
	de AWS déploiement qui déploie une application de référence AWS sur plusieurs étapes.
API Java avec AWS Fargate	Ce plan crée un projet de service Web conteneurisé. Le projet utilise la <a href="#">AWS CLI Copilot</a> pour créer et déployer un service Web Java <a href="#">Spring Boot</a> conteneurisé soutenu par Amazon DynamoDB sur Amazon ECS. Le projet déploie une application conteneurisée sur un cluster Amazon ECS sur AWS Fargate un calcul sans serveur. L'application stocke les données dans une table DynamoDB. Une fois votre flux de travail exécuté avec succès, l'exemple de service Web est accessible au public via l'Application Load Balancer.
Application Web moderne à trois niveaux	Ce plan génère du code en Python pour la couche application et le framework frontal Vue afin de créer et de déployer une application Web moderne à 3 niveaux bien architecturée.
Application sans serveur .NET	Ce plan crée des AWS Lambda fonctions à l'aide des outils Lambda de la CLI .NET. Le plan fournit des options pour les AWS Lambda fonctions, notamment le choix entre C# ou F#.

Nom du plan	Description du plan
API Node.js avec AWS Fargate	Ce plan crée un projet de service Web conteneurisé. Le projet utilise la <a href="#">AWS CLI Copilot</a> pour créer et déployer un service Web <a href="#">Express/Node.js</a> conteneurisé sur Amazon Elastic Container Service. Le projet déploie une application conteneurisée sur un cluster Amazon ECS sur AWS Fargate un calcul sans serveur. Une fois votre flux de travail exécuté avec succès, l'exemple de service Web est accessible au public via l'Application Load Balancer.
Modèle d'application sans serveur (SAM)	Ce plan crée un projet qui utilise un modèle d'application sans serveur (SAM) pour créer et déployer une API. Vous pouvez choisir SDK pour Java TypeScript ou SDK pour Python comme langage de programmation.
Gestionnaire d'images sans serveur	Ce plan permet de créer une application de traitement d'image à haute vitesse sans réduire la qualité de l'image.
Microservice RESTful sans serveur	Ce plan crée une API REST qui utilise AWS Lambda et Amazon API Gateway avec une référence de service To Do. Vous pouvez choisir SDK pour Java TypeScript ou SDK pour Python comme langage de programmation.
Application d'une seule page	Ce plan crée une application monopage (SPA) qui utilise les frameworks React, Vue et Angular. Pour l'hébergement, choisissez AWS Amplify Hosting ou Amazon CloudFront Amazon S3.

Nom du plan	Description du plan
Site Web statique	<p>Ce plan crée un site Web statique à l'aide des générateurs de sites statiques <a href="#">Hugo</a> ou <a href="#">Jekyll</a>. Les générateurs de sites statiques utilisent des fichiers de saisie de texte (tels que Markdown) pour générer des pages Web statiques. Ils sont idéaux pour les contenus informatifs rarement modifiés, tels que les pages de produits, la documentation et les blogs. Le plan utilise le AWS CDK pour déployer des pages Web statiques sur Amazon S3 + AWS Amplify ou sur Amazon S3 + CloudFront.</p>
Application Web To Do	<p>Ce plan crée une application Web sans serveur To Do avec des composants frontaux et principaux. Vous pouvez choisir SDK pour Java TypeScript ou SDK pour Python comme langage de programmation.</p>
Service video-on-demand Web V	<p>Ce plan crée un video-on-demand service qui permet d'intégrer, de transcoder et de diffuser du contenu. Le plan utilise AWS Lambda Amazon S3 Amazon CloudWatch et AWS Elemental MediaConvert.</p>

Nom du plan	Description du plan
S'abonner au plan externe	<p>Ce plan crée un flux de travail pour chaque package importé. Ces flux de travail s'exécutent une fois par jour pour vérifier la présence de nouvelles versions des packages dans NPM. Si une nouvelle version existe, le flux de travail tente de l'ajouter à votre CodeCatalyst espace sous forme de plan personnalisé. L'action échouera si un package est introuvable ou s'il ne s'agit pas d'un plan. Le package cible doit être sur NPM et le package doit être un plan. L'espace doit être souscrit à un niveau qui prend en charge les plans personnalisés.</p>
Chatbot Bedrock GenAI	<p>Ce plan crée un chatbot IA génératif avec <a href="#">Amazon Bedrock</a> et Claude d'<a href="#">Anthropic</a>. Avec ce plan, vous pouvez créer et déployer votre propre terrain de jeu LLM sécurisé et protégé par connexion, qui peut être personnalisé en fonction de vos données. Pour plus d'informations, consultez la documentation du <a href="#">Chatbot Bedrock GenAI</a>.</p>
Plans du kit de développement de projet AWS (AWS PDK)	<p>Ces plans PDK peuvent être composés ensemble pour créer une application composée d'un site Web React, d'une API Smithy et de l'infrastructure CDK de support pour le déployer sur AWS. Le kit AWS PDK fournit des éléments de base pour les modèles courants ainsi que des outils de développement pour gérer et développer vos projets. Pour plus d'informations, consultez le <a href="#">référentiel GitHub source AWS PDK</a> et <a href="#">Tutoriel : Création d'une application complète avec des plans PDK composables</a>.</p>

## Trouver des informations sur le plan de projet

Plusieurs plans de projet sont disponibles dans CodeCatalyst. Chaque plan est accompagné d'un résumé et d'un fichier README. Le résumé décrit les ressources installées par le plan, tandis que le fichier README explique le plan en détail et fournit des instructions sur son utilisation.

## Travailler avec des plans personnalisés dans CodeCatalyst

Vous pouvez standardiser le développement et les meilleures pratiques pour les projets de votre CodeCatalyst espace grâce à des plans personnalisés. Les plans personnalisés peuvent être utilisés pour définir différents aspects d'un CodeCatalyst projet, tels que les définitions de flux de travail et le code d'application. Une fois qu'un plan personnalisé est utilisé pour créer un nouveau projet ou appliqué à des projets existants, toutes les modifications apportées au plan sont disponibles pour ces projets sous forme de mises à jour par pull request. En tant qu'auteur de plans, vous pouvez consulter les détails des projets qui utilisent vos plans dans l'ensemble de votre espace, afin de voir comment les normes sont appliquées dans tous les projets. La gestion du cycle de vie d'un plan vous permet de gérer de manière centralisée le cycle de développement logiciel de chaque projet, ce qui vous permet de vous assurer que les projets de votre espace continuent de suivre les meilleures pratiques avec les dernières modifications ou corrections. Pour plus d'informations, consultez [Travailler avec la gestion du cycle de vie en tant qu'auteur de plans](#).

Les plans personnalisés permettent de mettre à jour les versions du plan par rapport au projet précédent par le biais d'une resynthèse. La resynthèse est le processus qui consiste à réexécuter la synthèse du plan avec des versions mises à jour ou à intégrer des correctifs et des modifications dans des projets existants. Pour plus d'informations, consultez [Concepts de plans personnalisés](#).

[Pour consulter le SDK de plans et des exemples de plans, consultez le référentiel open source. GitHub](#)

### Rubriques

- [Concepts de plans personnalisés](#)
- [Commencer à utiliser des plans personnalisés](#)
- [Tutoriel : Création et mise à jour d'une application React](#)
- [Travailler avec la gestion du cycle de vie en tant qu'auteur de plans](#)
- [Élaboration d'un plan personnalisé](#)
- [Publication d'un plan personnalisé](#)
- [Afficher les détails, les versions et les projets d'un plan personnalisé](#)



- [Ajouter et supprimer un plan personnalisé dans un espace](#)
- [Gestion des autorisations de publication pour un plan personnalisé](#)
- [Gestion des versions pour un plan personnalisé](#)
- [Suppression d'un plan ou d'une version personnalisés publiés](#)
- [Utilisation des dépendances et de l'outillage](#)
- [Contribuer](#)

## Concepts de plans personnalisés

Voici quelques concepts et termes que vous devez connaître lorsque vous travaillez avec des plans personnalisés dans CodeCatalyst.

### Rubriques

- [Projet Blueprint](#)
- [Plans spatiaux](#)
- [Catalogue de plans spatiaux](#)
- [Synthèse](#)
- [Resynthèse](#)
- [Options partielles](#)
- [Projen](#)

### Projet Blueprint

Un projet de plan vous permet de développer et de publier des plans dans votre espace. Un référentiel source est créé au cours du processus de création du projet, et le nom du référentiel est celui que vous avez choisi lors de la saisie des détails des ressources du projet. Au cours du processus de création du plan, si vous choisissez de générer une version du flux de travail, un flux de publication est créé dans votre plan avec le plan Blueprint Builder. Le flux de travail publie automatiquement votre dernière version.

### Plans spatiaux

Vous pouvez consulter et gérer tous les plans depuis le tableau des plans d'espace lorsque vous accédez à la section Plans de votre espace. Une fois les plans publiés dans votre espace, ils sont mis à disposition sous forme de plan d'espace à ajouter et à supprimer du catalogue de plans de

votre espace. Vous pouvez également gérer les autorisations de publication et supprimer des plans dans la section Blueprints de votre espace. Pour de plus amples informations, veuillez consulter [Afficher les détails, les versions et les projets d'un plan personnalisé](#).

## Catalogue de plans spatiaux

Vous pouvez consulter tous les plans personnalisés ajoutés à partir du catalogue de plans d'un espace. C'est ici qu'un membre de l'espace peut choisir votre plan personnalisé pour créer un nouveau projet. Ce catalogue est différent du CodeCatalyst catalogue, qui contient déjà des plans disponibles pour tous les membres de l'espace. Pour de plus amples informations, veuillez consulter [Référence du plan de projet](#).

## Synthèse

La synthèse est le processus de génération d'un ensemble de CodeCatalyst projets qui représente le code source, la configuration et les ressources d'un projet. Le bundle est ensuite utilisé par les opérations de l'API de CodeCatalyst déploiement pour le déployer dans un projet. Le processus peut être exécuté localement lors du développement de votre plan personnalisé afin d'émuler la création de projet sans avoir à créer un projet dans. CodeCatalyst Les commandes suivantes peuvent être utilisées pour effectuer une synthèse :

```
yarn blueprint:synth          # fast mode
yarn blueprint:synth --cache  # wizard emulation mode
```

Le plan commence par appeler la `blueprint.ts` classe principale avec cette option `fusionnéedefaults.json`. Un nouveau bundle de projets est généré sous le `synth/synth.[options-name]/proposed-bundle/` dossier. La sortie inclut l'ensemble de projets généré par un plan personnalisé, compte tenu des options que vous avez définies, y compris les [options partielles](#) que vous avez peut-être configurées.

## Resynthèse

La resynthèse est le processus qui consiste à régénérer un plan avec différentes options de plan ou des versions de plan de projets existants. En tant qu'auteur d'un plan, vous pouvez définir des stratégies de fusion personnalisées dans le code du plan personnalisé. Vous pouvez également définir des limites de propriété dans un `.ownership-file` pour spécifier dans quelles parties de la base de code un plan peut être mis à jour. Bien que le plan personnalisé puisse proposer des mises à jour `.ownership-file`, les développeurs de projets utilisant le plan personnalisé peuvent déterminer les limites de propriété de leurs projets. Vous pouvez exécuter la resynthèse localement,

la tester et la mettre à jour avant de publier votre plan personnalisé. Utilisez les commandes suivantes pour effectuer une resynthèse :

```
yarn blueprint:resynth          # fast mode
yarn blueprint:resynth --cache  # wizard emulation mode
```

Le plan commence par appeler la `blueprint.ts` classe principale avec cette option `fusionnéedefaults.json`. Un nouveau bundle de projets est généré sous le `synth/resynth.[options-name]` dossier. La sortie inclut l'ensemble de projets généré par un plan personnalisé, compte tenu des options que vous avez définies, y compris les [options partielles](#) que vous avez peut-être configurées.

Les contenus suivants sont créés après les processus de synthèse et de resynthèse :

- `proposed-bundle` - Le résultat de la synthèse lorsqu'elle est exécutée avec de nouvelles options pour la version du plan cible.
- `existing-bundle` - Une maquette de votre projet existant. S'il n'y a rien dans ce dossier, il est généré avec le même résultat que `proposed-bundle`.
- `ancestors-bundle` - Une maquette de ce que votre plan générerait s'il était exécuté avec une version antérieure, des options antérieures ou une combinaison des deux. S'il n'y a rien dans ce dossier, il est généré avec le même résultat que `proposed-bundle`.
- `resolved-bundle` - Le bundle est toujours régénéré et utilise par défaut une fusion à trois voies entre le, et le. `proposed-bundle existing-bundle ancestor-bundle` Ce bundle fournit une émulation de ce qu'une resynthèse produirait localement.

Pour en savoir plus sur les ensembles de sortie Blueprint, consultez. [Génération de fichiers par resynthèse](#)

## Options partielles

Vous pouvez ajouter des variantes d'`src/wizard-configuration/options` sans avoir à énumérer l'intégralité de l'`Optionsinterface`, et les options sont fusionnées en haut du `defaults.json` fichier. Cela vous permet d'adapter les scénarios de test à des options particulières.

Exemple :

`Optionsinterface` :

```
{
```

```
language: "Python" | "Java" | "Typescript",
repositoryName: string
...
}
```

Fichier `defaults.json` :

```
{
  language: "Python",
  repositoryName: "Myrepo"
  ...
}
```

Tests de configuration supplémentaires :

- ```
#wizard-config-typescript-test.json
{
  language: "Typescript",
}
```

- ```
#wizard-config-java-test.json
{
  language: "Java",
}
```

## Projen

Projen est un outil open source que les plans personnalisés utilisent pour rester à jour et cohérents. Les plans sont fournis sous forme de packages Projen, car ce framework vous permet de créer, de regrouper et de publier des projets, et vous pouvez utiliser l'interface pour gérer les configurations et les paramètres d'un projet.

Vous pouvez utiliser Projen pour mettre à jour les plans à grande échelle, même après leur création. L'outil Projen est la technologie sous-jacente à la synthèse du plan qui génère un ensemble de projets. Projen est propriétaire de la configuration d'un projet, et cela ne devrait pas avoir d'impact sur vous en tant qu'auteur du plan. Vous pouvez exécuter `yarn projen` pour régénérer la configuration de votre projet après avoir ajouté des dépendances, ou vous pouvez modifier les options du `projenrc.ts` fichier. Projen est également l'outil de génération sous-jacent pour les plans personnalisés afin de synthétiser un projet. Pour plus d'informations, consultez la [GitHub page](#)

[du projet](#). Pour en savoir plus sur l'utilisation de Projen, consultez la [documentation de Projen](#) et [Comment simplifier la configuration d'un projet avec Projen](#).

## Commencer à utiliser des plans personnalisés

Au cours du processus de création d'un plan, vous pouvez configurer le plan et générer un aperçu des ressources du projet. Chaque plan personnalisé est géré par un CodeCatalyst projet, qui contient un flux de travail par défaut pour la publication dans le catalogue de plans de l'espace.

### Rubriques

- [Prérequis](#)
- [Étape 1 : créer un plan personnalisé dans CodeCatalyst](#)
- [Étape 2 : développer un plan personnalisé avec des composants](#)
- [Étape 3 : prévisualiser un plan personnalisé](#)
- [\(Facultatif\) Étape 4 : Publier une version d'aperçu du plan personnalisé](#)

### Prérequis

Avant de créer un plan personnalisé, tenez compte des exigences suivantes :

- Votre CodeCatalyst espace doit être de niveau Enterprise. Pour plus d'informations, consultez [la section Gestion de la facturation](#) dans le manuel Amazon CodeCatalyst Administrator Guide.
- Vous devez avoir le rôle d'administrateur de l'espace ou d'utilisateur avancé pour créer des plans personnalisés. Pour de plus amples informations, veuillez consulter [Utilisation des rôles sur Amazon CodeCatalyst](#).

### Étape 1 : créer un plan personnalisé dans CodeCatalyst

Lorsque vous créez un plan personnalisé à partir des paramètres de votre espace, un référentiel est créé pour vous. Le référentiel inclut toutes les ressources nécessaires dont vous avez besoin pour développer votre plan avant de le publier dans le catalogue de plans de l'espace.

Pour créer un plan personnalisé

- a. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
- b. Dans la CodeCatalyst console, accédez à l'espace dans lequel vous souhaitez créer un plan personnalisé.

- c. Sur le tableau de bord spatial, cliquez sur l'onglet Paramètres, puis sur Blueprints.
- d. Choisissez Créer un plan.
- e. Sous Nommer votre plan, entrez le nom que vous souhaitez attribuer à votre projet et les noms de ressources associés. Le nom doit être unique dans votre espace.
- f. Sous Détails du plan, procédez comme suit :
  - i. Dans le champ de saisie du nom d'affichage du plan, entrez un nom qui apparaîtra dans le catalogue de plans de votre espace.
  - ii. Dans le champ de saisie de texte Description, entrez une description pour votre plan personnalisé.
  - iii. Dans le champ de saisie du nom de l'auteur, entrez le nom de l'auteur pour votre plan personnalisé.
  - iv. (Facultatif) Choisissez les paramètres avancés.
    - A. Choisissez + Ajouter pour ajouter des balises au package .json fichier.
    - B. Choisissez le menu déroulant Licence, puis choisissez une licence pour votre plan personnalisé.
    - C. Dans le champ de saisie du nom du package Blueprint, entrez un nom pour identifier votre package Blueprint.
    - D. Par défaut, un flux de publication est généré à l'aide d'un plan de publication au sein de votre projet appelé Blueprint Builder. Le flux de travail publie la dernière version du plan dans votre espace lorsque vous apportez des modifications, car les autorisations de publication sont activées par le flux de travail de publication. Pour désactiver la génération du flux de travail, décochez la case Libérer le flux de travail.
- g. (Facultatif) Un projet de plan est fourni avec un code prédéfini pour prendre en charge la publication du plan dans le catalogue spatial. Pour afficher les fichiers de définition avec des mises à jour en fonction des paramètres de projet que vous avez sélectionnés, choisissez Afficher le code ou Afficher le flux de travail dans Générer un aperçu du plan.
- h. Choisissez Créer un plan.

Si vous n'avez pas désactivé la génération de flux de travail pour votre plan personnalisé, le flux de travail commence automatiquement à s'exécuter lorsque votre plan est créé. Lorsque le flux de travail est terminé, votre plan personnalisé peut être ajouté par défaut au catalogue de plans de

votre espace. Vous pouvez désactiver les autorisations de publication si vous ne souhaitez pas que la dernière version du plan soit publiée automatiquement dans votre espace. Pour de plus amples informations, veuillez consulter [Appliquer et dissocier des plans dans un projet](#).

Étant donné que le flux de publication appelé `blueprint-release` est créé à l'aide d'un plan, celui-ci peut être trouvé sous forme de plan appliqué dans votre projet. Pour plus d'informations, consultez [Appliquer et dissocier des plans dans un projet](#) et [Utilisation des flux de travail](#).

## Étape 2 : développer un plan personnalisé avec des composants

Un assistant de plan est généré lorsque vous créez un plan personnalisé, et il peut être modifié à l'aide de composants lors du développement du plan personnalisé. Vous pouvez mettre à jour les `src/defaults.json` fichiers `src/blueprints.js` et pour modifier l'assistant.

### Important

Si vous souhaitez utiliser des packages de plans provenant de sources externes, prenez en compte les risques qui peuvent découler de ces packages. Vous êtes responsable des plans personnalisés que vous ajoutez à votre espace et du code qu'ils génèrent.

Créez un environnement de développement dans votre CodeCatalyst projet avec un environnement de développement intégré (IDE) pris en charge avant de configurer votre code de plan. Un environnement de développement est nécessaire pour travailler avec les outils et les packages requis.

Pour créer un environnement de développement

1. Dans le volet de navigation, effectuez l'une des opérations suivantes :
  - a. Choisissez Overview, puis accédez à la section Mes environnements de développement.
  - b. Choisissez Code, puis Dev Environments.
  - c. Choisissez Code, choisissez Référentiels sources et choisissez le référentiel que vous avez créé lors de la création de votre plan.
2. Choisissez Create Dev Environment.
3. Choisissez un IDE compatible dans le menu déroulant. Voir Environnements de [développement intégrés pris en charge pour les environnements](#) de développement pour plus d'informations.

4. Choisissez Travailler dans une branche existante, puis dans le menu déroulant Branche existante, sélectionnez la branche de fonctionnalités que vous avez créée.
5. (Facultatif) Dans le champ Alias - saisie de texte facultatif, entrez un alias pour identifier l'environnement de développement.
6. Choisissez Créer. Pendant la création de votre environnement de développement, la colonne d'état de l'environnement de développement affiche Démarrage et la colonne d'état indique En cours d'exécution lorsque l'environnement de développement a été créé.

Pour de plus amples informations, veuillez consulter [Environnements de développement dans CodeCatalyst](#).

Pour développer votre plan personnalisé

1. Dans un terminal fonctionnel, utilisez la yarn commande suivante pour installer les dépendances :

```
yarn
```

Les outils et packages requis sont disponibles via l'environnement de CodeCatalyst développement, y compris Yarn. Si vous travaillez sur un plan personnalisé sans environnement de développement, installez d'abord Yarn sur votre système. Pour plus d'informations, consultez la [documentation d'installation de Yarn](#).

2. Développez votre plan personnalisé afin qu'il soit configuré selon vos préférences. Vous pouvez modifier l'assistant de votre plan en ajoutant des composants. Pour plus d'informations, consultez [Élaboration d'un plan personnalisé](#), [Travailler avec un magicien](#) et [Publication d'un plan personnalisé](#).

### Étape 3 : prévisualiser un plan personnalisé

Après avoir configuré et développé votre plan personnalisé, vous pouvez prévisualiser et publier la version d'aperçu de votre plan dans votre espace. Une version d'aperçu vous permet de vérifier que le plan correspond à vos attentes avant de l'utiliser pour créer de nouveaux projets ou de l'appliquer à des projets existants.

Pour prévisualiser un plan personnalisé

1. Dans un terminal en état de fonctionnement, utilisez la yarn commande suivante :



```
yarn blueprint:preview
```

2. Accédez au `See this blueprint at:` lien fourni pour prévisualiser votre plan personnalisé.
3. Vérifiez que l'interface utilisateur, y compris le texte, s'affiche comme prévu en fonction de votre configuration. Si vous souhaitez modifier votre plan personnalisé, vous pouvez modifier le `blueprint.ts` fichier, resynthétiser le plan, puis publier à nouveau une version d'aperçu. Pour de plus amples informations, veuillez consulter [Resynthèse](#).

## (Facultatif) Étape 4 : Publier une version d'aperçu du plan personnalisé

Vous pouvez publier une version préliminaire de votre plan personnalisé dans votre espace si vous souhaitez l'ajouter au catalogue de plans de votre espace. Cela vous permet de visualiser le plan en tant qu'utilisateur avant d'ajouter la version non prévisualisée au catalogue. La version d'aperçu vous permet de publier sans passer à une version réelle. Par exemple, si vous travaillez sur une `0.0.1` version, vous pouvez publier et ajouter une version d'aperçu, afin que les nouvelles mises à jour d'une deuxième version puissent être publiées et ajoutées en tant que `0.0.2`.

Pour publier une version d'aperçu d'un plan personnalisé

Accédez au `Enable version [version number] at:` lien fourni pour activer votre plan personnalisé. Ce lien est fourni lors de l'exécution de la `yarn` commande [Étape 3 : prévisualiser un plan personnalisé](#).

Après avoir créé, développé, prévisualisé et publié votre plan personnalisé, vous pouvez publier et ajouter la version finale du plan au catalogue de plans de votre espace. Pour de plus amples informations, veuillez consulter [Ajouter et supprimer un plan personnalisé dans un espace](#).

## Tutoriel : Création et mise à jour d'une application React

En tant qu'auteur de plans, vous pouvez développer et ajouter des plans personnalisés au catalogue de plans de votre espace. Ces plans peuvent ensuite être utilisés par les membres de l'espace pour créer de nouveaux projets ou les appliquer à des projets existants. Vous pouvez continuer à apporter des modifications à vos plans, qui sont ensuite mis à disposition sous forme de mises à jour par le biais de pull requests.

Ce didacticiel fournit une présentation détaillée du point de vue de l'auteur du plan et du point de vue de l'utilisateur du plan. Le didacticiel montre comment créer un plan d'application Web d'une seule page React. Le plan est ensuite utilisé pour créer un nouveau projet. Lorsque le plan est mis à jour

avec des modifications, le projet créé à partir du plan intègre ces modifications par le biais d'une pull request.

## Rubriques

- [Prérequis](#)
- [Étape 1 : créer un plan personnalisé](#)
- [Étape 2 : Afficher le flux de production](#)
- [Étape 3 : ajouter un plan au catalogue](#)
- [Étape 4 : Création d'un projet avec un plan](#)
- [Étape 5 : Mettre à jour le plan](#)
- [Étape 6 : mise à jour de la version du catalogue publiée du plan vers la nouvelle version](#)
- [Étape 7 : Mettre à jour le projet avec la nouvelle version du plan](#)
- [Étape 8 : Afficher les modifications apportées au projet](#)

## Prérequis

Pour créer et mettre à jour un plan personnalisé, vous devez avoir effectué les tâches [Configuration CodeCatalyst](#) suivantes :

- Disposez d'un identifiant AWS Builder pour vous connecter à CodeCatalyst.
- Appartenez à un espace et le rôle d'administrateur de l'espace ou d'utilisateur avancé vous est attribué dans cet espace. Pour plus d'informations, consultez [Création d'un espace prenant en charge les utilisateurs de AWS Builder ID](#), [Gestion des utilisateurs de l'espace](#) et [Rôle d'administrateur de l'espace](#).

## Étape 1 : créer un plan personnalisé

Lorsque vous créez un plan personnalisé, un CodeCatalyst projet contenant le code source de votre plan ainsi que des outils et ressources de développement est créé. C'est dans votre projet que vous allez développer, tester et publier le plan.

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la CodeCatalyst console, accédez à l'espace dans lequel vous souhaitez créer un plan.
3. Choisissez Paramètres pour accéder aux paramètres de l'espace.
4. Dans l'onglet Paramètres de l'espace, choisissez Blueprints, puis Create Blueprint.

5. Mettez à jour les champs dans l'assistant de création de plans avec les valeurs suivantes :
  - Dans Nom du plan, entrez `react-app-blueprint`.
  - Dans Nom d'affichage du plan, entrez `react-app-blueprint`.
6. Vous pouvez éventuellement choisir Afficher le code pour prévisualiser le code source du plan de votre plan. De même, choisissez Afficher le flux de travail pour prévisualiser le flux de travail qui sera créé dans le projet qui crée et publie le plan.
7. Choisissez Créer un plan.
8. Une fois votre plan créé, vous êtes redirigé vers le projet du plan. Ce projet contient le code source du plan, ainsi que les outils et les ressources dont vous avez besoin pour développer, tester et publier le plan. Un flux de travail de publication a été généré et votre plan a été automatiquement publié dans l'espace.
9. Maintenant que votre plan et votre projet de plan sont créés, l'étape suivante consiste à le configurer en mettant à jour le code source. Vous pouvez utiliser les environnements de développement pour ouvrir et modifier votre référentiel source directement dans votre navigateur.

Dans le volet de navigation, choisissez Code, puis Dev Environments.

10. Choisissez Create Dev Environment, puis choisissez AWS Cloud9 (dans le navigateur).
11. Conservez les paramètres par défaut et choisissez Create.
12. Dans le AWS Cloud9 terminal, accédez au répertoire de votre projet de plan en exécutant la commande suivante :

```
cd react-app-blueprint
```

13. Un `static-assets` dossier est créé et rempli automatiquement lorsqu'un plan est créé. Dans ce didacticiel, vous allez supprimer le dossier par défaut et en générer un nouveau pour le plan d'une application React.

Supprimez le dossier `static-assets` en exécutant la commande suivante :

```
rm -r static-assets
```

AWS Cloud9 est construit sur une plate-forme basée sur Linux. Si vous utilisez un système d'exploitation Windows, vous pouvez utiliser la commande suivante à la place :

```
rmdir /s /q static-assets
```

14. Maintenant que le dossier par défaut est supprimé, créez un `static-assets` dossier pour un plan React-App en exécutant la commande suivante :

```
npx create-react-app static-assets
```

Si vous y êtes invité, entrez `y` pour continuer.

Une nouvelle application React a été créée dans le `static-assets` dossier contenant les packages nécessaires. Les modifications doivent être transmises à votre référentiel CodeCatalyst source distant.

15. Vérifiez que vous disposez des dernières modifications, puis validez et transférez les modifications dans le référentiel CodeCatalyst source du plan en exécutant les commandes suivantes :

```
git pull
```

```
git add .
```

```
git commit -m "Add React app to static-assets"
```

```
git push
```

Lorsqu'une modification est apportée au référentiel source du plan, le flux de production démarre automatiquement. Ce flux de travail augmente la version du plan, crée le plan et le publie dans votre espace. À l'étape suivante, vous allez accéder au flux de travail de publication exécuté pour voir comment il fonctionne.

## Étape 2 : Afficher le flux de production

1. Dans le volet de navigation de la CodeCatalyst console, choisissez CI/CD, puis Workflows.
2. Choisissez le flux de travail de publication du plan.
3. Vous pouvez voir que le flux de travail comporte des actions permettant de créer et de publier le plan.
4. Sous Dernière exécution, cliquez sur le lien d'exécution du flux de travail pour afficher l'exécution à partir de la modification de code que vous avez apportée.

5. Une fois l'exécution terminée, la nouvelle version de votre plan est publiée. Les versions de plans publiées sont visibles dans les paramètres de votre espace, mais ne peuvent pas être utilisées dans des projets tant qu'elles ne sont pas ajoutées au catalogue de plans de l'espace. À l'étape suivante, vous allez ajouter le plan au catalogue.

### Étape 3 : ajouter un plan au catalogue

L'ajout d'un plan au catalogue de plans de l'espace permet de l'utiliser dans tous les projets d'un espace. Les membres de l'espace peuvent utiliser le plan pour créer de nouveaux projets ou les appliquer à des projets existants.

1. Dans la CodeCatalyst console, revenez dans l'espace.
2. Choisissez Paramètres, puis Blueprints.
3. Choisissez react-app-blueprint, puis choisissez Ajouter au catalogue.
4. Choisissez Enregistrer.

### Étape 4 : Création d'un projet avec un plan

Maintenant que le plan est ajouté au catalogue, il peut être utilisé dans des projets. Au cours de cette étape, vous allez créer un projet avec le plan que vous venez de créer. Dans une étape ultérieure, vous mettrez à jour ce projet en mettant à jour et en publiant une nouvelle version du plan.

1. Choisissez l'onglet Projets, puis sélectionnez Créer un projet.
2. Choisissez Space Blueprints, puis choisissez react-app-blueprint.

#### Note

Une fois le plan choisi, vous pouvez voir le contenu du fichier du README .md plan.

3. Choisissez Suivant.

4.

#### Note

Le contenu de cet assistant de création de projet peut être configuré dans le plan.

Entrez le nom du projet en tant qu'utilisateur du plan. Dans le cadre de ce didacticiel, entrez `react-app-project`. Pour plus d'informations, consultez [Élaboration d'un plan personnalisé](#).

Vous allez ensuite mettre à jour le plan et ajouter la nouvelle version au catalogue, que vous utiliserez pour mettre à jour ce projet.

## Étape 5 : Mettre à jour le plan

Une fois qu'un plan est utilisé pour créer un nouveau projet ou qu'il est appliqué à des projets existants, vous pouvez continuer à le mettre à jour en tant qu'auteur du plan. Au cours de cette étape, vous allez apporter des modifications au plan et publier automatiquement une nouvelle version dans l'espace. La nouvelle version peut ensuite être ajoutée en tant que version du catalogue.

1. Accédez au `react-app-blueprint` projet créé dans [Tutoriel : Création et mise à jour d'une application React](#).
2. Ouvrez l'environnement de développement créé dans [Tutoriel : Création et mise à jour d'une application React](#).
  - a. Dans le volet de navigation, choisissez Code, puis Dev Environments.
  - b. Dans le tableau, recherchez l'environnement de développement, puis choisissez Ouvrir dans AWS Cloud9 (dans le navigateur).
3. Lorsque le flux de travail de publication du plan a été exécuté, il a incrémenté la version du plan en mettant à jour le fichier `package.json`. Intégrez cette modification en exécutant la commande suivante dans le AWS Cloud9 terminal :

```
git pull
```

4. Accédez au `static-assets` dossier en exécutant la commande suivante :

```
cd /projects/react-app-blueprint/static-assets
```

5. Créez un `hello-world.txt` fichier dans un `static-assets` dossier en exécutant la commande suivante :

```
touch hello-world.txt
```

AWS Cloud9 est construit sur une plate-forme basée sur Linux. Si vous utilisez un système d'exploitation Windows, vous pouvez utiliser la commande suivante à la place :

```
echo > hello-world.txt
```

6. Dans le menu de navigation de gauche, double-cliquez sur le `hello-world.txt` fichier pour l'ouvrir dans l'éditeur, puis ajoutez le contenu suivant :

```
Hello, world!
```

Enregistrez le fichier.

7. Vérifiez que vous disposez des dernières modifications, puis validez et transférez les modifications dans le référentiel CodeCatalyst source du plan en exécutant les commandes suivantes :

```
git pull
```

```
git add .
```

```
git commit -m "prettier setup"
```

```
git push
```

L'application des modifications a lancé le flux de travail de publication, qui publiera automatiquement la nouvelle version du plan dans l'espace.

## Étape 6 : mise à jour de la version du catalogue publiée du plan vers la nouvelle version

Une fois qu'un plan est utilisé pour créer un nouveau projet ou qu'il est appliqué à des projets existants, vous pouvez toujours le mettre à jour en tant qu'auteur du plan. Au cours de cette étape, vous allez apporter des modifications au plan et modifier la version du catalogue du plan.

1. Dans la CodeCatalyst console, revenez dans l'espace.
2. Choisissez Paramètres, puis Blueprints.
3. Choisissez `react-app-blueprint`, puis sélectionnez Gérer la version du catalogue.
4. Choisissez la nouvelle version, puis cliquez sur Enregistrer.

## Étape 7 : Mettre à jour le projet avec la nouvelle version du plan

Une nouvelle version est désormais disponible dans le catalogue des plans de l'espace. En tant qu'utilisateur du plan, vous pouvez mettre à jour la version du projet créé dans [Étape 4 : Création d'un projet avec un plan](#). Cela garantit que vous disposez des dernières modifications et corrections nécessaires pour respecter les meilleures pratiques.

1. Dans la CodeCatalyst console, accédez au react-app-projet projet créé dans [Étape 4 : Création d'un projet avec un plan](#).
2. Dans le panneau de navigation, sélectionnez Blueprints (Modèles).
3. Choisissez Mettre à jour le plan dans la boîte d'informations.
4. Dans le panneau de modification du code situé sur le côté droit, vous pouvez voir les package.json mises à jour hello-world.txt et.
5. Choisissez Appliquer la mise à jour.

Choisir Appliquer la mise à jour crée une pull request dans le projet avec les modifications apportées par rapport à la version mise à jour du plan. Pour mettre à jour le projet, vous devez fusionner la pull request. Pour plus d'informations, consultez [Révision d'une pull request](#) et [Fusion d'une pull request](#).

1. Dans le tableau des plans, recherchez le plan. Dans la colonne État, choisissez Pending pull request, puis cliquez sur le lien vers la pull request ouverte.
2. Passez en revue la pull request, puis choisissez Merge.
3. Choisissez Fusion rapide pour conserver les valeurs par défaut, puis choisissez Fusionner.

## Étape 8 : Afficher les modifications apportées au projet

Les modifications apportées au plan sont désormais disponibles dans votre projet par la suite [Étape 7 : Mettre à jour le projet avec la nouvelle version du plan](#). En tant qu'utilisateur du Blueprint, vous pouvez consulter les modifications dans le référentiel source.

1. Dans le volet de navigation, choisissez Référentiels sources, puis choisissez le nom du référentiel source créé lors de la création du projet.
2. Sous Fichiers, vous pouvez afficher le hello-world.txt fichier créé dans [Étape 5 : Mettre à jour le plan](#).
3. Choisissez le hello-world.txt pour afficher le contenu du fichier.



La gestion du cycle de vie permet aux auteurs de plans de gérer de manière centralisée le cycle de développement logiciel de chaque projet contenant un plan spécifique. Comme indiqué dans ce didacticiel, vous pouvez envoyer des mises à jour du plan qui peuvent ensuite être incorporées par les projets qui ont utilisé le plan pour créer un nouveau projet ou l'ont appliqué à un projet existant. Pour plus d'informations, voir [Travailler avec la gestion du cycle de vie en tant qu'auteur de plans](#).

## Travailler avec la gestion du cycle de vie en tant qu'auteur de plans

La gestion du cycle de vie vous permet de synchroniser un grand nombre de projets à partir d'une source commune unique de meilleures pratiques. Cela permet d'échelonner la propagation des correctifs et la maintenance d'un certain nombre de projets tout au long de leur cycle de vie de développement logiciel. La gestion du cycle de vie rationalise les campagnes internes, les correctifs de sécurité, les audits, les mises à niveau d'exécution, les modifications des meilleures pratiques et les autres pratiques de maintenance, car ces normes sont définies en un seul endroit et mises à jour automatiquement de manière centralisée lorsque de nouvelles normes sont publiées.

Lorsqu'une nouvelle version de votre plan est publiée, tous les projets contenant ce plan sont invités à passer à la dernière version. En tant qu'auteur d'un plan, vous pouvez également consulter la version d'un plan spécifique que contient chaque projet à des fins de conformité. En cas de conflit dans un référentiel source existant, la gestion du cycle de vie crée des pull requests. Pour toutes les autres ressources, telles que l'environnement de développement, toutes les mises à jour de gestion du cycle de vie créent strictement de nouvelles ressources. Les utilisateurs sont libres de fusionner ou de ne pas fusionner ces pull requests. Lorsque les pull requests en attente sont fusionnées, la version du plan, y compris les options, utilisée dans votre projet est ensuite mise à jour. Pour en savoir plus sur l'utilisation de la gestion du cycle de vie en tant qu'utilisateur du Blueprint, consultez [Utilisation de la gestion du cycle de vie sur les projets existants](#) et [Utilisation de la gestion du cycle de vie sur plusieurs plans d'un projet](#).

### Rubriques

- [Gestion du cycle de vie des tests](#)
- [Utilisation de stratégies de fusion](#)
- [Utilisation d'objets contextuels](#)

## Gestion du cycle de vie des tests

Vous pouvez tester localement la gestion du cycle de vie de votre plan et fusionner la résolution des conflits. Dans le synth/ répertoire, une série de bundles représentant les différentes phases d'une

mise à jour du cycle de vie est générée. Pour tester la gestion du cycle de vie, vous pouvez exécuter la commande yarn suivante sur votre plan :`yarn blueprint: resynth`. Pour en savoir plus sur la resynthèse et les bundles, consultez et. [Resynthèse Génération de fichiers par resynthèse](#)

## Utilisation de stratégies de fusion

### Rubriques

- [Génération de fichiers par resynthèse](#)
- [Utilisation de stratégies de fusion](#)
- [Spécification de fichiers pour les mises à jour de gestion du cycle](#)
- [Rédaction de stratégies de fusion](#)

### Génération de fichiers par resynthèse

La resynthèse peut fusionner le code source produit par un plan avec le code source précédemment généré par le même plan, ce qui permet de propager les modifications apportées à un plan aux projets existants. Les fusions sont exécutées à partir de la `resynth()` fonction sur les ensembles de sortie du plan. La resynthèse génère d'abord trois ensembles représentant différents aspects du plan et de l'état du projet. Il peut être exécuté manuellement localement à l'aide de la `yarn blueprint:resynth` commande, qui créera les bundles s'ils n'existent pas déjà. Travailler manuellement avec les ensembles vous permettra de simuler et de tester le comportement de resynthèse localement. Par défaut, les plans exécutent uniquement la resynthèse dans les référentiels situés sous, `src/*` car seule cette partie du bundle est généralement contrôlée par le code source.

- `existing-bundle`- Ce bundle est une représentation de l'état actuel du projet. Ceci est construit artificiellement par le calcul de synthèse pour donner au plan le contexte du contenu du projet dans lequel il est déployé (le cas échéant). Si quelque chose existe déjà à cet endroit lors de l'exécution de la resynthèse localement, il sera réinitialisé et considéré comme une maquette. Dans le cas contraire, il sera réglé sur le contenu du `ancestor-bundle`.
- `ancestor-bundle`- Il s'agit du bundle qui représente le résultat du plan s'il a été synthétisé avec certaines options et/ou versions précédentes. Si c'est la première fois que ce plan est ajouté à un projet, cela signifie que l'ancêtre n'existe pas. Il est donc défini sur le même contenu que le `existing-bundle`. Localement, si ce bundle existe déjà à cet endroit, il sera considéré comme une maquette.

- `proposed-bundle`- Il s'agit du bundle qui se moque du plan s'il a été synthétisé avec de nouvelles options et/ou versions. Il s'agit du même bundle qui serait produit par la `synth()` fonction. Localement, ce bundle est toujours remplacé.

Chaque bundle est créé lors d'une phase de resynthèse accessible depuis la classe `Blueprint` ci-dessous. `this.context.resynthesisPhase`

- `resolved-bundle`- Il s'agit du bundle final, qui est une représentation de ce qui est empaqueté et déployé dans un CodeCatalyst projet. Vous pouvez voir quels fichiers et différences sont envoyés aux mécanismes de déploiement. Il s'agit du résultat de la `resynth()` fonction résolvant les fusions entre les trois autres ensembles.

La fusion à trois voies est appliquée en prenant la différence entre `ancestor-bundle` et `proposed-bundle` et en l'appliquant au `existing-bundle` pour générer le `resolved-bundle`. Toutes les stratégies de fusion résolvent les fichiers au format `resolved-bundle`. La resynthèse résout la portée de ces ensembles grâce aux stratégies de fusion du plan pendant le processus `resynth()` et produit le bundle résolu à partir du résultat.

#### Utilisation de stratégies de fusion

Vous pouvez utiliser une stratégie de fusion proposée par la bibliothèque de plans. Ces stratégies fournissent des moyens de résoudre les sorties de fichiers et les conflits pour les fichiers mentionnés dans la [Génération de fichiers par resynthèse](#) section.

- `alwaysUpdate`- Une stratégie qui correspond toujours au fichier proposé.
- `neverUpdate`- Une stratégie qui répond toujours au fichier existant.
- `onlyAdd`- Une stratégie qui prend en compte le fichier proposé lorsqu'un fichier existant n'existe pas déjà. Dans le cas contraire, il est résolu dans le fichier existant.
- `threeWayMerge`- Une stratégie qui effectue une fusion à trois voies entre les fichiers d'ancêtres existants, proposés et communs. Le fichier résolu peut contenir des marqueurs de conflit si les fichiers ne peuvent pas être correctement fusionnés. Le contenu des fichiers fournis doit être codé en UTF-8 pour que la stratégie produise un résultat significatif. La stratégie tente de détecter si les fichiers d'entrée sont binaires. Si la stratégie détecte un conflit de fusion dans un fichier binaire, elle renvoie toujours le fichier proposé.

- `preferProposed`- Une stratégie qui effectue une fusion à trois voies entre les fichiers d'ancêtres existants, proposés et communs. Cette stratégie permet de résoudre les conflits en sélectionnant la version du fichier proposé pour chaque conflit.
- `preferExisting`- Une stratégie qui effectue une fusion à trois voies entre les fichiers d'ancêtres existants, proposés et communs. Cette stratégie permet de résoudre les conflits en sélectionnant le côté du fichier existant dans chaque conflit.

Pour consulter le code source des stratégies de fusion, consultez le [GitHub référentiel open source](#).

## Spécification de fichiers pour les mises à jour de gestion du cycle

Lors de la resynthèse, les plans contrôlent la manière dont les modifications sont fusionnées dans un référentiel source existant. Cependant, il se peut que vous ne souhaitiez pas envoyer des mises à jour à chaque fichier de votre plan. Par exemple, les exemples de code tels que les feuilles de style CSS sont conçus pour être spécifiques au projet. La stratégie de fusion à trois voies est l'option par défaut si vous ne spécifiez aucune autre stratégie. Les plans peuvent spécifier les fichiers qu'ils possèdent et ceux qu'ils ne possèdent pas en spécifiant des stratégies de fusion dans la structure du référentiel elle-même. Les plans peuvent mettre à jour leurs stratégies de fusion, et les dernières stratégies peuvent être utilisées lors de la resynthèse.

```
const sourceRepo = new SourceRepository(this, {
  title: 'my-repo',
});
sourceRepo.setResynthStrategies([
  {
    identifiant: 'dont-override-sample-code',
    description: 'This strategy is applied accross all sample code. The blueprint
will create sample code, but skip attempting to update it.',
    strategy: MergeStrategies.neverUpdate,
    globs: [
      '**/src/**',
      '**/css/**',
    ],
  },
]);
```

Plusieurs stratégies de fusion peuvent être spécifiées, la dernière étant prioritaire. Les fichiers non couverts sont par défaut three-way-merge similaires à Git. Plusieurs stratégies de fusion sont

proposées dans le MergeStrategies build, mais vous pouvez écrire la vôtre. Les stratégies fournies sont conformes au pilote de [stratégie git merge](#).

## Rédaction de stratégies de fusion

En plus d'utiliser l'une des stratégies de build merge fournies, vous pouvez également écrire vos propres stratégies. Les stratégies doivent respecter une interface de stratégie standard. Vous devez écrire une fonction de stratégie qui prend les versions d'un fichier à partir du `existing-bundle`, `proposed-bundle`, et `ancestor-bundle`, et les fusionne en un seul fichier résolu. Par exemple :

```
type StrategyFunction = (  
  /**  
   * file from the ancestor bundle (if it exists)  
   */  
  commonAncestorFile: ContextFile | undefined,  
  /**  
   * file from the existing bundle (if it exists)  
   */  
  existingFile: ContextFile | undefined,  
  /**  
   * file from the proposed bundle (if it exists)  
   */  
  proposedFile: ContextFile | undefined,  
  options?: {})  
  /**  
   * Return: file you'd like in the resolved bundle  
   * passing undefined will delete the file from the resolved bundle  
   */  
=> ContextFile | undefined;
```

Si les fichiers n'existent pas (ne sont pas définis), ce chemin de fichier n'existe pas dans cet ensemble d'emplacements en particulier.

## Exemple :

```
strategies: [  
  {  
    identifier: 'dont-override-sample-code',  
    description: 'This strategy is applied across all sample code. The  
    blueprint will create sample code, but skip attempting to update it.',  
    strategy: (ancestor, existing, proposed) => {
```

```
        const resolvedfile = ...
        ...
        // do something
        ...
        return resolvedfile
    },
    globs: [
        '**/src/**',
        '**/css/**',
    ],
},
],
```

## Utilisation d'objets contextuels

En tant qu'auteur d'un plan, vous pouvez accéder au contexte du projet du plan pendant la synthèse pour obtenir des informations telles que l'espace et les noms de projet, ou les fichiers existants dans le référentiel source d'un projet. Vous pouvez également obtenir des détails tels que la phase de resynthèse générée par le plan. Par exemple, vous pouvez accéder au contexte pour savoir si vous effectuez une resynthèse pour générer un bundle d'ancêtres ou un bundle proposé. Le contexte de code existant peut ensuite être utilisé pour transformer votre code dans votre référentiel. Par exemple, vous pouvez écrire votre propre stratégie de resynthèse pour définir des normes de code spécifiques. La stratégie peut être ajoutée au `blueprint.ts` fichier pour les petits plans, ou vous pouvez créer un fichier distinct pour les stratégies.

L'exemple suivant montre comment rechercher des fichiers dans le contexte d'un projet, définir un générateur de flux de travail et définir une stratégie de resynthèse basée sur un plan pour un fichier en particulier :

```
const contextFiles = this.context.project.src.findAll({
  fileGlobs: ['**/package.json'],
});

// const workflows = this.context.project.src.findAll({
//   fileGlobs: ['**/.codecatalyst/**/*.yaml'],
// });

const security = new WorkflowBuilder(this, {
  Name: 'security-workflow',
});
new Workflow(this, repo, security.getDefinition());
```

```
repo.setResynthStrategies([
  {
    identifiant: 'force-security',
    globs: ['**/.codecatalyst/security-workflow.yaml'],
    strategy: MergeStrategies.alwaysUpdate,
  },
]);

for (const contextFile of contextFiles) {
  const packageObject = JSON.parse(contextFile.buffer.toString());
  new SourceFile(internalRepo, contextFile.path, JSON.stringify({
    ...packageObject,
  }, null, 2));
}
}
```

## Élaboration d'un plan personnalisé

Avant de publier un plan personnalisé, vous pouvez le développer pour répondre à des exigences spécifiques. Vous pouvez développer votre plan personnalisé et le tester en créant un projet lors de la prévisualisation. Vous pouvez développer le plan personnalisé pour inclure les composants du projet, tels que le code source spécifique, les connexions aux comptes, les flux de travail, les problèmes ou tout autre composant pouvant être créé dans CodeCatalyst.

### Important

Si vous souhaitez utiliser des packages de plans provenant de sources externes, prenez en compte les risques qui peuvent découler de ces packages. Vous êtes responsable des plans personnalisés que vous ajoutez à votre espace et du code qu'ils génèrent.

Pour développer ou mettre à jour un plan personnalisé

1. Reprenez votre environnement de développement. Pour de plus amples informations, veuillez consulter [Reprise d'un environnement de développement](#).

Si vous n'avez pas d'environnement de développement, vous devez d'abord en créer un. Pour de plus amples informations, veuillez consulter [Création d'un environnement de développement](#).

2. Ouvrez un terminal fonctionnel dans votre environnement de développement.

3. Si vous avez opté pour un flux de publication lors de la création de votre plan, la dernière version du plan est automatiquement publiée. Extrayez les modifications pour vous assurer que le package `.json` fichier possède la version incrémentée. Utilisez la commande suivante :

```
git pull
```

4. Dans le `src/blueprint.ts` fichier, modifiez les options de votre plan personnalisé. L'`Optionsinterface` est interprétée dynamiquement par l' `CodeCatalyst` assistant pour générer une interface utilisateur (UI) de sélection. Vous pouvez développer votre plan personnalisé en ajoutant des composants et des balises compatibles. Pour plus d'informations, consultez [Travailler avec un magicien](#), [Composants de l'environnement](#), [Composantes de la région](#), [Composants du référentiel et du code source](#), [Composants du flux de travail](#) et [Composants des environnements de développement](#).

Vous pouvez également consulter le SDK de plans et des exemples de plans pour bénéficier d'une assistance supplémentaire lors du développement de votre plan personnalisé. Pour plus d'informations, consultez le [GitHub référentiel open source](#).

## Travailler avec un magicien

Un assistant de sélection de plans activé `CodeCatalyst` est généré automatiquement par l'`Optionsinterface` du fichier `blueprint.ts`. L'assistant frontal prend en charge les modifications et les fonctionnalités d'un plan à l'`Optionsaide` de [commentaires et de balises de style JSDOC](#).

Vous pouvez utiliser des commentaires et des balises de style JSDOC pour effectuer des tâches. Par exemple, vous pouvez sélectionner le texte affiché au-dessus d'une option, activer des fonctionnalités telles que la validation des entrées ou rendre une option pliable. L'assistant fonctionne en interprétant un arbre syntaxique abstrait (AST) généré à partir du TypeScript type de l'`Optionsinterface`.

L'assistant se configure automatiquement selon le type décrit du mieux qu'il peut. Tous les types ne sont pas pris en charge. Les autres types pris en charge incluent le sélecteur de région et le sélecteur d'environnement.

Voici un exemple d'assistant qui utilise des commentaires et des balises JSDOC avec des plans :

### Options

```
export interface Options {  
  /**  
   * What do you want to call your new blueprint?  
   * @validationRegex /^[a-zA-Z0-9_]+$/
```



```
    * @validationMessage Must contain only upper and lowercase letters, numbers and
underscores
    */
    blueprintName: string;

    /**
    * Add a description for your new blueprint.
    */
    description?: string;

    /**
    * Tags for your Blueprint:
    * @collapsed true
    */
    tags?: string[];
}
```

Le nom d'affichage de chaque option de l'Optionsinterface apparaît camelCase par défaut. Le texte brut dans le commentaire de style JSDOC est affiché sous forme de texte au-dessus de l'option dans l'assistant.

## Rubriques

- [Tags pris en charge](#)
- [TypeScript Types pris en charge](#)
- [Communiquer avec l'utilisateur lors de la synthèse](#)

## Tags pris en charge

Les balises JSDOC suivantes sont prises en charge par un plan personnalisé Options dans l'assistant frontal.

`@inlinePolicy. /path/to/policy/file.json`

- Nécessite : l'option doit être un typeRole.
- Utilisation : vous permet de communiquer les politiques intégrées dont un rôle a besoin. Le `policy.json` chemin devrait se trouver sous le code source. Utilisez cette balise lorsque vous avez besoin d'une politique personnalisée pour un rôle.
- Dépendances - `blueprint-cli 0.1.12` et supérieures
- Exemple - `@inlinePolicy ./deployment-policy.json`

```
environment: EnvironmentDefinition{
  awsAccountConnection: AccountConnection{
    /**
     * @inlinePolicy ./path/to/deployment-policy.json
     */
    cdkRole: Role[];
  };
};
```

### @trustPolicy. /path/to/policy/file.json

- Nécessite : l'option doit être un typeRole.
- Utilisation : vous permet de communiquer les politiques de confiance dont un rôle a besoin. Le policy.json chemin devrait se trouver sous le code source. Utilisez cette balise lorsque vous avez besoin d'une politique personnalisée pour un rôle.
- Dépendances - blueprint-cli 0.1.12 et supérieures
- Exemple - @trustPolicy ./trust-policy.json

```
environment: EnvironmentDefinition{
  awsAccountConnection: AccountConnection{
    /**
     * @trustPolicy ./path/to/trust-policy.json
     */
    cdkRole: Role[];
  };
};
```

### Expression Regex @validationRegex

- Nécessite : l'option doit être une chaîne.
- Utilisation - Effectue la validation des entrées sur l'option en utilisant l'expression regex donnée et les affichages@validationMessage.
- Exemple - @validationRegex /^[a-zA-Z0-9\_]+\$
- Recommandation - À utiliser avec@validationMessage. Le message de validation est vide par défaut.

## chaîne @validationMessage

- Nécessite des erreurs @validationRegex ou d'autres erreurs pour vérifier l'utilisation.
- Utilisation : affiche un message de validation en @validation\* cas d'échec.
- Exemple -@validationMessage Must contain only upper and lowercase letters, numbers, and underscores.
- Recommandation - À utiliser avec@validationMessage. Le message de validation est vide par défaut.

## @collapsed boolean (facultatif)

- Nécessite - N/A
- Utilisation : booléen permettant à une sous-option d'être repliable. Si l'annotation réduite est présente, sa valeur par défaut est vraie. La définition de la valeur sur @collapsed false crée une section pliable qui est initialement ouverte.
- Exemple - @collapsed true

## chaîne @displayName

- Nécessite - N/A
- Utilisation - Modifie le nom d'affichage de l'option. Autorise des formats autres que CamelCase pour le nom d'affichage.
- Exemple - @displayName Blueprint Name

## chaîne @displayName

- Nécessite - N/A
- Utilisation - Modifie le nom d'affichage de l'option. Autorise des formats autres que [CamelCase](#) pour le nom d'affichage.
- Exemple - @displayName Blueprint Name

## Numéro @defaultEntropy

- Nécessite : l'option doit être une chaîne.
- Utilisation : ajoute une chaîne alphanumérique aléatoire d'une longueur spécifiée à l'option.

- Exemple - `@defaultEntropy 5`

chaîne `@placeholder` (facultatif)

- Nécessite - N/A
- Utilisation - Modifie l'espace réservé aux champs de texte par défaut.
- Exemple - `@placeholder type project name here`

Numéro `@textArea` (facultatif)

- Nécessite - N/A
- Utilisation - Convertit l'entrée sous forme de chaîne en composant de zone de texte pour les grandes sections de texte. L'ajout d'un nombre définit le nombre de lignes. La valeur par défaut est de cinq lignes.
- Exemple - `@textArea 10`

`@hidden` boolean (facultatif)

- Nécessite - N/A
- Utilisation : masque le fichier à l'utilisateur sauf si le contrôle de validation échoue. La valeur par défaut est true.
- Exemple - `@hidden`

`@button` boolean (facultatif)

- Nécessite - N/A
- Utilisation : l'annotation doit figurer sur une propriété booléenne. Ajoute un bouton qui sera synthétisé comme vrai une fois sélectionné. Ce n'est pas une bascule.
- Exemple - `buttonExample: boolean;`

```
/**
 * @button
 */
buttonExample: boolean;
```

### @showName boolean (facultatif)

- Nécessite - N/A
- Utilisation : ne peut être utilisé que sur un type de connexion à un compte. Affiche la saisie d'un nom masqué. La valeur par défaut est `default_environment`.
- Exemple - `@showName true`

```
/**
 * @showName true
 */
accountConnection: AccountConnection<{
  ...
}>;
```

### @ showEnvironmentType boolean (facultatif)

- Nécessite - N/A
- Utilisation : ne peut être utilisé que sur un type de connexion à un compte. Affiche le menu déroulant des types d'environnement masqués. Toutes les connexions sont définies par défaut sur `production`. Les options sont la non-production ou la production.
- Exemple - `@showEnvironmentType true`

```
/**
 * @showEnvironmentType true
 */
accountConnection: AccountConnection<{
  ...
}>;
```

### @forceDefault boolean (facultatif)

- Nécessite - N/A
- Utilisation : utilise la valeur par défaut fournie par l'auteur du plan au lieu de la valeur utilisée précédemment par l'utilisateur.
- Exemple - `forceDefaultExample: any;`

```
/**
```

```
* @forceDefault
*/
forceDefaultExample: any;
```

## Nom du plan @requires

- Nécessite : annote l'interface `Options`
- Utilisation : avertit l'utilisateur d'appliquer `blueprintName` la spécification spécifiée au projet comme exigence pour le plan actuel.
- Exemple - @requires '@amazon-codecatalyst/blueprints.blueprint-builder'

```
/*
 * @requires '@amazon-codecatalyst/blueprints.blueprint-builder'
 */
export interface Options extends ParentOptions {
  ...
}
```

## TypeScript Types pris en charge

Les TypeScript types suivants sont pris en charge par un plan personnalisé `Options` dans l'assistant frontal.

### Nombre

- Nécessite : l'option doit être un `number`.
- Utilisation - Générez un champ de saisie numérique.
- Exemple - `age: number`

```
{
  age: number
  ...
}
```

### Chaîne

- Nécessite : l'option doit être un `string`.
- Utilisation - Génère un champ de saisie de chaîne.

- Exemple - name: string

```
{
  age: string
  ...
}
```

### Liste de chaînes

- Nécessite : l'option doit être un typeboolean.
- Utilisation - Générez une case à cocher.
- Exemple - isProduction: boolean

```
{
  isProduction: boolean
  ...
}
```

### Radio

- Nécessite : cette option doit être une union de trois chaînes ou moins.
- Utilisation - Génère une radio sélectionnée.

#### Note

Lorsqu'il y a quatre éléments ou plus, ce type s'affiche sous forme de liste déroulante.

- Exemple - color: 'red' | 'blue' | 'green'

```
{
  color: 'red' | 'blue' | 'green'
  ...
}
```

### Liste déroulante

- Nécessite : option devant être une union de quatre chaînes ou plus.

- Utilisation : générez une liste déroulante.
- Exemple - `runtimes: 'nodejs' | 'python' | 'java' | 'dotnetcore' | 'ruby'`

```
{  
  runtimes: 'nodejs' | 'python' | 'java' | 'dotnetcore' | 'ruby'  
  ...  
}
```

## Section extensible

- Nécessite : option pour être un objet.
- Utilisation - Générez une section extensible. Les options de l'objet seront imbriquées dans la section extensible de l'assistant.
- Exemple -

```
{  
  expandableSectionTitle: {  
    nestedString: string;  
    nestedNumber: number;  
  }  
}
```

## Tuple

- Nécessite : l'option doit être de type `Tuple`.
- Utilisation - Générez une entrée payante à valeur clé.
- Exemple - `tuple: Tuple[string, string]>`

```
{  
  tuple: Tuple[string, string]>;  
  ...  
}
```

## Liste de tuples

- Nécessite : l'option doit être un tableau de types `Tuple`.



- Utilisation - Générez une entrée de liste de tuples.
- Exemple - `tupleList: Tuple[string, string]>[]`

```
{
  tupleList: Tuple[string, string]>[];
  ...
}
```

## Selector

- Nécessite : l'option doit être de type `Selector`.
- Utilisation : générez une liste déroulante des référentiels sources ou des plans appliqués à un projet.
- Exemple - `sourceRepo: Selector<SourceRepository>`

```
{
  sourceRepo: Selector<SourceRepository>;
  sourceRepoOrAdd: Selector<SourceRepository | string>;
  blueprintInstantiation: Selector<BlueprintInstantiation>;
  ...
}
```

## Sélection multiple

- Nécessite : l'option doit être de type `Selector`.
- Utilisation - Génère une entrée à sélection multiple.
- Exemple - `multiselect: MultiSelect['A' | 'B' | 'C' | 'D' | 'E']>`

```
{
  multiselect: MultiSelect['A' | 'B' | 'C' | 'D' | 'E']>;
  ...
}
```

## Communiquer avec l'utilisateur lors de la synthèse

En tant qu'auteur d'un plan, vous pouvez communiquer avec les utilisateurs au-delà des simples messages de validation. Par exemple, un membre de l'espace peut afficher une combinaison

d'options qui produit un plan qui n'est pas clair. Les plans personnalisés permettent de communiquer des messages d'erreur aux utilisateurs en invoquant la synthèse. Le plan de base implémente une `throwSynthesisError(...)` fonction qui attend un message d'erreur clair. Vous pouvez appeler le message en utilisant ce qui suit :

```
//blueprint.ts
this.throwSynthesisError({
  name: BlueprintSynthesisErrorTypes.BlueprintSynthesisError,
  message: 'hello from the blueprint! This is a custom error communicated to the
user.'
})
```

## Composants de l'environnement

L'assistant de création de plans personnalisés est généré dynamiquement à partir de l'`Optionsinterface` exposée par l'intermédiaire de l'assistant. Les plans permettent de générer des composants d'interface utilisateur (UI) à partir de types exposés.

Pour importer les composants de l'environnement Amazon CodeCatalyst Blueprints

Dans votre `blueprint.ts` fichier, ajoutez les éléments suivants :

```
import {...} from '@amazon-codecatalyst/codecatalyst-environments'
```

## Rubriques

- [Création d'environnements de développement](#)
- [Exemples d'interfaces fictives](#)

## Création d'environnements de développement

L'exemple suivant montre comment déployer votre application dans le cloud :

```
export interface Options extends ParentOptions {
  ...
  myNewEnvironment: EnvironmentDefinition{
    thisIsMyFirstAccountConnection: AccountConnection{
      thisIsARole: Role['lambda', 's3', 'dynamo'];
    };
  };
};
```

```
}

```

L'interface génère un composant d'interface utilisateur qui demande un nouvel environnement (`myNewEnvironment`) avec une connexion à un seul compte (`thisIsMyFirstAccountConnection`). Un rôle sur la connexion au compte (`thisIsARole`) est également généré avec `['lambda', 's3', 'dynamo']` les capacités de rôle minimales requises. Tous les utilisateurs ne sont pas connectés à un compte. Vous devez donc vérifier s'il n'y a pas de cas où un utilisateur ne connecte pas de compte ou n'associe pas un compte à un rôle. Les rôles peuvent également être annotés avec `@inlinePolicies`. Pour de plus amples informations, veuillez consulter [@inlinePolicy. /path/to/policy/file.json](#).

Le composant environnemental nécessite un `environmentType`. Le code suivant est la forme par défaut minimale requise :

```
{
  ...
  "myNewEnvironment": {
    "name": "myProductionEnvironment",
    "environmentType": "PRODUCTION"
  },
}
```

Le composant d'interface utilisateur vous invite ensuite à saisir différents champs. Au fur et à mesure que vous remplissez les champs, le plan prend une forme complètement développée. Il peut être utile d'inclure la maquette complète dans le `defaults.json` fichier à des fins de test et de développement.

## Exemples d'interfaces fictives

### Interface fictive simple

```
{
  ...
  "thisIsMyEnvironment": {
    "name": "myProductionEnvironment",
    "environmentType": "PRODUCTION",
    "thisIsMySecondAccountConnection": {
      "id": "12345678910",
      "name": "my-account-connection-name",
      "secondAdminRole": {
        "arn": "arn:aws:iam::12345678910:role/ConnectedQuokkaRole",

```

```

        "name": "ConnectedQuokkaRole",
        "capabilities": [
            "lambda",
            "s3",
            "dynamo"
        ]
    }
}
}
}
}

```

## Interface fictive complexe

```

export interface Options extends ParentOptions {
    /**
     * The name of an environment
     * @displayName This is a Environment Name
     * @collapsed
     */
    thisIsMyEnvironment: EnvironmentDefinition{
        /**
         * comments about the account that is being deployed into
         * @displayName This account connection has an overridden name
         * @collapsed
         */
        thisIsMyFirstAccountConnection: AccountConnection{
            /**
             * Blah blah some information about the role that I expect
             * e.g. here's a copy-pastable policy: [to a link]
             * @displayName This role has an overridden name
             */
            adminRole: Role['admin', 'lambda', 's3', 'cloudfront'];
            /**
             * Blah blah some information about the second role that I expect
             * e.g. here's a copy-pastable policy: [to a link]
             */
            lambdaRole: Role['lambda', 's3'];
        };
    };
    /**
     * comments about the account that is being deployed into
     */
    thisIsMySecondAccountConnection: AccountConnection{
        /**

```

```

    * Blah blah some information about the role that I expect
    * e.g. here's a copy-pastable policy: [to a link]
    */
secondAdminRole: Role['admin', 'lambda', 's3', 'cloudfront'];
/**
    * Blah blah some information about the second role that I expect
    * e.g. here's a copy-pastable policy: [to a link]
    */
secondLambdaRole: Role['lambda', 's3'];
};
};
}

```

## Interface fictive complète

```

{
  ...
  "thisIsMyEnvironment": {
    "name": "my-production-environment",
    "environmentType": "PRODUCTION",
    "thisIsMySecondAccountConnection": {
      "id": "12345678910",
      "name": "my-connected-account",
      "secondAdminRole": {
        "name": "LambdaQuokkaRole",
        "arn": "arn:aws:iam::12345678910:role/LambdaQuokkaRole",
        "capabilities": [
          "admin",
          "lambda",
          "s3",
          "cloudfront"
        ]
      },
      "secondLambdaRole": {
        "name": "LambdaQuokkaRole",
        "arn": "arn:aws:iam::12345678910:role/LambdaQuokkaRole",
        "capabilities": [
          "lambda",
          "s3"
        ]
      }
    },
    "thisIsMyFirstAccountConnection": {

```

```
"id": "12345678910",
"name": "my-connected-account",
"adminRole": {
  "name": "LambdaQuokkaRole",
  "arn": "arn:aws:iam::12345678910:role/LambdaQuokkaRole",
  "capabilities": [
    "admin",
    "lambda",
    "s3",
    "cloudfront"
  ]
},
"lambdaRole": {
  "name": "LambdaQuokkaRole",
  "arn": "arn:aws:iam::12345678910:role/LambdaQuokkaRole",
  "capabilities": [
    "lambda",
    "s3"
  ]
}
},
}
```

## Composants secrets

Les secrets peuvent être utilisés CodeCatalyst pour stocker des données sensibles qui peuvent être référencées dans les flux de travail. Vous pouvez ajouter un secret à votre plan personnalisé et le référencer dans votre flux de travail. Pour de plus amples informations, veuillez consulter [Travailler avec des secrets](#).

Pour importer des CodeCatalyst plans Amazon, tapez le type de région

Dans votre `blueprint.ts` fichier, ajoutez les éléments suivants :

```
import { Secret, SecretDefinition } from '@amazon-codecatalyst/blueprint-
component.secrets'
```

## Rubriques

- [Création d'un secret](#)
- [Référencer un secret dans un flux de travail](#)

## Création d'un secret

L'exemple suivant crée un composant d'interface utilisateur qui invite l'utilisateur à saisir une valeur secrète et une description facultative :

```
export interface Options extends ParentOptions {
  ...
  mySecret: SecretDefinition;
}

export class Blueprint extends ParentBlueprint {
  constructor(options_: Options) {
    new Secret(this, options.secret);
  }
}
```

Le composant secret nécessite unname. Le code suivant est la forme par défaut minimale requise :

```
{
  ...
  "secret": {
    "name": "secretName"
  },
}
```

## Référencer un secret dans un flux de travail

L'exemple de plan suivant crée un secret et un flux de travail qui fait référence à la valeur secrète. Pour de plus amples informations, veuillez consulter [Référencer un secret dans un flux de travail](#).

```
export interface Options extends ParentOptions {
  ...
  /**
   *
   * @validationRegex /^\\w+$/
   */
  username: string;

  password: SecretDefinition;
}
```

```
export class Blueprint extends ParentBlueprint {
  constructor(options_: Options) {
    const password = new Secret(this, options_.password);

    const workflowBuilder = new WorkflowBuilder(this, {
      Name: 'my_workflow',
    });

    workflowBuilder.addBuildAction({
      actionName: 'download_files',
      input: {
        Sources: ['WorkflowSource'],
      },
      output: {
        Artifacts: [{ Name: 'download', Files: ['file1'] }],
      },
      steps: [
        `curl -u ${options_.username}:${password.reference} https://example.com`,
      ],
    });

    new Workflow(
      this,
      repo,
      workflowBuilder.getDefinition(),
    );
  }
}
```

Pour en savoir plus sur l'utilisation de secrets dans CodeCatalyst, consultez [Travailler avec des secrets](#).

## Composantes de la région

Le type de région peut être ajouté à l'`Options` interface de votre plan personnalisé pour générer un composant. Dans l'assistant de plan, vous pouvez saisir une ou plusieurs régions AWS. Le type de région peut être importé depuis votre plan de base dans votre `blueprint.ts` fichier. Pour plus d'informations, consultez la section [Régions AWS](#).

Pour importer des CodeCatalyst plans Amazon, tapez le type de région



Dans votre `blueprint.ts` fichier, ajoutez les éléments suivants :

```
import { Region } from '@amazon-codecatalyst/blueprints.blueprint'
```

Le paramètre de type de région est un tableau de codes de région AWS parmi lesquels choisir, ou que vous pouvez utiliser `*` pour inclure toutes les régions AWS prises en charge.

## Rubriques

- [Annotations](#)
- [Exemples de composants régionaux](#)

## Annotations

Des balises JSDoc peuvent être ajoutées à chaque champ de l'`Optionsinterface` pour personnaliser la façon dont un champ apparaît et se comporte dans l'assistant. Pour le type de région, les balises suivantes sont prises en charge :

- L'`@displayName` annotation peut être utilisée pour modifier l'étiquette du champ dans l'assistant.

Exemple : `@displayName AWS Region`

- L'`@placeholder` annotation peut être utilisée pour modifier l'espace réservé du composant de sélection ou de sélection multiple.

Exemple : `@placeholder Choose AWS Region`

## Exemples de composants régionaux

### Choix d'une région dans une liste spécifiée

```
export interface Options extends ParentOptions {  
    ...  
    /**  
     * @displayName Region  
     */  
    region: Region<['us-east-1', 'us-east-2', 'us-west-1', 'us-west-2']>;  
}
```

## Choix d'une ou de plusieurs régions dans une liste spécifiée

```
export interface Options extends ParentOptions {
  ...
  /**
   * @displayName Regions
   */
  multiRegion: Region<['us-east-1', 'us-east-2', 'us-west-1', 'us-west-2']>[];
}
```

## Choisir une région AWS

```
export interface Options extends ParentOptions {
  ...
  /**
   * @displayName Region
   */
  region: Region<['*']>;
}
```

## Choix d'une ou de plusieurs régions dans une liste spécifiée

```
export interface Options extends ParentOptions {
  ...
  /**
   * @displayName Regions
   */
  multiRegion: Region<['us-east-1', 'us-east-2', 'us-west-1', 'us-west-2']>[];
}
```

## Composants du référentiel et du code source

Un référentiel est utilisé par Amazon CodeCatalyst pour stocker le code. Le référentiel prend un nom en entrée. La plupart des composants sont stockés dans un référentiel, tels que les fichiers de code source, les flux de travail et d'autres composants tels que les environnements de développement gérés (MDE). Le composant du référentiel source exporte également les composants utilisés pour gérer les fichiers et les actifs statiques. Les référentiels sont soumis à des contraintes de nom. Pour de plus amples informations, veuillez consulter [Référentiels de sources dans CodeCatalyst](#).

```
const repository = new SourceRepository(this, {
```

```
title: 'my-new-repository-title',
});
```

Pour importer le référentiel Amazon CodeCatalyst Blueprints et les composants du code source

Dans votre `blueprint.ts` fichier, ajoutez les éléments suivants :

```
import {...} from '@caws-blueprint-component/caws-source-repositories'
```

## Rubriques

- [Ajouter un fichier](#)
- [Ajouter un fichier générique](#)
- [Copier des fichiers](#)
- [Cibler plusieurs fichiers](#)
- [Création d'un nouveau dépôt et ajout de fichiers](#)

## Ajouter un fichier

Vous pouvez écrire un fichier texte dans un dépôt à l'aide de cette `SourceFile` construction. L'opération est l'un des cas d'utilisation les plus courants et nécessite un référentiel, un chemin de fichier et le contenu du texte. Si le chemin du fichier n'existe pas dans un référentiel, le composant crée tous les dossiers requis.

```
new SourceFile(repository, `path/to/my/file/in/repo/file.txt`, 'my file contents');
```

### Note

Si vous écrivez deux fichiers au même emplacement dans le même référentiel, l'implémentation la plus récente remplace la précédente. Vous pouvez utiliser cette fonctionnalité pour superposer le code généré, et elle est particulièrement utile pour étendre le code que les plans personnalisés ont pu générer.

## Ajouter un fichier générique

Vous pouvez écrire des bits arbitraires dans votre dépôt. Vous pouvez lire à partir d'une mémoire tampon et utiliser la `File` construction.

```
new File(repository, `path/to/my/file/in/repo/file.img`, new Buffer(...));

new File(repository, `path/to/my/file/in/repo/new-img.img`, new StaticAsset('path/to/
image.png').content());
```

## Copier des fichiers

Vous pouvez commencer avec le code généré en copiant et en collant le code de démarrage, puis en générant du code supplémentaire au-dessus de cette base. Placez le code dans le `static-assets` répertoire, puis ciblez ce code avec la `StaticAsset` construction. Dans ce cas, le chemin commence toujours à la racine du `static-assets` répertoire.

```
const starterCode = new StaticAsset('path/to/file/file.txt')
const starterCodeText = new StaticAsset('path/to/file/file.txt').toString()
const starterCodeRawContent = new StaticAsset('path/to/image/hello.png').content()

const starterCodePath = new StaticAsset('path/to/image/hello.png').path()
// starterCodePath is equal to 'path/to/image/hello.png'
```

Une sous-classe de `StaticAsset` est `SubstitutionAsset`. La sous-classe fonctionne exactement de la même manière, mais vous pouvez plutôt exécuter une substitution par moustache sur le fichier. Cela peut être utile pour générer des `copy-and-replace` styles.

La substitution d'actifs statiques utilise un moteur de création de modèles à moustache pour afficher les fichiers statiques qui sontensemencés dans le référentiel source généré. Les règles de modélisation des moustaches sont appliquées lors du rendu, ce qui signifie que toutes les valeurs sont codées en HTML par défaut. Pour afficher du code HTML non filtré, utilisez la syntaxe à triple moustache. `{{{name}}}` Pour plus d'informations, consultez les règles relatives à la modélisation des [moustaches](#).

### Note

L'exécution d'un substitut sur des fichiers qui ne sont pas interprétables par du texte peut générer des erreurs.

```
const starterCodeText = new SubstitutionAsset('path/to/file/file.txt').subsitite({
  'my_variable': 'subbed value1',
  'another_variable': 'subbed value2'
```

```
})
```

## Cibler plusieurs fichiers

Les actifs statiques prennent en charge le ciblage global grâce à une fonction statique appelée `StaticAsset` et à ses sous-classes appelées `findAll(...)`, qui renvoie une liste d'actifs statiques préchargés avec leurs chemins, leur contenu, etc. Vous pouvez enchaîner la liste avec des `File` constructions pour copier et coller le contenu dans le `static-assets` répertoire.

```
new File(repository, `path/to/my/file/in/repo/file.img`, new Buffer(...));

new File(repository, `path/to/my/file/in/repo/new-img.img`, new StaticAsset('path/to/
image.png').content());
```

## Création d'un nouveau dépôt et ajout de fichiers

Vous pouvez utiliser un composant de référentiel pour créer un nouveau référentiel dans un projet généré. Vous pouvez ensuite ajouter des fichiers ou des flux de travail au référentiel créé.

```
import { SourceRepository } from '@amazon-codecatalyst/codecatalyst-source-
repositories';
...
const repository = new SourceRepository(this, { title: 'myRepo' });
```

L'exemple suivant montre comment ajouter des fichiers et des flux de travail à un référentiel existant :

```
import { SourceFile } from '@amazon-codecatalyst/codecatalyst-source-repositories';
import { Workflow } from '@amazon-codecatalyst/codecatalyst-workflows';
...
new SourceFile(repository, 'README.md', 'This is the content of my readme');
new Workflow(this, repository, {/**...workflowDefinition...**/});
```

La combinaison des deux éléments de code génère un référentiel unique nommé `myRepo` avec un fichier source `README.md` et un CodeCatalyst flux de travail à la racine.

## Composants du flux de travail

Un flux de travail est utilisé par les CodeCatalyst projets Amazon pour exécuter des actions basées sur des déclencheurs. Vous pouvez utiliser des composants de flux de travail pour créer et assembler des fichiers YAML de flux de travail. Pour de plus amples informations, veuillez consulter [Référence de définition du flux de travail](#).

## Pour importer des composants de flux de travail Amazon CodeCatalyst Blueprints

Dans votre `blueprint.ts` fichier, ajoutez les éléments suivants :

```
import { WorkflowBuilder, Workflow } from '@amazon-codecatalyst/codecatalyst-workflows'
```

### Rubriques

- [Exemples de composants de flux de travail](#)
- [Connexion à un environnement](#)

### Exemples de composants de flux de travail

#### WorkflowBuilder composant

Vous pouvez utiliser une classe pour créer une définition de flux de travail. La définition peut être donnée à un composant de flux de travail pour le rendu dans un référentiel.

```
import { WorkflowBuilder } from '@amazon-codecatalyst/codecatalyst-workflows'

const workflowBuilder = new WorkflowBuilder({} as Blueprint, {
  Name: 'my_workflow',
});

// trigger the workflow on pushes to branch 'main'
workflowBuilder.addBranchTrigger(['main']);

// add a build action
workflowBuilder.addAction({
  // give the action a name
  actionName: 'build_and_do_some_other_stuff',

  // the action pulls from source code
  input: {
    Sources: ['WorkflowSource'],
  },

  // the output attempts to autodiscover test reports, but not in the node modules
  output: {
    AutoDiscoverReports: {
      Enabled: true,
      ReportNamePrefix: AutoDiscovered,
    }
  }
});
```

```
    IncludePaths: ['**/*'],
    ExcludePaths: ['*/node_modules/**/*'],
  },
},
// execute some arbitrary steps
steps: [
  'npm install',
  'npm run myscript',
  'echo hello-world',
],
// add an account connection to the workflow
environment: convertToWorkflowEnvironment(myEnv),
});
```

## Composant du projet Workflow

L'exemple suivant montre comment un composant Projen peut être utilisé pour écrire un flux de travail YAML dans un référentiel :

```
import { Workflow } from '@amazon-codecatalyst/codecatalyst-workflows'

...

const repo = new SourceRepository
const blueprint = this;
const workflowDef = workflowBuilder.getDefinition()

// creates a workflow.yaml at .aws/workflows/${workflowDef.name}.yaml
new Workflow(blueprint, repo, workflowDef);

// can also pass in any object and have it rendered as a yaml. This is unsafe and may
not produce a valid workflow
new Workflow(blueprint, repo, {... some object ...});
```

## Connexion à un environnement

De nombreux flux de travail doivent être exécutés via une connexion à un compte AWS. Les flux de travail gèrent cela en autorisant les actions à se connecter à des environnements avec des spécifications de nom de compte et de rôle.

```
import { convertToWorkflowEnvironment } from '@amazon-codecatalyst/codecatalyst-workflows'
```

```
const myEnv = new Environment(...);

// can be passed into a workflow constructor
const workflowEnvironment = convertToWorkflowEnvironment(myEnv);

// add a build action
workflowBuilder.addAction({
  ...
  // add an account connection to the workflow
  environment: convertToWorkflowEnvironment(myEnv),
});
```

## Composants des environnements de développement

Les environnements de développement gérés (MDE) sont utilisés pour créer et installer des espaces de travail MDE dans CodeCatalyst. Le composant génère un `devfile.yaml` fichier. Pour plus d'informations, consultez [Introduction à Devfile](#) et [Déplacer un fichier de développement d'un référentiel pour un environnement de développement](#).

```
new Workspace(this, repository, SampleWorkspaces.default);
```

Pour importer des composants d'espaces de travail Amazon CodeCatalyst Blueprints

Dans votre `blueprint.ts` fichier, ajoutez les éléments suivants :

```
import {...} from '@amazon-codecatalyst/codecatalyst-workspaces'
```

## Émet des composants

Dans CodeCatalyst, vous pouvez surveiller les fonctionnalités, les tâches, les bogues et tout autre travail impliqué dans votre projet. Chaque œuvre est conservée dans un dossier distinct appelé numéro. Chaque problème peut avoir une description, un responsable, un statut et d'autres propriétés, que vous pouvez rechercher, regrouper et filtrer. Vous pouvez consulter vos problèmes à l'aide des vues par défaut ou créer vos propres vues avec un filtrage, un tri ou un regroupement personnalisés. Pour plus d'informations sur les concepts liés aux problèmes, consultez [Concepts des problèmes](#) et [Quotas pour les numéros parus dans CodeCatalyst](#).

Le composant `issue` génère une représentation JSON d'un problème. Le composant prend en entrée un champ ID et une définition du problème.



## Pour importer des CodeCatalyst plans Amazon, émet des composants

Dans votre `blueprint.ts` fichier, ajoutez les éléments suivants :

```
import {...} from '@amazon-codecatalyst/blueprint-component.issues'
```

### Rubriques

- [Exemples de composants de problèmes](#)

### Exemples de composants de problèmes

#### Création d'un problème

```
import { Issue } from '@amazon-codecatalyst/blueprint-component.issues';
...
new Issue(this, 'myFirstIssue', {
  title: 'myFirstIssue',
  content: 'This is an example issue.',
});
```

#### Création d'un problème hautement prioritaire

```
import { Workflow } from '@amazon-codecatalyst/codecatalyst-workflows'
...
const repo = new SourceRepository
const blueprint = this;
const workflowDef = workflowBuilder.getDefinition()

// Creates a workflow.yaml at .aws/workflows/${workflowDef.name}.yaml
new Workflow(blueprint, repo, workflowDef);

// Can also pass in any object and have it rendered as a yaml. This is unsafe and may
  not produce a valid workflow
new Workflow(blueprint, repo, {... some object ...});
```

#### Création d'un problème peu prioritaire avec les étiquettes

```
import { Issue } from '@amazon-codecatalyst/blueprint-component.issues';
...
new Issue(this, 'myThirdIssue', {
```

```
title: 'myThirdIssue',
content: 'This is an example of a low priority issue with a label.',
priority: 'LOW',
labels: ['exampleLabel'],
});
```

## Utilisation de l'outillage Blueprint et de la CLI

La [CLI Blueprint](#) fournit des outils pour gérer et utiliser vos plans personnalisés.

### Rubriques

- [Utilisation de l'outillage Blueprint](#)
- [Outil de téléchargement d'images](#)

### Utilisation de l'outillage Blueprint

#### Pour travailler avec les outils du plan

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Reprenez votre environnement de développement. Pour de plus amples informations, veuillez consulter [Reprise d'un environnement de développement](#).

Si vous n'avez pas d'environnement de développement, vous devez d'abord en créer un. Pour de plus amples informations, veuillez consulter [Création d'un environnement de développement](#).

3. Dans un terminal fonctionnel, exécutez la commande suivante pour installer la CLI Blueprint :

```
npm install -g @amazon-codecatalyst/blueprint-util.cli
```

4. Dans le `blueprint.ts` fichier, importez les outils que vous souhaitez utiliser au format suivant :

```
import { <tooling-function-name> } from '@amazon-codecatalyst/blueprint-util.cli/lib/<tooling-folder-name>/<tooling-file-name>;
```

#### Tip

Vous pouvez accéder au [CodeCatalyst blueprints GitHub repository](#) pour trouver le nom de l'outillage que vous souhaitez utiliser.

Si vous souhaitez utiliser l'outil de téléchargement d'images, ajoutez ce qui suit à votre script :

```
import { uploadImagePublicly } from '@amazon-codecatalyst/blueprint-util.cli/lib/  
image-upload-tool/upload-image-to-aws';
```

## Exemples

- Si vous souhaitez utiliser la fonction de publication, ajoutez ce qui suit à votre script :

```
import { publish } from '@amazon-codecatalyst/blueprint-util.cli/lib/publish/  
publish';
```

- Si vous souhaitez utiliser l'outil de téléchargement d'images, ajoutez ce qui suit à votre script :

```
import { uploadImagePublicly } from '@amazon-codecatalyst/blueprint-util.cli/lib/  
image-upload-tool/upload-image-to-aws';
```

## 5. Appelez la fonction.

### Exemples :

- Si vous souhaitez utiliser la fonction de publication, ajoutez ce qui suit à votre script :

```
await publish(logger, config.publishEndpoint, {<your publishing options>});
```

- Si vous souhaitez utiliser l'outil de téléchargement d'images, ajoutez ce qui suit à votre script :

```
const {imageUrl, imageName} = await uploadImagePublicly(logger, 'path/to/  
image');
```

## Outil de téléchargement d'images

L'outil de téléchargement d'images vous permet de télécharger votre propre image dans un compartiment S3 de votre compte AWS, puis de diffuser cette image publiquement derrière CloudFront. L'outil prend un chemin d'image dans le stockage local (et un nom de compartiment facultatif) comme entrée, et renvoie l'URL de l'image accessible au public. Pour plus d'informations, consultez [Qu'est-ce qu'Amazon CloudFront ?](#) et [qu'est-ce qu'Amazon S3 ?](#)

Pour utiliser l'outil de téléchargement d'images

1. Clonez le [GitHub référentiel de plans open source](#) qui donne accès au SDK de plans et à des exemples de plans. Dans un terminal fonctionnel, exécutez la commande suivante :

```
git clone https://github.com/aws/codecatalyst-blueprints.git
```

2. Exécutez la commande suivante pour accéder au GitHub référentiel de plans :

```
cd codecatalyst-blueprints
```

3. Exécutez la commande suivante pour installer les dépendances :

```
yarn && yarn build
```

4. Exécutez la commande suivante pour vous assurer que la dernière version de Blueprint CLI est installée :

```
yarn upgrade @amazon-codecatalyst/blueprint-util.cli
```

5. Connectez-vous au compte AWS avec le compartiment S3 dans lequel vous souhaitez télécharger votre image. Pour plus d'informations, consultez [Configurer l'interface de ligne de commande AWS](#) et [connectez-vous via l'interface de ligne de commande AWS](#).

6. Exécutez la commande suivante depuis la racine de votre CodeCatalyst dépôt pour accéder au répertoire avec la CLI Blueprint :

```
cd packages/utils/blueprint-cli
```

7. Exécutez la commande suivante pour télécharger votre image dans un compartiment S3 :

```
yarn blueprint upload-image-public <./path/to/your/image>  
    <optional:optional-bucket-name>
```

Une URL vers votre image est générée. L'URL ne sera pas disponible immédiatement car le déploiement de la CloudFront distribution nécessite un certain temps. Vérifiez l'état de distribution pour connaître le dernier statut de déploiement. Pour plus d'informations, consultez la section [Utilisation des distributions](#).

## Tests instantanés

Les tests instantanés générés sur plusieurs configurations de votre plan sont pris en charge.

Les plans permettent de [tester des instantanés](#) sur les configurations que vous avez fournies en tant qu'auteur du plan. Les configurations sont des remplacements partiels qui sont fusionnés au-dessus du fichier `defaults.json` à la racine d'un plan. Lorsque le test instantané est activé et configuré, le processus de création et de test synthétise les configurations données et vérifie que les sorties synthétisées n'ont pas changé par rapport à l'instantané de référence. Pour consulter le code de test des instantanés, consultez le [GitHub référentiel de CodeCatalyst plans](#).

Pour activer les tests instantanés

1. Dans le `.projenrc.ts` fichier, mettez à jour l'objet d'entrée `ProjenBlueprint` avec les fichiers que vous souhaitez capturer. Par exemple :

```
{
  ....
  blueprintSnapshotConfiguration: {
    snapshotGlobs: ['**', '!environments/**', '!aws-account-to-environment/**'],
  },
}
```

2. Resynthétisez le plan pour créer des TypeScript fichiers dans votre projet de plan. Ne modifiez pas les fichiers source car ils sont gérés et régénérés par Projen. Utilisez la commande suivante :

```
yarn projen
```

3. Accédez au `src/snapshot-configurations` répertoire pour afficher le `default-config.json` fichier contenant un objet vide. Mettez à jour ou remplacez le fichier par une ou plusieurs de vos propres configurations de test. Chaque configuration de test est ensuite fusionnée avec le `default.ts.json` fichier du projet, synthétisée et comparée aux instantanés lors des tests. Utilisez la commande suivante pour tester :

```
yarn test
```

La première fois que vous utilisez une commande de test, le message suivant s'affiche :  
Snapshot Summary > NN snapshots written from 1 test suite Les tests suivants vérifient que la sortie synthétisée n'a pas changé par rapport aux instantanés et affichent le message suivant :. Snapshots: NN passed, NN total

Si vous modifiez intentionnellement votre plan pour produire une sortie différente, exécutez la commande suivante pour mettre à jour les instantanés de référence :

```
yarn test:update
```

Les instantanés s'attendent à ce que les sorties synthétisées soient constantes entre chaque exécution. Si votre plan génère des fichiers qui varient, vous devez exclure ces fichiers du test des instantanés. Mettez à jour l'`blueprintSnapshotConfiguration` objet de votre objet `ProjenBlueprint` d'entrée pour ajouter la `snapshotGlobs` propriété. La `snapshotGlobs` propriété est un tableau de [globs](#) qui détermine quels fichiers sont inclus ou exclus de la capture instantanée.

#### Note

Il existe une liste de globs par défaut. Si vous spécifiez votre propre liste, vous devrez peut-être rétablir explicitement les entrées par défaut.

## Publication d'un plan personnalisé

Les plans personnalisés fournissent des ensembles de prévisualisation à la suite d'une synthèse réussie. Le bundle de projets représente le code source, la configuration et les ressources d'un projet, et il est utilisé par les opérations d'API de CodeCatalyst déployement pour le déployement dans un projet. Si vous souhaitez continuer à développer votre plan personnalisé, relancez le processus de synthèse du plan. Pour de plus amples informations, veuillez consulter [Concepts de plans personnalisés](#).

#### Important

Si vous souhaitez utiliser des packages de plans provenant de sources externes, prenez en compte les risques qui peuvent découler de ces packages. Vous êtes responsable des plans personnalisés que vous ajoutez à votre espace et du code qu'ils génèrent.

## Rubriques

- [Affichage et publication d'une version d'aperçu d'un plan personnalisé](#)

- [Affichage et publication d'une version normale d'un plan personnalisé](#)
- [Publication et application d'un plan personnalisé dans des espaces et des projets spécifiques](#)

## Affichage et publication d'une version d'aperçu d'un plan personnalisé

Vous pouvez publier une version préliminaire de votre plan personnalisé dans votre espace si vous souhaitez l'ajouter au catalogue de plans de votre espace. Cela vous permet de visualiser le plan en tant qu'utilisateur avant d'ajouter la version non prévisualisée au catalogue. La version d'aperçu vous permet de publier sans passer à une version réelle. Par exemple, si vous travaillez sur une `0.0.1` version, vous pouvez publier et ajouter une version d'aperçu, afin que les nouvelles mises à jour d'une deuxième version puissent être publiées et ajoutées en tant que `0.0.2`.

Après avoir apporté des modifications, reconstruisez le package de votre plan personnalisé en exécutant le package `.json` fichier et prévisualisez vos modifications.

Pour afficher et publier une version d'aperçu d'un plan personnalisé

1. Reprenez votre environnement de développement. Pour de plus amples informations, veuillez consulter la page [Reprise d'un environnement de développement](#).
2. Ouvrez un terminal fonctionnel dans votre environnement de développement.
3. (Facultatif) Dans un terminal fonctionnel, installez les dépendances nécessaires à votre projet si vous ne les avez pas déjà installées. Utilisez la commande suivante :

```
yarn
```

4. (Facultatif) Si vous avez apporté des modifications au `.projenrc.ts` fichier, régénérez la configuration de votre projet avant de créer et de prévisualiser votre plan. Utilisez la commande suivante :

```
yarn projen
```

5. Reconstruisez et prévisualisez votre plan personnalisé à l'aide de la commande suivante. Utilisez la commande suivante :

```
yarn blueprint:preview
```

Accédez au `See this blueprint at:` lien fourni pour prévisualiser votre plan personnalisé. Vérifiez que l'interface utilisateur, y compris le texte, s'affiche comme prévu en fonction de

vosre configuration. Si vous souhaitez modifier votre plan personnalisé, vous pouvez modifier le `blueprint.ts` fichier, resynthétiser le plan, puis publier à nouveau une version d'aperçu. Pour de plus amples informations, veuillez consulter [Resynthèse](#).

6. (Facultatif) Vous pouvez publier une version préliminaire de votre plan personnalisé, qui peut ensuite être ajoutée au catalogue de plans de votre espace. Accédez au `Enable version [preview version number]` at : lien pour publier une version préliminaire dans votre espace.

Vous pouvez émuler la création de projets sans avoir à en créer un dans CodeCatalyst. Pour synthétiser votre projet, utilisez la commande suivante :

```
yarn blueprint:synth
```

Un plan est généré dans le `synth/synth.[options-name]/proposed-bundle/` dossier. Pour de plus amples informations, veuillez consulter [Synthèse](#).

Si vous mettez à jour votre plan personnalisé, utilisez plutôt la commande suivante pour resynthétiser votre projet :

```
yarn blueprint:resynth
```

Un plan est généré dans le `synth/synth.[options-name]/proposed-bundle/` dossier. Pour de plus amples informations, veuillez consulter [Resynthèse](#).

Après avoir publié votre version d'aperçu, vous pouvez ajouter le plan afin que les membres de l'espace puissent l'utiliser pour créer de nouveaux projets ou l'appliquer à des projets existants. Pour de plus amples informations, veuillez consulter [Ajouter un plan personnalisé à un catalogue spatial](#).

## Affichage et publication d'une version normale d'un plan personnalisé

Une fois que vous avez terminé de développer et de prévisualiser votre plan personnalisé, vous pouvez consulter et publier la nouvelle version que vous souhaitez ajouter au catalogue de plans de votre espace. Le flux de publication généré lors de la création d'un projet publie automatiquement les modifications qui sont transmises. Si vous avez désactivé la génération du flux de travail lors de la création du plan, celui-ci n'est pas automatiquement mis à disposition pour être ajouté au catalogue de plans de votre espace. Vous pouvez toujours publier votre plan personnalisé sur votre espace après avoir exécuté une `yarn` commande.

Pour afficher et publier un plan personnalisé



1. Reprenez votre environnement de développement. Pour de plus amples informations, veuillez consulter la page [Reprise d'un environnement de développement](#).
2. Ouvrez un terminal fonctionnel dans votre environnement de développement.
3. • Si vous avez choisi de ne pas générer le flux de travail de publication lors de la création de votre plan, utilisez la commande suivante :

```
yarn blueprint:release
```

Vous pouvez toujours accéder au `See this blueprint at:` lien fourni pour afficher votre plan personnalisé.

Publiez la version mise à jour de votre plan personnalisé, qui peut ensuite être ajoutée au catalogue de plans de votre espace. Accédez au `Enable version [release version number]` at : lien pour publier la dernière version sur votre espace.

- Si vous avez opté pour un flux de publication lors de la création de votre plan, la dernière version du plan est automatiquement publiée lorsque les modifications sont apportées. Utilisez les commandes suivantes :

```
git add .
```

```
git commit -m "commit message"
```

```
git push
```

Après avoir publié votre version normale, vous pouvez ajouter le plan afin que les membres de l'espace puissent l'utiliser pour créer de nouveaux projets ou l'appliquer à des projets existants. Pour de plus amples informations, veuillez consulter [Ajouter un plan personnalisé à un catalogue spatial](#).

## Publication et application d'un plan personnalisé dans des espaces et des projets spécifiques

Par défaut, les `blueprint:release` commandes `blueprint:preview` et sont publiées dans l' CodeCatalyst espace dans lequel vous avez créé le plan. Si vous disposez de plusieurs espaces Enterprise, vous pouvez également prévisualiser et publier le même plan dans ces espaces. Vous pouvez également appliquer un plan à un projet existant d'un autre espace.

## Pour publier ou appliquer un plan personnalisé dans un espace spécifié

1. Reprenez votre environnement de développement. Pour de plus amples informations, veuillez consulter [Reprise d'un environnement de développement](#).
2. Ouvrez un terminal fonctionnel dans votre environnement de développement.
3. (Facultatif) Installez les dépendances nécessaires pour votre projet si vous ne les avez pas déjà installées. Utilisez la commande suivante :

```
yarn
```

4. Utilisez la `--space` balise pour publier un aperçu ou une version normale dans un espace spécifié. Par exemple :

- ```
yarn blueprint:preview --space my-awesome-space # publishes under a "preview" version tag to 'my-awesome-space'
```

### Exemple de sortie :

```
Enable version 0.0.1-preview.0 at: https://codecatalyst.aws/spaces/my-awesome-space/blueprints
Blueprint applied to [NEW]: https://codecatalyst.aws/spaces/my-awesome-space/blueprints/%40amazon-codecatalyst%2Fmyspace.my-blueprint/publishers/1524817d-a69b-4abe-89a0-0e4a9a6c53b2/versions/0.0.1-preview.0/projects/create
```

- ```
yarn blueprint:release --space my-awesome-space # publishes normal version to 'my-awesome-space'
```

### Exemple de sortie :

```
Enable version 0.0.1 at: https://codecatalyst.aws/spaces/my-awesome-space/blueprints
Blueprint applied to [NEW]: https://codecatalyst.aws/spaces/my-awesome-space/blueprints/%40amazon-codecatalyst%2Fmyspace.my-blueprint/publishers/1524817d-a69b-4abe-89a0-0e4a9a6c53b2/versions/0.0.1/projects/create
```

Utilisez le `--project` pour appliquer une version d'aperçu d'un plan personnalisé à un projet existant dans un espace spécifié. Par exemple :

```
yarn blueprint:preview --space my-awesome-space --project my-project # previews
blueprint application to an existing project
```

Exemple de sortie :

```
Enable version 0.0.1-preview.1 at: https://codecatalyst.aws/spaces/my-awesome-
space/blueprints
Blueprint applied to [my-project]: https://codecatalyst.aws/spaces/my-awesome-
space/projects/my-project/blueprints/%40amazon-codecatalyst%2FmySpace.my-blueprint/
publishers/1524817d-a69b-4abe-89a0-0e4a9a6c53b2/versions/0.0.1-preview.1/add
```

## Afficher les détails, les versions et les projets d'un plan personnalisé

Vous pouvez consulter les plans personnalisés publiés dans votre espace, y compris les détails d'un plan, les versions et les projets créés avec ou appliquant le plan.

Rubriques

- [Afficher les plans personnalisés d'un espace](#)
- [Affichage de projets créés avec ou application d'un plan personnalisé](#)

### Afficher les plans personnalisés d'un espace

Pour consulter les plans personnalisés d'un espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la CodeCatalyst console, accédez à l'espace dans lequel vous souhaitez afficher un plan personnalisé.
3. Sur le tableau de bord spatial, choisissez l'onglet Paramètres, puis choisissez Blueprints pour afficher les plans spatiaux. Les informations suivantes sont affichées dans le tableau :
  - Nom : nom du plan personnalisé.
  - État du catalogue : indique si le plan personnalisé est publié dans le catalogue de plans de l'espace.
  - Dernière version : dernière version du plan personnalisé.
  - Dernière modification : date à laquelle le plan spatial a été mis à jour pour la dernière fois.

## Affichage de projets créés avec ou application d'un plan personnalisé

Pour afficher les projets créés avec ou appliquant un plan personnalisé

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la CodeCatalyst console, accédez à l'espace dans lequel vous souhaitez afficher un plan personnalisé.
3. Sur le tableau de bord spatial, cliquez sur l'onglet Paramètres, puis sur Blueprints.
4. Dans le tableau des plans spatiaux, choisissez le nom d'un plan personnalisé pour afficher les projets utilisant un plan et les projets n'utilisant pas de tableaux de plans.

## Ajouter et supprimer un plan personnalisé dans un espace

Une fois que vous avez publié un plan personnalisé dans votre espace, il peut être ajouté au catalogue de plans de votre espace. Un plan personnalisé peut également être supprimé du catalogue de plans de votre espace si vous ne souhaitez plus qu'il soit utilisé pour créer de nouveaux projets ou appliqué à des projets existants.

### Rubriques

- [Ajouter un plan personnalisé à un catalogue spatial](#)
- [Supprimer un plan personnalisé d'un catalogue spatial](#)

## Ajouter un plan personnalisé à un catalogue spatial

Si vous ajoutez un plan personnalisé au catalogue de plans de votre CodeCatalyst espace, tous les membres de l'espace peuvent l'utiliser lors de la création d'un projet ou de son application à un projet existant. Avant d'ajouter un plan personnalisé au catalogue de plans de l'espace, l'autorisation de publication du plan doit être activée. Si vous avez opté pour la génération de versions de flux de travail, les autorisations de publication sont activées par défaut. Pour plus d'informations, veuillez consulter [Gestion des autorisations de publication pour un plan personnalisé](#) et [Publication d'un plan personnalisé](#).

Pour ajouter un plan au catalogue de plans de l'espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Le plan ne peut être ajouté qu'à partir de la branche par défaut du référentiel source. Si vous avez développé le plan sur une branche d'entités, fusionnez votre branche d'entités avec les

modifications apportées à la branche par défaut. Créez une pull request pour fusionner les modifications apportées à la branche par défaut. Pour en savoir plus, consultez [Utilisation des pull requests sur Amazon CodeCatalyst](#).

3. Dans la CodeCatalyst console, accédez au tableau de bord spatial avec votre plan personnalisé.
4. Sur le tableau de bord spatial, cliquez sur l'onglet Paramètres, puis sur Blueprints.
5. Choisissez le nom du plan que vous souhaitez ajouter, puis choisissez Ajouter au catalogue. Si vous avez plusieurs versions, choisissez-en une dans le menu déroulant Version du catalogue
6. Choisissez Save (Enregistrer).

## Supprimer un plan personnalisé d'un catalogue spatial

### Note

Si vous supprimez un plan personnalisé d'un catalogue spatial, cela n'affecte pas le projet créé à partir du plan ou le projet qui a appliqué le plan. Les ressources du plan ne sont pas supprimées du projet.

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la CodeCatalyst console, accédez au tableau de bord spatial avec votre plan personnalisé
3. Sur le tableau de bord spatial, cliquez sur l'onglet Paramètres, puis sur Blueprints.
4. Choisissez le nom du plan que vous souhaitez supprimer, puis choisissez Supprimer le plan du catalogue.

## Gestion des autorisations de publication pour un plan personnalisé

Par défaut, l'autorisation d'un plan personnalisé est activée si une version du flux de travail a été générée lors de la création du projet. Lorsque les autorisations de publication sont activées, le plan peut être publié dans l'espace. Vous pouvez désactiver l'autorisation afin que le plan ne puisse pas être publié. Lorsque l'autorisation est désactivée, le flux de production généré lors de la création du plan ne s'exécute pas. Les nouvelles modifications apportées à un plan ne peuvent être publiées que si les autorisations du plan sont activées.

**⚠ Important**

Pour activer ou désactiver les autorisations de publication d'un projet de plan, vous devez avoir le rôle d'administrateur de l'espace.

Pour gérer les autorisations de publication d'un projet de plan

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la CodeCatalyst console, accédez à l'espace dans lequel vous souhaitez gérer les autorisations de publication d'un plan personnalisé.
3. Sur le tableau de bord spatial, cliquez sur l'onglet Paramètres, puis sur Blueprints.
4. Choisissez l'onglet Autorisations de publication du projet pour afficher les autorisations de publication pour tous les plans de votre espace.
5. Choisissez le plan que vous souhaitez gérer, puis sélectionnez Activer ou Désactiver pour modifier les autorisations de publication. Si vous activez les autorisations, passez en revue les détails de la modification des autorisations, puis choisissez Activer la publication du plan pour confirmer la modification.

## Gestion des versions pour un plan personnalisé

En tant qu'auteur de plans, vous pouvez gérer la version que vous souhaitez publier dans le catalogue de plans de l'espace. La modification de la version du catalogue d'un plan n'affecte pas les projets qui utilisent une version de plan différente.

Pour gérer une version personnalisée du plan

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la CodeCatalyst console, accédez à l'espace dans lequel vous souhaitez modifier la version d'un plan personnalisé.
3. Sur le tableau de bord spatial, cliquez sur l'onglet Paramètres, puis sur Blueprints.
4. Dans le tableau des plans spatiaux, cliquez sur le bouton radio correspondant au plan personnalisé que vous souhaitez gérer.
5. Choisissez Créer une version du catalogue, puis choisissez une version dans le menu déroulant Version du catalogue.

## 6. Choisissez Save (Enregistrer).

# Suppression d'un plan ou d'une version personnalisés publiés

Lorsque vous supprimez la version d'un plan personnalisé ou le plan lui-même de votre CodeCatalyst espace Amazon, tous vos accès aux ressources du projet de plan ou de la version du plan sont supprimés. Lorsque vous avez supprimé une version du plan ou le plan, les membres du projet ne pourront pas accéder aux ressources du projet et tous les flux de travail demandés par des référentiels de sources tiers seront arrêtés.

### Note

Si vous supprimez un plan, cela n'affecte pas le projet auquel le plan est appliqué. Les ressources du plan ne sont pas supprimées du projet.

Si une version de plan est publiée dans le catalogue de plans de l'espace, choisissez une nouvelle version pour le catalogue avant de supprimer la version publiée.

Pour supprimer la version du catalogue d'un plan personnalisé

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la CodeCatalyst console, accédez à l'espace dans lequel vous souhaitez supprimer la version du catalogue d'un plan personnalisé.
3. Sur le tableau de bord spatial, cliquez sur l'onglet Paramètres, puis sur Blueprints.
4. Choisissez le nom du plan avec la version du catalogue que vous souhaitez supprimer.
5. Cliquez sur le bouton radio correspondant à la version du catalogue que vous souhaitez supprimer, puis sélectionnez Supprimer la version.
6. Passez en revue les détails, puis choisissez une autre version du plan dans le menu déroulant Choisir une nouvelle version du catalogue de plans.
7. Entrez `delete` pour confirmer la sélection de la version du catalogue de plans.
8. Sélectionnez Supprimer.

Si une version de plan ne figure pas dans le catalogue de plans de l'espace, vous pouvez la supprimer sans en choisir une nouvelle.

## Pour supprimer une version personnalisée du plan

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la CodeCatalyst console, accédez à l'espace dans lequel vous souhaitez supprimer une version personnalisée du plan.
3. Sur le tableau de bord spatial, cliquez sur l'onglet Paramètres, puis sur Blueprints.
4. Choisissez le nom du plan avec la version que vous souhaitez supprimer.
5. Cliquez sur le bouton radio correspondant à la version que vous souhaitez supprimer, puis sélectionnez Supprimer la version.
6. Entrez `delete` pour confirmer la suppression de la version du plan.
7. Sélectionnez Supprimer.

La suppression d'un plan du catalogue de plans de l'espace entraîne la suppression de toutes les versions du plan. Les projets de l'espace qui utilisent le plan ne sont pas affectés par la suppression.

## Pour supprimer une version personnalisée du plan

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la CodeCatalyst console, accédez à l'espace dans lequel vous souhaitez supprimer un plan personnalisé.
3. Sur le tableau de bord spatial, cliquez sur l'onglet Paramètres, puis sur Blueprints.
4. Dans le tableau des plans spatiaux, cliquez sur le bouton radio correspondant au plan personnalisé que vous souhaitez supprimer, puis choisissez Supprimer le plan.
5. Entrez `delete` pour confirmer la suppression du plan personnalisé.
6. Sélectionnez Supprimer.

## Utilisation des dépendances et de l'outillage

### Rubriques

- [Ajouter des dépendances](#)
- [Gestion des incohérences entre les types de dépendance](#)
- [Utiliser du fil et du npm](#)
- [Mise à niveau de l'outillage et des composants](#)



## Ajouter des dépendances

En tant qu'auteur de plan, vous devrez peut-être ajouter des packages à votre plan, tels que `@amazon-codecatalyst/blueprint-component.environments`. Vous devez mettre à jour le `projen.ts` fichier avec ce package, puis régénérer la configuration de votre [projet avec Projen](#). Projen sert de modèle de projet pour chaque base de code de plan, ce qui permet de mettre à jour les outils rétrocompatibles en modifiant la façon dont le modèle affiche les fichiers de configuration. Le package `.json` fichier est un fichier qui appartient partiellement au modèle Projen. Projen reconnaît les versions de dépendance incluses dans le fichier `package.json`, mais les autres options doivent provenir du modèle.

Pour ajouter une dépendance et mettre à jour un **projenrc.ts** fichier

1. Dans le `projen.ts` fichier, accédez à la section `deps`.
2. Ajoutez la dépendance que vous souhaitez utiliser dans votre plan.
3. Utilisez la commande suivante pour régénérer la configuration de votre projet :

```
yarn projen && yarn
```

## Gestion des incohérences entre les types de dépendance

Après une mise à jour de [Yarn](#), l'erreur suivante peut s'afficher concernant un paramètre de dépôt :

```
Type 'SourceRepository' is missing the following properties from type  
'SourceRepository': synthesisSteps, addSynthesisStep
```

L'erreur est due à une incompatibilité de dépendance qui se produit lorsqu'un composant repose sur une version plus récente d'un autre composant, mais que le composant de base est épinglé à une version plus ancienne. L'erreur peut être corrigée en faisant en sorte que tous vos composants reposent sur la même version afin que la version soit synchronisée entre eux. Il est préférable de conserver tous les packages fournis par Blueprint sous la même dernière version (`0.0.x`), sauf si vous êtes certain de la façon dont vous gérez les versions. L'exemple suivant montre comment le `package.json` fichier peut être configuré de telle sorte que toutes les dépendances reposent sur la même version :

```
...  
"@caws-blueprint-component/caws-environments": "^0.1.12345",
```

```
"@caws-blueprint-component/caws-source-repositories": "^0.1.12345",
"@caws-blueprint-component/caws-workflows": "^0.1.12345",
"@caws-blueprint-component/caws-workspaces": "^0.1.12345",
"@caws-blueprint-util/blueprint-utils": "^0.1.12345",
...
"@caws-blueprint/blueprints.blueprint": "*",
```

Après avoir configuré les versions pour toutes les dépendances, utilisez la commande suivante :

```
yarn install
```

## Utiliser du fil et du npm

Les plans utilisent du [fil](#) pour l'outillage. L'utilisation de [npm](#) et de Yarn posera des problèmes d'outillage car la manière dont les arbres de dépendance sont résolus par chacun est différente. Pour éviter de tels problèmes, il est préférable d'utiliser uniquement du fil.

Si vous avez accidentellement installé des dépendances à l'aide de npm, vous pouvez supprimer le `package-lock.json` fichier généré et vous assurer que votre `.projenrc.ts` fichier est mis à jour avec les dépendances dont vous avez besoin. Vous régénérez la configuration de votre projet avec Projen.

Utilisez ce qui suit pour effectuer une régénération à partir du modèle :

```
yarn projen
```

Après avoir vérifié que votre fichier `.projenrc.ts` est mis à jour avec les dépendances nécessaires, utilisez la commande suivante :

```
yarn
```

## Mise à niveau de l'outillage et des composants

De temps en temps, vous souhaitez peut-être mettre à niveau votre outillage et vos composants pour intégrer les nouvelles fonctionnalités disponibles. Il est recommandé de conserver tous les composants sur la même version, sauf si vous êtes certain de la façon dont vous gérez les versions. Les versions sont synchronisées entre les composants, de sorte que les mêmes versions pour tous les composants garantissent une bonne dépendance entre eux.

## Utilisation du monorepo de l'espace de travail Yarn

Utilisez la commande suivante pour mettre à niveau les utilitaires et les composants à partir de la racine du référentiel d'un plan personnalisé :

```
yarn upgrade @amazon-codecatalyst/*
```

Utilisez la commande suivante si vous n'utilisez pas de monorepo :

```
yarn upgrade --pattern @amazon-codecatalyst/*
```

Autres options que vous pouvez utiliser pour mettre à niveau l'outillage et les composants :

- Utilisez `npm view @caws-blueprint-component/<some-component>` pour obtenir la dernière version.
- Passez manuellement à la dernière version en définissant la version dans votre fichier `package.json` et en utilisant la commande suivante : `yarn` Tous les composants et utilitaires doivent avoir la même version.

## Contribuer

Le kit de développement logiciel (SDK) Blueprints est une bibliothèque open source à laquelle vous pouvez contribuer. En tant que contributeur, tenez compte des directives de contribution, des commentaires et des défauts. Pour plus d'informations, consultez le [GitHub référentiel de plans](#).

## Quotas pour les plans dans CodeCatalyst

Le tableau suivant décrit les quotas et les limites pour les plans sur Amazon CodeCatalyst. Pour plus d'informations sur les quotas sur Amazon CodeCatalyst, consultez [Quotas pour CodeCatalyst](#).

Nombre maximum de plans appliqués par CodeCatalyst projet	100
---	-----

# Référentiels de sources dans CodeCatalyst

CodeCatalyst les référentiels sources sont des référentiels Git hébergés sur Amazon. CodeCatalyst Vous pouvez utiliser des référentiels sources CodeCatalyst pour stocker, versionner et gérer en toute sécurité les actifs d'un projet.

Les actifs d'un CodeCatalyst référentiel peuvent inclure :

- Documents
- code source
- fichiers binaires

CodeCatalyst utilise également le référentiel source d'un projet pour stocker les informations de configuration de votre projet, telles que les fichiers de configuration du flux de travail.

Vous pouvez avoir plusieurs référentiels sources dans un CodeCatalyst projet. Par exemple, vous souhaitez peut-être disposer de référentiels sources distincts pour le code source frontal, le code source principal, les utilitaires et la documentation.

Voici un flux de travail possible pour travailler avec du code dans les référentiels sources, les pull requests et les environnements de développement dans CodeCatalyst :

Mary Major crée un projet d'application Web à CodeCatalyst l'aide d'un plan, qui crée un référentiel source contenant un exemple de code. Elle invite ses amis Li Juan, Saanvi Sarkar et Jorge Souza à travailler sur le projet avec elle. Li Juan examine l'exemple de code dans le référentiel source et décide d'apporter quelques modifications rapides pour ajouter un test au code. Li crée un environnement de développement, le choisit AWS Cloud9 comme IDE et spécifie une nouvelle branche, le code de *test*. L'environnement de développement s'ouvre. Li ajoute rapidement le code, puis valide et introduit la branche contenant les modifications apportées au CodeCatalyst référentiel source. Li crée ensuite une pull request. Dans le cadre de la création de cette pull request, Li ajoute Jorge Souza et Saanvi Sarkar en tant que réviseurs pour s'assurer que le code est révisé.

Lors de la révision du code, Jorge Souza se souvient qu'il possède son propre référentiel de projets GitHub contenant un prototype de l'application sur laquelle ils travaillent. Il demande à Mary Major d'installer et de configurer l'extension qui lui permettra de lier le GitHub dépôt au projet en tant que dépôt source supplémentaire. Mary passe en revue le référentiel GitHub et travaille avec Jorge pour configurer l' GitHub extension afin qu'il puisse lier le GitHub référentiel en tant que référentiel source supplémentaire pour le projet.

CodeCatalyst les référentiels sources prennent en charge les fonctionnalités standard de Git et fonctionnent avec vos outils Git existants. Vous pouvez créer et utiliser des jetons d'accès personnels (PAT) comme mot de passe spécifique à une application lors du clonage et de l'utilisation de référentiels sources à partir d'un client Git ou d'environnements de développement intégrés (IDE). Ces PAT sont associés à votre identité CodeCatalyst d'utilisateur. Pour plus d'informations, consultez [Gestion des jetons d'accès personnels sur Amazon CodeCatalyst](#).

CodeCatalyst les référentiels sources prennent en charge les pull requests. Il s'agit d'un moyen simple pour vous et les autres membres du projet d'examiner et de commenter les modifications de code avant de les fusionner d'une branche à l'autre. Vous pouvez consulter les modifications dans la CodeCatalyst console et commenter les lignes de code.

Les push vers les branches d'un référentiel CodeCatalyst source peuvent démarrer automatiquement une exécution dans un flux de travail, où les modifications peuvent être créées, testées et déployées. Si votre référentiel source a été créé dans le cadre d'un projet à l'aide d'un modèle de projet, un ou plusieurs flux de travail sont configurés pour vous dans le cadre du projet. Vous pouvez ajouter des flux de travail supplémentaires pour les référentiels à tout moment. Les fichiers de configuration YAML pour les flux de travail d'un projet sont stockés dans les référentiels sources configurés dans l'action source pour ces flux de travail. Pour plus d'informations, voir [Commencer à utiliser les flux de travail dans CodeCatalyst](#).

## Rubriques

- [Concepts du référentiel source](#)
- [Configuration pour travailler avec les référentiels sources](#)
- [Commencer à utiliser les référentiels de CodeCatalyst sources et le plan d'application d'une seule page](#)
- [Utilisation des référentiels de sources dans CodeCatalyst](#)
- [Travailler avec des succursales sur Amazon CodeCatalyst](#)
- [Utilisation de fichiers sur Amazon CodeCatalyst](#)
- [Utilisation des pull requests sur Amazon CodeCatalyst](#)
- [Utilisation des commits sur Amazon CodeCatalyst](#)
- [Quotas pour les référentiels sources dans CodeCatalyst](#)

# Concepts du référentiel source

Voici quelques concepts à connaître lorsque vous travaillez avec des référentiels CodeCatalyst sources.

## Rubriques

- [Projets](#)
- [Référentiels de sources](#)
- [Environnements de développement](#)
- [Jetons d'accès personnels \(PAT\)](#)
- [Branches](#)
- [Branches par défaut](#)
- [Validations](#)
- [Demandes d'extraction](#)
- [Révisions](#)
- [Flux de travail](#)

## Projets

Un projet représente un effort de collaboration CodeCatalyst qui soutient les équipes et les tâches de développement. Une fois que vous avez un projet, vous pouvez ajouter, mettre à jour ou supprimer des utilisateurs et des ressources, personnaliser le tableau de bord de votre projet et suivre l'avancement du travail de votre équipe. Vous pouvez avoir plusieurs projets au sein d'un même espace.

Les référentiels sources sont spécifiques aux projets dans lesquels vous les créez ou les liez dans un espace. Vous ne pouvez pas partager un référentiel entre des projets et vous ne pouvez pas lier un référentiel à plusieurs projets dans un espace. Les utilisateurs ayant le rôle de contributeur ou d'administrateur de projet dans un projet peuvent interagir avec les référentiels sources associés à ce projet conformément aux autorisations accordées à ces rôles. Pour plus d'informations, consultez [Utilisation des rôles sur Amazon CodeCatalyst](#).

## Référentiels de sources

Un référentiel source est l'endroit où vous stockez en toute sécurité le code et les fichiers de votre projet. Il enregistre également l'historique des versions de vos fichiers. Par défaut, un référentiel

source est partagé avec les autres utilisateurs de votre CodeCatalyst projet. Vous pouvez disposer de plusieurs référentiels sources pour un projet. Vous pouvez créer des référentiels sources pour les projets dans CodeCatalyst, ou vous pouvez choisir de lier un référentiel source existant hébergé par un autre service si ce service est pris en charge par une extension installée. Par exemple, vous pouvez lier un GitHub dépôt à un projet après avoir installé l'extension GitHub Repositories. Pour plus d'informations, consultez [Utilisation des référentiels de sources dans CodeCatalyst](#) et [Démarrage rapide : utilisation de GitHub référentiels dans CodeCatalyst](#).

## Environnements de développement

Un environnement de développement est un environnement de développement basé sur le cloud que vous pouvez utiliser CodeCatalyst pour travailler rapidement sur le code stocké dans les référentiels sources de votre projet. Les outils de projet et les bibliothèques d'applications inclus dans votre environnement de développement sont définis par un fichier de développement dans le référentiel source de votre projet. Si vous n'avez pas de fichier de développement dans votre dépôt source, un fichier de développement par défaut sera automatiquement appliqué. Le fichier de développement par défaut inclut des outils pour les langages de programmation et les frameworks les plus fréquemment utilisés. Par défaut, un environnement de développement est configuré pour disposer d'un processeur à 2 cœurs, de 4 Go de RAM et de 16 Go de stockage persistant.

Vous pouvez choisir de cloner une branche existante de votre référentiel source dans votre environnement de développement, ou vous pouvez choisir de créer une nouvelle branche dans le cadre de la création de l'environnement de développement.

## Jetons d'accès personnels (PAT)

Un jeton d'accès personnel (PAT) est similaire à un mot de passe. Il est associé à votre identité d'utilisateur pour être utilisé dans tous les espaces et projets de CodeCatalyst. Vous utilisez les PAT pour accéder à CodeCatalyst des ressources qui incluent des environnements de développement intégrés (IDE) et des référentiels de sources basés sur Git. Les PAT vous représentent CodeCatalyst et vous pouvez les gérer dans vos paramètres utilisateur. Un utilisateur peut avoir plusieurs PAT. Les jetons d'accès personnels ne s'affichent qu'une seule fois. Il est recommandé de les stocker en toute sécurité sur votre ordinateur local. Par défaut, les PAT expirent au bout d'un an.

Lorsque vous travaillez avec des environnements de développement intégrés (IDE), les PAT sont l'équivalent d'un mot de passe Git. Fournissez le PAT lorsqu'on vous demande un mot de passe lors de la configuration de votre IDE pour qu'il fonctionne avec un dépôt Git. Pour plus d'informations sur la façon de connecter votre IDE à un référentiel basé sur Git, consultez la documentation de votre IDE.

## Branches

Une branche est un pointeur ou une référence vers un commit dans Git et dans CodeCatalyst. Vous pouvez utiliser des branches pour organiser votre travail. Par exemple, vous pouvez utiliser des branches pour travailler sur une version nouvelle ou différente de fichiers sans affecter les fichiers des autres branches. Vous pouvez utiliser des branches pour développer de nouvelles fonctionnalités, stocker une version spécifique de votre projet, etc. Un dépôt source peut comporter une ou plusieurs branches. Lorsque vous créez un projet à l'aide d'un modèle, le référentiel source créé pour le projet contient des exemples de fichiers dans une branche appelée main. La branche principale est la branche par défaut du référentiel.

### Branches par défaut

Les référentiels source CodeCatalyst ont une branche par défaut, quelle que soit la manière dont vous les créez. Si vous choisissez de créer un projet à l'aide d'un modèle, le référentiel source créé pour ce projet inclut un fichier README.md en plus des exemples de code, des définitions de flux de travail et d'autres ressources. Si vous créez un référentiel source sans utiliser de modèle, un fichier README.md est ajouté pour vous lors de la première validation et une branche par défaut est créée pour vous dans le cadre de la création du référentiel. Cette branche par défaut est nommée main. Cette branche par défaut est celle utilisée comme branche de base ou par défaut dans les référentiels locaux (dépôts) lorsque les utilisateurs clonent le référentiel. Vous pouvez modifier la branche utilisée comme branche par défaut. Pour plus d'informations, consultez [Afficher et modifier la branche par défaut d'un référentiel](#).

Vous ne pouvez pas supprimer la branche par défaut d'un dépôt source. Les résultats de recherche incluent uniquement les résultats de la branche par défaut.

## Validations

Un commit est une modification apportée à un fichier ou à un ensemble de fichiers. Dans la CodeCatalyst console Amazon, un commit enregistre vos modifications et les envoie vers un référentiel source. Le commit inclut des informations sur le changement, notamment l'identité de l'utilisateur qui a effectué le changement, l'heure et la date du changement, le titre du commit et tout message inclus concernant le changement. Pour plus d'informations, consultez [Utilisation des commits sur Amazon CodeCatalyst](#).

Dans le contexte d'un dépôt source en CodeCatalyst, les validations sont des instantanés du contenu et des modifications apportées au contenu de votre dépôt. Vous pouvez également ajouter des balises Git aux validations, afin d'identifier des validations spécifiques.



## Demandes d'extraction

Une pull request est le principal moyen par lequel vous et les autres utilisateurs examinez, commentez et fusionnez les modifications de code d'une branche à l'autre dans un référentiel source. Vous pouvez utiliser des pull requests pour examiner les modifications de code de manière collaborative afin de détecter des modifications ou des correctifs mineurs, des ajouts de fonctionnalités majeures ou de nouvelles versions de vos logiciels publiés. Dans une pull request, vous pouvez examiner les modifications entre les branches source et de destination ou les différences entre les révisions de ces branches. Vous pouvez ajouter des commentaires à des lignes de code individuelles ainsi que des commentaires sur la pull request dans son ensemble.

### Tip

Lorsque vous créez une pull request, la différence affichée est la différence entre l'extrémité de la branche source et la pointe de la branche de destination. Une fois la pull request créée, la différence affichée se situera entre la révision de la pull request que vous avez choisie et le commit qui figurait au début de la branche de destination lorsque vous avez créé la pull request. Pour plus d'informations sur les différences et les bases de fusion dans Git, consultez [git-merge-base](#) la documentation Git.

## Révisions

Une révision est une version mise à jour d'une pull request. Chaque envoi vers la branche source d'une pull request crée une révision qui contient les modifications apportées aux validations incluses dans ce push. Vous pouvez voir les différences entre les révisions d'une pull request en plus des différences entre les branches source et de destination. Pour plus d'informations, consultez [Utilisation des pull requests sur Amazon CodeCatalyst](#).

## Flux de travail

Un flux de travail est une procédure automatisée qui décrit comment créer, tester et déployer votre code dans le cadre d'un système d'intégration et de livraison continues (CI/CD). Un flux de travail définit une série d'étapes, ou d'actions, à effectuer lors de l'exécution d'un flux de travail. Un flux de travail définit également les événements, ou déclencheurs, qui déclenchent le démarrage du flux de travail. Pour configurer un flux de travail, vous devez créer un fichier de définition de flux de travail à l'aide de [l'éditeur visuel ou YAML](#) de la CodeCatalyst console.

**i** Tip

Pour un aperçu rapide de la manière dont vous pouvez utiliser les flux de travail dans un projet, [créez un projet avec un plan](#). Chaque plan déploie un flux de travail fonctionnel que vous pouvez examiner, exécuter et tester.

Un référentiel source peut également stocker les fichiers de configuration et d'autres informations relatives aux flux de travail, aux notifications, aux problèmes et aux autres informations de configuration du projet. Les fichiers de configuration sont créés et stockés dans le référentiel source lorsque vous créez des ressources qui nécessitent des fichiers de configuration ou lorsque vous spécifiez le référentiel comme action source pour un flux de travail. Si vous créez un projet à partir d'un plan, les fichiers de configuration seront déjà stockés dans le référentiel source créé pour vous dans le cadre du projet. Ces informations de configuration sont stockées dans un dossier nommé `.codecatalyst` dans la branche par défaut de votre référentiel. Chaque fois que vous créez une branche de la branche par défaut, vous créez une copie de ce dossier et de sa configuration en plus de tous les autres fichiers et dossiers de cette branche.

## Configuration pour travailler avec les référentiels sources

Lorsque vous travaillez avec des référentiels de sources dans Amazon CodeCatalyst sur votre machine locale, vous pouvez utiliser Git seul ou dans un environnement de développement intégré (IDE) pris en charge pour apporter des modifications au code et envoyer et extraire votre code. En tant que bonne pratique, nous vous recommandons d'utiliser les dernières versions de Git et des autres logiciels.

**i** Note

Si vous utilisez des environnements de développement, il n'est pas nécessaire d'installer Git. Une version récente de Git est incluse dans votre environnement de développement.

### Informations de compatibilité des versions pour CodeCatalyst

Composant	Version
Git	dernières

## Installez Git

Pour travailler avec des fichiers, des validations, des branches et d'autres informations dans des référentiels sources à partir d'un client Git sans IDE, installez Git sur votre machine locale.

Pour installer Git, nous recommandons des sites Web tels que [Git Downloads](#).

## Créez un jeton d'accès personnel

Pour cloner des référentiels sources sur votre machine locale ou sur votre IDE préféré, vous devez créer un jeton d'accès personnel (PAT).

Pour créer un jeton d'accès personnel (PAT)

1. Dans la barre de menu supérieure, choisissez votre badge de profil, puis sélectionnez Mes paramètres.

### Tip

Vous pouvez également accéder à votre profil utilisateur en vous rendant sur la page des membres d'un projet ou d'un espace et en choisissant votre nom dans la liste des membres.

2. Dans Nom du PAT, entrez un nom descriptif pour votre PAT.
3. Dans Date d'expiration, laissez la date par défaut ou cliquez sur l'icône du calendrier pour sélectionner une date personnalisée. La date d'expiration par défaut est d'un an à compter de la date actuelle.
4. Choisissez Créer.

Vous pouvez également créer ce jeton lorsque vous choisissez Clone un référentiel pour un référentiel source.

5. Enregistrez le secret PAT dans un emplacement sécurisé.

### Important

Le secret PAT ne s'affiche qu'une seule fois. Vous ne pouvez pas le récupérer après avoir fermé la fenêtre.

# Commencer à utiliser les référentiels de CodeCatalyst sources et le plan d'application d'une seule page

Suivez les étapes de ce didacticiel pour apprendre à utiliser les référentiels de sources sur Amazon CodeCatalyst.

Le moyen le plus rapide de commencer à travailler avec les référentiels de sources sur Amazon CodeCatalyst est de créer un projet à l'aide d'un modèle. Lorsque vous créez un projet à l'aide d'un modèle, des ressources sont créées pour vous, notamment un référentiel de sources contenant un exemple de code. Vous pouvez utiliser ce référentiel et cet exemple de code pour apprendre à :

- Afficher les référentiels sources d'un projet et parcourir leur contenu
- Créez un environnement de développement avec une nouvelle branche dans laquelle vous pouvez travailler sur le code
- Modifiez un fichier, validez et publiez vos modifications
- Créez une pull request et passez en revue les modifications apportées au code avec les autres membres du projet
- Consultez le flux de travail de votre projet : créez et testez automatiquement les modifications dans la branche source de la pull request
- Fusionnez vos modifications de votre branche source dans la branche de destination et fermez la pull request
- Découvrez les modifications fusionnées créées et déployées automatiquement

Pour tirer le meilleur parti de ce didacticiel, invitez d'autres personnes à rejoindre votre projet afin de pouvoir travailler ensemble sur une pull request. Vous pouvez également explorer des fonctionnalités supplémentaires dans CodeCatalyst, telles que la création de problèmes et leur association à une pull request, ou la configuration de notifications et la réception d'alertes lorsque le flux de travail associé s'exécute. Pour une exploration complète de CodeCatalyst, voir [Tutoriels de mise en route](#).

## Création d'un projet à l'aide d'un plan

La création d'un projet est la première étape pour pouvoir travailler ensemble. Vous pouvez utiliser un plan pour créer votre projet, qui créera également un référentiel source avec un exemple de code et un flux de travail qui créera et déploiera automatiquement votre code lorsque vous le modifiez. Dans ce didacticiel, nous allons vous présenter un projet créé à l'aide du plan d'application d'une seule page, mais vous pouvez suivre les procédures de tout projet

comportant un référentiel source. Assurez-vous de choisir un rôle IAM ou d'ajouter un rôle IAM si vous n'en avez pas un lors de la création du projet. Nous vous recommandons d'utiliser le rôle CodeCatalystWorkflowDevelopmentRole-*spaceName* de service pour ce projet.

Si vous avez déjà un projet, vous pouvez passer à [Afficher les référentiels d'un projet](#).

#### Note

Seuls les utilisateurs dotés du rôle d'administrateur de l'espace ou d'utilisateur avancé peuvent créer des projets dans CodeCatalyst. Si vous n'avez pas ce rôle et que vous avez besoin d'un projet sur lequel travailler pour ce didacticiel, demandez à une personne possédant l'un de ces rôles de créer un projet pour vous et de vous ajouter au projet créé. Pour plus d'informations, consultez [Utilisation des rôles sur Amazon CodeCatalyst](#).

Pour créer un projet à l'aide d'un plan

1. Dans la CodeCatalyst console, accédez à l'espace dans lequel vous souhaitez créer un projet.
2. Sur le tableau de bord de l'espace, choisissez Créer un projet.
3. Choisissez Commencer par un plan.
4. Choisissez un plan, puis cliquez sur Suivant.
5. Sous Donnez un nom à votre projet, entrez le nom que vous souhaitez attribuer à votre projet et les noms de ressources associés. Le nom doit être unique dans votre espace.
6. Sous Ressources du projet, configurez les paramètres communs du projet.
7. (Facultatif) Pour afficher les fichiers de définition mis à jour en fonction des paramètres de projet que vous avez sélectionnés, choisissez Afficher le code ou Afficher le flux de travail dans Générer un aperçu du projet.
8. (Facultatif) Choisissez Afficher les détails dans la fiche du plan pour afficher des détails spécifiques sur le plan, tels qu'un aperçu de l'architecture du plan, les connexions et autorisations requises, ainsi que le type de ressources créé par le plan.
9. Sélectionnez Create a project (Créer un projet).

Pour plus d'informations sur les plans de projet, consultez [Référence du plan de projet](#).

La page d'aperçu du projet s'ouvre dès que vous créez un projet ou que vous acceptez une invitation à participer à un projet et que vous terminez le processus de connexion. La page de présentation

d'un nouveau projet ne contient aucun problème en suspens ni aucune pull request. Vous pouvez éventuellement choisir de créer un problème et de vous l'attribuer. Vous pouvez également choisir d'inviter quelqu'un d'autre à rejoindre votre projet. Pour plus d'informations, consultez [Création d'un problème dans CodeCatalyst](#) et [Inviter un utilisateur à rejoindre votre projet](#).

## Afficher les référentiels d'un projet

En tant que membre d'un projet, vous pouvez consulter les référentiels sources du projet. Vous pouvez également choisir de créer des référentiels supplémentaires. Si une personne ayant le rôle d'administrateur d'espace a installé et configuré l'extension de GitHub référentiels, vous pouvez également ajouter des liens vers des GitHub référentiels dans les GitHub comptes configurés pour l'extension. Pour plus d'informations, consultez [Création d'un référentiel source](#) et [Démarrage rapide : utilisation de GitHub référentiels dans CodeCatalyst](#).

### Note

***Pour les projets créés avec le plan d'application d'une seule page, le nom par défaut du référentiel source contenant l'exemple de code est spa-app.***

Pour accéder aux référentiels sources d'un projet

1. Accédez à votre projet, puis effectuez l'une des opérations suivantes :
  - Sur la page de résumé de votre projet, choisissez le référentiel souhaité dans la liste, puis choisissez Afficher le référentiel.
  - Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources. Dans Référentiels sources, choisissez le nom du référentiel dans la liste. Vous pouvez filtrer la liste des référentiels en saisissant une partie du nom du référentiel dans la barre de filtre.
2. Sur la page d'accueil du référentiel, consultez le contenu du référentiel et les informations sur les ressources associées, telles que le nombre de pull requests et les flux de travail. Par défaut, le contenu de la branche par défaut est affiché. Vous pouvez modifier l'affichage en choisissant une autre branche dans la liste déroulante.

La page de présentation du référentiel inclut des informations sur les flux de travail et les pull requests configurés pour les branches de ce référentiel et ses fichiers. Si vous venez de créer le projet, les flux de travail initiaux pour créer, tester et déployer le code seront toujours en cours

d'exécution, car leur exécution prend quelques minutes. Vous pouvez consulter les flux de travail associés et leur statut en choisissant le numéro sous Flux de travail associés, mais cela ouvre la page Flux de travail dans CI/CD. Pour ce didacticiel, restez sur la page de présentation et explorez le code du référentiel. Le contenu du README .md fichier est affiché sur cette page, sous les fichiers du référentiel. Dans Fichiers, le contenu de la branche par défaut est affiché. Vous pouvez modifier l'affichage du fichier pour afficher le contenu d'une autre branche si vous en avez une. Le .codecatalyst dossier contient le code utilisé pour d'autres parties du projet, tels que les fichiers YAML du flux de travail.

Pour afficher le contenu des dossiers, cliquez sur la flèche située à côté du nom du dossier pour le développer. Par exemple, cliquez sur la flèche à côté src pour afficher les fichiers de l'application Web d'une seule page contenue dans ce dossier. Pour afficher le contenu d'un fichier, sélectionnez ce dernier dans la liste. Cela ouvre la fenêtre Afficher les fichiers, dans laquelle vous pouvez parcourir le contenu de plusieurs fichiers. Vous pouvez également modifier des fichiers uniques dans la console, mais pour modifier plusieurs fichiers, vous devez créer un environnement de développement.

## Création d'un environnement de développement

Vous pouvez ajouter et modifier des fichiers dans un référentiel source dans la CodeCatalyst console Amazon. Toutefois, pour travailler efficacement avec plusieurs fichiers et branches, nous vous recommandons d'utiliser un environnement de développement ou de cloner le référentiel sur votre ordinateur local. Dans ce didacticiel, nous allons créer un AWS Cloud9 environnement de développement avec une branche nommée **develop**. Vous pouvez choisir un autre nom de branche, mais en nommant la branche **develop**, un flux de travail s'exécutera automatiquement pour générer et tester votre code lorsque vous créerez une pull request plus loin dans ce didacticiel.

### Tip

Si vous décidez de cloner un dépôt localement au lieu ou en plus d'utiliser un environnement de développement, assurez-vous que Git est installé sur votre ordinateur local ou que votre IDE inclut Git. Pour plus d'informations, consultez [Configuration pour travailler avec les référentiels sources](#).

Pour créer un environnement de développement avec une nouvelle branche

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).

2. Accédez au projet dans lequel vous souhaitez créer un environnement de développement.
3. Choisissez le nom du référentiel dans la liste des référentiels sources du projet. Dans le volet de navigation, vous pouvez également choisir Code, choisir Référentiels sources et choisir le référentiel pour lequel vous souhaitez créer un environnement de développement.
4. Sur la page d'accueil du référentiel, choisissez Create Dev Environment.
5. Choisissez un IDE compatible dans le menu déroulant. Pour plus d'informations, consultez [Environnements de développement intégrés pris en charge pour les environnements de développement](#).
6. Choisissez le référentiel à cloner, choisissez Travailler dans une nouvelle branche, entrez le nom de la branche dans le champ Nom de la branche et choisissez une branche à partir de laquelle créer la nouvelle branche dans le menu déroulant Créer une branche depuis.
7. Ajoutez éventuellement un alias pour l'environnement de développement.
8. Vous pouvez éventuellement cliquer sur le bouton d'édition de la configuration de l'environnement de développement pour modifier la configuration de calcul, de stockage ou de temporisation de l'environnement de développement.
9. Choisissez Créer. Pendant la création de votre environnement de développement, la colonne d'état de l'environnement de développement affichera Démarrage, et la colonne d'état affichera En cours d'exécution une fois l'environnement de développement créé. Un nouvel onglet s'ouvre avec votre environnement de développement dans l'IDE de votre choix. Vous pouvez modifier le code, valider et appliquer vos modifications.

Une fois que vous avez créé l'environnement de développement, vous pouvez modifier des fichiers, valider vos modifications et les transférer vers la **test** branche. Pour ce didacticiel, modifiez le contenu entre les `<p>` balises du `App.tsx` fichier du `src` dossier afin de modifier le texte affiché sur la page Web. Validez et validez votre modification, puis revenez à l' CodeCatalyst onglet.

Pour apporter et appliquer un changement à partir d'un environnement AWS Cloud9 de développement

1. Dans AWS Cloud9, développez le menu de navigation latéral pour parcourir les fichiers. `src`Développez et ouvrez `App.tsx`.
2. Modifiez le texte à l'intérieur des `<p>` balises.
3. Enregistrez le fichier, puis validez et publiez vos modifications à l'aide du menu Git. Vous pouvez également, dans la fenêtre du terminal, valider et appliquer vos modifications à l'aide `git commit` des `git push` commandes et.



```
git commit -am "Making an example change"  
git push
```

**i** Tip

Vous devrez peut-être remplacer les répertoires du terminal par le répertoire du référentiel Git avant de pouvoir exécuter correctement les commandes Git.

## Création d'une demande d'extraction

Vous pouvez utiliser des pull requests pour examiner les modifications de code de manière collaborative afin de détecter des modifications ou des correctifs mineurs, des ajouts de fonctionnalités majeures ou de nouvelles versions de vos logiciels publiés. Dans ce didacticiel, vous allez créer une pull request pour examiner les modifications que vous avez apportées à la branche de *test* par rapport à la branche principale. La création d'une pull request dans un projet créé à l'aide d'un modèle lancera également l'exécution des flux de travail associés, le cas échéant.

Pour créer une pull request

1. Accédez à votre projet.
2. Effectuez l'une des actions suivantes :
  - Dans le volet de navigation, choisissez Code, choisissez Pull requests, puis Create pull request.
  - Sur la page d'accueil du référentiel, choisissez Plus, puis Create pull request.
  - Sur la page du projet, choisissez Create pull request.
3. Dans le référentiel source, assurez-vous que le référentiel source spécifié est celui qui contient le code validé. Cette option n'apparaît que si vous n'avez pas créé la pull request depuis la page principale du dépôt.
4. Dans Branche de destination, choisissez la branche dans laquelle vous souhaitez fusionner le code une fois celui-ci révisé.
5. Dans Branche source, choisissez la branche qui contient le code validé.
6. Dans Titre de la demande d'extraction, entrez un titre qui aide les autres utilisateurs à comprendre ce qui doit être revu et pourquoi.


7. (Facultatif) Dans la description de la Pull request, fournissez des informations telles qu'un lien vers les problèmes ou une description de vos modifications.

 Tip

Vous pouvez choisir Write description for me afin de générer CodeCatalyst automatiquement une description des modifications contenues dans la pull request. Vous pouvez apporter des modifications à la description générée automatiquement après l'avoir ajoutée à la pull request.

Cette fonctionnalité nécessite que les fonctionnalités d'IA génératives soient activées pour l'espace. Pour plus d'informations, consultez [la section Gestion des fonctionnalités d'IA générative](#).

8. (Facultatif) Dans Problèmes, choisissez Lier les problèmes, puis choisissez un problème dans la liste ou entrez son ID. Pour dissocier un problème, cliquez sur l'icône de dissociation.
9. (Facultatif) Dans Réviseurs obligatoires, sélectionnez Ajouter les réviseurs requis. Choisissez parmi la liste des membres du projet pour les ajouter. Les réviseurs requis doivent approuver les modifications avant que la pull request puisse être fusionnée dans la branche de destination.

 Note

Vous ne pouvez pas ajouter un réviseur à la fois en tant que réviseur obligatoire et en tant que réviseur facultatif. Vous ne pouvez pas vous ajouter en tant que réviseur.

10. (Facultatif) Dans Réviseurs facultatifs, sélectionnez Ajouter des réviseurs facultatifs. Choisissez parmi la liste des membres du projet pour les ajouter. Les réviseurs facultatifs n'ont pas à approuver les modifications comme condition préalable pour que la pull request puisse être fusionnée dans la branche de destination.
11. Passez en revue les différences entre les branches. La différence affichée dans une pull request réside dans les modifications entre la révision dans la branche source et la base de fusion, qui est le commit principal de la branche de destination au moment de la création de la pull request. Si aucune modification ne s'affiche, il se peut que les branches soient identiques ou que vous ayez choisi la même branche pour la source et pour la destination.
12. Lorsque vous êtes certain que la pull request contient le code et les modifications que vous souhaitez vérifier, choisissez Create.

**Note**

Après avoir créé la pull request, vous pouvez ajouter des commentaires. Des commentaires peuvent être ajoutés à la pull request ou à des lignes individuelles des fichiers, ainsi qu'à la pull request globale. Vous pouvez ajouter des liens vers des ressources, telles que des fichiers, en utilisant le signe @ suivi du nom du fichier.

Vous pouvez consulter les informations relatives aux flux de travail associés lancés par la création de cette pull request en choisissant Vue d'ensemble, puis en consultant les informations dans la zone des détails de la demande d'extraction sous Exécutions du flux de travail. Pour afficher le flux de travail exécuté, choisissez l'exécution.

**Tip**

Si vous avez donné un autre nom à votre branche **develop**, aucun flux de travail ne s'exécutera automatiquement pour créer et tester vos modifications. Si vous souhaitez configurer cela, modifiez le fichier YAML du `onPullRequestBuildAndTest` flux de travail. Pour plus d'informations, consultez [Création, modification et suppression d'un flux de travail](#).

Vous pouvez commenter cette pull request et demander aux autres membres du projet de le faire. Vous pouvez également choisir d'ajouter ou de modifier des réviseurs facultatifs ou obligatoires. Vous pouvez choisir d'apporter d'autres modifications à la branche source du référentiel et voir comment ces modifications validées créent des révisions pour la pull request. Pour plus d'informations, reportez-vous aux [Mettre à jour une pull request](#) sections [Révision d'une pull request](#), [Utilisation des pull requests sur Amazon CodeCatalyst](#), et [Afficher le statut et les détails de l'exécution du flux de travail](#).

## Fusion d'une pull request

Une fois qu'une pull request a été examinée et qu'elle a reçu les approbations des réviseurs requis, vous pouvez fusionner sa branche source avec la branche de destination dans la CodeCatalyst console. La fusion d'une pull request lancera également une série de modifications via tous les flux de travail associés à la branche de destination. Dans ce didacticiel, vous allez fusionner la branche de test avec la branche principale, ce qui lancera une exécution du `onPushToMainDeployPipeline` flux de travail.

## Pour fusionner une pull request (console)

1. Dans Pull requests, choisissez la pull request que vous avez créée à l'étape précédente. Dans la pull request, choisissez Merge.
2. Choisissez parmi les stratégies de fusion disponibles pour la pull request. Sélectionnez ou désélectionnez éventuellement l'option permettant de supprimer la branche source après avoir fusionné la pull request, puis choisissez Merge. Une fois la fusion terminée, le statut de la pull request passe à Merged et n'apparaît plus dans la vue par défaut des pull requests. La vue par défaut affiche les pull requests dont le statut est Ouvert. Vous pouvez toujours consulter une pull request fusionnée, mais vous ne pouvez pas l'approuver ni modifier son statut.

### Note

Si le bouton Fusionner n'est pas actif ou si le libellé Non fusionnable s'affiche, cela signifie que le réviseur requis n'a pas encore approuvé la pull request ou que la pull request ne peut pas être fusionnée dans la CodeCatalyst console. Un réviseur qui n'a pas approuvé une pull request est indiqué par une icône en forme d'horloge dans la section Vue d'ensemble, dans la zone des détails de la pull request. Si tous les réviseurs requis ont approuvé la pull request mais que le bouton Fusionner n'est toujours pas actif, il se peut que vous rencontriez un conflit de fusion ou que vous ayez dépassé le quota de stockage pour l'espace. Vous pouvez résoudre les conflits de fusion pour la branche de destination dans un environnement de développement, appliquer les modifications, puis fusionner la pull request, ou vous pouvez résoudre les conflits et fusionner localement, puis transférer le commit contenant la fusion vers CodeCatalyst. Pour plus d'informations, consultez [Fusion d'une pull request \(Git\)](#) votre documentation Git.

## Afficher le code déployé

Il est maintenant temps de consulter le code initialement déployé qui se trouvait dans la branche par défaut, ainsi que les modifications que vous avez fusionnées une fois qu'elles ont été automatiquement créées, testées et déployées. Pour ce faire, vous pouvez revenir à la page d'aperçu du référentiel et choisir le numéro à côté de l'icône des flux de travail associés, ou dans le volet de navigation, choisir CI/CD, puis choisir Workflows.

## Pour afficher le code déployé

1. Dans Workflows, dans `PushToMainDeployPipeline`, développez `Recent runs`.

### Note

Il s'agit du nom par défaut du flux de travail pour les projets créés avec le plan d'application d'une seule page.

2. L'exécution la plus récente est celle lancée par la validation de votre pull request fusionnée dans la `main` branche et affichera probablement le statut `En cours`. Choisissez une exécution terminée avec succès dans la liste pour afficher les détails de cette exécution.
3. Choisissez `Variables`. Copiez la valeur de `AppURL`. Il s'agit de l'URL de l'application Web à page unique déployée. Ouvrez un nouvel onglet de navigateur et collez la valeur pour afficher le code créé et déployé. Laissez l'onglet ouvert.
4. Revenez à la liste des exécutions du flux de travail et attendez que l'exécution la plus récente soit terminée. Lorsque c'est le cas, retournez à l'onglet que vous avez ouvert pour afficher l'application Web et actualiser votre navigateur. Vous devriez voir les modifications que vous avez apportées dans votre pull request fusionnée.

## Nettoyage des ressources

Une fois que vous aurez exploré l'utilisation d'un référentiel source et d'une pull request, vous souhaitez peut-être supprimer les ressources dont vous n'avez pas besoin. Vous ne pouvez pas supprimer les pull requests, mais vous pouvez les fermer. Vous pouvez supprimer toutes les branches que vous avez créées.

Si vous n'avez plus besoin du référentiel source ou du projet, vous pouvez également supprimer ces ressources. Pour plus d'informations, consultez [Supprimer un référentiel source](#) et [Supprimer un projet sur Amazon CodeCatalyst](#).

## Utilisation des référentiels de sources dans CodeCatalyst

Un référentiel source est l'endroit où vous stockez en toute sécurité le code et les fichiers de votre projet. Il stocke également l'historique de vos sources, depuis le premier commit jusqu'aux dernières modifications. Si vous choisissez un plan qui inclut un référentiel source, ce référentiel contient également les fichiers de configuration et d'autres informations relatives aux flux de travail et aux

notifications du projet. Ces informations de configuration sont stockées dans un dossier nommé `.codecatalyst`.

Vous pouvez créer un référentiel source CodeCatalyst soit en créant un projet avec un plan qui crée un référentiel source dans le cadre de la création d'un projet, soit en créant un référentiel source dans un projet existant. Les utilisateurs du projet verront et pourront utiliser automatiquement les référentiels que vous créez pour un projet. Vous pouvez également choisir de lier un dépôt Git hébergé sur GitHub votre projet. Lorsque vous le faites, les utilisateurs de votre projet peuvent consulter et accéder à ce référentiel lié dans la liste des référentiels du projet.

#### Note

Avant de lier le référentiel, vous devez installer l'extension pour le service qui l'héberge. Vous ne pouvez pas lier un dépôt archivé. Bien que vous puissiez lier un dépôt vide, vous ne pouvez pas l'utiliser CodeCatalyst tant que vous ne l'avez pas initialisé avec un commit initial qui crée une branche par défaut. Pour plus d'informations, consultez [Installation d'une extension](#).

Par défaut, un référentiel source est partagé avec les autres membres de votre CodeCatalyst projet Amazon. Vous pouvez créer des référentiels de sources supplémentaires pour un projet ou lier des référentiels au projet. Tous les membres d'un projet peuvent afficher, ajouter, modifier et supprimer des fichiers et des dossiers dans les référentiels sources du projet.

Pour travailler rapidement sur le code d'un référentiel source, vous pouvez créer un environnement de développement qui clone un référentiel spécifique et y ajouter des branches dans lequel vous pouvez travailler sur le code dans l'environnement de développement intégré (IDE) que vous avez choisi pour l'environnement de développement. Vous pouvez cloner un dépôt source sur votre ordinateur local et extraire et transférer les modifications entre votre dépôt local et le dépôt distant. CodeCatalyst Vous pouvez également travailler avec les référentiels sources en configurant l'accès à ceux-ci dans votre IDE préféré, à condition que cet IDE prenne en charge la gestion des informations d'identification.

Les noms des référentiels doivent être uniques au sein d'un CodeCatalyst projet.

#### Rubriques

- [Création d'un référentiel source](#)
- [Lier un référentiel source](#)

- [Affichage d'un référentiel source](#)
- [Modification des paramètres d'un référentiel source](#)
- [Clonage d'un dépôt source](#)
- [Supprimer un référentiel source](#)

## Création d'un référentiel source

Lorsque vous créez un projet à l'aide d'un plan sur Amazon CodeCatalyst, vous CodeCatalyst créez un référentiel de sources. Ce référentiel source contient un exemple de code en plus des informations de configuration pour les flux de travail et les autres ressources créées pour vous. Il s'agit de la méthode recommandée pour commencer à utiliser les référentiels dans CodeCatalyst. Vous pouvez choisir de créer des référentiels pour un projet. Ces référentiels contiendront un seul fichier, un fichier README.md que vous pouvez modifier ou supprimer à tout moment. En fonction de vos choix lors de la création d'un référentiel source, les référentiels peuvent également contenir un `.gitignore` fichier.

Pour créer un référentiel de sources

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre projet.
3. Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources.
4. Choisissez Ajouter un référentiel, puis sélectionnez Créer un référentiel.
5. Dans Nom du référentiel, saisissez un nom pour le référentiel. Dans ce guide, nous utilisons un autre nom *codecatalyst-source-repository*, mais vous pouvez en choisir un autre. Les noms des référentiels doivent être uniques au sein d'un projet. Pour plus d'informations sur les exigences relatives aux noms de référentiels, consultez [Quotas pour les référentiels sources dans CodeCatalyst](#).
6. (Facultatif) Dans Description, ajoutez une description du référentiel afin d'aider les autres utilisateurs du projet à comprendre à quoi sert le référentiel.
7. (Facultatif) Ajoutez un `.gitignore` fichier correspondant au type de code que vous souhaitez envoyer.
8. Choisissez Créer.

**Note**

CodeCatalyst ajoute un README .md fichier à votre dépôt lorsque vous le créez. CodeCatalyst crée également un commit initial pour le dépôt dans une branche par défaut nommée main. Vous pouvez modifier ou supprimer le fichier README.md, mais vous ne pouvez pas modifier ou supprimer la branche par défaut.

## Lier un référentiel source

Lorsque vous liez un référentiel source à un projet, vous pouvez inclure des référentiels dotés d'une CodeCatalyst extension pour le service hébergeant le référentiel, si cette extension est installée pour votre espace. Seuls les utilisateurs dotés du rôle d'administrateur de l'espace peuvent installer des extensions. Une fois l'extension installée, vous pouvez créer un lien vers les référentiels configurés pour être accessibles par cette extension. Pour plus d'informations, consultez [Installation et désinstallation d'extensions dans CodeCatalyst](#) ou suivez [Démarrage rapide : utilisation de GitHub référentiels dans CodeCatalyst](#).

**Note**

Vous ne pouvez lier un dépôt qu'à un seul projet dans un espace. Vous ne pouvez pas lier un dépôt archivé. Bien que vous puissiez lier un dépôt vide, vous ne pouvez pas l'utiliser CodeCatalyst tant que vous ne l'avez pas initialisé avec un commit initial qui crée une branche par défaut.

### Pour lier un référentiel source

1. Accédez au projet auquel vous souhaitez lier un référentiel.


**Note**

Avant de pouvoir lier un référentiel, un utilisateur ayant le rôle d'administrateur de Space doit d'abord installer l'extension pour le fournisseur qui héberge le référentiel. Pour plus d'informations, consultez [Installation d'une extension](#).

2. Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources.




3. Choisissez Ajouter un référentiel, puis choisissez Lier le référentiel.
4. Dans Fournisseur de référentiel, choisissez le nom du fournisseur. Dans le GitHub compte, choisissez le compte lié que vous souhaitez utiliser pour travailler dans le référentiel, puis dans le référentiel, choisissez le nom du référentiel.

 Note

Vous ne pouvez lier qu'un dépôt actif. Vous ne pouvez pas lier un dépôt archivé.

5. Passez en revue les informations, puis choisissez Link.

 Note

Vous ne pouvez lier un dépôt qu'à un seul projet dans un espace.

Pour dissocier un dépôt d'un projet, vous devez avoir le rôle d'administrateur de l'espace. Pour plus d'informations, voir [Gestion des GitHub référentiels dans CodeCatalyst](#).

## Affichage d'un référentiel source

Vous pouvez consulter les référentiels de sources associés à un projet sur Amazon CodeCatalyst. Pour les référentiels sources dans CodeCatalyst, la page de présentation d'un référentiel fournit un aperçu rapide des informations et des activités dans ce référentiel, notamment :

- Description du référentiel, le cas échéant
- Le nombre de branches dans le référentiel
- Le nombre de pull requests ouvertes pour le référentiel
- Le nombre de flux de travail associés pour le référentiel
- Les fichiers et dossiers de la branche par défaut, ou de la branche que vous choisissez
- Le titre, l'auteur et la date du dernier commit dans la branche affichée
- Le contenu du fichier README.md affiché dans Markdown, si un fichier README.md est inclus

Cette page fournit également des liens vers les commits, les branches et les pull requests du référentiel, ainsi qu'un moyen rapide d'ouvrir, de visualiser et de modifier des fichiers individuels.

**Note**

Vous ne pouvez pas afficher ces informations sur les référentiels liés dans la CodeCatalyst console. Pour afficher des informations sur les référentiels liés, cliquez sur le lien dans la liste des référentiels pour ouvrir ce référentiel dans le service qui l'héberge.

Pour accéder aux référentiels sources d'un projet

1. Accédez à votre projet, puis effectuez l'une des opérations suivantes :
  - Sur la page de résumé de votre projet, choisissez le référentiel souhaité dans la liste, puis choisissez Afficher le référentiel.
  - Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources. Dans Référentiels sources, choisissez le nom du référentiel dans la liste. Vous pouvez filtrer la liste des référentiels en saisissant une partie du nom du référentiel dans la barre de filtre.
2. Sur la page d'accueil du référentiel, consultez le contenu du référentiel et les informations sur les ressources associées, telles que le nombre de pull requests et les flux de travail. Par défaut, le contenu de la branche par défaut est affiché. Vous pouvez modifier l'affichage en choisissant une autre branche dans la liste déroulante.

**Tip**

Vous pouvez également accéder rapidement aux référentiels de votre projet en choisissant Voir le code du projet sur la page de résumé du projet.

## Modification des paramètres d'un référentiel source

Vous pouvez gérer les paramètres de votre référentiel, notamment en modifiant la description d'un référentiel, en choisissant la branche par défaut, en créant et en gérant des règles de branche, ainsi qu'en créant et en gérant des règles d'approbation pour les pull requests CodeCatalyst. Cela peut aider les membres du projet à comprendre à quoi sert le référentiel et vous aider à appliquer les meilleures pratiques et les meilleurs processus utilisés par l'équipe.

**Note**

Vous ne pouvez pas modifier le nom d'un dépôt source.

Vous ne pouvez pas modifier le nom, la description ou les autres informations d'un référentiel lié dans CodeCatalyst. Pour modifier les informations relatives à un référentiel lié, vous devez les modifier dans le fournisseur qui héberge le référentiel lié. Pour plus d'informations, consultez la documentation du service qui héberge le référentiel lié.

## Pour modifier les paramètres d'un référentiel

1. Dans la CodeCatalyst console, accédez au projet qui contient le référentiel source dont vous souhaitez modifier les paramètres.
2. Sur la page de résumé de votre projet, choisissez le référentiel souhaité dans la liste, puis choisissez Afficher le référentiel. Dans le volet de navigation, vous pouvez également choisir Code, puis Référentiels sources. Choisissez le nom du référentiel dans la liste des référentiels sources du projet.
3. Sur la page d'aperçu du référentiel, choisissez Plus, puis sélectionnez Gérer les paramètres.
4. Effectuez une ou plusieurs des actions suivantes :
  - Modifiez la description du référentiel, puis choisissez Enregistrer.
  - Pour modifier la branche par défaut du référentiel, dans Branche par défaut, choisissez Modifier. Pour plus d'informations, consultez [Afficher et modifier la branche par défaut d'un référentiel](#).
  - Pour ajouter, supprimer ou modifier une règle indiquant les rôles du projet autorisés à effectuer certaines actions dans une branche, dans Règles de branche, sélectionnez Modifier. Pour plus d'informations, consultez [Gérer les actions autorisées pour une branche à l'aide de règles de branche](#).
  - Pour ajouter, supprimer ou modifier une règle d'approbation pour la fusion des pull requests vers une branche, dans Règles d'approbation, choisissez Modifier. Pour plus d'informations, voir [Gérez les exigences relatives à la fusion d'une pull request avec les règles d'approbation](#).

## Clonage d'un dépôt source

Pour travailler efficacement avec plusieurs fichiers, branches et validations dans les référentiels sources, clonez le référentiel source sur votre ordinateur local et utilisez un client Git ou un environnement de développement intégré (IDE) pour apporter des modifications. Validez et transférez vos modifications dans le référentiel source afin de gérer des CodeCatalyst fonctionnalités telles que les problèmes et les pull requests. Vous pouvez également choisir de créer un environnement

de développement pour travailler sur le code. La création d'un environnement de développement clone automatiquement le référentiel et la branche que vous spécifiez dans l'environnement de développement.

### Note

Vous ne pouvez pas cloner des référentiels liés dans la CodeCatalyst console ou créer des environnements de développement pour ceux-ci. Pour cloner un dépôt lié localement, cliquez sur le lien dans la liste des référentiels pour ouvrir ce référentiel dans le service qui l'héberge, puis clonez-le. Pour plus d'informations, consultez la documentation du service qui héberge le référentiel lié.


Pour créer un environnement de développement à partir d'un référentiel source

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources.
3. Choisissez le référentiel source dans lequel vous souhaitez travailler sur le code.
4. Choisissez Create Dev Environment.
5. Choisissez un IDE compatible dans le menu déroulant. Pour plus d'informations, consultez [Environnements de développement intégrés pris en charge pour les environnements de développement](#).
6. Effectuez l'une des actions suivantes :
  - Choisissez Travailler dans une branche existante, puis choisissez une branche dans le menu déroulant Branche existante.
  - Choisissez Travailler dans une nouvelle branche, entrez un nom de branche dans le champ Nom de la branche et choisissez une branche à partir de laquelle créer la nouvelle branche dans le menu déroulant Créer une branche à partir de.
7. Ajoutez éventuellement un nom pour l'environnement de développement ou modifiez sa configuration.
8. Choisissez Créer.

Pour cloner un référentiel source

1. Accédez à votre projet.

2. Sur la page de résumé de votre projet, choisissez le référentiel souhaité dans la liste, puis choisissez Afficher le référentiel. Dans le volet de navigation, vous pouvez également choisir Code, puis Référentiels sources. Choisissez le nom du référentiel dans la liste des référentiels sources du projet. Vous pouvez filtrer la liste des référentiels en saisissant une partie du nom du référentiel dans la barre de filtre.
- 3.
4. Choisissez Cloner le référentiel. Copiez l'URL du clone du référentiel.

 Note


Si vous n'avez pas de jeton d'accès personnel (PAT), choisissez Create token. Copiez le jeton et enregistrez-le dans un emplacement sécurisé. Vous utiliserez ce PAT lorsque votre client Git ou votre environnement de développement intégré (IDE) vous demandera de saisir un mot de passe.

5. Effectuez l'une des actions suivantes :

- Pour cloner un dépôt sur votre ordinateur local, ouvrez un terminal ou une ligne de commande et exécutez la git clone commande avec l'URL du clone après la commande. Par exemple :

```
git clone https://LiJuan@git.us-west-2.codecatalyst.aws/v1/ExampleCorp/MyExampleProject/MyExampleRepo
```

Lorsque vous êtes invité à saisir un mot de passe, collez le PAT que vous avez enregistré précédemment.

 Note

Si votre système d'exploitation assure la gestion des informations d'identification ou si vous avez installé un système de gestion des informations d'identification, vous ne devez fournir le PAT qu'une seule fois. Si ce n'est pas le cas, vous devrez peut-être fournir le PAT pour chaque opération Git. La meilleure pratique consiste à vous assurer que votre système de gestion des accreditations stocke votre PAT en toute sécurité. N'incluez pas le PAT dans la chaîne URL du clone.

- Pour cloner un dépôt à l'aide d'un IDE, suivez la documentation de votre IDE. Choisissez l'option permettant de cloner un dépôt Git et de fournir l'URL. Lorsque vous êtes invité à saisir un mot de passe, indiquez le PAT.

## Supprimer un référentiel source

Si un référentiel source pour un CodeCatalyst projet Amazon n'est plus nécessaire, vous pouvez le supprimer. La suppression d'un référentiel source entraîne également la suppression de toutes les informations de projet stockées dans le référentiel. Si des flux de travail dépendent du référentiel source, ils seront supprimés de la liste des flux de travail du projet une fois le référentiel supprimé. Les problèmes faisant référence au référentiel source ne seront ni supprimés ni modifiés, mais les liens vers le référentiel source ajoutés aux problèmes échoueront une fois le référentiel supprimé.

### Important

La suppression d'un dépôt source ne peut pas être annulée. Une fois que vous avez supprimé un dépôt source, vous ne pouvez plus le cloner, en extraire des données ou y transférer des données. La suppression d'un dépôt source ne supprime aucune copie locale de ce dépôt (dépôts locaux). Pour supprimer un dépôt local, utilisez le répertoire et les outils de gestion de fichiers de votre ordinateur local.

### Note

Vous ne pouvez pas supprimer un référentiel lié dans la CodeCatalyst console. Pour supprimer un référentiel lié, cliquez sur le lien dans la liste des référentiels pour ouvrir ce référentiel dans le service qui l'héberge, puis supprimez-le. Pour plus d'informations, consultez la documentation du service qui héberge le référentiel lié.

Pour supprimer un référentiel lié d'un projet, consultez [Gestion des GitHub référentiels dans CodeCatalyst](#).

### Pour supprimer un référentiel source

1. Accédez au projet qui contient le référentiel source que vous souhaitez supprimer.
2. Sur la page de résumé de votre projet, choisissez le référentiel souhaité dans la liste, puis choisissez Afficher le référentiel. Dans le volet de navigation, vous pouvez également choisir

Code, puis Référentiels sources. Choisissez le nom du référentiel dans la liste des référentiels sources du projet.

3. Sur la page d'accueil du référentiel, choisissez Plus, puis choisissez Supprimer le référentiel.
4. Passez en revue les informations relatives à la branche, à la pull request et au flux de travail associé pour vous assurer que vous ne supprimez pas un référentiel toujours utilisé ou dont le travail n'est pas terminé. Si vous souhaitez continuer, tapez Supprimer, puis choisissez Supprimer.

## Travailler avec des succursales sur Amazon CodeCatalyst

Dans Git, les branches sont des pointeurs ou des références à un commit. Dans le développement, ils constituent un moyen pratique d'organiser votre travail. Vous pouvez utiliser des branches pour séparer le travail sur une nouvelle version ou une version différente des fichiers sans affecter le travail dans les autres branches. Vous pouvez utiliser des branches pour développer de nouvelles fonctionnalités, stocker une version spécifique de votre projet, etc. Vous pouvez configurer des règles pour les branches des référentiels sources afin de limiter certaines actions sur une branche à des rôles spécifiques dans ce projet.

Les référentiels sources d'Amazon CodeCatalyst contiennent du contenu et une branche par défaut, quelle que soit la manière dont vous les créez. Les référentiels liés peuvent ne pas avoir de branche ou de contenu par défaut, mais ils ne sont pas utilisables CodeCatalyst tant que vous ne les avez pas initialisés et créé une branche par défaut. Lorsque vous créez un projet à l'aide d'un plan, vous CodeCatalyst créez un référentiel source pour ce projet qui inclut un fichier README.md, un exemple de code, des définitions de flux de travail et d'autres ressources. Lorsque vous créez un référentiel source sans utiliser de plan, un fichier README.md est ajouté pour vous en tant que premier commit, et une branche par défaut est créée pour vous. Cette branche par défaut est nommée main. Cette branche par défaut est celle utilisée comme branche de base ou par défaut dans les référentiels locaux (dépôts) lorsque les utilisateurs clonent le référentiel.

### Note

Vous ne pouvez pas supprimer la branche par défaut. La première branche créée pour un dépôt source est la branche par défaut de ce dépôt. En outre, la recherche affiche uniquement les résultats de la branche par défaut. Vous ne pouvez pas rechercher de code dans d'autres branches.

La création d'un dépôt dans crée CodeCatalyst également un premier commit, qui crée une branche par défaut contenant un fichier README.md. Le nom de cette branche par défaut est main. Il s'agit du nom de branche par défaut utilisé dans les exemples de ce guide.

## Rubriques

- [Création et suppression d'une branche](#)
- [Afficher et modifier la branche par défaut d'un référentiel](#)
- [Gérer les actions autorisées pour une branche à l'aide de règles de branche](#)
- [Commandes Git pour les branches](#)
- [Afficher les branches et les détails](#)

## Création et suppression d'une branche

Vous pouvez utiliser la CodeCatalyst console pour créer et supprimer des branches dans un CodeCatalyst référentiel. Les branches que vous créez seront visibles par les autres utilisateurs la prochaine fois qu'ils extrairont des modifications du référentiel. Lorsque vous supprimez une branche, des copies de cette branche restent dans les clones du référentiel sur les ordinateurs locaux jusqu'à ce que les utilisateurs extraient et synchronisent ces modifications.

### Tip

Vous pouvez également créer des branches dans le cadre de la création d'un environnement de développement pour travailler sur votre code. Pour plus d'informations, consultez [Création d'un environnement de développement](#).

Vous pouvez également utiliser Git pour créer et supprimer des branches. Pour plus d'informations, consultez [Commandes Git communes pour les branches](#).

Pour créer une branche (console)

1. Dans la CodeCatalyst console, accédez au projet dans lequel se trouve votre référentiel source.
2. Choisissez le nom du référentiel dans la liste des référentiels sources du projet. Dans le volet de navigation, vous pouvez également choisir Code, puis Référentiels sources.
3. Choisissez le référentiel dans lequel vous souhaitez créer une branche.
4. Sur la page d'aperçu du référentiel, choisissez Plus, puis sélectionnez Créer une branche.



5. Entrez le nom de la branche.
6. Choisissez une branche à partir de laquelle créer la branche, puis choisissez Create.

#### Pour supprimer une branche (console)

1. Accédez au projet dans lequel se trouve votre référentiel.
2. Choisissez le nom du référentiel dans la liste des référentiels sources du projet. Dans le volet de navigation, vous pouvez également choisir Code, puis Référentiels sources.

Choisissez le référentiel dans lequel vous souhaitez supprimer une branche.

3. Sur la page d'aperçu du référentiel, choisissez le sélecteur déroulant à côté du nom de la branche, puis choisissez Afficher tout.
4. Choisissez la branche que vous souhaitez supprimer, puis choisissez Supprimer la branche.

#### Note

Vous ne pouvez pas supprimer la branche par défaut d'un référentiel.

5. Une boîte de dialogue de confirmation s'affiche. Il indique le référentiel, le nombre de pull requests ouvertes et le nombre de flux de travail associés à la branche.
6. Pour confirmer la suppression de la branche, tapez supprimer dans la zone de texte, puis choisissez Supprimer.

## Afficher et modifier la branche par défaut d'un référentiel

Vous pouvez spécifier la branche à utiliser comme branche par défaut dans un référentiel source sur Amazon CodeCatalyst. Tous les référentiels sources CodeCatalyst contiennent du contenu et une branche par défaut, quelle que soit la manière dont vous les créez. Si vous utilisez un plan pour créer un projet, la branche par défaut du référentiel source créé pour ce projet est nommée main. Le contenu de la branche par défaut s'affiche automatiquement sur la page d'aperçu de ce référentiel.

La branche par défaut est traitée un peu différemment de toutes les autres branches d'un dépôt source. Il possède une étiquette spéciale à côté de son nom, Default. La branche par défaut est celle utilisée comme branche de base ou par défaut dans les référentiels locaux (dépôts) lorsque les utilisateurs clonent le référentiel sur des ordinateurs locaux avec un client Git. Il s'agit également de la valeur par défaut utilisée lors de la création de flux de travail pour le stockage des fichiers YAML

des flux de travail et pour le stockage des informations relatives aux problèmes. Lorsque vous utilisez la fonction de recherche dans CodeCatalyst, seule la branche par défaut d'un dépôt est recherchée. La branche par défaut étant fondamentale pour de nombreux aspects des projets, vous ne pouvez pas supprimer une branche si elle est spécifiée comme branche par défaut. Cependant, vous pouvez choisir d'utiliser une autre branche comme branche par défaut. Dans ce cas, toutes [les règles de branche](#) appliquées à l'ancienne branche par défaut seront automatiquement appliquées à la branche que vous spécifiez comme branche par défaut.

#### Note

Vous devez avoir le rôle d'administrateur de projet pour modifier la branche par défaut des référentiels sources dans les CodeCatalyst projets. Cela ne s'applique pas aux référentiels liés.

Pour afficher et modifier la branche par défaut d'un référentiel

1. Accédez au projet dans lequel se trouve votre référentiel.
2. Choisissez le nom du référentiel dans la liste des référentiels sources du projet. Dans le volet de navigation, vous pouvez également choisir Code, puis Référentiels sources.

Choisissez le référentiel dans lequel vous souhaitez afficher les paramètres, y compris la branche par défaut.

3. Sur la page d'aperçu du référentiel, choisissez Plus, puis sélectionnez Gérer les paramètres.
4. Dans Branche par défaut, le nom de la branche spécifiée comme branche par défaut est affiché avec une étiquette intitulée Default à côté du nom. Cette même étiquette apparaît à côté du nom de la branche dans la liste des branches dans Branches.
5. Pour modifier la branche par défaut, choisissez Modifier.

#### Note

Vous devez avoir le rôle d'administrateur de projet dans le projet pour modifier la branche par défaut.

6. Choisissez le nom de la branche que vous souhaitez définir comme branche par défaut dans la liste déroulante, puis choisissez Enregistrer.

## Gérer les actions autorisées pour une branche à l'aide de règles de branche

Lorsque vous créez une branche, certaines actions sont autorisées pour cette branche en fonction des autorisations associées à ce rôle. Vous pouvez modifier les actions autorisées pour une branche spécifique en configurant les règles de branche. Les règles de branche sont basées sur le rôle de l'utilisateur dans votre projet. Vous pouvez choisir de limiter certaines actions prédéfinies, telles que le transfert de validations vers une branche, aux utilisateurs ayant un rôle particulier dans un projet. Cela peut vous aider à protéger des branches spécifiques d'un projet en limitant les rôles autorisés à effectuer certaines actions. Par exemple, si vous configurez une règle de branche pour autoriser uniquement les utilisateurs ayant le rôle d'administrateur de projet à fusionner ou à transférer vers cette branche, les utilisateurs ayant d'autres rôles dans le projet ne pourront pas modifier le code de cette branche.

Vous devez examiner attentivement toutes les implications de la création d'une règle pour une branche. Par exemple, si vous choisissez de limiter les push vers une branche aux utilisateurs ayant le rôle d'administrateur de projet, les utilisateurs ayant le rôle de contributeur ne pourront pas créer ou modifier des flux de travail dans cette branche, car le flux de travail YAML est stocké dans cette branche, et ces utilisateurs ne peuvent pas valider et transmettre des modifications au YAML. Il est recommandé de tester les règles de branche une fois que vous les avez créées afin de vous assurer qu'elles n'ont aucun impact que vous n'avez pas prévu. Vous pouvez également utiliser les règles de branche conjointement avec les règles d'approbation pour les pull requests. Pour plus d'informations, consultez [Gérez les exigences relatives à la fusion d'une pull request avec les règles d'approbation](#).

### Note

Vous devez avoir le rôle d'administrateur de projet pour gérer les règles de branche pour les référentiels sources dans les CodeCatalyst projets. Vous ne pouvez pas créer de règles de branche pour les référentiels liés.

Vous ne pouvez créer que des règles de branche plus restrictives que les autorisations par défaut pour le rôle. Vous ne pouvez pas créer de règles de branche plus permissives que ne le permet le rôle d'un utilisateur dans le projet. Par exemple, vous ne pouvez pas créer de règle de branche qui autorise les utilisateurs dotés du rôle de réviseur à accéder à la branche.

Les règles de branche appliquées à la branche par défaut de votre dépôt source se comporteront légèrement différemment des règles de branche appliquées aux autres branches. Toute règle appliquée à la branche par défaut sera automatiquement appliquée à toute branche que vous

spécifiez comme branche par défaut. La branche précédemment définie comme branche par défaut conservera les règles qui lui sont appliquées, sauf qu'elle ne sera plus protégée contre la suppression. Cette protection ne s'applique qu'à la branche par défaut actuelle.

Les règles de branche ont deux états, Standard et Personnalisé. Standard indique que les actions autorisées sur une branche sont celles qui correspondent aux autorisations associées au rôle que l'utilisateur a CodeCatalyst pour les actions de branche. Pour en savoir plus sur les rôles dotés de quelles autorisations, consultez [Utilisation des rôles sur Amazon CodeCatalyst](#). Personnalisé indique qu'une ou plusieurs actions de branche comportent des actions associées à une liste spécifique de rôles autorisés à effectuer cette action, différente des autorisations par défaut accordées par le rôle d'un utilisateur dans le projet.

### Note

Si vous créez une règle de branche pour restreindre une ou plusieurs actions pour une branche, l'action Supprimer la branche est automatiquement définie pour autoriser uniquement les utilisateurs ayant le rôle d'administrateur de projet à supprimer cette branche.

Le tableau suivant répertorie les actions et les paramètres par défaut des rôles autorisés à effectuer ces actions sur une branche.

### Actions et rôles des succursales

Action sur les branches	Rôles autorisés à effectuer cette action lorsqu'aucune règle de branche n'est appliquée
Fusionner avec la branche (cela inclut la fusion d'une pull request avec la branche)	Administrateur de projet, contributeur
Appuyez sur la branche	Administrateur de projet, contributeur
Supprimer la branche	Administrateur de projet, contributeur
Supprimer la branche (branche par défaut)	Non autorisée

Vous ne pouvez pas supprimer les règles de branche, mais vous pouvez les mettre à jour pour autoriser les actions de tous les rôles qui seraient autorisés à effectuer cette action sur une branche, ce qui supprime effectivement la règle.

**Note**

Vous devez avoir le rôle d'administrateur de projet pour configurer les règles de branche pour les référentiels sources dans les CodeCatalyst projets. Cela ne s'applique pas aux référentiels liés. Les référentiels liés ne prennent pas en charge les règles de branche dans CodeCatalyst.

Pour afficher et modifier les règles de branche d'un référentiel

1. Accédez au projet dans lequel se trouve votre référentiel.
2. Choisissez le nom du référentiel dans la liste des référentiels sources du projet. Dans le volet de navigation, vous pouvez également choisir Code, puis Référentiels sources.

Choisissez le référentiel dans lequel vous souhaitez consulter les règles de branche.

3. Sur la page d'aperçu du référentiel, choisissez Branches.
4. Dans la colonne Règles de branche, consultez l'état des règles pour chaque branche du référentiel. Standard indique que les règles d'action des branches sont les règles par défaut pour toutes les branches créées dans un référentiel source et qu'elles correspondent aux autorisations accordées à ces rôles dans un projet. Personnalisé indique qu'une ou plusieurs actions de branche sont soumises à des règles qui limitent une ou plusieurs actions autorisées pour cette branche à un ensemble de rôles différent.

Pour consulter les détails des règles de branche d'une branche, choisissez le mot Standard ou Personnalisé à côté de la branche que vous souhaitez consulter.

5. Pour créer ou modifier une règle de branche, choisissez Gérer les paramètres. Sur la page des paramètres du référentiel source, dans Règles de branche, choisissez Modifier.
6. Dans Branche, choisissez le nom de la branche pour laquelle vous souhaitez configurer une règle dans la liste déroulante. Pour chacun des types d'action autorisés, choisissez les rôles que vous souhaitez autoriser à effectuer cette action dans la liste déroulante, puis choisissez Enregistrer.

## Commandes Git pour les branches

Vous pouvez utiliser Git pour créer, gérer et supprimer des branches dans le clone du dépôt source que vous avez sur votre ordinateur (votre dépôt local) ou dans vos environnements de

développement, puis valider et transférer vos modifications dans votre dépôt CodeCatalyst source (le dépôt distant). Par exemple :

## Commandes Git communes pour les branches

Répertorie toutes les branches du dépôt local avec un astérisque (*) affiché à côté de votre branche actuelle.	<code>git branch</code>
Extrait les informations relatives à toutes les branches existantes du référentiel distant vers le dépôt local.	<code>git fetch</code>
Répertorie toutes les branches du dépôt local et les branches de suivi à distance du dépôt local.	<code>git branch -a</code>
Répertorie uniquement les branches de suivi à distance dans le dépôt local.	<code>git branch -r</code>
Crée une branche dans le dépôt local en utilisant le nom de branche spécifié. Cette branche n'apparaîtra pas dans le référentiel distant tant que vous n'aurez pas validé et appliqué la modification.	<code>git branch <i>branch-name</i></code>
Crée une branche dans le dépôt local en utilisant le nom de branche spécifié, puis y passe.	<code>git checkout -b <i>branch-name</i></code>
Passe à une autre branche du dépôt local en utilisant le nom de branche spécifié.	<code>git checkout <i>other-branch-name</i></code>
Transfère une branche du dépôt local vers le dépôt distant en utilisant le surnom spécifié par le dépôt local pour le dépôt distant et le nom de branche spécifié. Configure également les informations de suivi en amont pour la branche dans le dépôt local.	<code>git push -u <i>remote-name</i> <i>branch-name</i></code>

Fusionne les modifications d'une autre branche du dépôt local vers la branche actuelle du dépôt local.

```
git merge from-other-branch-name
```

Supprime une branche du dépôt local sauf si elle contient du travail qui n'a pas été fusionné.

```
git branch -d branch-name
```

Supprime une branche du référentiel distant en utilisant le surnom spécifié par le dépôt local pour le référentiel distant et le nom de branche spécifié. (Notez l'utilisation du signe deux points (:).) Vous pouvez également `--delete` le spécifier dans le cadre de la commande.

```
git push remote-name :branch-name
```

```
git push remote-name --delete  
branch-name
```

Pour plus d'informations, consultez votre documentation Git.

## Afficher les branches et les détails

Vous pouvez consulter des informations sur les succursales distantes d'Amazon CodeCatalyst, notamment les détails des fichiers, des dossiers et le dernier commit pour une branche spécifique, dans la CodeCatalyst console Amazon. Vous pouvez également utiliser les commandes Git et votre système d'exploitation local pour afficher ces informations pour les branches distantes et locales.

Pour afficher les branches (console)

1. Dans la CodeCatalyst console, accédez au projet qui contient le référentiel source dans lequel vous souhaitez afficher les branches. Choisissez Code, choisissez Référentiels sources, puis choisissez le référentiel source.
2. Choisissez le nom du référentiel dans la liste des référentiels sources du projet. Dans le volet de navigation, vous pouvez également choisir Code, puis Référentiels sources.

Choisissez le référentiel dans lequel vous souhaitez afficher une branche.

3. La branche par défaut du référentiel s'affiche. Vous pouvez consulter la liste des fichiers et des dossiers de la branche, des informations sur le dernier commit et le contenu du fichier README.md, s'il existe dans la branche. Pour afficher les informations relatives à une autre branche, sélectionnez-la dans la liste déroulante des branches du référentiel.

4. Pour afficher toutes les branches d'un référentiel, choisissez Afficher tout. La page Branches affiche des informations sur le nom, le dernier commit et les règles de chaque branche.

Pour plus d'informations sur l'utilisation de Git et de votre système d'exploitation pour afficher les branches et les détails, consultez [Commandes Git communes pour les branches](#), votre documentation Git et la documentation de votre système d'exploitation.

## Utilisation de fichiers sur Amazon CodeCatalyst

Sur Amazon CodeCatalyst, un fichier est une information autonome dont la version est contrôlée et qui est accessible à vous et aux autres utilisateurs du référentiel source et de la succursale où le fichier est stocké. Vous pouvez organiser les fichiers de votre référentiel à l'aide d'une structure de répertoires. CodeCatalyst suit automatiquement toutes les modifications apportées à un fichier. Vous pouvez stocker différentes versions d'un fichier dans différentes branches du référentiel.

Pour ajouter ou modifier plusieurs fichiers dans un référentiel source, vous pouvez utiliser un client Git, un environnement de développement ou un environnement de développement intégré (IDE). Pour ajouter ou modifier un seul fichier, vous pouvez utiliser la CodeCatalyst console.

### Rubriques

- [Création ou ajout d'un fichier](#)
- [Affichage d'un fichier](#)
- [Modification d'un fichier](#)
- [Modification du nom ou suppression d'un fichier](#)

## Création ou ajout d'un fichier

Pour créer et ajouter des fichiers à un référentiel source, vous pouvez utiliser la CodeCatalyst console Amazon, un environnement de développement intégré (IDE) connecté ou un client Git. La CodeCatalyst console inclut un éditeur de code pour créer des fichiers. Cet éditeur est un moyen pratique de créer ou de modifier un fichier simple, tel qu'un fichier README.md, dans une branche d'un référentiel. Lorsque vous travaillez sur plusieurs fichiers, pensez à [créer un environnement de développement](#).

Pour créer un environnement de développement à partir d'un référentiel source

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).



2. Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources.
3. Choisissez le référentiel source dans lequel vous souhaitez travailler sur le code.
4. Choisissez Create Dev Environment.
5. Choisissez un IDE compatible dans le menu déroulant. Pour plus d'informations, consultez [Environnements de développement intégrés pris en charge pour les environnements de développement](#).
6. Effectuez l'une des actions suivantes :
  - Choisissez Travailler dans une branche existante, puis choisissez une branche dans le menu déroulant Branche existante.
  - Choisissez Travailler dans une nouvelle branche, entrez un nom de branche dans le champ Nom de la branche et choisissez une branche à partir de laquelle créer la nouvelle branche dans le menu déroulant Créer une branche depuis.
7. Ajoutez éventuellement un nom pour l'environnement de développement ou modifiez sa configuration.
8. Choisissez Créer.

#### Pour créer un fichier dans la CodeCatalyst console

1. Accédez au projet dans lequel vous souhaitez créer un fichier. Pour plus d'informations sur la manière d'accéder à un référentiel, consultez [Affichage d'un référentiel source](#).
2. Choisissez le nom du référentiel dans la liste des référentiels sources du projet. Dans le volet de navigation, vous pouvez également choisir Code, puis Référentiels sources.

Choisissez le référentiel dans lequel vous souhaitez créer le fichier.

3. (Facultatif) Choisissez la branche dans laquelle vous souhaitez créer le fichier, si vous souhaitez créer le fichier dans une branche différente de la branche par défaut.
4. Choisissez Créer un fichier.
5. Entrez le nom du fichier dans Nom du fichier. Ajoutez le contenu du fichier dans l'éditeur.

#### Tip

Si vous souhaitez créer le fichier dans un sous-dossier ou un sous-répertoire à la racine de la branche, incluez cette structure dans le nom du fichier.

Lorsque vous êtes satisfait de vos modifications, choisissez Commit.

6. Dans Nom du fichier, vérifiez le nom du fichier et apportez-y les modifications souhaitées. Choisissez éventuellement la branche dans laquelle vous souhaitez créer le fichier dans la liste des branches disponibles dans Branch. Dans le message de validation, entrez éventuellement une description brève mais informative de la raison pour laquelle vous avez effectué cette modification. Cela sera affiché sous forme d'informations de validation de base pour le commit qui ajoute le fichier au référentiel source.
7. Choisissez Commit pour valider et transférer le fichier vers le référentiel source.

Vous pouvez également ajouter des fichiers à un référentiel source en le clonant sur votre ordinateur local et en utilisant un client Git ou un environnement de développement intégré (IDE) connecté pour transférer vos fichiers et modifications.

#### Note

Si vous souhaitez ajouter un sous-module Git, vous devez utiliser un client Git ou un environnement de développement et exécuter la `git submodule add` commande. Vous ne pouvez pas ajouter ou afficher des sous-modules Git dans la CodeCatalyst console, ni visualiser les différences entre les sous-modules Git dans les pull requests. Pour plus d'informations sur les sous-modules Git, consultez la [documentation Git](#).

Pour ajouter un fichier à l'aide d'un client Git ou d'un environnement de développement intégré (IDE) connecté

1. Clonez votre dépôt source sur votre ordinateur local. Pour plus d'informations, consultez [Clonage d'un dépôt source](#).
2. Créez des fichiers dans votre dépôt local ou copiez-les dans votre dépôt local.
3. Créez et envoyez un commit en effectuant l'une des opérations suivantes :
  - Si vous utilisez un client Git, sur le terminal ou sur la ligne de commande, exécutez la `git add` commande en spécifiant les noms des fichiers que vous souhaitez ajouter. Sinon, pour ajouter tous les fichiers ajoutés ou modifiés, exécutez la `git add` commande suivie d'un point simple ou double pour indiquer si vous souhaitez inclure toutes les modifications au niveau du répertoire actuel (point unique) ou toutes les modifications dans le répertoire actuel et tous les sous-

répertoires (point double). Pour valider les modifications, exécutez la `git commit -m` commande et fournissez un message de validation. Pour transférer vos modifications dans le référentiel source CodeCatalyst, exécutez `git push`. Pour plus d'informations sur les commandes Git, consultez votre documentation Git et [Commandes Git pour les branches](#).

- Si vous utilisez un environnement de développement ou un IDE, créez des fichiers et ajoutez-y des fichiers, puis validez et appliquez vos modifications. Pour plus d'informations, consultez [Environnements de développement dans CodeCatalyst](#) ou consultez la documentation de votre IDE.

## Affichage d'un fichier

Vous pouvez consulter les fichiers de votre référentiel source dans la CodeCatalyst console Amazon. Vous pouvez afficher les fichiers dans la branche par défaut et dans toutes les autres branches. Le contenu du fichier peut varier en fonction de la branche que vous choisissez d'afficher.

Pour afficher des fichiers dans la CodeCatalyst console

1. Accédez au projet dans lequel vous souhaitez afficher les fichiers. Pour plus d'informations, consultez [Affichage d'un référentiel source](#).
2. Choisissez le nom du référentiel dans la liste des référentiels sources du projet. Dans le volet de navigation, vous pouvez également choisir Code, puis Référentiels sources.  
  
Choisissez le référentiel dans lequel vous souhaitez afficher les fichiers.
3. Une liste de fichiers et de dossiers s'affiche pour la branche par défaut. Les fichiers sont indiqués par une icône en forme de papier, tandis que les dossiers sont indiqués par une icône de dossier.
4. Effectuez l'une des actions suivantes :
  - Pour afficher les fichiers et les dossiers d'une autre branche, sélectionnez-la dans la liste des branches.
  - Pour développer un dossier, sélectionnez-le dans la liste.
5. Pour afficher le contenu d'un fichier spécifique, sélectionnez-le dans la liste. Le contenu du fichier sera affiché dans la branche. Pour afficher le contenu du fichier dans une autre branche, choisissez la branche souhaitée dans le sélecteur de branche.

**i** Tip

Lorsque vous consultez le contenu d'un fichier, vous pouvez sélectionner des fichiers supplémentaires à afficher dans Afficher les fichiers. Pour modifier un fichier, choisissez Modifier.

Vous pouvez afficher plusieurs fichiers dans la console. Vous pouvez également afficher les fichiers que vous avez clonés sur votre ordinateur local à l'aide d'un client Git ou d'un environnement de développement intégré (IDE). Pour plus d'informations, consultez la documentation de votre client Git ou de votre IDE.

**i** Note

Vous ne pouvez pas afficher les sous-modules Git dans la CodeCatalyst console. Pour plus d'informations sur les sous-modules Git, consultez la [documentation Git](#).

## Modification d'un fichier

Vous pouvez modifier des fichiers individuels dans la CodeCatalyst console Amazon. Pour modifier plusieurs fichiers à la fois, créez un environnement de développement ou clonez le référentiel et apportez vos modifications à l'aide d'un client Git ou d'un environnement de développement intégré (IDE). Pour plus d'informations, consultez [Environnements de développement dans CodeCatalyst](#) ou [Clonage d'un dépôt source](#).

Pour modifier un fichier dans la CodeCatalyst console

1. Accédez au projet dans lequel vous souhaitez modifier un fichier. Pour plus d'informations sur la manière d'accéder à un référentiel, consultez [Affichage d'un référentiel source](#).
2. Choisissez le référentiel dans lequel vous souhaitez modifier le fichier. Choisissez Afficher les branches, puis choisissez la branche dans laquelle vous souhaitez travailler. Choisissez le fichier dans la liste des fichiers et dossiers de cette branche.

Le contenu du fichier s'affiche.

3. Choisissez Modifier.

4. Dans l'éditeur, modifiez le contenu du fichier, puis choisissez Commit. Facultativement, dans la section Valider les modifications, ajoutez des informations supplémentaires sur la modification dans le message de validation. Lorsque vous êtes satisfait de vos modifications, choisissez Commit.

## Modification du nom ou suppression d'un fichier

Vous pouvez renommer ou supprimer des fichiers dans un environnement de développement, localement sur votre ordinateur ou dans un environnement de développement intégré (IDE). Une fois que vous avez renommé ou supprimé les fichiers, validez et transférez ces modifications dans le référentiel source. Vous ne pouvez pas renommer ou supprimer des fichiers dans la CodeCatalyst console Amazon.

## Utilisation des pull requests sur Amazon CodeCatalyst

Une pull request est le principal moyen pour vous et les autres membres du projet de consulter, commenter et fusionner les modifications de code d'une branche à l'autre. Vous pouvez utiliser des pull requests pour examiner les modifications de code de manière collaborative afin de détecter des modifications ou des correctifs mineurs, des ajouts de fonctionnalités majeures ou de nouvelles versions de vos logiciels publiés. Si vous utilisez les problèmes pour suivre le travail sur votre projet, vous pouvez associer des problèmes spécifiques aux pull requests pour vous aider à suivre les problèmes résolus par les modifications de code apportées dans la pull request. Lorsque vous créez, mettez à jour, commentez, fusionnez ou fermez une pull request, un e-mail est automatiquement envoyé à l'auteur de la pull request ainsi qu'à tous les réviseurs requis ou facultatifs pour la pull request.

### Tip

Vous pouvez configurer les événements de pull request pour lesquels vous recevrez des e-mails dans le cadre de votre profil. Pour plus d'informations, consultez [Gestion des notifications sur Amazon CodeCatalyst](#).

Les pull requests nécessitent deux branches dans un référentiel source : une branche source contenant le code que vous souhaitez réviser et une branche de destination, dans laquelle vous souhaitez fusionner le code révisé. La branche source contient l'APRÈS-validation, qui est la validation qui contient les modifications que vous souhaitez fusionner dans la branche de destination.

La branche de destination contient l'AVANT-validation, qui représente l'état du code avant que la branche de la demande d'extraction ne soit fusionnée dans la branche de destination.

### Note

Lorsque vous créez une pull request, la différence affichée est la différence entre l'extrémité de la branche source et la pointe de la branche de destination. Une fois que vous avez créé la pull request, la différence affichée se situe entre la révision de la pull request que vous avez choisie et le commit qui figurait au début de la branche de destination lorsque vous avez créé la pull request. Pour plus d'informations sur les différences et les bases de fusion dans Git, consultez [git-merge-base](#) la documentation Git.

Lorsqu'une pull request est créée pour un référentiel source et des branches spécifiques, vous pouvez les créer, les afficher, les revoir et les fermer dans le cadre de votre projet. Il n'est pas nécessaire de consulter le référentiel source pour visualiser et utiliser les pull requests. L'état d'une pull request est défini sur Ouvert lorsque vous la créez. La pull request reste ouverte jusqu'à ce que vous la fusionniez dans la CodeCatalyst console, ce qui change l'état en Merged, ou que vous la fermiez, ce qui change l'état en Closed.

Lorsque votre code a été révisé, vous pouvez modifier l'état de la pull request de plusieurs manières :

- Fusionnez la pull request dans la CodeCatalyst console. Le code de la branche source de la pull request sera fusionné dans la branche de destination. Le statut de la pull request deviendra Merged. Il ne peut pas être redéfini sur Ouvert.
- Fusionnez les branches localement et appliquez vos modifications, puis fermez la pull request dans la CodeCatalyst console.
- Utilisez la CodeCatalyst console pour fermer la pull request sans la fusionner. Cela changera le statut en Fermé et ne fusionnera pas le code de la branche source dans la branche de destination.

Avant de créer une demande d'extraction :

- Validez et transférez les modifications de code que vous souhaitez vérifier dans une branche (la branche source).
- Configurez des notifications pour votre projet afin que les autres utilisateurs puissent être informés des flux de travail exécutés lorsque vous créez une pull request. (Cette étape est facultative mais recommandée.)

## Rubriques

- [Création d'une demande d'extraction](#)
- [Afficher les pull requests](#)
- [Gérez les exigences relatives à la fusion d'une pull request avec les règles d'approbation](#)
- [Révision d'une pull request](#)
- [Mettre à jour une pull request](#)
- [Fusion d'une pull request](#)
- [Clôture d'une pull request](#)

## Création d'une demande d'extraction

Les fonctionnalités d'IA générative d'Amazon CodeCatalyst sont en version préliminaire et sont susceptibles d'être modifiées. Ils ne sont disponibles que dans la région ouest des États-Unis (Oregon). L'accès aux fonctionnalités génératives de l'IA varie selon le niveau. Pour plus d'informations, consultez [Tarification d'](#).

La création de demandes d'extraction permet aux autres utilisateurs de voir et de vérifier vos modifications de code avant de les fusionner dans une autre branche. Tout d'abord, vous devez créer une branche pour vos modifications de code. C'est ce que l'on appelle la branche source d'une demande d'extraction. Après avoir validé et transféré les modifications apportées au référentiel, vous pouvez créer une pull request qui compare le contenu de la branche source au contenu de la branche de destination.

Vous pouvez créer une pull request dans la CodeCatalyst console Amazon depuis une branche spécifique, depuis la page des pull requests ou depuis l'aperçu du projet. La création d'une pull request à partir d'une branche spécifique fournit automatiquement le nom du référentiel et la branche source sur la page de création de la pull request. Lorsque vous créez une pull request, vous recevez automatiquement des e-mails concernant toute mise à jour de la pull request, ainsi que la date de fusion ou de fermeture de la pull request.

### Note

Lorsque vous créez une pull request, la différence affichée est la différence entre l'extrémité de la branche source et celle de la branche de destination. Une fois la pull request créée,

la différence affichée se situera entre la révision de la pull request que vous avez choisie et le commit qui figurait au début de la branche de destination lorsque vous avez créé la pull request. Pour plus d'informations sur les différences et les bases de fusion dans Git, consultez [git-merge-base](#) la documentation Git.

Vous pouvez utiliser la fonctionnalité Rédiger une description pour moi lorsque vous créez des pull requests pour qu'Amazon Q crée automatiquement une description des modifications contenues dans une pull request. Lorsque vous choisissez cette option, Amazon Q analyse les différences entre la branche source contenant les modifications de code et la branche de destination dans laquelle vous souhaitez fusionner ces modifications. Il crée ensuite un résumé de ces modifications, ainsi que sa meilleure interprétation de l'intention et de l'effet de ces modifications.

#### Note

Propulsé par Amazon Bedrock : AWS implémente la [détection automatique des abus](#). Les fonctionnalités Write description for me et Create content summary étant basées sur Amazon Bedrock, les utilisateurs peuvent tirer pleinement parti des contrôles mis en œuvre dans Amazon Bedrock pour renforcer la sûreté, la sécurité et l'utilisation responsable de l'intelligence artificielle (IA).

Pour créer une pull request

1. Accédez à votre projet.
2. Effectuez l'une des actions suivantes :
  - Dans le volet de navigation, choisissez Code, choisissez Pull requests, puis Create pull request.
  - Sur la page d'accueil du référentiel, choisissez Plus, puis Create pull request.
  - Sur la page du projet, choisissez Create pull request.
3. Dans le référentiel source, assurez-vous que le référentiel source spécifié est celui qui contient le code validé. Cette option n'apparaît que si vous n'avez pas créé la pull request depuis la page principale du dépôt.
4. Dans Branche de destination, choisissez la branche dans laquelle vous souhaitez fusionner le code une fois celui-ci révisé.
5. Dans Branche source, choisissez la branche qui contient le code validé.




6. Dans Titre de la demande d'extraction, entrez un titre qui aide les autres utilisateurs à comprendre ce qui doit être revu et pourquoi.
7. (Facultatif) Dans la description de la Pull request, fournissez des informations telles qu'un lien vers les problèmes ou une description de vos modifications.

 Tip

Vous pouvez choisir Write description for me afin de générer CodeCatalyst automatiquement une description des modifications contenues dans la pull request. Vous pouvez apporter des modifications à la description générée automatiquement après l'avoir ajoutée à la pull request.


Cette fonctionnalité nécessite que les fonctionnalités d'IA génératives soient activées pour l'espace. Pour plus d'informations, consultez [la section Gestion des fonctionnalités d'IA générative](#).

8. (Facultatif) Dans Problèmes, choisissez Lier les problèmes, puis choisissez un problème dans la liste ou entrez son ID. Pour dissocier un problème, cliquez sur l'icône de dissociation.
9. (Facultatif) Dans Réviseurs obligatoires, sélectionnez Ajouter les réviseurs requis. Choisissez parmi la liste des membres du projet pour les ajouter. Les réviseurs requis doivent approuver les modifications avant que la pull request puisse être fusionnée dans la branche de destination.

 Note

Vous ne pouvez pas ajouter un réviseur à la fois comme réviseur obligatoire et comme réviseur facultatif. Vous ne pouvez pas vous ajouter en tant que réviseur.


10. (Facultatif) Dans Réviseurs facultatifs, sélectionnez Ajouter des réviseurs facultatifs. Choisissez parmi la liste des membres du projet pour les ajouter. Les réviseurs facultatifs n'ont pas à approuver les modifications comme condition préalable pour que la pull request puisse être fusionnée dans la branche de destination.
11. Passez en revue les différences entre les branches. La différence affichée dans une pull request réside dans les modifications entre la révision dans la branche source et la base de fusion, qui est le commit principal de la branche de destination au moment de la création de la pull request. Si aucune modification ne s'affiche, il se peut que les branches soient identiques ou que vous ayez choisi la même branche pour la source et pour la destination.
12. Lorsque vous êtes certain que la pull request contient le code et les modifications que vous souhaitez vérifier, choisissez Create.

 Note

Après avoir créé la pull request, vous pouvez ajouter des commentaires. Des commentaires peuvent être ajoutés à la pull request ou à des lignes individuelles des fichiers, ainsi qu'à la pull request globale. Vous pouvez ajouter des liens vers des ressources, telles que des fichiers, en utilisant le signe @ suivi du nom du fichier.

## Pour créer une pull request depuis une branche


1. Accédez au projet dans lequel vous souhaitez créer une pull request.
2. Dans le volet de navigation, choisissez Référentiels sources, puis choisissez le référentiel contenant la branche dans laquelle vous devez vérifier les modifications de code.
3. Cliquez sur la flèche déroulante à côté du nom de branche par défaut, puis sélectionnez la branche souhaitée dans la liste. Pour afficher toutes les branches d'un référentiel, choisissez Afficher tout.
4. Choisissez Plus, puis choisissez Create pull request.
5. Le référentiel et la branche source sont présélectionnés pour vous. Dans la branche de destination, choisissez la branche dans laquelle vous allez fusionner le code une fois qu'il aura été révisé. Dans Titre de la demande Pull, entrez un titre qui aidera les autres utilisateurs du projet à comprendre ce qui doit être révisé et pourquoi. Vous pouvez éventuellement fournir des informations supplémentaires dans la description de la demande Pull, par exemple en collant un lien vers des problèmes connexes ou en CodeCatalyst ajoutant une description des modifications que vous avez apportées.

 Note

Les flux de travail configurés pour être exécutés lors d'événements de création de pull request s'exécuteront après la création de la pull request, si la branche de destination de la pull request correspond à l'une des branches spécifiées dans le flux de travail.

6. Passez en revue les différences entre les branches. Si aucune modification n'est affichée, les branches sont peut-être identiques ou vous avez peut-être choisi la même branche pour la source et pour la destination.
7. (Facultatif) Dans Problèmes, choisissez Lier les problèmes, puis choisissez un problème dans la liste ou entrez son ID. Pour dissocier un problème, cliquez sur l'icône de dissociation.

8. (Facultatif) Dans Réviseurs obligatoires, sélectionnez Ajouter les réviseurs requis. Choisissez parmi la liste des membres du projet pour les ajouter. Les réviseurs requis doivent approuver les modifications avant que la pull request puisse être fusionnée dans la branche de destination.

 Note

Vous ne pouvez pas ajouter de réviseur à la fois comme obligatoire et facultatif. Vous ne pouvez pas vous ajouter en tant que réviseur.

9. (Facultatif) Dans Réviseurs facultatifs, sélectionnez Ajouter des réviseurs facultatifs. Choisissez parmi la liste des membres du projet pour les ajouter. Les réviseurs facultatifs n'ont pas à approuver les modifications pour que la pull request puisse être fusionnée dans la branche de destination.
10. Lorsque vous êtes certain que la pull request contient les modifications que vous souhaitez vérifier et inclut les réviseurs requis, choisissez Create.

Si vous avez configuré des flux de travail pour s'exécuter là où la branche correspond à la branche de destination dans la pull request, vous verrez des informations sur ces flux de travail dans la section Vue d'ensemble, dans la zone des détails de la pull request après la création de la pull request. Pour plus d'informations, voir [Ajouter un déclencheur push, pull ou Schedule](#).

## Afficher les pull requests

Les fonctionnalités d'IA générative d'Amazon CodeCatalyst sont en version préliminaire et sont susceptibles d'être modifiées. Ils ne sont disponibles que dans la région ouest des États-Unis (Oregon). L'accès aux fonctionnalités génératives de l'IA varie selon le niveau. Pour plus d'informations, consultez [Tarification d'](#).

Vous pouvez consulter les pull requests pour un projet dans la CodeCatalyst console Amazon. La page de résumé du projet affiche toutes les pull requests ouvertes pour un projet. Pour afficher toutes les pull requests, quel que soit leur état, accédez à la page des pull requests de votre projet. Lorsque vous consultez une pull request, vous pouvez choisir de créer pour vous un résumé de tous les commentaires laissés sur les modifications apportées à la pull request.

**Note**

Propulsé par Amazon Bedrock : AWS implémente la [détection automatique des abus](#). Les fonctionnalités Write description for me et Create content summary étant basées sur Amazon Bedrock, les utilisateurs peuvent tirer pleinement parti des contrôles mis en œuvre dans Amazon Bedrock pour renforcer la sûreté, la sécurité et l'utilisation responsable de l'intelligence artificielle (IA).

## Pour consulter les pull requests ouvertes


1. Accédez au projet dans lequel vous souhaitez afficher les pull requests.
2. Sur la page du projet, les pull requests ouvertes sont affichées, y compris des informations sur le créateur de la pull request, le référentiel contenant les branches de la pull request et la date de création de la pull request. Vous pouvez filtrer la vue de la pull request ouverte par référentiel source.
3. Pour afficher toutes les pull requests, choisissez Afficher tout. Vous pouvez utiliser les sélecteurs pour choisir entre les options. Par exemple, pour afficher toutes les pull requests, choisissez Any status et Any author.

Dans le volet de navigation, vous pouvez également choisir Code, puis Pull requests, puis utiliser les sélecteurs pour affiner votre affichage.

4. Sur la page des demandes d'extraction, vous pouvez trier les demandes d'extraction par identifiant, titre, statut, etc. Pour personnaliser les informations et la quantité d'informations affichées sur la page des pull requests, choisissez l'icône en forme de roue dentée.
5. Pour consulter une pull request spécifique, sélectionnez-la dans la liste.
6. Pour consulter l'état des exécutions de flux de travail associées à cette pull request, le cas échéant, choisissez Overview et consultez les informations dans la zone des détails de la pull request de la pull request sous Exécutions de flux de travail.


Un flux de travail sera exécuté si le flux de travail est configuré avec des événements de création ou de révision de demandes d'extraction, et si les exigences de la branche de destination du flux de travail correspondent à la branche de destination spécifiée dans la demande d'extraction. Pour plus d'informations, consultez [Ajouter un déclencheur push, pull ou Schedule](#).

7. Pour consulter les problèmes liés, le cas échéant, choisissez Vue d'ensemble et consultez les informations figurant dans les détails de la demande d'extraction sous Problèmes. Si vous souhaitez consulter un numéro lié, choisissez son ID dans la liste.
8. (Facultatif) Pour créer un résumé des commentaires laissés sur les modifications de code dans les révisions de la pull request, choisissez Créer un résumé du contenu. Le résumé n'inclura aucun commentaire laissé sur l'ensemble de la pull request.

 Note

Cette fonctionnalité nécessite que les fonctionnalités d'IA génératives soient activées pour l'espace. Pour plus d'informations, consultez [la section Gestion des fonctionnalités d'IA générative](#).

9. Pour afficher les modifications de code dans la pull request, choisissez Changes. Vous pouvez rapidement voir combien de fichiers ont été modifiés dans la pull request, et quels fichiers de la pull request contiennent des commentaires, dans Fichiers modifiés. Le nombre de commentaires affichés à côté d'un dossier indique le nombre de commentaires sur les fichiers de ce dossier. Développez le dossier pour afficher le nombre de commentaires pour chaque fichier qu'il contient. Vous pouvez également consulter les commentaires laissés sur des lignes de code spécifiques.

 Note

Les modifications apportées à une pull request ne peuvent pas toutes être affichées dans la console. Par exemple, vous ne pouvez pas afficher les sous-modules Git dans la console. Vous ne pouvez donc pas voir les différences entre les sous-modules dans une pull request. Certaines différences sont peut-être trop importantes pour être affichées. Pour plus d'informations, consultez [Quotas pour les référentiels sources dans CodeCatalyst](#) et [Affichage d'un fichier](#).

10. Pour consulter les rapports de qualité relatifs à cette pull request, choisissez Reports.

**Note**


Un flux de travail doit être configuré pour générer des rapports afin qu'ils apparaissent dans vos pull requests. Pour plus d'informations, voir [Tests à l'aide de workflows dans CodeCatalyst](#).

## Gérez les exigences relatives à la fusion d'une pull request avec les règles d'approbation

Lorsque vous créez une pull request, vous pouvez choisir d'ajouter des réviseurs obligatoires ou facultatifs à cette pull request individuelle. Cependant, vous pouvez également créer des exigences auxquelles toutes les pull requests doivent répondre lors de la fusion vers une branche de destination spécifique. Ces exigences sont appelées règles d'approbation. Les règles d'approbation sont configurées pour les branches d'un référentiel. Lorsque vous créez une demande d'extraction pour laquelle une règle d'approbation est configurée pour la branche de destination, les exigences de cette règle doivent être satisfaites en plus des approbations de tous les réviseurs requis avant de pouvoir fusionner la demande d'extraction avec cette branche. La création de règles d'approbation peut vous aider à maintenir les normes de qualité pour les fusions avec des succursales telles que votre succursale par défaut.

Les règles d'approbation appliquées à la branche par défaut de votre référentiel source se comporteront légèrement différemment des règles d'approbation appliquées aux autres branches. Toute règle appliquée à la branche par défaut sera automatiquement appliquée à toute branche que vous spécifiez comme branche par défaut. La branche précédemment définie comme branche par défaut conservera toujours les règles qui lui sont appliquées.

Lorsque vous créez des règles d'approbation, vous devez réfléchir à la manière dont ces règles seront respectées par les utilisateurs de votre projet à la fois dans le présent et dans le futur. Par exemple, si votre projet compte six utilisateurs et que vous créez une règle d'approbation qui nécessite cinq approbations avant de pouvoir être fusionnée avec la branche de destination, vous avez effectivement créé une règle qui oblige tout le monde, sauf la personne qui a créé les pull request, à approuver cette pull request avant qu'elle ne puisse être fusionnée.

 Note

Vous devez avoir le rôle d'administrateur de projet pour créer et gérer les règles d'approbation dans les CodeCatalyst projets. Vous ne pouvez pas créer de règles d'approbation pour les référentiels liés.

Vous ne pouvez pas supprimer les règles d'approbation, mais vous pouvez les mettre à jour pour qu'elles n'exigent aucune approbation, ce qui supprime effectivement la règle.

Pour consulter et modifier les règles d'approbation des succursales de destination pour les pull requests


1. Accédez au projet dans lequel se trouve votre référentiel.
2. Choisissez le nom du référentiel dans la liste des référentiels sources du projet. Dans le volet de navigation, vous pouvez également choisir Code, puis Référentiels sources.

Choisissez le référentiel dans lequel vous souhaitez consulter les règles d'approbation.

3. Sur la page d'aperçu du référentiel, choisissez Branches.
4. Dans la colonne Règles d'approbation, choisissez Afficher pour voir le statut de toutes les règles pour chaque branche du référentiel.

Dans Nombre minimum d'approbations, le nombre correspond au nombre d'approbations requises avant qu'une pull request puisse être fusionnée avec cette branche.

5. Pour créer ou modifier une règle d'approbation, choisissez Gérer les paramètres. Sur la page des paramètres du référentiel source, dans Règles d'approbation, sélectionnez Modifier.

 Note

Vous devez avoir le rôle d'administrateur de projet pour modifier les règles d'approbation.

6. Dans Branche, choisissez le nom de la succursale pour laquelle vous souhaitez configurer une règle d'approbation dans la liste déroulante. Dans Nombre minimum d'approbations, entrez un nombre, puis choisissez Enregistrer.

## Révision d'une pull request

Les fonctionnalités d'IA générative d'Amazon CodeCatalyst sont en version préliminaire et sont susceptibles d'être modifiées. Ils ne sont disponibles que dans la région ouest des États-Unis (Oregon). L'accès aux fonctionnalités génératives de l'IA varie selon le niveau. Pour plus d'informations, consultez [Tarification d'](#).

Vous pouvez utiliser la CodeCatalyst console Amazon pour examiner et commenter de manière collaborative les modifications incluses dans une pull request. Vous pouvez ajouter des commentaires à des lignes de code individuelles en faisant la différence entre les branches source et destination, ou en fonction de la différence entre les révisions de la pull request. Vous pouvez choisir de créer un résumé des commentaires laissés sur les modifications de code dans la pull request afin de vous aider à comprendre rapidement les commentaires laissés par les autres utilisateurs. Vous pouvez également choisir de créer un environnement de développement pour travailler sur le code.

### Note

Propulsé par Amazon Bedrock : AWS implémente la [détection automatique des abus](#). Les fonctionnalités Write description for me et Create content summary étant basées sur Amazon Bedrock, les utilisateurs peuvent tirer pleinement parti des contrôles mis en œuvre dans Amazon Bedrock pour renforcer la sûreté, la sécurité et l'utilisation responsable de l'intelligence artificielle (IA).

### Tip

Vous pouvez configurer les événements de pull request pour lesquels vous recevrez des e-mails dans le cadre de votre profil. Pour plus d'informations, consultez [Gestion des notifications sur Amazon CodeCatalyst](#).

Les pull requests indiquent quelle sera la différence entre la révision de la pull request et le commit qui figurait au début de la branche de destination lorsque vous avez créé la pull request. C'est ce qu'on appelle la base de fusion. Pour plus d'informations sur les différences et les bases de fusion dans Git, consultez [git-merge-base](#) la documentation Git.



**i** Tip

Lorsque vous travaillez dans la console, en particulier si une pull request est ouverte depuis un certain temps, pensez à actualiser votre navigateur pour vous assurer que vous disposez de la dernière version disponible pour une pull request avant de commencer à la consulter.


## Pour consulter une pull request dans la CodeCatalyst console

1. Accédez à votre projet.
2. Accédez aux pull requests en effectuant l'une des opérations suivantes :
  - Si la pull request est répertoriée sur la page du projet, choisissez-la dans la liste.
  - Si la pull request n'est pas répertoriée sur la page du projet, choisissez Afficher tout. Utilisez les filtres et triez pour trouver la pull request, puis choisissez-la dans la liste.
  - Dans le volet de navigation, choisissez Code, puis choisissez Pull requests.
3. Choisissez la pull request que vous souhaitez consulter dans la liste. Vous pouvez filtrer la liste des pull requests en saisissant une partie de son nom dans la barre de filtre.
4. Dans Vue d'ensemble, vous pouvez consulter le nom et le titre de la pull request. Vous pouvez créer et consulter les commentaires laissés sur la pull request elle-même. Vous pouvez également consulter les détails de la pull request, y compris les informations sur les exécutions du flux de travail, les problèmes liés, les réviseurs, l'auteur de la pull request et les stratégies de fusion réalisables.

**i** Note

Les commentaires laissés sur des lignes de code spécifiques apparaissent dans Modifications.

5. (Facultatif) Pour ajouter un commentaire qui s'applique à l'intégralité de la pull request, développez Commentaires sur la pull request, puis choisissez Create comment.
6. (Facultatif) Pour afficher un résumé de tous les commentaires laissés concernant les modifications apportées aux révisions de cette pull request, choisissez Créer un résumé des commentaires.

 Note

Cette fonctionnalité nécessite que les fonctionnalités d'IA génératives soient activées pour l'espace. Pour plus d'informations, consultez [la section Gestion des fonctionnalités d'IA générative](#).


7. Dans Changes, vous pouvez voir les différences entre la branche de destination et la version la plus récente de la pull request. S'il existe plusieurs révisions, vous pouvez modifier la différence entre les révisions comparées. Pour plus d'informations sur les révisions, consultez [Révisions](#).

 Tip

Vous pouvez rapidement voir combien de fichiers ont été modifiés dans la pull request, et quels fichiers de la pull request contiennent des commentaires, dans Fichiers modifiés. Le nombre de commentaires affichés à côté d'un dossier indique le nombre de commentaires sur les fichiers de ce dossier. Développez le dossier pour afficher le nombre de commentaires pour chaque fichier qu'il contient.


8. Pour modifier la façon dont les différences sont affichées, choisissez entre Unified et Split.
9. Pour ajouter un commentaire à une ligne de la pull request, rendez-vous sur la ligne que vous souhaitez commenter. Choisissez l'icône de commentaire qui apparaît pour cette ligne, entrez un commentaire, puis choisissez Enregistrer.
10. Pour afficher les modifications entre les révisions d'une pull request, ou entre ses branches source et de destination, choisissez l'une des options disponibles dans Comparing. Les commentaires sur les lignes des révisions sont conservés dans ces révisions.
11. Si vous avez configuré votre flux de travail pour générer un rapport de couverture de code sur les déclencheurs de pull request, vous pouvez consulter les résultats de couverture des lignes et des succursales dans la pull request correspondante. Pour masquer les résultats relatifs à la couverture du code, choisissez Masquer la couverture du code. Pour plus d'informations, consultez [Rapports sur la couverture du code](#).
12. Si vous souhaitez modifier le code de la pull request, vous pouvez créer un environnement de développement à partir de la pull request. Choisissez Create Dev Environment. Ajoutez éventuellement un nom pour l'environnement de développement ou modifiez sa configuration, puis choisissez Create.

13. Dans **Rapports**, vous pouvez consulter les rapports de qualité contenus dans cette pull request. S'il existe plusieurs révisions, vous pouvez modifier la différence entre les révisions comparées. Vous pouvez filtrer les rapports par nom, statut, flux de travail, action et type.

 Note

Un flux de travail doit être configuré pour générer des rapports afin qu'ils apparaissent dans vos pull requests. Pour plus d'informations, consultez [Configuration des rapports](#).

14. Pour consulter un rapport spécifique, sélectionnez-le dans la liste. Pour plus d'informations, consultez [Tests à l'aide de workflows dans CodeCatalyst](#).
15. Si vous êtes répertorié comme réviseur de cette pull request et que vous souhaitez approuver les modifications, assurez-vous de consulter la révision la plus récente, puis choisissez **Approuver**.

 Note

Tous les réviseurs requis doivent approuver une pull request avant de pouvoir la fusionner.

## Mettre à jour une pull request

Vous pouvez faciliter la révision du code par les autres membres du projet en mettant à jour la pull request. Vous pouvez mettre à jour une pull request pour modifier ses réviseurs, ses liens vers des problèmes, le titre de la pull request ou sa description. Par exemple, vous souhaitez peut-être modifier les réviseurs requis pour une pull request afin de supprimer une personne en vacances et d'ajouter quelqu'un d'autre. Vous pouvez également mettre à jour une pull request avec d'autres modifications de code en envoyant des validations vers la branche source d'une pull request ouverte. Chaque envoi vers la branche source d'une pull request dans le référentiel CodeCatalyst source crée une révision. Les membres du projet peuvent voir les différences entre les révisions dans une pull request.

Pour mettre à jour les réviseurs d'une pull request

1. Accédez au projet dans lequel vous souhaitez mettre à jour les réviseurs d'une pull request.
2. Sur la page du projet, sous **Ouvrir les pull requests**, choisissez la pull request dans laquelle vous souhaitez mettre à jour les réviseurs. Sinon, dans le volet de navigation, choisissez **Code**, choisissez **Pull requests**, puis choisissez la pull request que vous souhaitez mettre à jour.

3. (Facultatif) Dans Vue d'ensemble, dans la zone des détails de la demande d'extraction, choisissez le signe plus pour ajouter des réviseurs obligatoires ou facultatifs. Cliquez sur le X à côté d'un réviseur pour le supprimer en tant que réviseur facultatif ou obligatoire.
4. (Facultatif) Dans Vue d'ensemble, dans la zone Détails de la demande d'extraction, choisissez Lier les problèmes pour lier un problème à la demande d'extraction, puis choisissez un problème dans la liste ou entrez son ID. Pour dissocier un problème, cliquez sur l'icône de dissociation à côté du problème que vous souhaitez dissocier.

### Pour mettre à jour les fichiers et le code dans la branche source d'une pull request

1. Pour mettre à jour plusieurs fichiers, [créez un environnement](#) de développement ou clonez le référentiel et sa branche source et utilisez un client Git ou un environnement de développement intégré (IDE) pour apporter des modifications aux fichiers de la branche source. Validez et envoyez les modifications à la branche source dans le référentiel CodeCatalyst source pour mettre automatiquement à jour la pull request avec les modifications. Pour plus d'informations, consultez [Clonage d'un dépôt source](#) et [Utilisation des commits sur Amazon CodeCatalyst](#).
2. Pour mettre à jour un fichier individuel dans une branche source, vous pouvez utiliser un client Git ou un IDE comme vous le feriez pour plusieurs fichiers. Vous pouvez également le modifier directement dans la CodeCatalyst console. Pour plus d'informations, consultez [Modification d'un fichier](#).

### Pour mettre à jour le titre et la description d'une pull request

1. Accédez au projet dans lequel vous souhaitez mettre à jour le titre ou la description d'une pull request.
2. La page du projet affiche les pull requests ouvertes, y compris des informations sur le créateur de la pull request, le référentiel contenant les branches de la pull request et la date de création de la pull request. Vous pouvez filtrer la vue de la pull request ouverte par référentiel source. Choisissez la pull request que vous souhaitez modifier dans la liste.
3. Pour afficher toutes les pull requests, choisissez Afficher tout. Dans le volet de navigation, vous pouvez également choisir Code, puis Pull requests. Utilisez la boîte de filtre ou les fonctions de tri pour trouver la pull request que vous souhaitez modifier, puis choisissez-la.
4. Dans Vue d'ensemble, choisissez Modifier.
5. Modifiez le titre ou la description, puis choisissez Enregistrer.

## Fusion d'une pull request

Une fois que votre code a été révisé et que tous les réviseurs requis l'ont approuvé, vous pouvez fusionner une pull request dans la CodeCatalyst console en utilisant une stratégie de fusion compatible, telle que fast-forward. Toutes les stratégies de fusion prises en charge dans la CodeCatalyst console ne sont pas disponibles en tant que choix pour toutes les pull requests. CodeCatalyst évalue la fusion et vous permet uniquement de choisir entre les stratégies de fusion disponibles dans la console et capables de fusionner la branche source avec la branche de destination. Vous pouvez également fusionner une pull request avec les stratégies de fusion Git de votre choix en exécutant la git merge commande sur votre ordinateur local ou dans un environnement de développement pour fusionner la branche source dans la branche de destination. Vous pouvez ensuite transférer ces modifications de la branche de destination vers le référentiel source CodeCatalyst.

### Note

La fusion de la branche et l'application des modifications dans Git ne ferme pas automatiquement la pull request.

Si vous avez le rôle d'administrateur de projet, vous pouvez également choisir de fusionner une pull request qui ne répond pas encore à toutes les exigences en matière d'approbations et de règles d'approbation.

### Fusion d'une pull request (console)


Vous pouvez fusionner une pull request dans la CodeCatalyst console s'il n'y a aucun conflit de fusion entre les branches source et destination et si tous les réviseurs requis ont approuvé la pull request. En cas de conflit ou si la fusion ne peut pas être terminée, le bouton de fusion est inactif et une étiquette Non fusionnable s'affiche. Dans ce cas, vous devez obtenir l'approbation de tous les approbateurs requis, résoudre les conflits localement si nécessaire et appliquer ces modifications avant de pouvoir fusionner. La fusion d'une pull request enverra automatiquement un e-mail au créateur de la pull request ainsi qu'à tous les réviseurs obligatoires ou facultatifs. Il ne fermera ni ne modifiera automatiquement le statut des problèmes liés à la pull request.

 Tip

Vous pouvez configurer les événements de pull request pour lesquels vous recevrez des e-mails dans le cadre de votre profil. Pour plus d'informations, consultez [Gestion des notifications sur Amazon CodeCatalyst](#).

## Pour fusionner une pull request

1. Accédez au projet dans lequel vous souhaitez fusionner une pull request.
2. Sur la page du projet, sous Ouvrir les pull requests, choisissez la pull request que vous souhaitez fusionner. Si vous ne voyez pas la pull request, choisissez Afficher toutes les pull requests, puis choisissez-la dans la liste. Sinon, dans le volet de navigation, choisissez Code, choisissez Pull requests, puis choisissez la pull request que vous souhaitez fusionner. Choisissez Merge (Fusionner).
3. Choisissez parmi les stratégies de fusion disponibles pour la pull request. Vous pouvez éventuellement sélectionner ou désélectionner l'option permettant de supprimer la branche source après avoir fusionné la pull request, puis choisissez Merge.

 Note

Si le bouton Fusionner est inactif ou si le libellé Non fusionnable s'affiche, cela signifie que les réviseurs requis n'ont pas encore approuvé la pull request ou que la pull request ne peut pas être fusionnée dans la CodeCatalyst console. Un réviseur qui n'a pas approuvé une pull request est indiqué par une icône en forme d'horloge dans la zone des détails de la pull request dans Vue d'ensemble. Si tous les réviseurs requis ont approuvé la pull request mais que le bouton Fusionner est toujours inactif, il se peut que vous ayez un conflit de fusion. Choisissez le libellé Non fusionnable souligné pour obtenir plus de détails sur les raisons pour lesquelles la pull request ne peut pas être fusionnée. Vous pouvez résoudre les conflits de fusion pour la branche de destination dans un environnement de développement ou dans la CodeCatalyst console, puis fusionner la pull request, ou vous pouvez résoudre les conflits et fusionner localement, puis transférer le commit contenant la fusion vers la branche source in CodeCatalyst. Pour plus d'informations, consultez [Fusion d'une pull request \(Git\)](#) votre documentation Git.

## Ignorer les exigences relatives à la fusion

Si vous avez le rôle d'administrateur de projet, vous pouvez choisir de fusionner une pull request qui ne répond pas encore à toutes les exigences en matière d'approbations et de règles d'approbation requises. C'est ce que l'on appelle le remplacement des exigences d'une pull request. Vous pouvez choisir de le faire si le réviseur requis n'est pas disponible ou s'il est urgent de fusionner une pull request spécifique dans une branche dont les règles d'approbation ne peuvent pas être respectées rapidement.

### Pour fusionner une pull request

1. Dans la pull request dans laquelle vous souhaitez annuler les exigences et fusionner, cliquez sur la flèche déroulante à côté du bouton Fusionner. Choisissez Ignorer les exigences d'approbation.
2. Dans Raison de dérogation, expliquez pourquoi vous fusionnez cette pull request sans qu'elle réponde aux règles d'approbation et aux exigences requises en matière de révision. Bien que cela soit facultatif, cela est fortement recommandé.
3. Choisissez éventuellement une stratégie de fusion ou acceptez la stratégie par défaut. Vous pouvez également choisir de mettre à jour le message de validation généré automatiquement avec plus de détails.
4. Sélectionnez ou désélectionnez l'option permettant de supprimer la branche source lors de la fusion. Nous vous recommandons de conserver la branche source lorsque vous passez outre aux exigences relatives à la fusion d'une pull request jusqu'à ce que vous ayez eu l'occasion de revoir la décision avec les autres membres de l'équipe.
5. Choisissez Merge (Fusionner).

### Fusion d'une pull request (Git)

Git propose de nombreuses options de fusion et de gestion de branches. Les commandes suivantes sont quelques-unes des options que vous pouvez utiliser. Pour plus d'informations, consultez la documentation disponible sur le [site Web de Git](#). Une fois que vous avez fusionné et appliqué vos modifications, fermez manuellement la pull request. Pour plus d'informations, consultez [Clôture d'une pull request](#).

## Commandes Git communes pour la fusion de branches

Fusionne les modifications de la branche source du dépôt local vers la branche de destination du dépôt local.

```
git checkout destination-branch-name
```

```
git merge source-branch-name
```

Fusionne la branche source dans la branche de destination, en spécifiant une fusion rapide. Cela fusionne les branches et déplace le pointeur de branche de destination vers l'extrémité de la branche source.

```
git checkout destination-branch-name
```

```
git merge --ff-only source-branch-name
```

Fusionne la branche source avec la branche de destination, en spécifiant une fusion par squash. Cela combine tous les commits de la branche source en un seul commit de fusion dans la branche de destination.

```
git checkout destination-branch-name
```

```
git merge --squash source-branch-name
```

Fusionne la branche source dans la branche de destination, en spécifiant une fusion à trois voies. Cela crée un commit de fusion et ajoute les validations individuelles de la branche source à la branche de destination.

```
git checkout destination-branch-name
```

```
git merge --no-ff source-branch-name
```

Supprime la branche source dans le dépôt local. Cela est utile pour nettoyer votre dépôt local après avoir fusionné avec la branche de destination et transféré les modifications au référentiel source.

```
git branch -d source-branch-name
```

Supprime la branche source du référentiel distant (le référentiel source dans CodeCatalyst) en utilisant le surnom spécifié par le dépôt local pour le référentiel distant. (Notez l'utilisation du signe deux points (:).) Vous pouvez également `--delete` le spécifier dans le cadre de la commande.

```
git push remote-name :source-branch-name
```

```
git push remote-name --delete source-branch-name
```



## Clôture d'une pull request

Vous pouvez marquer une pull request comme étant fermée. Cela ne fusionne pas les pull requests, mais cela peut vous aider à déterminer quelles sont les pull requests qui nécessitent une action et celles qui ne sont plus pertinentes. Nous vous recommandons de fermer une pull request après l'avoir fusionnée. La fermeture d'une pull request enverra automatiquement un e-mail au créateur de la pull request ainsi qu'à tous les réviseurs obligatoires ou facultatifs. Cela ne modifiera pas automatiquement le statut des problèmes liés à la pull request.

### Note

Vous ne pouvez pas rouvrir une pull request une fois qu'elle a été fermée.

Pour fermer une pull request

1. Accédez au projet dans lequel vous souhaitez fermer une pull request.
2. Sur la page du projet, les pull requests ouvertes sont affichées. Choisissez la pull request que vous souhaitez fermer.
3. Choisissez Fermer.
4. Passez en revue les informations, puis choisissez Fermer la pull request.

## Utilisation des commits sur Amazon CodeCatalyst

Les validations sont des instantanés du contenu et des modifications du contenu de votre référentiel. Chaque fois qu'un utilisateur valide et apporte une modification à une branche, ces informations sont enregistrées. Les informations de validation Git incluent l'auteur de la validation, la personne qui a effectué la modification, la date et l'heure, ainsi que les modifications apportées. Des informations similaires sont automatiquement incluses lorsque vous créez ou modifiez un fichier dans la CodeCatalyst console Amazon, mais le nom de l'auteur est votre nom CodeCatalyst d'utilisateur. Vous pouvez également ajouter des balises Git aux validations pour vous aider à identifier des validations spécifiques.

Sur Amazon CodeCatalyst, vous pouvez :

- Afficher la liste des validations pour une branche.

- Affichez les validations individuelles, y compris les modifications apportées à une validation par rapport à son ou ses parents.

Vous pouvez également consulter des fichiers et des dossiers. Pour plus d'informations, consultez [Utilisation de fichiers sur Amazon CodeCatalyst](#).

## Rubriques

- [Afficher les validations d'une branche](#)
- [Modifier le mode d'affichage des validations \(CodeCatalystconsole\)](#)

## Afficher les validations d'une branche

Vous pouvez consulter l'historique des modifications apportées à une branche en consultant les validations de la branche dans la CodeCatalyst console. Cela vous permet de comprendre qui a apporté des modifications à la succursale et à quel moment. Vous pouvez également consulter les modifications apportées dans un commit spécifique.

Vous pouvez également consulter les validations à l'aide de votre client Git. Pour plus d'informations, consultez votre documentation Git.

### Pour afficher les validations (console)

1. Accédez au projet qui contient le référentiel source dans lequel vous souhaitez afficher les validations.
2. Choisissez le nom du référentiel dans la liste des référentiels sources du projet. Dans le volet de navigation, vous pouvez également choisir Code, puis Référentiels sources.

Choisissez le dépôt dans lequel vous souhaitez afficher les validations d'une branche.

3. La branche par défaut du référentiel s'affiche, y compris les informations relatives au dernier commit effectué sur la branche. Choisissez Commits. Vous pouvez également choisir Plus, puis Afficher les validations.
4. Pour afficher les validations pour une autre branche, choisissez le sélecteur de branche, puis choisissez le nom de la branche.
5. Pour afficher les détails d'un commit en particulier, choisissez son titre dans Titre du commit. Les détails du commit sont affichés, y compris des informations sur son commit parent et les modifications apportées au code en comparant le commit parent au commit spécifié.

**i** Tip

Si un commit a plusieurs parents, vous pouvez choisir le commit parent dont vous souhaitez afficher les informations et afficher les modifications en cliquant sur l'icône déroulante à côté de l'ID de validation parent.

## Modifier le mode d'affichage des validations (CodeCatalystconsole)

Vous pouvez modifier les informations affichées dans la vue des validations. Vous pouvez choisir de masquer ou d'afficher des colonnes telles que l'auteur et l'ID de validation.

Pour modifier le mode d'affichage des validations (console)

1. Accédez au projet qui contient le référentiel source dans lequel vous souhaitez afficher les validations.
2. Choisissez le nom du référentiel dans la liste des référentiels sources du projet. Dans le volet de navigation, vous pouvez également choisir Code, puis Référentiels sources.

Choisissez le référentiel dans lequel vous souhaitez modifier la façon dont vous visualisez les validations.

3. La branche par défaut du référentiel s'affiche, y compris les informations relatives au dernier commit effectué sur la branche. Choisissez Commits.
4. Choisissez l'icône d'engrenage.
5. Dans les préférences, choisissez le nombre de validations à afficher et indiquez si vous souhaitez afficher les informations relatives à l'auteur, à la date de validation et à l'ID de validation.

**i** Note

Vous ne pouvez pas masquer le titre du commit dans l'affichage des informations.



6. Lorsque vous avez apporté vos modifications, choisissez Enregistrer pour les enregistrer ou Annuler pour les ignorer.

## Quotas pour les référentiels sources dans CodeCatalyst

Le tableau suivant décrit les quotas et les limites pour les référentiels sources sur Amazon CodeCatalyst. Pour plus d'informations sur les quotas sur Amazon CodeCatalyst, consultez [Quotas pour CodeCatalyst](#).

Ressource	Informations
Noms des branches	<p>Toute combinaison de caractères autorisés comprise entre 1 et 256 caractères doit être unique au sein d'un référentiel. Les noms de branche ne peuvent pas :</p> <ul style="list-style-type: none"> <li>commencer ou se terminer par une barre oblique (/) ou un point (.)</li> <li>consister uniquement du caractère @</li> <li>contenir deux points (..) ou barres obliques (//) consécutifs ou plus, ou la combinaison de caractères suivante : @{</li> <li>contenir des espaces ou l'un des caractères suivants : ? ^ * [ \ ~ :</li> </ul> <p>Les noms de branche sont des références. Un grand nombre des restrictions qui s'appliquent aux noms de branche sont basées sur le standard de référence Git. Pour plus d'informations, consultez <a href="#">Git Internals et git-check-ref-format</a>.</p>
Commentaires sur une pull request	Maximum de 1 000 sur une pull request.
Valider le message	Maximum de 1024 caractères.
Chemins de fichier	<p>Toute combinaison de caractères autorisés pour une longueur de 1 à 4,096 caractères. Un chemin de fichier doit être un nom non ambigu qui spécifie le fichier et son emplacement</p>

Ressource	Informations
	<p>nt exact. Un chemin de fichier ne peut pas avoir une profondeur de plus de 20 répertoires. En outre, un chemin de fichier ne peut pas :</p> <ul style="list-style-type: none"><li>• contenir des chaînes vides</li><li>• être un chemin de fichier relatif</li><li>• inclure les combinaisons de caractères suivantes :</li></ul> <p style="margin-left: 40px;">/./</p> <p style="margin-left: 40px;">/..</p> <p style="margin-left: 40px;">//</p> <ul style="list-style-type: none"><li>• se terminer par une barre oblique ni une barre oblique inverse</li></ul> <p>Les noms et les chemins de fichier doivent être complets. Le nom et le chemin d'un fichier sur votre ordinateur local doivent suivre les normes définies pour ce système d'exploitation. Lorsque vous spécifiez le chemin d'accès à un fichier dans un référentiel, utilisez les normes d'Amazon Linux.</p>
Taille de fichier	Maximum de 6 Mo pour chaque fichier individuel lors de l'utilisation de la CodeCatalyst console.
Taille du fichier visible dans la console CodeCatalyst	Maximum de 6 Mo pour chaque fichier individuel lors de l'utilisation de la CodeCatalyst console.

Ressource	Informations
Taille d'objet blob git	<p>Maximum de 2 Go.</p> <div data-bbox="829 302 1507 856"><p> <b>Note</b></p><p>Aucune limite ne s'applique au nombre ou à la taille totale de tous les fichiers d'une même validation, dans la mesure où les métadonnées ne dépassent pas 6 Mo et où un objet blob unique ne dépasse pas 2 Go. Toutefois, la meilleure pratique consiste à envisager de faire plusieurs petits engagements plutôt qu'un seul grand commit.</p></div>
Métadonnées pour une validation	<p>Maximum de 6 Mo pour les <a href="#">métadonnées combinées d'un commit</a> (par exemple, la combinaison des informations sur l'auteur, de la date, de la liste de validation parent et des messages de validation).</p> <div data-bbox="829 1163 1507 1617"><p> <b>Note</b></p><p>Aucune limite ne s'applique au nombre ou à la taille totale de tous les fichiers d'une même validation, dans la mesure où les données ne dépassent pas 20 Mo, où un fichier individuel ne dépasse pas 6 Mo et où un objet blob unique ne dépasse pas 6 Go.</p></div>
Nombre de CodeCatalyst problèmes pouvant être liés à une pull request	50

Ressource	Informations
Nombre de problèmes liés à Jira pouvant être liés à une pull request	50
Nombre de pull requests ouvertes dans un espace	Maximum de 1 000 pour un CodeCatalyst espace Amazon.
Nombre total de pull requests dans un espace	Maximum de 10 000 pour un CodeCatalyst espace Amazon.
Nombre de références dans une seule commande push	Maximum de 4 000, y compris pour créer, supprimer et mettre à jour. Le nombre total de références dans le référentiel n'est pas limité.
Nombre de référentiels dans un espace	Maximum de 5 000 pour un CodeCatalyst espace Amazon.
Descriptions de référentiel	Toute combinaison de caractères pour une longueur de 0 à 1 000 caractères. Les descriptions de référentiel sont facultatives.
Noms de référentiel	Les noms des référentiels doivent être uniques au sein d'un projet. Ils peuvent contenir n'importe quelle combinaison de lettres, de chiffres, de points, de traits de soulignement et de tirets d'une longueur comprise entre 1 et 100 caractères. Les noms ne distinguent pas les majuscules et minuscules. Les noms de dépôt ne peuvent pas se terminer par .git, ne peuvent pas contenir d'espaces et ne peuvent contenir aucun des caractères suivants : ! ? @ # \$ % ^ & * ( ) + = { } [ ]   \ / > < ~ ` ' " ; :

Ressource	Informations
Taille du référentiel	La taille des référentiels dépend des limites de stockage globales de votre espace. Pour plus d'informations, veuillez consulter <a href="#">Tarification d'</a> et <a href="#">Résolution des problèmes liés aux référentiels sources</a> .
Réviseurs pour une pull request	Maximum de 100 réviseurs au total (facultatif ou obligatoire) pour une pull request.
Résumés écrits pour les pull requests	Le nombre maximum de résumés écrits pour les pull requests dépend du niveau de facturation de votre espace. Pour plus d'informations, consultez <a href="#">Pricing</a> (Tarification CTlong).



# Environnements de développement dans CodeCatalyst

Les environnements de développement sont des environnements de développement basés sur le cloud. Dans Amazon CodeCatalyst, vous utilisez des environnements de développement pour travailler sur le code stocké dans les référentiels sources de votre projet. Lorsque vous créez un environnement de développement, plusieurs options s'offrent à vous :

- Créez un environnement de développement spécifique au projet CodeCatalyst pour travailler sur du code avec un environnement de développement intégré (IDE) pris en charge.
- Créez un environnement de développement vide, clonez du code dans celui-ci à partir d'un référentiel source et travaillez sur ce code avec un IDE compatible.
- Créez un environnement de développement dans un IDE et clonez un référentiel source dans l'environnement de développement

Un fichier de développement est un fichier YAML standard ouvert qui normalise vos environnements de développement. En d'autres termes, ce fichier codifie les outils de développement requis pour votre environnement de développement. Ainsi, vous pouvez rapidement configurer un environnement de développement, passer d'un projet à l'autre et répliquer la configuration de l'environnement de développement entre les membres de l'équipe. Les environnements de développement réduisent le temps que vous consacrez à la création et à la maintenance d'un environnement de développement local, car ils utilisent un fichier de développement qui configure tous les outils dont vous avez besoin pour coder, tester et déboguer pour un projet donné.

Les outils de projet et les bibliothèques d'applications inclus dans votre environnement de développement sont définis par le fichier de développement dans le référentiel source de votre projet. Si vous n'avez pas de fichier de développement dans votre dépôt source, applique CodeCatalyst automatiquement un fichier de développement par défaut. Ce fichier de développement par défaut inclut des outils pour les langages de programmation et les frameworks les plus fréquemment utilisés. Si votre projet a été créé à l'aide d'un plan, un fichier de développement est automatiquement créé par. CodeCatalyst Pour plus d'informations sur le fichier de développement, consultez <https://devfile.io>.

Après avoir créé un environnement de développement, vous êtes le seul à pouvoir y accéder. Dans votre environnement de développement, vous pouvez consulter et travailler sur le code de votre référentiel source dans un IDE compatible.

Par défaut, un environnement de développement est configuré pour disposer d'un processeur à 2 cœurs, de 4 Go de RAM et de 16 Go de stockage persistant. Si vous disposez des autorisations d'administrateur de l'espace, vous pouvez modifier le niveau de facturation de votre espace afin d'utiliser différentes options de configuration de l'environnement de développement et de gérer les limites de calcul et de stockage. Voici un flux de travail possible pour un environnement de développement :

Akua Mansa est une nouvelle développeuse chez Example Corp. Akua a rejoint sa nouvelle équipe juste avant la sortie d'une nouvelle version du produit de l'équipe. L'équipe d'Akua a besoin d'elle pour travailler rapidement sur une nouvelle fonctionnalité pour la prochaine version du produit. Pour éviter un long processus de configuration, Akua accepte une invitation à rejoindre le CodeCatalyst projet de son équipe. Elle s'attribue ensuite le problème associé et crée un environnement de développement à partir d'une branche existante du référentiel source de son équipe. L'environnement de développement d'Akua ouvre le code du référentiel source dans l'IDE de son choix. L'instance IDE ouverte est connectée à son environnement de développement qui contient un fichier de développement qui a été automatiquement identifié et appliqué. Le fichier de développement indique tous les outils dont elle a besoin pour démarrer. Akua écrit le code de la nouvelle fonctionnalité du produit, valide ses modifications de code, les transmet à la branche existante, puis supprime son environnement de développement une fois son travail terminé. Akua a contribué au code du référentiel source de sa nouvelle équipe sans un long processus de configuration.

## Création d'un environnement de développement

Vous pouvez créer un environnement de développement de plusieurs manières :

- Créez un environnement de développement CodeCatalyst avec un référentiel CodeCatalyst source ou un [référentiel source lié à](#) partir des pages Présentation, Environnements de développement ou Référentiels sources
- Créez un environnement de développement vide CodeCatalyst qui n'est pas connecté à un référentiel source à partir de la page Environnements de développement
- Créez un environnement de développement dans l'IDE de votre choix et clonez n'importe quel référentiel source dans l'environnement de développement

Vous pouvez créer un environnement de développement par branche d'un dépôt. Un projet peut avoir plusieurs référentiels. Les environnements de développement que vous créez ne peuvent être gérés qu'avec votre CodeCatalyst compte, mais vous pouvez ouvrir l'environnement de développement et y travailler avec n'importe quel IDE compatible. Vous devez l'avoir AWS Toolkit

installé pour utiliser les environnements de développement dans votre IDE. Pour plus d'informations, consultez [Environnements de développement intégrés pris en charge pour les environnements de développement](#). Par défaut, les environnements de développement sont créés avec un processeur à 2 cœurs, 4 Go de RAM et 16 Go de stockage persistant.

#### Note

Si vous avez créé un environnement de développement associé à un référentiel source, la colonne Ressource indique toujours la branche que vous avez spécifiée lors de la création de cet environnement de développement. Cela s'applique même si vous créez une autre branche, passez à une autre branche dans l'environnement de développement ou clonez un dépôt supplémentaire. Si vous avez créé un environnement de développement vide, la colonne Ressource sera vide.

## Environnements de développement intégrés pris en charge pour les environnements de développement

Vous pouvez utiliser les environnements de développement avec les environnements de développement intégrés (IDE) pris en charge suivants :

- [AWS Cloud9](#)
- [JetBrains IDE](#)
  - [IntelliJ IDEA Ultimate](#)
  - [GoLand](#)
  - [PyCharm Professionnel](#)
- [Code Visual Studio](#)


## Création d'un environnement de développement dans CodeCatalyst

Pour commencer à travailler avec Dev Environment in CodeCatalyst, authentifiez-vous et connectez-vous à l'aide de votre [AWS Builder ID](#) ou de votre [SSO](#).

Pour créer un environnement de développement à partir d'une branche


1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).

2. Accédez au projet dans lequel vous souhaitez créer un environnement de développement.
3. Dans le volet de navigation, effectuez l'une des opérations suivantes :
  - Choisissez Overview, puis accédez à la section Mes environnements de développement.
  - Choisissez Code, puis Dev Environments.
  - Choisissez Code, choisissez Référentiels sources et choisissez le référentiel pour lequel vous souhaitez créer un environnement de développement.
4. Choisissez Create Dev Environment.
5. Choisissez un IDE compatible dans le menu déroulant. Pour plus d'informations, consultez [Environnements de développement intégrés pris en charge pour les environnements de développement](#).
6. Choisissez Cloner un dépôt.
7. Effectuez l'une des actions suivantes :
  - a. Choisissez le référentiel à cloner, choisissez Travailler dans une branche existante, puis choisissez une branche dans le menu déroulant Branche existante.

 Note

Si vous choisissez un dépôt tiers, vous devez travailler dans une branche existante.

- b. Choisissez le référentiel à cloner, choisissez Travailler dans une nouvelle branche, entrez le nom de la branche dans le champ Nom de la branche et choisissez une branche à partir de laquelle créer la nouvelle branche dans le menu déroulant Créer une branche depuis.


 Note

Si vous créez un environnement de développement à partir de la page Référentiels sources ou à partir d'un référentiel source spécifique, il n'est pas nécessaire de choisir un référentiel. L'environnement de développement sera créé à partir du référentiel source que vous avez choisi sur la page Référentiels source.

8. (Facultatif) Dans Alias - facultatif, entrez un alias pour l'environnement de développement.
9. (Facultatif) Cliquez sur le bouton d'édition de la configuration de l'environnement de développement pour modifier la configuration de calcul, de stockage ou de temporisation de l'environnement de développement.

10. (Facultatif) Dans Amazon Virtual Private Cloud (Amazon VPC), sélectionnez une connexion VPC que vous souhaitez associer à votre environnement de développement dans le menu déroulant.

Si un VPC par défaut est défini pour votre espace, vos environnements de développement s'exécuteront connectés à ce VPC. Vous pouvez contourner cela en associant une autre connexion VPC. Notez également que les environnements de développement connectés au VPC ne sont pas pris en charge. AWS Toolkit

 Note

Lorsque vous créez un environnement de développement avec une connexion VPC, une nouvelle interface réseau est créée au sein du VPC. CodeCatalyst interagit avec cette interface en utilisant le rôle VPC associé. Assurez-vous également que votre bloc d'adresse CIDR IPv4 n'est pas configuré pour la plage d'adresses 172.16.0.0/12 IP.

11. Choisissez Créer. Pendant la création de votre environnement de développement, la colonne d'état de l'environnement de développement affichera Démarrage, et la colonne d'état affichera En cours d'exécution une fois l'environnement de développement créé.

Pour créer un environnement de développement vide

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez au projet dans lequel vous souhaitez créer un environnement de développement.
3. Dans le volet de navigation, effectuez l'une des opérations suivantes :
  - Choisissez Overview, puis accédez à la section Mes environnements de développement.
  - Choisissez Code, puis Dev Environments.
4. Choisissez Create Dev Environment.
5. Choisissez un IDE compatible dans le menu déroulant. Pour plus d'informations, consultez [Environnements de développement intégrés pris en charge pour les environnements de développement](#).
6. Choisissez Créer un environnement de développement vide.
7. (Facultatif) Dans Alias - facultatif, entrez un alias pour l'environnement de développement.
8. (Facultatif) Cliquez sur le bouton d'édition de la configuration de l'environnement de développement pour modifier la configuration de calcul, de stockage ou de temporisation de l'environnement de développement.

9. (Facultatif) Dans Amazon Virtual Private Cloud (Amazon VPC), sélectionnez une connexion VPC que vous souhaitez associer à votre environnement de développement dans le menu déroulant.

Si un VPC par défaut est défini pour votre espace, vos environnements de développement s'exécuteront connectés à ce VPC. Vous pouvez contourner cela en associant une autre connexion VPC. Notez également que les environnements de développement connectés au VPC ne sont pas pris en charge. AWS Toolkit

#### Note

Lorsque vous créez un environnement de développement avec une connexion VPC, une nouvelle interface réseau est créée au sein du VPC. CodeCatalyst interagit avec cette interface en utilisant le rôle VPC associé. Assurez-vous également que votre bloc d'adresse CIDR IPv4 n'est pas configuré pour la plage d'adresses 172.16.0.0/12 IP.

10. Choisissez Créer. Pendant la création de votre environnement de développement, la colonne d'état de l'environnement de développement affichera Démarrage, et la colonne d'état affichera En cours d'exécution une fois l'environnement de développement créé.

#### Note

La création et l'ouverture d'un environnement de développement pour la première fois peuvent prendre une à deux minutes.

#### Note

Une fois l'environnement de développement ouvert dans l'IDE, vous devrez peut-être remplacer le répertoire par le référentiel source avant de valider et de transférer les modifications à votre code.

## Création d'un environnement de développement dans un IDE

Vous pouvez utiliser un environnement de développement intégré (IDE) pris en charge pour travailler sur le code stocké dans les référentiels sources de votre projet.

- [Travailler avec Amazon CodeCatalyst dans AWS Cloud9](#)

- [Amazon CodeCatalyst dans VS Code](#)

Pour obtenir des instructions sur l'authentification, consultez [Authentification et connexion à AWS partir de votre environnement de développement](#) et [Authentification pour Amazon CodeCatalyst](#) dans le guide de l' AWS Toolkit for Visual Studio Code utilisateur.

- [Amazon CodeCatalyst en JetBrains](#)

Pour obtenir des instructions sur l'authentification, voir [Authentifier et connecter JetBrains Gateway à Gateway CodeCatalyst](#) dans le guide de l' AWS Toolkit for JetBrains utilisateur.

## Arrêt d'un environnement de développement

Le `/projects` répertoire d'un environnement de développement stocke les fichiers extraits du référentiel source et le fichier de développement utilisé pour configurer l'environnement de développement. Le `/home` répertoire, qui est vide lors de la création de l'environnement de développement, stocke les fichiers que vous créez lors de l'utilisation de votre environnement de développement. Tout ce qui se trouve dans les `/home` répertoires `/projects` et d'un environnement de développement est stocké de manière persistante. Vous pouvez donc arrêter de travailler dans un environnement de développement si vous devez passer à un autre environnement de développement, à un autre référentiel ou à un autre projet.

### Warning

Un environnement de développement n'expirera pas si des instances, y compris les navigateurs Web, les shells distants et les IDE, restent connectées. Veillez donc à fermer toutes les instances connectées pour éviter d'encourir des coûts supplémentaires.

Un environnement de développement s'arrête automatiquement s'il est inactif pendant la durée sélectionnée dans les champs Timeout lors de la création de l'environnement de développement. Vous pouvez arrêter l'environnement de développement avant qu'il ne soit inactif. Si vous avez choisi Aucun délai d'attente lors de la création de votre environnement de développement, celui-ci ne s'arrêtera pas automatiquement. Au lieu de cela, il fonctionnera en continu.

**⚠ Warning**

Si vous arrêtez un environnement de développement associé à une connexion VPC supprimée, il ne peut pas être rétabli.

Pour arrêter un environnement de développement depuis la page Environnements de développement

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez au projet dans lequel vous souhaitez arrêter un environnement de développement.
3. Dans le volet de navigation, sélectionnez Code.
4. Choisissez Dev Environments.
5. Cliquez sur le bouton radio correspondant à l'environnement de développement que vous souhaitez arrêter.
6. Dans le menu Actions, choisissez Stop.

**i Note**


L'utilisation du calcul est facturée uniquement lorsque l'environnement de développement est en cours d'exécution, mais l'utilisation du stockage est facturée pendant toute la durée d'existence de l'environnement de développement. Arrêtez votre environnement de développement lorsqu'il n'est pas utilisé pour arrêter la facturation des calculs.

## Reprise d'un environnement de développement

Le `/projects` répertoire d'un environnement de développement stocke les fichiers extraits du référentiel source et le fichier de développement utilisé pour configurer l'environnement de développement. Le `/home` répertoire, qui est vide lors de la création de l'environnement de développement, stocke les fichiers que vous créez lors de l'utilisation de votre environnement de développement. Tout ce qui se trouve dans les `/home` répertoires `/projects` et d'un environnement de développement est stocké de manière persistante. Vous pouvez donc arrêter de travailler dans un environnement de développement si vous devez passer à un autre environnement de développement, référentiel ou projet et reprendre le travail dans votre environnement de développement ultérieurement.



Un environnement de développement s'arrête automatiquement s'il est inactif pendant la durée sélectionnée dans les champs Timeout lors de la création de l'environnement de développement. Vous devez fermer l'onglet du AWS Cloud9 navigateur pour que l'environnement de développement soit inactif.

 Note

L'environnement de développement est toujours disponible et fonctionne même si vous supprimez la branche avec laquelle vous l'avez créé. Si vous souhaitez recommencer à travailler dans un environnement de développement pour lequel vous avez supprimé la branche, créez une nouvelle branche et apportez-y vos modifications.

Pour reprendre un environnement de développement à partir de la page de présentation

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez au projet dans lequel vous souhaitez reprendre un environnement de développement, puis accédez à la section Mes environnements de développement.
3. Choisissez Resume in (IDE).
  - Pour les JetBrains IDE, choisissez JetBrains Gateway-EAP lorsque vous êtes invité à Choisir une application pour ouvrir le lien -gateway. JetBrains Choisissez Ouvrir le lien pour confirmer lorsque vous y êtes invité.
  - Pour l'IDE VS Code, choisissez VS Code lorsque vous êtes invité à Choisir une application pour ouvrir le lien VS Code. Choisissez Ouvrir le lien pour confirmer.

Pour reprendre un environnement de développement à partir du référentiel source

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez au projet dans lequel vous souhaitez reprendre un environnement de développement.
3. Dans le volet de navigation, sélectionnez Code.
4. Choisissez Sources Repositories.
5. Choisissez le référentiel source qui contient l'environnement de développement que vous souhaitez reprendre.
6. Choisissez le nom de la succursale pour afficher un menu déroulant de vos succursales, puis choisissez votre succursale.

## 7. Choisissez Resume Dev Environment.

- Pour les JetBrains IDE, choisissez Open Link pour confirmer lorsque vous êtes invité à Autoriser ce site à ouvrir le lien JetBrains -gateway avec JetBrains Gateway ? .
- Pour l'IDE VS Code, choisissez Open Link pour confirmer lorsque vous êtes invité à Autoriser ce site à ouvrir le lien VS Code avec Visual Studio Code ? .

Pour reprendre un environnement de développement à partir de la page Environnements de développement

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez au projet dans lequel vous souhaitez reprendre un environnement de développement.
3. Dans le volet de navigation, sélectionnez Code.
4. Choisissez Dev Environments.
5. Dans la colonne IDE, choisissez Resume in (IDE) pour l'environnement de développement.
  - Pour les JetBrains IDE, choisissez Open Link pour confirmer lorsque vous êtes invité à Autoriser ce site à ouvrir le lien JetBrains -gateway avec JetBrains Gateway ? .
  - Pour l'IDE VS Code, choisissez Open Link pour confirmer lorsque vous êtes invité à Autoriser ce site à ouvrir le lien VS Code avec Visual Studio Code ? .

### Note

La reprise d'un environnement de développement peut prendre quelques minutes.

## Modification d'un environnement de développement

Pendant que votre IDE est en cours d'exécution, vous pouvez modifier l'environnement de développement. Si vous modifiez votre délai de calcul ou d'inactivité, votre environnement de développement redémarrera une fois que vous aurez enregistré vos modifications.

Pour modifier un environnement de développement

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez au projet dans lequel vous souhaitez modifier un environnement de développement.

3. Dans le volet de navigation, sélectionnez Code.
4. Choisissez Dev Environments.
5. Choisissez l'environnement de développement que vous souhaitez modifier.
6. Choisissez Modifier.
7. Apportez les modifications souhaitées au délai de calcul ou d'inactivité.
8. Choisissez Enregistrer.

## Suppression d'un environnement de développement

Lorsque vous avez fini de travailler sur le contenu stocké dans votre environnement de développement, vous pouvez supprimer l'environnement de développement. Créez un nouvel environnement de développement pour travailler sur de nouveaux contenus. Si vous supprimez votre environnement de développement, le contenu persistant sera définitivement supprimé. Avant de supprimer votre environnement de développement, assurez-vous de valider et de transférer les modifications de code dans le référentiel source d'origine de l'environnement de développement. Une fois que vous avez supprimé votre environnement de développement, la facturation du calcul et du stockage pour l'environnement de développement s'arrête.

Après avoir supprimé votre environnement de développement, la mise à jour du quota de stockage peut prendre quelques minutes. Si vous avez atteint le quota de stockage, vous ne pourrez pas créer de nouvel environnement de développement pendant cette période.

### Important

La suppression d'un environnement de développement ne peut pas être annulée. Une fois que vous avez supprimé un environnement de développement, vous n'êtes plus en mesure de le récupérer.

Pour supprimer un environnement de développement

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez au projet dans lequel vous souhaitez supprimer un environnement de développement.
3. Dans le volet de navigation, sélectionnez Code.
4. Choisissez Dev Environments.

5. Choisissez l'environnement de développement que vous souhaitez supprimer.
6. Sélectionnez Delete (Supprimer).
7. Entrez **delete** pour confirmer la suppression de l'environnement de développement.
8. Sélectionnez Delete (Supprimer).

#### Note

Avant de supprimer une connexion VPC dans votre espace, assurez-vous de supprimer l'environnement de développement associé à ce VPC.

Même si vous supprimez un environnement de développement, il est possible que vous ne supprimiez pas l'interface réseau du VPC. Assurez-vous de nettoyer vos ressources selon les besoins. Si une erreur se produit lorsque vous supprimez un environnement de développement connecté à un VPC, vous devez [détacher](#) votre connexion périmée et la [supprimer](#) après avoir confirmé qu'elle n'est pas utilisée.

## Connexion à un environnement de développement via SSH

Vous pouvez vous connecter à votre environnement de développement via SSH pour effectuer des actions sans limitation, telles que la redirection de port, le téléchargement de fichiers et l'utilisation d'autres IDE.

#### Note

Si vous souhaitez continuer à utiliser SSH pendant une période prolongée après avoir fermé l'onglet ou la fenêtre IDE, assurez-vous de définir un délai d'expiration élevé pour votre environnement de développement afin qu'il ne s'arrête pas en raison de l'inactivité de l'IDE.

### Prérequis

- Vous avez besoin de l'un des systèmes d'exploitation suivants :
  - Windows 10 ou version ultérieure et OpenSSH activé
  - macOS et Bash version 3 ou supérieure
  - Linux avec yum dpkg ou gestionnaires de rpm paquets et Bash version 3 ou supérieure

- Vous avez également besoin de AWS CLI la version 2.9.4 ou supérieure.

Pour se connecter à un environnement de développement via SSH

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez au projet dans lequel vous souhaitez vous connecter à un environnement de développement via SSH.
3. Dans le volet de navigation, sélectionnez Code.
4. Choisissez Dev Environments.
5. Choisissez un environnement de développement en cours d'exécution auquel vous souhaitez vous connecter via SSH.
6. Choisissez Connect via SSH, choisissez le système d'exploitation de votre choix, puis procédez comme suit :
  - Si ce n'est pas déjà fait, collez et exécutez la première commande dans le terminal que vous avez spécifié. La commande télécharge un script et exécute les modifications suivantes dans votre environnement local afin que vous puissiez vous connecter à votre environnement de développement via SSH :
    - Installe le [plug-in Session Manager pour AWS CLI](#)
    - Modifie votre local AWS Config et ajoute un CodeCatalyst profil afin que vous puissiez effectuer la connexion SSO. Pour plus d'informations, consultez [Configuration pour utiliser AWS CLI le CodeCatalyst](#).
    - Modifie votre configuration SSH locale et ajoute la configuration requise pour la connexion à votre environnement de développement via SSH.
    - Ajoute un script dans le `~/ .aws/codecatalyst-dev-env` répertoire utilisé par le client SSH pour se connecter à votre environnement de développement. Ce script appelle l'[CodeCatalyst StartDevEnvironmentSession API](#) et utilise un AWS Systems Manager Session Manager plugin pour établir une AWS Systems Manager session avec votre environnement de développement qui est utilisée par le client SSH local pour se connecter en toute sécurité à l'environnement de développement distant.
  - Connectez-vous à Amazon à l' CodeCatalyst aide de l' AWS authentification unique à l'aide de la deuxième commande. Cette commande demande et récupère les informations d'identification afin que le script du `~/ .aws/codecatalyst-dev-env` répertoire puisse appeler l'[CodeCatalyst StartDevEnvironmentSession API](#). Cette commande doit être exécutée chaque fois que vos informations d'identification expirent. Lorsque vous exécutez

la dernière commande dans le modal (ssh<destination>), un message d'erreur s'affiche si vos informations d'identification ont expiré ou si vous n'avez pas effectué la connexion SSO comme indiqué dans cette étape.

- Connectez-vous à l'environnement de développement que vous avez spécifié via SSH à l'aide de la troisième commande. La structure de cette commande est la suivante :

```
ssh codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>
```

Vous pouvez également utiliser cette commande pour effectuer d'autres actions autorisées par le client SSH, telles que la redirection de port ou le chargement et le téléchargement de fichiers :

- Redirection de port :

```
ssh -L <local-port>:127.0.0.1:<remote-port> codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>
```

- Téléchargement d'un fichier dans le répertoire de base de votre environnement de développement :

```
scp -0 </path-to-local-file> codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>:</path-to-remote-file-or-directory>
```

## Configuration de votre environnement de développement

Un fichier de développement est un standard ouvert qui vous permet de personnaliser vos environnements de développement au sein de votre équipe. Un fichier de développement est un fichier YAML qui codifie les outils de développement dont vous avez besoin. En configurant un fichier de développement, vous pouvez prédéterminer les outils de projet et les bibliothèques d'applications dont vous avez besoin et Amazon les CodeCatalyst installe pour vous dans votre environnement de développement. Le fichier de développement est spécifique au référentiel pour lequel il a été créé, et vous pouvez créer un fichier de développement distinct pour chaque référentiel. Votre environnement de développement prend en charge les commandes et les événements et fournit une image de fichier de développement universelle par défaut.

Si vous créez un projet à l'aide du plan vide, vous pouvez créer un fichier de développement manuellement. Si vous créez un projet à l'aide d'un plan différent, CodeCatalyst crée automatiquement un fichier de développement. Le /projects répertoire d'un environnement de

développement stocke les fichiers extraits du référentiel source et du fichier de développement. Le `/home` répertoire, qui est vide lorsque vous créez un environnement de développement pour la première fois, stocke les fichiers que vous créez lors de l'utilisation de votre environnement de développement. Tout ce qui se trouve dans les `/home` répertoires `/projects` et d'un environnement de développement est stocké de manière persistante.

#### Note

Le `/home` dossier ne change que si vous modifiez le nom du fichier de développement ou le nom du composant de fichier de développement. Si vous modifiez le nom du devfile ou du composant devfile, le contenu du répertoire est remplacé et les données de votre `/home /home` répertoire précédent ne peuvent pas être récupérées.

Si vous créez un environnement de développement avec un référentiel source qui ne contient pas de fichier de développement à sa racine, ou si vous créez un environnement de développement sans référentiel source, un fichier de développement universel par défaut est automatiquement appliqué au référentiel source. La même image de fichier de développement universelle par défaut est utilisée pour tous les IDE. CodeCatalyst supporte actuellement la version 2.0.0 de devfile. Pour plus d'informations sur le fichier de développement, voir [Schéma de fichier de développement - Version 2.0.0](#).

#### Note

Vous pouvez uniquement inclure des images de conteneur publiques dans votre devfile.

Notez que les environnements de développement connectés à VPC ne prennent en charge que les images de fichier de développement suivantes :

- Image universelle
- Images Amazon ECR privées, si le référentiel se trouve dans la même région que le VPC

## Rubriques

- [Modification d'un fichier de développement d'un référentiel pour un environnement de développement dans CodeCatalyst](#)

- [Modification d'un fichier de développement d'un référentiel pour un environnement de développement dans un IDE](#)
- [Déplacer un fichier de développement d'un référentiel pour un environnement de développement](#)
- [Mode de récupération](#)
- [Fonctionnalités de Devfile prises en charge par CodeCatalyst](#)
- [Exemple : Configuration du fichier de développement pour votre environnement de développement](#)
- [Commandes Devfile](#)
- [Événements Devfile](#)
- [Composants Devfile](#)
- [Images de fichier universelles](#)

## Modification d'un fichier de développement d'un référentiel pour un environnement de développement dans CodeCatalyst

Pour modifier le fichier de développement du référentiel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez au projet qui contient le référentiel source pour lequel vous souhaitez modifier le fichier de développement.
3. Dans le volet de navigation, sélectionnez Code.
4. Choisissez Sources Repositories.
5. Choisissez le référentiel source qui contient le fichier de développement que vous souhaitez modifier.
6. Dans la liste des fichiers, sélectionnez le `devfile.yaml` fichier.
7. Choisissez Modifier.
8. Modifiez le fichier de développement.
9. Choisissez Commit ou créez une pull request afin qu'un membre de l'équipe puisse examiner et approuver les modifications.

### Note

Si vous modifiez votre fichier de développement, vous devez le redémarrer pour que les modifications prennent effet. Cela peut être fait en courant `aws/mde/mde start --`



`location devfile.yaml`. En cas de problème lors du démarrage de votre fichier de développement, celui-ci passe en mode de restauration. Toutefois, si vous modifiez un fichier de développement associé à un environnement de développement connecté à un VPC, vous devez plutôt redémarrer l'environnement de développement pour que les modifications prennent effet.

Vous pouvez vérifier quel fichier de développement est utilisé en exécutant `/aws/mde/mde status`. Le champ d'emplacement contient le chemin du fichier de développement par rapport au `/projects` dossier de l'environnement.

```
{
  "status": "STABLE",
  "location": "devfile.yaml"
}
```

## Modification d'un fichier de développement d'un référentiel pour un environnement de développement dans un IDE

Pour modifier la configuration d'un environnement de développement, vous devez modifier le fichier de développement. Nous vous recommandons de modifier le fichier de développement dans un IDE compatible, puis de mettre à jour votre environnement de développement, mais vous pouvez également modifier le fichier de développement à la racine du référentiel source dans CodeCatalyst. Si vous modifiez le fichier de développement dans un IDE compatible, vous devez valider et transférer vos modifications dans le référentiel source ou créer une pull request afin qu'un membre de l'équipe puisse examiner et approuver les modifications du fichier de développement.

- [Modification du fichier de développement du référentiel pour un environnement de développement dans AWS Cloud9](#)
- [Modification du fichier de développement du référentiel pour un environnement de développement dans VS Code](#)
- [Modification du fichier de développement du référentiel pour un environnement de développement dans JetBrains](#)

## Déplacer un fichier de développement d'un référentiel pour un environnement de développement

Vous pouvez déplacer le fichier de développement par défaut dans `/projects/devfile.yaml` votre référentiel de code source. Pour mettre à jour l'emplacement du fichier de développement, utilisez la commande suivante : `/aws/mde/mde start --location repository-name/devfile.yaml`.

### Mode de récupération

En cas de problème lors du démarrage de votre fichier de développement, celui-ci passe en mode de restauration afin que vous puissiez toujours vous connecter à votre environnement et corriger votre fichier de développement. En mode de restauration, l'exécution `/aws/mde/mde status` ne contiendra pas l'emplacement de votre fichier de développement.

```
{
  "status": "STABLE"
}
```

Vous pouvez vérifier l'erreur dans les journaux ci-dessous `/aws/mde/logs`, corriger le fichier de développement et réessayer de l'exécuter `/aws/mde/mde start`.

## Fonctionnalités de Devfile prises en charge par CodeCatalyst

CodeCatalyst prend en charge les fonctionnalités de fichier de développement suivantes sur la version 2.0.0. Pour plus d'informations sur le fichier de développement, voir [Schéma de fichier de développement - Version 2.0.0](#).

Fonctionnalité	Type
<code>exec</code>	Command
<code>postStart</code>	Événement
<code>container</code>	Composant
<code>args</code>	Propriétés des composants

Fonctionnalité	Type
env	Propriétés des composants
mountSources	Propriétés des composants
volumeMounts	Propriétés des composants

## Exemple : Configuration du fichier de développement pour votre environnement de développement

Voici un exemple de fichier de développement simple.

```
schemaVersion: 2.0.0
metadata:
  name: al2
components:
  - name: test
    container:
      image: public.ecr.aws/amazonlinux/amazonlinux:2
      mountSources: true
      command: ['sleep', 'infinity']
  - name: dockerstore
commands:
  - id: setupscript
    exec:
      component: test
      commandLine: "chmod +x script.sh"
      workingDir: /projects/devfiles
  - id: executescript
    exec:
      component: test
      commandLine: "/projects/devfiles/script.sh"
  - id: yumupdate
    exec:
      component: test
      commandLine: "yum -y update --security"
events:
  postStart:
    - setupscript
    - executescript
```

```
- yumupdate
```

Les journaux de démarrage, de commande et d'événements de Devfile sont capturés et stockés dans `/aws/mde/logs`. Pour déboguer le comportement des fichiers de développement, démarrez votre environnement de développement à l'aide d'un fichier de développement fonctionnel et accédez aux journaux.

## Commandes Devfile

Actuellement, CodeCatalyst seules les `exec` commandes de votre fichier de développement sont prises en charge. Pour plus d'informations, consultez la section [Ajout de commandes](#) dans la documentation de DevFile.io.

L'exemple suivant vous montre comment spécifier des `exec` commandes dans votre fichier de développement.

```
commands:
  - id: setupscript
    exec:
      component: test
      commandLine: "chmod +x script.sh"
      workingDir: /projects/devfiles
  - id: executescript
    exec:
      component: test
      commandLine: "./projects/devfiles/script.sh"
  - id: updateyum
    exec:
      component: test
      commandLine: "yum -y update --security"
```

Une fois connecté à votre environnement de développement, vous pouvez exécuter des commandes définies via le terminal.

```
/aws/mde/mde command <command-id>
/aws/mde/mde command executescript
```

Pour les commandes de longue durée, vous pouvez utiliser l'indicateur `-s` de streaming pour afficher l'exécution de la commande en temps réel.

```
/aws/mde/mde -s command <command-id>
```

### Note

command-id doit être en minuscules.

## Paramètres d'exécution pris en charge par CodeCatalyst

CodeCatalyst prend en charge les exec paramètres suivants sur la version 2.0.0 de devfile.

- `commandLine`
- `component`
- `id`
- `workingDir`

## Événements Devfile

Actuellement, CodeCatalyst ne prend en charge que `postStart` les événements de votre devfile. Pour plus d'informations, consultez la [postStartObject](#) documentation de Devfile.io.

L'exemple suivant vous montre comment ajouter des liaisons d'`postStart` événements dans votre fichier de développement.

```
commands:
  - id: executescript
    exec:
      component: test
      commandLine: "./projects/devfiles/script.sh"
  - id: updateyum
    exec:
      component: test
      commandLine: "yum -y update --security"
events:
  postStart:
```

- updateyum
- executescript

Après le démarrage, votre environnement de développement exécutera les `postStart` commandes spécifiées dans l'ordre dans lequel elles ont été définies. Si une commande échoue, l'environnement de développement continue de fonctionner et le résultat de l'exécution est stocké dans les journaux `ci-dessous/aws/mde/logs`.

## Composants Devfile

Actuellement, CodeCatalyst seules les `container` composants de votre fichier de développement sont pris en charge. Pour plus d'informations, consultez la section [Ajout de composants](#) dans la documentation de DevFile.io.

L'exemple suivant vous montre comment ajouter une commande de démarrage à votre conteneur dans votre fichier de développement.

```
components:  
  - name: test  
    container:  
      image: public.ecr.aws/amazonlinux/amazonlinux:2  
      command: ['sleep', 'infinity']
```

### Note

Lorsque le conteneur possède une commande d'entrée de courte durée, vous devez l'inclure `command: ['sleep', 'infinity']` pour que le conteneur continue de fonctionner.

CodeCatalyst prend également en charge les propriétés suivantes dans votre composant conteneur : `argsenv`, `mountSources`, et `volumeMounts`.

## Images de fichier universelles

L'image universelle par défaut inclut les langages de programmation les plus couramment utilisés et les outils associés qui peuvent être utilisés pour votre IDE. Si aucune image n'est spécifiée, CodeCatalyst fournit cette image et contient des outils gérés par CodeCatalyst. Pour rester informé des nouvelles publications d'images, consultez [Notifications d'image universelles avec SNS](#).

**Note**

L'`public.ecr.aws/aws-mde/universal-image:latest` image obtient l'`public.ecr.aws/aws-mde/universal-image:3.0` image.

Amazon CodeCatalyst prend en charge les images de fichier de développement suivantes.

Version de l'image	Identificateur de l'image
Universal image 1.0	<code>public.ecr.aws/aws-mde/universal-image:1.0</code>
Universal image 2.0	<code>public.ecr.aws/aws-mde/universal-image:2.0</code>
Universal image 3.0	<code>public.ecr.aws/aws-mde/universal-image:3.0</code>

**Note**

Si vous l'utilisez AWS Cloud9, la saisie automatique ne fonctionnera pas pour PHP, Ruby et CSS après la mise à `universal-image:3.0` niveau vers.

## Rubriques

- [Notifications d'image universelles avec SNS](#)
- [Versions d'exécution d'Universal Image 1.0](#)
- [Versions d'exécution d'Universal Image 2.0](#)
- [Versions d'exécution d'Universal Image 3.0](#)

## Notifications d'image universelles avec SNS

CodeCatalyst fournit un service universel de notification par image. Vous pouvez l'utiliser pour vous abonner à une rubrique Amazon Simple Notification Service (SNS) qui vous avertit lorsque des mises

à jour CodeCatalyst universelles des images ont été publiées. Pour plus d'informations sur les sujets liés aux réseaux sociaux, consultez [Qu'est-ce qu'Amazon Simple Notification Service ?](#).

Chaque fois que de nouvelles images universelles sont publiées, nous envoyons des notifications aux abonnés ; cette section explique comment s'abonner aux mises à jour CodeCatalyst universelles des images.

### Exemple de message

```
{
  "Type": "Notification",
  "MessageId": "123456789",
  "TopicArn": "arn:aws:sns:us-east-1:1234657890:universal-image-updates",
  "Subject": "New Universal Image Release",
  "Message": {
    "v1": {
      "Message": "A new version of the Universal Image has been released. You are now able to launch new DevEnvironments using this image.",
      "image ": {
        "release_type": "MAJOR VERSION",
        "image_name": "universal-image",
        "image_version": "2.0",
        "image_uri": "public.ecr.aws/amazonlinux/universal-image:2.0"
      }
    }
  },
  "Timestamp": "2021-09-03T19:05:57.882Z",
  "UnsubscribeURL": "example url"
}
```

Pour vous abonner aux mises à jour CodeCatalyst universelles des images à l'aide de la console Amazon SNS

1. [Ouvrez la console Amazon SNS sur le tableau de bord.](#)
2. Dans la barre de navigation, choisissez votre Région AWS.
3. Dans le panneau de navigation, choisissez Abonnements, puis Créer un abonnement.
4. Dans Topic ARN, entrez `arn:aws:sns:us-east-1:089793673375:universal-image-updates`.
5. Dans Protocole, choisissez E-mail.



6. Dans Endpoint, saisissez une adresse e-mail. Cette adresse e-mail sera utilisée pour recevoir les notifications.
7. Choisissez Créer un abonnement.
8. Vous recevrez un e-mail de confirmation avec pour objet « AWS Notification - Confirmation d'abonnement ». Ouvrez l'e-mail et choisissez Confirmer l'abonnement.

Pour vous désabonner des mises à jour CodeCatalyst universelles des images à l'aide de la console Amazon SNS

1. [Ouvrez la console Amazon SNS sur le tableau de bord.](#)
2. Dans la barre de navigation, choisissez votre Région AWS.
3. Dans le volet de navigation, choisissez Abonnements, puis sélectionnez l'abonnement dont vous souhaitez vous désabonner.
4. Choisissez Actions, puis sélectionnez Supprimer les abonnements.
5. Sélectionnez Delete (Supprimer).

## Versions d'exécution d'Universal Image 1.0

Le tableau suivant répertorie les environnements d'exécution disponibles pour `universal-image:1.0`.

### **universal-image:1.0** versions d'exécution

Nom de l'exécution	Version	Version spécifique majeure et dernière version mineure
agrafe	2.11	aws-cli: 2.x
docker compose	2,16	docker-compose: 2.x
dotnet	6.0	dotnet: 6.x
	7.0	dotnet: 7.x
golang	1,19	golang: 1.x
java	corretto11	java: corretto11.x

Nom de l'exécution	Version	Version spécifique majeure et dernière version mineure
	corretto17	java: corretto17.x
nodejs	14,20	nodejs: 14.x
	16,19	nodejs: 16.x
openssl	1.0	openssl: 1.x
	1.1	
php	7.2	php: 7.x
python	3.9	python: 3.x
	3,10	
ruby	3.1	ruby: 3.x
terraform	1.4	terraform: 1.x

## Versions d'exécution d'Universal Image 2.0

Le tableau suivant répertorie les environnements d'exécution disponibles pour `universal-image:2.0`.

### **universal-image:2.0** versions d'exécution

Nom de l'exécution	Version	Version spécifique majeure et dernière version mineure
agrafe	2.11	aws-cli: 2.x
docker compose	2,17	docker-compose: 2.x
dotnet	6.0	dotnet: 6.x
	7.0	dotnet: 7.x

Nom de l'exécution	Version	Version spécifique majeure et dernière version mineure
golang	1,20	golang: 1.x
java	corretto11	java: corretto11.x
	corretto17	java: corretto17.x
nodejs	16,19	nodejs: 16.x
openssl	1.0	openssl: 1.x
	1.1	
php	7.2	php: 7.x
python	3.9	python: 3.x
	3,10	
ruby	3.2	ruby: 3.x
terraform	1.4	terraform: 1.x

## Versions d'exécution d'Universal Image 3.0

Le tableau suivant répertorie les environnements d'exécution disponibles pour `universal-image:3.0`.

### **universal-image:3.0** versions d'exécution

Nom de l'exécution	Version	Version spécifique majeure et dernière version mineure
agrafe	2.11	aws-cli: 2.x
docker compose	2,17	docker-compose: 2.x
dotnet	6.0	dotnet: 6.x

Nom de l'exécution	Version	Version spécifique majeure et dernière version mineure
	7.0	dotnet: 7.x
golang	1,21	golang: 1.x
java	corretto11	java: corretto11.x
	corretto17	java: corretto17.x
nodejs	18,17	nodejs: 18.x
	20,6	nodejs: 20.x
openssl	3.0	openssl: 3.x
php	8.2	php: 8.x
python	3.9	python: 3.x
	3,11	
ruby	3.2	ruby: 3.x
terraform	1.5	terraform: 1.x

## Utilisation d'environnements de développement avec une connexion VPC

Une connexion VPC est une CodeCatalyst ressource qui contient toutes les configurations nécessaires pour que votre flux de travail accède à un VPC. Les administrateurs de l'espace peuvent ajouter leurs propres connexions VPC dans la CodeCatalyst console Amazon au nom des membres de l'espace. En ajoutant une connexion VPC, les membres de l'espace peuvent exécuter des actions de flux de travail et créer des environnements de développement conformes aux règles du réseau et pouvant accéder aux ressources du VPC associé.

Vous ne pouvez associer un environnement de développement à une connexion VPC que lors de la création de l'environnement de développement. Vous ne pouvez pas modifier la connexion VPC


associée à votre environnement de développement après l'avoir créé. Si vous souhaitez utiliser une autre connexion VPC, vous devez supprimer votre environnement de développement actuel et en créer un nouveau.

 Important

Les environnements de développement dotés d'une connexion VPC ne prennent pas en charge les [référentiels sources tiers](#) liés à CodeCatalyst

Notez que les environnements de développement utilisent plusieurs AWS ressources et services lors de leur création. Cela signifie que les environnements de développement se connectent aux AWS services suivants :

- Amazon CodeCatalyst
- AWS SSM
- AWS KMS
- Amazon ECR
- Amazon CloudWatch
- Amazon ECS

 Note

AWS Toolkit ne prend pas en charge la création d'environnements de développement avec une connexion VPC associée. Notez également que si vous utilisez un IDE autre que AWS Cloud9, vous risquez de rencontrer des temps de chargement d'environ cinq minutes.

Vous devez avoir le rôle d'administrateur de l'espace ou le rôle d'utilisateur avancé pour gérer les connexions VPC au niveau de l'espace. Pour plus d'informations sur les VPC, consultez [Managing Amazon VPC CodeCatalyst dans](#) le guide de l'CodeCatalyst administrateur.

## Utilisation d'environnements de développement avec un IDE

Vous pouvez utiliser les environnements de développement pour travailler rapidement sur le code stocké dans les référentiels sources de votre projet. Les environnements de développement

augmentent votre vitesse de développement car vous pouvez commencer à coder immédiatement dans un environnement de développement cloud entièrement fonctionnel et spécifique à un projet avec un environnement de développement intégré (IDE) pris en charge.

Pour plus d'informations sur l'utilisation à CodeCatalyst partir d'un IDE, consultez la documentation suivante.

- [Amazon CodeCatalyst pour les environnements de JetBrains développement intégrés](#)
- [Amazon CodeCatalyst pour VS Code](#)
- [Amazon CodeCatalyst pour AWS Cloud9](#)

## Quotas pour les environnements de développement dans CodeCatalyst

Le tableau suivant décrit les quotas et les limites pour les environnements de développement sur Amazon CodeCatalyst. Pour plus d'informations sur les quotas sur Amazon CodeCatalyst, consultez [Quotas pour CodeCatalyst](#).

<p>Nombre d'heures d'environnement de développement par mois</p>	<p>Les heures d'ouverture de l'environnement de développement sont influencées par les limites de stockage globales de votre espace. Pour plus d'informations, veuillez consulter <a href="#">Tarification d'</a> et <a href="#">Résolution des problèmes liés aux environnements de développement</a>.</p>
<p>Quantité de stockage de l'environnement de développement par espace</p>	<p>Le stockage dans l'environnement de développement est affecté par les limites de stockage globales de votre espace. Pour plus d'informations, veuillez consulter <a href="#">Tarification d'</a> et <a href="#">Résolution des problèmes liés aux environnements de développement</a>.</p>
<p>Volume de calcul de l'environnement de développement</p>	<p>Le calcul de l'environnement de développement est influencé par les limites de stockage globales de votre espace. Pour plus d'informations, veuillez consulter <a href="#">Tarification d'</a> et</p>

## [Résolution des problèmes liés aux environnements de développement.](#)

# Packages en CodeCatalyst

Amazon CodeCatalyst propose un service de référentiel de packages entièrement géré qui permet à votre équipe de développement de stocker et de partager facilement en toute sécurité les packages logiciels utilisés pour le développement d'applications. Ces packages sont stockés dans des référentiels de packages, qui sont créés et organisés au sein de projets dans CodeCatalyst.

CodeCatalyst prend en charge les formats de package suivants :

- npm

Les packages d'un référentiel de packages peuvent être découverts et partagés entre les membres du projet qui contient le référentiel.

Pour ajouter des packages à un référentiel, configurez un gestionnaire de packages pour utiliser le point de terminaison (URL) du référentiel. Vous pouvez ensuite utiliser le gestionnaire de packages pour publier des packages dans le référentiel.

Vous pouvez configurer CodeCatalyst des flux de travail pour publier des packages dans des référentiels de packages et consommer des packages à partir de CodeCatalyst ces référentiels. Pour plus d'informations sur l'utilisation de packages dans les flux de travail, consultez [Travailler avec des packages](#).

Vous pouvez mettre les packages d'un référentiel de packages à la disposition d'un autre référentiel du même projet en l'ajoutant en tant que référentiel en amont. Toutes les versions de package disponibles dans le référentiel en amont sont également disponibles dans le référentiel en aval.

Vous pouvez mettre des packages open source à la disposition de votre CodeCatalyst dépôt en y connectant des référentiels externes publics. Pour plus d'informations sur les référentiels en amont et la connexion à des référentiels externes, y compris une liste des référentiels pris en charge, consultez [Utilisation de référentiels en amont](#)

## Rubriques

- [Concepts de packages](#)
- [Utilisation des référentiels de packages](#)
- [Utilisation de référentiels en amont](#)
- [Connexion à des référentiels externes publics](#)



- [Travailler avec des packages](#)
- [Utilisation de npm](#)
- [Quotas pour les packages](#)

## Concepts de packages

Voici quelques concepts et termes à connaître lors de la gestion, de la publication ou de la consommation de packages dans CodeCatalyst.

### Packages

Un package est un ensemble qui inclut à la fois le logiciel et les métadonnées nécessaires à l'installation du logiciel et à la résolution des dépendances. CodeCatalyst prend en charge le format de package npm.

Un package comprend :

- Un nom (par exemple, webpack c'est le nom d'un package npm populaire)
- Un espace de [noms](#) facultatif (par exemple, @types dans @types/node)
- Un ensemble de [versions](#) (par exemple, 1.0.0, 1.0.1, 1.0.2)
- Métadonnées au niveau du package (par exemple, balises npm dist)

### Espaces de noms de packages

Certains formats de package prennent en charge les noms de packages hiérarchiques afin d'organiser les packages en groupes logiques et d'éviter les collisions de noms. Les packages portant le même nom peuvent être stockés dans différents espaces de noms. Par exemple, npm prend en charge les étendues, et le package npm @types/node a une portée @types et un nom de node. Il existe de nombreux autres noms de packages dans le @types champ d'application. Dans CodeCatalyst, la portée (« types ») est appelée espace de noms du package, et le nom (« nœud ») est appelé nom du package. Si vous n'avez aucun moyen de regrouper les noms de packages, il peut être plus difficile d'éviter les collisions de noms.

### Versions du package

Une version de package identifie la version spécifique d'un package, telle que @types/node@12.6.9. Le format et la sémantique du numéro de version varient selon les formats de

package. Par exemple, les versions du package npm doivent être conformes à la spécification de [version sémantique](#). Dans CodeCatalyst, une version de package comprend l'identifiant de version, package-version-level les métadonnées et un ensemble de ressources.

## Assets

Un actif est un fichier individuel stocké dans CodeCatalyst lequel est associé à une version de package, tel qu'un .tgz fichier npm.

## Référentiels de packages

Un référentiel de CodeCatalyst packages contient un ensemble de [packages contenant des versions](#) de packages, chacune correspondant à un ensemble de [ressources](#). Chaque référentiel de packages fournit des points de terminaison pour récupérer et publier des packages à l'aide d'outils tels que la CLI Node.js (npm). Vous pouvez créer jusqu'à 1 000 référentiels de packages dans chaque espace.

Vous pouvez lier un référentiel de packages à un autre référentiel en utilisant des référentiels en amont. Lorsque vous liez un référentiel de packages en tant que référentiel en amont, vous pouvez utiliser les packages du référentiel lié via le référentiel configuré. Pour plus d'informations, consultez [Référentiels en amont](#).

Les référentiels Gateway sont un type spécial de référentiel de packages qui extrait et stocke les packages provenant d'autorités de packages externes officielles. Pour plus d'informations, consultez [Référentiels Gateway](#).

## Référentiels Gateway

Un dépôt de passerelle est un type spécial de dépôt de packages connecté à une autorité de package externe officielle prise en charge. Lorsque vous ajoutez un référentiel de passerelle [en tant que référentiel en amont](#), vous pouvez consommer des packages provenant de l'autorité officielle des packages correspondante. Votre référentiel en aval ne communique pas avec le référentiel public, mais tout est intermédiaire par le référentiel de passerelle. Les packages consommés de cette manière sont stockés à la fois dans le référentiel de passerelle et dans le référentiel en aval qui a reçu la demande initiale.

Les référentiels Gateway sont prédéfinis, mais ils doivent être créés dans chaque projet pour être utilisés. La liste suivante contient tous les référentiels de passerelle qui peuvent être créés CodeCatalyst et l'autorité de package à laquelle ils sont connectés.

- `npm-public-registry-gateway` fournit des packages npm de `npmjs.com`.

## Référentiels en amont

Vous pouvez l'utiliser CodeCatalyst pour créer une relation en amont entre deux référentiels de packages. Un référentiel de packages est situé en amont d'un autre lorsque les versions de packages qu'il contient sont accessibles depuis le point de terminaison du référentiel de packages du référentiel en aval. Avec une relation en amont, le contenu des deux référentiels de packages est efficacement fusionné du point de vue du client.

Par exemple, si un gestionnaire de packages demande une version de package qui n'existe pas dans un référentiel, il CodeCatalyst recherchera alors la version du package dans les référentiels en amont configurés. Les référentiels en amont sont recherchés dans l'ordre dans lequel ils sont configurés et, une fois qu'un package est trouvé, la recherche CodeCatalyst s'arrête.

## Utilisation des référentiels de packages

Dans CodeCatalyst, les packages sont stockés et gérés dans des référentiels de packages. Pour publier des packages CodeCatalyst ou pour consommer des packages à partir d'un CodeCatalyst (ou de tout autre référentiel de packages public pris en charge), vous devez créer un référentiel de packages et y connecter votre gestionnaire de packages.

### Rubriques

- [Création d'un référentiel de packages](#)
- [Connexion à un référentiel de packages](#)
- [Modification d'un référentiel de packages](#)
- [Suppression d'un dépôt de packages](#)

## Création d'un référentiel de packages

Procédez comme suit pour créer un référentiel de packages dans CodeCatalyst.

Pour créer un référentiel de packages

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez au projet dans lequel vous souhaitez créer un référentiel de packages.

3. Dans le volet de navigation, sélectionnez Packages.
4. Sur la page Référentiels de packages, choisissez Créer un référentiel de packages.
5. Dans la section Détails du référentiel du package, ajoutez ce qui suit :
  - a. Nom du référentiel. Envisagez d'utiliser un nom descriptif avec des détails tels que le nom de votre projet ou de votre équipe, ou la manière dont le référentiel sera utilisé.
  - b. (Facultatif) Description du référentiel. La description d'un référentiel est particulièrement utile lorsque vous disposez de plusieurs référentiels répartis dans plusieurs équipes au sein d'un projet.
6. Dans la section Modifier les référentiels en amont, ajoutez tous les référentiels de packages auxquels vous souhaitez accéder via votre référentiel de CodeCatalyst packages. Vous pouvez ajouter des référentiels Gateway pour vous connecter à des référentiels de packages externes ou à d'autres référentiels de CodeCatalyst packages.
  - Lorsqu'un package est demandé à partir d'un référentiel de packages, les référentiels en amont sont recherchés dans l'ordre dans lequel ils apparaissent dans cette liste. Une fois qu'un package est trouvé, la recherche CodeCatalyst s'arrête. Pour modifier l'ordre des référentiels en amont, vous pouvez glisser-déposer les référentiels dans la liste ou utiliser les boutons Réorganiser.
7. Choisissez Create pour créer votre référentiel de packages.

## Connexion à un référentiel de packages

Pour publier des packages CodeCatalyst ou pour les consommer CodeCatalyst, vous devez configurer votre gestionnaire de packages avec les informations et les informations d'identification du point de terminaison de votre référentiel de packages. Si vous n'avez pas créé de référentiel, vous pouvez le faire en suivant les instructions figurant dans [Création d'un référentiel de packages](#).

Pour obtenir des instructions sur la façon de connecter le gestionnaire de packages npm à un référentiel de CodeCatalyst packages, consultez [Configuration et utilisation de npm](#).

## Modification d'un référentiel de packages

Procédez comme suit pour modifier la description d'un référentiel de packages et les référentiels en amont.

## Pour modifier un référentiel de packages

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez au projet qui contient le référentiel de packages que vous souhaitez modifier.
3. Dans le volet de navigation, sélectionnez Packages.
4. Sur la page Référentiels de packages, choisissez le référentiel que vous souhaitez supprimer.
5. Choisissez le menu déroulant Actions, puis sélectionnez Modifier.
6. Modifiez la description du référentiel et les référentiels en amont. Pour plus d'informations sur les référentiels en amont, consultez [Utilisation de référentiels en amont](#).
7. Choisissez Enregistrer.

## Suppression d'un dépôt de packages

Procédez comme suit pour supprimer un référentiel de packages dans CodeCatalyst.

### Pour supprimer un dépôt de packages

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez au projet qui contient le référentiel de packages que vous souhaitez supprimer.
3. Dans le volet de navigation, sélectionnez Packages.
4. Sur la page Référentiels de packages, choisissez le référentiel que vous souhaitez supprimer.
5. Choisissez le menu déroulant Actions, puis sélectionnez Supprimer.
6. Consultez les informations fournies concernant les effets de la suppression d'un dépôt de packages.
7. Entrez delete dans le champ de saisie et choisissez Supprimer.

## Utilisation de référentiels en amont

Vous pouvez connecter à la fois des référentiels de passerelle et d'autres référentiels de packages en amont de vos référentiels de packages. Cela permet à un client du gestionnaire de packages d'accéder aux packages contenus dans plusieurs référentiels de packages en utilisant un seul point de terminaison de référentiel de packages. Les principaux avantages de l'utilisation de référentiels en amont sont les suivants :

- Il vous suffit de configurer votre gestionnaire de packages avec un seul point de terminaison de référentiel à extraire de plusieurs sources.
- Les packages consommés à partir d'un référentiel en amont sont stockés dans votre référentiel en aval, ce qui garantit la disponibilité de vos packages même si le référentiel en amont connaît des pannes inattendues.

Vous pouvez ajouter des référentiels en amont lorsque vous créez un référentiel de packages. Vous pouvez également ajouter ou supprimer des référentiels en amont des référentiels de packages existants dans la CodeCatalyst console.

Lorsque vous ajoutez un référentiel de passerelle en tant que référentiel en amont, le référentiel de packages est connecté au référentiel de packages public correspondant du référentiel de passerelle. Pour obtenir la liste des référentiels de packages publics pris en charge, consultez [Référentiels de packages externes pris en charge et leurs référentiels de passerelle](#).

Vous pouvez lier plusieurs référentiels ensemble en tant que référentiels en amont. Par exemple, supposons que votre équipe crée un référentiel nommé `project-repo` et utilise déjà un autre référentiel nommé `team-repo` qui a été `npm-public-registry-gateway` ajouté en tant que référentiel en amont, qui est connecté au référentiel public `npm, npmjs.com`. Vous pouvez l'ajouter en `team-repo` tant que référentiel en amont à `project-repo`. Dans ce cas, il vous suffit de configurer votre gestionnaire de packages pour l'utiliser `project-repo` pour extraire des packages depuis `project-repo, team-repo, npm-public-registry-gateway, et npmjs.com`.

## Rubriques

- [Ajouter un référentiel en amont](#)
- [Modification de l'ordre de recherche des référentiels en amont](#)
- [Demande d'une version de package avec des référentiels en amont](#)
- [Supprimer un référentiel en amont](#)

## Ajouter un référentiel en amont

L'ajout d'un référentiel de packages public ou d'un autre CodeCatalyst référentiel de packages en tant que référentiel en amont à votre référentiel en aval met tous les packages du référentiel en amont à la disposition des gestionnaires de packages connectés au référentiel en aval.

## Pour ajouter un référentiel en amont

1. Dans le panneau de navigation, choisissez Packages.
2. Sur la page Référentiels de packages, choisissez le référentiel de packages auquel vous souhaitez ajouter un référentiel en amont.
3. Choisissez le menu déroulant Actions, puis sélectionnez Modifier.
4. Dans la section Référentiels en amont, choisissez Ajouter un référentiel en amont.
5. Choisissez la barre de recherche pour afficher et rechercher une liste des référentiels disponibles. Vous pouvez ajouter des référentiels de packages publics pris en charge ou d'autres référentiels en tant que CodeCatalyst référentiels en amont. Lorsque vous avez trouvé le référentiel que vous souhaitez ajouter, sélectionnez-le dans la liste.

### Note

Dans les référentiels Gateway, il existe un npm-public-registry-gatewayréférentiel. Pour se connecter aux autorités publiques chargées des packages externes, telles que npmjs.com, un référentiel passerelle est CodeCatalyst utilisé comme référentiel intermédiaire qui recherche et stocke les packages extraits de référentiels externes. Cela permet d'économiser du temps et de transférer des données, car tous les référentiels de packages d'un projet utiliseront des packages provenant du référentiel de passerelle.

6. Lorsque vous avez sélectionné tous les référentiels que vous souhaitez ajouter en tant que référentiels en amont, choisissez Ajouter.
7. Pour plus d'informations sur la modification de l'ordre de recherche des référentiels en amont, consultez [Modification de l'ordre de recherche des référentiels en amont](#).

Lorsque vous avez ajouté un dépôt en amont, vous pouvez utiliser un gestionnaire de packages connecté à votre dépôt local pour récupérer les packages depuis le dépôt en amont. Il n'est pas nécessaire de mettre à jour la configuration de votre gestionnaire de packages. Pour plus d'informations sur la demande de versions de package à partir d'un référentiel en amont, consultez [Demande d'une version de package avec des référentiels en amont](#).

## Modification de l'ordre de recherche des référentiels en amont

CodeCatalyst recherche les référentiels en amont dans leur ordre de recherche configuré. Lorsqu'un package est trouvé, la recherche CodeCatalyst s'arrête. Vous pouvez modifier l'ordre dans lequel les packages sont recherchés dans les référentiels en amont.

## Pour modifier l'ordre de recherche des référentiels en amont

1. Dans le panneau de navigation, choisissez Packages.
2. Sur la page Référentiels, choisissez le référentiel de packages dont vous souhaitez modifier l'ordre de recherche du référentiel en amont.
3. Choisissez le menu déroulant Actions, puis sélectionnez Modifier.
4. Dans la section Référentiels en amont, vous pouvez consulter les référentiels en amont et leur ordre de recherche. Pour modifier l'ordre de recherche, faites glisser les référentiels dans la liste ou utilisez les boutons Réorganiser.
5. Lorsque vous avez terminé de modifier l'ordre de recherche des référentiels en amont, choisissez Enregistrer.

## Demande d'une version de package avec des référentiels en amont

L'exemple suivant montre les scénarios possibles lorsqu'un gestionnaire de packages demande un package à partir d'un référentiel de CodeCatalyst packages contenant des référentiels en amont.

Dans cet exemple, un gestionnaire de packages, tel que `qupm`, demande une version de package à partir d'un référentiel de packages nommé `downstream` qui possède plusieurs référentiels en amont. Lorsque le package est demandé, les événements suivants peuvent se produire :

- S'il `downstream` contient la version du package demandée, il est renvoyé au client.
- S'il `downstream` ne contient pas la version du package demandée, CodeCatalyst recherchez-la dans `downstream` les référentiels en amont, dans l'ordre de recherche configuré. Si la version du package est trouvée, une référence à celle-ci est `downstream` copiée et la version du package est renvoyée au client.
- Si aucun `downstream` des référentiels en amont ne contient la version du package, une `Not Found` réponse HTTP 404 est renvoyée au client.

Le nombre maximum de référentiels en amont autorisés pour un référentiel est de 10. Le nombre maximum de CodeCatalyst recherches dans les référentiels lorsqu'une version de package est demandée est de 25.

## Conservation des packages depuis les référentiels en amont

Si une version de package demandée est trouvée dans un référentiel en amont, une référence à celle-ci est conservée et est toujours disponible dans le référentiel qui l'a demandée. Cela garantit

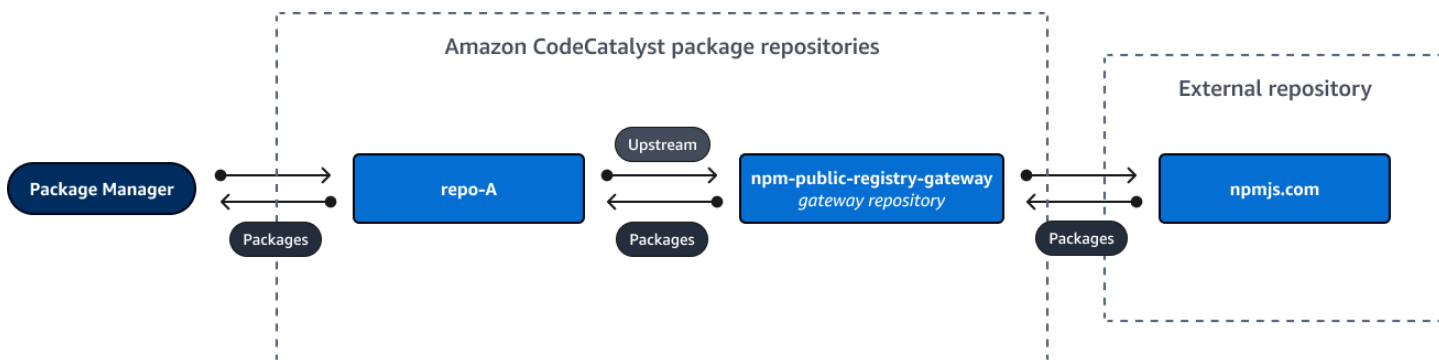


que vous avez accès à vos packages en cas de panne inattendue du référentiel en amont. La version du package conservée n'est affectée par aucun des éléments suivants :

- Suppression du référentiel en amont.
- Déconnexion du référentiel en amont du référentiel en aval.
- Suppression de la version du package du référentiel en amont.
- Modification de la version du package dans le référentiel en amont (par exemple, en y ajoutant une nouvelle ressource).

## Récupération de packages via une relation en amont

CodeCatalyst peut récupérer des packages via plusieurs référentiels liés appelés référentiels en amont. Si un référentiel de CodeCatalyst packages possède une connexion en amont à un autre référentiel de CodeCatalyst packages doté d'une connexion en amont à un référentiel de passerelle, les demandes de packages ne figurant pas dans le référentiel en amont sont copiées depuis le référentiel externe. Par exemple, considérez la configuration suivante : un référentiel nommé `repo-A` possède une connexion en amont avec le référentiel de passerelle, `npm-public-registry-gateway`. `npm-public-registry-gateway` dispose d'une connexion en amont vers le référentiel public de packages, <https://npmjs.com>.



S'il npm est configuré pour utiliser le `repo-A` référentiel, l'exécution `npm install` initie la copie des packages depuis <https://npmjs.com> vers `npm-public-registry-gateway`. Les versions installées sont également intégrées `repo-A`. L'exemple suivant installe `lodash`

```
$ npm config get registry
https://packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/
$ npm install lodash
+ lodash@4.17.20
added 1 package from 2 contributors in 6.933s
```

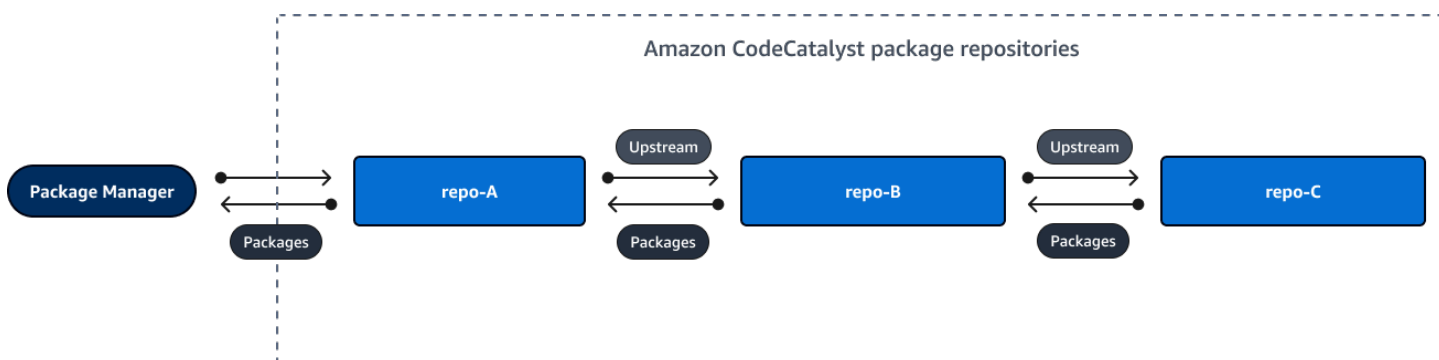
Après l'exécution `npm install`, `repo-A` contient que la dernière version (`lodash 4.17.20`) car c'est la version qui a été récupérée par `npm`. `repo-A`

Comme il `npm-public-registry-gateway` dispose d'une connexion externe en amont à <https://npmjs.com>, toutes les versions de package importées depuis <https://npmjs.com> sont stockées dans `npm-public-registry-gateway`. Ces versions de package auraient pu être récupérées par n'importe quel dépôt en aval disposant d'une connexion en amont à `npm-public-registry-gateway`.

Le contenu de `npm-public-registry-gateway` vous permet de voir tous les packages et versions de packages importés depuis <https://npmjs.com> au fil du temps.

## Rétention des packages dans des référentiels intermédiaires

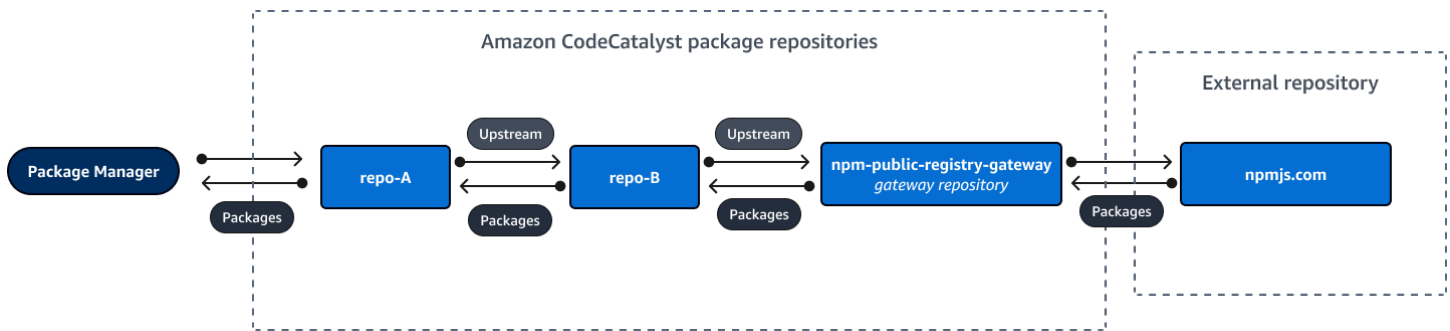
CodeCatalyst vous permet d'enchaîner les référentiels en amont. Par exemple, `repo-A` peut avoir `repo-B` comme référentiel en amont et `repo-B` peut avoir `repo-C` comme référentiel en amont. Cette configuration permet aux versions du package d'être intégrées `repo-B` et `repo-C` disponibles auprès de `repo-A`.



Lorsqu'un gestionnaire de packages se connecte au référentiel `repo-A` et récupère une version de package à partir du référentiel `repo-C`, la version du package n'est pas conservée dans le référentiel `repo-B`. La version du package n'est conservée que dans le référentiel le plus en aval, c'est-à-dire dans cet exemple. `repo-A`. Il n'est conservé dans aucun référentiel intermédiaire. Cela est également vrai pour les chaînes plus longues ; par exemple, s'il y avait quatre référentiels : `repo-A`, `repo-B`, `repo-C`, et `repo-D`, et `repo-A` qu'un gestionnaire de packages était connecté pour `repo-A` récupérer une version de package `repo-D`, la version du package serait conservée dans ou pas dans `repo-B` ou `repo-C`.

Le comportement de rétention des packages est similaire lors de l'extraction d'une version de package depuis un référentiel de packages public, sauf que la version du package est toujours conservée dans le référentiel de passerelle qui dispose d'une connexion directe en amont avec le

référentiel public. Par exemple, `repo-A` a `repo-B` comme référentiel en amont. `repo-B` dispose de `npm-public-registry-gateway` d'un dépôt en amont, qui dispose d'une connexion en amont au dépôt public, `npmjs.com` ; voir le schéma ci-dessous.



Si un gestionnaire de packages connecté à **repo-A** demande une version de package spécifique, `lodash 4.17.20` par exemple, et que la version du package n'est présente dans aucun des trois référentiels, elle sera récupérée sur `npmjs.com`. Lorsque `lodash 4.17.20` est récupéré, il est conservé **repo-A** car il s'agit du dépôt le plus en aval et **npm-public-registry-gateway** puisqu'il dispose d'une connexion en amont au référentiel externe public, `npmjs.com`. `lodash 4.17.20` n'est pas conservé `repo-B` car il s'agit d'un dépôt intermédiaire.

## Supprimer un référentiel en amont

Si vous ne souhaitez plus accéder aux packages d'un référentiel en amont, vous pouvez supprimer le référentiel en amont d'un référentiel de packages.

### Warning

Lorsque vous supprimez un référentiel en amont, vous risquez de rompre les chaînes de relations en amont, ce qui pourrait interrompre vos projets ou vos builds.

Pour supprimer un référentiel en amont

1. Dans le panneau de navigation, choisissez Packages.
2. Sur la page Référentiels de packages, choisissez le référentiel de packages dont vous souhaitez supprimer un référentiel en amont.
3. Choisissez le menu déroulant Actions, puis sélectionnez Modifier.
4. Dans la section Référentiels en amont, recherchez le référentiel en amont que vous souhaitez supprimer et choisissez Supprimer.

5. Lorsque vous avez terminé de supprimer les référentiels en amont, choisissez Enregistrer.

## Connexion à des référentiels externes publics

Vous pouvez connecter les référentiels de CodeCatalyst packages aux référentiels publics externes pris en charge en ajoutant le référentiel de passerelle correspondant en tant que référentiel en amont. Les référentiels Gateway agissent comme des référentiels intermédiaires qui recherchent et stockent des packages extraits de référentiels externes. Cela permet d'économiser du temps et de transférer des données, car tous les référentiels de packages d'un projet utilisent des packages provenant du référentiel de passerelle.

Pour vous connecter à un référentiel public à l'aide de référentiels de passerelle

1. Dans le panneau de navigation, choisissez Packages.
2. Sur la page Référentiels de packages, dans Référentiels de passerelle, vous pouvez consulter la liste des référentiels de passerelle pris en charge et leurs descriptions. Pour utiliser un référentiel de passerelle, vous devez d'abord le créer. Si le référentiel de passerelle a été créé, la date et l'heure de sa création sont affichées. Si ce n'est pas le cas, choisissez Create pour le créer.
3. Choisissez le référentiel de packages que vous souhaitez connecter au référentiel public.
4. Choisissez le menu déroulant Actions, puis sélectionnez Modifier.
5. Pour vous connecter au référentiel public, ajoutez le référentiel de passerelle qui correspond au référentiel public auquel vous souhaitez vous connecter en tant que référentiel en amont.

Dans la section Modifier les référentiels en amont, choisissez Ajouter un CodeCatalyst référentiel.

6. La section Référentiels de passerelle répertorie tous les référentiels de passerelle disponibles. Lorsque vous trouvez le référentiel de passerelle correspondant au référentiel externe public auquel vous souhaitez vous connecter, sélectionnez-le dans la liste, puis choisissez Ajouter.
7. Lorsqu'un package est demandé à partir d'un référentiel, CodeCatalyst recherche les référentiels en amont dans l'ordre dans lequel ils apparaissent dans la liste Modifier les référentiels en amont. Lorsqu'un package est trouvé, la recherche CodeCatalyst s'arrête. Pour modifier l'ordre des référentiels en amont, faites-les glisser dans la liste ou utilisez les flèches de réorganisation.
8. Lorsque vous avez terminé d'ajouter et de commander des référentiels en amont, choisissez Enregistrer.

Lorsque vous avez ajouté un référentiel de passerelle en tant que référentiel en amont, vous pouvez utiliser un gestionnaire de packages connecté à votre référentiel local pour récupérer les packages depuis le référentiel de packages externe public qui lui correspond. Il n'est pas nécessaire de mettre à jour la configuration de votre gestionnaire de packages. Les packages consommés de cette manière sont stockés à la fois dans le référentiel de passerelle et dans votre référentiel de packages local. Pour plus d'informations sur la demande de versions de package à partir d'un référentiel en amont, consultez [Demande d'une version de package avec des référentiels en amont](#).

## Référentiels de packages externes pris en charge et leurs référentiels de passerelle

CodeCatalyst prend en charge l'ajout d'une connexion en amont aux autorités de package officielles suivantes avec des référentiels de passerelle.

Type de package de référentiel	Description	Nom du référentiel Gateway
npm	registre public npm	npm-public-registry-gateway

## Travailler avec des packages

Un package contient CodeCatalyst un ensemble de logiciels et les métadonnées nécessaires pour résoudre les dépendances et installer le logiciel. CodeCatalyst prend en charge le format de package npm. Cette section fournit des informations sur la publication, l'affichage et la suppression de packages, ainsi que sur la mise à jour du statut d'une version de package.

### Rubriques

- [Publication de packages](#)
- [Afficher les détails de la version du package](#)
- [Supprimer une version de package](#)
- [Mettre à jour le statut d'une version de package](#)
- [Modification des contrôles d'origine des packages](#)

## Publication de packages

Vous pouvez publier des versions de n'importe quel type de package pris en charge dans un référentiel de CodeCatalyst packages à l'aide des outils du gestionnaire de packages.

Pour plus d'informations sur la connexion d'un gestionnaire de packages à un CodeCatalyst référentiel de packages pour publier des packages, consultez [Connexion à un référentiel de packages](#).

### Table des matières

- [Publication et référentiels en amont](#)
- [Packages privés et référentiels publics](#)
- [Remplacement des actifs du package](#)

## Publication et référentiels en amont

Dans CodeCatalyst, vous ne pouvez pas publier les versions de package présentes dans des référentiels en amont ou des référentiels publics accessibles. Supposons, par exemple, que vous souhaitez publier un package npm dans un référentiel de packages et myrepo que vous soyez myrepo connecté à npmjs.com via un référentiel de passerelle configuré en tant que référentiel en amont. `lodash@1.0` S'il `lodash@1.0` est présent dans le référentiel en amont ou dans npmjs.com, CodeCatalyst rejette toute tentative de publication dans ce référentiel en myrepo émettant une erreur de conflit 409. Cela vous permet d'éviter de publier accidentellement un package portant le même nom et la même version qu'un package dans un référentiel en amont, ce qui peut entraîner un comportement inattendu.

Vous pouvez toujours publier les différentes versions d'un nom de package qui existent dans un référentiel en amont. Par exemple, s'il `lodash@1.0` est présent dans un référentiel en amont, mais `lodash@1.1` qu'il ne l'est pas, vous pouvez `lodash@1.1` publier dans le référentiel en aval.

## Packages privés et référentiels publics

CodeCatalyst ne publie pas les packages stockés dans CodeCatalyst des référentiels publics, tels que npmjs.com. CodeCatalyst importe le package des référentiels publics vers un CodeCatalyst référentiel, mais il ne déplace pas les packages dans la direction opposée. Les packages que vous publiez dans CodeCatalyst des référentiels restent privés et ne sont disponibles que pour le CodeCatalyst projet auquel appartient le référentiel.

## Remplacement des actifs du package

Vous ne pouvez pas republier une ressource de package qui existe déjà avec un contenu différent. Étant donné que npm ne prend en charge qu'un seul actif par version de package, pour modifier une version de package publiée, vous devez d'abord la supprimer.

## Afficher les détails de la version du package

Vous pouvez utiliser la CodeCatalyst console pour afficher les détails d'une version de package spécifique.

Pour afficher les détails de la version du package

1. Dans le panneau de navigation, choisissez Packages.
2. Sur la page Référentiels de packages, choisissez le référentiel contenant la version du package dont vous souhaitez afficher les détails.
3. Recherchez la version du package dans le tableau Packages. Vous pouvez utiliser la barre de recherche pour filtrer les packages par nom de package. Choisissez le package dans la liste.
4. Sur la page Détails du package, choisissez Versions, puis choisissez la version que vous souhaitez consulter.

## Supprimer une version de package

Vous pouvez supprimer une version du package depuis la page des détails de la version du package de la CodeCatalyst console.

Pour supprimer une version de package

1. Dans le panneau de navigation, choisissez Packages.
2. Sur la page Référentiels de packages, choisissez le référentiel contenant la version du package que vous souhaitez supprimer.
3. Recherchez et choisissez le package dans le tableau.
4. Sur la page Détails du package, choisissez Versions et choisissez la version que vous souhaitez supprimer.
5. Sur la page des détails de la version du package, sélectionnez Actions de version, puis sélectionnez Supprimer.
6. Entrez Supprimer dans le champ de texte et choisissez Supprimer.

## Mettre à jour le statut d'une version de package

Chaque version de package CodeCatalyst possède un statut qui décrit l'état actuel et la disponibilité de la version du package. Vous pouvez modifier le statut de la version du package dans la CodeCatalyst console. Pour plus d'informations sur les valeurs d'état possibles des versions de package et leur signification, consultez [État de la version du package](#).

Pour mettre à jour le statut d'une version de package

1. Dans le panneau de navigation, choisissez Packages.
2. Sur la page Référentiels de packages, choisissez le référentiel contenant la version du package dont vous souhaitez mettre à jour le statut.
3. Recherchez et choisissez le package dans le tableau.
4. Sur la page Détails du package, choisissez Versions, puis choisissez la version que vous souhaitez consulter.
5. Sur la page des détails de la version du package, sélectionnez Actions, puis choisissez Délistier, Archiver ou Supprimer. Pour plus d'informations sur le statut de chaque version de package, consultez [État de la version du package](#).
6. Entrez le texte de confirmation dans le champ de texte, puis choisissez Dérépertorier, Archiver ou Supprimer, selon le statut vers lequel vous souhaitez effectuer la mise à jour.

### État de la version du package

Les valeurs possibles pour le statut de version du package sont les suivantes. Vous pouvez modifier le statut de la version du package dans la console. Pour plus d'informations, consultez [Mettre à jour le statut d'une version de package](#).

- **Publié** : La version du package a été publiée avec succès et peut être demandée par un gestionnaire de packages. La version du package sera incluse dans les listes de versions de package renvoyées aux gestionnaires de packages ; par exemple, dans la sortie `denpm view <package-name> versions`. Toutes les ressources de la version du package sont disponibles dans le référentiel.
- **Non répertorié** : les ressources de la version du package peuvent être téléchargées depuis le référentiel, mais la version du package n'est pas incluse dans la liste des versions renvoyées aux gestionnaires de packages. Par exemple, pour un package npm, la sortie de `n'npm view <package-name> versions` inclut pas la version du package. Cela signifie que la logique de



résolution des dépendances npm ne sélectionne pas la version du package car celle-ci n'apparaît pas dans la liste des versions disponibles. Toutefois, si la version du package non répertorié est déjà référencée dans un `npm package-lock.json` fichier, elle peut toujours être téléchargée et installée, par exemple lors de l'exécution `npm ci`.

- **Archivé** : Les ressources de la version du package ne peuvent pas être téléchargées. La version du package ne sera pas incluse dans la liste des versions renvoyées aux gestionnaires de packages. Comme les actifs ne sont pas disponibles, la consommation de la version du package par les clients est bloquée. Si le build de votre application dépend d'une version mise à jour vers Archivé, le build échouera, sauf si la version du package a été mise en cache localement. Vous ne pouvez pas utiliser un gestionnaire de packages ou un outil de génération pour republier une version de package archivée, car celle-ci est toujours présente dans le référentiel. Cependant, vous pouvez rétablir le statut de la version du package sur Non répertorié ou Publié dans la console.
- **Supprimé** : la version du package n'apparaît pas dans les listes et les ressources ne peuvent pas être téléchargées depuis le référentiel. La principale différence entre Disposé et Archivé est que lorsque le statut est Disposé, les actifs de la version du package sont définitivement supprimés par CodeCatalyst. Pour cette raison, vous ne pouvez pas déplacer une version de package du statut Disposé vers Archivé, Non répertorié ou Publié. La version du package ne peut pas être utilisée car les actifs ont été supprimés. Lorsqu'une version du package est marquée comme supprimée, le stockage des actifs du package ne vous est pas facturé.

Outre les statuts de la liste précédente, une version de package peut également être supprimée. Après sa suppression, une version de package ne se trouve pas dans le référentiel et vous pouvez librement republier cette version de package à l'aide d'un gestionnaire de packages ou d'un outil de génération.

## Modification des contrôles d'origine des packages

Sur Amazon CodeCatalyst, les versions de packages peuvent être ajoutées à un référentiel de packages en les publiant directement, en les extrayant d'un référentiel en amont ou en les ingérant depuis un référentiel public externe. Si vous autorisez l'ajout de versions d'un package à la fois par publication directe et par ingestion à partir de référentiels publics, vous êtes vulnérable à une attaque de substitution de dépendance. Pour plus d'informations, consultez [Attaques de substitution de la dépendance](#). Pour vous protéger contre une attaque de substitution de dépendance, configurez les contrôles d'origine des packages sur un package dans un référentiel afin de limiter la manière dont les versions de ce package peuvent être ajoutées au référentiel.

Vous devriez envisager de configurer les contrôles d'origine des packages pour que les nouvelles versions des différents packages proviennent à la fois de sources internes, telles que la publication directe, et de sources externes, telles que des référentiels publics. Par défaut, les contrôles d'origine des packages sont configurés en fonction de la manière dont la première version d'un package est ajoutée au référentiel.

## Paramètres de contrôle de l'origine des packages

Grâce aux contrôles de l'origine des packages, vous pouvez configurer la manière dont les versions des packages peuvent être ajoutées à un référentiel. Les listes suivantes incluent les paramètres et valeurs de contrôle d'origine des packages disponibles.

### Publish

Ce paramètre permet de configurer si les versions des packages peuvent être publiées directement dans le référentiel à l'aide de gestionnaires de packages ou d'outils similaires.

- **AUTORISER** : les versions du package peuvent être publiées directement.
- **BLOCK** : Les versions du package ne peuvent pas être publiées directement.

### En amont

Ce paramètre définit si les versions des packages peuvent être ingérées à partir de référentiels publics externes ou conservées à partir de référentiels en amont à la demande d'un gestionnaire de packages.

- **AUTORISER** : Toute version de package peut être conservée à partir d'autres CodeCatalyst référentiels configurés en tant que référentiels en amont ou ingérée à partir d'une source publique via une connexion externe.
- **BLOCAGE** : Les versions de package ne peuvent pas être conservées à partir d'autres CodeCatalyst référentiels configurés comme référentiels en amont ou ingérées à partir d'une source publique avec une connexion externe.

### Paramètres de contrôle de l'origine des packages par défaut

Les contrôles d'origine des packages par défaut pour un package seront basés sur la manière dont la première version de ce package est ajoutée au référentiel de packages.

- Si la première version du package est publiée directement par un gestionnaire de packages, les paramètres seront Publish : ALLOW et Upstream : BLOCK.
- Si la première version du package est ingérée à partir d'une source publique, les paramètres seront Publish : BLOCK et Upstream : ALLOW.

## Scénarios courants de contrôle d'accès aux packages

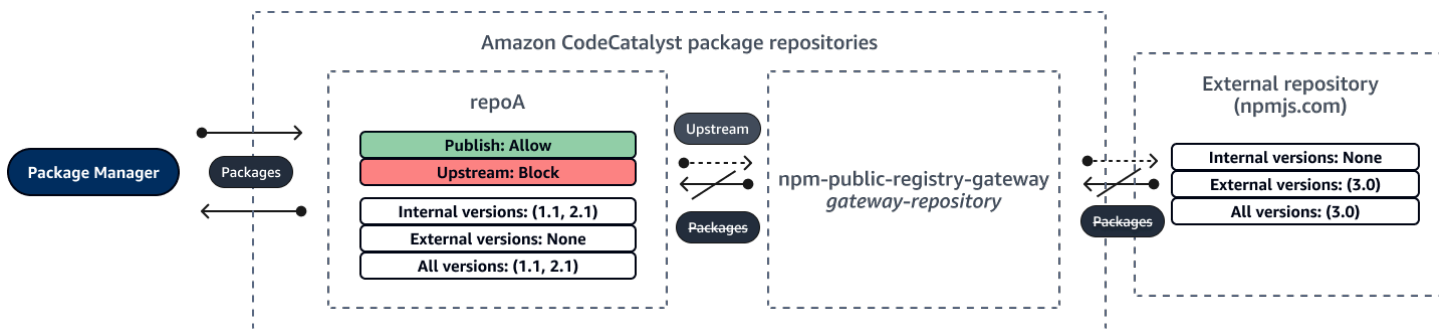
Cette section décrit certains scénarios courants d'ajout d'une version de package à un référentiel de CodeCatalyst packages. Les paramètres de contrôle de l'origine des packages sont définis pour les nouveaux packages en fonction de la manière dont la première version du package est ajoutée.

Dans les scénarios suivants, un package interne est publié directement depuis un gestionnaire de packages vers votre référentiel, tel qu'un package que vous gérez. Un package externe est un package existant dans un dépôt public qui peut être ingéré dans votre référentiel via une connexion externe.

Une version de package externe est publiée pour un package interne existant

Dans ce scénario, considérez un package interne, PackageA. Votre équipe publie la première version du package pour PackageA dans un référentiel de CodeCatalyst packages. Comme il s'agit de la première version de package pour ce package, les paramètres de contrôle de l'origine du package sont automatiquement définis sur Publier : Autoriser et Upstream : Bloquer. Une fois le package publié dans votre référentiel, un package portant le même nom est publié dans un référentiel public connecté à votre référentiel de CodeCatalyst packages. Il peut s'agir d'une tentative d'attaque de substitution de dépendance contre le package interne ou d'une coïncidence. Quoi qu'il en soit, les contrôles d'origine des packages sont configurés pour bloquer l'ingestion de la nouvelle version externe afin de se protéger contre une attaque potentielle.

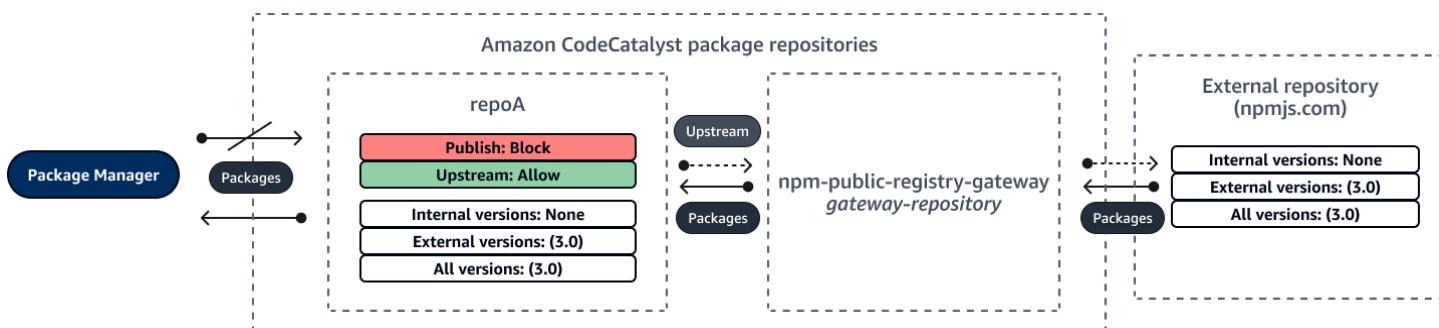
Dans l'image suivante, RePoA est votre dépôt de CodeCatalyst packages avec une connexion externe à un dépôt public. Votre dépôt contient les versions 1.1 et 2.1 de PackageA, mais la version 3.0 est publiée dans le dépôt public. Normalement, RePoA ingère la version 3.0 une fois que le package a été demandé par un gestionnaire de packages. L'ingestion de packages étant définie sur Bloquer, la version 3.0 n'est pas ingérée dans votre référentiel de CodeCatalyst packages et n'est pas disponible pour les gestionnaires de packages qui y sont connectés.



Une version de package interne est publiée pour un package externe existant

Dans ce scénario, un package, PackageB, existe en externe dans un référentiel public que vous avez connecté à votre référentiel. Lorsqu'un gestionnaire de packages connecté à votre référentiel demande PackageB, la version du package est ingérée dans votre référentiel depuis le référentiel public. Comme il s'agit de la première version de package de PackageB ajoutée à votre référentiel, les paramètres d'origine du package sont configurés sur Publish : BLOCK et Upstream : ALLOW. Plus tard, vous essayez de publier une version portant le même nom de package dans le référentiel. Vous ne connaissez peut-être pas le package public et vous essayez de publier un package indépendant portant le même nom, ou vous essayez peut-être de publier une version corrigée, ou vous essayez peut-être de publier directement la version exacte du package qui existe déjà en externe. CodeCatalyst rejette la version que vous essayez de publier, mais vous pouvez annuler explicitement le rejet et publier la version, si nécessaire.

Dans l'image suivante, RePoA est votre dépôt de CodeCatalyst packages avec une connexion externe à un dépôt public. Votre dépôt de packages contient la version 3.0 qu'il a ingérée depuis le dépôt public. Vous souhaitez publier la version 1.2 dans votre référentiel de packages. Généralement, vous pouvez publier la version 1.2 sur RePoA, mais comme la publication est définie sur Bloquer, la version 1.2 ne peut pas être publiée.



Publication d'une version de package corrigée d'un package externe existant

Dans ce scénario, un package, PackageB, existe en externe dans un référentiel public que vous avez connecté à votre référentiel de packages. Lorsqu'un gestionnaire de packages connecté à votre référentiel demande PackageB, la version du package est ingérée dans votre référentiel depuis le référentiel public. Comme il s'agit de la première version de package de PackageB ajoutée à votre référentiel, les paramètres d'origine du package sont configurés sur Publish : BLOCK et Upstream : ALLOW. Votre équipe décide de publier les versions corrigées de ce package dans le référentiel. Pour pouvoir publier directement les versions des packages, votre équipe modifie les paramètres de contrôle de l'origine des packages en Publish : ALLOW et Upstream : BLOCK. Les versions de ce package peuvent désormais être publiées directement dans votre dépôt et ingérées à partir de référentiels publics. Une fois que votre équipe a publié les versions du package corrigées, elle rétablit les paramètres d'origine du package sur Publish : BLOCK et Upstream : ALLOW.

## Modification des contrôles d'origine des packages

Les contrôles d'origine des packages sont configurés automatiquement en fonction de la manière dont la première version d'un package est ajoutée au référentiel de packages. Pour plus d'informations, consultez [Paramètres de contrôle de l'origine des packages par défaut](#). Pour ajouter ou modifier des contrôles d'origine de package pour un package dans un référentiel de CodeCatalyst packages, effectuez les étapes de la procédure suivante.

Pour ajouter ou modifier les contrôles d'origine des packages

1. Dans le panneau de navigation, choisissez Packages.
2. Choisissez le référentiel de packages qui contient le package que vous souhaitez modifier.
3. Dans le tableau Packages, recherchez et choisissez le package que vous souhaitez modifier.
4. Sur la page récapitulative du package, choisissez Modifier les contrôles d'origine.
5. Dans Contrôles d'origine, choisissez les contrôles d'origine du package que vous souhaitez définir pour ce package. Les deux paramètres de contrôle de l'origine du package, Publish et Upstream, doivent être définis en même temps.
  - Pour autoriser la publication directe des versions du package, dans Publier, sélectionnez Autoriser. Pour bloquer la publication des versions du package, choisissez Bloquer.
  - Pour autoriser l'ingestion de packages provenant de référentiels externes et l'extraction de packages depuis des référentiels en amont, dans Sources en amont, sélectionnez Autoriser. Pour bloquer toute ingestion et extraction de versions de packages depuis des référentiels externes et en amont, choisissez Bloquer.
6. Choisissez Enregistrer.

## Publication et référentiels en amont

Dans CodeCatalyst, vous ne pouvez pas publier les versions de package présentes dans des référentiels en amont ou des référentiels publics accessibles. Par exemple, supposons que vous souhaitez publier un package `lodash@1.0` npm dans un référentiel et `myrepo` que vous disposiez d'un référentiel en amont avec une connexion externe à `npmjs.com`. `myrepo` Envisagez les scénarios suivants.

1. Les paramètres de contrôle de l'origine du package `lodash` sont `Publish : ALLOW` et `Upstream : ALLOW`. S'il `lodash@1.0` est présent dans le référentiel en amont ou dans `npmjs.com`, CodeCatalyst rejette toute tentative de publication dans ce référentiel en `myrepo` émettant une erreur de conflit 409. Vous pouvez toujours publier une version différente, telle que `lodash@1.1`.
2. Les paramètres de contrôle de l'origine du package `lodash` sont `Publish : ALLOW` et `Upstream : BLOCK`. Vous pouvez publier n'importe quelle version `lodash` de dans votre référentiel qui n'existe pas encore car les versions des packages ne sont pas accessibles.
3. Les paramètres de contrôle de l'origine du package `lodash` sont `Publish : BLOCK` et `Upstream : ALLOW`. Vous ne pouvez publier aucune version de package directement dans votre référentiel.

## Attaques de substitution de la dépendance

Les gestionnaires de packages simplifient le processus d'empaquetage et de partage de code réutilisable. Ces packages peuvent être des packages privés développés par une organisation pour être utilisés dans ses applications, ou il peut s'agir de packages publics, généralement open source, développés en dehors d'une organisation et distribués par des référentiels de packages publics. Lorsqu'ils demandent des packages, les développeurs s'appuient sur leur gestionnaire de packages pour récupérer les nouvelles versions de leurs dépendances. Les attaques de substitution de dépendances, également appelées attaques de confusion des dépendances, exploitent le fait qu'un gestionnaire de packages n'a généralement aucun moyen de distinguer les versions légitimes d'un package des versions malveillantes.

Les attaques par substitution de dépendance appartiennent à un sous-ensemble d'attaques connues sous le nom d'attaques de la chaîne logistique logicielle. Une attaque de chaîne d'approvisionnement logicielle est une attaque qui tire parti de vulnérabilités situées à n'importe quel niveau de la chaîne d'approvisionnement logicielle.

Une attaque de substitution de dépendance peut cibler toute personne utilisant à la fois des packages développés en interne et des packages extraits de référentiels publics. Les attaquants identifient

les noms de packages internes, puis placent stratégiquement le code malveillant portant le même nom dans des référentiels de packages publics. Généralement, le code malveillant est publié dans un package dont le numéro de version est élevé. Les gestionnaires de packages récupèrent le code malveillant à partir de ces flux publics car ils pensent que les packages malveillants sont les dernières versions du package. Cela provoque une « confusion » ou une « substitution » entre le package souhaité et le package malveillant, ce qui entraîne la compromission du code.

Pour empêcher les attaques de substitution de dépendance, Amazon CodeCatalyst fournit des contrôles de l'origine des colis. Les contrôles d'origine des packages sont des paramètres qui contrôlent la manière dont les packages peuvent être ajoutés à vos référentiels. Les contrôles sont configurés automatiquement lorsque la première version d'un nouveau package est ajoutée à un CodeCatalyst référentiel. Ils peuvent garantir que les versions des packages ne peuvent pas être publiées directement dans votre référentiel et ingérées à partir de sources publiques, ce qui vous protège des attaques de substitution de dépendances. Pour plus d'informations sur les contrôles d'origine des packages et sur la manière de les modifier, consultez [Modification des contrôles d'origine des packages](#).

## Utilisation de npm

Ces rubriques décrivent comment vous pouvez utiliser npm le gestionnaire de packages Node.js avec CodeCatalyst.

### Note

CodeCatalyst supports node v4.9.1 et plus tard npm v5.0.0 et plus tard.

### Rubriques

- [Configuration et utilisation de npm](#)
- [gestion des balises npm](#)

## Configuration et utilisation de npm

Pour l'utiliser npm avec CodeCatalyst, vous devez vous connecter npm à votre référentiel de packages et fournir un jeton d'accès personnel (PAT) pour l'authentification. Vous pouvez consulter les instructions de connexion npm à votre référentiel de packages dans la CodeCatalyst console.

## Table des matières

- [Configuration de npm avec CodeCatalyst](#)
- [Installation de packages npm à partir d'un référentiel CodeCatalyst de packages](#)
- [Installation de packages npm depuis npmjs via CodeCatalyst](#)
- [Publication de packages npm dans votre référentiel de CodeCatalyst packages](#)
- [support des commandes npm](#)
  - [Commandes prises en charge qui interagissent avec un référentiel de packages](#)
  - [Commandes côté client prises en charge](#)
  - [Commandes non prises en charge](#)

## Configuration de npm avec CodeCatalyst

Les instructions suivantes expliquent comment vous authentifier et vous connecter npm à votre référentiel de CodeCatalyst packages. Pour plus d'informations sur npm, consultez la documentation [officielle de npm](#).

Pour vous connecter **npm** à votre référentiel de CodeCatalyst packages

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre projet.
3. Dans le panneau de navigation, choisissez Packages.
4. Choisissez votre référentiel de packages dans la liste.
5. Choisissez Connect to repository.
6. Dans Détails de configuration, dans Package manager client, choisissez npm client.
7. Choisissez votre système d'exploitation pour afficher les étapes de configuration correspondantes.
8. Un jeton d'accès personnel (PAT) est requis pour authentifier npm avec CodeCatalyst. Si vous avez déjà un jeton, vous pouvez l'utiliser. Si ce n'est pas le cas, vous pouvez en créer un en suivant les étapes ci-dessous.
  - a. (Facultatif) : Mettez à jour le nom PAT et la date d'expiration.
  - b. Choisissez Créer un jeton.
  - c. Copiez et stockez votre PAT dans un endroit sûr.



**⚠ Warning**

Vous ne pourrez plus voir ou copier votre PAT une fois que vous aurez fermé la boîte de dialogue. Les informations d'identification doivent être de courte durée afin de minimiser le temps pendant lequel un attaquant peut les utiliser après les avoir détournées.

9. Exécutez les commandes suivantes depuis le répertoire racine de votre projet pour configurer npm avec votre référentiel de packages. Les commandes effectueront les opérations suivantes :
  - Créez un `.npmrc` fichier au niveau du projet si votre projet n'en possède pas.
  - Ajoutez les informations de point de terminaison du référentiel de packages à votre fichier au niveau du projet `.npmrc`.
  - Ajoutez vos informations d'identification (PAT) à votre `.npmrc` fichier de niveau utilisateur.

Remplacez les valeurs suivantes.

**i Note**

Si vous copiez les instructions de la console, les valeurs des commandes suivantes sont mises à jour pour vous et il n'est pas nécessaire de les modifier.

- Remplacez le *nom d'utilisateur* par votre nom CodeCatalyst d'utilisateur.
- Remplacez *PAT* par votre CodeCatalyst PAT.
- Remplacez *space\_name par le nom* de votre CodeCatalyst espace.
- Remplacez *proj\_name par le nom* de votre CodeCatalyst projet.
- Remplacez *repo\_name par le nom* CodeCatalyst du dépôt de votre package.

```
npm set registry=https://packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/ --location project
npm set //packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/:_authToken=username:PAT
```

Pour npm 6 ou inférieur : pour que npm transmette toujours le jeton d'authentification à CodeCatalyst, même pour les GET demandes, définissez la variable de configuration `always-auth` comme suit. `npm config set`

```
npm set //packages.region.codecatalyst.aws/npm/space-name/proj-name/repo-name/:always-auth=true --location project
```

## Installation de packages npm à partir d'un référentiel CodeCatalyst de packages

Après avoir connecté npm à votre référentiel en suivant les étapes décrites dans [Configuration de npm avec CodeCatalyst](#), vous pouvez exécuter des npm commandes sur votre référentiel.

Vous pouvez installer un package npm qui se trouve dans votre référentiel de CodeCatalyst packages ou dans l'un de ses référentiels en amont à l'aide de la `npm install` commande.

```
npm install Lodash
```

## Installation de packages npm depuis npmjs via CodeCatalyst

Vous pouvez installer des packages npm depuis [npmjs.com](#) via un CodeCatalyst référentiel en configurant le référentiel avec une connexion en amont au référentiel de passerelle connecté à npmjs.com, . npm-public-registry-gateway Les packages installés à partir de npmjs sont ingérés et stockés dans le référentiel de passerelle, le référentiel de packages le plus éloigné en aval.

Pour installer des packages depuis npmjs

1. Si ce n'est pas déjà fait, configurez npm avec votre référentiel de CodeCatalyst packages en suivant les étapes décrites dans [Configuration de npm avec CodeCatalyst](#).
2. Vérifiez que votre référentiel a ajouté le référentiel de passerelle npm-public-registry-gateway, en tant que connexion en amont. Vous pouvez vérifier quelles sources en amont sont ajoutées ou ajoutées en npm-public-registry-gateway tant que source en amont en suivant les instructions [Ajouter un référentiel en amont](#) et en choisissant le npm-public-registry-gateway référentiel.
3. Installez les packages à l'aide de la `npm install` commande.

```
npm install package_name
```

Pour plus d'informations sur la demande de packages à partir de référentiels en amont, consultez [Demande d'une version de package avec des référentiels en amont](#).

## Publication de packages npm dans votre référentiel de CodeCatalyst packages

Une fois que vous avez terminé [Configuration de npm avec CodeCatalyst](#), vous pouvez exécuter npm des commandes.

Vous pouvez publier un package npm dans un référentiel de CodeCatalyst packages à l'aide de la `npm publish` commande.

```
npm publish
```

Pour plus d'informations sur la création de packages npm, voir [Création de modules Node.js](#) sur npm Docs.

## support des commandes npm

Les sections suivantes résument les npm commandes prises en charge par les référentiels de CodeCatalyst packages, en plus de répertorier les commandes spécifiques qui ne sont pas prises en charge.

### Rubriques

- [Commandes prises en charge qui interagissent avec un référentiel de packages](#)
- [Commandes côté client prises en charge](#)
- [Commandes non prises en charge](#)

### Commandes prises en charge qui interagissent avec un référentiel de packages

Cette section répertorie npm les commandes dans lesquelles le npm client envoie une ou plusieurs requêtes au registre pour lequel il est configuré (par exemple, `npm config set registry`). Il a été vérifié que ces commandes fonctionnent correctement lorsqu'elles sont invoquées dans un référentiel de CodeCatalyst packages.

Command	Description
<a href="#">punaises</a>	Devine l'emplacement de l'URL de suivi des bogues d'un package, puis tente de l'ouvrir.

Command	Description
<a href="#">ci</a>	Installe un projet sur une table rase.
<a href="#">déprécier</a>	Déprécie une version d'un package.
<a href="#">dist-tag</a>	Modifie les balises de distribution des packages.
<a href="#">documents</a>	Devine l'emplacement de l'URL de documentation d'un package, puis tente de l'ouvrir à l'aide du paramètre de <code>--browser</code> configuration.
<a href="#">médecin</a>	Exécute un ensemble de vérifications pour valider que votre installation npm peut gérer vos JavaScript packages.
<a href="#">installer</a>	Installe un package.
<a href="#">install-ci-test</a>	Installe un projet sur une table rase et exécute des tests. Pseudonyme : <code>npm ci</code> . Cette commande exécute <code>npm ci</code> , suivi immédiatement d' <code>npm test</code> .
<a href="#">installation-test</a>	Installe le package et exécute des tests. Exécute <code>npm install</code> , suivi immédiatement d' <code>npm test</code> .
<a href="#">désuet</a>	Vérifie le registre configuré pour déterminer si les packages installés sont obsolètes.
<a href="#">ping</a>	Envoie une requête ping au registre npm configuré ou donné et vérifie l'authentification.
<a href="#">publier</a>	Publie une version du package dans le registre.
<a href="#">update</a>	Devine l'emplacement de l'URL du dépôt d'un package, puis tente de l'ouvrir à l'aide du paramètre de <code>--browser</code> configuration.

Command	Description
<a href="#">vue</a>	Affiche les métadonnées du package. Peut également être utilisé pour imprimer les propriétés des métadonnées.

## Commandes côté client prises en charge

Ces commandes ne nécessitent aucune interaction directe avec un dépôt de packages et CodeCatalyst ne nécessite donc rien pour les prendre en charge.

Command	Description
<a href="#">poubelle (héritage)</a>	Affiche le <code>bin</code> répertoire npm.
<a href="#">construire</a>	Construit un package.
<a href="#">cache</a>	Manipule le cache des packages.
<a href="#">achèvement</a>	Permet de compléter les onglets dans toutes les commandes npm.
<a href="#">configuration</a>	Met à jour le contenu des <code>npmrc</code> fichiers utilisateur et globaux.
<a href="#">dédupliquez</a>	Effectue une recherche dans l'arborescence des packages locaux et tente de simplifier la structure en déplaçant les dépendances plus haut dans l'arborescence, où elles peuvent être partagées plus efficacement par plusieurs packages dépendants.
<a href="#">modifier</a>	Modifie un package installé. Sélectionne une dépendance dans le répertoire de travail actuel et ouvre le répertoire du package dans l'éditeur par défaut.

Command	Description
<a href="#">explorez</a>	Parcourt un package installé. Génère un sous-shell dans le répertoire du package installé spécifié. Si une commande est spécifiée, elle est exécutée dans le sous-shell, qui s'arrête alors immédiatement.
<a href="#">help</a>	Obtient de l'aide sur npm.
<a href="#">aide-recherche</a>	Recherche la documentation d'aide de npm.
<a href="#">initialisation</a>	Crée un package .json fichier.
<a href="#">lien</a>	Établit des liens symboliques vers un répertoire de packages.
<a href="#">ls</a>	Répertorie les packages installés.
<a href="#">emballer</a>	Crée une archive tar à partir d'un package.
<a href="#">prefix</a>	Affiche un préfixe. Il s'agit du répertoire parent le plus proche à contenir un package .json fichier, sauf indication contraire -g.
<a href="#">pruneau</a>	Supprime les packages qui ne figurent pas dans la liste des dépendances du package parent.
<a href="#">reconstruire</a>	Exécute la npm build commande sur les dossiers correspondants.
<a href="#">redémarrer</a>	Exécute les scripts d'arrêt, de redémarrage et de démarrage d'un package ainsi que les pré-scripts et post-scripts associés.
<a href="#">racine</a>	Imprime le node_modules répertoire effectif en sortie standard.
<a href="#">exécuter un script</a>	Exécute des scripts de package arbitraires.

Command	Description
<a href="#">shrinkwrap</a>	Verrouille les versions de dépendance pour publication.
<a href="#">désinstaller</a>	Désinstalle un package.

### Commandes non prises en charge

Ces npm commandes ne sont pas prises en charge par les référentiels de CodeCatalyst packages.

Command	Description	Remarques
<a href="#">accès</a>	Définit le niveau d'accès aux packages publiés.	CodeCatalyst utilise un modèle d'autorisation différent de celui du référentiel public npmjs.
<a href="#">ajouter un utilisateur</a>	Ajoute un compte utilisateur au registre	CodeCatalyst utilise un modèle utilisateur différent de celui du dépôt public npmjs.
<a href="#">audit</a>	Exécute un audit de sécurité.	CodeCatalyst ne vend actuellement pas de données relatives aux failles de sécurité.
<a href="#">crochet</a>	Gère les hooks npm, y compris l'ajout, la suppression, la liste et la mise à jour.	CodeCatalyst ne prend actuellement en charge aucun mécanisme de notification des modifications.
<a href="#">login</a>	Authentifie un utilisateur. Ceci est un alias pour npm adduser.	CodeCatalyst utilise un modèle d'authentification différent de celui du dépôt public npmjs. Pour plus d'informations, consultez

Command	Description	Remarques
		<a href="#">Configuration de npm avec CodeCatalyst.</a>
<a href="#">logout</a>	Se déconnecte du registre.	CodeCatalyst utilise un modèle d'authentification différent de celui du dépôt public npmjs. Il n'est pas possible de se déconnecter d'un CodeCatalyst référentiel, mais les jetons d'authentification expirent après leur date d'expiration configurable. La durée du jeton par défaut est de 12 heures.
<a href="#">propriétaire</a>	Gère les propriétaires de packages.	CodeCatalyst utilise un modèle d'autorisations différent de celui du référentiel public npmjs.
<a href="#">profile</a>	Modifie les paramètres de votre profil de registre.	CodeCatalyst utilise un modèle utilisateur différent de celui du dépôt public npmjs.
<a href="#">search</a>	Recherche dans le registre les packages correspondant aux termes de recherche.	CodeCatalyst ne prend pas en charge la search commande.
<a href="#">étoile</a>	Marque vos packages préférés.	CodeCatalyst ne prend actuellement en charge aucun mécanisme de favoris.
<a href="#">étoiles</a>	Affiche les packages marqués comme favoris.	CodeCatalyst ne prend actuellement en charge aucun mécanisme de favoris.



Command	Description	Remarques
<a href="#">équipe</a>	Gère les équipes et les adhésions aux équipes.	CodeCatalyst utilise un modèle d'adhésion d'utilisateurs et de groupes différent de celui du référentiel public npmjs.
<a href="https://docs.npmjs.com/cli/tokentoken">https://docs.npmjs.com/cli/tokentoken</a>	Gère vos jetons d'authentification.	CodeCatalyst utilise un modèle différent pour obtenir des jetons d'authentification. Pour plus d'informations, consultez <a href="#">Configuration de npm avec CodeCatalyst</a> .
<a href="#">dépublier</a>	Supprime un package du registre.	CodeCatalyst ne prend pas en charge la suppression d'une version de package d'un référentiel à l'aide du client npm. Vous pouvez supprimer un package dans la console.
<a href="#">whoami</a>	Affiche le nom d'utilisateur npm.	CodeCatalyst utilise un modèle utilisateur différent de celui du dépôt public npmjs.

## gestion des balises npm

Les registres npm prennent en charge les balises, qui sont des alias de chaîne pour les versions de packages. Vous pouvez utiliser des balises pour fournir un alias au lieu d'utiliser des numéros de version. Par exemple, vous avez un projet comportant plusieurs flux de développement et vous utilisez une balise différente pour chaque flux (par exemple `stable`, `beta`, `dev`, `canary`). Pour plus d'informations, consultez [dist-tag](#) sur npm Docs.

Par défaut, npm utilise la `latest` balise pour identifier la version actuelle d'un package. `npm install pkg` (sans `@version` ou `@tag` spécificateur) installe la dernière balise. Généralement,

les projets n'utilisent la balise la plus récente que pour les versions stables. D'autres balises sont utilisées pour les versions instables ou préliminaires.

## Modification des balises avec le client npm

Les trois `npm dist-tag` commandes (`add`, `rm`, et `ls`) fonctionnent de la même manière dans les référentiels de CodeCatalyst packages que dans le registre [npm par défaut](#).

## balises npm et référentiels en amont

Lorsque npm les balises d'un package et les versions de ce package sont également présentes dans un référentiel en amont, CodeCatalyst fusionne les balises avant de les renvoyer au client. Par exemple, un référentiel nommé R possède un référentiel en amont nommé U. Le tableau suivant indique les balises d'un package nommé `web-helper` présent dans les deux référentiels.

Référentiel.	Nom du package	Étiquettes du package
R	<code>web-helper</code>	dernier (alias pour la version 1.0.0)
U	<code>web-helper</code>	alpha (alias pour la version 1.0.1)

Dans ce cas, lorsque le client npm récupère les balises du `web-helper` package depuis le référentiel R, il reçoit à la fois les balises les plus récentes et les balises alpha. Les versions vers lesquelles pointent les balises ne changeront pas.

Lorsque la même balise est présente sur le même package à la fois dans le référentiel en amont et dans le référentiel local, CodeCatalyst utilise la balise qui a été mise à jour pour la dernière fois. Supposons, par exemple, que les balises de `webhelper` aient été modifiées pour ressembler à ce qui suit.

Référentiel.	Nom du package	Étiquettes du package	Dernière mise à jour
R	<code>web-helper</code>	dernier (alias pour la version 1.0.0)	1er janvier 2023

Référentiel.	Nom du package	Étiquettes du package	Dernière mise à jour
U	web-helper	dernier (alias pour la version 1.0.1)	1er juin 2023

Dans ce cas, lorsque le client npm récupère les balises du package web-helper depuis le référentielR, la dernière balise alias la version 1.0.1 car elle a été mise à jour pour la dernière fois. Cela permet de consommer facilement de nouvelles versions de package dans un référentiel en amont qui ne sont pas encore présentes dans un référentiel local en exécutant `npm update`.

## Quotas pour les packages

Le tableau suivant décrit les quotas et les limites pour les packages sur Amazon CodeCatalyst. Pour plus d'informations sur les quotas sur Amazon CodeCatalyst, consultez [Quotas pour CodeCatalyst](#).

Ressource	Quota par défaut
Référentiels de packages	Maximum de 1 000 par espace.
Référentiels en amont direct	Maximum de 10 par référentiel de packages.
Référentiels de packages recherchés en amont	Maximum de 25 référentiels en amont recherchés par version de package demandée.
Taille du fichier d'actifs du package	Maximum de 5 Go par actif du package.
Actifs du package	Maximum de 150 par version de package.

# Créez, testez et déployez avec des flux de travail dans CodeCatalyst

Après avoir écrit le code de votre application dans [CodeCatalyst un environnement de développement](#) et l'avoir transféré dans votre [référentiel CodeCatalyst source](#), vous êtes prêt à le déployer. Pour ce faire automatiquement, il faut utiliser un flux de travail.

Un flux de travail est une procédure automatisée qui décrit comment créer, tester et déployer votre code dans le cadre d'un système d'intégration et de livraison continues (CI/CD). Un flux de travail définit une série d'étapes, ou d'actions, à effectuer lors de son exécution. Un flux de travail définit également les événements, ou déclencheurs, qui déclenchent le démarrage du flux de travail. Pour configurer un flux de travail, vous devez créer un fichier de définition de flux de travail à l'aide de [l'éditeur visuel ou YAML](#) de la CodeCatalyst console.

## Tip

Pour un aperçu rapide de la manière dont vous pouvez utiliser les flux de travail dans un projet, [créez un projet avec un plan](#). Chaque plan déploie un flux de travail fonctionnel que vous pouvez examiner, exécuter et tester.

## À propos du fichier de définition du flux de travail

Un fichier de définition de flux de travail est un fichier YAML qui décrit votre flux de travail. Le fichier est stocké dans un `~/codecatalyst/workflows/` dossier à la racine de votre [dépôt source](#). Le fichier peut avoir une extension `.yaml` ou `.yml`.

Voici un exemple de fichier de définition de flux de travail simple. Nous expliquons chaque ligne de cet exemple dans le tableau ci-dessous.

```
Name: MyWorkflow
SchemaVersion: 1.0
RunMode: QUEUED
Triggers:
  - Type: PUSH
  Branches:
    - main
```

**Actions:****Build:**

Identifiant: aws/build@v1

**Inputs:****Sources:**

- WorkflowSource

**Configuration:****Steps:**

- Run: docker build -t MyApp:latest .

Line	Description
Name: MyWorkflow	Spécifie le nom du flux de travail. Pour plus d'informations sur la Name propriété, consultez <a href="#">Propriétés de haut niveau</a> .
SchemaVersion: 1.0	Spécifie la version du schéma du flux de travail. Pour plus d'informations sur la SchemaVersion propriété, consultez <a href="#">Propriétés de haut niveau</a> .
RunMode: QUEUED	Indique comment CodeCatalyst gérer plusieurs exécutions. Pour plus d'informations sur le mode d'exécution, consultez <a href="#">Configuration des exécutions en file d'attente, remplacées et parallèles</a> .
Triggers:	Spécifie la logique qui déclenchera l'exécution d'un flux de travail. Pour plus d'informations sur les déclencheurs, consultez <a href="#">Utilisation de déclencheurs</a> .
- Type: PUSH Branches: - main	Indique que le flux de travail doit démarrer chaque fois que vous envoyez du code à la main branche du référentiel source par défaut. Pour plus d'informations sur la source du flux de travail, consultez <a href="#">Travailler avec des sources</a> .

Line	Description
<pre>Actions:</pre>	Définit les tâches à effectuer lors de l'exécution d'un flux de travail. Dans cet exemple, la Actions section définit une action unique appelée Build. Pour plus d'informations sur les actions, consultez <a href="#">Utilisation des actions</a> .
<pre>Build:</pre>	Définit les propriétés de l'Build action. Pour plus d'informations sur l'action de génération, consultez <a href="#">Création à l'aide de flux de travail dans CodeCatalyst</a> .
<pre>Identifiant: aws/build@v1</pre>	Spécifie l'identifiant unique et codé en dur pour l'action de génération.
<pre>Inputs:   Sources:     - WorkflowSource</pre>	Indique que l'action de génération doit rechercher dans le référentiel WorkflowSource source les fichiers dont elle a besoin pour terminer son traitement. Pour plus d'informations, consultez <a href="#">Travailler avec des sources</a> .
<pre>Configuration:</pre>	Contient les propriétés de configuration spécifiques à l'action de génération.
<pre>Steps:   - Run: docker build -t MyApp:latest .</pre>	Indique à l'action de construction de créer une image Docker appelée MyApp et de l'étiqueter avec latest.

Pour obtenir la liste complète de toutes les propriétés disponibles dans le fichier de définition du flux de travail, consultez le [Référence de définition du flux de travail](#).

## Utilisation des éditeurs visuels et YAML de la CodeCatalyst console

Pour créer et modifier le fichier de définition du flux de travail, vous pouvez utiliser votre éditeur préféré, mais nous vous recommandons d'utiliser l'éditeur visuel ou l'éditeur YAML de la

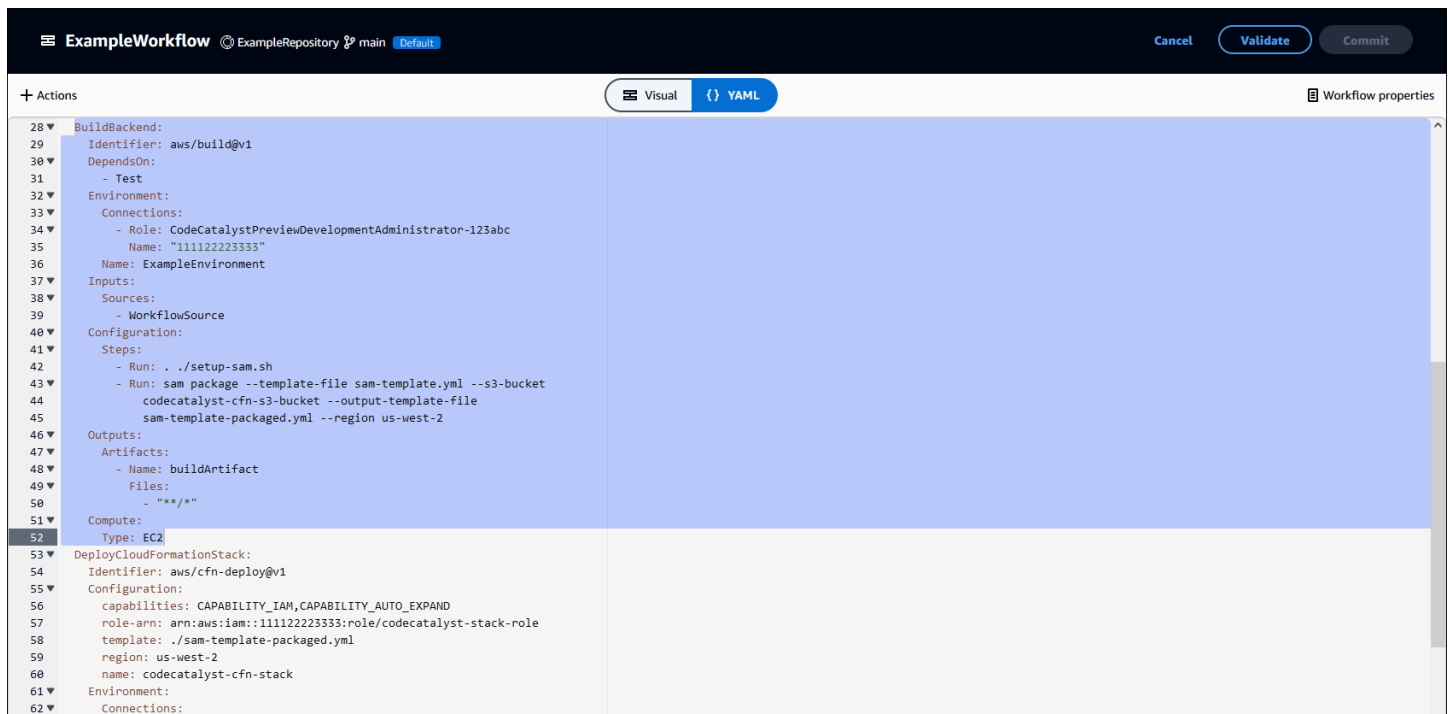
CodeCatalyst console. Ces éditeurs proposent une validation de fichier utile pour garantir l'exactitude des noms de propriétés YAML, des valeurs, de l'imbrication, de l'espacement, des majuscules, etc.

L'image suivante montre un flux de travail dans l'éditeur visuel. L'éditeur visuel vous propose une interface utilisateur complète grâce à laquelle vous pouvez créer et configurer votre fichier de définition de flux de travail. L'éditeur visuel inclut un diagramme de flux de travail (1) présentant les principaux composants du flux de travail, ainsi qu'une zone de configuration (2).

The screenshot displays the Amazon CodeCatalyst console interface. At the top, there is a navigation bar with 'ExampleWorkflow', 'ExampleRepository', and 'main' tabs, along with 'Cancel', 'Validate', and 'Commit' buttons. Below the navigation bar, there are tabs for '+ Actions', 'Visual', and 'YAML'. The main area is divided into two sections:

- Workflow diagram (1):** A visual flowchart showing the workflow steps: 'Source' (workflowSource, ExampleRepository, main) with 'Triggers: Push', followed by 'Test' (aws/managed-test@v1), 'BuildBackend' (aws/build@v1, Environment: ExampleEnvironment, Production), and 'DeployCloudFormationStack' (aws/cfn-deploy@v1, Environment: ExampleEnvironment, Production).
- Configuration area (2):** A detailed configuration panel for the 'BuildBackend' action. It includes sections for 'Inputs', 'Configuration', and 'Outputs'. The 'Configuration' section is active, showing fields for 'Action name' (BuildBackend), 'Compute type' (EC2), 'Compute fleet - optional' (Choose compute), 'Environment/account/role - optional' (Environment: ExampleEnvironment), and 'AWS account connection' (111122223333).

Vous pouvez également utiliser l'éditeur YAML, illustré dans l'image suivante. Utilisez l'éditeur YAML pour coller de gros blocs de code (issus d'un didacticiel, par exemple) ou pour ajouter des propriétés avancées qui ne sont pas proposées par l'éditeur visuel.



```
28 BuildBackend:
29   Identifier: aws/build@v1
30   DependsOn:
31     - Test
32   Environment:
33     Connections:
34       - Role: CodeCatalystPreviewDevelopmentAdministrator-123abc
35         Name: "111122223333"
36     Name: ExampleEnvironment
37   Inputs:
38     Sources:
39       - WorkflowSource
40   Configuration:
41     Steps:
42       - Run: ./setup-sam.sh
43       - Run: sam package --template-file sam-template.yml --s3-bucket
44             codecatalyst-cfn-s3-bucket --output-template-file
45             sam-template-packaged.yml --region us-west-2
46   Outputs:
47     Artifacts:
48       - Name: buildArtifact
49     Files:
50       - "*"/*"
51   Compute:
52     Type: EC2
53 DeployCloudFormationStack:
54   Identifier: aws/cfn-deploy@v1
55   Configuration:
56     capabilities: CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND
57     role-arn: arn:aws:iam::111122223333:role/codecatalyst-stack-role
58     template: ./sam-template-packaged.yml
59     region: us-west-2
60     name: codecatalyst-cfn-stack
61   Environment:
62     Connections:
```

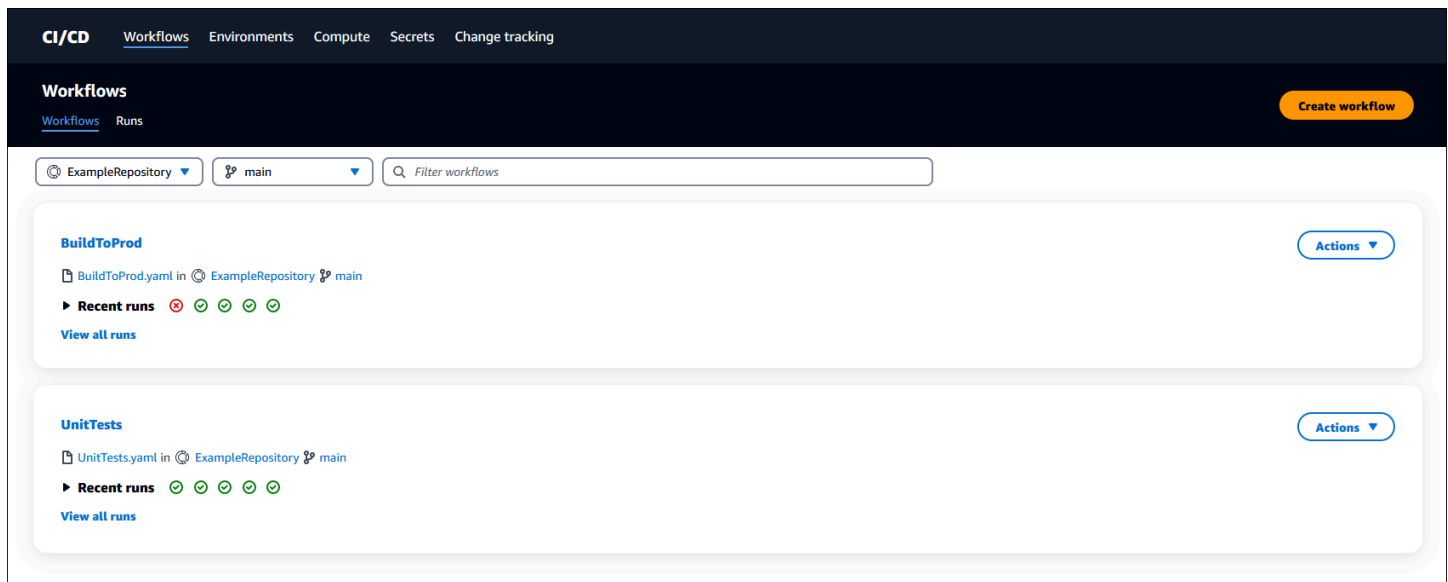
Vous pouvez passer de l'éditeur visuel à l'éditeur YAML pour voir l'effet de vos configurations sur le code YAML sous-jacent.

## Découvrir les flux de travail

Vous pouvez consulter votre flux de travail sur la page récapitulative des flux de travail, ainsi que les autres flux de travail que vous avez configurés dans le même projet.

L'image suivante montre la page récapitulative des flux de travail. Il est alimenté par deux flux de travail : BuildToProd et UnitTests. Vous pouvez voir que les deux ont été exécutés plusieurs fois. Vous pouvez choisir Recent runs pour consulter rapidement l'historique des exécutions, ou choisir le nom du flux de travail pour voir le code YAML du flux de travail et d'autres informations détaillées.





## Afficher les détails de l'exécution du workflow

Vous pouvez consulter les détails d'un flux de travail exécuté en choisissant l'exécution dans la page récapitulative des flux de travail.

L'image suivante montre les détails d'un flux de travail appelé Run-CC11d qui a été lancé automatiquement lors d'une validation vers la source. Le diagramme du flux de travail indique qu'une action a échoué (1). Vous pouvez accéder aux journaux (2) pour consulter les messages détaillés des journaux et résoudre les problèmes. Pour plus d'informations sur les exécutions de flux de travail, consultez [Travailler avec des courses](#).

The screenshot displays the Amazon CodeCatalyst console interface. At the top, the workflow 'Run-cc11d' is shown with a 'Failed' status. Below this, a summary table provides details: Status (Failed), Run mode (Queued), Trigger (Started by 0f8bf654), Start time (11 minutes ago), Duration (3 minutes 17 seconds), and End time (8 minutes ago). The main area shows a workflow graph with three actions: 'WorkflowSource', 'Test', and 'BuildBackend'. The 'BuildBackend' action is marked as 'Failed' with a red circle and the number '1'. A 'Detailed log messages' panel is open for the 'BuildBackend' action, showing a list of steps: 'Restore cache' (1 second), './setup-sam.sh' (14 seconds), and 'sam package --template-file sam-temple.yml --s3-bucket codecatalyst-cfn-s3-bucket --output-template-file sam-template-packaged.yml --region us-west-2' (3 seconds). The log for the failed step shows an error message: 'Error: Template file not found at /codecatalyst/output/src3862/src/git-codecommit.us'. A red circle with the number '2' is placed over the 'Detailed log messages' panel title.

## Étapes suivantes

Pour en savoir plus sur les concepts de flux de travail, voir [Concepts de workflows](#).

Pour créer votre premier flux de travail, voir [Commencer à utiliser les flux de travail dans CodeCatalyst](#).

## Concepts de workflows

Voici quelques concepts et termes à connaître lors de la création, du test ou du déploiement de votre code avec des flux de travail intégrés CodeCatalyst.

## Flux de travail

Un flux de travail est une procédure automatisée qui décrit comment créer, tester et déployer votre code dans le cadre d'un système d'intégration et de livraison continues (CI/CD). Un flux de travail définit une série d'étapes, ou d'actions, à effectuer lors de son exécution. Un flux de travail définit également les événements, ou déclencheurs, qui déclenchent le démarrage du flux de travail. Pour configurer un flux de travail, vous devez créer un fichier de définition de flux de travail à l'aide de [l'éditeur visuel ou YAML](#) de la CodeCatalyst console.

**i** Tip

Pour un aperçu rapide de la manière dont vous pouvez utiliser les flux de travail dans un projet, [créez un projet avec un plan](#). Chaque plan déploie un flux de travail fonctionnel que vous pouvez examiner, exécuter et tester.

Pour de plus amples informations sur les flux de travail, veuillez consulter [Utilisation des flux de travail](#).

## Fichiers de définition du flux de travail

Un fichier de définition de flux de travail est un fichier YAML qui décrit votre flux de travail. Le fichier est stocké dans un `~/ .codecatalyst/workflows/` dossier à la racine de votre [dépôt source](#). Le fichier peut avoir une extension `.yml` ou `.yaml`.

Pour plus d'informations sur le fichier de définition du flux de travail, consultez [Référence de définition du flux de travail](#).

## Actions

Une action est le principal élément constitutif d'un flux de travail et définit une unité logique de travail à effectuer lors de l'exécution d'un flux de travail. Généralement, un flux de travail inclut plusieurs actions qui s'exécutent de manière séquentielle ou en parallèle selon la façon dont vous les avez configurées.

Pour plus d'informations sur les actions, consultez [Utilisation des actions](#).

## Groupes d'action

Un groupe d'actions contient une ou plusieurs actions. Le regroupement des actions dans des groupes d'actions vous aide à organiser votre flux de travail et vous permet également de configurer les dépendances entre les différents groupes d'actions.

Pour plus d'informations sur les groupes d'actions, consultez [Regrouper les actions dans des groupes d'action](#).

## Artefacts

Un artefact est le résultat d'une action de flux de travail et consiste généralement en un dossier ou une archive de fichiers. Les artefacts sont importants car ils vous permettent de partager des fichiers et des informations entre les actions.

Pour plus d'informations sur les artefacts, consultez [Travailler avec des artefacts](#).

## Calcul

Le calcul fait référence au moteur informatique (le processeur, la mémoire et le système d'exploitation) géré et maintenu CodeCatalyst pour exécuter les actions du flux de travail.

Pour plus d'informations sur le calcul, consultez [Travailler avec le calcul](#).

## Environnements

Un environnement, à ne pas confondre avec un [environnement de développement](#), est l'endroit où le code est déployé. Il contient généralement une instance d'une application en cours d'exécution ainsi que son infrastructure associée. Vous pouvez donner à votre environnement un nom tel que development, test, staging ou production. Tous les déploiements générés par CodeCatalyst un environnement apparaîtront sur la page Environnements. Pour configurer un environnement, vous lui donnez un nom, par exemple my-production-environment, puis vous l'associez à votre Compte AWS.

Outre l'affichage des informations de déploiement, un environnement sert également de mécanisme permettant d'attribuer des rôles AWS IAM aux [actions](#) du flux de travail.

Pour plus d'informations sur les environnements, consultez [Utilisation des environnements](#).

## Rapports

Un rapport contient des informations détaillées sur les tests effectués lors de l'exécution d'un flux de travail. Vous pouvez créer des rapports tels qu'un rapport de test, un rapport de couverture du code, un rapport d'analyse de la composition logicielle et un rapport d'analyse statique. Vous pouvez utiliser un rapport pour résoudre un problème au cours d'un flux de travail. Si vous disposez de nombreux rapports issus de plusieurs flux de travail, vous pouvez les utiliser pour visualiser les tendances et les taux d'échec afin d'optimiser vos applications et vos configurations de déploiement.

Pour plus d'informations sur les rapports, consultez [Types de rapports de test](#).

## Exécutions

Une exécution est une itération unique d'un flux de travail. Au cours d'une exécution, CodeCatalyst exécute les actions définies dans le fichier de configuration du flux de travail et génère les journaux, artefacts et variables associés.

Pour plus d'informations sur les courses, consultez [Travailler avec des courses](#).

## Sources

Une source, également appelée source d'entrée, est un référentiel de sources auquel une [action de flux](#) de travail doit accéder pour exécuter ses tâches. Par exemple, une action de flux de travail peut nécessiter l'accès à une source pour obtenir des tests unitaires et les exécuter sur les fichiers source de votre application.

Pour plus d'informations sur les sources, consultez [Travailler avec des sources](#).

## Variables

Une variable est une paire clé-valeur qui contient des informations auxquelles vous pouvez faire référence dans votre CodeCatalyst flux de travail.

Pour plus d'informations sur les variables, consultez [Utilisation des variables](#).

## Déclencheurs de

Un déclencheur de flux de travail, ou simplement un déclencheur, vous permet de démarrer automatiquement un flux de travail lorsque certains événements se produisent, comme un envoi de code. Vous souhaitez peut-être configurer des déclencheurs pour éviter à vos développeurs de logiciels d'avoir à lancer des exécutions de flux de travail manuellement via la CodeCatalyst console.

Vous pouvez utiliser trois types de déclencheurs :

- **Push** : un déclencheur de code push déclenche l'exécution d'un flux de travail chaque fois qu'un commit est envoyé.
- **Demande d'extraction** : un déclencheur de demande d'extraction entraîne le démarrage d'un flux de travail chaque fois qu'une demande d'extraction est créée, révisée ou fermée.
- **Planification** : un déclencheur de planification entraîne le démarrage d'un flux de travail selon un calendrier que vous définissez. Envisagez d'utiliser un déclencheur de planification pour exécuter

des versions nocturnes de votre logiciel afin que les développeurs puissent travailler sur la dernière version le lendemain matin.

Vous pouvez utiliser les déclencheurs push, pull request et schedule seuls ou en combinaison dans le même flux de travail.

Pour plus d'informations sur les déclencheurs, consultez [Utilisation de déclencheurs](#).

## Commencer à utiliser les flux de travail dans CodeCatalyst

Dans ce didacticiel, vous allez apprendre à créer et à configurer votre premier flux de travail.

### Tip

Vous préférez commencer par un flux de travail préconfiguré ? Voir [Création d'un projet à l'aide d'un plan](#), qui inclut des instructions pour configurer un projet avec un flux de travail fonctionnel, un exemple d'application et d'autres ressources.

### Rubriques

- [Prérequis](#)
- [Étape 1 : créer et configurer votre flux de travail](#)
- [Étape 2 : Enregistrez votre flux de travail avec un commit](#)
- [Étape 3 : Afficher les résultats de la course](#)
- [\(Facultatif\) Étape 4 : Nettoyer](#)

## Prérequis

Avant de commencer :

- Tu as besoin d'un CodeCatalyst espace. Pour plus d'informations, consultez [Création d'un espace prenant en charge les utilisateurs de AWS Builder ID](#).
- Dans votre CodeCatalyst espace, vous avez besoin d'un CodeCatalyst projet vide, créé à partir de zéro, appelé :

```
codecatalyst-project
```

Pour plus d'informations, consultez [Création d'un projet vide dans Amazon CodeCatalyst](#).

- Dans votre projet, vous avez besoin d'un CodeCatalyst dépôt appelé :

```
codecatalyst-source-repository
```

Pour plus d'informations, consultez [Création d'un référentiel source](#).

#### Note

Si vous avez un projet et un référentiel de sources existants, vous pouvez les utiliser ; toutefois, la création de nouveaux référentiels facilitera le nettoyage à la fin de ce didacticiel.

## Étape 1 : créer et configurer votre flux de travail

Au cours de cette étape, vous créez et configurez un flux de travail qui génère et teste automatiquement votre code source lorsque des modifications sont apportées.

Pour créer votre flux de travail

1. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
2. Choisissez Créer un flux de travail.

Le fichier de définition du flux de travail apparaît dans l'éditeur YAML de la CodeCatalyst console.

Pour configurer votre flux de travail

Vous pouvez configurer votre flux de travail dans l'éditeur visuel ou dans l'éditeur YAML. Commençons par l'éditeur YAML, puis passons à l'éditeur visuel.

1. Choisissez + Actions pour afficher la liste des actions de flux de travail que vous pouvez ajouter à votre flux de travail.
2. Dans l'action Créer, choisissez + pour ajouter le code YAML de l'action à votre fichier de définition du flux de travail. Votre flux de travail ressemble désormais à ce qui suit.

```
Name: Workflow_fe47
```

```
SchemaVersion: "1.0"

# Optional - Set automatic triggers.
Triggers:
  - Type: Push
    Branches:
      - main

# Required - Define action configurations.
Actions:
  Build_f0:
    Identifier: aws/build@v1

    Inputs:
      Sources:
        - WorkflowSource # This specifies that the action requires this workflow as
a source

    Outputs:
      AutoDiscoverReports:
        Enabled: true
        # Use as prefix for the report files
        ReportNamePrefix: rpt

    Configuration:
      Steps:
        - Run: echo "Hello, World!"
        - Run: echo "<?xml version=\"1.0\" encoding=\"UTF-8\" ?>" >> report.xml
        - Run: echo "<testsuite tests=\"1\" name=\"TestAgentJunit\" >" >>
report.xml
        - Run: echo "<testcase classname=\"TestAgentJunit\" name=\"Dummy
Test\"/></testsuite>" >> report.xml
```

Le flux de travail copie les fichiers du référentiel WorkflowSource source sur la machine de calcul qui exécute l'Build\_f0action, les imprime Hello, World! dans les journaux, découvre les rapports de test sur la machine de calcul et les affiche sur la page Rapports de la CodeCatalyst console.

3. Choisissez Visual pour afficher le fichier de définition du flux de travail dans l'éditeur visuel. Les champs de l'éditeur visuel vous permettent de configurer les propriétés YAML affichées dans l'éditeur YAML.



## Étape 2 : Enregistrez votre flux de travail avec un commit

Au cours de cette étape, vous devez enregistrer vos modifications. Les flux de travail étant stockés sous forme de `.yaml` fichiers dans votre référentiel, vous enregistrez vos modifications par le biais de validations.

Pour valider les modifications apportées à votre flux de travail

1. (Facultatif) Choisissez Valider pour vous assurer que le code YAML du flux de travail est valide.
2. Choisissez Commit (Valider).
3. Dans Nom du fichier de flux de travail, entrez un nom pour le fichier de configuration de votre flux de travail, par exemple **my-first-workflow**.
4. Dans Message de validation, entrez un message pour identifier votre validation, par exemple **create my-first-workflow.yaml**.
5. Dans Référentiel, choisissez le référentiel dans lequel vous souhaitez enregistrer le flux de travail (codecatalyst-repository).
6. Dans Nom de la branche, choisissez la branche dans laquelle vous souhaitez enregistrer le flux de travail (main).
7. Choisissez Commit (Valider).

Votre nouveau flux de travail apparaît dans la liste des flux de travail. L'affichage peut prendre plusieurs minutes.

Étant donné que les flux de travail sont enregistrés avec des validations et qu'un déclencheur push de code est configuré pour le flux de travail, l'enregistrement du flux de travail démarre automatiquement l'exécution d'un flux de travail.

## Étape 3 : Afficher les résultats de la course

Au cours de cette étape, vous accédez à l'exécution qui a été lancée à partir de votre validation et vous visualisez les résultats.

Pour afficher les résultats d'une course

1. Choisissez le nom de votre flux de travail, par exemple, `Workflow_fe47`.

Un diagramme de flux de travail montrant l'étiquette de votre référentiel source (WorkflowSource) et l'action de génération (par exemple, Build\_f0).

2. Dans le diagramme d'exécution du flux de travail, choisissez l'action de génération (par exemple, Build\_f0).
3. Passez en revue le contenu des onglets Journaux, Rapports, Configuration et Variables. Ces onglets vous montrent les résultats de votre action de création.

Pour plus d'informations, consultez [Afficher les résultats d'une action de construction](#).

## (Facultatif) Étape 4 : Nettoyer

Au cours de cette étape, vous allez nettoyer les ressources que vous avez créées dans ce didacticiel.

Pour supprimer des ressources

1. Si vous avez créé un nouveau projet pour ce didacticiel, supprimez-le. Pour obtenir des instructions, veuillez consulter [Supprimer un projet sur Amazon CodeCatalyst](#). La suppression du projet entraîne également la suppression du référentiel source et du flux de travail.
2. Supprimez le flux de travail s'il n'est pas encore supprimé. Pour obtenir des instructions, veuillez consulter [Pour supprimer un flux de travail](#).
3. Supprimez le dépôt s'il n'est pas encore supprimé. Pour obtenir des instructions, consultez [Supprimer un référentiel source](#).

## Affichage de la qualité du code et de l'état du déploiement par validation CodeCatalyst


À tout moment du cycle de développement, il est important de connaître l'état de déploiement de validations spécifiques, telles que les corrections de bogues, les nouvelles fonctionnalités ou les autres modifications importantes. Envisagez les scénarios suivants dans lesquels la fonctionnalité de suivi de l'état du déploiement est utile aux équipes de développement :

- En tant que développeur, vous avez apporté un correctif pour corriger un bogue et vous souhaitez signaler l'état de son déploiement dans les environnements de déploiement de votre équipe.
- En tant que responsable de publication, vous souhaitez consulter la liste des validations déployées afin de suivre et de signaler l'état de leur déploiement.

CodeCatalyst fournit une vue que vous pouvez utiliser pour déterminer en un coup d'œil où les validations ou modifications individuelles ont été déployées, et dans quel environnement. Cette vue inclut :

- Une liste de validations.
- État des déploiements qui incluent les validations.
- Les environnements dans lesquels les validations sont déployées avec succès.
- État de tous les tests exécutés par rapport aux validations de votre flux de travail CI/CD.

La procédure suivante explique comment accéder à cette vue et comment l'utiliser pour suivre les modifications apportées à votre projet.

 Note

Le suivi de l'état du déploiement par validation n'est pris en charge qu'avec [CodeCatalyst les référentiels](#). Vous ne pouvez pas utiliser cette fonctionnalité avec un [GitHub référentiel](#).

Pour suivre l'état du déploiement par validation

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis choisissez Suivi des modifications.
4. Dans les deux listes déroulantes situées en haut du volet principal, choisissez le référentiel source et la branche contenant les validations dont vous souhaitez consulter l'état de publication.
5. Choisissez Afficher les modifications.

La liste des validations s'affiche.

Pour chaque validation, vous pouvez consulter les informations suivantes :

- Validez des informations telles que l'identifiant, l'auteur, le message et la date de validation. Pour plus d'informations, consultez [Référentiels de sources dans CodeCatalyst](#).
- État des déploiements dans chaque environnement. Pour plus d'informations, consultez [Utilisation des environnements](#).
- Résultats des tests et de la couverture du code. Pour plus d'informations, consultez [Tests à l'aide de workflows dans CodeCatalyst](#).

**Note**

Les résultats de l'analyse de composition logicielle (SCA) ne sont pas affichés.

- (Facultatif) Pour afficher plus d'informations sur les modifications liées à un commit spécifique, y compris le dernier déploiement, la couverture détaillée du code et les informations relatives aux tests unitaires, choisissez Afficher les détails de ce commit.

## Création à l'aide de flux de travail dans CodeCatalyst

À l'aide [CodeCatalyst de flux de travail](#), vous pouvez créer des applications et d'autres ressources.

### Rubriques

- [Comment créer une application ?](#)
- [Avantages de l'action Build](#)
- [Alternatives à l'action de construction](#)
- [Ajouter l'action de construction](#)
- [Afficher les résultats d'une action de construction](#)
- [Tutoriel : télécharger des artefacts sur Amazon S3](#)

## Comment créer une application ?

Pour créer une application ou une ressource CodeCatalyst, vous devez d'abord créer un flux de travail, puis spécifier une action de génération à l'intérieur de celui-ci.

Une action de génération est un élément constitutif du flux de travail qui compile votre code source, exécute des tests unitaires et produit des artefacts prêts à être déployés.

Vous pouvez ajouter une action de génération à votre flux de travail à l'aide de l'éditeur visuel ou de l'éditeur YAML de la CodeCatalyst console.

Les étapes de haut niveau pour créer une application ou une ressource sont les suivantes.

Pour créer une application (tâches de haut niveau)

1. Dans CodeCatalyst, vous ajoutez le code source d'une application que vous souhaitez créer. Pour plus d'informations, consultez [Utilisation des référentiels de sources dans CodeCatalyst](#).

2. Dans CodeCatalyst, vous créez un flux de travail. Le flux de travail vous permet de définir comment créer, tester et déployer votre application. Pour plus d'informations, consultez [Commencer à utiliser les flux de travail dans CodeCatalyst](#).
3. (Facultatif) Dans le flux de travail, vous ajoutez un déclencheur qui indique les événements qui provoqueront le démarrage automatique du flux de travail. Pour plus d'informations, consultez [Utilisation de déclencheurs](#).
4. Dans le flux de travail, vous ajoutez une action de génération qui compile et empaquète le code source de votre application ou de votre ressource. Facultativement, vous pouvez également demander à l'action de génération d'exécuter des tests unitaires, de générer des rapports et de déployer votre application si vous ne souhaitez pas utiliser d'action de test ou de déploiement à ces fins. Pour en savoir plus sur les actions de test et de déploiement, consultez [Ajouter l'action de construction](#).
5. (Facultatif) Dans le flux de travail, vous ajoutez une action de test et une action de déploiement pour tester et déployer votre application ou votre ressource. Vous pouvez choisir parmi plusieurs actions préconfigurées pour déployer votre application sur différentes cibles, telles qu'Amazon ECS. Pour plus d'informations, consultez [Tests à l'aide de workflows dans CodeCatalyst](#) et [Déploiement à l'aide de workflows dans CodeCatalyst](#).
6. Vous démarrez le flux de travail manuellement ou automatiquement par le biais d'un déclencheur. Le flux de travail exécute les actions de génération, de test et de déploiement en séquence pour créer, tester et déployer votre application et vos ressources sur la cible. Pour plus d'informations, consultez [Démarrage d'une exécution de flux de travail](#).

## Avantages de l'action Build

L'utilisation de l'action de création dans un flux de travail présente les avantages suivants :

- Entièrement géré : l'action de génération élimine le besoin de configurer, de corriger, de mettre à jour et de gérer vos propres serveurs de génération.
- À la demande — L'action de création évolue à la demande pour répondre à vos besoins de création. Vous ne payez que pour le nombre de minutes de génération que vous consommez. Pour plus d'informations, consultez [Travailler avec le calcul](#).
- Prêt à l'emploi : CodeCatalyst inclut des images Docker d'environnement d'exécution préemballées qui sont utilisées pour exécuter toutes les actions de votre flux de travail, y compris les actions de génération. Ces images sont préconfigurées avec des outils utiles pour créer des applications telles que le AWS CLI et Node.js. Vous pouvez configurer CodeCatalyst pour utiliser une image de

build que vous fournissez à partir d'un registre public ou privé. Pour plus d'informations, consultez [Utilisation des images Docker de l'environnement d'exécution](#).

## Alternatives à l'action de construction

Si vous utilisez une action de génération pour déployer votre application, pensez plutôt à utiliser une action de CodeCatalyst déploiement. Les actions de déploiement exécutent une behind-the-scenes configuration que vous devriez autrement écrire manuellement si vous utilisez une action de génération. Pour plus d'informations sur les actions de déploiement disponibles, consultez [Liste des actions de déploiement](#).

Vous pouvez également l'utiliser AWS CodeBuild pour créer vos applications. Pour plus d'informations, voir [Qu'est-ce que c'est CodeBuild ?](#).

### Rubriques

- [Ajouter l'action de construction](#)
- [Afficher les résultats d'une action de construction](#)
- [Tutoriel : télécharger des artefacts sur Amazon S3](#)

## Ajouter l'action de construction

Utilisez la procédure suivante pour ajouter une action de génération à votre CodeCatalyst flux de travail.

### Visual

Pour ajouter une action de génération à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
3. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
4. Choisissez Modifier.
5. Choisissez Visual.
6. Choisissez Actions.
7. Dans Actions, sélectionnez Construire.

8. Dans les onglets Entrées et Configuration, complétez les champs selon vos besoins. Pour une description de chaque champ, consultez le [Créez et testez une référence d'actions](#). Cette référence fournit des informations détaillées sur chaque champ (et la valeur de propriété YAML correspondante) tel qu'il apparaît dans les éditeurs YAML et visuels.
9. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
10. Choisissez Commit, entrez un message de validation, puis choisissez Commit à nouveau.

## YAML

Pour ajouter une action de génération à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
3. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
4. Choisissez Modifier.
5. Choisissez YAML.
6. Choisissez Actions.
7. Dans Actions, sélectionnez Construire.
8. Modifiez les propriétés du code YAML en fonction de vos besoins. Une explication de chaque propriété disponible est fournie dans le [Créez et testez une référence d'actions](#).
9. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
10. Choisissez Commit, entrez un message de validation, puis choisissez Commit à nouveau.

## Définir une action

L'action de génération est définie comme un ensemble de propriétés YAML dans votre fichier de définition de flux de travail. Pour plus d'informations sur ces propriétés, consultez [Créez et testez une référence d'actions](#) le [Référence de définition du flux de travail](#).

## Afficher les résultats d'une action de construction

Utilisez les instructions suivantes pour afficher les résultats d'une action de génération, y compris les journaux, les rapports et les variables générés.

## Pour afficher les résultats d'une action de création

1. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
2. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
3. Dans le diagramme du flux de travail, choisissez le nom de votre action de génération, par exemple, Construire.
4. Pour afficher les journaux de l'exécution de la compilation, choisissez Logs. Les journaux des différentes phases de construction sont affichés. Vous pouvez développer les journaux selon vos besoins.
5. Pour afficher les rapports de test produits par l'action de génération, choisissez Rapports, ou dans le volet de navigation, sélectionnez Rapports. Pour plus d'informations, consultez [Types de rapports de test](#).
6. Pour afficher la configuration utilisée pour l'action de génération, choisissez Configuration. Pour plus d'informations, consultez [Ajouter l'action de construction](#).
7. Pour afficher les variables utilisées par l'action de génération, choisissez Variables. Pour plus d'informations, voir [Utilisation des variables](#).

## Tutoriel : télécharger des artefacts sur Amazon S3

Dans ce didacticiel, vous apprendrez à télécharger des artefacts dans un compartiment Amazon S3 à l'aide d'un CodeCatalyst [flux de travail](#) qui inclut quelques [actions de génération](#). Ces actions sont exécutées en série lorsque le flux de travail démarre. La première action de construction génère deux fichiers `Hello.txt` et `Goodbye.txt` les regroupe dans un artefact de construction. La deuxième action de génération télécharge l'artefact sur Amazon S3. Vous allez configurer le flux de travail pour qu'il s'exécute chaque fois que vous envoyez un commit à votre dépôt source.

### Rubriques

- [Prérequis](#)
- [Étape 1 : créer un AWS rôle](#)
- [Étape 2 : Création d'un compartiment Amazon S3](#)
- [Étape 3 : Création d'un référentiel source](#)
- [Étape 4 : Création d'un flux de travail](#)
- [Étape 5 : Vérifiez les résultats](#)



- [Nettoyage](#)

## Prérequis

Avant de commencer, vous avez besoin de ce qui suit :

- Vous avez besoin d'un CodeCatalyst espace avec un AWS compte connecté. Pour plus d'informations, consultez [Création d'un espace prenant en charge les utilisateurs de AWS Builder ID](#).

- Dans votre espace, vous avez besoin d'un CodeCatalyst projet vide, créé à partir de zéro, appelé :

```
codecatalyst-artifact-project
```

Pour plus d'informations, consultez [Création d'un projet vide dans Amazon CodeCatalyst](#).

- Dans votre projet, vous avez besoin d'un CodeCatalyst environnement appelé :

```
codecatalyst-artifact-environment
```

Configurez cet environnement comme suit :

- Choisissez n'importe quel type, tel que Développement.
- AWS Connectez-y votre compte.

Pour plus d'informations, consultez [Utilisation des environnements](#).

## Étape 1 : créer un AWS rôle

Au cours de cette étape, vous créez un rôle AWS IAM que vous attribuerez ultérieurement à l'action de création dans votre flux de travail. Ce rôle accorde à l'action de CodeCatalyst création l'autorisation d'accéder à votre AWS compte et d'écrire sur Amazon S3 où votre artefact sera stocké. Ce rôle s'appelle le rôle Build.

### Note

Si vous possédez déjà un rôle de build que vous avez créé pour un autre didacticiel, vous pouvez également l'utiliser pour ce didacticiel. Assurez-vous simplement qu'il dispose des autorisations et de la politique de confiance indiquées dans la procédure suivante.

Pour plus d'informations sur les rôles IAM, consultez la section Rôles [IAM dans le Guide](#) de l'AWS AWS Identity and Access Management utilisateur.

Pour créer un rôle de build

1. Créez une politique pour le rôle, comme suit :
  - a. Connectez-vous à AWS.
  - b. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
  - c. Dans le panneau de navigation, choisissez Politiques (Politiques).
  - d. Sélectionnez Create policy (Créer une politique).
  - e. Choisissez l'onglet JSON.
  - f. Supprimez le code existant.
  - g. Collez le code suivant :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource": "*"
    }
  ]
}
```

#### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

- h. Choisissez Suivant : Balises.
- i. Choisissez Suivant : Vérification.
- j. Dans Nom, entrez :

```
codecatalyst-s3-build-policy
```

- k. Choisissez Créer une politique.

Vous venez de créer une politique d'autorisation.

2. Créez le rôle de build, comme suit :

- a. Dans le volet de navigation, sélectionnez Rôles, puis Créer un rôle.
- b. Choisissez Politique de confiance personnalisée.
- c. Supprimez la politique de confiance personnalisée existante.
- d. Ajoutez la politique de confiance personnalisée suivante :

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

- e. Choisissez Suivant.
- f. Dans Politiques d'autorisations, recherchez `codecatalyst-s3-build-policy` et cochez la case correspondante.
- g. Choisissez Suivant.
- h. Dans Nom du rôle, entrez :

```
codecatalyst-s3-build-role
```

- i. Pour la description du rôle, entrez :

```
CodeCatalyst build role
```

- j. Sélectionnez Créer un rôle.

Vous avez maintenant créé un rôle de build avec une politique de confiance et une politique d'autorisations.

## Étape 2 : Création d'un compartiment Amazon S3

Au cours de cette étape, vous créez un compartiment Amazon S3 dans lequel les `Goodbye.txt` artefacts `Hello.txt` et seront chargés.

Pour créer un compartiment Amazon S3

1. Ouvrez la console Amazon S3 sur <https://console.aws.amazon.com/s3/>.
2. Dans le volet principal, choisissez Create bucket.
3. Dans le champ Nom du compartiment, entrez :

```
codecatalyst-artifact-bucket
```

4. Choisissez une région dans Région AWS . Ce didacticiel part du principe que vous avez choisi US West (Oregon) us-west-2. Pour plus d'informations sur les régions prises en charge par Amazon S3, consultez la section [Points de terminaison et quotas Amazon Simple Storage Service](#) dans le Références générales AWS.
5. Au bas de la page, choisissez Create bucket.
6. Copiez le nom du bucket que vous venez de créer, par exemple :

```
codecatalyst-artifact-bucket
```

Vous venez de créer un bucket appelé **codecatalyst-artifact-bucket** dans la région us-west-2 de l'ouest des États-Unis (Oregon).

## Étape 3 : Création d'un référentiel source

Au cours de cette étape, vous créez un référentiel source dans CodeCatalyst. Ce référentiel est utilisé pour stocker le fichier de définition du flux de travail du didacticiel.

Pour plus d'informations sur les référentiels sources, consultez [Création d'un référentiel source](#).

Pour créer un référentiel source

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre projet, `codecatalyst-artifact-project`.
3. Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources.
4. Choisissez Ajouter un référentiel, puis sélectionnez Créer un référentiel.
5. Dans Nom du référentiel, entrez :

`codecatalyst-artifact-source-repository`

6. Choisissez Créer.

Vous venez de créer un dépôt appelé `codecatalyst-artifact-source-repository`.

## Étape 4 : Création d'un flux de travail

Au cours de cette étape, vous allez créer un flux de travail composé des éléments de base suivants qui s'exécutent de manière séquentielle :

- Un déclencheur : ce déclencheur lance automatiquement l'exécution du flux de travail lorsque vous apportez une modification à votre référentiel source. Pour plus d'informations sur les déclencheurs, consultez [Utilisation de déclencheurs](#).
- Une action de génération appelée `GenerateFiles` — Au moment du déclenchement, l'`GenerateFiles` action crée deux fichiers `Goodbye.txt`, `Hello.txt` puis les regroupe dans un artefact de sortie appelé `codecatalystArtifact`.
- Une autre action de Upload génération appelée — Une fois l'`GenerateFiles` action terminée, l'`Upload` action exécute la AWS CLI commande `aws s3 sync` pour télécharger les fichiers dans `codecatalystArtifact` et dans votre référentiel source vers votre compartiment Amazon S3. Il AWS CLI est préinstallé et préconfiguré sur la plate-forme CodeCatalyst informatique, vous n'avez donc pas besoin de l'installer ou de le configurer.

Pour plus d'informations sur le logiciel préinstallé sur la plate-forme de CodeCatalyst calcul, consultez [Utilisation des images Docker de l'environnement d'exécution](#). Pour plus d'informations sur la `aws s3 sync` commande AWS CLI's, voir [sync](#) dans le manuel de référence des AWS CLI commandes.

Pour plus d'informations sur l'action de génération, consultez [Création à l'aide de flux de travail dans CodeCatalyst](#).

Pour créer un flux de travail

1. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
2. Choisissez Créer un flux de travail.
3. Supprimez l'exemple de code YAML.
4. Ajoutez le code YAML suivant :

```
Name: codecatalyst-artifact-workflow
SchemaVersion: 1.0

Triggers:
  - Type: Push
    Branches:
      - main
Actions:
  GenerateFiles:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        # Create the output files.
        - Run: echo "Hello, World!" > "Hello.txt"
        - Run: echo "Goodbye!" > "Goodbye.txt"
    Outputs:
      Artifacts:
        - Name: codecatalystArtifact
          Files:
            - "**/*"
  Upload:
    Identifier: aws/build@v1
    DependsOn:
      - GenerateFiles
    Environment:
```

```
Name: codecatalyst-artifact-environment
Connections:
  - Name: codecatalyst-account-connection
    Role: codecatalyst-s3-build-role
Inputs:
  Artifacts:
    - codecatalystArtifact
Configuration:
  Steps:
    # Upload the output artifact to the S3 bucket.
    - Run: aws s3 sync . s3://codecatalyst-artifact-bucket
```

Dans le code ci-dessus, remplacez :

- *codecatalyst-artifact-environment* avec le nom de l'environnement dans lequel vous l'avez créé [Prérequis](#).
- *codecatalyst-account-connection* avec le nom de la connexion au compte que vous avez créée [Prérequis](#).
- *codecatalyst-s3-build-role* avec le nom du rôle de construction que vous avez créé dans. [Étape 1 : créer un AWS rôle](#)
- *codecatalyst-artifact-bucket* avec le nom de l'Amazon S3 dans lequel vous l'avez créé [Étape 2 : Création d'un compartiment Amazon S3](#).

Pour plus d'informations sur les propriétés de ce fichier, consultez le [Créez et testez une référence d'actions](#).

5. (Facultatif) Choisissez Valider pour vous assurer que le code YAML est valide avant de valider.
6. Choisissez Commit (Valider).
7. Dans la boîte de dialogue du flux de travail de validation, entrez ce qui suit :
  - a. Pour le nom du fichier de flux de travail, laissez la valeur par défaut, `codecatalyst-artifact-workflow`.
  - b. Pour le message de validation, entrez :

```
add initial workflow file
```

- c. Pour Repository, choisissez `codecatalyst-artifact-source-repository`.
- d. Pour Nom de la branche, choisissez `main`.

e. Choisissez Commit (Valider).

Vous venez de créer un flux de travail. L'exécution d'un flux de travail démarre automatiquement en raison du déclencheur défini en haut du flux de travail. Plus précisément, lorsque vous avez validé (et transféré) le `codecatalyst-artifact-workflow.yaml` fichier dans votre référentiel source, le déclencheur a lancé l'exécution du flux de travail.

Pour consulter le flux de travail en cours d'exécution

1. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
2. Choisissez le flux de travail que vous venez de créer : `codecatalyst-artifact-workflow`.
3. Choisissez GenerateFilesde voir la progression de la première action de construction.
4. Choisissez Upload pour voir la progression de la deuxième action de build.
5. Lorsque l'action Charger est terminée, procédez comme suit :
  - Si l'exécution du flux de travail a réussi, passez à la procédure suivante.
  - Si l'exécution du flux de travail a échoué, choisissez Logs pour résoudre le problème.

## Étape 5 : Vérifiez les résultats

Une fois le flux de travail exécuté, accédez au service Amazon S3 et examinez votre `codecatalyst-artifact-bucket` compartiment. Il devrait désormais inclure les fichiers et dossiers suivants :

```
.
|- .aws/
|- .git/
|Goodbye.txt
|Hello.txt
|README.md
```

Les Hello.txt fichiers Goodbye.txt et ont été téléchargés car ils faisaient partie de l'`codecatalystArtifact`artefact. Les README.md fichiers .aws/, .git/, et ont été chargés parce qu'ils se trouvaient dans votre référentiel source.



## Nettoyage

Faites le ménage CodeCatalyst et évitez AWS d'être facturé pour ces services.

Pour nettoyer dans CodeCatalyst

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Supprimez le référentiel `codecatalyst-artifact-source-repository` source.
3. Supprimez le `codecatalyst-artifact-workflow` flux de travail.

Pour nettoyer dans AWS

1. Nettoyez dans Amazon S3, comme suit :
  - a. Ouvrez la console Amazon S3 sur <https://console.aws.amazon.com/s3/>.
  - b. Supprimez les fichiers du `codecatalyst-artifact-bucket` compartiment.
  - c. Supprimez le `codecatalyst-artifact-bucket` compartiment.
2. Nettoyez dans IAM, comme suit :
  - a. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
  - b. Supprimez la `codecatalyst-s3-build-policy`.
  - c. Supprimez la `codecatalyst-s3-build-role`.

## Tests à l'aide de workflows dans CodeCatalyst

Dans CodeCatalyst, vous pouvez exécuter des tests dans le cadre de différentes actions de flux de travail, telles que la création et le test. Ces actions de flux de travail peuvent toutes générer des rapports. Une action de test est une action de flux de travail qui produit des rapports de test, de couverture de code, d'analyse de la composition logicielle et d'analyse statique. Ces rapports sont affichés dans la CodeCatalyst console.

Rubriques

- [Types de rapports de test](#)
- [Ajouter l'action de test](#)
- [Configuration des rapports](#)
- [En utilisant universal-test-runner](#)

- [Bonnes pratiques](#)
- [Travailler avec des tests](#)

## Types de rapports de test

L'action de CodeCatalyst test Amazon prend en charge les types de rapports suivants. Pour un exemple de mise en forme de ces rapports dans votre fichier YAML, consultez [Exemple : Configuration de rapports](#).

### Rubriques

- [Rapports d'essais](#)
- [Rapports sur la couverture du code](#)
- [Rapports d'analyse de la composition des logiciels](#)
- [Rapports d'analyse statiques](#)

## Rapports d'essais

Dans CodeCatalyst, vous pouvez configurer des tests unitaires, des tests d'intégration et des tests système exécutés pendant les builds. CodeCatalyst Vous pouvez ensuite créer des rapports contenant les résultats de vos tests.

Vous pouvez utiliser un rapport de test pour résoudre les problèmes liés à vos tests. Si vous disposez de nombreux rapports de test provenant de plusieurs versions, vous pouvez utiliser vos rapports de test pour afficher les taux d'échec afin de vous aider à optimiser vos versions.

Vous pouvez utiliser les formats de fichier de rapport de test suivants :

- Concombre JSON (.json)
- JUnit XML (.xml)
- NUnit XML (.xml)
- Unit3 XML (.xml)
- TestNG XML (.xml)
- Visual Studio TRX (.trx, .xml)

## Rapports sur la couverture du code

Dans CodeCatalyst, vous pouvez générer des rapports de couverture de code pour vos tests. CodeCatalyst fournit les mesures de couverture du code suivantes :

### Couverture de la ligne

Mesure le nombre de déclarations couvertes par vos tests. Une instruction est une instruction unique, sans les commentaires.

$$\text{line coverage} = (\text{total lines covered}) / (\text{total number of lines})$$

### Couverture des succursales

Mesure le nombre de branches couvertes par vos tests parmi toutes les branches possibles d'une structure de contrôle, telle qu'une case instruction `if` or.

$$\text{branch coverage} = (\text{total branches covered}) / (\text{total number of branches})$$

Les formats de fichier de rapport de couverture de code suivants sont pris en charge :

- JaCoCo XML (.xml)
- SimpleCov [JSON \(généré par simplecov, pas par simplecov-json, .json\)](#)
- Clover XML (version 3, .xml)
- Couverture XML (.xml)
- LCOV (.info)

## Rapports d'analyse de la composition des logiciels

Dans CodeCatalyst, vous pouvez utiliser les outils d'analyse de la composition logicielle (SCA) pour analyser les composants de votre application et vérifier les vulnérabilités de sécurité connues. Vous pouvez découvrir et analyser les rapports SARIF qui détaillent les vulnérabilités de gravité variable et les moyens de les corriger. Les valeurs de gravité valides, de la plus sévère à la moins sévère, sont les suivantes : CRITICALHIGH,MEDIUM,LOW,,INFORMATIONAL.

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

- SARIF (.sarif, .json)

## Rapports d'analyse statiques

Vous pouvez utiliser des rapports d'analyse statique (SA) pour identifier les défauts du code au niveau de la source. Dans CodeCatalyst, vous pouvez générer des rapports SA pour aider à résoudre les problèmes liés à votre code avant de le déployer. Ces problèmes incluent des bogues, des failles de sécurité, des problèmes de qualité et d'autres vulnérabilités. Les valeurs de gravité valides, de la plus sévère à la moins sévère, sont les suivantes : CRITICAL HIGHMEDIUM,LOW,, etINFORMATIONAL.

CodeCatalyst fournit les métriques SA suivantes :

### Insectes

Identifie un certain nombre de bogues potentiels trouvés dans votre code source. Ces bogues peuvent inclure des problèmes liés à la sécurité de la mémoire. Voici un exemple de bogue.

```
// The while loop will inadvertently index into array x out-of-bounds
int x[64];
while (int n = 0; n <= 64; n++) {
    x[n] = 0;
}
```

### Failles de sécurité

Identifie un certain nombre de failles de sécurité possibles détectées dans votre code source. Ces failles de sécurité peuvent inclure des problèmes tels que le stockage de vos jetons secrets en texte clair.

### Problèmes de qualité

Identifie un certain nombre de problèmes de qualité potentiels trouvés dans votre code source. Ces problèmes de qualité peuvent inclure des problèmes liés aux conventions de style. Voici un exemple de problème de qualité.

```
// The function name doesn't adhere to the style convention of camelCase
int SUBTRACT(int x, int y) {
    return x-y
}
```

### Autres vulnérabilités

Identifie un certain nombre d'autres vulnérabilités possibles détectées dans votre code source.

CodeCatalyst prend en charge les formats de fichier de rapport SA suivants :

- PyLint (.py)
- ESLint (.js, .jsx, .ts, .tsx)
- SARIF (.sarif, .json)

## Ajouter l'action de test

Utilisez la procédure suivante pour ajouter une action de test à votre flux de travail.

Pour ajouter une action de test

- Suivez les instructions de la section [Ajouter l'action de construction](#). Les instructions pour ajouter une action de génération et une action de test sont exactement les mêmes.

## Définition de l'action de test

L'action de test est définie comme un ensemble de propriétés YAML dans votre fichier de définition de flux de travail. Pour plus d'informations sur ces propriétés, consultez [Créez et testez une référence d'actions](#) le [Référence de définition du flux de travail](#).

## Configuration des rapports

Cette section décrit comment configurer un rapport de qualité.

Rubriques

- [Configuration de la découverte automatique et des rapports manuels](#)
- [Configuration des critères de réussite pour les rapports](#)
- [Exemple : Configuration de rapports](#)
- [Support SARIF pour les rapports d'analyse de composition logicielle et d'analyse statique](#)

## Configuration de la découverte automatique et des rapports manuels

Lorsque la découverte automatique est activée, CodeCatalyst recherche toutes les entrées transmises à l'action et tous les fichiers générés par l'action elle-même, à la recherche de rapports de test, de couverture de code, d'analyse de la composition logicielle (SCA) et d'analyse statique (SA). Vous pouvez consulter et manipuler chacun de ces rapports dans CodeCatalyst.

Vous pouvez également configurer manuellement les rapports qui sont générés. Vous pouvez spécifier le type de rapport que vous souhaitez générer ainsi que le format de fichier. Pour plus d'informations, consultez [Types de rapports de test](#).

## Configuration des critères de réussite pour les rapports

Vous pouvez définir les valeurs qui déterminent les critères de réussite d'un test, d'un rapport de couverture du code, d'analyse de la composition logicielle (SCA) ou d'un rapport d'analyse statique (SA).

Les critères de réussite sont des seuils qui déterminent si un rapport est réussi ou non. CodeCatalyst génère d'abord votre rapport, qui peut être un rapport de test, de couverture de code, SCA ou SA, puis applique les critères de réussite aux rapports générés. Il indique ensuite si les critères de réussite ont été remplis, et dans quelle mesure. Si un rapport ne répond pas aux critères de succès spécifiés, l' action CodeCatalyst qui a spécifié les critères de réussite échoue.

Par exemple, lorsque vous définissez les critères de réussite de votre rapport SCA, les valeurs de vulnérabilité valides, allant de la plus grave à la moins grave, sont les suivantes : CRITICALHIGH,, MEDIUMLOW,INFORMATIONAL. Si vous définissez les critères d'analyse d'une vulnérabilité en termes de HIGH gravité, le rapport échouera s'il existe au moins une vulnérabilité grave ou aucune vulnérabilité grave, mais au moins une vulnérabilité de niveau de gravité supérieur, telle qu'une vulnérabilité de CRITICAL gravité supérieure. HIGH HIGH

Si vous ne spécifiez aucun critère de réussite, alors :

- Le CodeCatalyst rapport généré sur la base de vos rapports bruts n'affichera aucun critère de réussite.
- Les critères de réussite ne seront pas utilisés pour déterminer si l'action de flux de travail associée est réussie ou échoue.

## Visual

Pour configurer les critères de réussite

1. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
2. Choisissez un flux de travail contenant une action qui génère un rapport. Il s'agit du rapport pour lequel vous souhaitez appliquer des critères de réussite. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.

3. Choisissez Modifier.
4. Choisissez Visual.
5. Dans le diagramme du flux de travail, choisissez l'action que vous avez configurée pour générer des CodeCatalyst rapports.
6. Choisissez l'onglet Outputs.
7. Sous Rapports de découverte automatique ou sous Configuration manuelle des rapports, sélectionnez Critères de réussite.

Les critères de réussite apparaissent. En fonction de vos sélections précédentes, vous pouvez voir l'une ou l'ensemble des options suivantes :

#### Taux de réussite

Spécifiez le pourcentage de tests dans un rapport de test qui doivent être réussis pour que le CodeCatalyst rapport associé soit marqué comme réussi. Les valeurs valides incluent les nombres décimaux. Par exemple, 50, 60.5. Les critères relatifs au taux de réussite ne sont appliqués qu'aux rapports de test. Pour plus d'informations sur les rapports de test, consultez [Rapports d'essais](#).

#### Couverture de la ligne

Spécifiez le pourcentage de lignes d'un rapport de couverture de code qui doivent être couvertes pour que le CodeCatalyst rapport associé soit marqué comme transmis. Les valeurs valides incluent les nombres décimaux. Par exemple, 50, 60.5. Les critères de couverture de ligne sont appliqués uniquement aux rapports de couverture de code. Pour plus d'informations sur les rapports de couverture du code, consultez [Rapports sur la couverture du code](#).

#### Couverture des succursales

Spécifiez le pourcentage de branches dans un rapport de couverture de code qui doivent être couvertes pour que le CodeCatalyst rapport associé soit marqué comme transmis. Les valeurs valides incluent les nombres décimaux. Par exemple, 50, 60.5. Les critères de couverture des succursales sont appliqués uniquement aux rapports de couverture par code. Pour plus d'informations sur les rapports de couverture du code, consultez [Rapports sur la couverture du code](#).

#### Vulnérabilités (SCA)

Spécifiez le nombre maximum et la gravité des vulnérabilités autorisées dans le rapport SCA pour que le CodeCatalyst rapport associé soit marqué comme transmis. Pour définir les vulnérabilités, vous devez spécifier :

- La gravité minimale des vulnérabilités que vous souhaitez inclure dans le décompte. Les valeurs valides, de la plus sévère à la moins sévère, sont les suivantes : CRITICALHIGH,MEDIUM,LOW,,INFORMATIONAL.

Par exemple, si vous le souhaitezHIGH, HIGH les CRITICAL vulnérabilités seront comptabilisées.

- Le nombre maximum de vulnérabilités de la gravité spécifiée que vous souhaitez autoriser. Le dépassement de ce nombre entraîne le marquage CodeCatalyst du rapport comme ayant échoué. Les valeurs valides sont des nombres entiers.

Les critères de vulnérabilité ne sont appliqués qu'aux rapports SCA. Pour plus d'informations sur les rapports SCA, consultez[Rapports d'analyse de la composition des logiciels](#).

## Insectes

Spécifiez le nombre maximum et la gravité des bogues autorisés dans le rapport SA pour que le CodeCatalyst rapport associé soit marqué comme passé. Pour spécifier les bogues, vous devez spécifier :

- La gravité minimale des bogues que vous souhaitez inclure dans le décompte. Les valeurs valides, de la plus sévère à la moins sévère, sont les suivantes : CRITICALHIGH,MEDIUM,LOW,,INFORMATIONAL.

Par exemple, si vous le souhaitezHIGH, HIGH les CRITICAL bogues seront comptabilisés.

- Le nombre maximum de bogues de la gravité spécifiée que vous souhaitez autoriser. Le dépassement de ce nombre entraîne le marquage CodeCatalyst du rapport comme ayant échoué. Les valeurs valides sont des nombres entiers.

Les critères de bogues ne sont appliqués qu' PyLint aux rapports ESLint SA. Pour plus d'informations sur les rapports SA, consultez[Rapports d'analyse statiques](#).

## Failles de sécurité



Spécifiez le nombre maximum et la gravité des vulnérabilités de sécurité autorisées dans le rapport SA pour que le CodeCatalyst rapport associé soit marqué comme transmis. Pour spécifier les failles de sécurité, vous devez spécifier :

- La gravité minimale des failles de sécurité que vous souhaitez inclure dans le décompte. Les valeurs valides, de la plus sévère à la moins sévère, sont les suivantes : CRITICALHIGH,MEDIUM,LOW,,INFORMATIONAL.

Par exemple, si vous le souhaitezHIGH, HIGH les failles CRITICAL de sécurité seront comptabilisées.

- Le nombre maximum de failles de sécurité de la gravité spécifiée que vous souhaitez autoriser. Le dépassement de ce nombre entraîne le marquage CodeCatalyst du rapport comme ayant échoué. Les valeurs valides sont des nombres entiers.

Les critères de vulnérabilité de sécurité ne s'appliquent qu' PyLint aux rapports ESLint SA. Pour plus d'informations sur les rapports SA, consultez[Rapports d'analyse statiques](#).

### Problèmes de qualité

Spécifiez le nombre maximum et la gravité des problèmes de qualité autorisés dans le rapport SA pour que le CodeCatalyst rapport associé soit marqué comme réussi. Pour définir les problèmes de qualité, vous devez spécifier :

- La gravité minimale des problèmes de qualité que vous souhaitez inclure dans le décompte. Les valeurs valides, de la plus sévère à la moins sévère, sont les suivantes : CRITICALHIGH,MEDIUM,LOW,,INFORMATIONAL.

Par exemple, si vous le souhaitezHIGH, HIGH les problèmes CRITICAL de qualité seront pris en compte.

- Le nombre maximum de problèmes de qualité de la gravité spécifiée que vous souhaitez autoriser. Le dépassement de ce nombre entraîne le marquage CodeCatalyst du rapport comme ayant échoué. Les valeurs valides sont des nombres entiers.

Les critères relatifs aux problèmes de qualité ne s'appliquent qu' PyLint aux rapports ESLint SA. Pour plus d'informations sur les rapports SA, consultez[Rapports d'analyse statiques](#).

8. Choisissez Commit (Valider).

9. Exécutez votre flux de travail pour CodeCatalyst appliquer des critères de réussite à vos rapports bruts, et régénérez les CodeCatalyst rapports associés en incluant les informations sur les critères de réussite. Pour plus d'informations, consultez [Démarrage d'une exécution de flux de travail](#).

## YAML

Pour configurer les critères de réussite

1. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
2. Choisissez un flux de travail contenant une action qui génère un rapport. Il s'agit du rapport pour lequel vous souhaitez appliquer des critères de réussite. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
3. Choisissez Modifier.
4. Choisissez YAML.
5. Dans le diagramme du flux de travail, choisissez l'action que vous avez configurée pour générer des CodeCatalyst rapports.
6. Dans le volet de détails, choisissez l'onglet Sorties.
7. Dans l'action, dans la AutoDiscoverReports section ou dans la Reports section, ajoutez une SuccessCriteria propriété, ainsi que PassRateLineCoverage, BranchCoverage, Vulnerabilities, StaticAnalysisBugStaticAnalysisSecurity, et StaticAnalysisQuality des propriétés.

Pour une explication de chacune de ces propriétés, consultez le [Créez et testez une référence d'actions](#).

8. Choisissez Commit (Valider).
9. Exécutez votre flux de travail pour CodeCatalyst appliquer des critères de réussite à vos rapports bruts, et régénérez les CodeCatalyst rapports associés avec les informations sur les critères de succès incluses. Pour plus d'informations sur le démarrage d'un flux de travail, consultez [Démarrage d'une exécution de flux de travail](#).

## Exemple : Configuration de rapports

L'exemple suivant montre comment configurer manuellement quatre rapports : un rapport de test, un rapport de couverture du code, un rapport d'analyse de la composition logicielle et un rapport d'analyse statique.

```
Reports:
  MyTestReport:
    Format: JUNITXML
    IncludePaths:
      - "*.xml"
    ExcludePaths:
      - report1.xml
    SuccessCriteria:
      PassRate: 90
  MyCoverageReport:
    Format: CLOVERXML
    IncludePaths:
      - output/coverage/jest/clover.xml
    SuccessCriteria:
      LineCoverage: 75
      BranchCoverage: 75
  MySCARReport:
    Format: SARIFSCA
    IncludePaths:
      - output/sca/reports.xml
    SuccessCriteria:
      Vulnerabilities:
        Number: 5
        Severity: HIGH
  MySARReport:
    Format: ESLINTJSON
    IncludePaths:
      - output/static/eslint.xml
    SuccessCriteria:
      StaticAnalysisBug:
        Number: 10
        Severity: MEDIUM
      StaticAnalysisSecurity:
        Number: 5
        Severity: CRITICAL
      StaticAnalysisQuality:
        Number: 0
```

Severity: INFORMATIONAL

## Support SARIF pour les rapports d'analyse de composition logicielle et d'analyse statique

Le format SARIF (Static Analysis Results Interchange Format) est un format de fichier de sortie disponible dans les rapports d'analyse de composition logicielle et d'analyse statique en. CodeCatalyst L'exemple suivant montre comment configurer manuellement le SARIF dans un rapport d'analyse statique :

```
Reports:
  MySARReport:
    Format: SARIFSA
    IncludePaths:
      - output/sa_report.json
    SuccessCriteria:
      StaticAnalysisFinding:
        Number: 25
        Severity: HIGH
```

CodeCatalyst prend en charge les propriétés SARIF suivantes qui peuvent être utilisées pour optimiser la façon dont les résultats d'analyse apparaîtront dans vos rapports.

### Rubriques

- [Objet sarifLog](#)
- [Objet run](#)
- [Objet toolComponent](#)
- [Objet reportingDescriptor](#)
- [Objet result](#)
- [Objet location](#)
- [Objet physicalLocation](#)
- [Objet logicalLocation](#)
- [Objet fix](#)

## Objet **sarifLog**

Nom	Obligatoire	Description
<code>\$schema</code>	Oui	L'URI du schéma JSON SARIF pour la version <a href="#">2.1.0</a> .
<code>version</code>	Oui	CodeCatalyst ne prend en charge que la version 2.1.0 de SARIF.
<code>runs[]</code>	Oui	Un fichier SARIF contient un tableau d'une ou de plusieurs exécutions, chacune représentant une exécution unique de l'outil d'analyse.

## Objet **run**

Nom	Obligatoire	Description
<code>tool.driver</code>	Oui	<code>toolComponent</code> Objet qui décrit l'outil d'analyse.
<code>tool.name</code>	Non	Propriété qui indique le nom de l'outil utilisé pour effectuer l'analyse.
<code>results[]</code>	Oui	Les résultats de l'outil d'analyse qui sont affichés sur CodeCatalyst.

Objet **toolComponent**

Nom	Obligatoire	Description
name	Oui	Nom de l'outil d'analyse.
properties.artifactScanned	Non	Nombre total d'artefacts analysés par l'outil.
rules[]	Oui	Tableau d' reporting Descriptor objets représentant des règles. Sur la base de ces règles, l'outil d'analyse détecte les problèmes dans le code analysé.

Objet **reportingDescriptor**

Nom	Obligatoire	Description
id	Oui	Identifiant unique de la règle utilisée pour référencer un résultat.  Longueur maximale : 1 024 caractères
name	Non	Le nom d'affichage de la règle.  Longueur maximale : 1 024 caractères
shortDescription.text	Non	Description abrégée de la règle.  Longueur maximale : 3 000 caractères

Nom	Obligatoire	Description
<code>fullDescription.text</code>	Non	Description complète de la règle.  Longueur maximale : 3 000 caractères
<code>helpUri</code>	Non	Chaîne qui peut être localisée pour contenir l'URI absolu de la documentation principale de la règle.  Longueur maximale : 3 000 caractères
<code>properties.unscore</code>	Non	Un indicateur qui indique si le résultat du scan a été noté.
<code>properties.score.severity</code>	Non	Ensemble fixe de chaînes qui spécifient le niveau de gravité du résultat.  Longueur maximale : 1 024 caractères
<code>properties.cvssv3_baseSeverity</code>	Non	Une évaluation qualitative de la gravité du <a href="#">Common Vulnerability Scoring System v3.1</a> .
<code>properties.cvssv3_baseScore</code>	Non	Un score de base CVSS v3 compris entre <a href="#">0,0 et 10,0</a> .
<code>properties.cvssv2_severity</code>	Non	Si les valeurs CVSS v3 ne sont pas disponibles, CodeCatalyst recherche les valeurs CVSS v2.

Nom	Obligatoire	Description
<code>properties.cvssv2_score</code>	Non	Un score de base CVSS v2 compris entre <a href="#">0,0 et 10,0</a> .
<code>properties.severity</code>	Non	Ensemble fixe de chaînes qui spécifient le niveau de gravité du résultat.  Longueur maximale : 1 024 caractères
<code>defaultConfiguration.level</code>	Non	La sévérité par défaut d'une règle.

### Objet **result**

Nom	Obligatoire	Description
<code>ruleId</code>	Oui	Identifiant unique de la règle utilisée pour référencer un résultat.  Longueur maximale : 1 024 caractères
<code>ruleIndex</code>	Oui	Index de la règle associée dans le composant de <code>l'outilrules[]</code> .
<code>message.text</code>	Oui	Un message qui décrit le résultat et affiche le message correspondant à chaque résultat.  Longueur maximale : 3 000 caractères



Nom	Obligatoire	Description
rank	Non	Une valeur comprise entre 0,0 et 100,0 inclus qui représente la priorité ou l'importance du résultat. Cette échelle indique 0,0 étant la priorité la plus basse et 100,0 la priorité la plus élevée.
level	Non	La sévérité du résultat.  Longueur maximale : 1 024 caractères
properties.unscore	Non	Un indicateur qui indique si le résultat du scan a été noté.
properties.score.severity	Non	Ensemble fixe de chaînes qui spécifient le niveau de gravité du résultat.  Longueur maximale : 1 024 caractères
properties.cvssv3_baseSeverity	Non	Une évaluation qualitative de la gravité du <a href="#">Common Vulnerability Scoring System v3.1</a> .
properties.cvssv3_baseScore	Non	Un score de base CVSS v3 compris entre <a href="#">0,0 et 10,0</a> .
properties.cvssv2_severity	Non	Si les valeurs CVSS v3 ne sont pas disponibles, CodeCatalyst recherche les valeurs CVSS v2.

Nom	Obligatoire	Description
<code>properties.cvssv2_score</code>	Non	Un score de base CVSS v2 compris entre <a href="#">0,0</a> et <a href="#">10,0</a> .
<code>properties.severity</code>	Non	Ensemble fixe de chaînes qui spécifient le niveau de gravité du résultat.  Longueur maximale : 1 024 caractères
<code>locations[]</code>	Oui	Ensemble d'emplacements où le résultat a été détecté. Un seul emplacement doit être inclus, sauf si le problème ne peut être résolu qu'en apportant une modification à chaque emplacement spécifié. CodeCatalyst utilise la première valeur du tableau d'emplacements pour annoter le résultat.  Nombre maximum d'locationobjets : 10
<code>relatedLocations[]</code>	Non	Liste des références de lieux supplémentaires figurant dans le résultat.  Nombre maximum d'locationobjets : 50

Nom	Obligatoire	Description
<code>fixes[]</code>	Non	Tableau d' <code>fix</code> objets représentant les recommandations fournies par l'outil de numérisation. CodeCatalyst utilise la première recommandation du <code>fixes</code> tableau.

### Objet **location**

Nom	Obligatoire	Description
<code>physicalLocation</code>	Oui	Identifie l'artefact et la région.
<code>logicalLocations[]</code>	Non	Ensemble de lieux décrits par leur nom sans référence à l'artefact.

### Objet **physicalLocation**

Nom	Obligatoire	Description
<code>artifactLocation.uri</code>	Oui	L'URI indiquant l'emplacement d'un artefact, généralement un fichier dans le référentiel ou généré lors d'une compilation.
<code>fileLocation.uri</code>	Non	L'URI de secours indiquant l'emplacement du fichier. Ceci est utilisé si les <code>artifactLocation.uri</code> retours sont vides.
<code>region.startLine</code>	Oui	Numéro de ligne du premier caractère de la région.

Nom	Obligatoire	Description
<code>region.startColumn</code>	Oui	Numéro de colonne du premier caractère de la région.
<code>region.endLine</code>	Oui	Numéro de ligne du dernier caractère de la région.
<code>region.endColumn</code>	Oui	Numéro de colonne du dernier caractère de la région.

### Objet **logicalLocation**

Nom	Obligatoire	Description
<code>fullyQualifiedName</code>	Non	Informations supplémentaires qui décrivent l'emplacement du résultat.  Longueur maximale : 1 024 caractères

### Objet **fix**

Nom	Obligatoire	Description
<code>description.text</code>	Non	Un message qui affiche une recommandation pour chaque résultat.  Longueur maximale : 3 000 caractères
<code>artifactChanges.[0].artifactLocation.uri</code>	Non	L'URI indiquant l'emplacement de l'artefact qui doit être mis à jour.

## En utilisant universal-test-runner

Les actions de test s'intègrent à l'outil `universal-test-runner` de ligne de commande open source. Cet outil fournit des fonctionnalités de test avancées, telles que le fait de réessayer un ou plusieurs cas de test à partir d'un rapport de test. `universal-test-runner` utilise le [protocole d'exécution des tests](#) pour exécuter vos tests pour n'importe quel langage dans un framework donné. `universal-test-runner` prend en charge les frameworks suivants :

- [Gradle](#)
- [Jest](#)
- [Maven](#)
- [pytest](#)
- [.NET](#)

`universal-test-runner` est installé uniquement sur les images sélectionnées pour les actions de test. Si vous configurez une action de test pour utiliser un Docker Hub ou Amazon ECR personnalisé, vous devez l'installer manuellement `universal-test-runner` pour activer les fonctionnalités de test avancées. Pour ce faire, installez Node.js (14 ou supérieur) sur l'image, puis installez-le `universal-test-runner` à l'aide de la commande shell- Run: `npm install -g @aws/universal-test-runner`. Pour plus d'informations sur l'installation de Node.js dans votre conteneur via des commandes shell, consultez la section [Installation et mise à jour du gestionnaire de versions de Node](#).

Pour plus d'informations `universal-test-runner`, voir [Qu'est-ce que c'est universal-test-runner ?](#)

### Visual

À utiliser `universal-test-runner` dans l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
3. Choisissez le nom de votre flux de travail.
4. Choisissez Modifier.
5. Choisissez Visual.
6. Choisissez Actions.
7. Dans Actions, sélectionnez Test.

8. Dans l'onglet Configuration, complétez le champ des commandes Shell en mettant à jour l'exemple de code avec les frameworks pris en charge de votre choix. Par exemple, pour utiliser un framework pris en charge, vous devez utiliser une Run commande similaire à la suivante.

```
- Run: run-tests <framework>
```

Si le framework que vous souhaitez n'est pas pris en charge, pensez à fournir un adaptateur ou un coureur personnalisé. Pour une description du champ de commandes Shell, consultez [Steps](#).

9. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
10. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

À utiliser universal-test-runner dans l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
3. Choisissez le nom de votre flux de travail.
4. Choisissez Modifier.
5. Choisissez YAML.
6. Choisissez Actions.
7. Dans Actions, sélectionnez Test.
8. Modifiez le code YAML en fonction de vos besoins. Par exemple, pour utiliser un framework pris en charge, vous devez utiliser une Run commande similaire à la suivante.

```
Configuration:  
Steps:  
  - Run: run-tests <framework>
```

Si le framework que vous souhaitez n'est pas pris en charge, pensez à fournir un adaptateur ou un coureur personnalisé. Pour une description de la propriété Steps, voir [Steps](#).

9. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
10. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Bonnes pratiques

Lorsque vous utilisez les fonctionnalités de test fournies par CodeCatalyst, nous vous recommandons de suivre ces meilleures pratiques.

### Rubriques

- [Découverte automatique](#)
- [Critères de réussite](#)
- [Inclure/exclure des chemins](#)

### Découverte automatique

Lorsque vous configurez des actions dans CodeCatalyst, la découverte automatique vous permet de découvrir automatiquement les résultats de divers outils, tels que les rapports de test JUnit, et de générer des CodeCatalyst rapports pertinents à partir de ceux-ci. La découverte automatique permet de garantir que les rapports continuent d'être générés même si les noms ou les chemins des sorties découvertes changent. Lorsque de nouveaux fichiers sont ajoutés, il les découvre CodeCatalyst automatiquement et produit des rapports pertinents. Toutefois, si vous utilisez la détection automatique, il est important de prendre en compte certains des aspects suivants de cette fonctionnalité :

- Lorsque vous activez la détection automatique dans votre action, tous les rapports découverts automatiquement du même type partagent les mêmes critères de réussite. Par exemple, un critère partagé tel que le taux de réussite minimum s'appliquerait à tous les rapports de test découverts automatiquement. Si vous avez besoin de critères différents pour les rapports du même type, vous devez configurer explicitement chacun de ces rapports.
- La détection automatique peut également détecter les rapports produits par vos dépendances et, si des critères de réussite sont configurés, l'action sur ces rapports risque d'échouer. Ce problème peut être résolu en mettant à jour la configuration du chemin d'exclusion.
- Il n'est pas garanti que la découverte automatique produise la même liste de rapports à chaque fois, car elle analyse l'action au moment de l'exécution. Dans le cas où vous souhaitez qu'un rapport particulier soit toujours produit, vous devez configurer les rapports de manière explicite. Par exemple, si les tests cessaient de s'exécuter dans le cadre de votre build, le framework de test ne produirait aucun résultat et, par conséquent, aucun rapport de test ne serait produit et l'action pourrait réussir. Si vous souhaitez que le succès de votre action dépende de ce test en particulier, vous devez configurer explicitement ce rapport.

**i** Tip

Lorsque vous commencez à travailler sur un projet nouveau ou existant, utilisez la détection automatique pour l'ensemble du répertoire du projet (y compris `**/*`). Cela implique la génération de rapports pour tous les fichiers de votre projet, y compris ceux des sous-répertoires.

Pour plus d'informations, consultez [Configuration des rapports](#).

## Critères de réussite

Vous pouvez appliquer des seuils de qualité à vos rapports en configurant des critères de réussite. Par exemple, si deux rapports de couverture de code ont été découverts automatiquement, l'un avec une couverture de ligne de 80 % et l'autre avec une couverture de ligne de 60 %, les options suivantes s'offrent à vous :

- Définissez les critères de réussite de la détection automatique pour la couverture des lignes à 80 %. Cela entraînerait l'adoption du premier rapport et l'échec du second rapport, ce qui entraînerait l'échec de l'action globale. Pour débloquer le flux de travail, ajoutez de nouveaux tests à votre projet jusqu'à ce que la couverture linéaire du deuxième rapport dépasse 80 %.
- Définissez les critères de réussite de la détection automatique pour la couverture des lignes à 60 %. Cela entraînerait la transmission des deux rapports, ce qui se traduirait par le succès de l'action. Vous pourriez ensuite travailler à augmenter la couverture du code dans le deuxième rapport. Toutefois, avec cette approche, vous ne pouvez pas garantir que le taux de couverture indiqué dans le premier rapport ne descendra pas en dessous de 80 %.
- Configurez explicitement l'un des rapports ou les deux à l'aide de l'éditeur visuel ou en ajoutant une section YAML et un chemin explicites pour chaque rapport. Cela vous permettra de configurer des critères de réussite distincts et des noms personnalisés pour chaque rapport. Toutefois, avec cette approche, l'action peut échouer si les chemins des rapports changent.


Pour plus d'informations, consultez [Configuration des critères de réussite pour les rapports](#).

## Inclure/exclure des chemins

Lorsque vous examinez les résultats des actions, vous pouvez ajuster la liste des rapports générés CodeCatalyst par en configurant `IncludePaths` et `ExcludePaths`.



- Permet `IncludePaths` de spécifier les fichiers et les chemins de fichiers que vous CodeCatalyst souhaitez inclure lors de la recherche de rapports. Par exemple, si vous le spécifiez `"/test/report/*"`, CodeCatalyst recherche l'intégralité de l'image de construction utilisée par l'action à la recherche du `/test/report/` répertoire. Lorsqu'il trouve ce répertoire, CodeCatalyst il recherche des rapports dans ce répertoire.

 Note

Pour les rapports configurés manuellement, `IncludePaths` il doit s'agir d'un modèle global correspondant à un seul fichier.

- Permet `ExcludePaths` de spécifier les fichiers et les chemins de fichiers que vous CodeCatalyst souhaitez exclure lors de la recherche de rapports. Par exemple, si vous le spécifiez `"/test/reports/**/*"`, ne CodeCatalyst recherchera pas de fichiers dans le `/test/reports/` répertoire. Pour ignorer tous les fichiers d'un répertoire, utilisez le modèle `**/*` glob.

Vous trouverez ci-dessous des exemples de modèles globaux possibles.

Modèle	Description
<code>*.*</code>	Correspond à tous les noms d'objets du répertoire actuel contenant un point
<code>*.xml</code>	Correspond à tous les noms d'objets du répertoire actuel se terminant par <code>.xml</code>
<code>*.{xml,txt}</code>	Correspond à tous les noms d'objets du répertoire actuel se terminant par <code>.xml</code> ou <code>.txt</code>
<code>**/*.xml</code>	Fait correspondre les noms d'objets dans tous les répertoires se terminant par <code>.xml</code>
<code>testFolder</code>	Correspond à un objet appelé <code>testFolder</code> , en le traitant comme un fichier

Modèle	Description
<code>testFolder/*</code>	Fait correspondre les objets d'un niveau du sous-dossier à partir de <code>testFolder</code> , tels que <code>testFolder/file.xml</code>
<code>testFolder/*/*</code>	Fait correspondre les objets situés à deux niveaux du sous-dossier à partir de <code>testFolder</code> , tels que <code>testFolder/reportsFolder/file.xml</code>
<code>testFolder/**</code>	Correspond au sous-dossier <code>testFolder</code> , ainsi qu'aux fichiers sous <code>testFolder</code> , tels que <code>testFolder/file.xml</code> et <code>testFolder/otherFolder/file.xml</code>

CodeCatalyst interprète les modèles globulaires comme suit :

- Le caractère slash (/) sépare les répertoires dans les chemins de fichiers.
- L'astérisque (\*) correspond à zéro ou plusieurs caractères d'un composant de nom sans dépasser les limites d'un dossier.
- Un double astérisque (\*\*) correspond à zéro ou plusieurs caractères d'un composant de nom dans tous les répertoires.

#### Note

`ExcludePaths` a la priorité sur `IncludePaths`. Si `IncludePaths` et `ExcludePaths` incluent le même dossier, ce dossier n'est pas scanné pour les rapports.

## Travailler avec des tests

Cette section décrit comment afficher et configurer vos actions de test.

### Rubriques

- [Afficher les résultats d'une action de test](#)

- [Ignorer les tests ayant échoué](#)
- [Réessayer les scénarios de test](#)

## Afficher les résultats d'une action de test

Utilisez les instructions suivantes pour afficher les résultats d'une action de test, y compris les journaux, les rapports et les variables générés.

Pour afficher les résultats d'une action de test

1. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
2. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
3. Dans le diagramme du flux de travail, choisissez le nom de votre action de test (par exemple, Test).
4. Pour afficher les journaux générés par une action, choisissez Logs. Les journaux des différentes phases d'action sont affichés. Vous pouvez agrandir ou réduire les journaux selon vos besoins.
5. Pour afficher les rapports de test produits par l'action de test, sélectionnez Rapports. Pour plus d'informations, consultez [Types de rapports de test](#).
6. Pour afficher la configuration utilisée pour l'action de test, choisissez Configuration. Pour plus d'informations, consultez [Ajouter l'action de test](#).
7. Pour afficher les variables utilisées par l'action de test, sélectionnez Variables. Pour plus d'informations, voir [Utilisation des variables](#).

## Ignorer les tests ayant échoué

Si votre action comporte plusieurs commandes de test, vous souhaitez peut-être autoriser les commandes de test suivantes à s'exécuter même si une commande précédente échoue. Par exemple, dans les commandes suivantes, vous souhaitez peut-être toujours `test2` exécuter, même en cas d'`test1` échec.

Steps:

- Run: `npm install`
- Run: `npm run test1`
- Run: `npm run test2`

Normalement, lorsqu'une étape renvoie une erreur, Amazon CodeCatalyst arrête l'action du flux de travail et la marque comme ayant échoué. Vous pouvez autoriser la poursuite des étapes d'action en redirigeant le résultat d'erreur vers `null`. Vous pouvez le faire en ajoutant `2>/dev/null` à la commande. Avec cette modification, l'exemple précédent ressemblerait à ce qui suit.

**Steps:**

- Run: `npm install`
- Run: `npm run test1 2>/dev/null`
- Run: `npm run test2`

Dans le second extrait de code, le statut de la `npm install` commande sera respecté, mais toute erreur renvoyée par la `npm run test1` commande sera ignorée. Par conséquent, la `npm run test2` commande est exécutée. Ainsi, vous pouvez consulter les deux rapports en même temps, qu'une erreur se produise ou non.

## Réessayer les scénarios de test

Si votre rapport échoue en raison de plusieurs cas de test, vous ne pouvez réessayer que ces tests individuels. Cela vous permet de vérifier rapidement la qualité de vos scénarios de test et de déterminer les prochaines étapes pour résoudre vos problèmes, par exemple en cas de rupture de dépendance ou en lançant une nouvelle exécution du flux de travail. Votre action de test intègre le fait de `universal-test-runner` réessayer uniquement les cas de test sélectionnés, au lieu de réessayer l'ensemble de l'action. Vous ne pouvez réessayer qu'un seul ensemble de cas de test sélectionnés par action à la fois, et ne réessayer que cinq fois par rapport de test. Pour plus d'informations, consultez [En utilisant universal-test-runner](#).

**Note**

Si vous réessayez des scénarios de test sur un rapport, cela n'aura aucun effet sur le statut du flux de travail qui a généré le rapport d'origine.

Suivez les instructions ci-dessous pour réessayer les scénarios de test dans vos rapports.

Pour réessayer les scénarios de test d'un rapport

1. Dans le volet de navigation, choisissez Rapports.
2. Choisissez le nom de votre rapport. Vous pouvez filtrer en fonction du nom, du statut, du référentiel, de la branche ou du type de rapport.

3. Sous le nom du rapport, sélectionnez Résultats.
4. Sélectionnez les scénarios de test que vous souhaitez réessayer, choisissez Réexécuter, puis sélectionnez Scénarios de test sélectionnés.
5. Lorsque votre nouvelle tentative est terminée, choisissez Actualiser sur la bannière et consultez les résultats mis à jour.

## Déploiement à l'aide de workflows dans CodeCatalyst

À l'aide de [CodeCatalyst flux de travail](#), vous pouvez déployer des applications et d'autres ressources sur diverses cibles telles qu'Amazon ECS AWS Lambda, etc.

### Rubriques

- [Comment déployer une application ?](#)
- [Liste des actions de déploiement](#)
- [Avantages des actions de déploiement](#)
- [Alternatives aux actions de déploiement](#)
- [Tutoriel : Déployer une application sans serveur à l'aide de AWS CloudFormation](#)
- [Tutoriel : Déploiement d'une application sur Amazon ECS](#)
- [Tutoriel : Déploiement d'une application sur Amazon EKS](#)
- [Ajout de l'action « Deploy AWS CloudFormation stack »](#)
- [Ajout de l'action « Déployer sur Amazon ECS »](#)
- [Ajout de l'action « Déployer vers le cluster Kubernetes »](#)
- [Ajout de l'action « AWS CDK déployer »](#)
- [Utilisation de déploiements](#)

## Comment déployer une application ?

Pour déployer une application ou une ressource CodeCatalyst, vous devez d'abord créer un flux de travail, puis spécifier une action de déploiement à l'intérieur de celui-ci. Une action de déploiement est un élément constitutif du flux de travail qui définit ce que vous souhaitez déployer, où vous souhaitez le déployer et comment vous souhaitez le déployer (par exemple, en utilisant un schéma bleu/vert). Vous ajoutez une action de déploiement à votre flux de travail à l'aide de l'éditeur visuel de la CodeCatalyst console, ou éditeur YAML.

Les étapes de haut niveau pour déployer une application ou une ressource sont les suivantes.

Pour déployer une application (tâches de haut niveau)

1. Dans votre CodeCatalyst projet, vous ajoutez le code source d'une application que vous souhaitez déployer. Pour plus d'informations, consultez [Utilisation des référentiels de sources dans CodeCatalyst](#).
2. Dans votre CodeCatalyst projet, vous créez un flux de travail. Le flux de travail vous permet de définir comment créer, tester et déployer votre application. Pour plus d'informations, consultez [Commencer à utiliser les flux de travail dans CodeCatalyst](#).
3. Dans le flux de travail, vous ajoutez un déclencheur, une action de création et éventuellement une action de test. Pour plus d'informations, consultez [Utilisation de déclencheurs](#), [Ajouter l'action de construction](#) et [Ajouter l'action de test](#).
4. Dans le flux de travail, vous ajoutez une action de déploiement. Vous pouvez choisir parmi plusieurs actions de déploiement CodeCatalyst proposées sur votre application vers différentes cibles, telles qu'Amazon ECS. (Vous pouvez également utiliser une action de génération ou une GitHub action pour déployer votre application. Pour plus d'informations sur l'action de génération et GitHub les actions, consultez [Alternatives aux actions de déploiement](#).)
5. Vous démarrez le flux de travail manuellement ou automatiquement par le biais d'un déclencheur. Le flux de travail exécute les actions de génération, de test et de déploiement en séquence pour déployer votre application et vos ressources vers la cible. Pour plus d'informations, consultez [Démarrage d'une exécution de flux de travail](#).

## Liste des actions de déploiement

Les actions de déploiement disponibles sont les suivantes :

- Déployer une AWS CloudFormation pile : cette action crée une CloudFormation pile sur la AWS base d'un [AWS CloudFormation modèle](#) ou d'un [AWS Serverless Application Model modèle](#) que vous fournissez. Pour plus d'informations, consultez [Ajout de l'action « Deploy AWS CloudFormation stack »](#).
- Déployer sur Amazon ECS : cette action enregistre un fichier de [définition de tâche](#) que vous fournissez. Pour plus d'informations, consultez [Ajout de l'action « Déployer sur Amazon ECS »](#).
- Déployer vers un cluster Kubernetes : cette action déploie une application sur un cluster Amazon Elastic Kubernetes Service. Pour plus d'informations, consultez [Ajout de l'action « Déployer vers le cluster Kubernetes »](#).

- AWS CDK déployer — Cette action déploie une [AWS CDK application](#) dans. AWS Pour plus d'informations, consultez [Ajout de l'action « AWS CDK déployer »](#).

### Note

D'autres CodeCatalyst actions peuvent déployer des ressources ; toutefois, elles ne sont pas considérées comme des actions de déploiement car leurs informations de déploiement n'apparaissent pas sur la page Environnements. Pour en savoir plus sur la page Environnements et sur l'affichage des déploiements, consultez [Utilisation des environnements](#) et [Utilisation de déploiements](#).

## Avantages des actions de déploiement

L'utilisation d'actions de déploiement dans un flux de travail présente les avantages suivants :

- Historique des déploiements : consultez l'historique de vos déploiements pour vous aider à gérer et à communiquer les modifications apportées aux logiciels que vous avez déployés.
- Traçabilité : suivez l'état de vos déploiements via la CodeCatalyst console et découvrez quand et où chaque révision d'application a été déployée.
- Annulations : annulez automatiquement les déploiements en cas d'erreur. Vous pouvez également configurer des alarmes pour activer les annulations de déploiement.
- Surveillance : surveillez votre déploiement au fur et à mesure qu'il progresse au cours des différentes étapes de votre flux de travail.
- Intégration avec d'autres CodeCatalyst fonctionnalités : stockez le code source, puis créez-le, testez-le et déployez-le, le tout à partir d'une seule application.

## Alternatives aux actions de déploiement

Vous n'êtes pas obligé d'utiliser des actions de déploiement, bien qu'elles soient recommandées car elles offrent les avantages décrits dans la section précédente. Au lieu de cela, vous pouvez utiliser les [CodeCatalyst actions](#) suivantes :

- Une action de construction.

Généralement, vous utilisez des actions de génération si vous souhaitez effectuer un déploiement sur une cible pour laquelle aucune action de déploiement correspondante n'existe, ou si vous souhaitez mieux contrôler la procédure de déploiement. Pour plus d'informations sur l'utilisation des actions de génération pour déployer des ressources, consultez [Création à l'aide de flux de travail dans CodeCatalyst](#).

- Une GitHub action.

Vous pouvez utiliser une [GitHub action](#) dans un CodeCatalyst flux de travail pour déployer des applications et des ressources (au lieu d'une CodeCatalyst action). Pour plus d'informations sur l'utilisation GitHub des actions dans un CodeCatalyst flux de travail, voir [Ajouter des GitHub actions](#)

Vous pouvez également utiliser les AWS services suivants pour déployer votre application, si vous ne souhaitez pas utiliser de CodeCatalyst flux de travail pour ce faire :

- AWS CodeDeploy — voir [Qu'est-ce que c'est CodeDeploy ?](#)
- AWS CodeBuild et AWS CodePipeline — voir [Qu'est-ce que c'est AWS CodeBuild ?](#) et [qu'est-ce que c'est AWS CodePipeline ?](#)
- AWS CloudFormation — voir [Qu'est-ce que c'est AWS CloudFormation ?](#)

Utilisation CodeDeploy, CodeBuild CodePipeline, et CloudFormation services pour les déploiements complexes en entreprise.

## Tutoriel : Déployer une application sans serveur à l'aide de AWS CloudFormation

Dans ce didacticiel, vous apprendrez à créer, tester et déployer une application sans serveur à l'aide d'un flux de travail et AWS CloudFormation.

L'application présentée dans ce didacticiel est une simple application Web qui produit un message « Hello World ». Il se compose d'une AWS Lambda fonction et d'un Amazon API Gateway, et vous le créez à l'aide de [AWS Serverless Application Model \(AWS SAM\)](#), qui est une extension de [AWS CloudFormation](#).

### Rubriques

- [Prérequis](#)
- [Étape 1 : Création d'un référentiel source](#)



- [Étape 2 : créer des AWS rôles](#)
- [Étape 3 : ajouter AWS des rôles à CodeCatalyst](#)
- [Étape 4 : Création d'un compartiment Amazon S3](#)
- [Étape 5 : Ajouter des fichiers source](#)
- [Étape 6 : créer et exécuter un flux de travail](#)
- [Étape 7 : Apporter une modification](#)
- [Nettoyage](#)

## Prérequis

Avant de commencer :

- Vous avez besoin d'un CodeCatalyst espace avec un AWS compte connecté. Pour plus d'informations, consultez [Création d'un espace prenant en charge les utilisateurs de AWS Builder ID](#).
- Dans votre espace, vous avez besoin d'un CodeCatalyst projet vide, créé à partir de zéro, appelé :

```
codecatalyst-cfn-project
```

Pour plus d'informations, consultez [Création d'un projet vide dans Amazon CodeCatalyst](#).

- Dans votre projet, vous avez besoin d'un CodeCatalyst environnement appelé :

```
codecatalyst-cfn-environment
```

Configurez cet environnement comme suit :

- Choisissez n'importe quel type, tel que Non-production.
- AWS Connectez-y votre compte.

Pour plus d'informations, consultez [Utilisation des environnements](#).

## Étape 1 : Création d'un référentiel source

Au cours de cette étape, vous créez un référentiel source dans CodeCatalyst. Ce référentiel est utilisé pour stocker les fichiers sources du didacticiel, tels que le fichier de fonction Lambda.

Pour plus d'informations sur les référentiels sources, consultez [Création d'un référentiel source](#).

## Pour créer un référentiel source

1. Dans CodeCatalyst le volet de navigation, sélectionnez Code, puis sélectionnez Référentiels sources.
2. Choisissez Ajouter un référentiel, puis sélectionnez Créer un référentiel.
3. Dans Nom du référentiel, entrez :

```
codecatalyst-cfn-source-repository
```

4. Choisissez Créer.

Vous venez de créer un dépôt appelé `codecatalyst-cfn-source-repository`.

## Étape 2 : créer des AWS rôles

Au cours de cette étape, vous allez créer les rôles AWS IAM suivants :

- Rôle de déploiement : accorde à l'action CodeCatalyst Deploy AWS CloudFormation Stack l'autorisation d'accéder à votre AWS compte et au CloudFormation service sur lesquels vous déploierez votre application sans serveur. L'action Deploy AWS CloudFormation stack fait partie de votre flux de travail.
- Rôle de création : accorde à l'action de CodeCatalyst création l'autorisation d'accéder à votre AWS compte et d'écrire sur Amazon S3 où votre package d'application sans serveur sera stocké. L'action de création fait partie de votre flux de travail.
- Rôle Stack : CloudFormation autorise la lecture et la modification des ressources spécifiées dans le AWS SAM modèle que vous fournirez ultérieurement. Accorde également l'autorisation de CloudWatch.

Pour plus d'informations sur les rôles IAM, consultez la section Rôles [IAM dans le Guide](#) de l'AWS Identity and Access Management utilisateur.

### Note

Pour gagner du temps, vous pouvez créer un seul rôle, appelé `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle, au lieu des trois rôles répertoriés précédemment. Pour plus d'informations, consultez [Création du CodeCatalystWorkflowDevelopmentRole-\*spaceName\* rôle pour votre compte et votre](#)

espace. Sachez que le CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle dispose d'autorisations très étendues qui peuvent présenter un risque de sécurité. Nous vous recommandons de n'utiliser ce rôle que dans les didacticiels et les scénarios où la sécurité est moins préoccupante. Ce didacticiel part du principe que vous créez les trois rôles répertoriés précédemment.

### Note

Un rôle d'exécution Lambda est également requis, mais vous n'avez pas besoin de le créer maintenant car le `sam-template.yml` fichier le crée pour vous lorsque vous exécutez le flux de travail à l'étape 5.

## Pour créer un rôle de déploiement

1. Créez une politique pour le rôle, comme suit :
  - a. Connectez-vous à AWS.
  - b. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
  - c. Dans le panneau de navigation, choisissez Politiques (Politiques).
  - d. Sélectionnez Create policy (Créer une politique).
  - e. Choisissez l'onglet JSON.
  - f. Supprimez le code existant.
  - g. Collez le code suivant :

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "cloudformation:CreateStack",
      "cloudformation>DeleteStack",
      "cloudformation:Describe*",
      "cloudformation:UpdateStack",
      "cloudformation:CreateChangeSet",
      "cloudformation>DeleteChangeSet",
      "cloudformation:ExecuteChangeSet",
```

```
        "cloudformation:SetStackPolicy",
        "cloudformation:ValidateTemplate",
        "cloudformation:List*",
        "iam:PassRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
}]
}
```

### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

- h. Choisissez Suivant : Balises.
- i. Choisissez Suivant : Vérification.
- j. Dans Nom, entrez :

```
codecatalyst-deploy-policy
```

- k. Choisissez Créer une politique.

Vous venez de créer une politique d'autorisation.

2. Créez le rôle de déploiement comme suit :
  - a. Dans le volet de navigation, sélectionnez Rôles, puis Créer un rôle.
  - b. Choisissez Politique de confiance personnalisée.
  - c. Supprimez la politique de confiance personnalisée existante.
  - d. Ajoutez la politique de confiance personnalisée suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
        "Sid": "",
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "codecatalyst-runner.amazonaws.com",
                "codecatalyst.amazonaws.com"
            ]
        },
        "Action": "sts:AssumeRole"
    }
]
```

- e. Choisissez Suivant.
- f. Dans Politiques d'autorisations, recherchez `codecatalyst-deploy-policy` et cochez la case correspondante.
- g. Choisissez Suivant.
- h. Dans Nom du rôle, entrez :

**codecatalyst-deploy-role**

- i. Pour la description du rôle, entrez :

**CodeCatalyst deploy role**

- j. Sélectionnez Créer un rôle.

Vous venez de créer un rôle de déploiement avec une politique de confiance et une politique d'autorisations.

3. Obtenez l'ARN du rôle de déploiement, comme suit :
  - a. Dans le panneau de navigation, choisissez Roles (Rôles).
  - b. Dans le champ de recherche, entrez le nom du rôle que vous venez de créer (`codecatalyst-deploy-role`).
  - c. Choisissez le rôle dans la liste.

La page Résumé du rôle apparaît.


- d. En haut, copiez la valeur ARN.

Vous avez maintenant créé le rôle de déploiement avec les autorisations appropriées et obtenu son ARN.

Pour créer un rôle de build

1. Créez une politique pour le rôle, comme suit :
  - a. Connectez-vous à AWS.
  - b. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
  - c. Dans le panneau de navigation, choisissez Politiques (Politiques).
  - d. Sélectionnez Create policy (Créer une politique).
  - e. Choisissez l'onglet JSON.
  - f. Supprimez le code existant.
  - g. Collez le code suivant :

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

 Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

- h. Choisissez Suivant : Balises.

- i. Choisissez Suivant : Vérification.
- j. Dans Nom, entrez :

**codecatalyst-build-policy**

- k. Choisissez Créer une politique.

Vous venez de créer une politique d'autorisation.

2. Créez le rôle de build, comme suit :

- a. Dans le volet de navigation, sélectionnez Rôles, puis Créer un rôle.
- b. Choisissez Politique de confiance personnalisée.
- c. Supprimez la politique de confiance personnalisée existante.
- d. Ajoutez la politique de confiance personnalisée suivante :

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

- e. Choisissez Suivant.
- f. Dans Politiques d'autorisations, recherchez `codecatalyst-build-policy` et cochez la case correspondante.
- g. Choisissez Suivant.
- h. Dans Nom du rôle, entrez :

**codecatalyst-build-role**

- i. Pour la description du rôle, entrez :

**CodeCatalyst build role**

- j. Sélectionnez Créer un rôle.

Vous avez maintenant créé un rôle de build avec une politique de confiance et une politique d'autorisations.

3. Obtenez l'ARN du rôle de build, comme suit :
  - a. Dans le panneau de navigation, choisissez Roles (Rôles).
  - b. Dans le champ de recherche, entrez le nom du rôle que vous venez de créer (codecatalyst-build-role).
  - c. Choisissez le rôle dans la liste.

La page Résumé du rôle apparaît.

- d. En haut, copiez la valeur ARN.

Vous avez maintenant créé le rôle de build avec les autorisations appropriées et obtenu son ARN.

Pour créer un rôle de pile

1. Connectez-vous à AWS l'aide du compte sur lequel vous souhaitez déployer votre stack.
2. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
3. Créez le rôle de pile comme suit :
  - a. Dans le panneau de navigation, choisissez Roles (Rôles).
  - b. Sélectionnez Create role (Créer un rôle).
  - c. Choisissez Service AWS .
  - d. Dans la section Cas d'utilisation, choisissez CloudFormation dans la liste déroulante.
  - e. Sélectionnez le bouton CloudFormation radio.
  - f. En bas de la page, choisissez Next.
  - g. À l'aide de la zone de recherche, recherchez les politiques d'autorisation suivantes, puis cochez leurs cases respectives.



**Note**

Si vous recherchez une politique et qu'elle n'apparaît pas, assurez-vous de choisir Effacer les filtres et réessayez.

- CloudWatchFullAccess
- AWS CloudFormationFullAccess
- IAM FullAccess
- AWS Lambda\_ FullAccess
- API Amazon GatewayAdministrator
- Amazon S3 FullAccess
- Amazon EC2 ContainerRegistryFullAccess

La première politique autorise l'accès CloudWatch pour activer les rollbacks de pile lorsqu'une alarme se produit.

Les politiques restantes permettent AWS SAM d'accéder aux services et aux ressources de la pile qui seront déployés dans ce didacticiel. Pour plus d'informations, consultez la section [Permissions](#) dans le guide du AWS Serverless Application Model développeur.

- h. Choisissez Suivant.
- i. Dans Nom du rôle, entrez :

**codecatalyst-stack-role**

- j. Sélectionnez Créer un rôle.
4. Obtenez l'ARN du rôle de pile, comme suit :
    - a. Dans le panneau de navigation, choisissez Roles (Rôles).
    - b. Dans le champ de recherche, entrez le nom du rôle que vous venez de créer (codecatalyst-stack-role).
    - c. Choisissez le rôle dans la liste.
    - d. Dans la section Résumé, copiez la valeur ARN. Vous en aurez besoin ultérieurement.

Vous avez maintenant créé le rôle stack avec les autorisations appropriées, et vous avez obtenu son ARN.

### Étape 3 : ajouter AWS des rôles à CodeCatalyst

Au cours de cette étape, vous ajoutez le rôle de création (`codecatalyst-build-role`) et le rôle de déploiement (`codecatalyst-deploy-role`) à la connexion au CodeCatalyst compte dans votre espace.

#### Note

Il n'est pas nécessaire d'ajouter le rôle stack (`codecatalyst-stack-role`) à la connexion. Cela est dû au fait que le rôle de pile est utilisé CloudFormation(et non CodeCatalyst), une fois qu'une connexion est déjà établie entre CodeCatalyst et AWS en utilisant le rôle de déploiement. Comme le rôle stack n'est pas utilisé CodeCatalyst pour accéder à AWS, il n'est pas nécessaire de l'associer à une connexion à un compte.

Pour ajouter des rôles de création et de déploiement à votre compte, connectez-vous

1. Dans CodeCatalyst, naviguez jusqu'à votre espace.
2. Choisissez AWS des comptes. La liste des connexions au compte s'affiche.
3. Choisissez la connexion au compte qui représente le AWS compte sur lequel vous avez créé vos rôles de création et de déploiement.
4. Choisissez Gérer les rôles depuis AWS la console de gestion.

La page Ajouter un rôle IAM à Amazon CodeCatalyst Space s'affiche. Vous devrez peut-être vous connecter pour accéder à la page.

5. Sélectionnez Ajouter un rôle existant que vous avez créé dans IAM.

Une liste déroulante apparaît. La liste affiche tous les rôles IAM dotés d'une politique de confiance qui inclut les principaux de `codecatalyst.amazonaws.com` service `codecatalyst-runner.amazonaws.com` et.

6. Dans la liste déroulante, sélectionnez `codecatalyst-build-role`, puis choisissez Ajouter un rôle.

7. Choisissez Ajouter un rôle IAM, choisissez Ajouter un rôle existant que vous avez créé dans IAM, puis dans la liste déroulante, sélectionnez `codecatalyst-deploy-role` Choisissez Ajouter un rôle.

Vous avez maintenant ajouté les rôles de création et de déploiement à votre espace.

8. Copiez la valeur du nom CodeCatalyst d'affichage Amazon. Vous aurez besoin de cette valeur ultérieurement, lors de la création de votre flux de travail.

## Étape 4 : Création d'un compartiment Amazon S3

Au cours de cette étape, vous créez un compartiment Amazon S3 dans lequel vous stockez le fichier `.zip` du package de déploiement de votre application sans serveur.

Pour créer un compartiment Amazon S3

1. Ouvrez la console Amazon S3 sur <https://console.aws.amazon.com/s3/>.
2. Dans le volet principal, choisissez Create bucket.
3. Dans le champ Nom du compartiment, entrez :

```
codecatalyst-cfn-s3-bucket
```

4. Choisissez une région dans Région AWS . Ce didacticiel part du principe que vous avez choisi US West (Oregon) `us-west-2`. Pour plus d'informations sur les régions prises en charge par Amazon S3, consultez la section [Points de terminaison et quotas Amazon Simple Storage Service](#) dans le Références générales AWS.
5. Au bas de la page, choisissez Create bucket.

Vous venez de créer un bucket appelé **codecatalyst-cfn-s3-bucket** dans la région `us-west-2` de l'ouest des États-Unis (Oregon).

## Étape 5 : Ajouter des fichiers source

Au cours de cette étape, vous allez ajouter plusieurs fichiers source de l'application à votre référentiel CodeCatalyst source. Le `hello-world` dossier contient les fichiers d'application que vous allez déployer. Le `tests` dossier contient les tests unitaires. La structure des dossiers est la suivante :

```
.
|- hello-world
```

```
| |- tests
|   |- unit
|     |- test-handler.js
|   |- app.js
|- .npmignore
|- package.json
|- sam-template.yml
|- setup-sam.sh
```

## fichier .npmignore

Le `.npmignore` fichier indique les fichiers et dossiers que npm doit exclure du package de l'application. Dans ce didacticiel, npm exclut le `tests` dossier car il ne fait pas partie de l'application.

Pour ajouter le fichier `.npmignore`

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet, `codecatalyst-cfn-project`
3. Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources.
4. Dans la liste des référentiels sources, choisissez votre référentiel, `codecatalyst-cfn-source-repository`.
5. Dans Fichiers, choisissez Créer un fichier.
6. Dans le champ Nom du fichier, entrez :

```
.npmignore
```

7. Dans la zone de texte, entrez le code suivant :

```
tests/*
```

8. Choisissez Commit, puis sélectionnez Commit à nouveau.

Vous venez de créer un fichier appelé `.npmignore` à la racine de votre dépôt.

## fichier package.json

Le `package.json` fichier contient des métadonnées importantes concernant votre projet Node, telles que le nom du projet, le numéro de version, la description, les dépendances et d'autres détails qui décrivent comment interagir avec votre application et l'exécuter.

package.json Dans ce didacticiel, vous trouverez une liste de dépendances et un test script. Le script de test effectue les opérations suivantes :

- À l'aide de [mocha](#), le script de test exécute les tests unitaires spécifiés dans hello-world/tests/unit/ et écrit les résultats dans un junit.xml fichier à l'aide du rapporteur [xunit](#).
- À l'aide de [d'Istanbul \(NYC\)](#), le script de test génère un rapport de couverture de code (clover.xml) à l'aide du [Clover Reporter](#). Pour plus d'informations, consultez la section [Utilisation de reporters alternatifs](#) dans la documentation d'Istanbul.

Pour ajouter le fichier package.json

1. Dans votre référentiel, dans Fichiers, choisissez Créer un fichier.
2. Dans le champ Nom du fichier, entrez :

```
package.json
```

3. Dans la zone de texte, entrez le code suivant :

```
{
  "name": "hello_world",
  "version": "1.0.0",
  "description": "hello world sample for NodeJS",
  "main": "app.js",
  "repository": "https://github.com/aws-labs/aws-sam-cli/tree/develop/samcli/local/init/templates/cookiecutter-aws-sam-hello-nodejs",
  "author": "SAM CLI",
  "license": "MIT",
  "dependencies": {
    "axios": "^0.21.1",
    "nyc": "^15.1.0"
  },
  "scripts": {
    "test": "nyc --reporter=clover mocha hello-world/tests/unit/ --reporter xunit --reporter-option output=junit.xml"
  },
  "devDependencies": {
    "aws-sdk": "^2.815.0",
    "chai": "^4.2.0",
    "mocha": "^8.2.1"
  }
}
```

```
}
```

4. Choisissez Commit, puis sélectionnez Commit à nouveau.

Vous venez d'ajouter un fichier appelé `package.json` à la racine du dépôt.

fichier `sam-template.yml`

Le `sam-template.yml` fichier contient les instructions pour déployer la fonction Lambda et API Gateway et pour les configurer ensemble. Il suit la [spécification du AWS Serverless Application Model modèle](#), qui étend la spécification du AWS CloudFormation modèle.

Dans ce didacticiel, vous utilisez un AWS SAM modèle au lieu d'un AWS CloudFormation modèle normal car il AWS SAM propose un type de ressource utile [AWS::Serverless::Function](#). Ce type effectue une grande partie de la behind-the-scenes configuration que vous devez normalement écrire pour utiliser la CloudFormation syntaxe de base. Par exemple, `AWS::Serverless::Function` crée une fonction Lambda, un rôle d'exécution Lambda et des mappages de sources d'événements qui démarrent la fonction. Vous devez coder tout cela si vous voulez l'écrire en utilisant basic CloudFormation.

Bien que ce didacticiel utilise un modèle pré-écrit, vous pouvez en générer un dans le cadre de votre flux de travail à l'aide d'une action de génération. Pour plus d'informations, consultez [Ajout de l'action « Deploy AWS CloudFormation stack »](#).

Pour ajouter le fichier `sam-template.yml`

1. Dans votre référentiel, dans Fichiers, choisissez Créer un fichier.
2. Dans le champ Nom du fichier, entrez :

```
sam-template.yml
```

3. Dans la zone de texte, entrez le code suivant :

```
AWSTemplateFormatVersion: '2010-09-09'  
Transform: AWS::Serverless-2016-10-31  
Description: >  
  serverless-api  
  
Sample SAM Template for serverless-api
```

```
# More info about Globals: https://github.com/awslabs/serverless-application-model/
blob/master/docs/globals.rst
Globals:
  Function:
    Timeout: 3

Resources:
  HelloWorldFunction:
    Type: AWS::Serverless::Function # For details on this resource type,
    see https://github.com/awslabs/serverless-application-model/blob/master/
versions/2016-10-31.md#awsserverlessfunction
    Properties:
      CodeUri: hello-world/
      Handler: app.lambdaHandler
      Runtime: nodejs12.x
      Events:
        HelloWorld:
          Type: Api # For details on this event source type, see
          https://github.com/awslabs/serverless-application-model/blob/master/
versions/2016-10-31.md#api
          Properties:
            Path: /hello
            Method: get

Outputs:
  # ServerlessRestApi is an implicit API created out of the events key under
  Serverless::Function
  # Find out about other implicit resources you can reference within AWS SAM at
  # https://github.com/awslabs/serverless-application-model/blob/master/docs/
internals/generated_resources.rst#api
  HelloWorldApi:
    Description: "API Gateway endpoint URL for the Hello World function"
    Value: !Sub "https://${ServerlessRestApi}.execute-api.
${AWS::Region}.amazonaws.com/Prod/hello/"
  HelloWorldFunction:
    Description: "Hello World Lambda function ARN"
    Value: !GetAtt HelloWorldFunction.Arn
  HelloWorldFunctionIamRole:
    Description: "Implicit Lambda execution role created for the Hello World
function"
    Value: !GetAtt HelloWorldFunctionRole.Arn
```

#### 4. Choisissez Commit, puis sélectionnez Commit à nouveau.

Vous venez d'ajouter un fichier appelé `sam-template.yml` sous le dossier racine de votre dépôt.

### fichier `setup-sam.sh`

Le `setup-sam.sh` fichier contient les instructions de téléchargement et d'installation de l'utilitaire AWS SAM CLI. Le flux de travail utilise cet utilitaire pour empaqueter la `hello-world` source.

Pour ajouter le fichier `setup-sam.sh`

1. Dans votre référentiel, dans Fichiers, choisissez Créer un fichier.
2. Dans le champ Nom du fichier, entrez :

```
setup-sam.sh
```

3. Dans la zone de texte, entrez le code suivant :

```
#!/usr/bin/env bash
echo "Setting up sam"

yum install unzip -y

curl -LO https://github.com/aws/aws-sam-cli/releases/latest/download/aws-sam-cli-linux-x86_64.zip
unzip -qq aws-sam-cli-linux-x86_64.zip -d sam-installation-directory

./sam-installation-directory/install; export AWS_DEFAULT_REGION=us-west-2
```

Dans le code précédent, remplacez *us-west-2* par votre région. AWS

4. Choisissez Commit, puis sélectionnez Commit à nouveau.

Vous venez d'ajouter un fichier appelé `setup-sam.sh` à la racine du dépôt.

### fichier `app.js`

`app.js` Contient le code de la fonction Lambda. Dans ce didacticiel, le code renvoie le texte `hello world`.



## Pour ajouter le fichier app.js

1. Dans votre référentiel, dans Fichiers, choisissez Créer un fichier.
2. Dans le champ Nom du fichier, entrez :

```
hello-world/app.js
```

3. Dans la zone de texte, entrez le code suivant :

```
// const axios = require('axios')
// const url = 'http://checkip.amazonaws.com/';
let response;

/**
 *
 * Event doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html#api-gateway-simple-proxy-for-lambda-input-format
 * @param {Object} event - API Gateway Lambda Proxy Input Format
 *
 * Context doc: https://docs.aws.amazon.com/lambda/latest/dg/nodejs-prog-model-context.html
 * @param {Object} context
 *
 * Return doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html
 * @returns {Object} object - API Gateway Lambda Proxy Output Format
 */
exports.lambdaHandler = async (event, context) => {
  try {
    // const ret = await axios(url);
    response = {
      'statusCode': 200,
      'body': JSON.stringify({
        message: 'hello world',
        // location: ret.data.trim()
      })
    }
  } catch (err) {
    console.log(err);
    return err;
  }
}
```

```
    return response
  };
```

4. Choisissez Commit, puis sélectionnez Commit à nouveau.

Vous avez maintenant créé un dossier appelé `hello-world` et un fichier appelé `app.js`.

fichier `test-handler.js`

Le `test-handler.js` fichier contient des tests unitaires pour la fonction Lambda.

Pour ajouter le fichier `test-handler.js`

1. Dans votre référentiel, dans Fichiers, choisissez Créer un fichier.
2. Dans le champ Nom du fichier, entrez :

```
hello-world/tests/unit/test-handler.js
```

3. Dans la zone de texte, entrez le code suivant :

```
'use strict';

const app = require('../../app.js');
const chai = require('chai');
const expect = chai.expect;
var event, context;

describe('Tests index', function () {
  it('verifies successful response', async () => {
    const result = await app.lambdaHandler(event, context)

    expect(result).to.be.an('object');
    expect(result.statusCode).to.equal(200);
    expect(result.body).to.be.an('string');

    let response = JSON.parse(result.body);

    expect(response).to.be.an('object');
    expect(response.message).to.be.equal("hello world");
    // expect(response.location).to.be.an("string");
  });
});
```

#### 4. Choisissez Commit, puis sélectionnez Commit à nouveau.

Vous venez d'ajouter un fichier appelé `test-handler.js` sous le `hello-world/tests/unit` dossier.

Vous avez maintenant ajouté tous vos fichiers sources.

Prenez le temps de vérifier votre travail et de vous assurer que vous avez placé tous les fichiers dans les bons dossiers. La structure des dossiers est la suivante :

```
.
|- hello-world
|  |- tests
|     |- unit
|        |- test-handler.js
|  |- app.js
|- .npmignore
|- README.md
|- package.json
|- sam-template.yml
|- setup-sam.sh
```

### Étape 6 : créer et exécuter un flux de travail

Au cours de cette étape, vous créez un flux de travail qui empaquette votre code source Lambda et le déploie. Le flux de travail comprend les éléments de base suivants qui s'exécutent de manière séquentielle :

- Un déclencheur : ce déclencheur lance automatiquement l'exécution du flux de travail lorsque vous apportez une modification à votre référentiel source. Pour plus d'informations sur les déclencheurs, consultez [Utilisation de déclencheurs](#).
- Une action de test (Test) — Au déclenchement, cette action installe le [gestionnaire de packages Node \(npm\)](#), puis exécute la `npm run test` commande. Cette commande indique à npm d'exécuter le test script défini dans le `package.json` fichier. Le test script exécute à son tour les tests unitaires et génère deux rapports : un rapport de test ( `junit.xml` ) et un rapport de couverture de code ( `clover.xml` ). Pour plus d'informations, consultez [fichier package.json](#).

Ensuite, l'action de test transforme les rapports XML en CodeCatalyst rapports et les affiche dans la CodeCatalyst console, sous l'onglet Rapports de l'action de test.

Pour plus d'informations sur l'action de test, consultez [Tests à l'aide de workflows dans CodeCatalyst](#).

- Une action de construction (BuildBackend) : à la fin de l'action de test, l'action de génération télécharge et installe la AWS SAM CLI, empaquète la hello-world source et copie le package dans votre compartiment Amazon S3, là où le service Lambda s'attend à ce qu'il se trouve. L'action génère également un nouveau fichier AWS SAM modèle appelé `sam-template-packaged.yml` et le place dans un artefact de sortie appelé `buildArtifact`.

Pour plus d'informations sur l'action de génération, consultez [Création à l'aide de flux de travail dans CodeCatalyst](#).

- Une action de déploiement (DeployCloudFormationStack) : à la fin de l'action de génération, l'action de déploiement recherche l'artefact de sortie généré par l'action de génération (`buildArtifact`), trouve le AWS SAM modèle à l'intérieur de celui-ci, puis exécute le modèle. Le AWS SAM modèle crée une pile qui déploie l'application sans serveur.

Pour créer un flux de travail

1. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
2. Choisissez Créer un flux de travail.
3. Pour Référentiel source, choisissez `codecatalyst-cfn-source-repository`.
4. Pour Branch, choisissez `main`.
5. Choisissez Créer.
6. Supprimez l'exemple de code YAML.
7. Ajoutez le code YAML suivant :

```
Name: codecatalyst-cfn-workflow
SchemaVersion: 1.0

Triggers:
  - Type: PUSH
    Branches:
      - main

Actions:
  Test:
    Identifiant: aws/managed-test@v1
    Inputs:
```

```
Sources:
  - WorkflowSource
Outputs:
Reports:
  CoverageReport:
    Format: CLOVERXML
    IncludePaths:
      - "coverage/*"
  TestReport:
    Format: JUNITXML
    IncludePaths:
      - junit.xml
Configuration:
  Steps:
    - Run: npm install
    - Run: npm run test
BuildBackend:
  Identifier: aws/build@v1
  DependsOn:
    - Test
  Environment:
    Name: codecatalyst-cfn-environment
  Connections:
    - Name: codecatalyst-account-connection
      Role: codecatalyst-build-role
Inputs:
  Sources:
    - WorkflowSource
  Configuration:
    Steps:
      - Run: . ./setup-sam.sh
      - Run: sam package --template-file sam-template.yml --s3-
bucket codecatalyst-cfn-s3-bucket --output-template-file sam-template-packaged.yml
--region us-west-2
    Outputs:
      Artifacts:
        - Name: buildArtifact
          Files:
            - "**/*"
  DeployCloudFormationStack:
    Identifier: aws/cfn-deploy@v1
    DependsOn:
      - BuildBackend
  Environment:
```

```
Name: codecatalyst-cfn-environment
Connections:
  - Name: codecatalyst-account-connection
    Role: codecatalyst-deploy-role
Inputs:
  Artifacts:
    - buildArtifact
  Sources: []
Configuration:
  name: codecatalyst-cfn-stack
  region: us-west-2
  role-arn: arn:aws:iam::111122223333:role/StackRole
  template: ./sam-template-packaged.yml
  capabilities: CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND
```

Dans le code précédent, remplacez :

- Les deux instances *codecatalyst-cfn-environment* portent le nom de votre environnement.
- Dans les deux cas, *codecatalyst-account-connection* avec le nom d'affichage de la connexion à votre compte. Le nom d'affichage peut être un chiffre. Pour plus d'informations, consultez [Étape 3 : ajouter AWS des rôles à CodeCatalyst](#).
- *codecatalyst-build-role* avec le nom du rôle de build que vous avez créé dans [Étape 2 : créer des AWS rôles](#).
- *codecatalyst-cfn-s3 compartiments avec le nom du compartiment* Amazon S3 dans [Étape 4 : Création d'un compartiment Amazon S3](#) lequel vous l'avez créé.
- Les deux instances de *us-west-2* avec la région où réside votre compartiment Amazon S3 (première instance) et où votre stack sera déployée (deuxième instance). Ces régions peuvent être différentes. Ce didacticiel part du principe que les deux régions sont définies sur *us-west-2*. Pour en savoir plus sur les régions prises en charge par Amazon S3 AWS CloudFormation, consultez la section [Points de terminaison et quotas du service](#) dans le Références générales AWS.
- *codecatalyst-deploy-role* avec le nom du rôle de déploiement que vous avez créé dans [Étape 2 : créer des AWS rôles](#).
- *codecatalyst-cfn-environment* avec le nom de l'environnement dans lequel vous l'avez créé [Prérequis](#).
- *arn:aws:iam : :111122223333:role/ avec* le nom de ressource Amazon (ARN) StackRole du rôle de pile que vous avez créé dans. [Étape 2 : créer des AWS rôles](#)

**Note**

Si vous avez décidé de ne pas créer de rôles de construction, de déploiement et d'empilement, remplacez `codecatalyst-build-role``codecatalyst-deploy-role`, et `arn:aws:iam : StackRole :111122223333:role/` par le nom ou l'ARN du rôle. `CodeCatalystWorkflowDevelopmentRole-spaceName` Pour plus d'informations sur ce rôle, consultez [Étape 2 : créer des AWS rôles](#).

Pour plus d'informations sur les propriétés du code présenté précédemment, consultez le [Référence d'action « Deploy AWS CloudFormation Stack »](#).

8. (Facultatif) Choisissez Valider pour vous assurer que le code YAML est valide avant de valider.
9. Choisissez Commit (Valider).
10. Dans la boîte de dialogue du flux de travail de validation, entrez ce qui suit :
  - a. Pour le nom du fichier de flux de travail, conservez la valeur par défaut, `codecatalyst-cfn-workflow`.
  - b. Pour le message de validation, entrez :

```
add initial workflow file
```

- c. Pour Repository, choisissez `codecatalyst-cfn-source-repository`.
- d. Pour Nom de la branche, choisissez `main`.
- e. Choisissez Commit (Valider).

Vous venez de créer un flux de travail. L'exécution d'un flux de travail démarre automatiquement en raison du déclencheur défini en haut du flux de travail. Plus précisément, lorsque vous avez validé (et transféré) le `codecatalyst-cfn-workflow.yaml` fichier dans votre référentiel source, le déclencheur a lancé l'exécution du flux de travail.

Pour consulter le flux de travail en cours d'exécution

1. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
2. Choisissez le flux de travail que vous venez de créer : `codecatalyst-cfn-workflow`.
3. Choisissez l'onglet Runs.

4. Dans la colonne Run ID, choisissez l'Run ID.
5. Choisissez Test pour voir la progression des tests.
6. Choisissez BuildBackendde voir la progression de la construction.
7. Choisissez DeployCloudFormationStackde voir la progression du déploiement.

Pour plus d'informations sur l'affichage des détails des exécutions, consultez [Afficher le statut et les détails de l'exécution du flux de travail](#).

8. Lorsque l'DeployCloudFormationStackaction est terminée, procédez comme suit :
  - Si l'exécution du flux de travail a réussi, passez à la procédure suivante.
  - Si l'exécution du flux de travail a échoué lors du test ou de BuildBackendl'action, choisissez Logs pour résoudre le problème.
  - Si l'exécution du flux de travail a échoué lors de l'DeployCloudFormationStackaction, choisissez l'action de déploiement, puis cliquez sur l'onglet Résumé. Accédez à la section CloudFormation des événements pour afficher le message d'erreur détaillé. En cas de restauration, supprimez la codecatalyst-cfn-stack pile via la AWS CloudFormation console AWS avant de réexécuter le flux de travail.

#### Pour vérifier le déploiement

1. Une fois le déploiement réussi, choisissez Variables (7) dans la barre de menu horizontale située en haut. (Ne choisissez pas Variables dans le volet de droite.)
2. Ensuite HelloWorldApi, collez l'`https://URL` dans un navigateur.

Un message JSON « hello world » de la fonction Lambda s'affiche, indiquant que le flux de travail a déployé et configuré correctement la fonction Lambda et l'API Gateway.

#### Tip

Vous pouvez avoir CodeCatalyst affiché cette URL dans le diagramme de flux de travail avec quelques petites configurations. Pour plus d'informations, consultez [Afficher l'URL de l'application déployée](#).

#### Pour vérifier les résultats des tests unitaires et la couverture du code

1. Dans le diagramme du flux de travail, choisissez Test, puis Reports.



2. Choisissez `TestReport` d'afficher les résultats des tests unitaires ou `CoverageReport` d'afficher les détails de couverture du code des fichiers testés, dans ce cas, `app.js` et `test-handler.js`.

Pour vérifier les ressources déployées

1. Connectez-vous à la console API Gateway AWS Management Console et ouvrez-la à l'[adresse https://console.aws.amazon.com/apigateway/](https://console.aws.amazon.com/apigateway/).
2. Observez l'`codecatalyst-cfn-stack` API créée par le AWS SAM modèle. Le nom de l'API provient de la `Configuration/name` valeur du fichier de définition du flux de travail (`codecatalyst-cfn-workflow.yaml`).
3. Ouvrez la AWS Lambda console à l'[adresse https://console.aws.amazon.com/lambda/](https://console.aws.amazon.com/lambda/).
4. Dans le volet de navigation, choisissez Fonctions.
5. Choisissez votre fonction Lambda, `codecatalyst-cfn-stack-HelloWorldFunction-string`
6. Vous pouvez voir comment l'API Gateway déclenche cette fonction. Cette intégration a été automatiquement configurée en fonction du type de AWS SAM `AWS::Serverless::Function` ressource.

## Étape 7 : Apporter une modification

Au cours de cette étape, vous apportez une modification à votre code source Lambda et vous le validez. Ce commit lance une nouvelle exécution du flux de travail. Cette exécution déploie la nouvelle fonction Lambda dans un schéma bleu-vert qui utilise la configuration de transfert de trafic par défaut spécifiée dans la console Lambda.

Pour apporter une modification à votre source Lambda

1. Dans CodeCatalyst, accédez à votre projet.
2. Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources.
3. Choisissez votre référentiel source `codecatalyst-cfn-source-repository`.
4. Modifiez le dossier de candidature :
  - a. Choisissez le dossier `hello-world`.
  - b. Choisissez le `app.js` fichier.
  - c. Choisissez Modifier.

- d. À la ligne 23, changez `hello world` en **Tutorial complete!**.
- e. Choisissez Commit, puis sélectionnez Commit à nouveau.

La validation entraîne le démarrage d'un flux de travail. Cette exécution échouera car vous n'avez pas mis à jour les tests unitaires pour refléter le changement de nom.

5. Mettez à jour les tests unitaires :
  - a. Sélectionnez `hello-world\tests\unit\test-handler.js`.
  - b. Choisissez Modifier.
  - c. À la ligne 19, changez `hello world` en **Tutorial complete!**.
  - d. Choisissez Commit, puis sélectionnez Commit à nouveau.

La validation entraîne le démarrage d'un autre flux de travail. Cette course sera couronnée de succès.

6. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
7. Choisissez `codecatalyst-cfn-workflow`, puis choisissez Runs.
8. Choisissez l'ID d'exécution de la dernière exécution. Il devrait toujours être en cours.
9. Choisissez Test BuildBackend, et DeployCloudFormationStack pour voir la progression de l'exécution du flux de travail.
10. Lorsque le flux de travail est terminé, choisissez Variables (7) en haut.
11. Ensuite HelloWorldApi, collez l'`https://URL` dans un navigateur.

Un `Tutorial complete!` message s'affiche dans le navigateur, indiquant que votre nouvelle application a été déployée avec succès.

## Nettoyage

Nettoyez les fichiers et les services utilisés dans ce didacticiel pour éviter de vous les faire facturer.

Pour nettoyer dans la CodeCatalyst console

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Supprimer `codecatalyst-cfn-workflow`.
3. Supprimer `codecatalyst-cfn-environment`.
4. Supprimer `codecatalyst-cfn-source-repository`.

## 5. Supprimer `codecatalyst-cfn-project`.

Pour nettoyer dans le AWS Management Console

### 1. Nettoyez en CloudFormation procédant comme suit :

- a. Ouvrez la AWS CloudFormation console à l'[adresse https://console.aws.amazon.com/cloudformation](https://console.aws.amazon.com/cloudformation).
- b. Supprimez la `codecatalyst-cfn-stack`.

La suppression de la pile supprime toutes les ressources du didacticiel des services API Gateway et Lambda.

### 2. Nettoyez dans Amazon S3, comme suit :

- a. Ouvrez la console Amazon S3 sur <https://console.aws.amazon.com/s3/>.
- b. Cliquez sur le bouton `codecatalyst-cfn-s3-bucket`.
- c. Supprimez le contenu du compartiment.
- d. Supprimez le compartiment.

### 3. Nettoyez dans IAM, comme suit :

- a. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
- b. Supprimez la `codecatalyst-deploy-policy`.
- c. Supprimez la `codecatalyst-build-policy`.
- d. Supprimez la `codecatalyst-stack-policy`.
- e. Supprimez la `codecatalyst-deploy-role`.
- f. Supprimez la `codecatalyst-build-role`.
- g. Supprimez la `codecatalyst-stack-role`.

Dans ce didacticiel, vous avez appris à déployer une application sans serveur sous forme de CloudFormation pile à l'aide d'un CodeCatalyst flux de travail et d'une action de déploiement de AWS CloudFormation pile.

## Tutoriel : Déploiement d'une application sur Amazon ECS

Dans ce didacticiel, vous allez apprendre à déployer une application sans serveur dans Amazon Elastic Container Service (Amazon ECS) à l'aide d'un flux de travail, Amazon ECS, et de quelques

autres services. AWS L'application déployée est un simple site Web Hello World construit sur une image Docker du serveur Web Apache. Le didacticiel explique les tâches de préparation requises, telles que la configuration d'un cluster, puis décrit comment créer un flux de travail pour créer et déployer l'application.

### Tip

Au lieu de suivre ce didacticiel, vous pouvez utiliser un plan qui effectue une configuration complète d'Amazon ECS pour vous. Vous devrez utiliser l'API Node.js avec AWS Fargate ou l'API Java avec AWS Fargate Blueprint. Pour plus d'informations, consultez [Création d'un projet à l'aide d'un plan](#).

## Rubriques

- [Prérequis](#)
- [Étape 1 : configurer un AWS utilisateur et AWS CloudShell](#)
- [Étape 2 : Déployer une application fictive dans Amazon ECS](#)
- [Étape 3 : Création d'un référentiel d'images Amazon ECR](#)
- [Étape 4 : Création de AWS rôles](#)
- [Étape 5 : Ajouter AWS des rôles à CodeCatalyst](#)
- [Étape 6 : Création d'un référentiel source](#)
- [Étape 7 : Ajouter des fichiers source](#)
- [Étape 8 : créer et exécuter un flux de travail](#)
- [Étape 9 : Apportez une modification à vos fichiers source](#)
- [Nettoyage](#)

## Prérequis

Avant de commencer :

- Vous avez besoin d'un CodeCatalyst espace avec un AWS compte connecté. Pour plus d'informations, consultez [Création d'un espace prenant en charge les utilisateurs de AWS Builder ID](#).
- Dans votre espace, vous avez besoin d'un CodeCatalyst projet vide, créé à partir de zéro, appelé :

```
codecatalyst-ecs-project
```

Pour plus d'informations, consultez [Création d'un projet vide dans Amazon CodeCatalyst](#).

- Dans votre projet, vous avez besoin d'un CodeCatalyst environnement appelé :

```
codecatalyst-ecs-environment
```

Configurez cet environnement comme suit :

- Choisissez n'importe quel type, tel que Non-production.
- AWS Connectez-y votre compte.

Pour plus d'informations, consultez [Utilisation des environnements](#).

## Étape 1 : configurer un AWS utilisateur et AWS CloudShell

La première étape de ce didacticiel consiste à créer un utilisateur dans AWS IAM Identity Center et à lancer une AWS CloudShell instance sous le nom de cet utilisateur. Pendant la durée de ce didacticiel, CloudShell c'est votre ordinateur de développement et c'est là que vous configurez les AWS ressources et les services. Supprimez cet utilisateur après avoir terminé le didacticiel.

### Note

N'utilisez pas votre utilisateur root pour ce didacticiel. Vous devez créer un utilisateur distinct, sinon vous risquez de rencontrer des problèmes lors de l'exécution d'actions dans la AWS Command Line Interface (CLI) ultérieurement.

Pour plus d'informations sur les utilisateurs d'IAM Identity Center CloudShell, consultez le guide de l'AWS IAM Identity Center utilisateur et le guide de AWS CloudShell l'utilisateur.

Pour créer un utilisateur IAM Identity Center

1. Connectez-vous à la AWS IAM Identity Center console AWS Management Console et ouvrez-la à l'[adresse https://console.aws.amazon.com/singlesignon/](https://console.aws.amazon.com/singlesignon/).

**Note**

Assurez-vous de vous connecter à l'aide de Compte AWS celui qui est connecté à votre CodeCatalyst espace. Vous pouvez vérifier quel compte est connecté en accédant à votre espace et en choisissant l'onglet Comptes AWS. Pour plus d'informations, consultez [Création d'un espace prenant en charge les utilisateurs de AWS Builder ID](#).

2. Dans le panneau de navigation, choisissez Users (Utilisateurs), puis Add user (Ajouter un utilisateur).

3. Dans Nom d'utilisateur, entrez :

**CodeCatalystECSUser**

4. Sous Mot de passe, choisissez Générer un mot de passe à usage unique que vous pouvez partager avec cet utilisateur.

5. Dans Adresse e-mail et Confirmer l'adresse e-mail, entrez une adresse e-mail qui n'existe pas encore dans IAM Identity Center.

6. Dans Prénom et nom de famille, entrez :

**CodeCatalystECSUser**

7. Dans Nom d'affichage, conservez le nom généré automatiquement :

**CodeCatalystECSUser CodeCatalystECSUser**

8. Choisissez Suivant.

9. Sur la page Ajouter un utilisateur aux groupes, choisissez Next.

10. Sur la page Vérifier et ajouter un utilisateur, passez en revue les informations et choisissez Ajouter un utilisateur.

Une boîte de dialogue de mot de passe à usage unique s'affiche.

11. Choisissez Copier, puis collez les informations de connexion, y compris l'URL du portail AWS d'accès et le mot de passe à usage unique.

12. Choisissez Fermer.

## Pour créer un jeu d'autorisations

Vous attribuerez cet ensemble d'autorisations à une `CodeCatalystECSUser` date ultérieure.

1. Dans le volet de navigation, choisissez Ensembles d'autorisations, puis choisissez Créer un ensemble d'autorisations.
2. Choisissez Ensemble d'autorisations prédéfini, puis sélectionnez `AdministratorAccess`. Cette politique fournit des autorisations complètes à tous Services AWS.
3. Choisissez Suivant.
4. Dans Nom du jeu d'autorisations, entrez :

`CodeCatalystECSPermissionSet`

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

## Pour attribuer l'ensemble d'autorisations à `CodeCatalyst ECSUser`

1. Dans le volet de navigation, choisissez Comptes AWS, puis cochez la case à côté de Compte AWS celle à laquelle vous êtes actuellement connecté.
2. Choisissez Attribuer des utilisateurs ou des groupes.
3. Sélectionnez l'onglet Utilisateurs.
4. Cochez la case à côté de `CodeCatalystECSUser`.
5. Choisissez Suivant.
6. Cochez la case à côté de `CodeCatalystECSPermissionSet`.
7. Choisissez Suivant.
8. Vérifiez les informations et choisissez Soumettre.

Vous les avez maintenant assignés `CodeCatalystECSUser` et `CodeCatalystECSPermissionSet` à votre Compte AWS, en les liant ensemble.

## Pour vous déconnecter et vous reconnecter en tant qu' `CodeCatalystECSUser`

1. Avant de vous déconnecter, assurez-vous d'avoir l'URL du portail AWS d'accès ainsi que le nom d'utilisateur et le mot de passe à usage unique pour `CodeCatalystECSUser`. Vous devriez avoir copié ces informations dans un éditeur de texte plus tôt.

**Note**

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

2. Déconnectez-vous de AWS.
3. Collez l'URL du portail d' AWS accès dans la barre d'adresse de votre navigateur.
4. Connectez-vous avec le nom d'utilisateur et le mot de passe à usage unique pourCodeCatalystECSUser.
5. Dans Nouveau mot de passe, entrez un mot de passe, puis choisissez Définir un nouveau mot de passe.

Une Compte AWSboîte apparaît à l'écran.

6. Choisissez Compte AWS, puis choisissez le nom de l'utilisateur Compte AWS auquel vous avez attribué l'CodeCatalystECSUserutilisateur et le jeu d'autorisations.
7. À côté deCodeCatalystECSPermissionSet, choisissez Console de gestion.

Le AWS Management Console apparaît. Vous êtes maintenant connecté CodeCatalystECSUser avec les autorisations appropriées.

Pour lancer une AWS CloudShell instance

1. Par CodeCatalystECSUser exemple, dans la barre de navigation supérieure, choisissez l' AWS icône



La page principale du AWS Management Console apparaît.

2. Dans la barre de navigation supérieure, choisissez l' AWS CloudShell icône



CloudShell ouvre. Patientez pendant que l' CloudShell environnement est créé.



**Note**

Si l'icône CloudShell ne s'affiche pas, assurez-vous que vous vous trouvez dans une [région prise en charge par CloudShell](#). Ce didacticiel part du principe que vous vous trouvez dans la région de l'ouest des États-Unis (Oregon).

Pour vérifier AWS CLI que le

1. Dans le CloudShell terminal, saisissez :

```
aws --version
```

2. Vérifiez qu'une version apparaît.

Il AWS CLI est déjà configuré pour l'utilisateur actuel `CodeCatalystECSUser`, il n'est donc pas nécessaire de configurer AWS CLI les clés et les informations d'identification, comme c'est normalement le cas.

## Étape 2 : Déployer une application fictive dans Amazon ECS

Dans cette section, vous déployez manuellement une application fictive dans Amazon ECS. Cette application fictive sera remplacée par l'application Hello World déployée par votre flux de travail. L'application fictive est le serveur Web Apache.

Pour plus d'informations sur Amazon ECS, consultez le guide du développeur Amazon Elastic Container Service.

Suivez la série de procédures ci-dessous pour déployer l'application fictive.

Pour créer le rôle d'exécution des tâches

Ce rôle accorde à Amazon ECS l'AWS Fargate (Fargate) autorisation d'effectuer des appels d'API en votre nom.

1. Créez une politique de confiance :
  - a. Dans AWS CloudShell, entrez la commande suivante :

```
cat > codecatalyst-ecs-trust-policy.json
```

Un message clignotant apparaît dans le CloudShell terminal.

- b. Entrez le code suivant à l'invite :

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

- c. Placez votre curseur après le dernier crochet (}).
- d. Appuyez sur **Enter** puis **Ctrl+d** pour enregistrer le fichier et quitter cat.

2. Créez un rôle d'exécution de tâches :

```
aws iam create-role \
  --role-name codecatalyst-ecs-task-execution-role \
  --assume-role-policy-document file:///codecatalyst-ecs-trust-policy.json
```

3. Associez la AmazonECSTaskExecutionRolePolicy politique AWS gérée au rôle :

```
aws iam attach-role-policy \
  --role-name codecatalyst-ecs-task-execution-role \
  --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy
```

4. Afficher les détails du rôle :

```
aws iam get-role \
  --role-name codecatalyst-ecs-task-execution-role
```

5. Notez la "Arn" : valeur du rôle, par exemple `arn:aws:iam::111122223333:role/codecatalyst-ecs-task-execution-role`. Vous aurez besoin de cet Amazon Resource Name (ARN) ultérieurement.

## Pour créer un nouveau cluster Amazon ECS

Ce cluster contiendra l'application d'espace réservé Apache, puis l'application Hello World.

1. Comme dans CodeCatalystECSUser AWS CloudShell, créez un cluster vide :

```
aws ecs create-cluster --cluster-name codecatalyst-ecs-cluster
```

2. (Facultatif) Vérifiez que le cluster a été créé avec succès :

```
aws ecs list-clusters
```

L'ARN du `codecatalyst-ecs-cluster` cluster doit apparaître dans la liste, indiquant une création réussie.

## Pour créer un fichier de définition de tâche

Le fichier de définition de tâche indique d'exécuter l'image Docker (`httpd:2.4`) [du serveur Web Apache 2.4](#) dont elle est DockerHub extraite.

1. Comme dans CodeCatalystECSUser AWS CloudShell, créez un fichier de définition de tâche :

```
cat > taskdef.json
```

2. Collez le code suivant à l'invite :

```
{
  "executionRoleArn": "arn:aws:iam::111122223333:role/codecatalyst-ecs-task-
execution-role",
  "containerDefinitions": [
    {
      "name": "codecatalyst-ecs-container",
      "image": "httpd:2.4",
      "essential": true,
      "portMappings": [
        {
```

```
        "hostPort": 80,  
        "protocol": "tcp",  
        "containerPort": 80  
      }  
    ]  
  },  
  "requiresCompatibilities": [  
    "FARGATE"  
  ],  
  "cpu": "256",  
  "family": "codecatalyst-ecs-task-def",  
  "memory": "512",  
  "networkMode": "awsvpc"  
}
```

Dans le code précédent, remplacez `arn:aws:iam : :111122223333:role/` -role codecatalyst-ecs-task-execution

avec l'ARN du rôle d'exécution de tâche que vous avez noté dans [Pour créer le rôle d'exécution des tâches](#).

3. Placez votre curseur après le dernier crochet (}).
4. Appuyez sur **Enter** puis **Ctrl+d** pour enregistrer le fichier et quitter cat.

Pour enregistrer le fichier de définition de tâche auprès d'Amazon ECS

1. Comme dans CodeCatalystECSUser AWS CloudShell, enregistrez la définition de tâche :

```
aws ecs register-task-definition \  
  --cli-input-json file://taskdef.json
```

2. (Facultatif) Vérifiez que la définition de tâche a été enregistrée :

```
aws ecs list-task-definitions
```

La définition de la codecatalyst-ecs-task-def tâche doit apparaître dans la liste.

## Pour créer le service Amazon ECS

Le service Amazon ECS exécute les tâches (et les conteneurs Docker associés) de l'application fictive Apache, puis de l'application Hello World.

1. Par `CodeCatalystECSUser` exemple, passez à la console Amazon Elastic Container Service si ce n'est pas déjà fait.
2. Choisissez le cluster que vous avez créé précédemment, `codecatalyst-ecs-cluster`.
3. Dans l'onglet Services, choisissez Create.
4. Dans la page Créer, procédez comme suit :
  - a. Conservez tous les paramètres par défaut, à l'exception de ceux listés ci-dessous.
  - b. Pour Launch type (Type de lancement), choisissez FARGATE.
  - c. Sous Définition de la tâche, dans la liste déroulante Famille, sélectionnez :

`codecatalyst-ecs-task-def`

- d. Pour le nom du service, entrez :

`codecatalyst-ecs-service`

- e. Dans Tâches souhaitées, entrez :

`3`

Dans ce didacticiel, chaque tâche lance un conteneur Docker unique.

- f. Développez la section Mise en réseau.
- g. Pour VPC, choisissez n'importe quel VPC.
- h. Pour les sous-réseaux, choisissez n'importe quel sous-réseau.

### Note

Spécifiez un seul sous-réseau. C'est tout ce dont vous avez besoin pour ce tutoriel.

**Note**

Si vous n'avez pas de VPC ni de sous-réseau, créez-les. Consultez les [sections Créer un VPC](#) et [Créer un sous-réseau dans votre VPC dans le guide de l'utilisateur Amazon VPC](#).

- i. Pour Groupe de sécurité, choisissez Créer un nouveau groupe de sécurité, puis procédez comme suit :
    - i. Pour le nom du groupe de sécurité, entrez :

`codecatalyst-ecs-security-group`
    - ii. Pour la description du groupe de sécurité, entrez :

`CodeCatalyst ECS security group`
    - iii. Choisissez Ajouter une règle. Pour Type, choisissez HTTP, et pour Source, choisissez Anywhere.
  - j. En bas de la page, choisissez Create.
  - k. Patientez pendant la création du service. Cette opération peut prendre quelques minutes.
5. Cliquez sur l'onglet Tâches, puis sur le bouton d'actualisation. Vérifiez que la colonne Dernier état des trois tâches est définie sur En cours d'exécution.

(Facultatif) Pour vérifier que votre application d'espace réservé Apache est en cours d'exécution

1. Dans l'onglet Tâches, choisissez l'une des trois tâches.
2. Dans le champ IP publique, choisissez une adresse ouverte.

Une `It Works!` page apparaît. Cela indique que le service Amazon ECS a démarré avec succès une tâche qui a lancé un conteneur Docker avec l'image Apache.

À ce stade du didacticiel, vous avez déployé manuellement une définition de cluster, de service et de tâche Amazon ECS, ainsi qu'une application d'espace réservé Apache. Tous ces éléments étant en place, vous êtes maintenant prêt à créer un flux de travail qui remplacera l'application fictive Apache par l'application Hello World du didacticiel.

## Étape 3 : Création d'un référentiel d'images Amazon ECR

Dans cette section, vous allez créer un référentiel d'images privé dans Amazon Elastic Container Registry (Amazon ECR). Ce dépôt stocke l'image Docker du didacticiel qui remplacera l'image fictive d'Apache que vous avez déployée précédemment.

Pour plus d'informations sur Amazon ECR, consultez le guide de l'utilisateur d'Amazon Elastic Container Registry.

Pour créer un référentiel d'images dans Amazon ECR

1. Par `CodeCatalystECSUser` exemple, dans AWS CloudShell, créez un référentiel vide dans Amazon ECR :

```
aws ecr create-repository --repository-name codecatalyst-ecs-image-repo
```

2. Affichez les détails du référentiel Amazon ECR :

```
aws ecr describe-repositories \  
  --repository-names codecatalyst-ecs-image-repo
```

3. Notez la `repositoryUri` : valeur, par exemple, `111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo`.

Vous en aurez besoin ultérieurement lors de l'ajout du référentiel à votre flux de travail.

## Étape 4 : Création de AWS rôles

Dans cette section, vous allez créer les rôles AWS IAM dont votre CodeCatalyst flux de travail aura besoin pour fonctionner. Ces rôles sont les suivants :

- Rôle de création : accorde à l'action de CodeCatalyst création (dans le flux de travail) l'autorisation d'accéder à votre AWS compte et d'écrire sur Amazon ECR et Amazon EC2.
- Rôle de déploiement : accorde à l'action CodeCatalyst Deploy to ECS (dans le flux de travail) l'autorisation d'accéder à votre AWS compte, à Amazon ECS et à quelques autres AWS services.

Pour plus d'informations sur les rôles IAM, consultez la section Rôles [IAM dans le Guide](#) de l'AWS Identity and Access Management utilisateur.

**Note**

Pour gagner du temps, vous pouvez créer un seul rôle, appelé `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle, au lieu des deux rôles répertoriés précédemment. Pour plus d'informations, consultez [Création du CodeCatalystWorkflowDevelopmentRole-\*spaceName\* rôle pour votre compte et votre espace](#). Sachez que le `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle dispose d'autorisations très étendues, ce qui peut présenter un risque de sécurité. Nous vous recommandons de n'utiliser ce rôle que dans les didacticiels et les scénarios où la sécurité est moins préoccupante. Ce didacticiel part du principe que vous créez les deux rôles répertoriés précédemment.

Pour créer les rôles de génération et de déploiement, vous pouvez utiliser le AWS Management Console ou le AWS CLI.

### AWS Management Console

Pour créer les rôles de génération et de déploiement, suivez la série de procédures suivante.

Pour créer un rôle de build

1. Créez une politique pour le rôle, comme suit :
  - a. Connectez-vous à AWS.
  - b. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
  - c. Dans le panneau de navigation, choisissez Politiques (Politiques).
  - d. Sélectionnez Create policy (Créer une politique).
  - e. Choisissez l'onglet JSON.
  - f. Supprimez le code existant.
  - g. Collez le code suivant :


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



```

        "ec1:*",
        "ec2:*"
    ],
    "Resource": "*"
}
]
}

```

 Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"

```

- h. Choisissez Suivant : Balises.
- i. Choisissez Suivant : Vérification.
- j. Dans Nom, entrez :

```
codecatalyst-ecs-build-policy

```

- k. Choisissez Créer une politique.

Vous venez de créer une politique d'autorisation.

2. Créez le rôle de build, comme suit :
  - a. Dans le volet de navigation, sélectionnez Rôles, puis Créer un rôle.
  - b. Choisissez Politique de confiance personnalisée.
  - c. Supprimez la politique de confiance personnalisée existante.
  - d. Ajoutez la politique de confiance personnalisée suivante :

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",

```

```
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "codecatalyst-runner.amazonaws.com",
                "codecatalyst.amazonaws.com"
            ]
        },
        "Action": "sts:AssumeRole"
    }
]
```

- e. Choisissez Suivant.
- f. Dans Politiques d'autorisations `codecatalyst-ecs-build-policy`, recherchez et cochez la case correspondante.
- g. Choisissez Suivant.
- h. Dans Nom du rôle, entrez :

**codecatalyst-ecs-build-role**

- i. Pour la description du rôle, entrez :

**CodeCatalyst ECS build role**

- j. Sélectionnez Créer un rôle.

Vous avez maintenant créé un rôle de build avec une politique d'autorisation et une politique de confiance.

3. Obtenez l'ARN du rôle de build, comme suit :
  - a. Dans le panneau de navigation, choisissez Roles (Rôles).
  - b. Dans le champ de recherche, entrez le nom du rôle que vous venez de créer (`codecatalyst-ecs-build-role`).
  - c. Choisissez le rôle dans la liste.

La page Résumé du rôle apparaît.

- d. En haut, copiez la valeur de l'ARN. Vous en aurez besoin ultérieurement.

## Pour créer un rôle de déploiement

1. Créez une politique pour le rôle, comme suit :
  - a. Connectez-vous à AWS.
  - b. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
  - c. Dans le panneau de navigation, choisissez Politiques (Politiques).
  - d. Choisissez Create Policy (Créer une politique).
  - e. Choisissez l'onglet JSON.
  - f. Supprimez le code existant.
  - g. Collez le code suivant :

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "ecs:DescribeServices",
      "ecs:CreateTaskSet",
      "ecs>DeleteTaskSet",
      "ecs:ListClusters",
      "ecs:RegisterTaskDefinition",
      "ecs:UpdateServicePrimaryTaskSet",
      "ecs:UpdateService",
      "elasticloadbalancing:DescribeTargetGroups",
      "elasticloadbalancing:DescribeListeners",
      "elasticloadbalancing:ModifyListener",
      "elasticloadbalancing:DescribeRules",
      "elasticloadbalancing:ModifyRule",
      "lambda:InvokeFunction",
      "lambda:ListFunctions",
      "cloudwatch:DescribeAlarms",
      "sns:Publish",
      "sns:ListTopics",
      "s3:GetObject",
      "s3:GetObjectVersion",
      "codedeploy:CreateApplication",
      "codedeploy:CreateDeployment",
      "codedeploy:CreateDeploymentGroup",
      "codedeploy:GetApplication",
      "codedeploy:GetDeployment",
      "codedeploy:GetDeploymentGroup",
```

```

    "codedeploy:ListApplications",
    "codedeploy:ListDeploymentGroups",
    "codedeploy:ListDeployments",
    "codedeploy:StopDeployment",
    "codedeploy:GetDeploymentTarget",
    "codedeploy:ListDeploymentTargets",
    "codedeploy:GetDeploymentConfig",
    "codedeploy:GetApplicationRevision",
    "codedeploy:RegisterApplicationRevision",
    "codedeploy:BatchGetApplicationRevisions",
    "codedeploy:BatchGetDeploymentGroups",
    "codedeploy:BatchGetDeployments",
    "codedeploy:BatchGetApplications",
    "codedeploy:ListApplicationRevisions",
    "codedeploy:ListDeploymentConfigs",
    "codedeploy:ContinueDeployment"
  ],
  "Resource": "*",
  "Effect": "Allow"
}, {"Action": [
  "iam:PassRole"
],
  "Effect": "Allow",
  "Resource": "*",
  "Condition": {"StringLike": {"iam:PassedToService": [
    "ecs-tasks.amazonaws.com",
    "codedeploy.amazonaws.com"
  ]
  }
}
]]
}

```

### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources. Vous pouvez ensuite définir la politique avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

- h. Choisissez Suivant : Balises.
- i. Choisissez Suivant : Vérification.
- j. Dans Nom, entrez :

`codecatalyst-ecs-deploy-policy`

- k. Choisissez Créer une politique.

Vous venez de créer une politique d'autorisation.

- 2. Créez le rôle de déploiement comme suit :
  - a. Dans le volet de navigation, sélectionnez Rôles, puis Créer un rôle.
  - b. Choisissez Politique de confiance personnalisée.
  - c. Supprimez la politique de confiance personnalisée existante.
  - d. Ajoutez la politique de confiance personnalisée suivante :

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

- e. Choisissez Suivant.
- f. Dans Politiques d'autorisations `codecatalyst-ecs-deploy-policy`, recherchez et cochez la case correspondante.
- g. Choisissez Suivant.
- h. Dans Nom du rôle, entrez :

```
codecatalyst-ecs-deploy-role
```

- i. Pour la description du rôle, entrez :

```
CodeCatalyst ECS deploy role
```

- j. Sélectionnez Créer un rôle.

Vous venez de créer un rôle de déploiement avec une politique de confiance.

3. Obtenez l'ARN du rôle de déploiement, comme suit :
  - a. Dans le panneau de navigation, choisissez Roles (Rôles).
  - b. Dans le champ de recherche, entrez le nom du rôle que vous venez de créer (codecatalyst-ecs-deploy-role).
  - c. Choisissez le rôle dans la liste.

La page Résumé du rôle apparaît.

- d. En haut, copiez la valeur de l'ARN. Vous en aurez besoin ultérieurement.

## AWS CLI

Pour créer les rôles de génération et de déploiement, suivez la série de procédures suivante.

Pour créer une politique de confiance pour les deux rôles

Comme dans CodeCatalystECSUser AWS CloudShell, créez un fichier de politique de confiance :

1. Créez le fichier :

```
cat > codecatalyst-ecs-trust-policy.json
```

2. À l'invite du terminal, collez le code suivant :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
```

```
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "codecatalyst-runner.amazonaws.com",
                "codecatalyst.amazonaws.com"
            ]
        },
        "Action": "sts:AssumeRole"
    }
]
}
```

3. Placez votre curseur après le dernier crochet (}).
4. Appuyez sur **Enter** puis **Ctrl+d** pour enregistrer le fichier et quitter cat.

Pour créer la politique de construction et le rôle de construction

1. Créez la politique de construction :
  - a. Comme dans CodeCatalystECSUser AWS CloudShell, créez un fichier de politique de construction :

```
cat > codecatalyst-ecs-build-policy.json
```

- b. À l'invite, entrez le code suivant :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:*",
        "ec2:*"
      ],
      "Resource": "*"
    }
  ]
}
```

- c. Placez votre curseur après le dernier crochet (}).
    - d. Appuyez sur **Enter** puis **Ctrl+d** pour enregistrer le fichier et quitter cat.

## 2. Ajoutez la politique de construction à AWS :

```
aws iam create-policy \  
  --policy-name codecatalyst-ecs-build-policy \  
  --policy-document file://codecatalyst-ecs-build-policy.json
```

3. Dans la sortie de commande, notez la "arn" : valeur, par exemple, `arn:aws:iam::111122223333:policy/codecatalyst-ecs-build-policy`. Vous aurez besoin de cet ARN ultérieurement.

## 4. Créez le rôle de build et associez-y la politique de confiance :

```
aws iam create-role \  
  --role-name codecatalyst-ecs-build-role \  
  --assume-role-policy-document file://codecatalyst-ecs-trust-policy.json
```

## 5. Associez la politique de build au rôle de build :

```
aws iam attach-role-policy \  
  --role-name codecatalyst-ecs-build-role \  
  --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-ecs-build-policy
```

Où *arn:aws:iam::111122223333:policy/codecatalyst-ecs-build-policy* est remplacé par l'ARN de la politique de construction que vous avez mentionnée précédemment.

## 6. Affichez les détails du rôle de build :

```
aws iam get-role \  
  --role-name codecatalyst-ecs-build-role
```

7. Notez la "Arn" : valeur du rôle, par exemple `arn:aws:iam::111122223333:role/codecatalyst-ecs-build-role`. Vous aurez besoin de cet ARN ultérieurement.

Pour créer la politique de déploiement et le rôle de déploiement

## 1. Créez une politique de déploiement :

a. Dans AWS CloudShell, créez un fichier de politique de déploiement :



```
cat > codecatalyst-ecs-deploy-policy.json
```

b. À l'invite, entrez le code suivant :

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "ecs:DescribeServices",
      "ecs:CreateTaskSet",
      "ecs>DeleteTaskSet",
      "ecs:ListClusters",
      "ecs:RegisterTaskDefinition",
      "ecs:UpdateServicePrimaryTaskSet",
      "ecs:UpdateService",
      "elasticloadbalancing:DescribeTargetGroups",
      "elasticloadbalancing:DescribeListeners",
      "elasticloadbalancing:ModifyListener",
      "elasticloadbalancing:DescribeRules",
      "elasticloadbalancing:ModifyRule",
      "lambda:InvokeFunction",
      "lambda:ListFunctions",
      "cloudwatch:DescribeAlarms",
      "sns:Publish",
      "sns:ListTopics",
      "s3:GetObject",
      "s3:GetObjectVersion",
      "codedeploy:CreateApplication",
      "codedeploy:CreateDeployment",
      "codedeploy:CreateDeploymentGroup",
      "codedeploy:GetApplication",
      "codedeploy:GetDeployment",
      "codedeploy:GetDeploymentGroup",
      "codedeploy:ListApplications",
      "codedeploy:ListDeploymentGroups",
      "codedeploy:ListDeployments",
      "codedeploy:StopDeployment",
      "codedeploy:GetDeploymentTarget",
      "codedeploy:ListDeploymentTargets",
      "codedeploy:GetDeploymentConfig",
      "codedeploy:GetApplicationRevision",
      "codedeploy:RegisterApplicationRevision",
```

```

    "codedeploy:BatchGetApplicationRevisions",
    "codedeploy:BatchGetDeploymentGroups",
    "codedeploy:BatchGetDeployments",
    "codedeploy:BatchGetApplications",
    "codedeploy:ListApplicationRevisions",
    "codedeploy:ListDeploymentConfigs",
    "codedeploy:ContinueDeployment"
  ],
  "Resource": "*",
  "Effect": "Allow"
}, {"Action": [
  "iam:PassRole"
],
  "Effect": "Allow",
  "Resource": "*",
  "Condition": {"StringLike": {"iam:PassedToService": [
    "ecs-tasks.amazonaws.com",
    "codedeploy.amazonaws.com"
  ]
  }
}
}]
}

```

### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

- c. Placez votre curseur après le dernier crochet (}).
  - d. Appuyez sur **Enter** puis **Ctrl+d** pour enregistrer le fichier et quitter cat.
2. Ajoutez la politique de déploiement à AWS :

```

aws iam create-policy \
  --policy-name codecatalyst-ecs-deploy-policy \
  --policy-document file://codecatalyst-ecs-deploy-policy.json

```

3. Dans le résultat de la commande, notez la "arn" : valeur de la politique de déploiement, par exemple, `arn:aws:iam::111122223333:policy/codecatalyst-ecs-deploy-policy`. Vous aurez besoin de cet ARN ultérieurement.
4. Créez le rôle de déploiement et associez-y la politique de confiance :

```
aws iam create-role \  
  --role-name codecatalyst-ecs-deploy-role \  
  --assume-role-policy-document file://codecatalyst-ecs-trust-policy.json
```

5. Attachez la politique de déploiement au rôle de déploiement, où `arn:aws:iam::111122223333:policy/codecatalyst-ecs-deploy-policy` est remplacé par l'ARN de la politique de déploiement que vous avez mentionnée précédemment.

```
aws iam attach-role-policy \  
  --role-name codecatalyst-ecs-deploy-role \  
  --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-ecs-deploy-policy
```

6. Affichez les détails du rôle de déploiement :

```
aws iam get-role \  
  --role-name codecatalyst-ecs-deploy-role
```

7. Notez la "Arn" : valeur du rôle, par exemple `arn:aws:iam::111122223333:role/codecatalyst-ecs-deploy-role`. Vous aurez besoin de cet ARN ultérieurement.

## Étape 5 : Ajouter AWS des rôles à CodeCatalyst

Au cours de cette étape, vous ajoutez le rôle de création (`codecatalyst-ecs-build-role`) et le rôle de déploiement (`codecatalyst-ecs-deploy-role`) à la connexion au CodeCatalyst compte dans votre espace.

Pour ajouter des rôles de création et de déploiement à votre compte, connectez-vous

1. Dans CodeCatalyst, naviguez jusqu'à votre espace.
2. Choisissez AWS des comptes. La liste des connexions au compte apparaît.
3. Choisissez la connexion au compte qui représente le AWS compte sur lequel vous avez créé vos rôles de création et de déploiement.

4. Choisissez Gérer les rôles depuis AWS la console de gestion.

La page Ajouter un rôle IAM à Amazon CodeCatalyst Space s'affiche. Il se peut que vous deviez vous connecter pour accéder à la page.

5. Sélectionnez Ajouter un rôle existant que vous avez créé dans IAM.

Une liste déroulante apparaît. La liste affiche tous les rôles IAM dotés d'une politique de confiance qui inclut les principaux de `codecatalyst.amazonaws.com` service `codecatalyst-runner.amazonaws.com` et.

6. Dans la liste déroulante, sélectionnez `codecatalyst-ecs-build-role`, puis choisissez Ajouter un rôle.

#### Note

Si vous le voyez `The security token included in the request is invalid`, c'est peut-être parce que vous ne disposez pas des autorisations appropriées. Pour résoudre ce problème, déconnectez-vous et reconnectez-vous avec le AWS compte que vous avez utilisé lors de la création de votre CodeCatalyst espace. AWS

7. Choisissez Ajouter un rôle IAM, choisissez Ajouter un rôle existant que vous avez créé dans IAM, puis dans la liste déroulante, sélectionnez `codecatalyst-ecs-deploy-role`. Choisissez Ajouter un rôle.

Vous avez maintenant ajouté les rôles de création et de déploiement à votre espace.

8. Copiez la valeur du nom CodeCatalyst d'affichage Amazon. Vous aurez besoin de cette valeur ultérieurement, lors de la création de votre flux de travail.

## Étape 6 : Création d'un référentiel source

Au cours de cette étape, vous créez un référentiel source dans CodeCatalyst. Ce référentiel stocke les fichiers source du didacticiel, tels que le fichier de définition des tâches.

Pour plus d'informations sur les référentiels sources, consultez [Création d'un référentiel source](#).

Pour créer un référentiel source

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre projet, `codecatalyst-ecs-project`.

3. Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources.
4. Choisissez Ajouter un référentiel, puis sélectionnez Créer un référentiel.
5. Dans Nom du référentiel, entrez :

```
codecatalyst-ecs-source-repository
```

6. Choisissez Créer.

## Étape 7 : Ajouter des fichiers source

Dans cette section, vous allez ajouter les fichiers source de Hello World à votre CodeCatalyst dépôt, `codecatalyst-ecs-source-repository`. Ils sont composés de :

- Un `index.html` fichier — Affiche un message Hello World dans le navigateur.
- Un `Dockerfile` — Décrit l'image de base à utiliser pour votre image Docker et les commandes Docker à appliquer à celle-ci.
- Un `taskdef.json` fichier : définit l'image Docker à utiliser lors du lancement de tâches dans votre cluster.

La structure des dossiers est la suivante :

```
.  
|- public-html  
|  |- index.html  
|- Dockerfile  
|- taskdef.json
```

### Note

Les instructions suivantes vous montrent comment ajouter les fichiers à l'aide de la CodeCatalyst console, mais vous pouvez utiliser Git si vous préférez. Pour plus de détails, consultez [Clonage d'un dépôt source](#).

## Rubriques

- [index.html](#)
- [Dockerfile](#)

- [taskdef.json](#)

index.html

Le `index.html` fichier affiche un message Hello World dans le navigateur.

Pour ajouter le fichier `index.html`

1. Dans la CodeCatalyst console, accédez à votre référentiel source, `codecatalyst-ecs-source-repository`.
2. Dans Fichiers, choisissez Créer un fichier.
3. Dans Nom du fichier, entrez :

```
public-html/index.html
```

 Important

Assurez-vous d'inclure le `public-html/` préfixe pour créer un dossier du même nom. Le devrait `index.html` se trouver dans ce dossier.

4. Dans la zone de texte, entrez le code suivant :

```
<html>
  <head>
    <title>Hello World</title>
    <style>
      body {
        background-color: black;
        text-align: center;
        color: white;
        font-family: Arial, Helvetica, sans-serif;
      }
    </style>
  </head>
  <body>
    <h1>Hello World</h1>
  </body>
</html>
```

5. Choisissez Commit, puis sélectionnez Commit à nouveau.

Le `index.html` est ajouté à votre dépôt dans un `public-html` dossier.

## Dockerfile

Le Dockerfile décrit l'image Docker de base à utiliser et les commandes Docker à lui appliquer. Pour plus d'informations sur le Dockerfile, consultez le [Dockerfile Reference](#).

Le Dockerfile spécifié ici indique d'utiliser l'image de base d'Apache 2.4 (`httpd`). Il inclut également des instructions pour copier un fichier source appelé `index.html` dans un dossier du serveur Apache qui sert aux pages Web. L'`EXPOSE` instruction du Dockerfile indique à Docker que le conteneur écoute sur le port 80.

Pour ajouter le Dockerfile

1. Dans votre référentiel source, choisissez Créer un fichier.
2. Dans Nom du fichier, entrez :

**Dockerfile**

N'incluez pas d'extension de fichier.

### Important

Le Dockerfile doit résider dans le dossier racine de votre dépôt. La `Docker build` commande du flux de travail s'attend à ce qu'il soit présent.

3. Dans la zone de texte, entrez le code suivant :

```
FROM httpd:2.4
COPY ./public-html/index.html /usr/local/apache2/htdocs/index.html
EXPOSE 80
```

4. Choisissez Commit, puis sélectionnez Commit à nouveau.

Le Dockerfile est ajouté à votre dépôt.

## taskdef.json

Le `taskdef.json` fichier que vous ajoutez à cette étape est le même que celui que vous avez déjà spécifié [Étape 2 : Déployer une application fictive dans Amazon ECS](#) avec la différence suivante :

Au lieu de spécifier un nom d'image Docker codé en dur dans le `image` : champ (`httpd:2.4`), la définition de la tâche utilise ici quelques variables pour désigner l'image : `et. $REPOSITORY_URI $IMAGE_TAG` Ces variables seront remplacées par des valeurs réelles générées par l'action de création du flux de travail lorsque vous exécuterez le flux de travail lors d'une étape ultérieure.

Pour plus de détails sur les paramètres de définition des tâches, consultez la section [Paramètres de définition des tâches](#) dans le manuel Amazon Elastic Container Service Developer Guide.

Pour ajouter le fichier `taskdef.json`

1. Dans votre référentiel source, choisissez Créer un fichier.
2. Dans Nom du fichier, entrez :

```
taskdef.json
```

3. Dans la zone de texte, entrez le code suivant :

```
{
  "executionRoleArn": "arn:aws:iam::account_ID:role/codecatalyst-ecs-task-
  execution-role",
  "containerDefinitions": [
    {
      "name": "codecatalyst-ecs-container",
      # The $REPOSITORY_URI and $IMAGE_TAG variables will be replaced
      # by the workflow at build time (see the build action in the
      # workflow)
      "image": $REPOSITORY_URI:$IMAGE_TAG,
      "essential": true,
      "portMappings": [
        {
          "hostPort": 80,
          "protocol": "tcp",
          "containerPort": 80
        }
      ]
    }
  ],
}
```



```
"requiresCompatibilities": [
  "FARGATE"
],
"networkMode": "awsvpc",
"cpu": "256",
"memory": "512",
"family": "codecatalyst-ecs-task-def"
}
```

Dans le code précédent, remplacez

*arn:aws:iam : :account\_id:role/ -role codecatalyst-ecs-task-execution*

avec l'ARN du rôle d'exécution de tâche que vous avez noté dans [Pour créer le rôle d'exécution des tâches](#).

4. Choisissez Commit, puis sélectionnez Commit à nouveau.

Le `taskdef.json` fichier est ajouté à votre dépôt.

## Étape 8 : créer et exécuter un flux de travail

Au cours de cette étape, vous créez un flux de travail qui prend vos fichiers source, les intègre dans une image Docker, puis déploie l'image sur votre cluster Amazon ECS. Ce déploiement remplace l'application fictive Apache existante.

Le flux de travail comprend les éléments de base suivants qui s'exécutent de manière séquentielle :

- Un déclencheur : ce déclencheur lance automatiquement l'exécution du flux de travail lorsque vous apportez une modification à votre référentiel source. Pour plus d'informations sur les déclencheurs, consultez [Utilisation de déclencheurs](#).
- Une action de génération (BuildBackend) — Au déclenchement, l'action crée l'image Docker à l'aide du Dockerfile et envoie l'image vers Amazon ECR. L'action de génération met également à jour le `taskdef.json` avec la valeur de `image` champ correcte, puis crée un artefact de sortie de ce fichier. Cet artefact est utilisé comme entrée pour l'action de déploiement, qui est la suivante.

Pour plus d'informations sur l'action de génération, consultez [Création à l'aide de flux de travail dans CodeCatalyst](#).

- Une action de déploiement (DeployToECS) : une fois l'action de création terminée, l'action de déploiement recherche l'artefact de sortie généré par l'action de génération (TaskDefArtifact),

en trouve l'`taskdef.json` intérieur et l'enregistre auprès de votre service Amazon ECS. Le service suit ensuite les instructions du `taskdef.json` fichier pour exécuter trois tâches Amazon ECS et les conteneurs Hello World Docker associés au sein de votre cluster Amazon ECS.

Pour créer un flux de travail

1. Dans le volet de navigation de la CodeCatalyst console, choisissez CI/CD, puis Workflows.
2. Choisissez Créer un flux de travail.
3. Pour Référentiel source, choisissez `codecatalyst-ecs-source-repository`.
4. Pour Branch, choisissez `main`.
5. Choisissez Créer.
6. Supprimez l'exemple de code YAML.
7. Ajoutez le code YAML suivant :

```
Name: codecatalyst-ecs-workflow
SchemaVersion: 1.0

Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  BuildBackend:
    Identifiant: aws/build@v1
    Environment:
      Name: codecatalyst-ecs-environment
    Connections:
      - Name: codecatalyst-account-connection
        Role: codecatalyst-ecs-build-role
    Inputs:
      Sources:
        - WorkflowSource
      Variables:
        - Name: REPOSITORY_URI
          Value: 111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo
        - Name: IMAGE_TAG
          Value: ${WorkflowSource.CommitId}
    Configuration:
      Steps:
```

```
#pre_build:
- Run: echo Logging in to Amazon ECR...
- Run: aws --version
- Run: aws ecr get-login-password --region us-west-2 | docker login --
username AWS --password-stdin 111122223333.dkr.ecr.us-west-2.amazonaws.com
#build:
- Run: echo Build started on `date`
- Run: echo Building the Docker image...
- Run: docker build -t $REPOSITORY_URI:latest .
- Run: docker tag $REPOSITORY_URI:latest $REPOSITORY_URI:$IMAGE_TAG
#post_build:
- Run: echo Build completed on `date`
- Run: echo Pushing the Docker images...
- Run: docker push $REPOSITORY_URI:latest
- Run: docker push $REPOSITORY_URI:$IMAGE_TAG
# Replace the variables in taskdef.json
- Run: find taskdef.json -type f | xargs sed -i "s|\\$REPOSITORY_URI|
$REPOSITORY_URI|g"
- Run: find taskdef.json -type f | xargs sed -i "s|\\$IMAGE_TAG|$IMAGE_TAG|
g"
- Run: cat taskdef.json
# The output artifact will be a zip file that contains a task definition
file.
Outputs:
  Artifacts:
    - Name: TaskDefArtifact
      Files:
        - taskdef.json
DeployToECS:
  DependsOn:
    - BuildBackend
  Identifier: aws/ecs-deploy@v1
  Environment:
    Name: codecatalyst-ecs-environment
  Connections:
    - Name: codecatalyst-account-connection
      Role: codecatalyst-ecs-deploy-role
Inputs:
  Sources: []
  Artifacts:
    - TaskDefArtifact
Configuration:
  region: us-west-2
  cluster: codecatalyst-ecs-cluster
```

```
service: codecatalyst-ecs-service
task-definition: taskdef.json
```

Dans le code précédent, remplacez :

- Les deux instances portent *codecatalyst-ecs-environment* le nom de l'environnement dans lequel vous l'avez créé [Prérequis](#).
- Dans les deux cas, *codecatalyst-account-connection* avec le nom d'affichage de la connexion à votre compte. Le nom d'affichage peut être un chiffre. Pour plus d'informations, consultez [Étape 5 : Ajouter AWS des rôles à CodeCatalyst](#).
- *codecatalyst-ecs-build-role* avec le nom du rôle de build que vous avez créé dans [Étape 4 : Création de AWS rôles](#).
- *111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo* (dans la Value : propriété) avec l'URI du référentiel Amazon ECR dans lequel vous l'avez créé. [Étape 3 : Création d'un référentiel d'images Amazon ECR](#)
- *111122223333.dkr.ecr.us-west-2.amazonaws.com* (dans la Run : aws ecr commande) avec l'URI du référentiel Amazon ECR sans le suffixe d'image (). / *codecatalyst-ecs-image-repo*
- *codecatalyst-ecs-deploy-role* avec le nom du rôle de déploiement que vous avez créé dans [Étape 4 : Création de AWS rôles](#).
- Les deux instances de *us-west-2* avec votre code de région. AWS Pour obtenir la liste des codes de région, consultez la section [Points de terminaison régionaux](#) dans le Références générales AWS.

#### Note

Si vous avez décidé de ne pas créer de rôles de création et de déploiement, remplacez *codecatalyst-ecs-build-role* et *codecatalyst-ecs-deploy-role* par le nom du CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle. Pour plus d'informations sur ce rôle, consultez [Étape 4 : Création de AWS rôles](#).

**i** Tip

Au lieu d'utiliser les `sed` commandes `find` et indiquées dans le code de flux de travail précédent pour mettre à jour le référentiel et le nom de l'image, vous pouvez utiliser l'action de définition de tâche `Render Amazon ECS` à cette fin. Pour plus d'informations, consultez [Ajout de l'action « Render la définition de tâche Amazon ECS »](#).

8. (Facultatif) Choisissez `Valider` pour vous assurer que le code YAML est valide avant de valider.
9. Choisissez `Commit` (`Valider`).
10. Dans la boîte de dialogue du flux de travail de validation, entrez les informations suivantes :
  - a. Pour le message de validation, supprimez le texte et entrez :

**Add first workflow**

- b. Pour `Repository`, choisissez `codecatalyst-ecs-source-repository`.
- c. Pour `Nom de la branche`, choisissez `main`.
- d. Choisissez `Commit` (`Valider`).

Vous venez de créer un flux de travail. L'exécution d'un flux de travail démarre automatiquement en raison du déclencheur défini en haut du flux de travail. Plus précisément, lorsque vous avez validé (et transféré) le `workflow.yaml` fichier dans votre référentiel source, le déclencheur a lancé l'exécution du flux de travail.

Pour consulter la progression de l'exécution du flux de travail

1. Dans le volet de navigation de la CodeCatalyst console, choisissez `CI/CD`, puis `Workflows`.
2. Choisissez le flux de travail que vous venez de créer, `codecatalyst-ecs-workflow`.
3. Choisissez `BuildBackend` de voir la progression de la construction.
4. Choisissez `DeployToECS` pour voir la progression du déploiement.

Pour plus d'informations sur l'affichage des détails des exécutions, consultez [Afficher le statut et les détails de l'exécution du flux de travail](#).

## Pour vérifier le déploiement

1. Ouvrez la console classique Amazon ECS à partir de l'adresse <https://console.aws.amazon.com/ecs/>.
2. Choisissez votre cluster, `codecatalyst-ecs-cluster`.
3. Choisissez l'onglet Tasks.
4. Choisissez l'une des trois tâches.
5. Dans le champ IP publique, choisissez une adresse ouverte.

Une page « Hello World » apparaît dans le navigateur, indiquant que le service Amazon ECS a correctement déployé votre application.

## Étape 9 : Apportez une modification à vos fichiers source

Dans cette section, vous allez apporter une modification au `index.html` fichier dans votre référentiel source. Cette modification oblige le flux de travail à créer une nouvelle image Docker, à l'étiqueter avec un ID de validation, à la transmettre à Amazon ECR et à la déployer sur Amazon ECS.

### Pour modifier le fichier `index.html`

1. Dans la CodeCatalyst console, dans le volet de navigation, choisissez Code, puis Référentiels sources, puis choisissez votre référentiel, `codecatalyst-ecs-source-repository`.

2. Choisissez `public-html` et `index.html`.

Le contenu de `index.html` apparaît.

3. Choisissez Modifier.
4. À la ligne 14, remplacez le `Hello World` texte par `Tutorial complete!`.
5. Choisissez Commit, puis sélectionnez Commit à nouveau.

La validation entraîne le démarrage d'un nouveau flux de travail.

6. (Facultatif) Accédez à la page principale de votre dépôt source, choisissez Afficher les validations, puis notez l'ID de validation associé à la `index.html` modification.
7. Suivez la progression du déploiement :
  - a. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
  - b. Choisissez `codecatalyst-ecs-workflow` d'afficher la dernière édition.

- c. Choisissez BuildBackend, et DeployToECS pour voir la progression de l'exécution du flux de travail.
8. Vérifiez que votre application a été mise à jour, comme suit :
    - a. Ouvrez la console classique Amazon ECS à partir de l'adresse <https://console.aws.amazon.com/ecs/>.
    - b. Choisissez votre cluster, `codecatalyst-ecs-cluster`.
    - c. Choisissez l'onglet Tasks.
    - d. Choisissez l'une des trois tâches.
    - e. Dans le champ IP publique, choisissez une adresse ouverte.  
  
Une Tutorial complète! page apparaît.
  9. (Facultatif) Dans AWS, passez à la console Amazon ECR et vérifiez que la nouvelle image Docker a été étiquetée avec l'ID de validation indiqué à l'étape 6.

## Nettoyage

Nettoyez les fichiers et les services utilisés dans ce didacticiel pour éviter de vous les faire facturer.

Dans le AWS Management Console, nettoyez dans cet ordre :

1. Dans Amazon ECS, procédez comme suit :
  - a. Supprimer `codecatalyst-ecs-service`.
  - b. Supprimer `codecatalyst-ecs-cluster`.
  - c. Désenregistrer `codecatalyst-ecs-task-definition`.
2. Dans Amazon ECR, supprimez `codecatalyst-ecs-image-repo`.
3. Dans Amazon EC2, supprimez `codecatalyst-ecs-security-group`.
4. Dans IAM Identity Center, supprimez :
  - a. `CodeCatalystECSUser`
  - b. `CodeCatalystECSPermissionSet`

Dans la CodeCatalyst console, nettoyez comme suit :

1. Supprimer `codecatalyst-ecs-workflow`.
2. Supprimer `codecatalyst-ecs-environment`.

3. `Supprimercodecatalyst-ecs-source-repository`.
4. `Supprimercodecatalyst-ecs-project`.

Dans ce didacticiel, vous avez appris à déployer une application sur un service Amazon ECS à l'aide d'un CodeCatalyst flux de travail et d'une action Deploy to Amazon ECS.

## Tutoriel : Déploiement d'une application sur Amazon EKS

Dans ce didacticiel, vous apprendrez à déployer une application conteneurisée dans Amazon Elastic Kubernetes Service à l'aide d'un flux de travail Amazon, CodeCatalyst d'Amazon EKS et de quelques autres services. AWS L'application déployée est une simple application « Hello, World ! » site Web construit sur une image Docker d'un serveur Web Apache. Le didacticiel explique les tâches de préparation requises, telles que la configuration d'une machine de développement et d'un cluster Amazon EKS, puis décrit comment créer un flux de travail pour créer l'application et la déployer dans le cluster.

Une fois le déploiement initial terminé, le didacticiel vous demande de modifier la source de votre application. Cette modification entraîne la création d'une nouvelle image Docker et son transfert vers votre référentiel d'images Docker avec de nouvelles informations de révision. La nouvelle version de l'image Docker est ensuite déployée dans Amazon EKS.

### Tip

Au lieu de suivre ce didacticiel, vous pouvez utiliser un plan qui effectue une configuration complète d'Amazon EKS pour vous. Vous devez utiliser le plan de déploiement de l'application EKS. Pour plus d'informations, consultez [Création d'un projet à l'aide d'un plan](#).

### Rubriques

- [Prérequis](#)
- [Étape 1 : configurer votre machine de développement](#)
- [Étape 2 : créer un cluster Amazon EKS](#)
- [Étape 3 : Création d'un référentiel d'images Amazon ECR](#)
- [Étape 4 : Ajouter des fichiers source](#)
- [Étape 5 : Création de AWS rôles](#)
- [Étape 6 : Ajouter AWS des rôles à CodeCatalyst](#)



- [Étape 7 : mettez à jour le ConfigMap](#)
- [Étape 8 : créer et exécuter un flux de travail](#)
- [Étape 9 : Apportez une modification à vos fichiers source](#)
- [Nettoyage](#)

## Prérequis

Avant de commencer ce didacticiel :

- Vous avez besoin d'un CodeCatalyst espace Amazon avec un AWS compte connecté. Pour plus d'informations, consultez [Création d'un espace prenant en charge les utilisateurs de AWS Builder ID](#).
- Dans votre espace, vous avez besoin d'un CodeCatalyst projet vide, créé à partir de zéro, appelé :

```
codecatalyst-eks-project
```

Pour plus d'informations, consultez [Création d'un projet vide dans Amazon CodeCatalyst](#).

- Dans votre projet, vous avez besoin d'un dépôt CodeCatalyst source vide appelé :

```
codecatalyst-eks-source-repository
```

Pour plus d'informations, consultez [Référentiels de sources dans CodeCatalyst](#).

- Dans votre projet, vous avez besoin d'un environnement CodeCatalyst CI/CD (et non d'un environnement de développement) appelé :

```
codecatalyst-eks-environment
```

Configurez cet environnement comme suit :

- Choisissez n'importe quel type, tel que Non-production.
- AWS Connectez-y votre compte.

Pour plus d'informations, consultez [Utilisation des environnements](#).

## Étape 1 : configurer votre machine de développement

La première étape de ce didacticiel consiste à configurer une machine de développement avec quelques outils que vous utiliserez tout au long de ce didacticiel. Ces outils sont les suivants :

- l'`eksctl` utilitaire — pour la création de clusters
- l'`kubectl` utilité — une condition préalable à `eksctl`
- le AWS CLI — également une condition préalable pour `eksctl`

Vous pouvez installer ces outils sur votre machine de développement existante si vous en avez une, ou vous pouvez utiliser un CodeCatalyst environnement de développement basé sur le cloud. L'avantage d'un environnement CodeCatalyst de développement est qu'il est facile à installer et à démonter, et qu'il est intégré à d'autres CodeCatalyst services, ce qui vous permet de suivre ce didacticiel en moins d'étapes.

Ce didacticiel part du principe que vous utiliserez un CodeCatalyst environnement de développement.

Les instructions suivantes décrivent un moyen rapide de lancer un environnement de CodeCatalyst développement et de le configurer avec les outils requis, mais si vous souhaitez obtenir des instructions détaillées, consultez :

- [Création d'un environnement de développement](#) dans ce guide.
- [Installation de kubectl](#) dans le guide de l'utilisateur Amazon EKS.
- [Installation ou mise à niveau d'eksctl](#) dans le guide de l'utilisateur Amazon EKS.
- [Installation ou mise à jour de la dernière version du AWS CLI](#) dans le guide de AWS Command Line Interface l'utilisateur.

Pour lancer un environnement de développement

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre projet, `codecatalyst-eks-project`.
3. Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources.
4. Choisissez le nom de votre dépôt source, `codecatalyst-eks-source-repository`.
5. En haut, choisissez Create Dev Environment, puis choisissez AWS Cloud9 (dans le navigateur).
6. Assurez-vous que les options Travailler dans la branche et la branche principale existantes sont sélectionnées, puis choisissez Créer.

Votre environnement de développement s'ouvre dans un nouvel onglet du navigateur, et votre dépôt (codecatalyst-eks-source-repository) y est cloné.

### Pour installer et configurer kubectl

1. Dans le terminal Dev Environment, entrez :

```
curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.18.9/2020-11-02/bin/linux/amd64/kubectl
```

2. Entrez :

```
chmod +x ./kubectl
```

3. Entrez :

```
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$PATH:$HOME/bin
```

4. Entrez :

```
echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
```

5. Entrez :

```
kubectl version --short --client
```

6. Vérifiez qu'une version apparaît.

Vous êtes maintenant installé `kubectl`.

### Pour installer et configurer eksctl

#### Note

`eksctl` n'est pas strictement obligatoire car vous pouvez utiliser à la place `kubectl`. Cependant, `eksctl` présente l'avantage d'automatiser une grande partie de la configuration du cluster, et c'est donc l'outil recommandé pour ce didacticiel.

1. Dans le terminal Dev Environment, entrez :

```
curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_${uname -s}_amd64.tar.gz" | tar xz -C /tmp
```

2. Entrez :

```
sudo cp /tmp/eksctl /usr/bin
```

3. Entrez :

```
eksctl version
```

4. Vérifiez qu'une version apparaît.

Vous êtes maintenant installé `eksctl`.

Pour vérifier AWS CLI que le

1. Dans le terminal Dev Environment, entrez :

```
aws --version
```

2. Vérifiez qu'une version apparaît pour vérifier qu'elle AWS CLI est installée.


Effectuez les procédures restantes pour configurer le AWS CLI avec les autorisations d'accès nécessaires AWS.

Pour configurer le AWS CLI

Vous devez le configurer AWS CLI avec des clés d'accès et un jeton de session pour lui donner accès aux AWS services. Les instructions suivantes fournissent un moyen rapide de configurer les clés et le jeton, mais si vous souhaitez obtenir des instructions détaillées, reportez-vous AWS CLI à [la section Configuration du](#) guide de l'AWS Command Line Interface utilisateur.

1. Créez un utilisateur IAM Identity Center, comme suit :
  - a. Connectez-vous à la AWS IAM Identity Center console AWS Management Console et ouvrez-la à l'[adresse https://console.aws.amazon.com/singlesignon/](https://console.aws.amazon.com/singlesignon/).

(Vous devrez peut-être choisir Activer si vous ne vous êtes jamais connecté à IAM Identity Center auparavant.)

 Note

Assurez-vous de vous connecter à l'aide de Compte AWS celui qui est connecté à votre CodeCatalyst espace. Vous pouvez vérifier quel compte est connecté en accédant à votre espace et en choisissant l'onglet Comptes AWS. Pour plus d'informations, consultez [Création d'un espace prenant en charge les utilisateurs de AWS Builder ID](#).

- b. Dans le panneau de navigation, choisissez Users (Utilisateurs), puis Add user (Ajouter un utilisateur).
- c. Dans Nom d'utilisateur, entrez :

**codecatalyst-eks-user**

- d. Sous Mot de passe, choisissez Générer un mot de passe à usage unique que vous pouvez partager avec cet utilisateur.
- e. Dans Adresse e-mail et Confirmer l'adresse e-mail, entrez une adresse e-mail qui n'existe pas encore dans IAM Identity Center.
- f. Dans Prénom, entrez :

**codecatalyst-eks-user**

- g. Dans Nom de famille, entrez :

**codecatalyst-eks-user**


- h. Dans Nom d'affichage, conservez :

**codecatalyst-eks-user codecatalyst-eks-user**

- i. Choisissez Suivant.
- j. Sur la page Ajouter un utilisateur aux groupes, choisissez Next.
- k. Sur la page Vérifier et ajouter un utilisateur, passez en revue les informations et choisissez Ajouter un utilisateur.



- a. Assurez-vous de disposer de l'URL du portail AWS d'accès, du nom d'utilisateur et du mot de passe à usage unique pour `codecatalyst-eks-user`. Vous devriez avoir copié ces informations dans un éditeur de texte plus tôt.

 Note

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

- b. Déconnectez-vous de AWS.
- c. Collez l'URL du portail d' AWS accès dans la barre d'adresse de votre navigateur.
- d. Connectez-vous avec :

- Nom d'utilisateur :

```
codecatalyst-eks-user
```

- Mot de passe :

*one-time-password*

- e. Dans Définir un nouveau mot de passe, entrez un nouveau mot de passe et choisissez Définir un nouveau mot de passe.

Une Compte AWSboîte apparaît à l'écran.

- f. Choisissez Compte AWS, puis choisissez le nom auquel vous avez attribué l'`codecatalyst-eks-user`utilisateur et le jeu d'autorisations. Compte AWS
- g. À côté `codecatalyst-eks-permission-set`, choisissez Ligne de commande ou accès par programmation.
- h. Copiez les commandes au milieu de la page. Ils ressemblent à ce qui suit :

```
export AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"  
export AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"  
export AWS_SESSION_TOKEN="session-token"
```

... où le *jeton de session* est une longue chaîne aléatoire.

5. Ajoutez les clés d'accès et le jeton de session au AWS CLI, comme suit :
  - a. Retournez à votre CodeCatalyst environnement de développement.
  - b. À l'invite du terminal, collez les commandes que vous avez copiées. Appuyez sur Entrée.

Vous avez maintenant configuré le AWS CLI avec des clés d'accès et un jeton de session. Vous pouvez désormais utiliser AWS CLI pour effectuer les tâches requises par ce didacticiel.

#### Important

Si, à tout moment au cours de ce didacticiel, vous voyez des messages similaires aux suivants :

```
Unable to locate credentials. You can configure credentials by running "aws configure".
```

Ou:

```
ExpiredToken: The security token included in the request is expired
```

... c'est parce que votre AWS CLI session a expiré. Dans ce cas, n'exécutez pas la `aws configure` commande. Utilisez plutôt les instructions de l'étape 4 de cette procédure qui commence par `Obtain codecatalyst-eks-user's access key and session token` pour actualiser votre session.

## Étape 2 : créer un cluster Amazon EKS

Dans cette section, vous allez créer un cluster dans Amazon EKS. Les instructions ci-dessous décrivent un moyen rapide de créer le cluster en utilisant `eksctl`, mais si vous souhaitez obtenir des instructions détaillées, consultez :

- [Commencer à utiliser eksctl dans le guide](#) de l'utilisateur Amazon EKS

or

- [Démarrage avec la console et AWS CLI](#) dans le guide de l'utilisateur Amazon EKS (cette rubrique fournit des `kubectl` instructions pour créer le cluster)



**Note**

Les [clusters privés](#) ne sont pas pris en charge par l' CodeCatalyst intégration avec Amazon EKS.

## Avant de commencer

Assurez-vous d'avoir effectué les tâches suivantes sur votre machine de développement :

- L'`eksctl` utilitaire a été installé.
- L'`kubectl` utilitaire a été installé.
- Vous l'avez installé AWS CLI et configuré avec des clés d'accès et un jeton de session.

Pour plus d'informations sur la façon d'effectuer ces tâches, consultez [Étape 1 : configurer votre machine de développement](#).

## Pour créer un cluster

**⚠ Important**

N'utilisez pas l'interface utilisateur du service Amazon EKS pour créer le cluster car celui-ci ne sera pas configuré correctement. Utilisez l'`eksctl` utilitaire, comme décrit dans les étapes suivantes.

1. Accédez à votre environnement de développement.
2. Créez un cluster et des nœuds :

```
eksctl create cluster --name codecatalyst-eks-cluster --region us-west-2
```

Où :

- *codecatalyst-eks-cluster* est remplacé par le nom que vous souhaitez donner à votre cluster.
- *us-west-2* est remplacé par votre région.

Après 10 à 20 minutes, un message semblable au suivant apparaît :

```
EKS cluster "codecatalyst-eks-cluster" in "us-west-2" region is ready
```

**Note**

Vous verrez plusieurs `waiting for CloudFormation stack` messages lors de la AWS création de votre cluster. Ceci est normal.

3. Vérifiez que votre cluster a été créé avec succès :

```
kubectl cluster-info
```

Vous verrez un message similaire au suivant, indiquant une création de cluster réussie :

```
Kubernetes master is running at https://long-string.gr7.us-west-2.eks.amazonaws.com
CoreDNS is running at https://long-string.gr7.us-west-2.eks.amazonaws.com/api/v1/
namespaces/kube-system/services/kube-dns:dns/proxy
```

### Étape 3 : Création d'un référentiel d'images Amazon ECR

Dans cette section, vous allez créer un référentiel d'images privé dans Amazon Elastic Container Registry (Amazon ECR). Ce référentiel stocke l'image Docker pour le didacticiel.

Pour plus d'informations sur Amazon ECR, consultez le guide de l'utilisateur d'Amazon Elastic Container Registry.

Pour créer un référentiel d'images dans Amazon ECR

1. Accédez à votre environnement de développement.
2. Créez un référentiel vide dans Amazon ECR :

```
aws ecr create-repository --repository-name codecatalyst-eks-image-repo
```

*codecatalyst-eks-image-repo* Remplacez-le par le nom que vous souhaitez attribuer au référentiel Amazon ECR.

Ce didacticiel part du principe que vous avez donné un nom à votre dépôt `codecatalyst-eks-image-repo`.

### 3. Affichez les détails du référentiel Amazon ECR :

```
aws ecr describe-repositories \
  --repository-names codecatalyst-eks-image-repo
```

### 4. Notez la "repositoryUri" : valeur, par exemple, `111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-eks-image-repo`.

Vous en aurez besoin ultérieurement lors de l'ajout du référentiel à votre flux de travail.

## Étape 4 : Ajouter des fichiers source

Dans cette section, vous allez ajouter les fichiers source de l'application à votre référentiel source (`codecatalyst-eks-source-repository`). Ils sont composés de :

- Un `index.html` fichier — Affiche un message « Bonjour tout le monde ! » message dans le navigateur.
- Un `Dockerfile` — Décrit l'image de base à utiliser pour votre image Docker et les commandes Docker à appliquer à celle-ci.
- Un `deployment.yaml` fichier : le manifeste Kubernetes qui définit le service et le déploiement de Kubernetes.

La structure des dossiers est la suivante :

```
|– codecatalyst-eks-source-repository
  |– Kubernetes
    |– deployment.yaml
  |– public-html
  |   |– index.html
  |– Dockerfile
```

## Rubriques

- [index.html](#)
- [Dockerfile](#)

- [deployment.yaml](#)

index.html

Le `index.html` fichier affiche un message « Bonjour tout le monde ! » message dans le navigateur.

Pour ajouter le fichier `index.html`

1. Accédez à votre environnement de développement.
2. Dans `codecatalyst-eks-source-repository`, créez un dossier appelé `public-html`.
3. Dans `public-html`, créez un fichier appelé `index.html` avec le contenu suivant :

```
<html>
  <head>
    <title>Hello World</title>
    <style>
      body {
        background-color: black;
        text-align: center;
        color: white;
        font-family: Arial, Helvetica, sans-serif;
      }
    </style>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

4. À l'invite du terminal, entrez :

```
cd /projects/codecatalyst-eks-source-repository
```

5. Ajoutez, validez et envoyez :

```
git add .
git commit -m "add public-html/index.html"
git push
```

Le `index.html` est ajouté à votre dépôt dans un `public-html` dossier.

## Dockerfile

Le Dockerfile décrit l'image Docker de base à utiliser et les commandes Docker à lui appliquer. Pour plus d'informations sur le Dockerfile, consultez le [Dockerfile Reference](#).

Le Dockerfile spécifié ici indique d'utiliser l'image de base d'Apache 2.4 (`httpd`). Il inclut également des instructions pour copier un fichier source appelé `index.html` dans un dossier du serveur Apache qui sert aux pages Web. L'`EXPOSE` instruction du Dockerfile indique à Docker que le conteneur écoute sur le port 80.

### Pour ajouter le Dockerfile

1. Dans `codecatalyst-eks-source-repository`, créez un fichier appelé `Dockerfile` avec le contenu suivant :

```
FROM httpd:2.4
COPY ./public-html/index.html /usr/local/apache2/htdocs/index.html
EXPOSE 80
```

N'incluez pas d'extension de fichier.

#### Important

Le Dockerfile doit résider dans le dossier racine de votre dépôt. La `Docker build` commande du flux de travail s'attend à ce qu'il soit présent.

2. Ajoutez, validez et envoyez :

```
git add .
git commit -m "add Dockerfile"
git push
```

Le Dockerfile est ajouté à votre dépôt.

## deployment.yaml

Dans cette section, vous allez ajouter un `deployment.yaml` fichier à votre dépôt. Le `deployment.yaml` fichier est un manifeste Kubernetes qui définit deux types ou types de ressources Kubernetes à exécuter : un « service » et un « déploiement ».

- Le « service » déploie un équilibreur de charge dans Amazon EC2. L'équilibreur de charge vous fournit une URL publique connectée à Internet et un port standard (port 80) que vous pouvez utiliser pour accéder à « Hello, World ! » application.
- Le « déploiement » déploie trois pods, et chaque pod contiendra un conteneur Docker avec le message « Hello, World ! » application. Les trois pods sont déployés sur les nœuds créés lorsque vous avez créé le cluster.

Le manifeste présenté dans ce didacticiel est court ; toutefois, un manifeste peut inclure un certain nombre de types de ressources Kubernetes, tels que des pods, des jobs, des entrées et des politiques réseau. En outre, vous pouvez utiliser plusieurs fichiers manifestes si votre déploiement est complexe.

Pour ajouter un fichier de déploiement .yaml

1. Dans `codecatalyst-eks-source-repository`, créez un dossier appelé `Kubernetes`.
2. Dans `Kubernetes`, créez un fichier appelé `deployment.yaml` avec le contenu suivant :

```
apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
    app: my-app
spec:
  type: LoadBalancer
  selector:
    app: my-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deployment
  labels:
    app: my-app
spec:
  replicas: 3
```

```
selector:
  matchLabels:
    app: my-app
template:
  metadata:
    labels:
      app: my-app
  spec:
    containers:
      - name: codecatalyst-eks-container
        # The $REPOSITORY_URI and $IMAGE_TAG placeholders will be replaced by
        # actual values supplied by the build action in your workflow
        image: $REPOSITORY_URI:$IMAGE_TAG
        ports:
          - containerPort: 80
```

### 3. Ajoutez, validez et envoyez :

```
git add .
git commit -m "add Kubernetes/deployment.yaml"
git push
```

Le `deployment.yaml` fichier est ajouté à votre dépôt dans un dossier appelé `Kubernetes`.

Vous avez maintenant ajouté tous vos fichiers sources.

Prenez le temps de vérifier votre travail et de vous assurer que vous avez placé tous les fichiers dans les bons dossiers. La structure des dossiers est la suivante :

```
|– codecatalyst-eks-source-repository
  |– Kubernetes
    |– deployment.yaml
  |– public-html
  |   |– index.html
  |– Dockerfile
```

## Étape 5 : Création de AWS rôles

Dans cette section, vous allez créer les rôles AWS IAM dont votre CodeCatalyst flux de travail aura besoin pour fonctionner. Ces rôles sont les suivants :

- Rôle de création : accorde à l'action de CodeCatalyst création (dans le flux de travail) l'autorisation d'accéder à votre AWS compte et d'écrire sur Amazon ECR et Amazon EC2.
- Rôle de déploiement : accorde à l'action CodeCatalyst Deploy to Kubernetes cluster (dans le flux de travail) l'autorisation d'accéder à votre compte AWS et à Amazon EKS.

Pour plus d'informations sur les rôles IAM, consultez la section Rôles [IAM dans le Guide](#) de l'AWS Identity and Access Management utilisateur.

#### Note

Pour gagner du temps, vous pouvez créer un seul rôle, appelé `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle, au lieu des deux rôles répertoriés précédemment. Pour plus d'informations, consultez [Création du CodeCatalystWorkflowDevelopmentRole-\*spaceName\* rôle pour votre compte et votre espace](#). Sachez que le `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle dispose d'autorisations très étendues, ce qui peut présenter un risque de sécurité. Nous vous recommandons de n'utiliser ce rôle que dans les didacticiels et les scénarios où la sécurité est moins préoccupante. Ce didacticiel part du principe que vous créez les deux rôles répertoriés précédemment.

Pour créer les rôles de génération et de déploiement, suivez la série de procédures suivante.

1. Pour créer une politique de confiance pour les deux rôles
  1. Accédez à votre environnement de développement.
  2. Dans le Cloud9-*long-string* répertoire, créez un fichier appelé `codecatalyst-eks-trust-policy.json` avec le contenu suivant :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      }
    }
  ]
}
```



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

## 2. Pour créer la politique de build pour le rôle de build

- Dans le Cloud9-*long-string* répertoire, créez un fichier appelé `codecatalyst-eks-build-policy.json` avec le contenu suivant :

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "ecr:*",  
        "ec2:*"  
      ],  
      "Resource": "*"   
    }  
  ]  
}
```

### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

## 3. Pour créer la politique de déploiement pour le rôle de déploiement

- Dans le Cloud9-*long-string* répertoire, créez un fichier appelé `codecatalyst-eks-deploy-policy.json` avec le contenu suivant :

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

### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

Vous avez maintenant ajouté trois documents de politique à votre environnement de développement. La structure de votre répertoire ressemble désormais à ceci :

```
|– Cloud9-long-string
  |– .c9
  |– codecatalyst-eks-source-repository
    |– Kubernetes
    |– public-html
    |– Dockerfile
  codecatalyst-eks-build-policy.json
  codecatalyst-eks-deploy-policy.json
  codecatalyst-eks-trust-policy.json
```

## 4. Pour ajouter la politique de construction à AWS

1. Dans le terminal Dev Environment, entrez :

```
cd /projects
```

2. Entrez :

```
aws iam create-policy \  
  --policy-name codecatalyst-eks-build-policy \  
  --policy-document file:///codecatalyst-eks-build-policy.json
```

3. Appuyez sur Entrée.

4. Dans la sortie de commande, notez la "arn" : valeur, par exemple, `arn:aws:iam::111122223333:policy/codecatalyst-eks-build-policy`. Vous aurez besoin de cet ARN ultérieurement.

5. Pour ajouter la politique de déploiement à AWS

1. Entrez :

```
aws iam create-policy \  
  --policy-name codecatalyst-eks-deploy-policy \  
  --policy-document file:///codecatalyst-eks-deploy-policy.json
```

2. Appuyez sur Entrée.

3. Dans le résultat de la commande, notez la "arn" : valeur de la politique de déploiement, par exemple, `arn:aws:iam::111122223333:policy/codecatalyst-eks-deploy-policy`. Vous aurez besoin de cet ARN ultérieurement.

6. Pour créer le rôle de build

1. Entrez :

```
aws iam create-role \  
  --role-name codecatalyst-eks-build-role \  
  --assume-role-policy-document file:///codecatalyst-eks-trust-policy.json
```

2. Appuyez sur Entrée.

3. Entrez :

```
aws iam attach-role-policy \  
  --role-name codecatalyst-eks-build-role \  
  --policy-arn arn:aws:iam::111122223333:policy/codecatalyst-eks-build-policy
```

```
--role-name codecatalyst-eks-build-role \  
--policy-arn arn:aws:iam::111122223333:policy/codecatalyst-eks-build-policy
```

Où *arn:aws:iam::111122223333:policy/codecatalyst-eks-build-policy* est remplacé par l'ARN de la politique de construction que vous avez mentionnée précédemment.

4. Appuyez sur Entrée.
5. À l'invite du terminal, entrez :

```
aws iam get-role \  
--role-name codecatalyst-eks-build-role
```

6. Appuyez sur Entrée.
7. Notez la "Arn" : valeur du rôle, par exemple *arn:aws:iam::111122223333:role/codecatalyst-eks-build-role*. Vous aurez besoin de cet ARN ultérieurement.

## 7. Pour créer le rôle de déploiement

1. Entrez :

```
aws iam create-role \  
--role-name codecatalyst-eks-deploy-role \  
--assume-role-policy-document file://codecatalyst-eks-trust-policy.json
```

2. Appuyez sur Entrée.
3. Entrez :

```
aws iam attach-role-policy \  
--role-name codecatalyst-eks-deploy-role \  
--policy-arn arn:aws:iam::111122223333:policy/codecatalyst-eks-deploy-policy
```

Où *arn:aws:iam::111122223333:policy/codecatalyst-eks-deploy-policy* est remplacé par l'ARN de la politique de déploiement que vous avez mentionnée précédemment.

4. Appuyez sur Entrée.
5. Entrez :

```
aws iam get-role \  
--role-name codecatalyst-eks-deploy-role
```

6. Appuyez sur Entrée.
7. Notez la "Arn" : valeur du rôle, par exemple `arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role`. Vous aurez besoin de cet ARN ultérieurement.

Vous avez maintenant créé des rôles de construction et de déploiement et noté leurs ARN.

## Étape 6 : Ajouter AWS des rôles à CodeCatalyst

Au cours de cette étape, vous ajoutez le rôle de construction (`codecatalyst-eks-build-role`) et le rôle de déploiement (`codecatalyst-eks-deploy-role`) au rôle Compte AWS que vous avez connecté à votre espace. Les rôles peuvent ainsi être utilisés dans votre flux de travail.

Pour ajouter des rôles de création et de déploiement à votre Compte AWS

1. Dans la CodeCatalyst console, accédez à votre espace.
2. En haut, choisissez Réglages.
3. Dans le volet de navigation, sélectionnez AWS Accounts. La liste des comptes s'affiche.
4. Dans la colonne du nom CodeCatalyst d'affichage d'Amazon, copiez le nom d'affichage de l'Compte AWS endroit où vous avez créé vos rôles de création et de déploiement. (Il peut s'agir d'un chiffre.) Vous aurez besoin de cette valeur ultérieurement, lors de la création de votre flux de travail.
5. Choisissez le nom d'affichage.
6. Choisissez Gérer les rôles depuis AWS la console de gestion.

La page Ajouter un rôle IAM à Amazon CodeCatalyst Space s'affiche. Il se peut que vous deviez vous connecter pour accéder à la page.

7. Sélectionnez Ajouter un rôle existant que vous avez créé dans IAM.

Une liste déroulante apparaît. La liste affiche les rôles de création et de déploiement, ainsi que tous les autres rôles IAM dotés d'une politique de confiance incluant les principes `codecatalyst-runner.amazonaws.com` et les principaux `codecatalyst.amazonaws.com` de service.

8. Dans la liste déroulante, ajoutez :
  - `codecatalyst-eks-build-role`
  - `codecatalyst-eks-deploy-role`

**Note**

Si vous le voyez `The security token included in the request is invalid`, c'est peut-être parce que vous ne disposez pas des autorisations appropriées. Pour résoudre ce problème, déconnectez-vous et reconnectez-vous avec le AWS compte que vous avez utilisé lors de la création de votre CodeCatalyst espace. AWS

9. Retournez à la CodeCatalyst console et actualisez la page.

Les rôles de création et de déploiement devraient désormais apparaître sous les rôles IAM.

Ces rôles peuvent désormais être utilisés dans les CodeCatalyst flux de travail.

## Étape 7 : mettez à jour le ConfigMap

Vous devez ajouter le rôle de déploiement que vous avez créé dans le ConfigMap fichier Kubernetes [Étape 5 : Création de AWS rôles](#) pour permettre à l'action Deploy to Kubernetes cluster (dans votre flux de travail) d'accéder à votre cluster et d'interagir avec celui-ci. Vous pouvez utiliser `eksctl` ou `kubectl` pour exécuter cette tâche.

Pour configurer le fichier Kubernetes ConfigMap à l'aide de `eksctl`

- Dans le terminal Dev Environment, entrez :

```
eksctl create iamidentitymapping --cluster codecatalyst-eks-cluster --  
arn arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role --group  
system:masters --username codecatalyst-eks-deploy-role --region us-west-2
```

Où :

- codecatalyst-eks-cluster* est remplacé par le nom du cluster Amazon EKS.
- arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role* est remplacé par l'ARN du rôle de déploiement que vous avez créé dans [Étape 5 : Création de AWS rôles](#)
- codecatalyst-eks-deploy-role* (à côté de `--username`) est remplacé par le nom du rôle de déploiement que vous avez créé dans [Étape 5 : Création de AWS rôles](#).

**Note**

Si vous avez décidé de ne pas créer de rôle de déploiement, *codecatalyst-eks-deploy-role* remplacez-le par le nom du CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle. Pour plus d'informations sur ce rôle, consultez [Étape 5 : Création de AWS rôles](#).

- *us-west-2* est remplacé par votre région.

Pour plus de détails sur cette commande, voir [Gérer les utilisateurs et les rôles IAM](#).

Un message similaire au suivant s'affiche :

```
2023-06-09 00:58:29 [#] checking arn arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role against entries in the auth ConfigMap
2023-06-09 00:58:29 [#] adding identity "arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role" to auth ConfigMap
```

Pour configurer le fichier Kubernetes ConfigMap à l'aide de kubectl

1. Dans le terminal Dev Environment, entrez :

```
kubectl edit configmap -n kube-system aws-auth
```

Le ConfigMap fichier apparaît à l'écran.

2. Ajoutez le texte en italique rouge :

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file
# will be
# reopened with the relevant failures.
#
apiVersion: v1
data:
  mapRoles: |
    - groups:
      - system:bootstrappers
      - system:nodes
```

```
rolearn: arn:aws:iam::111122223333:role/eksctl-codecatalyst-eks-cluster-n-
NodeInstanceRole-16BC456ME6YR5
username: system:node:{{EC2PrivateDNSName}}
- groups:
  - system:masters
  rolearn: arn:aws:iam::111122223333:role/codecatalyst-eks-deploy-role
  username: codecatalyst-eks-deploy-role
mapUsers: |
  []
kind: ConfigMap
metadata:
  creationTimestamp: "2023-06-08T19:04:39Z"
  managedFields:
  ...
```

Où :

- *arn:aws:iam : :111122223333:role/ codecatalyst-eks-deploy-role* est remplacé par l'ARN du rôle de déploiement que vous avez créé dans [Étape 5 : Création de AWS rôles](#)
- *codecatalyst-eks-deploy-role* (à côté de `username:`) est remplacé par le nom du rôle de déploiement que vous avez créé dans [Étape 5 : Création de AWS rôles](#).

#### Note

Si vous avez décidé de ne pas créer de rôle de déploiement, *codecatalyst-eks-deploy-role* remplacez-le par le nom du CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle. Pour plus d'informations sur ce rôle, consultez [Étape 5 : Création de AWS rôles](#).

Pour plus de détails, consultez la section [Activation de l'accès principal IAM à votre cluster](#) dans le guide de l'utilisateur Amazon EKS.

Vous avez maintenant accordé au rôle de déploiement, et par extension à l'action Deploy to Amazon EKS, `system:masters` des autorisations sur votre cluster Kubernetes.



## Étape 8 : créer et exécuter un flux de travail

Au cours de cette étape, vous créez un flux de travail qui prend vos fichiers source, les intègre dans une image Docker, puis déploie l'image dans des capsules d'arbres de votre cluster Amazon EKS.

Le flux de travail comprend les éléments de base suivants qui s'exécutent de manière séquentielle :

- Un déclencheur : ce déclencheur lance automatiquement l'exécution du flux de travail lorsque vous apportez une modification à votre référentiel source. Pour plus d'informations sur les déclencheurs, consultez [Utilisation de déclencheurs](#).
- Une action de génération (BuildBackend) — Au déclenchement, l'action crée l'image Docker à l'aide du Dockerfile et envoie l'image vers Amazon ECR. L'action de génération met également à jour les `$IMAGE_TAG` variables `$REPOSITORY_URI` et du `deployment.yaml` fichier avec les valeurs correctes, puis crée un artefact de sortie de ce fichier et de tous les autres éléments du Kubernetes dossier. Dans ce didacticiel, le seul fichier du Kubernetes dossier l'est `deployment.yaml`, mais vous pouvez inclure d'autres fichiers. L'artefact est utilisé comme entrée pour l'action de déploiement, qui est la suivante.

Pour plus d'informations sur l'action de génération, consultez [Création à l'aide de flux de travail dans CodeCatalyst](#).

- Une action de déploiement (DeployToEKS) : une fois l'action de génération terminée, l'action de déploiement recherche l'artefact de sortie généré par l'action de génération (Manifests) et trouve le `deployment.yaml` fichier qu'il contient. L'action suit ensuite les instructions du `deployment.yaml` fichier pour exécuter trois pods, chacun contenant un seul « Hello, World ! » Conteneur Docker : à l'intérieur de votre cluster Amazon EKS.

Pour créer un flux de travail

1. Accédez à la CodeCatalyst console.
2. Accédez à votre projet (`codecatalyst-eks-project`).
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez Créer un flux de travail.
5. Pour Référentiel source, choisissez `codecatalyst-eks-source-repository`.
6. Pour Branch, choisissez `main`.
7. Choisissez Créer.
8. Supprimez l'exemple de code YAML.

## 9. Ajoutez le code YAML suivant pour créer un nouveau fichier de définition de flux de travail :

### Note

Pour plus d'informations sur le fichier de définition du flux de travail, consultez [Référence de définition du flux de travail](#).

```
Name: codecatalyst-eks-workflow
SchemaVersion: 1.0

Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  BuildBackend:
    Identifiant: aws/build@v1
    Environment:
      Name: codecatalyst-eks-environment
    Connections:
      - Name: codecatalyst-account-connection
        Role: codecatalyst-eks-build-role
    Inputs:
      Sources:
        - WorkflowSource
      Variables:
        - Name: REPOSITORY_URI
          Value: 111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-eks-  
image-repo
        - Name: IMAGE_TAG
          Value: ${WorkflowSource.CommitId}
    Configuration:
      Steps:
        #pre_build:
          - Run: echo Logging in to Amazon ECR...
          - Run: aws --version
          - Run: aws ecr get-login-password --region us-west-2 | docker login --  
username AWS --password-stdin 111122223333.dkr.ecr.us-west-2.amazonaws.com
        #build:
          - Run: echo Build started on `date`
          - Run: echo Building the Docker image...
```

```


- Run: docker build -t $REPOSITORY_URI:latest .
- Run: docker tag $REPOSITORY_URI:latest $REPOSITORY_URI:$IMAGE_TAG
#post_build:
- Run: echo Build completed on `date`
- Run: echo Pushing the Docker images...
- Run: docker push $REPOSITORY_URI:latest
- Run: docker push $REPOSITORY_URI:$IMAGE_TAG
# Replace the variables in deployment.yaml
- Run: find Kubernetes/ -type f | xargs sed -i "s|\|$REPOSITORY_URI|
$REPOSITORY_URI|g"
- Run: find Kubernetes/ -type f | xargs sed -i "s|\|$IMAGE_TAG|$IMAGE_TAG|g"
- Run: cat Kubernetes/*
# The output artifact will be a zip file that contains Kubernetes manifest
files.
  Outputs:
    Artifacts:
      - Name: Manifests
        Files:
          - "Kubernetes/*"
  DeployToEKS:
  DependsOn:
    - BuildBackend
  Identifier: aws/kubernetes-deploy@v1
  Environment:
    Name: codecatalyst-eks-environment
    Connections:
      - Name: codecatalyst-account-connection
        Role: codecatalyst-eks-deploy-role
  Inputs:
    Artifacts:
      - Manifests
  Configuration:
    Namespace: default
    Region: us-west-2
    Cluster: codecatalyst-eks-cluster
    Manifests: Kubernetes/

```

Dans le code précédent, remplacez :

- Les deux instances portent *codecatalyst-eks-environment* le nom de l'environnement dans lequel vous l'avez créé [Prérequis](#).

- Dans les deux cas, *codecatalyst-account-connection* avec le nom d'affichage de la connexion à votre compte. Le nom d'affichage peut être un chiffre. Pour plus d'informations, consultez [Étape 6 : Ajouter AWS des rôles à CodeCatalyst](#).
- *codecatalyst-eks-build-role* avec le nom du rôle de build que vous avez créé dans [Étape 5 : Création de AWS rôles](#).
- *111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-eks-image-repo* (dans la Value : propriété) avec l'URI du référentiel Amazon ECR dans lequel vous l'avez créé. [Étape 3 : Création d'un référentiel d'images Amazon ECR](#)
- *111122223333.dkr.ecr.us-west-2.amazonaws.com* (dans la Run : aws ecr commande) avec l'URI du référentiel Amazon ECR sans le suffixe d'image (). / *codecatalyst-eks-image-repo*
- *codecatalyst-eks-deploy-role* avec le nom du rôle de déploiement que vous avez créé dans [Étape 5 : Création de AWS rôles](#).
- Les deux instances de *us-west-2* avec votre code de région. AWS Pour obtenir la liste des codes de région, consultez la section [Points de terminaison régionaux](#) dans le Références générales AWS.

 Note

Si vous avez décidé de ne pas créer de rôles de création et de déploiement, remplacez *codecatalyst-eks-build-role* et *codecatalyst-eks-deploy-role* par le nom du CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle. Pour plus d'informations sur ce rôle, consultez [Étape 5 : Création de AWS rôles](#).

10. (Facultatif) Choisissez Valider pour vous assurer que le code YAML est valide avant de valider.
11. Choisissez Commit (Valider).
12. Dans la boîte de dialogue du flux de travail de validation, entrez les informations suivantes :
  - a. Pour le message de validation, supprimez le texte et entrez :

**Add first workflow**
  - b. Pour Repository, choisissez *codecatalyst-eks-source-repository*.
  - c. Pour Nom de la branche, choisissez main.
  - d. Choisissez Commit (Valider).

Vous venez de créer un flux de travail. L'exécution d'un flux de travail démarre automatiquement en raison du déclencheur défini en haut du flux de travail. Plus précisément, lorsque vous avez validé (et transféré) le `workflow.yaml` fichier dans votre référentiel source, le déclencheur a lancé l'exécution du flux de travail.

Pour consulter la progression de l'exécution du flux de travail

1. Dans le volet de navigation de la CodeCatalyst console, choisissez CI/CD, puis Workflows.
2. Choisissez le flux de travail que vous venez de créer, `codecatalyst-eks-workflow`.
3. Choisissez BuildBackend de voir la progression de la construction.
4. Choisissez DeployToEKS pour voir la progression du déploiement.

Pour plus d'informations sur l'affichage des détails des exécutions, consultez [Afficher le statut et les détails de l'exécution du flux de travail](#).

Pour vérifier le déploiement

1. Ouvrez la console Amazon EC2 à l'adresse <https://console.aws.amazon.com/ec2/>.
2. Sur la gauche, vers le bas, choisissez Load Balancers.
3. Sélectionnez l'équilibreur de charge créé dans le cadre de votre déploiement Kubernetes. Si vous ne savez pas quel équilibreur de charge choisir, recherchez les balises suivantes sous l'onglet Tags :
  - `kubernetes.io/service-name`
  - `kubernetes.io/cluster/ekstutorialcluster`
4. Une fois le bon équilibreur de charge sélectionné, cliquez sur l'onglet Description.
5. Copiez et collez la valeur du nom DNS dans la barre d'adresse de votre navigateur.

Le « Hello, World ! » Une page Web apparaît dans votre navigateur, indiquant que vous avez correctement déployé votre application.

## Étape 9 : Apportez une modification à vos fichiers source

Dans cette section, vous allez apporter une modification au `index.html` fichier dans votre dépôt source. Cette modification oblige le flux de travail à créer une nouvelle image Docker, à l'étiqueter avec un ID de validation, à la transmettre à Amazon ECR et à la déployer sur Amazon ECS.

Pour modifier le fichier `index.html`

1. Accédez à votre environnement de développement.
2. À l'invite du terminal, accédez à votre référentiel source :

```
cd /projects/codecatalyst-eks-source-repository
```

3. Importez les dernières modifications apportées au flux de travail :

```
git pull
```

4. Ouvrir `codecatalyst-eks-source-repository/public-html/index.html`.
5. À la ligne 14, remplacez le `Hello, World!` texte par `Tutorial complete!`.
6. Ajoutez, validez et envoyez :

```
git add .  
git commit -m "update index.html title"  
git push
```

L'exécution d'un flux de travail démarre automatiquement.

7. (Facultatif) Entrez :

```
git show HEAD
```

Notez l'ID de validation associé à la `index.html` modification. Cet ID de validation sera associé à l'image Docker qui sera déployée par le flux de travail que vous venez de démarrer.

8. Suivez la progression du déploiement :
  - a. Dans le volet de navigation de la CodeCatalyst console, choisissez `CI/CD`, puis `Workflows`.
  - b. Choisissez `codecatalyst-eks-workflow` d'afficher la dernière édition.
  - c. Choisissez `BuildBackend`, et `DeployToEKS` pour voir la progression de l'exécution du flux de travail.

9. Vérifiez que votre application a été mise à jour, comme suit :
  - a. Ouvrez la console Amazon EC2 à l'adresse <https://console.aws.amazon.com/ec2/>.
  - b. Sur la gauche, vers le bas, choisissez Load Balancers.
  - c. Sélectionnez l'équilibreur de charge créé dans le cadre de votre déploiement Kubernetes.
  - d. Copiez et collez la valeur du nom DNS dans la barre d'adresse de votre navigateur.

Le « Tutoriel terminé ! » Une page Web apparaît dans votre navigateur, indiquant que vous avez déployé avec succès une nouvelle version de votre application.
10. (Facultatif) Dans AWS, passez à la console Amazon ECR et vérifiez que la nouvelle image Docker a été étiquetée avec l'ID de validation indiqué à l'étape 7 de cette procédure.

## Nettoyage

Vous devez nettoyer votre environnement afin de ne pas être facturé inutilement pour les ressources de stockage et de calcul utilisées dans ce didacticiel.

Pour nettoyer

1. Supprimez votre cluster :
  - Dans le terminal Dev Environment, entrez :

```
eksctl delete cluster --region=us-west-2 --name=codecatalyst-eks-cluster
```

Où :

- *us-west-2* est remplacé par votre région.
- *codecatalyst-eks-cluster* est remplacé par le nom du cluster que vous avez créé.

Après 5 à 10 minutes, le cluster et les ressources associées sont supprimés, y compris, mais sans s'y limiter, les AWS CloudFormation piles, les groupes de nœuds (dans Amazon EC2) et les équilibreurs de charge.

**⚠ Important**

Si la `eksctl delete cluster` commande ne fonctionne pas, vous devrez peut-être actualiser vos AWS informations d'identification ou vos `kubectl` informations d'identification. Si vous n'êtes pas sûr des informations d'identification à actualiser, AWS actualisez-les d'abord. Pour actualiser vos AWS informations d'identification, consultez [Comment corriger les erreurs « Impossible de localiser les informations d'identification » et ExpiredToken « » ?](#). Pour actualiser vos `kubectl` informations d'identification, consultez [Comment corriger les erreurs « Impossible de se connecter au serveur » ?](#).

2. Dans la AWS console, nettoyez comme suit :
  1. Dans Amazon ECR, supprimez `codecatalyst-eks-image-repo`.
  2. Dans IAM Identity Center, supprimez :
    - a. `codecatalyst-eks-user`
    - b. `codecatalyst-eks-permission-set`
  3. Dans IAM, supprimez :
    - `codecatalyst-eks-build-role`
    - `codecatalyst-eks-deploy-role`
    - `codecatalyst-eks-build-policy`
    - `codecatalyst-eks-deploy-policy`
3. Dans la CodeCatalyst console, nettoyez comme suit :
  1. Supprimer `codecatalyst-eks-workflow`.
  2. Supprimer `codecatalyst-eks-environment`.
  3. Supprimer `codecatalyst-eks-source-repository`.
  4. Supprimez votre environnement de développement.
  5. Supprimer `codecatalyst-eks-project`.

Dans ce didacticiel, vous avez appris à déployer une application sur un service Amazon EKS à l'aide d'un CodeCatalyst flux de travail et d'une action de cluster Deploy to Kubernetes.



## Ajout de l'action « Deploy AWS CloudFormation stack »

### Tip

Pour consulter un didacticiel expliquant comment utiliser l'action Deploy AWS CloudFormation stack, consultez [Tutoriel : Déployer une application sans serveur à l'aide de AWS CloudFormation](#).

Cette section explique comment ajouter l'action Deploy AWS CloudFormation stack à votre flux de travail. L'action crée un CloudFormation stack in sur la AWS base d'un modèle que vous fournissez. Le modèle peut être :

- AWS CloudFormation modèle — Pour plus d'informations, consultez la section [Utilisation des AWS CloudFormation modèles](#).
- AWS SAM modèle — Pour plus d'informations, voir la [spécification AWS Serverless Application Model \(AWS SAM\)](#).

### Note

Pour utiliser un AWS SAM modèle, vous devez d'abord emballer votre AWS SAM application à l'aide de l'[sam package](#) opération. Pour un didacticiel expliquant comment créer automatiquement cet emballage dans le cadre d'un CodeCatalyst flux de travail Amazon, consultez [Tutoriel : Déployer une application sans serveur à l'aide de AWS CloudFormation](#).

Si la pile existe déjà, l'action exécute l' CloudFormation [CreateChangeSet](#) opération, puis l'[ExecuteChangeSet](#) opération. L'action attend ensuite que les modifications soient déployées et se marque comme réussie ou comme ayant échoué, en fonction des résultats.

Utilisez l'action Deploy AWS CloudFormation stack si vous possédez déjà un AWS SAM modèle AWS CloudFormation OR contenant des ressources que vous souhaitez déployer, ou si vous prévoyez d'en générer une automatiquement dans le cadre d'une [action de création de flux de travail à l'aide d'outils](#) tels que AWS SAM et [AWS Cloud Development Kit \(AWS CDK\)](#).

Il n'existe aucune restriction quant au modèle que vous pouvez utiliser, quel que soit le modèle que vous pouvez créer CloudFormation ou que AWS SAM vous pouvez utiliser avec l'action Deploy AWS CloudFormation stack.

## Visual

Pour ajouter l'action « Deploy AWS CloudFormation stack » à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez Visual.
7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
8. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst.
9. Recherchez l'action Deploy AWS CloudFormation stack et effectuez l'une des opérations suivantes :

- Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.

Ou

- Choisissez Deploy AWS CloudFormation stack. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
    - (Facultatif) Choisissez Télécharger pour [afficher le code source de l'action](#).
    - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.
10. Dans les onglets Entrées et Configuration, complétez les champs en fonction de vos besoins. Pour obtenir une description de chaque champ, consultez le [Référence d'action « Deploy AWS CloudFormation Stack »](#). Cette référence fournit des informations détaillées sur chaque champ (et la valeur de propriété YAML correspondante) tel qu'il apparaît dans les éditeurs YAML et visuels.
  11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.

12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour ajouter l'action « Deploy AWS CloudFormation stack » à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
  2. Choisissez votre projet.
  3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
  4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
  5. Choisissez Modifier.
  6. Choisissez YAML.
  7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
  8. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst.
  9. Recherchez l'action Deploy AWS CloudFormation stack et effectuez l'une des opérations suivantes :
    - Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.
- Ou
- Choisissez Deploy AWS CloudFormation stack. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
    - (Facultatif) Choisissez Télécharger pour [afficher le code source de l'action](#).
    - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.
  10. Modifiez les propriétés du code YAML en fonction de vos besoins. Une explication de chaque propriété disponible est fournie dans le [Référence d'action « Deploy AWS CloudFormation Stack »](#).
  11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
  12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Variables produites par l'action « Deploy AWS CloudFormation stack »

Lorsque l'action Deploy AWS CloudFormation stack s'exécute, elle produit des variables que vous pouvez utiliser dans les actions de flux de travail suivantes. Pour plus de détails, voir [Variables d'action « Déployer la AWS CloudFormation pile »](#) dans le [Liste des variables prédéfinies](#).

## Définition de l'action « Deploy AWS CloudFormation Stack »

L'action Deploy AWS CloudFormation stack est définie comme un ensemble de propriétés YAML dans votre fichier de définition de flux de travail. Pour plus d'informations sur ces propriétés, consultez [Référence d'action « Deploy AWS CloudFormation Stack »](#) le [Référence de définition du flux de travail](#).

## Ajout de l'action « Déployer sur Amazon ECS »

### Tip

Pour consulter un didacticiel expliquant comment utiliser l'action Deploy to Amazon ECS, consultez [Tutoriel : Déploiement d'une application sur Amazon ECS](#).

### Tip

Pour un exemple pratique de l'action Deploy to Amazon ECS, créez un projet avec l'API Node.js avec AWS Fargate ou avec l'API Java avec AWS Fargate Blueprint. Pour plus d'informations, consultez [Création d'un projet à l'aide d'un plan](#).

Cette section décrit comment ajouter l'action Deploy to Amazon ECS à votre flux de travail. Cette action enregistre un fichier de [définition de tâche](#) que vous fournissez. Lors de l'enregistrement, la définition de tâche est ensuite instanciée par votre [service Amazon ECS](#) exécuté dans votre cluster [Amazon ECS](#). « Instancier une définition de tâche » équivaut au déploiement d'une application dans Amazon ECS.

Pour utiliser cette action, vous devez disposer d'un cluster, d'un service et d'un fichier de définition de tâche Amazon ECS existant.

Pour plus d'informations sur Amazon ECS, consultez le guide du développeur Amazon Elastic Container Service.

## Visual

Pour ajouter l'action « Déployer sur Amazon ECS » à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez Visual.
7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
8. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst.
9. Recherchez l'action Deploy to Amazon ECS, puis effectuez l'une des opérations suivantes :
  - Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.

Ou

- Choisissez Deploy to Amazon ECS. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
    - (Facultatif) Choisissez Télécharger pour [afficher le code source de l'action](#).
    - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.
10. Dans les onglets Entrées et Configuration, complétez les champs en fonction de vos besoins. Pour une description de chaque champ, consultez le [Référence d'action « Déployer sur Amazon ECS »](#). Cette référence fournit des informations détaillées sur chaque champ (et la valeur de propriété YAML correspondante) tel qu'il apparaît dans les éditeurs YAML et visuels.
  11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
  12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour ajouter l'action « Déployer sur Amazon ECS » à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez YAML.
7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
8. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst.
9. Recherchez l'action Deploy to Amazon ECS, puis effectuez l'une des opérations suivantes :
  - Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.

Ou

- Choisissez Deploy to Amazon ECS. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
    - (Facultatif) Choisissez Télécharger pour [afficher le code source de l'action](#).
    - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.
10. Modifiez les propriétés du code YAML en fonction de vos besoins. Une explication de chaque propriété disponible est fournie dans le [Référence d'action « Déployer sur Amazon ECS »](#).
  11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
  12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Variables produites par l'action « Déployer sur Amazon ECS »

Lorsque l'action Deploy to Amazon ECS s'exécute, elle produit des variables que vous pouvez utiliser dans les actions de flux de travail suivantes. Pour plus de détails, voir [Variables d'action « Déployer sur Amazon ECS »](#) dans le [Liste des variables prédéfinies](#).

## Définition de l'action « Déployer sur Amazon ECS »

L'action Deploy to Amazon ECS est définie comme un ensemble de propriétés YAML dans votre fichier de définition de flux de travail. Pour plus d'informations sur ces propriétés, consultez [Référence d'action « Déployer sur Amazon ECS »](#) le [Référence de définition du flux de travail](#).

## Ajout de l'action « Déployer vers le cluster Kubernetes »

### Tip

Pour un didacticiel expliquant comment utiliser l'action Deploy to Kubernetes cluster, voir. [Tutoriel : Déploiement d'une application sur Amazon EKS](#)

Cette section décrit comment ajouter l'action Déployer vers le cluster Kubernetes à votre flux de travail. Cette action déploie votre application sur un cluster Kubernetes que vous avez configuré dans Amazon Elastic Kubernetes Service (EKS) à l'aide d'un ou de plusieurs fichiers manifestes Kubernetes que vous fournissez. Pour un exemple de manifeste, voir [deployment.yaml](#) dans [Tutoriel : Déploiement d'une application sur Amazon EKS](#).

[Pour plus d'informations sur Kubernetes, consultez la documentation Kubernetes.](#)

Pour plus d'informations sur Amazon EKS, consultez [Qu'est-ce qu'Amazon EKS ?](#) dans le guide de l'utilisateur Amazon EKS.

## Comment ça marche

Le cluster Deploy to Kubernetes fonctionne comme suit :

1. Au moment de l'exécution, l'action installe l'utilitaire Kubernetes sur la machine de CodeCatalyst calcul sur laquelle l'action est exécutée. L'action est configurée `kubectl` pour pointer vers le cluster Amazon EKS que vous avez fourni lors de la configuration de l'action. L'utilitaire est nécessaire pour exécuter la `kubectl apply` commande, ensuite.
2. L'action exécute la `kubectl apply -f my-manifest.yaml` commande, qui exécute les instructions du fichier `my-manifest.yaml` pour déployer votre application sous la forme d'un ensemble de conteneurs et de pods dans le cluster configuré. Pour plus d'informations sur cette commande, consultez la rubrique [kubectl apply](#) dans la documentation de référence de Kubernetes.

## Prérequis

Pour utiliser cette action, vous devez préparer les éléments suivants :

### Tip

Pour configurer rapidement ces prérequis, suivez les instructions figurant dans [Tutoriel : Déploiement d'une application sur Amazon EKS](#).

- Un cluster Kubernetes dans Amazon EKS. Pour plus d'informations sur les clusters, consultez la section [relative aux clusters Amazon EKS](#) dans le guide de l'utilisateur Amazon EKS.
- Au moins un Dockerfile qui décrit comment assembler votre application dans une image Docker. Pour plus d'informations sur Dockerfiles, consultez la référence [Dockerfile](#).
- Au moins un fichier manifeste Kubernetes, appelé fichier de configuration ou configuration dans la documentation de Kubernetes. Pour plus d'informations, consultez [la section Gestion des ressources](#) dans la documentation de Kubernetes.
- Rôle IAM qui permet à l'action Deploy to Kubernetes cluster d'accéder à votre cluster Amazon EKS et d'interagir avec celui-ci. Pour plus d'informations, consultez la rubrique [Role](#) dans le manuel [Référence d'action « Déployer vers un cluster Kubernetes »](#).

Après avoir créé ce rôle, vous devez l'ajouter à :

- Votre fichier Kubernetes ConfigMap . Pour savoir comment ajouter un rôle à un ConfigMap fichier, consultez la section [Activation de l'accès principal IAM à votre cluster](#) dans le guide de l'utilisateur Amazon EKS.
- CodeCatalyst. Pour savoir comment ajouter un rôle IAM à CodeCatalyst, consultez [Ajouter des rôles IAM aux connexions aux comptes](#).
- Un CodeCatalyst espace, un projet et un environnement. L'espace et l'environnement doivent tous deux être connectés au AWS compte sur lequel vous allez déployer votre application. Pour plus d'informations, consultez [Création d'un espace prenant en charge les utilisateurs de AWS Builder ID](#), [Création d'un projet vide dans Amazon CodeCatalyst](#) et [Utilisation des environnements](#).
- Un référentiel source pris en charge par CodeCatalyst. Le référentiel stocke les fichiers source de votre application, les Dockerfiles et les manifestes Kubernetes. Pour plus d'informations, consultez [Référentiels de sources dans CodeCatalyst](#).



## Ajout de l'action « Déployer vers un cluster »

Suivez les instructions suivantes pour ajouter l'action Déployer vers un cluster à votre flux de travail.

### Visual

Pour ajouter l'action « Déployer vers le cluster Kubernetes » à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez Visual.
7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
8. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst.
9. Recherchez l'action Déployer vers le cluster Kubernetes, puis effectuez l'une des opérations suivantes :

- Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.

Ou

- Choisissez Déployer vers un cluster Kubernetes. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
  - (Facultatif) Choisissez Télécharger pour [afficher le code source de l'action](#).
  - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.

10. Dans les onglets Entrées et Configuration, complétez les champs en fonction de vos besoins. Pour obtenir une description de chaque champ, consultez le [Référence d'action « Déployer vers un cluster Kubernetes »](#). Cette référence fournit des informations détaillées sur chaque champ (et la valeur de propriété YAML correspondante) tel qu'il apparaît dans les éditeurs YAML et visuels.
11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.

12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour ajouter l'action « Déployer vers le cluster Kubernetes » à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
  2. Choisissez votre projet.
  3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
  4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
  5. Choisissez Modifier.
  6. Choisissez YAML.
  7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
  8. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst.
  9. Recherchez l'action Déployer vers le cluster Kubernetes, puis effectuez l'une des opérations suivantes :
    - Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.
- Ou
- Choisissez Déployer vers un cluster Kubernetes. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
    - (Facultatif) Choisissez Télécharger pour [afficher le code source de l'action](#).
    - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.
  10. Modifiez les propriétés du code YAML en fonction de vos besoins. Une explication de chaque propriété disponible est fournie dans le [Référence d'action « Déployer vers un cluster Kubernetes »](#).
  11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
  12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Variables produites par l'action « Déployer vers le cluster Kubernetes »

Lorsque l'action Déployer vers le cluster Kubernetes s'exécute, elle produit des variables que vous pouvez utiliser dans les actions de flux de travail suivantes. Pour plus de détails, voir [Variables d'action « Déployer vers un cluster Kubernetes »](#) dans le [Liste des variables prédéfinies](#).

## Définition de l'action « Déployer vers un cluster Kubernetes »

L'action Déployer vers le cluster Kubernetes est définie comme un ensemble de propriétés YAML dans votre fichier de définition de flux de travail. Pour plus d'informations sur ces propriétés, consultez [Référence d'action « Déployer vers un cluster Kubernetes »](#) le [Référence de définition du flux de travail](#).

## Ajout de l'action « AWS CDK déployer »

Cette section décrit comment ajouter l'action de AWS CDK déploiement à votre flux de travail. L'action de AWS CDK déploiement synthétise et déploie votre AWS Cloud Development Kit (AWS CDK) application dans. AWS Si votre application existe déjà dans AWS, l'action la met à jour si nécessaire.

Pour des informations générales sur la création d'applications utilisant le AWS CDK, voir [Qu'est-ce que le AWS CDK ?](#) dans le Guide AWS Cloud Development Kit (AWS CDK) du développeur.

### Rubriques

- [Quand utiliser cette action](#)
- [Comment ça marche](#)
- [Versions de la CLI CDK utilisées par l'action « AWS CDK déploiement »](#)
- [Combien de piles l'action peut-elle déployer ?](#)
- [Prérequis](#)
- [Exemple de flux de travail](#)
- [Ajout de l'action « AWS CDK déployer »](#)
- [Variables produites par l'action « AWS CDK déployer »](#)
- [définition de l'action « AWS CDK déployer »](#)

## Quand utiliser cette action

Utilisez cette action si vous avez développé une application à l'aide de AWS CDK, et si vous souhaitez maintenant la déployer automatiquement dans le cadre d'un flux de travail automatisé d'intégration et de livraison continues (CI/CD). Par exemple, vous souhaitez peut-être déployer votre AWS CDK application automatiquement chaque fois que quelqu'un fusionne une pull request liée à la source de votre AWS CDK application.

## Comment ça marche

Le AWS CDK déploiement fonctionne comme suit :

1. [Au moment de l'exécution, si vous avez spécifié la version 1.0.12 ou une version antérieure de l'action, celle-ci télécharge la dernière CLI CDK \(également appelée AWS CDK Toolkit\) sur l'image de construction. CodeCatalyst](#)

Si vous avez spécifié la version 1.0.13 ou ultérieure, l'action est fournie avec une [version spécifique](#) de la CLI CDK, de sorte qu'aucun téléchargement n'a lieu.

2. L'action utilise la CLI CDK pour exécuter la `cdk deploy` commande. Cette commande synthétise et déploie votre AWS CDK application dans. AWS Pour plus d'informations sur cette commande, consultez la rubrique [AWS CDK Toolkit \(commande cdk\)](#) du Guide du AWS Cloud Development Kit (AWS CDK) développeur.

## Versions de la CLI CDK utilisées par l'action « AWS CDK déploiement »

Le tableau suivant indique quelle version de la CLI CDK est utilisée par défaut par les différentes versions de l'action de AWS CDK déploiement.

### Note

Vous pouvez peut-être remplacer la valeur par défaut. Pour de plus amples informations, veuillez consulter [CdkCliVersion](#) dans le [référence d'action « AWS CDK déployer »](#).

version d'action « AWS CDK déployer »	AWS CDK Version de la CLI
1.0.0 — 1.0.12	dernières

version d'action « AWS CDK déployer »	AWS CDK Version de la CLI
1.0.13 ou version ultérieure	2,99,1

## Combien de piles l'action peut-elle déployer ?

Le AWS CDK déploiement ne peut déployer qu'une seule pile. Si votre AWS CDK application est composée de plusieurs piles, vous devez créer une pile parent avec des piles imbriquées et déployer le parent à l'aide de cette action.

## Prérequis

Avant de pouvoir utiliser l'action de AWS CDK déploiement, effectuez les tâches suivantes :

1. Préparez une AWS CDK application. Vous pouvez écrire votre AWS CDK application en utilisant AWS CDK v1 ou v2, dans n'importe quel langage de programmation pris en charge par le AWS CDK. Assurez-vous que les fichiers de votre AWS CDK application sont disponibles dans :

- Un [référentiel CodeCatalyst source](#), ou
- [Artefact CodeCatalyst de sortie](#) généré par une autre action du flux de travail

2. Bootstrap votre AWS environnement. Pour démarrer, vous pouvez :

- Utilisez l'une des méthodes décrites dans la section [Comment démarrer du](#) manuel du AWS Cloud Development Kit (AWS CDK) développeur.
- Utilisez l'action AWS CDK bootstrap. Vous pouvez ajouter cette action dans le même flux de travail que celui de votre AWS CDK déploiement ou dans un autre. Assurez-vous simplement que l'action bootstrap s'exécute au moins une fois avant d'exécuter l'action de AWS CDK déploiement afin que les ressources nécessaires soient en place. Pour plus d'informations sur l'action AWS CDK bootstrap, consultez [Ajout de l'action « AWS CDK bootstrap »](#).

Pour plus d'informations sur le bootstrapping, voir [Bootstrapping](#) dans le manuel du développeur.AWS Cloud Development Kit (AWS CDK)

## Exemple de flux de travail

L'exemple de flux de travail suivant inclut l'action de AWS CDK déploiement, ainsi que l'action de AWS CDK démarrage. Le flux de travail comprend les éléments de base suivants qui s'exécutent de manière séquentielle :

**Note**

L'exemple de flux de travail suivant est fourni à titre d'illustration et ne fonctionnera pas sans configuration supplémentaire. Il est destiné à vous donner un exemple de ce à quoi peut ressembler un flux de travail lorsqu'il est configuré avec l'action de AWS CDK déploiement.

- Un déclencheur : ce déclencheur lance automatiquement l'exécution du flux de travail lorsque vous apportez une modification à votre référentiel source. Ce référentiel contient votre AWS CDK application. Pour plus d'informations sur les déclencheurs, consultez [Utilisation de déclencheurs](#).
- Une action AWS CDK bootstrap (CDKBootstrap) — Au déclenchement, l'action déploie la pile CDKToolkit bootstrap dans. AWS Si la CDKToolkit pile existe déjà dans l'environnement, elle sera mise à niveau si nécessaire ; dans le cas contraire, rien ne se passe et l'action est marquée comme réussie.
- Une action de AWS CDK déploiement (AWS CDK Deploy) — Une fois l'action de AWS CDK démarrage terminée, l'action de AWS CDK déploiement synthétise le code de votre AWS CDK application dans un AWS CloudFormation modèle et déploie la pile définie dans le modèle dans. AWS

```
Name: codecatalyst-cdk-deploy-workflow
SchemaVersion: 1.0

Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  CDKBootstrap:
    Identifiant: aws/cdk-bootstrap@v1
    Inputs:
      Sources:
        - WorkflowSource
  Environment:
    Name: codecatalyst-cdk-deploy-environment
    Connections:
      - Name: codecatalyst-account-connection
        Role: codecatalyst-cdk-bootstrap-role
  Configuration:
    Region: us-west-2
```

```
CDKDeploy:
  Identifiant: aws/cdk-deploy@v1
  DependsOn:
    - CDKBootstrap
  Environment:
    Name: codecatalyst-cdk-deploy-environment
  Connections:
    - Name: codecatalyst-account-connection
      Role: codecatalyst-cdk-deploy-role
  Inputs:
    Sources:
      - WorkflowSource
  Configuration:
    StackName: my-app-stack
    Region: us-west-2
```

## Ajout de l'action « AWS CDK déployer »

Suivez les instructions suivantes pour ajouter l'action de AWS CDK déploiement à votre flux de travail.

### Prérequis

Avant de commencer, assurez-vous d'avoir effectué les tâches décrites dans [Prérequis](#).

### Visual


Pour ajouter l'action « AWS CDK déployer » à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez Visual.
7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
8. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst.

9. Recherchez l'action de AWS CDK déploiement, puis effectuez l'une des opérations suivantes :
  - Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.

Ou

  - Choisissez AWS CDK Deploy. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
    - (Facultatif) Choisissez Télécharger pour [afficher le code source de l'action](#).
    - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.
10. Dans les onglets Entrées et Configuration, complétez les champs en fonction de vos besoins. Pour obtenir une description de chaque champ, consultez le [référence d'action « AWS CDK déployer »](#). Cette référence fournit des informations détaillées sur chaque champ (et la valeur de propriété YAML correspondante) tel qu'il apparaît dans les éditeurs YAML et visuels.
11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

 Note

Si votre action de AWS CDK déploiement échoue avec une `npm install` erreur, consultez [Comment corriger les erreurs « npm install » ?](#) pour plus d'informations sur la façon de corriger l'erreur.


## YAML

Pour ajouter l'action « AWS CDK déployer » à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.



5. Choisissez Modifier.
  6. Choisissez YAML.
  7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
  8. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst.
  9. Recherchez l'action de AWS CDK déploiement, puis effectuez l'une des opérations suivantes :
    - Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.
- Ou
- Choisissez AWS CDK Deploy. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
    - (Facultatif) Choisissez Télécharger pour [afficher le code source de l'action](#).
    - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.
  10. Modifiez les propriétés du code YAML en fonction de vos besoins. Une explication de chaque propriété disponible est fournie dans le [référence d'action « AWS CDK déployer »](#).
  11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
  12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

 Note

Si votre action de AWS CDK déploiement échoue avec une `npm install` erreur, consultez [Comment corriger les erreurs « npm install » ?](#) pour plus d'informations sur la façon de corriger l'erreur.

## Variables produites par l'action « AWS CDK déployer »

Lorsque l'action de AWS CDK déploiement s'exécute, elle produit des variables que vous pouvez utiliser dans les actions de flux de travail suivantes. Pour plus de détails, voir [variables d'action « AWS CDK déployer »](#) dans le [Liste des variables prédéfinies](#).

## définition de l'action « AWS CDK déployer »

L'action de AWS CDK déploiement est définie comme un ensemble de propriétés YAML dans votre fichier de définition de flux de travail. Pour plus d'informations sur ces propriétés, consultez [référence d'action « AWS CDK déployer »](#) le [Référence de définition du flux de travail](#).

## Utilisation de déploiements

Cette section explique comment gérer, surveiller et configurer les déploiements dans Amazon CodeCatalyst.

### Rubriques

- [Déploiement d'une application ou d'une ressource](#)
- [Réexécution d'un déploiement](#)
- [Affichage de l'état du déploiement, des validations et des pull requests](#)
- [Afficher les journaux de déploiement](#)
- [Configuration des annulations](#)
- [Afficher l'URL de l'application déployée](#)
- [Supprimer une cible de déploiement](#)

## Déploiement d'une application ou d'une ressource

Pour déployer une application ou une ressource à l'aide d'Amazon CodeCatalyst, vous devez utiliser un CodeCatalyst flux de travail. Ce flux de travail doit inclure au moins une action permettant de déployer quelque chose. Par exemple, si vous souhaitez déployer une AWS CloudFormation pile, vous devez créer un flux de travail incluant l'action Déployer une AWS CloudFormation pile.

Pour de plus amples informations sur les flux de travail, veuillez consulter [Créez, testez et déployez avec des flux de travail dans CodeCatalyst](#). Pour plus d'informations sur les actions, consultez [Utilisation des actions](#).

## Réexécution d'un déploiement

Pour réexécuter un déploiement dans Amazon CodeCatalyst, vous devez réexécuter le flux de travail associé. Pour obtenir des instructions, consultez [Démarrage d'une exécution de flux de travail](#).

## Affichage de l'état du déploiement, des validations et des pull requests

Vous pouvez consulter les informations suivantes concernant un déploiement sur Amazon CodeCatalyst :

- Activité de déploiement, y compris l'état du déploiement, l'heure de début, l'heure de fin, l'historique et la durée des événements.
- Nom de la pile Région AWS, heure de la dernière mise à jour et flux de travail associés.
- Validations et pull requests.
- Informations spécifiques à l'action, par exemple les CloudFormation événements et les résultats.

Vous pouvez consulter les informations de déploiement à partir d'un [flux de travail](#), d'un [environnement](#) ou d'une [action](#) de flux de travail.

Pour afficher les informations de déploiement à partir d'un flux de travail

- Accédez au flux de travail qui a déployé votre application. Pour obtenir des instructions, veuillez consulter [Afficher le statut et les détails de l'exécution du flux de travail](#).

Pour afficher les informations de déploiement à partir d'un environnement

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Environments.
4. Choisissez l'environnement dans lequel votre stack a été déployé, par exemple, Production.
5. Choisissez Activité de déploiement pour afficher l'historique des déploiements de vos piles, l'état des déploiements (par exemple, SUCCÈS ou ÉCHEC) et d'autres informations relatives au déploiement.
6. Choisissez la cible de déploiement pour afficher les informations relatives aux piles, aux clusters ou aux autres cibles déployées dans l'environnement. Vous pouvez consulter des informations telles que le nom de la pile, la région, le fournisseur et l'identifiant.

Pour afficher les informations de déploiement à partir d'une action

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.

3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Dans le diagramme de flux de travail, choisissez l'action de flux de travail qui a déployé votre application. Par exemple, vous pouvez choisir DeployCloudFormationStack.
6. Consultez le contenu du volet droit pour obtenir des informations de déploiement spécifiques à l'action.

## Afficher les journaux de déploiement

Vous pouvez consulter les journaux relatifs à des actions de déploiement spécifiques pour résoudre des problèmes sur Amazon CodeCatalyst.

Vous pouvez consulter les journaux à partir d'un [flux](#) de travail ou d'un [environnement](#).

Pour consulter les journaux d'une action de déploiement à partir d'un flux de travail

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Runs.
6. Choisissez l'exécution du flux de travail qui a déployé votre application.
7. Dans le diagramme du flux de travail, choisissez l'action dont vous souhaitez consulter les journaux.
8. Choisissez l'onglet Journaux et développez les sections pour afficher les messages du journal.
9. Pour afficher d'autres journaux, choisissez l'onglet Résumé, puis choisissez Afficher dans CloudFormation (s'il est disponible) pour y afficher d'autres journaux. Vous devrez peut-être vous connecter à AWS.

Pour consulter les journaux d'une action de déploiement à partir d'un environnement

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.

3. Dans le volet de navigation, choisissez CI/CD, puis Environments.
4. Choisissez l'environnement dans lequel votre application a été déployée.
5. Dans Activité de déploiement, recherchez la colonne ID d'exécution du flux de travail et choisissez l'exécution du flux de travail qui a déployé votre stack.
6. Dans le diagramme du flux de travail, choisissez l'action dont vous souhaitez consulter les journaux.
7. Choisissez l'onglet Journaux et développez les sections pour afficher les messages du journal.
8. Pour afficher d'autres journaux, choisissez l'onglet Résumé, puis choisissez Afficher dans CloudFormation (s'il est disponible) pour y afficher d'autres journaux. Vous devrez peut-être vous connecter à AWS.

## Configuration des annulations

Par défaut, si l'action Deploy AWS CloudFormation stack échoue, elle ramènera la pile au dernier état stable connu. AWS CloudFormation Vous pouvez modifier le comportement afin que les annulations se produisent non seulement en cas d'échec de l'action, mais également lorsqu'une CloudWatch alarme Amazon spécifiée se produit. Pour plus d'informations sur les CloudWatch alarmes, consultez la section [Utilisation des CloudWatch alarmes Amazon](#) dans le guide de CloudWatch l'utilisateur Amazon.

Vous pouvez également modifier le comportement par défaut afin de CloudFormation ne pas annuler la pile en cas d'échec de l'action.

Suivez les instructions ci-dessous pour configurer les annulations.

### Note

Vous ne pouvez pas démarrer une restauration manuellement.

## Visual

### Avant de commencer

1. Assurez-vous de disposer d'un [flux de travail](#) qui inclut une action Deploy AWS CloudFormation stack fonctionnelle. Pour plus d'informations, consultez [Ajout de l'action « Deploy AWS CloudFormation stack »](#).


2. Dans le rôle spécifié dans le champ Stack role - facultatif de l'action Deploy AWS CloudFormation stack, assurez-vous d'inclure l'CloudWatchFullAccess autorisation. Pour plus d'informations sur la création de ce rôle avec les autorisations appropriées, consultez [Étape 2 : créer des AWS rôles](#).

Pour configurer les alarmes d'annulation pour l'action « Deploy AWS CloudFormation stack »

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom d'un flux de travail qui inclut l'action Deploy AWS CloudFormation stack. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez Visual.
7. Choisissez votre action Deploy AWS CloudFormation stack.
8. Dans le volet de détails, sélectionnez Configuration.
9. En bas de la page, développez la section Avancé.
10. Sous Surveiller les ARN des alarmes, choisissez Ajouter une alarme.
11. Entrez les informations dans les champs suivants.

- ARN de l'alarme

Spécifiez le nom de ressource Amazon (ARN) d'une CloudWatch alarme Amazon à utiliser comme déclencheur d'annulation. Par exemple, `arn:aws:cloudwatch::123456789012:alarm/MyAlarm`. Vous pouvez disposer d'un maximum de cinq déclencheurs de restauration.

 Note

Si vous spécifiez un ARN CloudWatch d'alarme, vous devez également configurer des autorisations supplémentaires pour permettre à l'action d'y accéder CloudWatch. Pour plus d'informations, consultez [Configuration des annulations](#).

- Temps de surveillance

Spécifiez une durée, comprise entre 0 et 180 minutes, pendant laquelle CloudFormation les alarmes spécifiées seront surveillées. La surveillance commence une fois que toutes les ressources de la pile ont été déployées. Si l'alarme se produit dans le délai de surveillance spécifié, le déploiement échoue et CloudFormation annule l'ensemble du fonctionnement de la pile.

Par défaut : 0. CloudFormation surveille uniquement les alarmes pendant le déploiement des ressources de la pile, et non après.

## YAML

Pour configurer les déclencheurs d'annulation pour l'action « Deploy AWS CloudFormation stack »

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom d'un flux de travail qui inclut l'action Deploy AWS CloudFormation stack. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez YAML.
7. Ajoutez les `monitor-timeout-in-minutes` propriétés `monitor-alarm-arns` et dans le code YAML pour ajouter des déclencheurs de restauration. Pour une explication de chaque propriété, voir [Référence d'action « Deploy AWS CloudFormation Stack »](#).
8. Dans le rôle spécifié dans la `role-arn` propriété de l'action Deploy AWS CloudFormation stack, assurez-vous d'inclure l'`CloudWatchFullAccess` autorisation. Pour plus d'informations sur la création de ce rôle avec les autorisations appropriées, consultez [Étape 2 : créer des AWS rôles](#).

## Visual

Pour désactiver les annulations pour l'action « Deploy AWS CloudFormation stack »

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.

3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom d'un flux de travail qui inclut l'action Deploy AWS CloudFormation stack. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez Visual.
7. Choisissez votre action Deploy AWS CloudFormation stack.
8. Dans le volet de détails, sélectionnez Configuration.
9. En bas de la page, développez la section Avancé.
10. Activez Désactiver la restauration.

## YAML

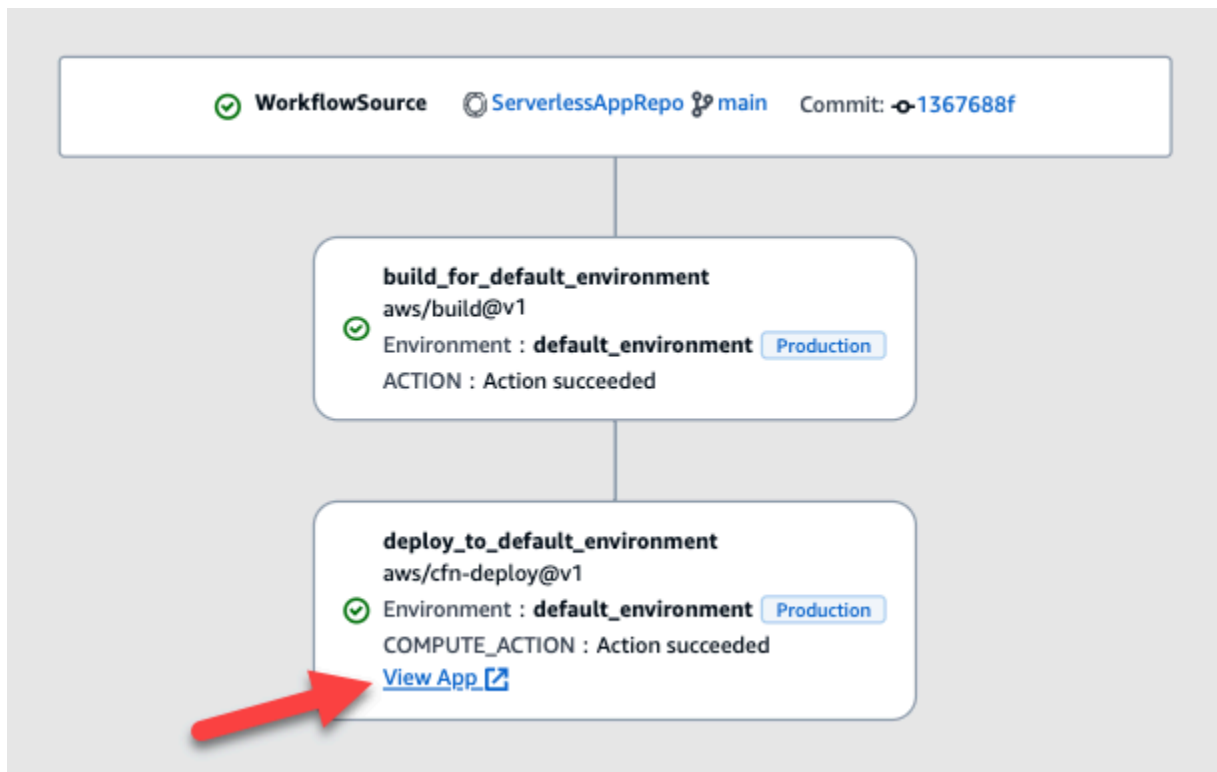
Pour désactiver les annulations pour l'action « Deploy AWS CloudFormation stack »

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom d'un flux de travail qui inclut l'action Deploy AWS CloudFormation stack. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez YAML.
7. Ajoutez la `disable-rollback: 1` propriété dans le code YAML pour arrêter les annulations. Pour une explication de cette propriété, voir [Référence d'action « Deploy AWS CloudFormation Stack »](#).

## Afficher l'URL de l'application déployée

Si votre flux de travail déploie une application, vous pouvez configurer Amazon CodeCatalyst pour afficher l'URL de l'application sous forme de lien cliquable. Ce lien apparaît dans la CodeCatalyst console, dans l'action qui l'a déployé. Le diagramme de flux de travail suivant montre l'URL de l'application View qui apparaît au bas d'une action.





En rendant cette URL cliquable dans la CodeCatalyst console, vous pouvez vérifier rapidement le déploiement de votre application.

#### Note

L'URL de l'application n'est pas prise en charge par l'action Deploy to Amazon ECS.

Pour activer cette fonctionnalité, ajoutez une variable de sortie à votre action dont le nom contient `appurl`, ou `endpointurl`. Vous pouvez utiliser un nom avec ou sans tiret (-), trait de soulignement (\_) ou espace ( ). La chaîne ne distingue pas les majuscules et minuscules. Définissez la valeur de la variable sur l'`httpsURL http` ou l'URL de votre application déployée.

#### Note

Si vous mettez à jour une variable de sortie existante pour inclure la `endpoint url` chaîne ou `app url`, mettez à jour toutes les références à cette variable pour utiliser le nouveau nom de variable.

Pour connaître les étapes détaillées, reportez-vous à l'une des procédures suivantes :

- [Pour afficher l'URL de l'application dans l'action « AWS CDK déployer »](#)
- [Pour afficher l'URL de l'application dans l'action « Deploy AWS CloudFormation stack »](#)
- [Pour afficher l'URL de l'application dans toutes les autres actions](#)

Lorsque vous avez terminé de configurer l'URL, vérifiez qu'elle s'affiche comme prévu en suivant ces instructions :

- [Pour vérifier que l'URL de l'application a été ajoutée](#)

Pour afficher l'URL de l'application dans l'action « AWS CDK déployer »

1. Si vous utilisez l'action de AWS CDK déploiement, ajoutez une `CfnOutput` construction (qui est une paire clé-valeur) dans le code de votre AWS CDK application :
  - Le nom de la clé doit contenir `rappurl`, ou `endpointurl`, avec ou sans tiret (-), un trait de soulignement (\_) ou un espace ( ). La chaîne ne distingue pas les majuscules et minuscules.
  - La valeur doit être l'URL `http` ou l'URL de votre application déployée.

Par exemple, votre AWS CDK code peut ressembler à ceci :

```
import { Duration, Stack, StackProps, CfnOutput, RemovalPolicy } from 'aws-cdk-lib';
import * as dynamodb from 'aws-cdk-lib/aws-dynamodb';
import * as s3 from 'aws-cdk-lib/aws-s3';
import { Construct } from 'constructs';
import * as cdk from 'aws-cdk-lib';
export class HelloCdkStack extends Stack {
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);
    const bucket = new s3.Bucket(this, 'my-bucket', {
      removalPolicy: RemovalPolicy.DESTROY,
    });
    new CfnOutput(this, 'APP-URL', {
      value: https://mycompany.myapp.com,
      description: 'The URL of the deployed application',
      exportName: 'myApp',
    });
    ...
  }
}
```

Pour plus d'informations sur la CfnOutput construction, consultez la section [interface CfnOutputProps](#) dans la référence de l'AWS Cloud Development Kit (AWS CDK) API.

2. Enregistrez et validez votre code.
3. Passez à [Pour vérifier que l'URL de l'application a été ajoutée.](#)

Pour afficher l'URL de l'application dans l'action « Deploy AWS CloudFormation stack »

1. Si vous utilisez l'action Deploy AWS CloudFormation stack, ajoutez une sortie à la Outputs section de votre CloudFormation modèle ou de votre AWS SAM modèle présentant les caractéristiques suivantes :
  - La clé (également appelée ID logique) doit contenir ou `appurl` `endpointurl` non un tiret (-), un trait de soulignement (\_) ou un espace ( ). La chaîne ne distingue pas les majuscules et minuscules.
  - La valeur doit être l'HTTPSURL http ou l'URL de votre application déployée.

Par exemple, votre CloudFormation modèle peut ressembler à ceci :

```
"Outputs" : {  
  "APP-URL" : {  
    "Description" : "The URL of the deployed app",  
    "Value" : "https://mycompany.myapp.com",  
    "Export" : {  
      "Name" : "My App"  
    }  
  }  
}
```

Pour plus d'informations sur CloudFormation les sorties, voir [Sorties](#) dans le guide de AWS CloudFormation l'utilisateur.

2. Enregistrez et validez votre code.
3. Passez à [Pour vérifier que l'URL de l'application a été ajoutée.](#)

## Pour afficher l'URL de l'application dans toutes les autres actions

Si vous utilisez une autre action pour déployer votre application, telle que l'action de création ou GitHub les actions, procédez comme suit pour afficher l'URL de l'application.

1. Définissez une variable d'environnement dans la Steps section Inputs ou de l'action du fichier de définition du flux de travail. La variable doit présenter les caractéristiques suivantes :
  - Ils name doivent contenir ou `appurlendpointurl`, avec ou sans tiret (-), soulignement (\_) ou espace ( ). La chaîne ne distingue pas les majuscules et minuscules.
  - La valeur doit être l'HTTPSURL http ou l'URL de votre application déployée.

Par exemple, une action de génération peut ressembler à ceci :

```
Build-action:
  Identifieur: aws/build@v1
  Inputs:
    Variables:
      - Name: APP-URL
        Value: https://mycompany.myapp.com
```

... ou ceci :

```
Actions:
  Build:
    Identifieur: aws/build@v1
    Configuration:
      Steps:
        - Run: APP-URL=https://mycompany.myapp.com
```

Pour plus d'informations sur la définition des variables d'environnement, consultez [Définition d'une variable](#).

2. Exportez la variable.

Par exemple, votre action de création peut ressembler à ceci :

```
Build-action:
  ...
  Outputs:
    Variables:
```

**- APP-URL**

Pour plus d'informations sur l'exportation de variables, consultez [Exporter une variable afin que d'autres actions puissent l'utiliser](#).

3. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
4. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.
5. Passez à [Pour vérifier que l'URL de l'application a été ajoutée](#).

Pour vérifier que l'URL de l'application a été ajoutée

- Lancez l'exécution d'un flux de travail s'il n'a pas démarré automatiquement. Lors de la nouvelle exécution, l'URL de l'application doit être affichée sous forme de lien cliquable dans son diagramme de flux de travail. Pour plus d'informations sur le démarrage de courses, consultez [Démarrage d'une exécution de flux de travail](#).

## Supprimer une cible de déploiement

Vous pouvez supprimer une cible de déploiement telle qu'un cluster ou une AWS CloudFormation pile Amazon ECS depuis la page Environnements de la CodeCatalyst console.

### Important

Lorsque vous supprimez une cible de déploiement, elle est supprimée de la CodeCatalyst console, mais elle reste disponible dans le AWS service qui l'héberge (si elle existe toujours).

Envisagez de supprimer une cible de déploiement si celle-ci est devenue obsolète. CodeCatalyst Les cibles peuvent devenir périmées si :

- Vous avez supprimé le flux de travail déployé sur la cible.
- Vous avez modifié la pile ou le cluster sur lequel vous effectuez le déploiement.
- Vous avez supprimé la pile CloudFormation ou le cluster du service Amazon ECS dans la AWS console.

## Pour supprimer une cible de déploiement

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Environments.
4. Choisissez le nom de l'environnement qui contient la cible de déploiement que vous souhaitez supprimer. Pour plus d'informations sur les environnements, consultez [Utilisation des environnements](#).
5. Choisissez l'onglet Cibles de déploiement.
6. Cliquez sur le bouton radio situé à côté de la cible de déploiement que vous souhaitez supprimer.
7. Sélectionnez Remove (Supprimer).

La cible est supprimée de la page.

## Utilisation des flux de travail

Un flux de travail est une procédure automatisée qui décrit comment créer, tester et déployer votre code dans le cadre d'un système d'intégration et de livraison continues (CI/CD). Un flux de travail définit une série d'étapes, ou d'actions, à effectuer lors de l'exécution d'un flux de travail. Un flux de travail définit également les événements, ou déclencheurs, qui déclenchent le démarrage du flux de travail. Pour configurer un flux de travail, vous devez créer un fichier de définition de flux de travail à l'aide de l'[éditeur visuel ou YAML](#) de la CodeCatalyst console.

### Tip

Pour un aperçu rapide de la manière dont vous pouvez utiliser les flux de travail dans un projet, [créez un projet avec un plan](#). Chaque plan déploie un flux de travail fonctionnel que vous pouvez examiner, exécuter et tester.

## Rubriques

- [Création, modification et suppression d'un flux de travail](#)
- [Affichage de l'état du flux de travail](#)
- [Utilisation des actions](#)

- [Travailler avec des artefacts](#)
- [Utilisation des images Docker de l'environnement de calcul et d'exécution](#)
- [Utilisation des environnements](#)
- [Utilisation de la mise en cache de fichiers](#)
- [Travailler avec des packages](#)
- [Travailler avec des courses](#)
- [Travailler avec des secrets](#)
- [Travailler avec des sources](#)
- [Utilisation de déclencheurs](#)
- [Utilisation des variables](#)

## Création, modification et suppression d'un flux de travail

Utilisez les procédures suivantes pour créer, modifier et supprimer un flux de travail.

### Visual

Pour créer un flux de travail à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez Créer un flux de travail.


La boîte de dialogue Créer un flux de travail apparaît.

5. Dans le champ Référentiel source, choisissez un référentiel source dans lequel résidera le fichier de définition du flux de travail. Le fichier sera stocké dans le `~/codecatalyst/workflows/` dossier du référentiel choisi. S'il n'existe aucun référentiel source, [créez-en un](#).
6. Dans le champ Branche, choisissez une branche dans laquelle résidera le fichier de définition du flux de travail.
7. Sélectionnez Create (Créer).

Amazon CodeCatalyst enregistre les informations du référentiel et de la succursale en mémoire, mais le flux de travail n'est pas encore validé.

8. Choisissez Visual.

9. Créez le flux de travail :
  - a. (Facultatif) Dans le diagramme du flux de travail, sélectionnez la case Source et déclencheurs. Un volet Déclencheurs apparaît. Choisissez Ajouter un déclencheur pour ajouter un déclencheur. Pour en savoir plus, consultez [Ajouter un déclencheur push, pull ou Schedule](#).
  - b. Choisissez + Actions (en haut à gauche). Le catalogue Actions apparaît.
  - c. Choisissez le signe plus (+) dans une action pour l'ajouter au flux de travail. Utilisez le volet de droite pour configurer l'action. Pour en savoir plus, consultez [Ajouter une action](#).
  - d. (Facultatif) Choisissez les propriétés du flux de travail (en haut à droite). Un volet des propriétés du flux de travail apparaît. Configurez le nom du flux de travail, le mode d'exécution et le calcul. Pour plus d'informations, veuillez consulter [Configuration des exécutions en file d'attente, remplacées et parallèles](#) et [Utilisation des images Docker de l'environnement de calcul et d'exécution](#).
10. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
11. Choisissez Commit, puis dans la boîte de dialogue Commit workflow, procédez comme suit :
  - a. Pour le nom du fichier Workflow, laissez le nom par défaut ou entrez le vôtre.
  - b. Pour le message de validation, laissez le message par défaut ou entrez le vôtre.
  - c. Pour Repository and Branch, choisissez le référentiel source et la branche pour le fichier de définition du flux de travail. Ces champs doivent être définis sur le référentiel et la branche que vous avez spécifiés précédemment dans la boîte de dialogue Créer un flux de travail. Vous pouvez modifier le dépôt et la branche dès maintenant, si vous le souhaitez.

 Note

Après avoir validé votre fichier de définition de flux de travail, celui-ci ne peut pas être associé à un autre référentiel ou branche. Veillez donc à les choisir avec soin.

- d. Choisissez Commit pour valider le fichier de définition du flux de travail.



## YAML

Pour créer un flux de travail à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez Créer un flux de travail.


La boîte de dialogue Créer un flux de travail apparaît.

5. Dans le champ Référentiel source, choisissez un référentiel source dans lequel résidera le fichier de définition du flux de travail. Le fichier sera stocké dans le `~/ .codecatalyst/workflows/` dossier du référentiel choisi. S'il n'existe aucun référentiel source, [créez-en un](#).
6. Dans le champ Branche, choisissez une branche dans laquelle résidera le fichier de définition du flux de travail.
7. Sélectionnez Create (Créer).

Amazon CodeCatalyst enregistre les informations du référentiel et de la succursale en mémoire, mais le flux de travail n'est pas encore validé.

8. Choisissez YAML.
9. Créez le flux de travail :
  - a. (Facultatif) Ajoutez un déclencheur au code YAML. Pour en savoir plus, consultez [Ajouter un déclencheur push, pull ou Schedule](#).
  - b. Choisissez + Actions (en haut à gauche). Le catalogue Actions apparaît.
  - c. Choisissez le signe plus (+) dans une action pour l'ajouter au flux de travail. Utilisez le volet de droite pour configurer l'action. Pour en savoir plus, consultez [Ajouter une action](#).
  - d. (Facultatif) Choisissez les propriétés du flux de travail (en haut à droite). Un volet des propriétés du flux de travail apparaît. Configurez le nom du flux de travail, le mode d'exécution et le calcul. Pour plus d'informations, veuillez consulter [Configuration des exécutions en file d'attente, remplacées et parallèles](#) et [Utilisation des images Docker de l'environnement de calcul et d'exécution](#).
10. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
11. Choisissez Commit, puis dans la boîte de dialogue Commit workflow, procédez comme suit :
  - a. Pour le nom du fichier Workflow, laissez le nom par défaut ou entrez le vôtre.

- b. Pour le message de validation, laissez le message par défaut ou entrez le vôtre.
- c. Pour Repository and Branch, choisissez le référentiel source et la branche pour le fichier de définition du flux de travail. Ces champs doivent être définis sur le référentiel et la branche que vous avez spécifiés précédemment dans la boîte de dialogue Créer un flux de travail. Vous pouvez modifier le dépôt et la branche dès maintenant, si vous le souhaitez.

 Note

Après avoir validé votre fichier de définition de flux de travail, celui-ci ne peut pas être associé à un autre référentiel ou branche. Veillez donc à les choisir avec soin.

- d. Choisissez Commit pour valider le fichier de définition du flux de travail.

### Pour modifier un flux de travail

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom du flux de travail que vous souhaitez modifier. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Modifiez le flux de travail. Choisissez Visual pour utiliser l'éditeur visuel ou YAML pour utiliser l'éditeur YAML. Vous pouvez :
  - a. Modifiez les déclencheurs. Pour en savoir plus, consultez [Utilisation de déclencheurs](#).
  - b. Modifier les actions. Pour en savoir plus, consultez [Utilisation des actions](#).
7. Modifiez les propriétés supplémentaires du flux de travail. Choisissez Propriétés du flux de travail (en haut à droite) pour modifier le nom du flux de travail, le mode d'exécution ou le calcul. Pour plus d'informations, veuillez consulter [Configuration des exécutions en file d'attente, remplacées et parallèles](#) et [Utilisation des images Docker de l'environnement de calcul et d'exécution](#).

## Pour supprimer un flux de travail

La suppression d'un flux de travail entraîne la suppression du fichier de définition du flux de travail et des exécutions associées. Tous AWS les services déployés à partir du flux de travail supprimé doivent être supprimés manuellement, faute de quoi des frais continueront d'être facturés pour ces services.

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom du flux de travail que vous souhaitez supprimer. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Sélectionnez Supprimer.
6. Choisissez Confirmer.

## Affichage de l'état du flux de travail

Vous souhaitez peut-être consulter l'état d'un flux de travail pour voir s'il existe des problèmes de configuration du flux de travail à résoudre ou pour résoudre les problèmes d'exécution qui ne démarrent pas. CodeCatalyst évalue l'état du flux de travail chaque fois que vous créez ou mettez à jour le [fichier de définition du flux de travail sous-jacent du flux](#) de travail.

### Note

Vous pouvez également consulter l'état d'exécution du flux de travail, qui est différent de l'état du flux de travail. Pour de plus amples informations, veuillez consulter [Afficher le statut et les détails de l'exécution du flux de travail](#).

Un flux de travail peut avoir l'un des statuts suivants :

- Valide — [Le flux de travail est exécutable et peut être activé par des déclencheurs](#).

Pour qu'un flux de travail soit marqué comme valide, les deux conditions suivantes doivent être remplies :

- Le fichier de définition du flux de travail doit être valide.

- Le flux de travail ne doit comporter aucun déclencheur, aucun déclencheur push ou un déclencheur push qui s'exécute à l'aide des fichiers de la branche actuelle. Pour de plus amples informations, veuillez consulter [Considérations relatives aux déclencheurs lors du branchement](#).
- Non valide : le fichier de définition du flux de travail n'est pas valide. Le flux de travail ne peut pas être exécuté manuellement ou automatiquement par le biais de déclencheurs. *Les flux de travail non valides apparaissent avec une **définition du flux de travail et aucun message d'erreur** (ou similaire) ne s'affiche dans la CodeCatalyst console.*

Pour qu'un flux de travail soit marqué comme non valide, la condition suivante doit être vraie :

- Le fichier de définition du flux de travail doit être mal configuré.

Pour corriger un fichier de définition de flux de travail mal configuré, consultez [Comment corriger les erreurs « La définition du flux de travail ne comporte aucune erreur » ?](#).

- Inactif : la définition du flux de travail est valide mais ne peut pas être exécutée manuellement ou automatiquement via des déclencheurs.

Pour qu'un flux de travail soit marqué comme inactif, les deux conditions suivantes doivent être remplies :

- Le fichier de définition du flux de travail doit être valide.
- Le fichier de définition du flux de travail doit inclure un déclencheur push qui spécifie une branche différente de celle sur laquelle se trouve actuellement le fichier de définition du flux de travail. Pour de plus amples informations, veuillez consulter [Considérations relatives aux déclencheurs lors du branchement](#).

Pour passer d'un flux de travail d'inactif à actif, voir [Comment corriger les messages « Le flux de travail est inactif » ?](#).

#### Note

Si le flux de travail indique une ressource que vous supprimez ultérieurement (par exemple, un référentiel de packages), il CodeCatalyst ne détectera pas cette modification et continuera à marquer le flux de travail comme valide. Ces types de problèmes seront détectés lors de l'exécution du flux de travail.

## Pour consulter le statut d'un flux de travail

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Recherchez le flux de travail dont vous souhaitez consulter le statut. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.

Le statut est affiché avec le flux de travail dans la liste.

5. (Facultatif) Choisissez le nom du flux de travail et recherchez le champ de définition du flux de travail. Il indique l'état du flux de travail.

## Utilisation des actions

Une action est le principal élément constitutif d'un flux de travail et définit une unité logique de travail à effectuer lors de l'exécution d'un flux de travail. Généralement, un flux de travail inclut plusieurs actions qui s'exécutent de manière séquentielle ou en parallèle selon la façon dont vous les avez configurées.

### Rubriques

- [Types d'action](#)
- [Ajouter, configurer et supprimer une action](#)
- [Développement d'une action personnalisée](#)
- [Regrouper les actions dans des groupes d'action](#)
- [Configuration des actions pour qu'elles dépendent d'autres actions](#)
- [Afficher le code source d'une action](#)
- [Utilisation des versions d'action](#)

### Types d'action

Dans un CodeCatalyst flux de travail Amazon, vous pouvez utiliser les types d'actions suivants.

### Types d'action

- [CodeCatalyst actions](#)
- [CodeCatalyst Actions menées par les laboratoires](#)
- [GitHub Actions](#)
- [Actions tierces](#)

## CodeCatalyst actions

Une CodeCatalyst action est une action créée, gérée et entièrement prise en charge par l'équipe de CodeCatalyst développement.

Il existe CodeCatalyst des actions permettant de créer, de tester et de déployer des applications, ainsi que d'effectuer diverses tâches, telles que l'appel d'une AWS Lambda fonction.

Les CodeCatalyst actions disponibles sont les suivantes :

- Build

Cette action crée vos artefacts et exécute vos tests unitaires dans un conteneur Docker. Pour de plus amples informations, veuillez consulter [Ajouter l'action de construction](#).

- Test

Cette action exécute des tests d'intégration et de système par rapport à votre application ou à vos artefacts. Pour de plus amples informations, veuillez consulter [Ajouter l'action de test](#).

- Publication d'Amazon S3

Cette action copie les artefacts de votre application dans un compartiment Amazon S3. Pour de plus amples informations, veuillez consulter [Ajout de l'action « Amazon S3 publish »](#).

- AWS CDK sangle

Cette action fournit les ressources dont ils ont AWS CDK besoin pour déployer votre application CDK. Pour de plus amples informations, veuillez consulter [Ajout de l'action « AWS CDK bootstrap »](#).

- AWS CDK déployer

Cette action synthétise et déploie une AWS Cloud Development Kit (AWS CDK) application. Pour de plus amples informations, veuillez consulter [Ajout de l'action « AWS CDK déployer »](#).

- AWS Lambda invoquer

Cette action appelle une AWS Lambda fonction. Pour de plus amples informations, veuillez consulter [Ajouter l'action « AWS Lambda invoquer »](#).

- Déployer AWS CloudFormation une pile

Cette action déploie des AWS CloudFormation piles. Pour de plus amples informations, veuillez consulter [Ajout de l'action « Deploy AWS CloudFormation stack »](#).

- Déploiement sur Amazon ECS

Cette action enregistre une définition de tâche Amazon ECS et la déploie sur un service Amazon ECS. Pour de plus amples informations, veuillez consulter [Ajout de l'action « Déployer sur Amazon ECS »](#).

- Déploiement sur un cluster Kubernetes

Cette action déploie une application sur un cluster Kubernetes. Pour de plus amples informations, veuillez consulter [Ajout de l'action « Déployer vers le cluster Kubernetes »](#).

- Afficher la définition de la tâche Amazon ECS

Cette action insère un URI d'image de conteneur dans un fichier JSON de définition de tâche Amazon ECS, créant ainsi un nouveau fichier de définition de tâche. Pour de plus amples informations, veuillez consulter [Ajout de l'action « Render la définition de tâche Amazon ECS »](#).

La documentation des CodeCatalyst actions est disponible dans ce guide et dans le fichier readme de chaque action.

Pour plus d'informations sur les CodeCatalyst actions disponibles et sur la façon d'en ajouter une à un flux de travail, consultez [Ajouter une action](#).

### CodeCatalyst Actions menées par les laboratoires

Une action CodeCatalyst Labs est une action qui fait partie d'Amazon CodeCatalyst Labs, un terrain d'essai pour les applications expérimentales. CodeCatalyst Des actions de laboratoire ont été développées pour présenter les intégrations aux AWS services.

Les actions CodeCatalyst Labs suivantes sont disponibles :

- Déployer vers un AWS Amplify hébergement

Cette action déploie une application sur Amplify Hosting.

- Déployer vers AWS App Runner

Cette action déploie la dernière image d'un référentiel d'images source dans App Runner.

- Déploiement sur Amazon CloudFront et Amazon S3

Cette action déploie une application vers Amazon S3 CloudFront et Amazon S3.

- Déployez avec AWS SAM

Cette action déploie votre application sans serveur avec AWS Serverless Application Model (AWS SAM).

- Invalider Amazon Cache CloudFront

Cette action invalide un CloudFront cache pour un ensemble de chemins donné.

- Action Webhook sortante

Cette action permet aux utilisateurs d'envoyer des messages dans un flux de travail à un serveur Web arbitraire à l'aide d'une requête HTTPS.

- Publier sur AWS CodeArtifact

Cette action publie des packages dans un CodeArtifact référentiel.

- Publier sur Amazon SNS

Cette action permet aux utilisateurs d'intégrer Amazon SNS en créant une rubrique, en publiant sur une rubrique ou en s'abonnant à une rubrique.

- Envoyer vers Amazon ECR

Cette action crée et publie une image Docker dans un référentiel Amazon Elastic Container Registry (Amazon ECR).

- Scannez avec Amazon CodeGuru Security

Cette action crée une archive zip d'un chemin de code configuré et utilise CodeGuru Security pour exécuter un scan de code.

- Édition communautaire Terraform

Cette action exécute Terraform Community Edition plan et apply ses opérations.

La documentation des actions CodeCatalyst Labs est disponible dans le fichier readme de chaque action.



Pour plus d'informations sur l'ajout d'une action CodeCatalyst Labs à un flux de travail et l'affichage de son fichier readme, consultez [Ajouter une action](#).

## GitHub Actions

Une GitHub action ressemble beaucoup à une [CodeCatalyst action](#), sauf qu'elle a été développée pour être utilisée avec des GitHub flux de travail. Pour plus de détails sur GitHub les actions, consultez la documentation sur [GitHub les actions](#).

Vous pouvez utiliser une GitHub action en même temps que CodeCatalyst des actions natives dans un CodeCatalyst flux de travail.

Pour vous faciliter la tâche, la CodeCatalyst console donne accès à plusieurs GitHub actions populaires. Vous pouvez également utiliser n'importe quelle GitHub action répertoriée [GitHub sur le Marketplace](#) (sous réserve de quelques restrictions).

La documentation relative aux GitHub actions est disponible dans le fichier readme de chaque action.

Pour de plus amples informations, veuillez consulter [Ajouter des GitHub actions](#).

## Actions tierces

Une action tierce est une action créée par un fournisseur tiers et mise à disposition dans la CodeCatalyst console. L'action Mend SCA, créée par Mend, est un exemple d'action tierce.

La documentation relative aux actions tierces est disponible dans le fichier readme de chaque action. Une documentation supplémentaire peut également être fournie par le fournisseur tiers.

Pour plus d'informations sur l'ajout d'une action tierce à un flux de travail et l'affichage de son fichier readme, consultez [Ajouter une action](#).

## Ajouter, configurer et supprimer une action

Choisissez l'une des rubriques suivantes pour savoir comment ajouter l'action à un flux de travail et comment la supprimer.

### Rubriques

- [Ajouter une action](#)
- [Ajout de l'action « Amazon S3 publish »](#)
- [Ajout de l'action « AWS CDK bootstrap »](#)
- [Ajouter l'action « AWS Lambda invoquer »](#)

- [Ajout de l'action « Render la définition de tâche Amazon ECS »](#)
- [Ajouter des GitHub actions](#)
- [Supprimer une action](#)

## Ajouter une action

Suivez les instructions ci-dessous pour ajouter une action à un flux de travail, puis pour le configurer.

Pour ajouter et configurer une action

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. En haut à gauche, choisissez + Actions. Le catalogue d'actions apparaît.
7. Dans la liste déroulante, effectuez l'une des opérations suivantes :
  - Choisissez Amazon CodeCatalyst pour afficher [CodeCatalyst](#), [CodeCatalyst Labs](#) ou les actions de [tiers](#).
    - CodeCatalyst les actions ont une AWSétiquette secondaire.
    - CodeCatalyst Les actions des laboratoires portent le label by CodeCatalyst Labs.
    - Les actions de tiers sont **libellées** par *fournisseur*, où *fournisseur* est le nom du fournisseur tiers.
  - Choisissez GitHub d'afficher une [liste organisée d' GitHub actions](#).
8. Dans le catalogue d'actions, recherchez une action, puis effectuez l'une des opérations suivantes :
  - Choisissez le signe plus (+) pour ajouter l'action à votre flux de travail.
  - Choisissez le nom de l'action pour afficher son fichier readme.
9. Configurez l'action. Choisissez Visual pour utiliser l'éditeur visuel ou YAML pour utiliser l'éditeur YAML. Pour obtenir des instructions détaillées, consultez les liens suivants.

Pour obtenir des instructions sur l'ajout d'[CodeCatalystactions](#), voir :

- [Ajouter l'action de construction](#)
- [Ajouter l'action de test](#)
- [Ajout de l'action « Amazon S3 publish »](#)
- [Ajout de l'action « AWS CDK bootstrap »](#)
- [Ajout de l'action « AWS CDK déployer »](#)
- [Ajouter l'action « AWS Lambda invoquer »](#)
- [Ajout de l'action « Deploy AWS CloudFormation stack »](#)
- [Ajout de l'action « Déployer sur Amazon ECS »](#)
- [Ajout de l'action « Déployer vers le cluster Kubernetes »](#)
- [Ajout de l'action « Render la définition de tâche Amazon ECS »](#)

Pour obtenir des instructions sur l'ajout d'[actions CodeCatalyst Labs](#), voir :

- L'action est readme. Vous pouvez trouver le fichier readme en choisissant le nom de l'action dans le catalogue d'actions.

Pour obtenir des instructions sur l'ajout d'[GitHub actions](#), voir :

- [Ajouter des GitHub actions](#)

Pour obtenir des instructions sur l'ajout d'[actions tierces](#), voir :

- L'action est readme. Vous pouvez trouver le fichier readme en choisissant le nom de l'action dans le catalogue d'actions.

10. (Facultatif) Choisissez Valider pour vous assurer que le code YAML est valide.

11. Choisissez Valider pour valider vos modifications.

### Ajout de l'action « Amazon S3 publish »

Cette section décrit comment ajouter l'action de publication Amazon S3 à votre flux de travail. L'action de publication Amazon S3 copie les fichiers d'un répertoire source vers un compartiment Amazon S3. Le répertoire source peut résider dans :

- Un [référentiel source](#), ou

- [Artefact de sortie](#) généré par une autre action du flux de travail

## Rubriques

- [Quand utiliser cette action](#)
- [Exemple de flux de travail](#)
- [Ajout de l'action « Amazon S3 publish »](#)
- [Variables produites par l'action « Amazon S3 publish »](#)
- [Définition de l'action « Amazon S3 publish »](#)

## Quand utiliser cette action

Utilisez cette action si :

- Vous disposez d'un flux de travail qui génère les fichiers que vous souhaitez stocker dans Amazon S3.

Par exemple, vous pouvez avoir un flux de travail qui crée un site Web statique que vous souhaitez héberger dans Amazon S3. Dans ce cas, votre flux de travail inclurait une [action](#) de création pour créer le code HTML du site et les fichiers de support, et une action de publication Amazon S3 pour copier les fichiers sur Amazon S3.

- Vous disposez d'un référentiel source qui contient les fichiers que vous souhaitez stocker dans Amazon S3.

Par exemple, vous pouvez disposer d'un référentiel source contenant des fichiers source d'application que vous souhaitez archiver tous les soirs sur Amazon S3.

## Exemple de flux de travail

L'exemple de flux de travail suivant inclut l'action de publication Amazon S3, ainsi qu'une action de génération. Le flux de travail crée un site Web de documentation statique, puis le publie sur Amazon S3, où il est hébergé. Le flux de travail comprend les éléments de base suivants qui s'exécutent de manière séquentielle :

**Note**

L'exemple de flux de travail suivant est fourni à titre d'illustration et ne fonctionnera qu'avec une configuration supplémentaire.

- Un déclencheur : ce déclencheur lance automatiquement l'exécution du flux de travail lorsque vous apportez une modification à votre référentiel source. Pour plus d'informations sur les déclencheurs, consultez [Utilisation de déclencheurs](#).
- Une action de construction (BuildDocs) — Au moment du déclenchement, l'action crée un site Web de documentation statique (mkdocs build) et ajoute les fichiers HTML associés et les métadonnées associées à un artefact appelé MyDocsSite. Pour plus d'informations sur l'action de génération, consultez [Création à l'aide de flux de travail dans CodeCatalyst](#).
- Une action de publication Amazon S3 (PublishToS3) : une fois l'action de création terminée, cette action copie le site contenu dans l'artefact MyDocsSite vers Amazon S3 pour l'hébergement.

```
Name: codecatalyst-s3-publish-workflow
SchemaVersion: 1.0

Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  BuildDocs:
    Identifiant: aws/build@v1
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - Run: echo BuildDocs started on `date`
        - Run: pip install --upgrade pip
        - Run: pip install mkdocs
        - Run: mkdocs build
        - Run: echo BuildDocs completed on `date`
    Outputs:
      Artifacts:
        - Name: MyDocsSite
```

```
Files:  
  - "site/**/*"
```

**PublishToS3:**

Identifiant: aws/s3-publish@v1

**Environment:**

Name: codecatalyst-s3-publish-environment

**Connections:**

- Name: codecatalyst-account-connection
- Role: codecatalyst-s3-publish-build-role

**Inputs:****Sources:**

- WorkflowSource

**Artifacts:**

- MyDocsSite

**Configuration:**

DestinationBucketName: my-bucket

SourcePath: /artifacts/PublishToS3/MyDocSite/site

TargetPath: my/docs/site

## Ajout de l'action « Amazon S3 publish »

Suivez les instructions suivantes pour ajouter l'action de publication Amazon S3 à votre flux de travail.

### Visual

Pour ajouter l'action « Amazon S3 publish » à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez Visual.
7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
8. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst.
9. Recherchez l'action de publication Amazon S3 et effectuez l'une des opérations suivantes :

- Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.

Ou

- Choisissez Amazon S3 Publish. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
  - (Facultatif) Choisissez Afficher la source pour [afficher le code source de l'action](#).
  - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.

10. Dans les onglets Entrées, Configuration et Sorties, complétez les champs en fonction de vos besoins. Pour une description de chaque champ, consultez le [Référence d'action « Amazon S3 publish »](#). Cette référence fournit des informations détaillées sur chaque champ (et la valeur de propriété YAML correspondante) tel qu'il apparaît dans les éditeurs YAML et visuels.
11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour ajouter l'action « Amazon S3 publish » à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
8. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst.
9. Recherchez l'action de publication Amazon S3 et effectuez l'une des opérations suivantes :
  - Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.

Ou

- Choisissez Amazon S3 Publish. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
  - (Facultatif) Choisissez Afficher la source pour [afficher le code source de l'action](#).
  - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.
- 10. Modifiez les propriétés du code YAML en fonction de vos besoins. Une explication de chaque propriété disponible est fournie dans le [Référence d'action « Amazon S3 publish »](#).
- 11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
- 12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

Variables produites par l'action « Amazon S3 publish »

Lorsque l'action de publication Amazon S3 s'exécute, elle produit des variables que vous pouvez utiliser dans les actions de flux de travail suivantes. Pour plus de détails, voir [Variables d'action « Amazon S3 publish »](#) dans le [Liste des variables prédéfinies](#).

Définition de l'action « Amazon S3 publish »

L'action de publication Amazon S3 est définie comme un ensemble de propriétés YAML dans votre fichier de définition de flux de travail. Pour plus d'informations sur ces propriétés, consultez [Référence d'action « Amazon S3 publish »](#) le [Référence de définition du flux de travail](#).

Ajout de l'action « AWS CDK bootstrap »

Cette section explique comment ajouter l'action AWS CDK bootstrap à votre flux de travail. L'action AWS CDK bootstrap fournit une pile bootstrap dans votre AWS environnement à l'aide du modèle [moderne](#). Si une pile bootstrap existe déjà, l'action la met à jour si nécessaire. La présence d'une pile bootstrap AWS est une condition préalable au déploiement d'une AWS CDK application.

Pour plus d'informations sur le bootstrapping, voir [Bootstrapping](#) dans le manuel du développeur.AWS Cloud Development Kit (AWS CDK)

Rubriques

- [Quand utiliser cette action](#)
- [Comment ça marche](#)



- [Versions de la CLI CDK utilisées par l'action « AWS CDK bootstrap »](#)
- [Prérequis](#)
- [Exemple de flux de travail](#)
- [Ajout de l'action « AWS CDK bootstrap »](#)
- [Variables produites par l'action « AWS CDK bootstrap »](#)
- [définition de l'action « AWS CDK bootstrap »](#)

## Quand utiliser cette action

Utilisez cette action si vous disposez d'un flux de travail qui déploie une AWS CDK application et que vous souhaitez déployer (et mettre à jour, si nécessaire) la pile bootstrap en même temps. Dans ce cas, vous devez ajouter l'action AWS CDK bootstrap au même flux de travail que celui qui déploie votre AWS CDK application.

N'utilisez pas cette action si l'une des conditions suivantes s'applique :

- Vous avez déjà déployé une pile de bootstrap à l'aide d'un autre mécanisme et vous souhaitez la conserver intacte (aucune mise à jour).
- Vous souhaitez utiliser un modèle de [bootstrap personnalisé](#), qui n'est pas pris en charge par l'action AWS CDK bootstrap.

## Comment ça marche

Le AWS CDK bootstrap fonctionne comme suit :

1. [Au moment de l'exécution, si vous avez spécifié la version 1.0.7 ou une version antérieure de l'action, celle-ci télécharge la dernière CLI CDK \(également appelée AWS CDK Toolkit\) sur l'image de construction. CodeCatalyst](#)

Si vous avez spécifié la version 1.0.8 ou ultérieure, l'action est fournie avec une [version spécifique](#) de la CLI CDK, de sorte qu'aucun téléchargement n'a lieu.

2. L'action utilise la CLI CDK pour exécuter la `cdk bootstrap` commande. Cette commande exécute les tâches d'amorçage décrites dans la rubrique [Bootstrapping](#) du Guide du développeur.AWS Cloud Development Kit (AWS CDK)

## Versions de la CLI CDK utilisées par l'action « AWS CDK bootstrap »

Le tableau suivant indique quelle version de la CLI CDK est utilisée par défaut par les différentes versions de l'action AWS CDK bootstrap.

### Note

Vous pouvez peut-être remplacer la valeur par défaut. Pour de plus amples informations, veuillez consulter [CdkCliVersion](#) dans le [référence d'action « AWS CDK bootstrap »](#).

version d'action « AWS CDK bootstrap »	AWS CDK Version de la CLI
10,0 — 1,0,7	dernières
1.0.8 ou version ultérieure	2,99,1

## Prérequis

Avant de pouvoir utiliser l'action AWS CDK bootstrap, assurez-vous qu'une AWS CDK application est prête. L'action bootstrap synthétisera l' AWS CDK application avant le démarrage. Vous pouvez écrire votre application dans n'importe quel langage de programmation pris en charge par le AWS CDK.

Assurez-vous que les fichiers de votre AWS CDK application sont disponibles dans :

- Un [référentiel CodeCatalyst source](#), ou
- [Artefact CodeCatalyst de sortie](#) généré par une autre action du flux de travail

## Exemple de flux de travail

Reportez-vous au [Exemple de flux de travail](#) in [Ajout de l'action « AWS CDK déployer »](#) pour un flux de travail incluant l'action AWS CDK bootstrap.

## Ajout de l'action « AWS CDK bootstrap »

Suivez les instructions ci-dessous pour ajouter l'action AWS CDK bootstrap à votre flux de travail.

## Prérequis

Avant de commencer, assurez-vous d'avoir effectué les tâches décrites dans [Prérequis](#).

## Visual

Pour ajouter l'action « AWS CDK bootstrap » à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez Visual.
7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
8. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst.
9. Recherchez l'action AWS CDK bootstrap, puis effectuez l'une des opérations suivantes :

- Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.

Ou

- Choisissez AWS CDK bootstrap. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
  - (Facultatif) Choisissez Afficher la source pour [afficher le code source de l'action](#).
  - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.

10. Dans les onglets Entrées, Configuration et Sorties, complétez les champs en fonction de vos besoins. Pour une description de chaque champ, consultez le [référence d'action « AWS CDK bootstrap »](#). Cette référence fournit des informations détaillées sur chaque champ (et la valeur de propriété YAML correspondante) tel qu'il apparaît dans les éditeurs YAML et visuels.
11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

**Note**

Si votre action AWS CDK bootstrap échoue avec une `npm install` erreur, reportez-vous à [Comment corriger les erreurs « npm install » ?](#) la section pour savoir comment corriger l'erreur.

## YAML

Pour ajouter l'action « AWS CDK bootstrap » à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez YAML.
7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
8. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst.
9. Recherchez l'action AWS CDK bootstrap, puis choisissez + pour l'ajouter au diagramme de flux de travail et ouvrir son volet de configuration.
10. Modifiez les propriétés du code YAML en fonction de vos besoins. Une explication de chaque propriété disponible est fournie dans le [référence d'action « AWS CDK bootstrap »](#).
11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

**Note**

Si votre action AWS CDK bootstrap échoue avec une `npm install` erreur, reportez-vous à [Comment corriger les erreurs « npm install » ?](#) la section pour savoir comment corriger l'erreur.

## Variables produites par l'action « AWS CDK bootstrap »

Lorsque l'action AWS CDK bootstrap est exécutée, elle produit des variables que vous pouvez utiliser dans les actions de flux de travail suivantes. Pour plus de détails, voir [variables d'action « AWS CDK bootstrap »](#) dans le [Liste des variables prédéfinies](#).

### définition de l'action « AWS CDK bootstrap »

L'action AWS CDK bootstrap est définie comme un ensemble de propriétés YAML dans le fichier de définition de votre flux de travail. Pour plus d'informations sur ces propriétés, consultez [référence d'action « AWS CDK bootstrap »](#) le [Référence de définition du flux de travail](#).

### Ajouter l'action « AWS Lambda invoquer »

Cette section décrit comment ajouter l'action d'AWS Lambdaappel à votre flux de travail. L'action AWS Lambdad'appel appelle une fonction Lambda que vous spécifiez.

[Outre l'appel de votre fonction, l'action d'AWS Lambdaappel convertit également chaque clé de niveau supérieur de la charge utile de réponse reçue de la fonction Lambda en une variable de sortie du flux de travail](#). Ces variables peuvent ensuite être référencées dans les actions de flux de travail suivantes. Si vous ne souhaitez pas que toutes les clés de niveau supérieur soient converties en variables, vous pouvez utiliser des filtres pour spécifier les clés exactes. Pour plus d'informations, consultez la description de la [ResponseFilters](#) propriété dans le [référence d'action « AWS Lambda invoquer »](#).

### Rubriques

- [Quand utiliser cette action](#)
- [Exemple de flux de travail](#)
- [Ajouter l'action « AWS Lambda invoquer »](#)
- [Variables produites par l'action « AWS Lambda invoke »](#)
- [définition de l'action « AWS Lambda invoquer »](#)

### Quand utiliser cette action

Utilisez cette action si vous souhaitez ajouter à votre flux de travail des fonctionnalités encapsulées et exécutées par une fonction Lambda.

Par exemple, vous souhaiterez peut-être que votre flux de travail envoie une `Build started` notification à une chaîne Slack avant de commencer à créer votre application. Dans ce cas, votre

flux de travail inclurait une AWS Lambda action d'appel pour invoquer un Lambda pour envoyer la notification Slack, et une [action de génération pour créer](#) votre application.

Autre exemple, vous souhaitez peut-être que votre flux de travail effectue une analyse des vulnérabilités de votre application avant son déploiement. Dans ce cas, vous devez utiliser une action de génération pour créer votre application, une action d'AWS Lambda appel pour appeler un Lambda pour rechercher des vulnérabilités et une action de déploiement pour déployer l'application scannée.

### Exemple de flux de travail

#### Note

L'exemple de flux de travail suivant est fourni à titre indicatif uniquement et nécessite un travail de configuration supplémentaire pour fonctionner correctement. Il est destiné à vous donner un exemple de ce à quoi peut ressembler un flux de travail lorsqu'il est configuré avec l'action AWS Lambda d'appel.

Le flux de travail suivant inclut l'AWS Lambda action d'appel, ainsi qu'une action de déploiement. Le flux de travail envoie une notification Slack indiquant qu'un déploiement a commencé, puis déploie une application en AWS utilisant un AWS CloudFormation modèle. Le flux de travail comprend les éléments de base suivants qui s'exécutent de manière séquentielle :

- Un déclencheur : ce déclencheur lance automatiquement l'exécution du flux de travail lorsque vous apportez une modification à votre référentiel source. Pour plus d'informations sur les déclencheurs, consultez [Utilisation de déclencheurs](#).
- Une action d'AWS Lambda appel (LambdaNotif y) — Au moment du déclenchement, cette action invoque la fonction Notif y-Start Lambda dans le compte et la région AWS spécifiés my-aws-account (, us-west-2 et). Lors de l'appel, la fonction Lambda envoie une notification Slack indiquant qu'un déploiement a commencé.
- Une action Deploy AWS CloudFormation stack (Deploy) — Une fois l'action AWS Lambda d'appel terminée, l'action Deploy AWS CloudFormation stack exécute le template (cfn-template.yml) pour déployer votre pile d'applications. Pour plus d'informations sur l'action Deploy AWS CloudFormation stack, consultez [Ajout de l'action « Deploy AWS CloudFormation stack »](#).

```
Name: codecatalyst-lambda-invoke-workflow
SchemaVersion: 1.0
```

```
Triggers:
- Type: PUSH
  Branches:
    - main
Actions:
  LambdaNotify:
    Identifiant: aws/lambda-invoke@v1
    Environnement:
      Name: my-production-environment
    Connexions:
      - Name: my-aws-account
        Role: codecatalyst-lambda-invoke-role
    Entrées:
      Sources:
        - WorkflowSource
    Configuration:
      Fonction: Notify-Start
      RégionAWS: us-west-2

  Deploy:
    Identifiant: aws/cfn-deploy@v1
    Environnement:
      Name: my-production-environment
    Connexions:
      - Name: my-aws-account
        Role: codecatalyst-deploy-role
    Entrées:
      Sources:
        - WorkflowSource
    Configuration:
      name: my-application-stack
      region: us-west-2
      role-arn: arn:aws:iam::111122223333:role/StackRole
      template: ./cfn-template.yml
      capacités: CAPABILITY_IAM,CAPABILITY_AUTO_EXPAND
```

Ajouter l'action « AWS Lambda invoquer »

Suivez les instructions suivantes pour ajouter l'action d'AWS Lambdaappel à votre flux de travail.

Prérequis

Avant de commencer, assurez-vous que votre AWS Lambda fonction et le rôle d'exécution Lambda associé sont prêts et disponibles dans. AWS Pour plus d'informations, consultez la rubrique relative au [rôle d'exécution Lambda](#) dans le Guide du AWS Lambda développeur.

## Visual

Pour ajouter l'action « AWS Lambda invoquer » à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez Visual.
7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
8. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst.
9. Recherchez l'action AWS Lambda d'appel, puis effectuez l'une des opérations suivantes :

- Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.

Ou

- Choisissez AWS LambdaInvoke. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
  - (Facultatif) Choisissez Afficher la source pour [afficher le code source de l'action](#).
  - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.

10. Dans les onglets Entrées, Configuration et Sorties, complétez les champs en fonction de vos besoins. Pour une description de chaque champ, consultez le [référence d'action « AWS Lambda invoquer »](#). Cette référence fournit des informations détaillées sur chaque champ (et la valeur de propriété YAML correspondante) tel qu'il apparaît dans les éditeurs YAML et visuels.
11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.



## YAML

Pour ajouter l'action « AWS Lambda invoke » à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez YAML.
7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
8. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst.
9. Recherchez l'action AWS Lambda d'appel, puis effectuez l'une des opérations suivantes :
  - Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.

Ou

- Choisissez AWS LambdaInvoke. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
    - (Facultatif) Choisissez Afficher la source pour [afficher le code source de l'action](#).
    - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.
10. Modifiez les propriétés du code YAML en fonction de vos besoins. Une explication de chaque propriété disponible est fournie dans le [référence d'action « AWS Lambda invoquer »](#).
  11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
  12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

Variables produites par l'action « AWS Lambda invoke »

Lorsque l'action AWS Lambda d'appel est exécutée, elle produit des variables que vous pouvez utiliser dans les actions de flux de travail suivantes. Pour plus de détails, voir [variables d'action « AWS Lambda invoquer »](#) dans le [Liste des variables prédéfinies](#).

## définition de l'action « AWS Lambda invoquer »

L'action AWS Lambda d'appel est définie comme un ensemble de propriétés YAML dans votre fichier de définition de flux de travail. Pour plus d'informations sur ces propriétés, consultez [référence d'action « AWS Lambda invoquer »](#) le [Référence de définition du flux de travail](#).

## Ajout de l'action « Render la définition de tâche Amazon ECS »

Cette section décrit comment ajouter l'action de définition de tâche Render Amazon ECS à votre flux de travail. Cette action met à jour le champ image d'un fichier de [définition de tâche Amazon Elastic Container Service \(Amazon ECS\)](#) avec un nom d'image Docker fourni par votre flux de travail lors de l'exécution.

### Note

Vous pouvez également utiliser cette action pour mettre à jour le environnement champ de définition de tâche avec des variables d'environnement.

## Rubriques

- [Quand utiliser cette action](#)
- [Comment ça marche](#)
- [Exemple de flux de travail](#)
- [Ajout de l'action « Render la définition de tâche Amazon ECS »](#)
- [Affichage du fichier de définition de tâche mis à jour](#)
- [Variables produites par l'action « Rendre la définition de la tâche Amazon ECS »](#)
- [Définition de l'action « Rendre la définition de la tâche Amazon ECS »](#)

## Quand utiliser cette action

Utilisez-le si vous disposez d'un flux de travail qui crée et balise une image Docker avec du contenu dynamique, tel qu'un identifiant de validation ou un horodatage.

N'utilisez pas cette action si votre fichier de définition de tâche contient une valeur d'image qui reste toujours la même. Dans ce cas, vous pouvez saisir manuellement le nom de votre image dans le fichier de définition de tâche.

## Comment ça marche

Vous devez utiliser l'action de définition de tâche Render Amazon ECS avec les actions Build et Deploy to Amazon ECS dans votre flux de travail. Ensemble, ces actions fonctionnent comme suit :

1. L'action de création crée votre image Docker et l'étiquette avec un nom, un identifiant de validation, un horodatage ou un autre contenu dynamique. Par exemple, votre action de création peut ressembler à ceci :

```
MyECSWorkflow
  Actions:
    BuildAction:
      Identifieur: aws/build@v1
      ...
    Configuration:
      Steps:
        # Build, tag, and push the Docker image...
        - Run: docker build -t MyDockerImage:${WorkflowSource.CommitId} .
        ...
```

Dans le code précédent, la `docker build -t` directive indique de créer l'image Docker et de l'étiqueter avec l'ID de validation lors de l'exécution de l'action. Le nom de l'image générée peut ressembler à ceci :

`MyDockerImage:a37bd7e`

2. L'action de définition de tâche Render Amazon ECS ajoute le nom de l'image générée dynamiquement à votre fichier de définition de tâche, comme suit : `MyDockerImage:a37bd7e`

```
{
  "executionRoleArn": "arn:aws:iam::account_ID:role/codecatalyst-ecs-task-execution-role",
  "containerDefinitions": [
    {
      "name": "codecatalyst-ecs-container",
      "image": MyDockerImage:a37bd7e,
      "essential": true,
      ...
      "portMappings": [
        {
          "hostPort": 80,
          "protocol": "tcp",
```

```

        "containerPort": 80
      }
    ]
  },
  ...
}

```

Vous pouvez également demander à l'action Render Amazon ECS de définition de tâche d'ajouter des variables d'environnement à la définition de tâche, comme suit :

```

{
  "executionRoleArn": "arn:aws:iam::account_ID:role/codecatalyst-ecs-task-execution-
role",
  "containerDefinitions": [
    {
      "name": "codecatalyst-ecs-container",
      "image": MyDockerImage:a37bd7e,
      ...
      "environment": [
        {
          "name": "ECS_LOGLEVEL",
          "value": "info"
        }
      ]
    }
  ],
  ...
}

```

Pour plus d'informations sur les variables d'environnement, consultez la section [Spécification des variables d'environnement](#) dans le manuel Amazon Elastic Container Service Developer Guide.

3. L'action Deploy to Amazon ECS enregistre le fichier de définition de tâche mis à jour auprès d'Amazon ECS. L'enregistrement du fichier de définition de tâche mis à jour déploie la nouvelle image MyDockerImage : a37bd7e dans Amazon ECS.

## Exemple de flux de travail

Voici un exemple de flux de travail complet qui inclut l'action de définition de tâche Render Amazon ECS, ainsi que les actions de création et de déploiement. L'objectif du flux de travail est de créer

et de déployer une image Docker dans votre cluster Amazon ECS. Le flux de travail comprend les éléments de base suivants qui s'exécutent de manière séquentielle :

- Un déclencheur : ce déclencheur lance automatiquement l'exécution du flux de travail lorsque vous apportez une modification à votre référentiel source. Pour plus d'informations sur les déclencheurs, consultez [Utilisation de déclencheurs](#).
- Une action de génération (`BuildDocker`) — Au moment du déclenchement, l'action crée l'image Docker à l'aide du Dockerfile, l'étiquette avec un ID de validation et envoie l'image vers Amazon ECR. Pour plus d'informations sur l'action de génération, consultez [Création à l'aide de flux de travail dans CodeCatalyst](#).
- Une action de définition de tâche Render Amazon ECS (`RenderTaskDef`) : à la fin de l'action de génération, cette action met à jour un `taskdef.json` élément existant situé à la racine de votre référentiel source avec une valeur de `image` champ qui inclut l'ID de validation correct. Il enregistre le fichier mis à jour sous un nouveau nom de fichier (`task-definition-random-string.json`), puis crée un artefact de sortie contenant ce fichier. L'action de rendu génère également une variable appelée `task-definition` et lui attribue le nom du nouveau fichier de définition de tâche. L'artefact et la variable seront utilisés lors de l'action de déploiement, qui est la suivante.
- Une action de déploiement vers Amazon ECS (`DeployToECS`) : une fois l'action de définition de tâche Render Amazon ECS terminée, l'action Deploy to Amazon ECS recherche l'artefact de sortie généré par l'action de rendu (`TaskDefArtifact`), trouve le `task-definition-random-string.json` fichier qu'il contient et l'enregistre auprès de votre service Amazon ECS. Le service Amazon ECS suit ensuite les instructions du `task-definition-random-string.json` fichier pour exécuter les tâches Amazon ECS et les conteneurs d'images Docker associés au sein de votre cluster Amazon ECS.

```
Name: codecatalyst-ecs-workflow
SchemaVersion: 1.0
```

Triggers:

- Type: PUSH
- Branches:
  - main

Actions:

```
BuildDocker:
  Identifier: aws/build@v1
  Environment:
```

```
Name: codecatalyst-ecs-environment
Connections:
  - Name: codecatalyst-account-connection
    Role: codecatalyst-ecs-build-role
Inputs:
  Variables:
    - Name: REPOSITORY_URI
      Value: 111122223333.dkr.ecr.us-east-2.amazonaws.com/codecatalyst-ecs-image-
repo
    - Name: IMAGE_TAG
      Value: ${WorkflowSource.CommitId}
Configuration:
  Steps:
    #pre_build:
      - Run: echo Logging in to Amazon ECR...
      - Run: aws --version
      - Run: aws ecr get-login-password --region us-east-2 | docker login --username
AWS --password-stdin 111122223333.dkr.ecr.us-east-2.amazonaws.com
    #build:
      - Run: echo Build started on `date`
      - Run: echo Building the Docker image...
      - Run: docker build -t $REPOSITORY_URI:latest .
      - Run: docker tag $REPOSITORY_URI:latest $REPOSITORY_URI:$IMAGE_TAG
    #post_build:
      - Run: echo Build completed on `date`
      - Run: echo Pushing the Docker images...
      - Run: docker push $REPOSITORY_URI:latest
      - Run: docker push $REPOSITORY_URI:$IMAGE_TAG

RenderTaskDef:
  DependsOn:
    - BuildDocker
  Identifier: aws/ecs-render-task-definition@v1
  Inputs:
    Variables:
      - Name: REPOSITORY_URI
        Value: 111122223333.dkr.ecr.us-east-2.amazonaws.com/codecatalyst-ecs-image-
repo
      - Name: IMAGE_TAG
        Value: ${WorkflowSource.CommitId}
  Configuration:
    task-definition: taskdef.json
    container-definition-name: codecatalyst-ecs-container
    image: $REPOSITORY_URI:$IMAGE_TAG
```

```
# The output artifact contains the updated task definition file.
# The new file is prefixed with 'task-definition'.
# The output variable is set to the name of the updated task definition file.
Outputs:
  Artifacts:
    - Name: TaskDefArtifact
      Files:
        - "task-definition*"
  Variables:
    - task-definition

DeployToECS:
  Identifier: aws/ecs-deploy@v1
  Environment:
    Name: codecatalyst-ecs-environment
  Connections:
    - Name: codecatalyst-account-connection
      Role: codecatalyst-ecs-deploy-role
  #Input artifact contains the updated task definition file.
  Inputs:
    Sources: []
    Artifacts:
      - TaskDefArtifact
  Configuration:
    region: us-east-2
    cluster: codecatalyst-ecs-cluster
    service: codecatalyst-ecs-service
    task-definition: ${RenderTaskDef.task-definition}
```

## Ajout de l'action « Render la définition de tâche Amazon ECS »

Suivez les instructions suivantes pour ajouter l'action de définition de tâche Render Amazon ECS à votre flux de travail.

### Prérequis

Avant de commencer, assurez-vous de disposer d'un flux de travail qui inclut une action de génération qui génère dynamiquement une image Docker. Reportez-vous à l'[exemple de flux de travail](#) précédent pour plus de détails.

## Visual

Pour ajouter l'action « Render la définition de tâche Amazon ECS » à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
  2. Choisissez votre projet.
  3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
  4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
  5. Choisissez Edit (Modifier).
  6. Choisissez Visual.
  7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
  8. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst.
  9. Recherchez l'action de définition de tâche Render Amazon ECS, puis effectuez l'une des opérations suivantes :
    - Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.
- Ou
- Choisissez la définition de tâche Render Amazon ECS. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
    - (Facultatif) Choisissez Afficher la source pour [afficher le code source de l'action](#).
    - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.
10. Dans les onglets Entrées et Configuration, complétez les champs en fonction de vos besoins. Pour obtenir une description de chaque champ, consultez le [Référence d'action « Render la définition de la tâche Amazon ECS »](#). Cette référence fournit des informations détaillées sur chaque champ (et la valeur de propriété YAML correspondante) tel qu'il apparaît dans les éditeurs YAML et visuels.
  11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
  12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.



## YAML

Pour ajouter l'action « Render la définition de tâche Amazon ECS » à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
  2. Choisissez votre projet.
  3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
  4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
  5. Choisissez Edit (Modifier).
  6. Choisissez YAML.
  7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
  8. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst.
  9. Recherchez l'action de définition de tâche Render Amazon ECS, puis effectuez l'une des opérations suivantes :
    - Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.
- Ou
- Choisissez la définition de tâche Render Amazon ECS. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
    - (Facultatif) Choisissez Afficher la source pour [afficher le code source de l'action](#).
    - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.
10. Modifiez les propriétés du code YAML en fonction de vos besoins. Une explication de chaque propriété disponible est fournie dans le [Référence d'action « Render la définition de la tâche Amazon ECS »](#).
  11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
  12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Étapes suivantes

Après avoir ajouté l'action de rendu, ajoutez l'action Deploy to Amazon ECS à votre flux de travail en suivant les instructions de [Ajout de l'action « Déployer sur Amazon ECS »](#). Lors de l'ajout de l'action de déploiement, procédez comme suit :

1. Dans l'onglet Entrées de l'action de déploiement, dans Artefacts - facultatif, sélectionnez l'artefact généré par l'action de rendu. Il contient le fichier de définition de tâche mis à jour.

Pour plus d'informations sur les artefacts, consultez [Travailler avec des artefacts](#).

2. Dans l'onglet Configuration de l'action de déploiement, dans le champ Définition de la tâche, spécifiez la variable d'action suivante : `${action-name.task-definition}` où *action-name* est le nom de votre action de rendu, par exemple, `RenderTaskDef`. L'action de rendu attribue à cette variable le nouveau nom du fichier de définition de tâche.

Pour plus d'informations sur les variables, consultez [Utilisation des variables](#).

Pour plus d'informations sur la configuration de l'action de déploiement, consultez l'[exemple de flux](#) de travail précédent.

## Affichage du fichier de définition de tâche mis à jour

Vous pouvez consulter le nom et le contenu du fichier de définition de tâche mis à jour.

Pour afficher le nom du fichier de définition de tâche mis à jour, une fois que l'action de définition de tâche Render Amazon ECS l'a traité.

1. Trouvez l'exécution qui inclut une action de rendu terminée :
  - a. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
  - b. Choisissez votre projet.
  - c. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
  - d. Choisissez le nom du flux de travail contenant l'action de rendu. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
  - e. Choisissez une exécution qui inclut l'action de rendu terminée.
2. Dans le diagramme du flux de travail, choisissez l'action de rendu.
3. Choisissez Sorties.

4. Choisissez Variables.
5. Le nom du fichier de définition de tâche s'affiche. Cela ressemble à `task-definition--259-0a2r7gx1TF5X-.json`.

Pour afficher le contenu du fichier de définition de tâche mis à jour

1. Trouvez l'exécution qui inclut une action de rendu terminée :
  - a. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
  - b. Choisissez votre projet.
  - c. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
  - d. Choisissez le nom du flux de travail contenant l'action de rendu. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
  - e. Choisissez une exécution qui inclut l'action de rendu terminée.
2. Lors de l'exécution du flux de travail, en haut, à côté de Visual et YAML, choisissez les sorties du flux de travail.
3. Dans la section Artefacts, choisissez Télécharger à côté de l'artefact contenant le fichier de définition de tâche mis à jour. La colonne Produit par sera définie sur le nom de votre action de rendu pour cet artefact.
4. Ouvrez le fichier .zip pour afficher le fichier .json de définition de tâche.

Variables produites par l'action « Rendre la définition de la tâche Amazon ECS »

Lorsque l'action de définition de tâche Render Amazon ECS est exécutée, elle produit des variables que vous pouvez utiliser dans les actions de flux de travail suivantes. Pour plus de détails, voir [Variables d'action « Afficher la définition de la tâche Amazon ECS »](#) dans le [Liste des variables prédéfinies](#).

Définition de l'action « Rendre la définition de la tâche Amazon ECS »

L'action de définition de tâche Render Amazon ECS est définie comme un ensemble de propriétés YAML dans votre fichier de définition de flux de travail. Pour plus d'informations sur ces propriétés, consultez [Référence d'action « Render la définition de la tâche Amazon ECS »](#) le [Référence de définition du flux de travail](#).

## Ajouter des GitHub actions

Une GitHub action ressemble beaucoup à une [CodeCatalyst action](#), sauf qu'elle a été développée pour être utilisée avec des GitHub flux de travail. Pour plus de détails sur GitHub les actions, consultez la documentation sur [GitHub les actions](#).

Vous pouvez utiliser une GitHub action en même temps que CodeCatalyst des actions natives dans un CodeCatalyst flux de travail.

Il existe deux manières d'ajouter une GitHub action à un CodeCatalyst flux de travail :

- Vous pouvez sélectionner l' GitHub action dans une liste organisée dans la CodeCatalyst console. Plusieurs GitHub actions populaires sont disponibles. Pour plus d'informations, consultez [Ajouter une action organisée GitHub](#) .
- Si l' GitHub action que vous souhaitez utiliser n'est pas disponible dans la CodeCatalyst console, vous pouvez l'ajouter à l'aide d'une action GitHub Actions.

Une action GitHub Actions est une CodeCatalyst action qui enveloppe une GitHub action et la rend compatible avec les CodeCatalyst flux de travail.

Voici un exemple d'action GitHub Actions encapsulant l'action [Super-Linter](#) : GitHub

```
Actions:
  GitHubAction:
    Identifier: aws/github-actions-runner@v1
    Configuration:
      Steps:
        - name: Lint Code Base
          uses: github/super-linter@v4
          env:
            VALIDATE_ALL_CODEBASE: "true"
            DEFAULT_BRANCH: main
```

Dans le code précédent, l'action CodeCatalyst GitHub Actions (identifiée par `aws/github-actions-runner@v1`) enveloppe l'action Super-Linter (identifiée par `github/super-linter@v4`), la faisant fonctionner dans un flux de travail. CodeCatalyst

Pour plus d'informations, consultez [Ajouter l'action « GitHub Actions »](#).

Toutes les GitHub actions, qu'elles soient organisées ou non, doivent être intégrées à une action GitHub Actions (`aws/github-actions-runner@v1`), comme indiqué dans l'exemple précédent. Le wrapper est nécessaire au bon fonctionnement de l'action.

## Rubriques

- [En quoi les GitHub actions diffèrent-elles des CodeCatalyst actions ?](#)
- [Les GitHub actions peuvent-elles interagir avec d'autres CodeCatalyst actions du flux de travail ?](#)
- [Quelles GitHub actions puis-je utiliser ?](#)
- [Limites des GitHub actions dans CodeCatalyst](#)
- [Comment ajouter une GitHub action \(étapes de haut niveau\) ?](#)
- [L' GitHub action se déroule-t-elle GitHub ?](#)
- [Puis-je également utiliser des GitHub flux de travail ?](#)
- [Tutoriel : code Lint à l'aide d'une action GitHub](#)
- [Ajouter l'action « GitHub Actions »](#)
- [Ajouter une action organisée GitHub](#)
- [Utilisation des paramètres de sortie des GitHub actions](#)

En quoi les GitHub actions diffèrent-elles des CodeCatalyst actions ?

GitHub Les actions utilisées dans un CodeCatalyst flux de travail n'ont pas le même niveau d'accès, d'intégration AWS et de CodeCatalyst fonctionnalités (telles que [les environnements](#) et [les problèmes](#)) que CodeCatalyst les actions.

Les GitHub actions peuvent-elles interagir avec d'autres CodeCatalyst actions du flux de travail ?

Oui. Par exemple, GitHub les actions peuvent utiliser des variables produites par d'autres CodeCatalyst actions comme entrée, et peuvent également partager des paramètres de sortie et des artefacts avec CodeCatalyst des actions. Pour plus d'informations, consultez [Utilisation des paramètres de sortie des GitHub actions](#).

Quelles GitHub actions puis-je utiliser ?

Vous pouvez utiliser n'importe quelle GitHub action disponible via la CodeCatalyst console et n'importe quelle GitHub action disponible [GitHub sur le Marketplace](#). Si vous décidez d'utiliser une GitHub Action depuis le Marketplace, gardez à l'esprit les [limites](#) suivantes.

## Limites des GitHub actions dans CodeCatalyst

- GitHub Les actions ne peuvent pas être utilisées avec le type de [calcul CodeCatalyst Lambda](#).
- GitHub Les actions s'exécutent sur l'image Docker de l'environnement d'exécution de [novembre 2022](#), qui inclut des outils plus anciens. Pour plus d'informations sur l'image et l'outillage, consultez [Utilisation des images Docker de l'environnement d'exécution](#).
- GitHub Les actions qui s'appuient en interne sur le [githubcontexte](#) ou qui font référence à GitHub des ressources spécifiques ne fonctionneront pas. CodeCatalyst Par exemple, les actions suivantes ne fonctionneront pas dans CodeCatalyst :
  - Actions visant à ajouter, modifier ou mettre à jour GitHub des ressources. Les exemples incluent les actions qui mettent à jour les pull requests ou créent des problèmes dans GitHub.
  - Presque toutes les actions sont répertoriées [sur https://github.com/actions](https://github.com/actions).
- GitHub Les actions qui sont des [actions de conteneur Docker](#) fonctionneront, mais elles doivent être exécutées par l'utilisateur Docker par défaut (root). N'exécutez pas l'action en tant qu'utilisateur 1001. (Au moment de la rédaction de cet article, l'utilisateur 1001 travaille dans GitHub, mais pas dans CodeCatalyst.) Pour plus d'informations, consultez la rubrique [UTILISATEUR](#) dans le [support Dockerfile pour les GitHub actions](#).

Pour obtenir la liste des GitHub actions disponibles via la CodeCatalyst console, consultez [Ajouter une action organisée GitHub](#) .

Comment ajouter une GitHub action (étapes de haut niveau) ?

Les étapes de haut niveau pour ajouter une GitHub action à un CodeCatalyst flux de travail sont les suivantes :

1. Dans votre CodeCatalyst projet, vous créez un flux de travail. Le flux de travail vous permet de définir comment créer, tester et déployer votre application. Pour plus d'informations, consultez [Commencer à utiliser les flux de travail dans CodeCatalyst](#).
2. Dans le flux de travail, vous ajoutez une GitHub action organisée ou vous ajoutez l'action GitHub Actions.
3. Vous devez effectuer l'une des opérations suivantes :
  - Si vous avez choisi d'ajouter une action organisée, configurez-la. Pour plus d'informations, consultez [Ajouter une action organisée GitHub](#) .

- Si vous avez choisi d'ajouter une action non organisée, dans l'action GitHubActions, vous collez le code YAML de l' GitHub action. Vous trouverez ce code sur la page détaillée de l' GitHubaction que vous avez choisie [GitHubsur le Marketplace](#). Vous devrez probablement modifier légèrement le code pour qu'il fonctionne CodeCatalyst. Pour plus d'informations, consultez [Ajouter l'action « GitHub Actions »](#).
4. (Facultatif) Dans le flux de travail, vous ajoutez d'autres actions, telles que les actions de création et de test. Pour plus d'informations, consultez [Créez, testez et déployez avec des flux de travail dans CodeCatalyst](#).
  5. Vous démarrez le flux de travail manuellement ou automatiquement par le biais d'un déclencheur. Le flux de travail exécute l' GitHub action et toutes les autres actions du flux de travail. Pour plus d'informations, consultez [Démarrage d'une exécution de flux de travail](#).

Pour connaître les étapes détaillées, voir :

- [Ajouter une action organisée GitHub](#) .
- [Ajouter l'action « GitHub Actions »](#).

L' GitHub action se déroule-t-elle GitHub ?

Non. L' GitHub action s'exécute CodeCatalyst en utilisant les [machines CodeCatalyst de construction](#).

Puis-je également utiliser des GitHub flux de travail ?

Non.

Tutoriel : code Lint à l'aide d'une action GitHub

Dans ce didacticiel, vous allez ajouter l' [GitHub action Super-Linter à un flux](#) de travail Amazon CodeCatalyst . L'action Super-Linter inspecte le code, détecte les zones dans lesquelles le code contient des erreurs, des problèmes de formatage et des constructions suspectes, puis affiche les résultats sur la console). CodeCatalyst Après avoir ajouté le linter à votre flux de travail, vous exécutez le flux de travail pour lint un exemple d'application Node.js ()app . js. Vous corrigez ensuite les problèmes signalés et réexécutez le flux de travail pour voir si les correctifs ont fonctionné.

**i** Tip

Envisagez d'utiliser Super-Linter pour colorer des fichiers YAML, tels que des modèles. [AWS CloudFormation](#)

## Rubriques

- [Prérequis](#)
- [Étape 1 : Création d'un référentiel source](#)
- [Étape 2 : Ajouter un fichier app.js](#)
- [Étape 3 : créer un flux de travail qui exécute l'action Super-Linter](#)
- [Étape 4 : Résoudre les problèmes détectés par le Super-Linter](#)
- [Nettoyage](#)

## Prérequis

Avant de commencer, vous aurez besoin des éléments suivants :

- Un CodeCatalyst espace connecté [Compte AWS](#). Pour en savoir plus, consultez [Création d'un espace prenant en charge les utilisateurs de AWS Builder ID](#).
- Un projet vide dans votre CodeCatalyst espace appelé `codecatalyst-linter-project`. Choisissez l'option Partir de zéro pour créer ce projet.

Pour en savoir plus, consultez [Création d'un projet vide dans Amazon CodeCatalyst](#).

## Étape 1 : Création d'un référentiel source

Au cours de cette étape, vous créez un référentiel source dans CodeCatalyst. Vous allez utiliser ce référentiel pour stocker l'exemple de fichier source de l'application `app.js`, dans le cadre de ce didacticiel.

Pour plus d'informations sur les référentiels sources, consultez [Création d'un référentiel source](#).

Pour créer un référentiel de sources

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).



2. Accédez à votre projet, `codecatalyst-linter-project`.
3. Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources.
4. Choisissez Ajouter un référentiel, puis sélectionnez Créer un référentiel.
5. Dans Nom du référentiel, entrez :

```
codecatalyst-linter-source-repository
```

6. Sélectionnez Create (Créer).

## Étape 2 : Ajouter un fichier `app.js`

Au cours de cette étape, vous ajoutez un `app.js` fichier à votre référentiel source. Le `app.js` contient un code de fonction contenant quelques erreurs que le linter trouvera.

Pour ajouter le fichier `app.js`

1. Dans la CodeCatalyst console, choisissez votre projet, `codecatalyst-linter-project`.
2. Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources.
3. Dans la liste des référentiels sources, choisissez votre référentiel, `codecatalyst-linter-source-repository`.
4. Dans Fichiers, choisissez Créer un fichier.
5. Dans la zone de texte, entrez le code suivant :

```
// const axios = require('axios')
// const url = 'http://checkip.amazonaws.com/';
let response;
/**
 *
 * Event doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html#api-gateway-simple-proxy-for-lambda-input-format
 * @param {Object} event - API Gateway Lambda Proxy Input Format
 *
 * Context doc: https://docs.aws.amazon.com/lambda/latest/dg/nodejs-prog-model-context.html
 * @param {Object} context
 *
 * Return doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html
 * @returns {Object} object - API Gateway Lambda Proxy Output Format
```

```
*
*/
exports.lambdaHandler = async (event, context) => {
  try {
    // const ret = await axios(url);
    response = {
      statusCode: 200,
      'body': JSON.stringify({
        message: 'hello world'
        // location: ret.data.trim()
      })
    }
  } catch (err) {
    console.log(err)
    return err
  }

  return response
}
```

6. Dans Nom du fichier, entrez `app.js`. Conservez les autres options par défaut.
7. Choisissez Commit (Valider).

Vous venez de créer un fichier appelé `app.js`.

### Étape 3 : créer un flux de travail qui exécute l'action Super-Linter

Au cours de cette étape, vous créez un flux de travail qui exécute l'action Super-Linter lorsque vous envoyez du code vers votre référentiel source. Le flux de travail comprend les éléments de base suivants, que vous définissez dans un fichier YAML :

- Un déclencheur : ce déclencheur lance automatiquement l'exécution du flux de travail lorsque vous apportez une modification à votre référentiel source. Pour plus d'informations sur les déclencheurs, consultez [Utilisation de déclencheurs](#).
- Une action « GitHub Actions » — Lorsqu'elle est déclenchée, l'action GitHub Actions exécute l'action Super-Linter, qui à son tour inspecte tous les fichiers de votre référentiel source. Si le linter détecte un problème, l'action du flux de travail échoue.

## Pour créer un flux de travail qui exécute l'action Super-Linter

1. Dans la CodeCatalyst console, choisissez votre projet, `codecatalyst-linter-project`.
2. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
3. Choisissez Créer un flux de travail.
4. Pour Référentiel source, choisissez `codecatalyst-linter-source-repository`.
5. Pour Branch, choisissez `main`.
6. Sélectionnez Create (Créer).
7. Supprimez l'exemple de code YAML.
8. Ajoutez le code YAML suivant :

```
Name: codecatalyst-linter-workflow
SchemaVersion: "1.0"
Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  SuperLinterAction:
    Identifiant: aws/github-actions-runner@v1
    Configuration:
      Steps:
        github-action-code
```

Dans le code précédent, remplacez-le *github-action-code* par le code d'action Super-Linter, comme indiqué dans les étapes suivantes de cette procédure.

9. Accédez à la [page Super-Linter sur le Marketplace](#). GitHub
10. Sous `steps` : (minuscules), recherchez le code et collez-le dans le CodeCatalyst flux de travail situé sous `Steps` : (majuscules).

Ajustez le code GitHub d'action pour qu'il soit conforme aux CodeCatalyst normes, comme indiqué dans le code suivant.

Votre CodeCatalyst flux de travail ressemble désormais à ceci :

```
Name: codecatalyst-linter-workflow
SchemaVersion: "1.0"
Triggers:
```

```
- Type: PUSH
  Branches:
    - main
  Actions:
    SuperLintAction:
      Identifiant: aws/github-actions-runner@v1
      Configuration:
        Steps:
          - name: Lint Code Base
            uses: github/super-linter@v4
            env:
              VALIDATE_ALL_CODEBASE: "true"
              DEFAULT_BRANCH: main
```

11. (Facultatif) Choisissez Valider pour vous assurer que le code YAML est valide avant de valider.
12. Choisissez Valider, entrez un message de validation, sélectionnez votre `codecatalyst-linter-source-repository` référentiel, puis choisissez à nouveau Valider.

Vous venez de créer un flux de travail. L'exécution d'un flux de travail démarre automatiquement en raison du déclencheur défini en haut du flux de travail.

Pour consulter le flux de travail en cours d'exécution

1. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
2. Choisissez le flux de travail que vous venez de créer `:codecatalyst-linter-workflow`.
3. Dans le diagramme du flux de travail, choisissez SuperLintAction.
4. Attendez que l'action échoue. Cet échec est attendu car le linter a détecté des problèmes dans le code.
5. Laissez la CodeCatalyst console ouverte et accédez à [Étape 4 : Résoudre les problèmes détectés par le Super-Linter](#).

#### Étape 4 : Résoudre les problèmes détectés par le Super-Linter

Le Super-Linter devrait avoir détecté des problèmes dans le `app.js` code, ainsi que dans le `README.md` fichier inclus dans votre dépôt source.

Pour résoudre les problèmes détectés par le linter

1. Dans la CodeCatalyst console, choisissez l'onglet Logs, puis Lint Code Base.

Les journaux générés par l'action Super-Linter sont affichés.

2. Dans les journaux de Super-Linter, faites défiler l'écran vers le bas jusqu'à la ligne 90, où vous trouverez le début des problèmes. Ils ressemblent à ce qui suit :

```
/github/workspace/hello-world/app.js:3:13: Extra semicolon.  
/github/workspace/hello-world/app.js:9:92: Trailing spaces not allowed.  
/github/workspace/hello-world/app.js:21:7: Unnecessarily quoted property 'body'  
found.  
/github/workspace/hello-world/app.js:31:1: Expected indentation of 2 spaces but  
found 4.  
/github/workspace/hello-world/app.js:32:2: Newline required at end of file but not  
found.
```

3. Corrigez `app.js` et `README.md` dans votre référentiel source et validez vos modifications.

#### Tip

Pour corriger le problème `README.md`, ajoutez-le markdown au bloc de code, comme ceci :

```
```markdown  
Setup examples:  
...  
```
```

Vos modifications démarrent automatiquement un autre flux de travail. Attendez que le flux de travail soit terminé. Si vous avez résolu tous les problèmes, le flux de travail devrait réussir.

## Nettoyage

Nettoyez CodeCatalyst pour supprimer les traces de ce didacticiel dans votre environnement.

Pour nettoyer CodeCatalyst

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Supprimer `codecatalyst-linter-source-repository`.
3. Supprimer `codecatalyst-linter-workflow`.

Dans ce didacticiel, vous avez appris à ajouter l' GitHub action Super-Linter à un CodeCatalyst flux de travail afin de créer du code.

Ajouter l'action « GitHub Actions »

Une action GitHub Actions est une CodeCatalyst action qui enveloppe une GitHub action et la rend compatible avec les CodeCatalyst flux de travail.

Pour en savoir plus, consultez [Ajouter des GitHub actions](#).

Pour ajouter l'action GitHub Actions à un flux de travail, procédez comme suit.

 Tip

Pour consulter un didacticiel expliquant comment utiliser l'action GitHub Actions, voir [Tutoriel : code Lint à l'aide d'une action GitHub](#).

## Visual

Pour ajouter l'action « GitHub Actions » à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez Visual.
7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
8. Dans la liste déroulante, sélectionnez GitHub.
9. Recherchez l'action GitHub Actions, puis effectuez l'une des opérations suivantes :
  - Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.

Ou

- Choisissez GitHub Actions. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
  - (Facultatif) Choisissez Afficher la source pour [afficher le code source de l'action](#).
  - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.
- 10. Dans les onglets Entrées et Configuration, complétez les champs en fonction de vos besoins. Pour une description de chaque champ, consultez le [Référence GitHub d'action « Actions »](#). Cette référence fournit des informations détaillées sur chaque champ (et la valeur de propriété YAML correspondante) tel qu'il apparaît dans les éditeurs YAML et visuels.
- 11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
- 12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour ajouter l'action « GitHub Actions » à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
8. Dans la liste déroulante, sélectionnez GitHub.
9. Recherchez l'action GitHub Actions, puis effectuez l'une des opérations suivantes :
  - Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.

Ou

- Choisissez GitHub Actions. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
  - (Facultatif) Choisissez Afficher la source pour [afficher le code source de l'action](#).

- Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.
10. Modifiez les propriétés du code YAML en fonction de vos besoins. Une explication de chaque propriété disponible est fournie dans le [Référence GitHub d'action « Actions »](#).
  11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
  12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Définition de l'action « GitHub Actions »

L'action GitHub Actions est définie comme un ensemble de propriétés YAML dans votre fichier de définition de flux de travail. Pour plus d'informations sur ces propriétés, consultez [Référence GitHub d'action « Actions »](#) le [Référence de définition du flux de travail](#).

## Ajouter une action organisée GitHub

Une GitHub action organisée est une GitHub action mise à disposition dans la CodeCatalyst console et sert d'exemple d'utilisation d'une GitHub action dans un CodeCatalyst flux de travail.

Les GitHub actions sélectionnées sont regroupées dans l'[action GitHub Actions](#) créée par CodeCatalyst l'auteur, identifiée par l'identifiant. `aws/github-actions-runner@v1` Par exemple, voici à quoi ressemble l' GitHub action sélectionnée, [TruffleHog OSS](#) :

```
Actions:
  TruffleHogOSS_e8:
    Identifiant: aws/github-actions-runner@v1
    Inputs:
      Sources:
        - WorkflowSource # This specifies that the action requires this Workflow as a
source
    Configuration:
      Steps:
        - uses: trufflesecurity/trufflehog@v3.16.0
          with:
            path: ' ' # Required; description: Repository path
            base: ' ' # Required; description: Start scanning from here (usually main
branch).
            head: ' ' # Optional; description: Scan commits until here (usually dev
branch).
            extra_args: ' ' # Optional; description: Extra args to be passed to the
trufflehog cli.
```



Dans le code précédent, l'action CodeCatalyst GitHub Actions (identifiée par `aws/github-actions-runner@v1`) englobe l'action TruffleHog OSS (identifiée par `trufflesecurity/trufflehog@v3.16.0`), la faisant fonctionner dans un CodeCatalyst flux de travail.

Pour configurer cette action, vous devez remplacer les chaînes vides ci-dessous `with:` par vos propres valeurs. Par exemple :

```
Actions:
  TruffleHogOSS_e8:
    Identifiant: aws/github-actions-runner@v1
    Inputs:
      Sources:
        - WorkflowSource # This specifies that the action requires this Workflow as a
source
    Configuration:
      Steps:
        - uses: trufflesecurity/trufflehog@v3.16.0
          with:
            path: ./
            base: main # Required; description: Start scanning from here (usually main
branch).
            head: HEAD # Optional; description: Scan commits until here (usually dev
branch).
            extra_args: '--debug --only-verified' # Optional; description: Extra args
to be passed to the trufflehog cli.
```

Pour ajouter une GitHub action sélectionnée à un flux de travail, procédez comme suit. Pour des informations générales sur l'utilisation GitHub des actions dans un CodeCatalyst flux de travail, consultez [Ajouter des GitHub actions](#).

#### Note

Si votre GitHub action ne figure pas dans la liste des actions sélectionnées, vous pouvez toujours l'ajouter à votre flux de travail à l'aide de l'action GitHub Actions. Pour en savoir plus, consultez [Ajouter l'action « GitHub Actions »](#).

## Visual

Pour ajouter une GitHub action organisée à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez Visual.
7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
8. Dans la liste déroulante, sélectionnez GitHub.
9. Parcourez ou recherchez une GitHub action, puis effectuez l'une des opérations suivantes :
  - Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.

Ou

- Choisissez le nom de l' GitHub action. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
    - (Facultatif) Choisissez Afficher la source pour [afficher le code source de l'action](#).
    - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.
10. Dans les onglets Entrées, Configuration et Sorties, complétez les champs en fonction de vos besoins. Pour une description de chaque champ, consultez le [Référence GitHub d'action « Actions »](#). Cette référence fournit des informations détaillées sur chaque champ (et la valeur de propriété YAML correspondante) disponible pour l'action GitHubActions, tel qu'il apparaît à la fois dans le YAML et dans les éditeurs visuels.

Pour plus d'informations sur les options de configuration disponibles pour l' GitHubaction sélectionnée, consultez sa documentation.

11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour ajouter une GitHub action organisée à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
8. Dans la liste déroulante, sélectionnez GitHub.
9. Parcourez ou recherchez une GitHub action, puis effectuez l'une des opérations suivantes :
  - Choisissez le signe plus (+) pour ajouter l'action au diagramme de flux de travail et ouvrir son volet de configuration.

Ou

- Choisissez le nom de l' GitHub action. La boîte de dialogue des détails de l'action apparaît. Dans cette boîte de dialogue :
    - (Facultatif) Choisissez Afficher la source pour [afficher le code source de l'action](#).
    - Choisissez Ajouter au flux de travail pour ajouter l'action au diagramme du flux de travail et ouvrir son volet de configuration.
10. Modifiez les propriétés du code YAML en fonction de vos besoins. Une explication de chaque propriété disponible pour l'action GitHub Actions est fournie dans le [Référence GitHub d'action « Actions »](#).

Pour plus d'informations sur les options de configuration disponibles pour l' GitHubaction sélectionnée, consultez sa documentation.

11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Utilisation des paramètres de sortie des GitHub actions

Vous pouvez intégrer des [paramètres GitHub de sortie](#) dans vos CodeCatalyst flux de travail.

### Note

Un autre mot pour le paramètre de sortie est variable. Comme le terme paramètre de sortie est GitHub utilisé dans sa documentation, nous utiliserons également ce terme.

## Rubriques

- [Exporter un paramètre GitHub de sortie afin que d'autres actions puissent l'utiliser](#)
- [Référencement d'un paramètre GitHub de sortie](#)

Exporter un paramètre GitHub de sortie afin que d'autres actions puissent l'utiliser

Suivez les instructions ci-dessous pour exporter un paramètre GitHub de sortie afin qu'il puisse être utilisé dans d'autres actions de CodeCatalyst flux de travail.

Pour exporter un paramètre GitHub de sortie

1. Ouvrez un flux de travail et choisissez Modifier. Pour des instructions, consultez [Pour modifier un flux de travail](#).
2. Dans l'action GitHub Actions qui génère le paramètre de sortie que vous souhaitez exporter, ajoutez une Outputs section avec une Variables propriété sous-jacente qui ressemble à ceci :

```
Actions:
  MyGitHubAction:
    Identifieur: aws/github-actions-runner@v1
    Outputs:
    Variables:
      - 'step-id_output-name'
```

Remplacer :

- *step-id* avec la valeur de la id: propriété dans la steps section de l' GitHub action.
- *output-name* avec le nom du paramètre de sortie. GitHub

## Exemple

L'exemple suivant montre comment exporter un paramètre GitHub de sortie appelé `SELECTEDCOLOR`.

```
Actions:
  MyGitHubAction:
    Identifieur: aws/github-actions-runner@v1
    Outputs:
      Variables:
        - 'random-color-generator_SELECTEDCOLOR'
    Configuration:
      Steps:
        - name: Set selected color
          run: echo "SELECTEDCOLOR=green" >> $GITHUB_OUTPUT
          id: random-color-generator
```

## Référencement d'un paramètre GitHub de sortie

Suivez les instructions ci-dessous pour référencer un paramètre GitHub de sortie.

Pour référencer un paramètre GitHub de sortie

1. Suivez les étapes de [Exporter un paramètre GitHub de sortie afin que d'autres actions puissent l'utiliser](#).

Le paramètre GitHub de sortie peut désormais être utilisé dans d'autres actions.

2. Notez la `Variables` valeur du paramètre de sortie. Il inclut un trait de soulignement (`_`).
3. Reportez-vous au paramètre de sortie à l'aide de la syntaxe suivante :

```
${action-name.output-name}
```

Remplacer :

- `action-name` avec le nom de l' `CodeCatalystGitHub` action qui produit le paramètre de sortie (n'utilisez pas le `name` ou `id` de l' `GitHub` action).
- `output-name` avec la `Variables` valeur du paramètre de sortie que vous avez notée plus tôt.

## Exemple

```
BuildActionB:
  Identifieur: aws/build@v1
  Configuration:
    Steps:
      - Run: echo ${MyGitHubAction.random-color-generator_SELECTEDCOLOR}
```

## Exemple avec contexte

L'exemple suivant montre comment définir une `SELECTEDCOLOR` variable dans `GitHubActionA`, la sortir, puis y faire référence dans `BuildActionB`.

```
Actions:
  GitHubActionA:
    Identifieur: aws/github-actions-runner@v1
    Configuration:
      Steps:
        - name: Set selected color
          run: echo "SELECTEDCOLOR=green" >> $GITHUB_OUTPUT
          id: random-color-generator
      Outputs:
        Variables:
          - 'random-color-generator_SELECTEDCOLOR'

  BuildActionB:
    Identifieur: aws/build@v1
    Configuration:
      Steps:
        - Run: echo ${GitHubActionA.random-color-generator_SELECTEDCOLOR}
```

## Supprimer une action

Suivez les instructions ci-dessous pour supprimer une action d'un flux de travail.

## Visual

Pour supprimer une action à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez Visual.
7. Dans le diagramme du flux de travail, dans l'action que vous souhaitez supprimer, cliquez sur l'icône représentant des points de suspension verticaux, puis sur Supprimer.
8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour supprimer une action à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. Recherchez la section du code YAML qui contient l'action que vous souhaitez supprimer.  
  
Sélectionnez la section et appuyez sur la touche de suppression de votre clavier.
8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Développement d'une action personnalisée

Vous pouvez développer une action personnalisée à utiliser dans vos flux de travail à l'aide du kit de développement CodeCatalyst d'actions (ADK). Vous pouvez ensuite publier l'action dans le catalogue d'actions CodeCatalyst afin que les autres utilisateurs puissent la consulter et l'utiliser dans leurs flux de travail.

Pour développer, tester et publier une action (tâches de haut niveau)

1. Installez les outils et packages nécessaires au développement d'une action.
2. Créez un référentiel CodeCatalyst pour stocker votre code d'action.
3. Initialisez l'action. Cela définit les fichiers source requis par l'action, y compris un fichier de définition d'action (`action.yml`) que vous pouvez mettre à jour avec votre propre code.
4. Démarrez le code d'action pour obtenir les outils et les bibliothèques nécessaires pour créer, tester et publier le projet d'action.
5. Créez l'action sur votre ordinateur local et transférez les modifications vers votre référentiel CodeCatalyst.
6. Testez l'action avec des tests unitaires en local et exécutez le flux de travail généré par ADK dans CodeCatalyst.
7. Publiez l'action dans le catalogue d'actions CodeCatalyst en cliquant sur le bouton Publier dans la console CodeCatalyst.

Pour connaître les étapes détaillées, consultez le [guide du développeur Amazon CodeCatalyst Action Development Kit](#).

## Regrouper les actions dans des groupes d'action

Un groupe d'actions contient une ou plusieurs actions. Le regroupement des actions dans des groupes d'actions vous aide à organiser votre flux de travail et vous permet également de configurer les dépendances entre les différents groupes d'actions.

### Note

Vous ne pouvez pas imbriquer des groupes d'actions dans d'autres actions ou groupes d'actions.



Suivez les instructions ci-dessous pour définir un groupe d'actions.

## Visual

Non disponible Choisissez YAML pour afficher les instructions YAML.

## YAML

Pour définir un groupe d'actions

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. Dans Actions, ajoutez un code similaire au suivant :

```
Actions:
  action-group-name:
    Actions:
      action-1:
        Identifiant: aws/build@v1
        Configuration:
          ...
      action-2:
        Identifiant: aws/ecs-deploy@v1
        Configuration:
          ...
```

Pour obtenir un autre exemple, consultez [Exemple : définition de deux groupes d'action](#). Pour plus d'informations, consultez la description [Actions](#) de la `action-group-name` propriété dans le [Référence de définition du flux de travail](#).

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Exemple : définition de deux groupes d'action

L'exemple suivant montre comment définir deux groupes d'actions : BuildAndTest et Deploy. Le groupe BuildAndTest d'actions comprend deux actions (Build et Test), et le groupe Deploy d'actions comprend également deux actions (DeployCloudFormationStack et DeployToECS).

```
Actions:
  BuildAndTest: # Action group 1
    Actions:
      Build:
        Identifier: aws/build@v1
        Configuration:
          ...
      Test:
        Identifier: aws/managed-test@v1
        Configuration:
          ...
  Deploy: #Action group 2
    Actions:
      DeployCloudFormationStack:
        Identifier: aws/cfn-deploy@v1
        Configuration:
          ...
      DeployToECS:
        Identifier: aws/ecs-deploy@v1
        Configuration:
          ...
```

## Configuration des actions pour qu'elles dépendent d'autres actions

Par défaut, lorsque vous ajoutez des actions à un flux de travail, elles sont ajoutées côte à côte dans [l'éditeur visuel](#). Cela signifie que les actions s'exécuteront en parallèle lorsque vous lancerez une exécution de flux de travail. Si vous souhaitez que les actions s'exécutent dans un ordre séquentiel (et apparaissent verticalement dans l'éditeur visuel), vous devez définir des dépendances entre elles. Par exemple, vous pouvez configurer une Test action pour qu'elle dépende de l'Build action afin que l'action de test soit exécutée après l'action de génération.

Vous pouvez configurer des dépendances entre les actions, entre les groupes d'actions et entre les actions et les groupes d'actions. Vous pouvez également configurer one-to-many les dépendances afin qu'une action dépende de plusieurs autres pour démarrer. Consultez le [Directives relatives à la configuration des dépendances](#) pour vous assurer que la configuration de vos dépendances est conforme à la syntaxe YAML du flux de travail.

## Rubriques

- [Configuration des dépendances](#)
- [Directives relatives à la configuration des dépendances](#)
- [Exemples](#)

## Configuration des dépendances

Utilisez les instructions suivantes pour configurer les dépendances entre les actions d'un flux de travail.

## Visual

Pour configurer les dépendances à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez Visual.
7. Dans le diagramme du flux de travail, choisissez l'action qui dépendra d'une autre.
8. Choisissez l'onglet Entrées.
9. Dans Depends on - optional, procédez comme suit :

Spécifiez une action ou un groupe d'actions qui doit être exécuté correctement pour que cette action soit exécutée.

Pour plus d'informations sur la fonctionnalité « dépend », consultez. [Configuration des actions pour qu'elles dépendent d'autres actions](#)

10. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
11. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour configurer les dépendances à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. Dans une action qui dépendra d'une autre, ajoutez un code similaire au suivant :

```
action-name:  
  DependsOn:  
    - action-1
```

Pour obtenir plus d'exemples, consultez [Exemples](#). Pour les directives générales, voir [Directives relatives à la configuration des dépendances](#). Pour plus d'informations, consultez la description de la DependsOn propriété dans le champ [Référence de définition du flux de travail](#) correspondant à votre action.

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

### Directives relatives à la configuration des dépendances

Lors de la configuration des dépendances, suivez les instructions suivantes :

- Si une action se trouve à l'intérieur d'un groupe d'actions, elle ne peut dépendre que d'autres actions au sein du même groupe d'actions.
- Les actions et les groupes d'actions peuvent dépendre d'autres actions et groupes d'actions situés au même niveau dans la hiérarchie YAML, mais pas à un niveau différent.

## Exemples

Les exemples suivants montrent comment configurer les dépendances entre les actions et les groupes d'actions dans le fichier de définition du flux de travail.

### Rubriques

- [Exemple : Configuration d'une dépendance simple](#)
- [Exemple : Configuration d'un groupe d'actions pour qu'il dépende d'une action](#)
- [Exemple : Configuration d'un groupe d'actions pour qu'il dépende d'un autre groupe d'actions](#)
- [Exemple : Configuration d'un groupe d'actions pour qu'il dépende de plusieurs actions](#)

#### Exemple : Configuration d'une dépendance simple

L'exemple suivant montre comment configurer une `Test` action pour qu'elle dépende de l'`Build` action utilisant la `DependsOn` propriété.

```
Actions:
  Build:
    Identifier: aws/build@v1
    Configuration:
      ...
  Test:
    DependsOn:
      - Build
    Identifier: aws/managed-test@v1
    Configuration:
      ...
```

#### Exemple : Configuration d'un groupe d'actions pour qu'il dépende d'une action

L'exemple suivant montre comment configurer un groupe d'`DeployGroup` actions pour qu'il dépende de l'`FirstAction` action. Notez que l'action et le groupe d'actions sont au même niveau.

```
Actions:
  FirstAction: #An action outside an action group
    Identifier: aws/github-actions-runner@v1
    Configuration:
      ...
  DeployGroup: #An action group containing two actions
    DependsOn:
```

**- FirstAction**

Actions:

DeployAction1:

...

DeployAction2:

...

Exemple : Configuration d'un groupe d'actions pour qu'il dépende d'un autre groupe d'actions

L'exemple suivant montre comment configurer un groupe d'DeployGroupactions pour qu'il dépende du groupe BuildAndTestGroup d'actions. Notez que les groupes d'action sont au même niveau.

Actions:

BuildAndTestGroup: # Action group 1

Actions:

BuildAction:

...

TestAction:

...

DeployGroup: #Action group 2

**DependsOn:****- BuildAndTestGroup**

Actions:

DeployAction1:

...

DeployAction2:

...

Exemple : Configuration d'un groupe d'actions pour qu'il dépende de plusieurs actions

L'exemple suivant montre comment configurer un groupe d'DeployGroupactions pour qu'il dépende de l'SecondActionaction, de l'action et du groupe BuildAndTestGroup d'actions. FirstAction Notez que DeployGroup c'est au même niveau que FirstActionSecondAction, etBuildAndTestGroup.

Actions:

FirstAction: #An action outside an action group

...

SecondAction: #Another action

...

BuildAndTestGroup: #Action group 1

Actions:

```
Build:
...
Test:
...
DeployGroup: #Action group 2
DependsOn:
- FirstAction
- SecondAction
- BuildAndTestGroup
Actions:
  DeployAction1:
  ...
  DeployAction2:
  ...
```

## Afficher le code source d'une action

Vous pouvez consulter le code source d'une action pour vous assurer qu'elle ne contient pas de code risqué, de failles de sécurité ou d'autres défauts.

Suivez les instructions ci-dessous pour afficher le code source d'une action [CodeCatalyst](#), d'une action de [CodeCatalyst Labs](#) ou d'une action [tiers](#).

### Note

Pour consulter le code source d'une [GitHubaction](#), rendez-vous sur la page de l'action dans le [GitHub Marketplace](#). La page inclut un lien vers le référentiel de l'action, où vous pouvez trouver le code source de l'action.

### Note

Vous ne pouvez pas afficher le code source des CodeCatalyst actions suivantes : [build](#), [test](#), [GitHub Actions](#).

### Note

AWS ne soutient ni ne garantit le code d'action des GitHub actions ou des actions de tiers.

## Pour afficher le code source d'une action

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Recherchez l'action dont vous souhaitez afficher le code :
  - a. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
  - b. Choisissez le nom de n'importe quel flux de travail ou créez-en un. Pour plus d'informations sur la création d'un flux de travail, consultez [Création, modification et suppression d'un flux de travail](#).
  - c. Choisissez Edit (Modifier).
  - d. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
  - e. Dans la liste déroulante, sélectionnez Amazon CodeCatalyst pour afficher CodeCatalyst, CodeCatalyst Labs et actions tierces.
  - f. Recherchez une action, puis choisissez son nom. Ne choisissez pas le signe plus (+).

Les détails de l'action apparaissent.

4. Dans la boîte de dialogue des détails de l'action, en bas de la page, sélectionnez Télécharger.

Une page s'affiche, indiquant le compartiment Amazon S3 dans lequel réside le code source de l'action. Pour plus d'informations sur Amazon S3, consultez [Qu'est-ce qu'Amazon S3 ?](#) dans le guide de l'utilisateur d'Amazon Simple Storage Service.

5. Inspectez le code pour vous assurer qu'il répond à vos attentes en matière de qualité et de sécurité.

## Utilisation des versions d'action

Par défaut, lorsque vous ajoutez une action à un flux de travail, Amazon CodeCatalyst ajoute la version complète au fichier de définition du flux de travail au format suivant :

*vmajor.minor.patch*

Par exemple :

```
My-Build-Action:  
  Identifiant: aws/build@v1.0.0
```



Vous pouvez raccourcir la version complète de la `Identifier` propriété afin que le flux de travail utilise toujours la dernière version mineure ou corrective de l'action.

Par exemple, si vous spécifiez :

```
My-CloudFormation-Action:  
  Identifier: aws/cfn-deploy@v1.0
```

... et la dernière version du correctif est `1.0.4`, alors l'action sera utilisée `1.0.4`. Si une version ultérieure est publiée, disons `1.0.5`, alors l'action utilisera `1.0.5`. Si une version mineure est publiée, par exemple `1.1.0`, l'action continuera à être utilisée `1.0.5`.

Pour obtenir des instructions détaillées sur la spécification des versions, consultez l'une des rubriques suivantes.

## Rubriques

- [Spécification de la version de l'action à utiliser](#)
- [Déterminer quelles versions d'une action sont disponibles](#)

## Spécification de la version de l'action à utiliser

Suivez les instructions ci-dessous pour indiquer la version d'une action que vous souhaitez utiliser dans votre flux de travail. Vous pouvez spécifier la dernière version majeure ou mineure, ou une version de correctif spécifique.

Nous vous recommandons d'utiliser la dernière version mineure ou corrective d'une action.

## Visual


Non disponible Choisissez YAML pour afficher les instructions YAML.

## YAML


Pour configurer un flux de travail afin d'utiliser la dernière version d'une action ou une version de correctif spécifique

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.

4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. Recherchez l'action dont vous souhaitez modifier la version.
8. Recherchez la `Identifier` propriété de l'action et définissez la version sur l'une des valeurs suivantes :
  - `action-identifier @v major` — Utilisez cette syntaxe pour que le flux de travail utilise une version majeure spécifique et que les dernières versions mineures et patches soient sélectionnées automatiquement.
  - `action-identifier @v major.minor` — Utilisez cette syntaxe pour que le flux de travail utilise une version mineure spécifique et que la dernière version du correctif soit sélectionnée automatiquement.
  - `action-identifier @v major.mineur.patch` — Utilisez cette syntaxe pour que le flux de travail utilise une version de correctif spécifique.

 Note

Si vous ne savez pas quelles versions sont disponibles, consultez [Déterminer quelles versions d'une action sont disponibles](#).

 Note

Vous ne pouvez pas omettre la version majeure.

9. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
10. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Déterminer quelles versions d'une action sont disponibles

Suivez les instructions suivantes pour déterminer les versions d'une action que vous pouvez utiliser dans un flux de travail.

## Visual

Pour déterminer quelles versions d'action sont disponibles

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Recherchez l'action dont vous souhaitez consulter les versions :
  - a. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
  - b. Choisissez le nom de n'importe quel flux de travail ou créez-en un. Pour plus d'informations sur la création d'un flux de travail, consultez [Création, modification et suppression d'un flux de travail](#).
  - c. Choisissez Edit (Modifier).
  - d. En haut à gauche, choisissez + Actions pour ouvrir le catalogue d'actions.
  - e. Dans la liste déroulante, choisissez Amazon CodeCatalyst pour afficher CodeCatalyst, CodeCatalyst Labs et actions tierces, ou choisissez GitHub pour afficher des GitHub actions sélectionnées.
  - f. Recherchez une action et choisissez son nom. Ne choisissez pas le signe plus (+).

Les détails de l'action apparaissent.

4. Dans la boîte de dialogue des détails de l'action, en haut à droite, choisissez la liste déroulante Versions pour afficher la liste des versions disponibles de l'action.

## YAML

Non disponible Choisissez « visuel » pour afficher les instructions de l'éditeur visuel.

## Travailler avec des artefacts

Un artefact est le résultat d'une action de flux de travail et consiste généralement en un dossier ou une archive de fichiers. Les artefacts sont importants car ils vous permettent de partager des fichiers et des informations entre les actions.

Par exemple, vous pouvez avoir une action de génération qui génère un `sample.yml` fichier, mais vous souhaitez qu'une action de déploiement l'utilise. Dans ce scénario, vous utiliseriez un artefact pour permettre à l'action de génération de partager le `sample.yml` fichier avec l'action de déploiement. Le code pourrait ressembler à ceci :

```
Actions:
  BuildAction:
    Identifier: aws/build@v1
    Steps:
      - Run: sam package --output-template-file sam-template.yml
    Outputs:
      Artifacts:
        - Name: MYARTIFACT
          Files:
            - sam-template.yml
  DeployAction:
    Identifier: aws/cfn-deploy@v1
    Inputs:
      Artifacts:
        - MYARTIFACT
    Configuration:
      template: sam-template.yml
```

Dans le code précédent, l'action de construction (BuildAction) génère un `sam-template.yml` fichier, puis l'ajoute à un artefact de sortie appelé MYARTIFACT. Une action de déploiement ultérieure (DeployAction) est spécifiée MYARTIFACT en entrée, lui donnant accès au `sam-template.yml` fichier.

## Rubriques

- [Puis-je partager des artefacts sans les spécifier en tant que sorties ou entrées ?](#)
- [Puis-je partager des artefacts entre les flux de travail ?](#)
- [Définition d'un artefact de sortie](#)
- [Définition d'un artefact d'entrée](#)
- [Référencement de fichiers dans un artefact](#)
- [Téléchargement d'artefacts](#)
- [Exemples](#)

## Puis-je partager des artefacts sans les spécifier en tant que sorties ou entrées ?

Oui, vous pouvez partager des artefacts entre des actions sans les spécifier dans les `Inputs` sections `Outputs` et du code YAML de vos actions. Pour ce faire, vous devez activer le partage de calcul. Pour plus d'informations sur le partage de calcul et sur la manière de spécifier des artefacts lorsqu'il est activé, consultez [Partage du calcul entre les actions](#).

**Note**

Bien que la fonctionnalité de partage de calcul vous permette de simplifier le code YAML de votre flux de travail en éliminant le besoin de Inputs sections Outputs et, cette fonctionnalité présente des limites que vous devez connaître avant de l'activer. Pour plus d'informations sur ces limitations, consultez [Considérations relatives au partage du calcul](#).

## Puis-je partager des artefacts entre les flux de travail ?

Non, vous ne pouvez pas partager d'artefacts entre différents flux de travail ; toutefois, vous pouvez partager des artefacts entre des actions au sein d'un même flux de travail.

## Définition d'un artefact de sortie

Utilisez les instructions suivantes pour définir un artefact que vous souhaitez qu'une action génère. Cet artefact devient alors disponible pour d'autres actions.

**Note**

Toutes les actions ne prennent pas en charge les artefacts de sortie. Pour déterminer si votre action les prend en charge, parcourez les instructions de l'éditeur visuel qui suivent et vérifiez si l'action inclut un bouton Artefacts de sortie dans l'onglet Sorties. Dans l'affirmative, les artefacts de sortie sont pris en charge.

## Visual

Pour définir un artefact de sortie à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez Visual.

7. Dans le diagramme du flux de travail, choisissez l'action qui produira l'artefact.
8. Choisissez l'onglet Outputs.
9. Sous Artefacts, choisissez Ajouter un artefact.
10. Choisissez Ajouter un artefact, puis entrez les informations dans les champs, comme suit.

#### Nom de l'artefact de construction

Spécifiez le nom d'un artefact généré par l'action. Les noms d'artefact doivent être uniques dans un flux de travail et sont limités aux caractères alphanumériques (a-z, A-Z, 0-9) et aux traits de soulignement (\_). Les espaces, les tirets (-) et les autres caractères spéciaux ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les espaces, les tirets et autres caractères spéciaux dans les noms d'artefacts en sortie.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

#### Fichiers produits par build

Spécifiez les fichiers CodeCatalyst inclus dans l'artefact généré par l'action. Ces fichiers sont générés par l'action du flux de travail lorsqu'elle s'exécute et sont également disponibles dans votre référentiel source. Les chemins de fichiers peuvent résider dans un référentiel source ou dans un artefact issu d'une action précédente, et sont relatifs au référentiel source ou à la racine de l'artefact. Vous pouvez utiliser des modèles globulaires pour définir des chemins.

Exemples :

- Pour spécifier un seul fichier situé à la racine de l'emplacement de votre build ou de votre référentiel source, utilisez `my-file.jar`.
- Pour spécifier un seul fichier dans un sous-répertoire, utilisez `directory/my-file.jar` ou `directory/subdirectory/my-file.jar`.
- Pour spécifier tous les fichiers, utilisez `**/*`. Le modèle `** glob` indique qu'il doit correspondre à un nombre quelconque de sous-répertoires.
- Pour spécifier tous les fichiers et répertoires d'un répertoire nommé `directory`, utilisez `directory/**/*`. Le modèle `** glob` indique qu'il doit correspondre à un nombre quelconque de sous-répertoires.
- Pour spécifier tous les fichiers d'un répertoire nommé `directory`, mais aucun de ses sous-répertoires, utilisez `directory/*`.

**Note**

Si le chemin de votre fichier comporte un ou plusieurs astérisques (\*) ou autres caractères spéciaux, mettez-le entre guillemets (). "" Pour plus d'informations sur les caractères spéciaux, consultez [Consignes et conventions de syntaxe](#).

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

**Note**

Vous devrez peut-être ajouter un préfixe au chemin du fichier pour indiquer dans quel artefact ou dans quelle source le trouver. Pour plus d'informations, consultez [Référencement de fichiers dans un référentiel source](#) et [Référencement de fichiers dans un artefact](#).

11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour définir un artefact de sortie à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez YAML.
7. Dans une action de flux de travail, ajoutez un code similaire au suivant :

```
action-name:  
  Outputs:
```

```
Artifacts:
  - Name: artifact-name
    Files:
      - file-path-1
      - file-path-2
```

Pour obtenir plus d'exemples, consultez [Exemples](#). Pour plus d'informations, consultez le [Référence de définition du flux de travail](#) correspondant à votre action.

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Définition d'un artefact d'entrée

Si vous souhaitez utiliser un artefact généré par une autre action, vous devez le spécifier comme entrée pour l'action en cours. Vous pouvez peut-être spécifier plusieurs artefacts en entrée, cela dépend de l'action. Pour plus d'informations, consultez le [Référence de définition du flux de travail](#) correspondant à votre action.

### Note

Vous ne pouvez pas faire référence à des artefacts provenant d'autres flux de travail.

Suivez les instructions ci-dessous pour spécifier un artefact issu d'une autre action en tant qu'entrée de l'action en cours.

### Prérequis

Avant de commencer, assurez-vous d'avoir généré l'artefact issu de l'autre action. Pour plus d'informations, consultez [Définition d'un artefact de sortie](#). La sortie de l'artefact le rend disponible pour d'autres actions.

### Visual

Pour spécifier un artefact en tant qu'entrée d'une action (éditeur visuel)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.



4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez Visual.
7. Dans le diagramme du flux de travail, choisissez l'action dans laquelle vous souhaitez spécifier un artefact en entrée.
8. Choisissez Entrées.
9. Dans Artefacts - optional, procédez comme suit :

Spécifiez les artefacts des actions précédentes que vous souhaitez fournir en entrée pour cette action. Ces artefacts doivent déjà être définis en tant qu'artefacts de sortie dans les actions précédentes.

Si vous ne spécifiez aucun artefact d'entrée, vous devez spécifier au moins un référentiel source sous `action-name/Inputs/Sources`.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

#### Note

Si la liste déroulante Artéfacts - optionnelle n'est pas disponible (éditeur visuel), ou si vous recevez des erreurs lors de la validation de votre YAML (éditeur YAML), c'est peut-être parce que l'action ne prend en charge qu'une seule entrée. Dans ce cas, essayez de supprimer l'entrée source.

10. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
11. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour spécifier un artefact en entrée d'une action (éditeur YAML)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.

4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez YAML.
7. Dans l'action dans laquelle vous souhaitez spécifier l'artefact en entrée, ajoutez un code similaire au suivant :

```
action-name:  
  Inputs:  
    Artifacts:  
      - artifact-name
```

Pour obtenir plus d'exemples, consultez [Exemples](#).

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Référencement de fichiers dans un artefact

Si un fichier se trouve dans un artefact et que vous devez y faire référence dans l'une des actions de votre flux de travail, suivez la procédure ci-dessous.

### Note

Voir aussi [Référencement de fichiers dans un référentiel source](#).

## Pour référencer des fichiers dans un artefact

- Dans l'action dans laquelle vous souhaitez référencer un fichier, ajoutez un code similaire au suivant :

```
Actions:  
  My-action:  
    Inputs:  
      Sources:  
        - WorkflowSource  
    Artifacts:  
      - artifact-name
```

Configuration:

Steps:

```
- run: cd $CATALYST_SOURCE_DIR_artifact-name/build-output && cat file.txt
```

Dans le code précédent, l'action recherche dans le build-output répertoire de l'*artefact nom de l'artefact* pour trouver et afficher le fichier. `file.txt`

Pour obtenir plus d'exemples, consultez [Exemples](#).

### Note

Vous pouvez peut-être omettre le `$CATALYST_SOURCE_DIR_artifact-name/` préfixe en fonction de la façon dont vous avez configuré votre action. Pour plus d'informations, consultez les instructions suivantes.

Conseils sur la façon de faire référence aux variables :

- Si votre action inclut un seul élément en dessous Inputs (par exemple, elle inclut un artefact d'entrée et aucune source), vous pouvez omettre le préfixe et spécifier uniquement le chemin du fichier par rapport à la racine de l'artefact.
- Vous pouvez également omettre le préfixe si le fichier se trouve dans l'entrée principale. L'entrée principale est soit le `WorkflowSource`, soit le premier artefact d'entrée répertorié, s'il n'y en a pas `WorkflowSource`.
- Le préfixe peut être différent en fonction de l'action que vous utilisez. Pour plus d'informations, consultez le tableau suivant.

| Type d'action  | Préfixe de chemin de fichier à utiliser                   | Exemple   |
|--|---|---|
| <a href="#">Créez une action</a> , <a href="#">testez une action</a> | <code>\$CATALYST_SOURCE_DIR_<i>artifact-name</i> /</code> | <code>\$CATALYST_SOURCE_DIR_MyArtifact/foolder1/file.txt</code> |
| Toutes les autres actions  | <code>\$CATALYST_SOURCE_DIR_<i>artifact-name</i> /</code> | <code>\$CATALYST_SOURCE_DIR_MyArtifact/foolder1/file.txt</code> |

| Type d'action | Préfixe de chemin de fichier à utiliser   | Exemple   |
|---------------|---|---|
|               | or<br><br><code>/artifacts/ <i>current-action-name</i> /artifact-name</code> <i>/(ce chemin est un lien symbolique vers \$CATALYST_SOURCE_DIR_artifact-name/)</i> | or<br><br><code>/artifacts/MyCurrentAction/MyArtifact/folder1/file.txt</code> |

## Téléchargement d'artefacts

Vous pouvez télécharger et inspecter les artefacts générés par les actions de votre flux de travail à des fins de résolution des problèmes. Il existe deux types d'artefacts que vous pouvez télécharger :

- Artefacts de source : artefact contenant un instantané du contenu du référentiel source tel qu'il existait au début de l'exécution.
- Artefacts de flux de travail : artefact défini dans la `Outputs` propriété du fichier de configuration de votre flux de travail.

Pour télécharger les artefacts produits par le flux de travail

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Sous le nom du flux de travail, choisissez Runs.
6. Dans Historique des exécutions, dans la colonne Run ID, sélectionnez une exécution. Par exemple, Run-95a4d.
7. Sous le nom de la course, sélectionnez Artifacts.

8. À côté d'un artefact, choisissez Télécharger. Un fichier d'archive est téléchargé. Son nom de fichier est composé de sept caractères aléatoires.
9. Extrayez l'archive à l'aide de l'utilitaire d'extraction d'archives de votre choix.

## Exemples

Les exemples suivants montrent comment générer et référencer des artefacts dans le fichier de définition du flux de travail.

### Rubriques

- [Exemple : sortie d'un artefact](#)
- [Exemple : saisie d'un artefact généré par une autre action](#)
- [Exemple : Référencement de fichiers dans plusieurs artefacts](#)
- [Exemple : Référencement d'un fichier dans un seul artefact](#)
- [Exemple : Référencer un fichier dans un artefact en présence d'un WorkflowSource](#)

### Exemple : sortie d'un artefact

L'exemple suivant montre comment générer un artefact qui inclut deux fichiers .jar.

```
Actions:
  Build:
    Identifieur: aws/build@v1
    Outputs:
      Artifacts:
        - Name: ARTIFACT1
          Files:
            - build-output/file1.jar
            - build-output/file2.jar
```

### Exemple : saisie d'un artefact généré par une autre action

L'exemple suivant vous montre comment générer un artefact appelé ARTIFACT4 in et BuildActionA le saisir dans BuildActionB.

```
Actions:
  BuildActionA:
    Identifieur: aws/build@v1
    Outputs:
```

```
Artifacts:
  - Name: ARTIFACT4
    Files:
      - build-output/file1.jar
      - build-output/file2.jar
BuildActionB:
  Identifieur: aws/build@v1
  Inputs:
    Artifacts:
      - ARTIFACT4
  Configuration:
```

### Exemple : Référencement de fichiers dans plusieurs artefacts

L'exemple suivant montre comment générer deux artefacts nommés ART5 et ART6 dans BuildActionC, puis référencer deux fichiers nommés file5.txt (dans artefactART5) et file6.txt (dans artefactART6) dans BuildActionD (sousSteps).

#### Note

Pour plus d'informations sur le référencement de fichiers, consultez [Référencement de fichiers dans un artefact](#).

#### Note

Bien que l'exemple montre le \$CATALYST\_SOURCE\_DIR\_ART5 préfixe utilisé, vous pouvez l'omettre. C'est parce que ART5 c'est l'entrée principale. Pour en savoir plus sur l'entrée principale, voir [Référencement de fichiers dans un artefact](#).

```
Actions:
  BuildActionC:
    Identifieur: aws/build@v1
    Outputs:
      Artifacts:
        - Name: ART5
          Files:
            - build-output/file5.txt
        - Name: ART6
          Files:
```

```
        - build-output/file6.txt
BuildActionD:
  Identifieur: aws/build@v1
  Inputs:
    Artifacts:
      - ART5
      - ART6
  Configuration:
    Steps:
      - run: cd $CATALYST_SOURCE_DIR_ART5/build-output && cat file5.txt
      - run: cd $CATALYST_SOURCE_DIR_ART6/build-output && cat file6.txt
```

### Exemple : Référencement d'un fichier dans un seul artefact

L'exemple suivant montre comment générer un artefact nommé ART7 dans BuildActionE, puis référencer file7.txt (dans artefactART7) dans BuildActionF (sousSteps).

Notez que la référence ne nécessite pas le préfixe \$CATALYST\_SOURCE\_DIR\_ *artefact-name* devant le build-output répertoire comme c'était le cas dans. [Exemple : Référencement de fichiers dans plusieurs artefacts](#) Cela est dû au fait qu'un seul élément est spécifié sousInputs.

#### Note

Pour plus d'informations sur le référencement de fichiers, consultez [Référencement de fichiers dans un artefact](#).

```
Actions:
BuildActionE:
  Identifieur: aws/build@v1
  Outputs:
    Artifacts:
      - Name: ART7
      Files:
        - build-output/file7.txt
BuildActionF:
  Identifieur: aws/build@v1
  Inputs:
    Artifacts:
      - ART7
  Configuration:
    Steps:
```

```
- run: cd build-output && cat file7.txt
```

Exemple : Référencer un fichier dans un artefact en présence d'un WorkflowSource

L'exemple suivant montre comment générer un artefact nommé ART8 dans BuildActionG, puis référencer file8.txt (dans artefactART8) dans BuildActionH (sousSteps).

Remarquez que la référence nécessite le préfixe `$CATALYST_SOURCE_DIR_`*artifact-name*, comme c'était le cas dans. [Exemple : Référencement de fichiers dans plusieurs artefacts](#) Cela est dû au fait que plusieurs éléments sont spécifiés sous Inputs (une source et un artefact). Vous avez donc besoin du préfixe pour indiquer où rechercher le fichier.

### Note

Pour plus d'informations sur le référencement de fichiers, consultez [Référencement de fichiers dans un artefact](#).

#### Actions:

##### BuildActionG:

Identifiant: aws/build@v1

##### Outputs:

##### Artifacts:

- Name: ART8

##### Files:

- build-output/file8.txt

##### BuildActionH:

Identifiant: aws/build@v1

##### Inputs:

##### Sources:

- WorkflowSource

##### Artifacts:

- ART8

##### Configuration:

##### Steps:

```
- run: cd $CATALYST_SOURCE_DIR_ART8/build-output && cat file8.txt
```

## Utilisation des images Docker de l'environnement de calcul et d'exécution

Dans un CodeCatalyst flux de travail, vous pouvez spécifier l'image de l'environnement de calcul et d'exécution à CodeCatalyst utiliser pour exécuter les actions du flux de travail.



Le calcul fait référence au moteur informatique (le processeur, la mémoire et le système d'exploitation) géré et maintenu par lequel les actions du flux CodeCatalyst de travail sont exécutées.

Une image d'environnement d'exécution est un conteneur Docker dans lequel CodeCatalyst s'exécute des actions de flux de travail. Le conteneur Docker s'exécute sur la plate-forme informatique que vous avez choisie et inclut un système d'exploitation et des outils supplémentaires dont une action de flux de travail peut avoir besoin AWS CLI, tels que Node.js et .tar.

## Rubriques

- [Travailler avec le calcul](#)
- [Partage du calcul entre les actions](#)
- [Utilisation des images Docker de l'environnement d'exécution](#)

## Travailler avec le calcul

Le calcul fait référence au moteur informatique (le processeur, la mémoire et le système d'exploitation) géré et maintenu par lequel les actions du flux CodeCatalyst de travail sont exécutées.

### Note

Si le calcul est défini comme une propriété du flux de travail, il ne peut être défini comme une propriété d'aucune action dans ce flux de travail. De même, si le calcul est défini comme une propriété d'une action, il ne peut pas être défini dans le flux de travail.

## Rubriques

- [À propos des types de calcul](#)
- [À propos des flottes informatiques](#)
- [Propriétés de flotte à la demande](#)
- [Propriétés du parc provisionné](#)
- [Création, modification et suppression d'une flotte provisionnée](#)
- [Affectation d'une flotte provisionnée ou d'un calcul à la demande à une action](#)

## À propos des types de calcul

CodeCatalyst propose les types de calcul suivants :

- Amazon EC2
- AWS Lambda

Amazon EC2 offre une flexibilité optimisée lors des exécutions d'actions et Lambda offre des vitesses de démarrage des actions optimisées. Lambda permet des exécutions d'actions de flux de travail plus rapides grâce à une latence de démarrage plus faible. Lambda vous permet d'exécuter des flux de travail de base permettant de créer, de tester et de déployer des applications sans serveur avec des environnements d'exécution courants. Ces environnements d'exécution incluent Node.js, Python, Java, .NET et Go. Cependant, Lambda ne prend pas en charge certains cas d'utilisation, et s'ils vous concernent, utilisez le type de calcul Amazon EC2 :

- Lambda ne prend pas en charge les images d'environnement d'exécution provenant d'un registre spécifié.
- Lambda ne prend pas en charge les outils qui nécessitent des autorisations root. Pour des outils tels que yum ou rpm, utilisez le type de calcul Amazon EC2 ou d'autres outils ne nécessitant pas d'autorisations root.
- Lambda ne prend pas en charge les builds ou les exécutions de Docker. Les actions suivantes qui utilisent des images Docker ne sont pas prises en charge : Deploy AWS CloudFormation stack, Deploy to Amazon ECS, Amazon S3 publish, AWS CDK bootstrap, AWS CDK deploy, AWS Lambda invoke et GitHub Actions. Les GitHub actions basées sur Docker qui s'exécutent dans le cadre d'une action CodeCatalyst GitHub Actions ne sont pas non plus prises en charge par le calcul Lambda. Vous pouvez utiliser des alternatives qui ne nécessitent pas d'autorisations root, telles que Podman.
- Lambda ne prend pas en charge l'écriture dans des fichiers externes. /tmp Lorsque vous configurez les actions de votre flux de travail, vous pouvez reconfigurer les outils à installer ou à utiliser pour /tmp écrire. Si vous avez une action de compilation qui s'installe avec npm, assurez-vous de la configurer pour l'installer sur /tmp.
- Lambda ne prend pas en charge les durées d'exécution supérieures à 15 minutes.

## À propos des flottes informatiques

CodeCatalyst propose les flottes de calcul suivantes :

- Flottes à la demande
- Flottes approvisionnées

Dans le cas des flottes à la demande, lorsqu'une action de flux de travail démarre, le flux de travail fournit les ressources dont il a besoin. Les machines sont détruites à la fin de l'action. Vous ne payez que pour le nombre de minutes pendant lesquelles vous exécutez vos actions. Les flottes à la demande sont entièrement gérées et incluent des fonctionnalités de mise à l'échelle automatique pour faire face aux pics de demande.

CodeCatalyst propose également des flottes provisionnées contenant des machines alimentées par Amazon EC2 et entretenues par CodeCatalyst. Avec les flottes provisionnées, vous configurez un ensemble de machines dédiées pour exécuter les actions de votre flux de travail. Ces machines restent inactives, prêtes à exécuter des actions immédiatement. Avec les flottes provisionnées, vos machines fonctionnent toujours et entraîneront des coûts tant qu'elles seront provisionnées.

Pour créer, mettre à jour ou supprimer une flotte, vous devez avoir le rôle d'administrateur de l'espace ou le rôle d'administrateur du projet.

### Propriétés de flotte à la demande

CodeCatalyst fournit les flottes à la demande suivantes :

| Nom                 | Système d'exploitation | Architecture | vCPU | Mémoire (Go) | Espace disque | Types de calcul pris en charge |
|---------------------|------------------------|--------------|------|--------------|---------------|--------------------------------|
| Linux.Arm64.Large   | Amazon Linux 2         | Bras 64      | 2    | 4            | 64 Go         | Amazon EC2                     |
|                     |                        |              |      |              | 10 Go         | Lambda                         |
| Linux.Arm64.XLarge  | Amazon Linux 2         | Bras 64      | 4    | 8            | 128 Go        | Amazon EC2                     |
|                     |                        |              |      |              | 10 Go         | Lambda                         |
| Linux.Arm64.2XLarge | Amazon Linux 2         | Bras 64      | 8    | 16           | 128 Go        | Amazon EC2                     |
| Linux.x86-64.Large  | Amazon Linux 2         | x86-64       | 2    | 4            | 64 Go         | Amazon EC2                     |
|                     |                        |              |      |              | 10 Go         | Lambda                         |

| Nom                  | Système d'exploitation | Architecture | vCPU | Mémoire (Go) | Espace disque | Types de calcul pris en charge |
|----------------------|------------------------|--------------|------|--------------|---------------|--------------------------------|
| Linux.x86-64.XLarge  | Amazon Linux 2         | x86-64       | 4    | 8            | 128 Go        | Amazon EC2                     |
|                      |                        |              |      |              | 10 Go         | Lambda                         |
| Linux.x86-64.2XLarge | Amazon Linux 2         | x86-64       | 8    | 16           | 128 Go        | Amazon EC2                     |

Si aucune flotte n'est sélectionnée, CodeCatalyst utilise Linux.x86-64.Large.

### Propriétés du parc provisionné

Une flotte provisionnée contient les propriétés suivantes :

#### Système d'exploitation

Le système d'exploitation. Les systèmes d'exploitation suivants sont disponibles :

- Amazon Linux 2
- Windows Server 2022

#### Note

Les flottes Windows ne sont prises en charge que dans l'action de génération. Les autres actions ne sont actuellement pas compatibles avec Windows.

#### Architecture

Architecture du processeur. Les architectures disponibles sont les suivantes :

- x86\_64
- Bras 64

#### Type de machine

Type de machine pour chaque instance. Les types de machines suivants sont disponibles :

| vCPU | Mémoire (Go) | Espace disque | Système d'exploitation                |
|------|--------------|---------------|---------------------------------------|
| 2    | 4            | 64 Go         | Amazon Linux 2                        |
| 4    | 8            | 128 Go        | Amazon Linux 2<br>Windows Server 2022 |
| 8    | 16           | 128 Go        | Amazon Linux 2<br>Windows Server 2022 |

## Capacité

Le nombre initial de machines allouées au parc, qui définit le nombre d'actions pouvant être exécutées en parallèle.

## Mode mise à l'échelle

Définit le comportement lorsque le nombre d'actions dépasse la capacité de la flotte.

Fournir des capacités supplémentaires à la demande


Des machines supplémentaires sont configurées à la demande. Elles augmentent automatiquement en réponse aux nouvelles actions en cours d'exécution, puis réduisent leur capacité de base à la fin des actions. Cela peut entraîner des coûts supplémentaires, car vous payez à la minute pour chaque machine en marche.

Attendez que la capacité de flotte supplémentaire soit disponible

Les exécutions d'actions sont placées dans une file d'attente jusqu'à ce qu'une machine soit disponible. Cela limite les coûts supplémentaires car aucune machine supplémentaire n'est allouée.

## Création, modification et suppression d'une flotte provisionnée

Suivez les instructions ci-dessous pour créer, modifier et supprimer des flottes provisionnées.

 Note

Les flottes approvisionnées seront désactivées après 2 semaines d'inactivité. S'ils sont réutilisés, ils seront automatiquement réactivés, mais cette réactivation peut entraîner une latence.

### Pour créer une flotte approvisionnée

1. Dans le volet de navigation, choisissez CI/CD, puis Compute.
2. Choisissez Créer une flotte provisionnée.
3. Dans le champ de texte Nom du parc approvisionné, entrez le nom de votre flotte.
4. Dans le menu déroulant Système d'exploitation, sélectionnez le système d'exploitation.
5. Dans le menu déroulant Type de machine, choisissez le type de machine pour votre machine.
6. Dans le champ de texte Capacité, entrez le nombre maximum de machines du parc.
7. Dans le menu déroulant du mode de mise à l'échelle, choisissez le comportement de débordement souhaité. Pour plus d'informations sur ces champs, consultez [Propriétés du parc provisionné](#).
8. Choisissez Créer.

Après avoir créé le parc approvisionné, vous êtes prêt à l'affecter à une action. Pour plus d'informations, consultez [Affectation d'une flotte provisionnée ou d'un calcul à la demande à une action](#).

### Pour modifier une flotte approvisionnée

1. Dans le volet de navigation, choisissez CI/CD, puis Compute.
2. Dans la liste des flottes approvisionnées, choisissez la flotte que vous souhaitez modifier.
3. Choisissez Modifier.
4. Dans le champ de texte Capacité, entrez le nombre maximum de machines du parc.
5. Dans le menu déroulant du mode de mise à l'échelle, choisissez le comportement de débordement souhaité. Pour plus d'informations sur ces champs, consultez [Propriétés du parc provisionné](#).
6. Choisissez Enregistrer.

## Pour supprimer une flotte approvisionnée

### Warning

Avant de supprimer un parc provisionné, supprimez-le de toutes les actions en supprimant la `Fleet` propriété du code YAML de l'action. Toute action qui continue de faire référence à une flotte approvisionnée après sa suppression échouera lors de la prochaine exécution de l'action.

1. Dans le volet de navigation, choisissez CI/CD, puis Compute.
2. Dans la liste des flottes approvisionnées, choisissez la flotte que vous souhaitez supprimer.
3. Sélectionnez Delete (Supprimer).
4. Entrez **delete** pour confirmer la suppression.
5. Sélectionnez Delete (Supprimer).

## Affectation d'une flotte provisionnée ou d'un calcul à la demande à une action

Par défaut, les actions de flux de travail utilisent le parc `Linux.x86-64.Large` à la demande avec un type de calcul Amazon EC2. Pour utiliser un parc provisionné à la place, ou pour utiliser un autre parc à la demande, par exemple `Linux.x86-64.2XLarge`, suivez les instructions suivantes.

### Visual

#### Avant de commencer

- Si vous souhaitez attribuer une flotte approvisionnée, vous devez d'abord créer la flotte approvisionnée. Pour plus d'informations, consultez [Création, modification et suppression d'une flotte provisionnée](#).

#### Pour affecter une flotte approvisionnée ou un type de flotte différent à une action

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.

5. Choisissez Modifier.
6. Choisissez Visual.
7. Dans le diagramme du flux de travail, choisissez l'action à laquelle vous souhaitez attribuer votre flotte approvisionnée ou votre nouveau type de flotte.
8. Cliquez sur l'onglet Configuration.
9. Dans Compute fleet, procédez comme suit :

Spécifiez la machine ou le parc qui exécutera votre flux de travail ou vos actions de flux de travail. Dans le cas des flottes à la demande, lorsqu'une action démarre, le flux de travail fournit les ressources dont il a besoin et les machines sont détruites à la fin de l'action. Exemples de flottes à la demande : `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Pour plus d'informations sur les flottes à la demande, consultez [Propriétés de flotte à la demande](#).

Avec les flottes provisionnées, vous configurez un ensemble de machines dédiées pour exécuter les actions de votre flux de travail. Ces machines restent inactives, prêtes à exécuter des actions immédiatement. Pour plus d'informations sur les flottes provisionnées, consultez [Propriétés du parc provisionné](#)

S'il `Fleet` est omis, la valeur par défaut est `Linux.x86-64.Large`.

10. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
11. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

### Avant de commencer

- Si vous souhaitez attribuer une flotte approvisionnée, vous devez d'abord créer la flotte approvisionnée. Pour plus d'informations, consultez [Création, modification et suppression d'une flotte provisionnée](#).

Pour affecter une flotte approvisionnée ou un type de flotte différent à une action

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.



4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail. /
5. Choisissez Modifier.
6. Choisissez YAML.
7. Recherchez l'action à laquelle vous souhaitez attribuer votre flotte approvisionnée ou votre nouveau type de flotte.
8. Dans l'action, ajoutez une Compute propriété et définissez Fleet le nom de votre flotte ou le type de flotte à la demande. Pour plus d'informations, consultez la description de la Fleet propriété dans le champ [Créez et testez une référence d'actions](#) correspondant à votre action.
9. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
10. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Partage du calcul entre les actions

### Rubriques

- [Exécution de plusieurs actions sur un ordinateur partagé](#)
- [Considérations relatives au partage du calcul](#)
- [Activer le partage de données](#)
- [Exemples](#)

Par défaut, les actions d'un flux de travail s'exécutent sur des instances distinctes d'une [flotte](#). Ce comportement permet aux actions d'isoler et de prévoir l'état des entrées. Le comportement par défaut nécessite une configuration explicite pour partager le contexte tel que les fichiers et les variables entre les actions.

Le partage de calcul est une fonctionnalité qui vous permet d'exécuter toutes les actions d'un flux de travail sur la même instance. L'utilisation du partage de calcul peut accélérer les temps d'exécution des flux de travail, car moins de temps est consacré au provisionnement des instances. Vous pouvez également partager des fichiers (artefacts) entre des actions sans configuration supplémentaire du flux de travail.

Lorsqu'un flux de travail est exécuté à l'aide du partage de calcul, une instance de la flotte par défaut ou spécifiée est réservée pour la durée de toutes les actions de ce flux de travail. Lorsque l'exécution du flux de travail est terminée, la réservation d'instance est publiée.

## Exécution de plusieurs actions sur un ordinateur partagé

Vous pouvez utiliser l'attribut `Compute` dans la définition YAML au niveau du flux de travail pour spécifier à la fois la flotte et les propriétés de partage de calcul des actions. Vous pouvez également configurer les propriétés de calcul à l'aide de l'éditeur visuel dans CodeCatalyst. Pour spécifier un parc, définissez le nom d'un parc existant, définissez le type de calcul sur EC2 et activez le partage de calcul.

### Note

Le partage de calcul n'est pris en charge que si le type de calcul est défini sur EC2, et il n'est pas pris en charge par le système d'exploitation Windows Server 2022. Pour plus d'informations sur les flottes de calcul, les types de calcul et les propriétés, consultez [Travailler avec le calcul](#).

### Note

Si vous êtes au niveau Free et que vous spécifiez la `Linux.x86-64.2XLarge` flotte `Linux.x86-64.XLarge` ou manuellement dans la définition du flux de travail YAML, l'action sera toujours exécutée sur la flotte par défaut (`Linux.x86-64.Large`). Pour plus d'informations sur la disponibilité et la tarification du calcul, consultez le [tableau des options de niveaux](#).

Lorsque le partage de calcul est activé, le dossier contenant la source du flux de travail est automatiquement copié d'une action à l'autre. Il n'est pas nécessaire de configurer les artefacts de sortie et de les référencer en tant qu'artefacts d'entrée dans une définition de flux de travail (fichier YAML). En tant qu'auteur de flux de travail, vous devez câbler les variables d'environnement à l'aide d'entrées et de sorties, comme vous le feriez sans utiliser le partage de calcul. Si vous souhaitez partager des dossiers entre des actions extérieures à la source du flux de travail, envisagez la mise en cache de fichiers. Pour plus d'informations, veuillez consulter [Travailler avec des artefacts](#) et [Utilisation de la mise en cache de fichiers](#).

Le référentiel source dans lequel réside le fichier de définition de votre flux de travail est identifié par l'étiquette `WorkflowSource`. Lors de l'utilisation du partage de calcul, la source du flux de travail est téléchargée lors de la première action qui la référence et est automatiquement mise à disposition pour les actions suivantes du flux de travail à utiliser. Toutes les modifications apportées

au dossier contenant la source du flux de travail par une action, telle que l'ajout, la modification ou la suppression de fichiers, sont également visibles dans les actions suivantes du flux de travail. Vous pouvez référencer les fichiers qui se trouvent dans le dossier source du flux de travail dans n'importe laquelle de vos actions de flux de travail, comme vous le pouvez sans utiliser le partage de calcul. Pour en savoir plus, consultez [Référencement de fichiers dans un référentiel source](#).

### Note

Les flux de travail de partage de calcul doivent spécifier une séquence d'actions stricte, de sorte que les actions parallèles ne peuvent pas être définies. Bien que les artefacts de sortie puissent être configurés à n'importe quelle action de la séquence, les artefacts d'entrée ne sont pas pris en charge.

## Considérations relatives au partage du calcul

Vous pouvez exécuter des flux de travail avec le partage de calcul afin d'accélérer les exécutions de flux de travail et de partager le contexte entre les actions d'un flux de travail utilisant la même instance. Tenez compte des points suivants pour déterminer si l'utilisation du partage de calcul convient à votre scénario :

|                             | Partage du calcul  | Sans partage de calcul  |
|-----------------------------|--|---|
| Type de calcul              | Amazon EC2   | Amazon EC2, AWS Lambda  |
| Provisionnement d'instances | Actions exécutées sur la même instance   | Actions exécutées sur des instances distinctes  |
| Système d'exploitation      | Amazon Linux 2   | Amazon Linux 2, Windows Server 2022 (action de génération uniquement)   |
| Référencement de fichiers   | <code>\$CATALYST_SOURCE_DIR_WorkflowSource</code> ,<br><code>/sources/WorkflowSource/</code> | <code>\$CATALYST_SOURCE_DIR</code> ,<br><code>IR_WorkflowSource</code> ,<br><code>/sources/WorkflowSource/</code> |

|  | Partage du calcul   | Sans partage de calcul  |
|--|---|---|
| Structure de flux de travail                         | Les actions ne peuvent être exécutées que de manière séquentielle         | Les actions peuvent être exécutées en parallèle                                       |
| Accès aux données via les actions du flux de travail | Accéder à la source de flux de travail mise en cache ()<br>WorkflowSource | Accès aux sorties des artefacts partagés (nécessite une configuration supplémentaire) |

## Activer le partage de données

Suivez les instructions ci-dessous pour activer le partage de calcul pour un flux de travail.

### Visual

Pour activer le partage de données à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez Visual.
7. Choisissez les propriétés du flux de travail.
8. Dans le menu déroulant Type de calcul, choisissez EC2.
9. (Facultatif) Dans le menu déroulant Compute fleet - facultatif, choisissez le parc que vous souhaitez utiliser pour exécuter des actions de flux de travail. Vous pouvez choisir une flotte à la demande ou créer et choisir une flotte provisionnée. Pour plus d'informations, consultez [Création, modification et suppression d'une flotte provisionnée](#) et [Affectation d'une flotte provisionnée ou d'un calcul à la demande à une action](#).
10. Activez ce bouton pour activer le partage de calcul et faire en sorte que les actions du flux de travail soient exécutées sur le même parc.
11. (Facultatif) Choisissez le mode d'exécution du flux de travail. Pour en savoir plus, consultez [Configuration des exécutions en file d'attente, remplacées et parallèles](#).

12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour activer le partage de calcul à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. Activez le partage de calcul en définissant le SharedInstance champ sur TRUE et Type sur EC2. Définissez Fleet un parc informatique que vous souhaitez utiliser pour exécuter des actions de flux de travail. Vous pouvez choisir une flotte à la demande ou créer et choisir une flotte provisionnée. Pour plus d'informations, consultez [Création, modification et suppression d'une flotte provisionnée](#) et [Affectation d'une flotte provisionnée ou d'un calcul à la demande à une action](#).

Dans un flux de travail YAML, ajoutez un code similaire au suivant :

```
Name: MyWorkflow
SchemaVersion: "1.0"
Compute: # Define compute configuration.
  Type: EC2
  Fleet: MyFleet # Optionally, choose an on-demand or provisioned fleet.
  SharedInstance: true # Turn on compute sharing. Default is False.
Actions:
  BuildFirst:
    Identifier: aws/build@v1
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - Run: ...
        ...
```

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Exemples

### Rubriques

- [Exemple : Amazon S3 Publish](#)

### Exemple : Amazon S3 Publish

Les exemples de flux de travail suivants montrent comment exécuter l'action Amazon Amazon S3 Publish de deux manières : d'abord en utilisant des artefacts d'entrée, puis en utilisant le partage de calcul. Avec le partage de calcul, les artefacts d'entrée ne sont pas nécessaires puisque vous pouvez accéder au `cacheWorkflowSource`. De plus, l'artefact de sortie de l'action Build n'est plus nécessaire. L'action S3 Publish est configurée pour utiliser la `DependsOn` propriété explicite afin de maintenir des actions séquentielles ; l'action Build doit être exécutée avec succès pour que l'action S3 Publish puisse s'exécuter.

- Sans partage de calcul, vous devez utiliser des artefacts d'entrée et partager les sorties avec les actions suivantes :

```
Name: S3PublishUsingInputArtifact
SchemaVersion: "1.0"
Actions:
  Build:
    Identifieur: aws/build@v1
    Outputs:
      Artifacts:
        - Name: ArtifactToPublish
          Files: [output.zip]
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - Run: ./build.sh # Build script that generates output.zip
  PublishToS3:
    Identifieur: aws/s3-publish@v1
    Inputs:
```

```

Artifacts:
- ArtifactToPublish
Environment:
Connections:
- Role: codecatalyst-deployment-role
  Name: dev-deployment-role
Name: dev-connection
Configuration:
SourcePath: output.zip
DestinationBucketName: dev-bucket

```

- Lorsque vous utilisez le partage de calcul en SharedInstance réglant sur TRUE, vous pouvez exécuter plusieurs actions sur la même instance et partager des artefacts en spécifiant une seule source de flux de travail. Les artefacts d'entrée ne sont pas obligatoires et ne peuvent pas être spécifiés :

```

Name: S3PublishUsingComputeSharing
SchemaVersion: "1.0"
Compute:
  Type: EC2
  Fleet: dev-fleet
  SharedInstance: TRUE
Actions:
  Build:
    Identifiant: aws/build@v1
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - Run: ./build.sh # Build script that generates output.zip
  PublishToS3:
    Identifiant: aws/s3-publish@v1
    DependsOn:
      - Build
    Environment:
      Connections:
        - Role: codecatalyst-deployment-role
          Name: dev-deployment-role
      Name: dev-connection
    Configuration:

```

```
SourcePath: output.zip
DestinationBucketName: dev-bucket
```

## Utilisation des images Docker de l'environnement d'exécution

Une image d'environnement d'exécution est un conteneur Docker dans lequel CodeCatalyst s'exécutent des actions de flux de travail. Le conteneur Docker s'exécute sur la plate-forme informatique que vous avez choisie et inclut un système d'exploitation et des outils supplémentaires dont une action de flux de travail peut avoir besoin AWS CLI, tels que Node.js et .tar.

Par défaut, les actions du flux de travail s'exécutent sur l'une [des images actives](#) fournies et gérées par CodeCatalyst. Seules les actions de création et de test prennent en charge les images personnalisées. Pour plus d'informations, consultez [Affectation d'une image Docker d'environnement d'exécution personnalisée à une action](#).

### Rubriques

- [Images actives](#)
- [Et si une image active n'inclut pas les outils dont j'ai besoin ?](#)
- [Affectation d'une image Docker d'environnement d'exécution personnalisée à une action](#)
- [Exemples](#)

### Images actives

Les images actives sont des images d'environnement d'exécution qui sont entièrement prises en charge par CodeCatalyst des outils préinstallés et qui incluent ceux-ci. Il existe actuellement deux séries d'images actives : l'une publiée en mars 2024 et l'autre publiée en novembre 2022.

Le fait qu'une action utilise une image de mars 2024 ou de novembre 2022 dépend de l'action :

- Les actions de création et de test ajoutées à un flux de travail le 26 mars 2024 ou après cette date incluront une `Container` section dans leur définition YAML qui spécifie explicitement une [image de mars 2024](#). Vous pouvez éventuellement supprimer la `Container` section pour revenir à [l'image de novembre 2022](#).
- Les actions de création et de test ajoutées à un flux de travail avant le 26 mars 2024 n'incluront aucune `Container` section dans leur définition YAML et utiliseront par conséquent une [image](#)



[de novembre 2022](#). Vous pouvez conserver l'image de novembre 2022 ou la mettre à jour. Pour mettre à niveau l'image, ouvrez l'action dans l'éditeur visuel, choisissez l'onglet Configuration, puis sélectionnez l'image de mars 2024 dans la liste déroulante des images docker de l'environnement d'exécution. Cette sélection ajoutera une `Container` section à la définition YAML de l'action qui sera remplie avec l'image de mars 2024 appropriée.

- Toutes les autres actions utiliseront une [image de novembre 2022](#), quelle que soit la date à laquelle elles ont été ajoutées au flux de travail. La mise à niveau de ces actions pour utiliser une image de mars 2024 n'est actuellement pas possible.

## Rubriques

- [Images du mois de mars 2024](#)
- [Images de novembre 2022](#)

### Images du mois de mars 2024

Les images de mars 2024 sont les dernières images fournies par CodeCatalyst. Il existe une image de mars 2024 par combinaison type de calcul/flotte.

Le tableau suivant présente les outils installés sur chaque image de mars 2024.

### Outils d'image de mars 2024

| Outil               | CodeCatalyst Amazon EC2 pour Linux x86_64 - <b>CodeCatalystLinux_x86_64:2024_03</b> | CodeCatalyst Lambda pour Linux x86_64 - <b>CodeCatalystLinuxLambda_x86_64:2024_03</b> | CodeCatalyst Amazon EC2 pour Linux Arm64 - <b>CodeCatalystLinux_Arm64:2024_03</b> | CodeCatalyst Lambda pour Linux Arm64 - <b>CodeCatalystLinuxLambda_Arm64:2024_03</b> |
|---------------------|---|---|---|---|
| AWS CLI             | 2,1,17  | 2,1,17  | 2,1,17  | 2,1,17  |
| AWS CLI du copilote | 1,32.1  | 1,32.1  | 1,32.1  | 1,32.1  |
| Docker              | 24,0,9  | N/A   | 24,0,9  | N/A   |
| Docker Compose      | 2.23.3  | N/A   | 2.23.3  | N/A   |

| Outil   | CodeCatalyst Amazon EC2 pour Linux x86_64 - CodeCatalystLinux_x86_64:2024_03 | CodeCatalyst Lambda pour Linux x86_64 - CodeCatalystLinuxLambda_x86_64:2024_03 | CodeCatalyst Amazon EC2 pour Linux Arm64 - CodeCatalystLinux_Arm64:2024_03 | CodeCatalyst Lambda pour Linux Arm64 - CodeCatalystLinuxLambda_Arm64:2024_03 |
|---------|--|--|--|--|
| Git     | 2.43.0   | 2.43.0   | 2.43.0   | 2.43.0   |
| Go      | 1.21.5   | 1.21.5   | 1.21.5   | 1.21.5   |
| Gradle  | 8,5  | 8,5  | 8,5  | 8,5  |
| Java    | Corretto 17  | Corretto 17  | Corretto 17  | Corretto 17  |
| Maven   | 3,9,6  | 3,9,6  | 3,9,6  | 3,9,6  |
| Node.js | 18,19,0  | 18,19,0  | 18,19,0  | 18,19,0  |
| npm     | 10.2.3   | 10.2.3   | 10.2.3   | 10.2.3   |
| Python  | 3,9,18   | 3,9,18   | 3,9,18   | 3,9,18   |
| Python3 | 3.11.6   | 3.11.6   | 3.11.6   | 3.11.6   |
| pip     | 22.3.1   | 22.3.1   | 22.3.1   | 22.3.1   |
| .NET    | 8,0,100  | 8,0,100  | 8,0,100  | 8,0,100  |

## Images de novembre 2022

Il existe une image de novembre 2022 par combinaison type de calcul/flotte. Une image Windows de novembre 2022 est également disponible avec l'action de génération si vous avez configuré un [parc provisionné](#).

Le tableau suivant montre les outils installés sur chaque image de novembre 2022.


## outils d'image de novembre 2022

| Outil               | CodeCatalyst Amazon EC2 pour Linux x86_64 - CodeCatalystLinux_x86_64:2022_11 | CodeCatalyst Lambda pour Linux x86_64 - CodeCatalystLinuxLambda_x86_64:2022_11 | CodeCatalyst Amazon EC2 pour Linux Arm64 - CodeCatalystLinux_Arm64:2022_11 | CodeCatalyst Lambda pour Linux Arm64 - CodeCatalystLinuxLambda_Arm64:2022_11 |
|---------------------|--|--|--|--|
| AWS CLI             | 2,1,17   | 2,1,17   | 2,1,17   | 2,1,17   |
| AWS CLI du copilote | 0.6.0  | 0.6.0  | N/A  | N/A  |
| Docker              | 23,01  | N/A  | 23,0.1   | N/A  |
| Docker Compose      | 2.16.0   | N/A  | 2.16.0   | N/A  |
| Git                 | 2.40.0   | 2.40.0   | 2,39,2   | 2,39,2   |
| Go                  | 1.20.2   | 1.20.2   | 1.20.1   | 1.20.1   |
| Gradle              | 8,0.2  | 8,0.2  | 8.0.1  | 8.0.1  |
| Java                | Corretto 17  | Corretto 17  | Corretto 17  | Corretto 17  |
| Maven               | 3.9.4  | 3.9.4  | 3.9.0  | 3.9.0  |
| Node.js             | 16,20,2  | 16,20,2  | 16,19.1  | 16,14.2  |
| npm                 | 8,19,4   | 8,19,4   | 8,19,3   | 8.5.0  |
| Python              | 3,9,15   | 2.7,18   | 3.11.2   | 2.7,18   |
| Python3             | N/A  | 3,9,15   | N/A  | 3.11.2   |
| pip                 | 22.2.2   | 22.2.2   | 23,0.1   | 23,0.1   |
| .NET                | 6,0,407  | 6,0,407  | 6,0,406  | 6,0,406  |

Et si une image active n'inclut pas les outils dont j'ai besoin ?

Si aucune des [images actives](#) fournies par n' CodeCatalyst inclut les outils dont vous avez besoin, plusieurs options s'offrent à vous :

- Vous pouvez fournir une image Docker d'environnement d'exécution personnalisée qui inclut les outils nécessaires. Pour plus d'informations, consultez [Affectation d'une image Docker d'environnement d'exécution personnalisée à une action](#).

 Note

Si vous souhaitez fournir une image Docker d'environnement d'exécution personnalisée, assurez-vous que Git est installé sur votre image personnalisée.

- Vous pouvez demander à l'action de création ou de test de votre flux de travail d'installer les outils dont vous avez besoin.

Par exemple, vous pouvez inclure les instructions suivantes dans la Steps section du code YAML de l'action de compilation ou de test :

```
Configuration:  
Steps:  
  - Run: ./setup-script
```

L'instruction *setup-script* exécuterait ensuite le script suivant pour installer le gestionnaire de packages Node (npm) :

```
#!/usr/bin/env bash  
echo "Setting up environment"  
  
touch ~/.bashrc  
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash  
source ~/.bashrc  
nvm install v16.1.0  
source ~/.bashrc
```

Pour plus d'informations sur l'action de génération YAML, consultez [Créez et testez une référence d'actions](#).

## Affectation d'une image Docker d'environnement d'exécution personnalisée à une action

Si vous ne souhaitez pas utiliser une [image active](#) fournie par CodeCatalyst, vous pouvez fournir une image Docker d'environnement d'exécution personnalisée. Si vous souhaitez fournir une image personnalisée, assurez-vous que Git y est installé. L'image peut résider dans Docker Hub, Amazon Elastic Container Registry ou dans n'importe quel référentiel public.

Pour savoir comment créer une image Docker personnalisée, consultez la section [Conteneuriser une application](#) dans la documentation Docker.

Suivez les instructions suivantes pour attribuer l'image Docker de votre environnement d'exécution personnalisé à une action. Après avoir spécifié une image, elle la CodeCatalyst déploie sur votre plateforme de calcul lorsque l'action démarre.

### Note

Les actions suivantes ne prennent pas en charge les images Docker de l'environnement d'exécution personnalisé : Deploy AWS CloudFormation stack, Deploy to ECS et GitHub Actions. Les images Docker de l'environnement d'exécution personnalisé ne prennent pas non plus en charge le type de calcul Lambda.

## Visual

Pour attribuer une image Docker à un environnement d'exécution personnalisé à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
3. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
4. Choisissez Modifier.
5. Choisissez Visual.
6. Dans le diagramme du flux de travail, choisissez l'action qui utilisera l'image Docker de votre environnement d'exécution personnalisé.
7. Cliquez sur l'onglet Configuration.
8. En bas de la page, remplissez les champs suivants.

## Image Docker de l'environnement d'exécution - facultatif

Spécifiez le registre dans lequel votre image est stockée. Les valeurs valides sont les suivantes :

- CODECATALYST(éditeur YAML)

L'image est enregistrée dans le CodeCatalyst registre.

- Docker Hub (éditeur visuel) ou DockerHub (éditeur YAML)

L'image est stockée dans le registre d'images de Docker Hub.

- Autre registre (éditeur visuel) ou Other (éditeur YAML)

L'image est stockée dans un registre d'images personnalisé. Tout registre accessible au public peut être utilisé.

- Amazon Elastic Container Registry (éditeur visuel) ou ECR (éditeur YAML)

L'image est stockée dans un référentiel d'images Amazon Elastic Container Registry. Pour utiliser une image dans un référentiel Amazon ECR, cette action nécessite l'accès à Amazon ECR. Pour activer cet accès, vous devez créer un [rôle IAM](#) qui inclut les autorisations suivantes et une politique de confiance personnalisée. (Vous pouvez modifier un rôle existant pour inclure les autorisations et la politique, si vous le souhaitez.)

Le rôle IAM doit inclure les autorisations suivantes dans sa politique de rôle :

- `ecr:BatchCheckLayerAvailability`
- `ecr:BatchGetImage`
- `ecr:GetAuthorizationToken`
- `ecr:GetDownloadUrlForLayer`

Le rôle IAM doit inclure la politique de confiance personnalisée suivante :

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

```
        "Service": [
            "codecatalyst-runner.amazonaws.com",
            "codecatalyst.amazonaws.com"
        ],
        "Action": "sts:AssumeRole"
    }
]
```

Pour plus d'informations sur la création de rôles IAM, consultez la section [Création d'un rôle à l'aide de politiques de confiance personnalisées \(console\)](#) dans le guide de l'utilisateur IAM.

Une fois le rôle créé, vous devez l'attribuer à l'action par le biais d'un environnement. Pour plus d'informations, consultez [Associer un environnement, une connexion à un compte et un rôle IAM à une action de flux de travail](#).

URL de l'image ECR, image Docker Hub ou URL de l'image

Spécifiez l'un des éléments suivants :

- Si vous utilisez un CODECATALYST registre, définissez l'image sur l'une des [images actives](#) suivantes :
  - CodeCatalystLinux\_x86\_64:2024\_03
  - CodeCatalystLinux\_x86\_64:2022\_11
  - CodeCatalystLinux\_Arm64:2024\_03
  - CodeCatalystLinux\_Arm64:2022\_11
  - CodeCatalystLinuxLambda\_x86\_64:2024\_03
  - CodeCatalystLinuxLambda\_x86\_64:2022\_11
  - CodeCatalystLinuxLambda\_Arm64:2024\_03
  - CodeCatalystLinuxLambda\_Arm64:2022\_11
  - CodeCatalystWindows\_x86\_64:2022\_11
- Si vous utilisez un registre Docker Hub, attribuez à l'image le nom de l'image Docker Hub et la balise facultative.

### Exemple : postgres:latest

- Si vous utilisez un registre Amazon ECR, définissez l'image sur l'URI du registre Amazon ECR.

Exemple : `111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo`

- Si vous utilisez un registre personnalisé, définissez l'image sur la valeur attendue par le registre personnalisé.
9. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
  10. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour attribuer une image Docker à un environnement d'exécution personnalisé à l'aide de l'éditeur YAML

1. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
2. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
3. Choisissez Modifier.
4. Choisissez YAML.
5. Recherchez l'action à laquelle vous souhaitez attribuer une image Docker de l'environnement d'exécution.
6. Dans l'action, ajoutez une `Container` section, un sous-jacent `Registry` et `Image` des propriétés. Pour plus d'informations, consultez la `Container description Registry` et les `Image` propriétés [Actions](#) de votre action.
7. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
8. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Exemples

Les exemples suivants montrent comment attribuer une image Docker d'environnement d'exécution personnalisée à une action dans le fichier de définition du flux de travail.

## Rubriques



- [Exemple : utilisation d'une image Docker d'environnement d'exécution personnalisée pour ajouter la prise en charge de Node.js 18 avec Amazon ECR](#)
- [Exemple : utilisation d'une image Docker d'environnement d'exécution personnalisée pour ajouter la prise en charge de Node.js 18 avec Docker Hub](#)

Exemple : utilisation d'une image Docker d'environnement d'exécution personnalisée pour ajouter la prise en charge de Node.js 18 avec Amazon ECR

L'exemple suivant montre comment utiliser une image Docker d'environnement d'exécution personnalisée pour ajouter la prise en charge de Node.js 18 avec [Amazon ECR](#).

```
Configuration:
Container:
  Registry: ECR
  Image: public.ecr.aws/amazonlinux/amazonlinux:2023
```

Exemple : utilisation d'une image Docker d'environnement d'exécution personnalisée pour ajouter la prise en charge de Node.js 18 avec Docker Hub

L'exemple suivant montre comment utiliser une image Docker d'environnement d'exécution personnalisée pour ajouter la prise en charge de Node.js 18 avec [Docker Hub](#).

```
Configuration:
Container:
  Registry: DockerHub
  Image: node:18.18.2
```

## Utilisation des environnements

Un environnement, à ne pas confondre avec un [environnement de développement](#), est l'endroit où le code est déployé. Il contient généralement une instance d'une application en cours d'exécution ainsi que son infrastructure associée. Vous pouvez donner à votre environnement un nom tel que development, test, staging ou production. Tous les déploiements générés par CodeCatalyst un environnement apparaîtront sur la page Environnements. Pour configurer un environnement, vous lui donnez un nom, tel quemy-production-environment, puis vous l'associez à votre Compte AWS.

Outre l'affichage des informations de déploiement, un environnement sert également de mécanisme permettant d'attribuer des rôles AWS IAM aux [actions](#) du flux de travail.

## Est-il possible de créer plusieurs environnements au sein d'un même flux de travail ?

Oui. Si un flux de travail inclut plusieurs actions, un environnement peut être attribué à chacune de ces actions. Par exemple, vous pouvez avoir un flux de travail qui inclut deux actions de déploiement, l'une étant affectée à un `my-staging-environment` environnement et l'autre à un `my-production-environment` environnement.

## Quelles actions soutiennent les environnements ?

Les actions suivantes permettent d'afficher leurs informations de déploiement sur la page Environnements :

- Déployer AWS CloudFormation une pile — Pour plus d'informations, voir [Ajout de l'action « Deploy AWS CloudFormation stack »](#)
- Déploiement sur Amazon ECS — Pour plus d'informations, consultez [Ajout de l'action « Déployer sur Amazon ECS »](#)
- Déploiement sur un cluster Kubernetes — Pour plus d'informations, voir [Ajout de l'action « Déployer vers le cluster Kubernetes »](#)
- AWS CDKdéployer — Pour plus d'informations, voir [Ajout de l'action « AWS CDK déployer »](#)

### Note

Si vous souhaitez autoriser une action à accéder à votre AWS compte et à y effectuer des opérations, vous devez l'associer à un environnement. De nombreuses actions soutiennent une association environnementale, y compris, mais sans s'y limiter, les actions répertoriées précédemment. Vous pouvez savoir quelles actions prennent en charge une association d'environnement, car elles incluront une liste déroulante Environnement dans leur onglet Configuration de l'[éditeur visuel](#).

## Régions prises en charge

La page Environnements peut afficher les ressources de n'importe quelle AWS région.

## Les environnements sont-ils obligatoires ?

Un environnement est obligatoire si l'action de flux de travail à laquelle il est affecté déploie des ressources dans le AWS cloud ou communique avec les AWS services pour d'autres raisons (telles que la surveillance et le reporting).

## Création d'un environnement

Suivez les instructions ci-dessous pour créer un environnement vide que vous pourrez associer ultérieurement à une action de flux de travail.

### Avant de commencer

Vous avez besoin des éléments suivants :

- Un CodeCatalyst espace. Pour de plus amples informations, veuillez consulter [Configuration CodeCatalyst](#).
- Un CodeCatalyst projet. Pour de plus amples informations, veuillez consulter [Création d'un projet à l'aide d'un plan](#).
- Une connexion à un AWS compte qui inclut les rôles IAM auxquels votre action de flux de travail devra accéder AWS. Vous pouvez utiliser au maximum une connexion à un compte par environnement. Pour de plus amples informations, veuillez consulter [Administration Comptes AWS d'un espace](#).

#### Note

Vous pouvez créer un environnement sans connexion à un compte, mais vous devrez revenir et ajouter la connexion ultérieurement.

Pour créer un environnement .

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Environments.
4. Dans Nom de l'environnement, entrez un nom, tel que **Production** ou **Staging**.
5. Dans Type d'environnement, sélectionnez l'une des options suivantes :

- Hors production : environnement dans lequel vous pouvez tester votre application pour vous assurer qu'elle fonctionne comme prévu avant de la mettre en production.
- Production : environnement « en direct » accessible au public qui héberge votre application finalisée.

Si vous choisissez Production, un badge de production apparaît dans l'interface utilisateur à côté de toutes les actions associées à l'environnement. Le badge vous permet de voir rapidement quelles actions sont déployées en production. Hormis l'apparence du badge, il n'existe aucune différence entre les environnements de production et les environnements hors production.

6. (Facultatif) Dans Connexion VPC, choisissez une connexion VPC que vous souhaitez associer à cet environnement. Pour plus d'informations sur la création de cette connexion VPC, consultez [Managing Amazon Virtual Private Clouds](#) dans le guide de l'CodeCatalyst administrateur.
7. (Facultatif) Dans Description, entrez une description telle que **Production environment for the hello-world app**.
8. Dans Connexion - facultatif, choisissez la connexion au AWS compte que vous souhaitez associer à cet environnement. Assurez-vous que la connexion au compte inclut les rôles IAM que vous souhaitez associer à l'environnement. Pour plus d'informations sur la création de cette connexion, consultez [Administration Comptes AWS d'un espace](#).
9. Choisissez Créer un environnement. CodeCatalyst crée un environnement vide.

## Étapes suivantes

- Maintenant que vous avez créé un environnement, vous êtes prêt à l'associer à une action de flux de travail. Pour de plus amples informations, veuillez consulter [Associer un environnement, une connexion à un compte et un rôle IAM à une action de flux de travail](#).

## Associer un environnement, une connexion à un compte et un rôle IAM à une action de flux de travail

Lorsque vous associez un environnement, une connexion à un compte et un rôle IAM à une [action de flux de travail prise en charge](#), le rôle IAM devient disponible pour être utilisé par l'action. Outre l'accès à un rôle IAM, l'action peut également voir ses informations de déploiement importées dans la page Environnements. Pour de plus amples informations, veuillez consulter [Quelles actions soutiennent les environnements ?](#).

Suivez les instructions suivantes pour associer un environnement, une connexion à un compte et un rôle IAM à une action.

Étape 1 : associer l'environnement, la connexion au compte et le rôle à une action de flux de travail

Utilisez la procédure suivante pour associer un environnement, une connexion à un compte et un rôle à une action de flux de travail.

## Visual

Pour associer un environnement, une connexion à un compte et un rôle à une action de flux de travail à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez Visual.
7. Dans le diagramme du flux de travail, choisissez une action prise en charge par les environnements. Pour de plus amples informations, veuillez consulter [Quelles actions soutiennent les environnements ?](#).
8. Choisissez l'onglet Configuration et spécifiez les informations dans les champs, comme suit.

## Environnement

Spécifiez l' CodeCatalyst environnement à utiliser avec l'action.

Pour plus d'informations sur les environnements, reportez-vous [Utilisation des environnements](#) aux sections et [Création d'un environnement](#).

Connexion au compte ou Connexion - optionnelle (selon les options disponibles)

Spécifiez la connexion au compte à associer à l'action. Vous pouvez spécifier un maximum d'une connexion à un compte sous `Environment`.

Pour plus d'informations sur les connexions aux comptes, consultez [Administration Comptes AWS d'un espace](#). Pour plus d'informations sur la façon d'associer une connexion à un compte à votre environnement, consultez [Création d'un environnement](#).

## Rôle

Spécifiez le nom du rôle IAM que cette action utilise pour accéder à des AWS services tels qu'Amazon S3 et Amazon ECR et y opérer. Assurez-vous que ce rôle est ajouté à la connexion de votre compte. Pour ajouter un rôle IAM à une connexion à un compte, consultez [Ajouter des rôles IAM aux connexions aux comptes](#).

### Note

Vous pouvez peut-être spécifier le nom du `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle ici, à condition qu'il dispose des autorisations suffisantes. Pour plus d'informations sur ce rôle, consultez [Création du CodeCatalystWorkflowDevelopmentRole-\*spaceName\* rôle pour votre compte et votre espace](#). Sachez que le `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle dispose d'autorisations très étendues, ce qui peut présenter un risque de sécurité. Nous vous recommandons de n'utiliser ce rôle que dans les didacticiels et les scénarios où la sécurité est moins préoccupante.

Si le rôle ne figure pas dans la liste, c'est parce que vous ne l'avez pas associé à la connexion au compte. Pour de plus amples informations, veuillez consulter [Ajouter des rôles IAM aux connexions aux comptes](#).

9. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
10. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour associer un environnement, une connexion de compte et un rôle à une action de flux de travail à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.

3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez YAML.
7. Dans l'action de flux de travail que vous souhaitez associer à un environnement, ajoutez un code similaire au suivant :

```
action-name
Environment:
  Name: environment-name
Connections:
  - Name: account-connection-name
    Role: iam-role-name
```

Pour plus d'informations, consultez le [Référence de définition du flux de travail](#) correspondant à votre action.

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Étape 2 : peupler l'environnement

Après avoir associé un environnement, une connexion à un compte et un rôle à une action de flux de travail, vous pouvez renseigner la page Environnements avec des informations de déploiement. Suivez les instructions ci-dessous pour remplir la page Environnements.

### Note

La page Environnements n'est prise en charge que par un sous-ensemble d'actions de flux de travail. Pour de plus amples informations, veuillez consulter [Quelles actions soutiennent les environnements ?](#).

## Pour peupler l'environnement

1. Si une exécution de flux de travail ne démarrait pas automatiquement lorsque vous avez validé vos modifications [Étape 1 : associer l'environnement, la connexion au compte et le rôle à une action de flux de travail](#), lancez-la manuellement comme suit :
  - a. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
  - b. Choisissez le nom du flux de travail dans lequel vous souhaitez démarrer une exécution. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
  - c. Cliquez sur Exécuter.

L'exécution du flux de travail lance un nouveau déploiement, ce qui entraîne CodeCatalyst l'ajout d'informations sur les ressources de votre application sous Environnements.

2. Vérifiez que les ressources de votre application apparaissent sous votre environnement :
  - a. Dans le volet de navigation, choisissez CI/CD, puis Environments.
  - b. Choisissez votre environnement (par exemple, `Production`).
  - c. Choisissez l'onglet Activité de déploiement et vérifiez qu'un déploiement apparaît avec le statut RÉUSSI. Cela indique qu'un flux de travail exécuté a correctement déployé les ressources de votre application.
  - d. Choisissez l'onglet Cibles de déploiement et vérifiez que les ressources de votre application apparaissent.

## Gestion d'un environnement

Utilisez les instructions suivantes pour gérer un environnement en l'associant à une connexion VPC ou à un. Compte AWS

### Associer une connexion VPC à un environnement

Lorsqu'une action est configurée avec un environnement doté d'une connexion VPC, l'action s'exécute connectée au VPC, conformément aux règles réseau et aux ressources d'accès spécifiées par le VPC associé. La même connexion VPC peut être utilisée par un ou plusieurs environnements.

Suivez les instructions suivantes pour associer une connexion VPC à un environnement.



## Pour associer une connexion VPC à un environnement

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Environments.
4. Choisissez votre environnement (par exemple, Production).
5. Choisissez l'onglet Propriétés de l'environnement.
6. Choisissez Gérer la connexion VPC, choisissez la connexion VPC de votre choix, puis cliquez sur Confirmer. Cela associe la connexion VPC que vous avez sélectionnée à cet environnement.

Pour plus d'informations, consultez [Managing Amazon Virtual Private Clouds](#) dans le guide de CodeCatalyst l'administrateur.

## Associer un Compte AWS à un environnement

Suivez les instructions ci-dessous pour un Compte AWS associer un à un environnement.

### Pour associer un Compte AWS à un environnement

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Environments.
4. Choisissez votre environnement (par exemple, Production).
5. Choisissez l'onglet Propriétés de l'environnement.
6. Choisissez Associer Compte AWS, choisissez ce que vous souhaitez Compte AWS, puis choisissez Associer. Cela associe votre sélection Compte AWS à cet environnement.

Pour de plus amples informations, veuillez consulter [Administration Comptes AWS d'un espace](#).

## Utilisation de la mise en cache de fichiers

Lorsque la mise en cache de fichiers est activée, les actions de génération et de test enregistrent les fichiers sur disque dans un cache et les restaurent à partir de ce cache lors des exécutions de flux de travail suivantes. La mise en cache réduit la latence causée par la création ou le téléchargement de dépendances qui n'ont pas changé entre les exécutions. CodeCatalyst prend également en charge les caches de secours, qui peuvent être utilisés pour restaurer des caches partiels contenant

certaines des dépendances nécessaires. Cela permet de réduire l'impact sur la latence d'un échec du cache.

### Note

La mise en cache des fichiers n'est disponible qu'avec les actions de CodeCatalyst [compilation](#) et de [test](#) Amazon, et uniquement lorsqu'elles sont configurées pour utiliser le type de [calcul](#) EC2.

## Rubriques

- [À propos de la mise en cache de fichiers](#)
- [Création d'un cache](#)
- [Constraints](#)

## À propos de la mise en cache de fichiers

La mise en cache des fichiers vous permet d'organiser vos données dans plusieurs caches, chacun référencé sous la FileCaching propriété. Chaque cache enregistre un répertoire spécifié par un chemin donné. Le répertoire spécifié sera restauré lors des futures exécutions du flux de travail. Voici un exemple d'extrait de code YAML pour la mise en cache avec plusieurs caches nommés et cacheKey1 cacheKey2

```
Actions:
  BuildMyNpmApp:
    Identifier: aws/build@v1
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - Run: npm install
        - Run: npm run test
    Caching:
      FileCaching:
        cacheKey1:
          Path: file1.txt
          RestoreKeys:
            - restoreKey1
```

```
cacheKey2:  
  Path: /root/repository  
  RestoreKeys:  
    - restoreKey2  
    - restoreKey3
```

### Note

CodeCatalyst utilise une mise en cache multicouche, composée d'un cache local et d'un cache distant. Lorsque des flottes provisionnées ou des machines à la demande rencontrent une erreur de cache dans un cache local, les dépendances sont restaurées à partir d'un cache distant. Par conséquent, le téléchargement d'un cache distant peut entraîner une latence lors de l'exécution de certaines actions.

CodeCatalyst applique des restrictions d'accès au cache pour garantir qu'une action dans un flux de travail ne peut pas modifier les caches d'un autre flux de travail. Cela protège chaque flux de travail contre d'autres qui pourraient transmettre des données incorrectes ayant un impact sur les builds ou les déploiements. Les restrictions sont appliquées grâce à des étendues de cache qui isolent les caches pour chaque flux de travail et chaque association de branches. Par exemple, `workflow-A` la branche `feature-A` possède un cache de fichiers différent de celui `workflow-A` de la branche `feature-B` sœur.

Des erreurs de cache se produisent lorsqu'un flux de travail recherche un cache de fichiers spécifié et ne parvient pas à le trouver. Cela peut se produire pour plusieurs raisons, par exemple lorsqu'une nouvelle branche est créée ou lorsqu'un nouveau cache est référencé alors qu'il n'a pas encore été créé. Cela peut également se produire lorsqu'un cache expire, ce qui se produit par défaut 14 jours après sa dernière utilisation. Pour atténuer les erreurs de cache et augmenter le taux d'accès au cache, CodeCatalyst prend en charge les caches de secours. Les caches de secours sont des caches alternatifs et permettent de restaurer des caches partiels, qui peuvent être une ancienne version d'un cache. Un cache est restauré en recherchant d'abord une correspondance sous le nom `FileCaching` de propriété et, s'il n'est pas trouvé, il est évalué. `RestoreKeys` En cas d'erreur de cache à la fois pour le nom de la propriété et pour l'ensemble des propriétés `RestoreKeys`, le flux de travail continuera de s'exécuter, car la mise en cache est une solution optimale et n'est pas garantie.

## Création d'un cache

Vous pouvez utiliser les instructions suivantes pour ajouter un cache à votre flux de travail.

## Visual

Pour ajouter un cache à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez Visual.
7. Dans le diagramme du flux de travail, choisissez l'action à laquelle vous souhaitez ajouter votre cache.
8. Choisissez Configuration.
9. Sous Mise en cache de fichiers - facultatif, choisissez Ajouter un cache et entrez les informations dans les champs, comme suit :

### Clé

Spécifiez le nom de votre propriété de cache principale. Les noms des propriétés du cache doivent être uniques dans votre flux de travail. Chaque action peut comporter jusqu'à cinq entrées `FileCaching`.

### Chemin

Spécifiez le chemin associé à votre cache.

Clés de restauration : facultatif

Spécifiez la clé de restauration à utiliser comme solution de secours lorsque la propriété de cache principale est introuvable. Les noms des clés de restauration doivent être uniques dans votre flux de travail. Chaque cache peut contenir jusqu'à cinq entrées `RestoreKeys`.

10. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
11. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour ajouter un cache à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez YAML.
7. Dans une action de flux de travail, ajoutez un code similaire au suivant :

```
action-name:
  Configuration:
    Steps: ...
  Caching:
    FileCaching:
      key-name:
        Path: file-path
        # # Specify any additional fallback caches
        # RestoreKeys:
        # - restore-key
```

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Constraints

Les contraintes relatives au nom de la propriété sont les RestoreKeys suivantes :

- Les noms doivent être uniques dans un flux de travail.
- Les noms sont limités aux caractères alphanumériques (A-Z, a-z, 0-9), aux tirets (-) et aux traits de soulignement (\_).
- Les noms peuvent comporter jusqu'à 180 caractères.
- Chaque action peut contenir jusqu'à cinq caches. FileCaching
- Chaque cache peut contenir jusqu'à cinq entréesRestoreKeys.

Les contraintes relatives aux chemins sont les suivantes :

- Les astérisques (\*) ne sont pas autorisés.
- Les chemins peuvent comporter jusqu'à 255 caractères.

## Travailler avec des packages

Un package est un ensemble qui inclut à la fois le logiciel et les métadonnées nécessaires à l'installation du logiciel et à la résolution des dépendances. CodeCatalyst prend en charge le format de package npm.

Un package comprend :

- Un nom (par exemple, webpack c'est le nom d'un package npm populaire)
- Un espace de [noms](#) facultatif (par exemple, @types dans@types/node)
- Un ensemble de [versions](#) (par exemple, 1.0.0, 1.0.1, 1.0.2)
- Métadonnées au niveau du package (par exemple, balises npm dist)

Dans CodeCatalyst, vous pouvez publier des packages et consommer des packages depuis des référentiels de CodeCatalyst packages dans vos flux de travail. Vous pouvez configurer une action de construction ou de test avec un référentiel de CodeCatalyst packages pour configurer automatiquement le client npm d'une action pour envoyer et extraire des packages depuis le référentiel spécifié.

Pour plus d'informations sur les packages, consultez [Packages en CodeCatalyst](#).

### Note

Actuellement, les actions de compilation et de test prennent en charge les référentiels de CodeCatalyst packages.

## Rubriques

- [Utilisation de référentiels de CodeCatalyst packages dans les flux de travail](#)
- [Exemples d'utilisation de packages dans les flux de travail](#)

## Utilisation de référentiels de CodeCatalyst packages dans les flux de travail

Dans CodeCatalyst, vous pouvez ajouter un référentiel de CodeCatalyst packages à vos actions de compilation et de test dans votre flux de travail. Votre référentiel de packages doit être configuré avec un format de package, tel que npm. Vous pouvez également choisir d'inclure une séquence de portées pour le référentiel de packages sélectionné.

Utilisez les instructions suivantes pour spécifier la configuration de package à utiliser avec une action de flux de travail.

### Visual

Pour spécifier la configuration du package qu'une action utilisera (éditeur visuel)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez Visual.
7. Dans le diagramme du flux de travail, choisissez l'action que vous souhaitez configurer avec un référentiel de packages.
8. Choisissez Packages.
9. Dans le menu déroulant Ajouter une configuration, choisissez la configuration du package que vous souhaitez utiliser avec vos actions de flux de travail.
10. Choisissez Ajouter un référentiel de packages.
11. Dans le menu déroulant du référentiel de packages, spécifiez le nom du référentiel de CodeCatalyst packages que vous souhaitez utiliser par l'action.

Pour plus d'informations sur les référentiels de packages, consultez [Référentiels de packages](#).

12. (Facultatif) Dans Étendue - facultatif, spécifiez une séquence de portées que vous souhaitez définir dans votre registre de packages.

Lors de la définition des étendues, le référentiel de packages spécifié est configuré comme registre pour toutes les étendues répertoriées. Si un package avec cette portée est demandé

via le client npm, il utilisera ce référentiel au lieu du dépôt par défaut. Le nom de chaque étendue doit être préfixé par « @ ».

S'il Scopes est omis, le référentiel de packages spécifié est configuré comme registre par défaut pour tous les packages utilisés par l'action.

Pour plus d'informations sur les étendues, reportez-vous à la section [Espaces de noms de packages](#) et Packages [délimités](#).

13. Choisissez Ajouter.
14. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
15. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour spécifier la configuration du package qu'une action utilisera (éditeur YAML)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. Dans une action, ajoutez un code similaire au suivant :

```
action-name:
  Configuration:
    Packages:
      NpmConfiguration:
        PackageRegistries:
          - PackagesRepository: package-repository
        Scopes:
          - "@scope"
```

Pour plus d'informations, consultez la description de la Packages propriété dans [Créez et testez une référence d'actions](#) Pour votre action.

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.



9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Exemples d'utilisation de packages dans les flux de travail

Les exemples suivants montrent comment référencer des packages dans le fichier de définition du flux de travail.

### Rubriques

- [Exemple : définition de packages avec NpmConfiguration](#)
- [Exemple : remplacement du registre par défaut](#)
- [Exemple : remplacement des étendues dans le registre de vos packages](#)

### Exemple : définition de packages avec **NpmConfiguration**

L'exemple suivant montre comment définir un package NpmConfiguration dans votre fichier de définition de flux de travail.

```
Actions:
  Build:
    Identifieur: aws/build-beta@v1
    Configuration:
      Packages:
        NpmConfiguration:
          PackageRegistries:
            - PackagesRepository: main-repo
            - PackagesRepository: scoped-repo
          Scopes:
            - "@scope1"
```

Cet exemple configure le client npm comme tel :

```
default: main-repo
@scope1: scoped-repo
```

Dans cet exemple, deux référentiels sont définis. Le registre par défaut est défini main-repo tel qu'il est défini sans étendue. @scope1Le champ d'application est configuré dans PackageRegistries pour scoped-repo.

## Exemple : remplacement du registre par défaut

L'exemple suivant montre comment remplacer le registre par défaut.

```
NpmConfiguration:
  PackageRegistries:
    - PackagesRepository: my-repo-1
    - PackagesRepository: my-repo-2
    - PackagesRepository: my-repo-3
```

Cet exemple configure le client npm comme tel :

```
default: my-repo-3
```

Si vous spécifiez plusieurs référentiels par défaut, le dernier sera prioritaire. Dans cet exemple, le dernier référentiel répertorié est `my-repo-3`, ce qui signifie que npm se connectera à `my-repo-3`. Cela remplace les référentiels `my-repo-1` et `my-repo-2`.

## Exemple : remplacement des étendues dans le registre de vos packages

L'exemple suivant vous montre comment remplacer une étendue dans votre registre de packages.

```
NpmConfiguration:
  PackageRegistries:
    - PackagesRepository: my-default-repo
    - PackagesRepository: my-repo-1
  Scopes:
    - "@scope1"
    - "@scope2"
  - PackagesRepository: my-repo-2
  Scopes:
    - "@scope2"
```

Cet exemple configure le client npm comme tel :

```
default: my-default-repo
@scope1: my-repo-1
@scope2: my-repo-2
```

Si vous incluez des étendues de remplacement, le dernier référentiel aura la priorité. Dans cet exemple, la dernière fois que l'étendue @scope2 a été configurée dans PackageRegistries est pourmy-repo-2. Cela remplace le champ d'application @scope2 configuré pourmy-repo-1.

## Travailler avec des courses

Une exécution est une itération unique d'un flux de travail. Au cours d'une exécution, CodeCatalyst exécute les actions définies dans le fichier de configuration du flux de travail et génère les journaux, artefacts et variables associés.

### Rubriques

- [Démarrage d'une exécution de flux de travail](#)
- [Arrêt d'une exécution de flux de travail](#)
- [Afficher le statut et les détails de l'exécution du flux de travail](#)
- [Configuration des exécutions en file d'attente, remplacées et parallèles](#)

### Démarrage d'une exécution de flux de travail

Utilisez la procédure suivante pour démarrer un flux de travail exécuté manuellement.

#### Note

Vous pouvez également démarrer automatiquement un flux de travail en [configurant un déclencheur](#).

### Pour démarrer un flux de travail exécuté manuellement

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom du flux de travail que vous souhaitez exécuter. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Cliquez sur Exécuter.

## Arrêt d'une exécution de flux de travail

Utilisez la procédure suivante pour arrêter une exécution de flux de travail en cours. Vous voudrez peut-être arrêter une course si elle a été lancée par accident.

Lorsque vous arrêtez l'exécution d'un flux de CodeCatalyst travail, attendez que les actions en cours soient terminées avant de marquer l'exécution comme arrêtée dans la CodeCatalyst console. Les actions qui n'ont pas eu la chance de démarrer ne seront pas lancées et seront marquées comme abandonnées.

### Note

Si une exécution est en file d'attente (c'est-à-dire qu'aucune action n'est en cours), elle est immédiatement arrêtée.

Pour arrêter l'exécution d'un flux de travail

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Sous Workflows, choisissez Exécutions et choisissez l'exécution en cours dans la liste.
5. Choisissez Arrêter.

## Afficher le statut et les détails de l'exécution du flux de travail

Vous pouvez consulter le statut et les détails d'une seule exécution de flux de travail ou de plusieurs exécutions en même temps.

### Note

Vous pouvez également consulter l'état du flux de travail, qui est différent de l'état d'exécution du flux de travail. Pour de plus amples informations, veuillez consulter [Affichage de l'état du flux de travail](#).

## Rubriques

- [Statuts d'exécution du flux de travail](#)

- [Afficher le statut et les détails d'une seule exécution](#)
- [Afficher le statut et les détails de toutes les exécutions de votre projet](#)
- [Afficher le statut et les détails de toutes les exécutions d'un flux de travail spécifique](#)
- [Affichage des exécutions d'un flux de travail dans le diagramme du flux de travail](#)

## Statuts d'exécution du flux de travail

L'exécution d'un flux de travail peut avoir l'un des statuts suivants :

- Succès : le flux de travail a été correctement exécuté.
- Échec : une ou plusieurs actions de l'exécution du flux de travail ont échoué.
- En cours — Le flux de travail est en cours de traitement.
- Arrêté : une personne a arrêté l'exécution du flux de travail alors qu'il était en cours.
- Arrêt — L'exécution du flux de travail est actuellement arrêtée.
- Annulé — L'exécution du flux de travail a été annulée CodeCatalyst parce que le flux de travail associé a été supprimé ou mis à jour alors que l'exécution était en cours d'exécution.
- Remplacé : ne se produit que si vous avez configuré le mode d'[exécution remplacé](#). L'exécution du flux de travail a été annulée CodeCatalyst car une exécution ultérieure l'a remplacée.

## Afficher le statut et les détails d'une seule exécution

Vous souhaitez peut-être consulter le statut et les détails d'une seule exécution de flux de travail pour vérifier si elle a réussi, à quelle heure elle s'est terminée ou pour savoir qui ou quoi l'a lancée.

Pour afficher le statut et les détails d'une seule exécution

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Sous le nom du flux de travail, choisissez Runs.
6. Dans Historique des exécutions, dans la colonne Run ID, sélectionnez une exécution. Par exemple, Run-95a4d.
7. Sous le nom de l'exécution, effectuez l'une des opérations suivantes :

- Visuel illustrant un diagramme de flux de travail montrant les actions de votre flux de travail exécutées et leur statut (voir [Statuts d'exécution du flux de travail](#)). Cette vue montre également le référentiel source et la branche utilisés pendant l'exécution.

Dans le diagramme du flux de travail, choisissez une action pour voir les détails tels que les journaux, les rapports et les résultats générés par l'action pendant l'exécution. Les informations affichées dépendent du type d'action sélectionné. Pour plus d'informations sur l'affichage des journaux de création ou de déploiement, consultez [Afficher les résultats d'une action de construction](#) ou [Afficher les journaux de déploiement](#).

- YAML pour voir le fichier de définition du flux de travail utilisé pour l'exécution.
- Artefacts permettant de voir les artefacts produits par le flux de travail exécuté. Pour plus d'informations sur les artefacts, consultez [Travailler avec des artefacts](#).
- Rapports permettant de consulter les rapports de test et les autres types de rapports produits par le flux de travail exécuté. Pour plus d'informations sur les rapports, consultez [Types de rapports de test](#).
- Variables permettant de voir les variables de sortie produites par le flux de travail exécuté. Pour plus d'informations sur les variables, consultez [Utilisation des variables](#).

#### Note

Si le flux de travail parent de l'exécution a été supprimé, un message indiquant ce fait apparaît en haut de la page des détails de l'exécution.

Afficher le statut et les détails de toutes les exécutions de votre projet

Vous souhaitez peut-être consulter le statut et les détails de toutes les exécutions de flux de travail au sein de votre projet, comprendre le niveau d'activité du flux de travail dans votre projet et en savoir plus sur l'état général de vos flux de travail.


Pour consulter le statut et les détails de toutes les exécutions de votre projet

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Sous Workflows, choisissez Runs.

Toutes les exécutions, pour tous les flux de travail, dans toutes les branches, dans tous les référentiels de votre projet, sont affichées.

La page comprend les colonnes suivantes :

- ID d'exécution : identifiant unique de l'exécution. Cliquez sur le lien Run ID pour afficher des informations détaillées sur l'exécution.
- État — État du traitement du flux de travail exécuté. Pour plus d'informations sur les statuts d'exécution, consultez [Statuts d'exécution du flux de travail](#).
- Déclencheur : personne, commit, pull request (PR) ou calendrier qui a lancé l'exécution du flux de travail. Pour de plus amples informations, veuillez consulter [Utilisation de déclencheurs](#).
- Flux de travail : nom du flux de travail pour lequel une exécution a été lancée, ainsi que du référentiel source et de la branche où réside le fichier de définition du flux de travail. Vous devrez peut-être augmenter la largeur des colonnes pour voir ces informations.

 Note

Si cette colonne est définie sur Non disponible, c'est généralement parce que le flux de travail associé a été supprimé ou déplacé.

- Heure de début : heure à laquelle l'exécution du flux de travail a commencé.
- Durée : durée de traitement du flux de travail. Des durées très longues ou très courtes peuvent indiquer des problèmes.
- Heure de fin : heure à laquelle l'exécution du flux de travail s'est terminée.

Afficher le statut et les détails de toutes les exécutions d'un flux de travail spécifique

Vous souhaitez peut-être consulter le statut et les détails de toutes les exécutions associées à un flux de travail spécifique pour voir si certaines d'entre elles créent des goulots d'étranglement dans le flux de travail, ou pour voir quelles exécutions sont en cours ou terminées.

Pour afficher le statut et les détails de toutes les exécutions d'un flux de travail spécifique

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.

4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Sous le nom du flux de travail, choisissez Runs.

Les exécutions associées au flux de travail choisi apparaissent.

La page est divisée en deux sections :

- Exécutions actives : affiche les séries en cours. Ces courses se dérouleront dans l'un des états suivants : En cours.
- Historique des exécutions : affiche les exécutions terminées (c'est-à-dire non en cours).

Pour plus d'informations sur les statuts d'exécution, consultez [Statuts d'exécution du flux de travail](#).

### Affichage des exécutions d'un flux de travail dans le diagramme du flux de travail

Vous pouvez consulter le statut de toutes les exécutions d'un flux de travail au fur et à mesure qu'elles progressent ensemble dans le flux de travail. Les exécutions sont affichées dans le diagramme du flux de travail (et non dans une vue de liste). Cela vous donne une représentation visuelle des exécutions traitées par quelles actions, et des exécutions en attente dans une file d'attente.

Pour afficher le statut de plusieurs exécutions au fur et à mesure qu'elles progressent ensemble dans un flux de travail


#### Note

Cette procédure s'applique uniquement si votre flux de travail utilise le mode d'exécution en file d'attente ou remplacé. Pour de plus amples informations, veuillez consulter [Configuration des exécutions en file d'attente, remplacées et parallèles](#).

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.



4. Choisissez le nom du flux de travail contenant les exécutions que vous souhaitez afficher. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.

 Note

Assurez-vous de consulter une page de flux de travail et non une page d'exécution.

5. Choisissez l'onglet Dernier état en haut à gauche.

Un diagramme de flux de travail apparaît.

6. Consultez le diagramme du flux de travail. Le diagramme montre toutes les exécutions en cours dans le flux de travail, ainsi que les dernières exécutions terminées. Plus précisément :
  - Les exécutions qui apparaissent en haut, avant Sources, sont mises en file d'attente et attendent de démarrer.
  - Les exécutions qui apparaissent entre les actions sont mises en file d'attente et attendent d'être traitées par l'action suivante.
  - Les exécutions qui apparaissent dans une action sont 1. en cours de traitement par l'action, 2. ont fini d'être traitées par l'action, ou 3. n'ont pas été traitées par l'action (généralement parce qu'une action précédente a échoué).

## Configuration des exécutions en file d'attente, remplacées et parallèles

Par défaut, lorsque plusieurs exécutions de flux de travail ont lieu en même temps, elles sont mises en CodeCatalyst file d'attente et traitées une par une, dans l'ordre dans lequel elles ont été démarrées. Vous pouvez modifier ce comportement par défaut en spécifiant un mode d'exécution. Il existe quelques modes d'exécution :

- (Par défaut) Mode d'exécution en file d'attente : CodeCatalyst les processus s'exécutent un par un
- Mode d'exécution remplacé : CodeCatalyst les processus s'exécutent un par un, les nouvelles exécutions remplaçant les anciennes
- Mode d'exécution parallèle : CodeCatalyst les processus s'exécutent en parallèle

## Rubriques

- [À propos du mode d'exécution en file d'attente](#)

- [À propos du mode d'exécution remplacé](#)
- [À propos du mode d'exécution parallèle](#)
- [Modification du mode d'exécution](#)

## À propos du mode d'exécution en file d'attente

En mode exécution en file d'attente, les exécutions se produisent en série, les séries en attente formant une file d'attente.

Les files d'attente se forment aux points d'entrée des actions et des groupes d'actions. Vous pouvez donc avoir plusieurs files d'attente dans le même flux de travail (voir [Figure 1](#)). Lorsqu'une exécution en file d'attente entre dans une action, celle-ci est verrouillée et aucune autre exécution ne peut y participer. Lorsque l'exécution se termine et quitte l'action, l'action est déverrouillée et prête pour la prochaine exécution.

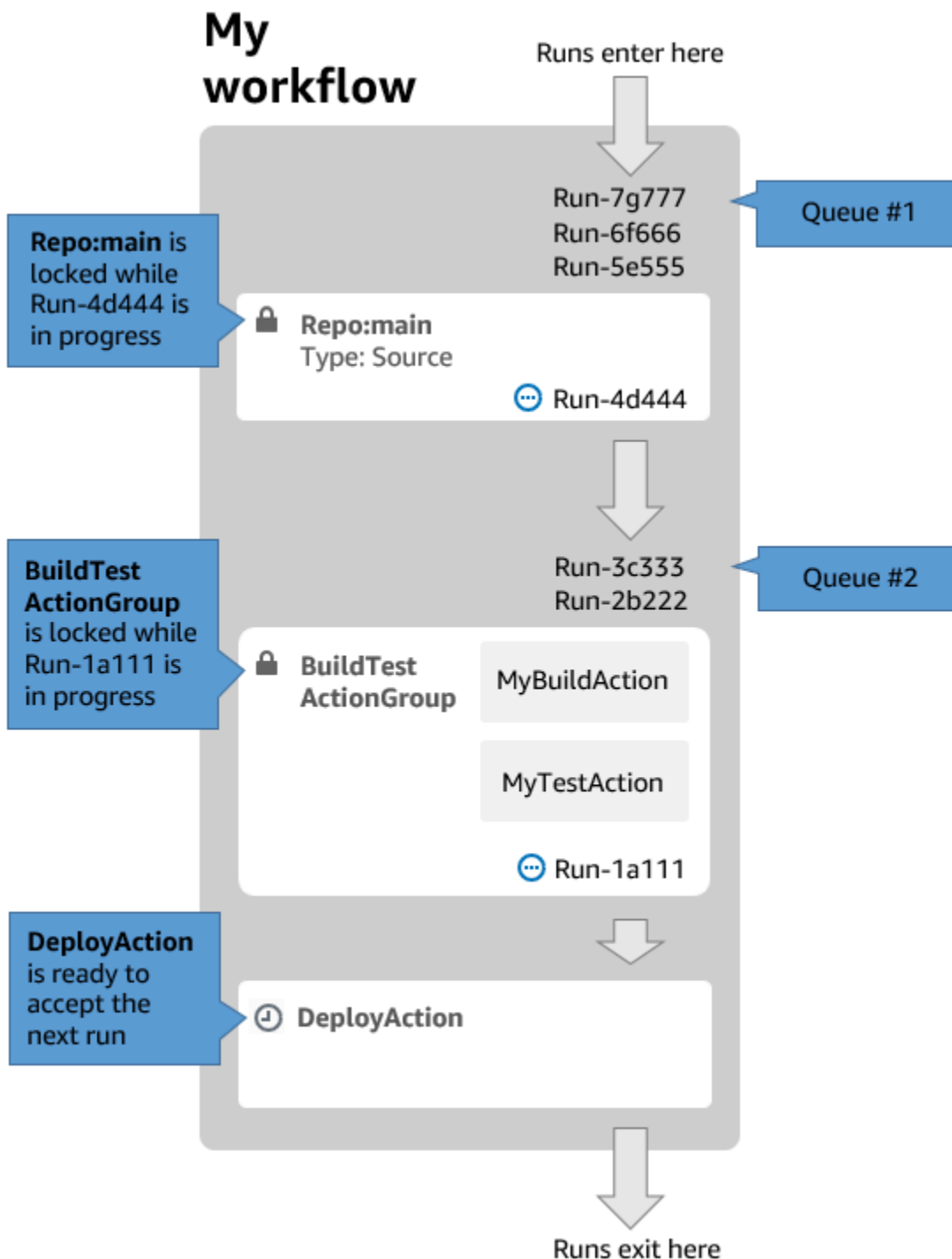
[Figure 1](#) illustre un flux de travail configuré en mode d'exécution en file d'attente. Cela montre :

- Sept cycles se frayent un chemin dans le flux de travail.
- Deux files d'attente : une en dehors de l'entrée de la source d'entrée (Repo:Main) et une en dehors de l'entrée de l'action. BuildTestActionGroup
- Deux blocs verrouillés : la source d'entrée (Repo:main) et le. BuildTestActionGroup

Voici comment les choses se dérouleront au fur et à mesure que le flux de travail terminera le traitement :

- Lorsque Run-4d444 aura fini de cloner le référentiel source, il quittera la source d'entrée et rejoindra la file d'attente située derrière Run-3c333. Ensuite, RUN-5e555 entrera dans la source d'entrée.
- Lorsque Run-1a111 aura fini de créer et de tester, il quittera l'BuildTestActionGroupaction et entrera dans l'action. DeployAction Ensuite, Run-2b222 entrera dans l'action. BuildTestActionGroup

Figure 1 : Un flux de travail configuré en « mode exécution en file d'attente »



Utilisez le mode d'exécution en file d'attente si :

- Vous souhaitez conserver une one-to-one relation entre les fonctionnalités et les exécutions. Ces fonctionnalités peuvent être regroupées lorsque vous utilisez le mode remplacé. Par exemple, lorsque vous fusionnez la fonctionnalité 1 dans le commit 1, l'exécution 1 démarre, et lorsque vous fusionnez l'entité 2 dans le commit 2, l'exécution 2 démarre, etc. Si vous deviez utiliser le

mode remplacé au lieu du mode en file d'attente, vos fonctionnalités (et vos validations) seront regroupées lors de l'exécution qui remplacera les autres.

- Vous souhaitez éviter les conditions de course et les problèmes inattendus qui peuvent survenir lors de l'utilisation du mode parallèle. Par exemple, si deux développeurs de logiciels, Wang et Saanvi, lancent des exécutions de flux de travail à peu près au même moment pour les déployer sur un cluster Amazon ECS, l'exécution de Wang peut lancer des tests d'intégration sur le cluster tandis que celle de Saanvi déploie un nouveau code d'application sur le cluster, ce qui entraîne l'échec des tests de Wang ou le test du mauvais code. Autre exemple, vous pouvez avoir une cible dépourvue de mécanisme de verrouillage, auquel cas les deux essais peuvent remplacer les modifications de l'autre de manière inattendue.
- Vous souhaitez limiter la charge sur les ressources de calcul CodeCatalyst utilisées pour traiter vos exécutions. Par exemple, si votre flux de travail comporte trois actions, vous pouvez exécuter au maximum trois exécutions en même temps. L'imposition d'une limite au nombre d'exécutions pouvant être effectuées simultanément rend le débit d'exécution plus prévisible.
- Vous souhaitez limiter le nombre de demandes adressées à des services tiers par le flux de travail. Par exemple, votre flux de travail peut comporter une action de génération qui inclut des instructions pour extraire une image de Docker Hub. [Docker Hub limite le nombre de pull requests](#) que vous pouvez effectuer à un certain nombre par heure et par compte, et vous serez bloqué si vous dépassez cette limite. L'utilisation du mode d'exécution en file d'attente pour ralentir votre débit d'exécution aura pour effet de générer moins de demandes à Docker Hub par heure, limitant ainsi les risques de verrouillages et d'échecs de compilation et d'exécution qui en résultent.

Taille maximale de la file d'attente : 50

Remarques sur la taille maximale de la file d'attente :

- La taille maximale de la file d'attente fait référence au nombre maximum d'exécutions autorisées dans toutes les files d'attente du flux de travail.
- Si une file d'attente dépasse 50 essais, la 51e et les suivantes sont supprimées. CodeCatalyst

Comportement de défaillance :

Si une exécution ne répond plus alors qu'elle est traitée par une action, les exécutions qui la suivent sont maintenues dans la file d'attente jusqu'à ce que l'action expire. Les actions expirent au bout d'une heure.

Si une exécution échoue dans le cadre d'une action, la première exécution en file d'attente qui la suit est autorisée à se poursuivre.

### À propos du mode d'exécution remplacé

Le mode d'exécution remplacé est identique au mode d'exécution en file d'attente sauf que :

- Si une exécution en file d'attente rattrape une autre course de la file d'attente, la dernière remplace (remplace) la précédente, et la précédente est annulée et marquée comme « remplacée ».
- En raison du comportement décrit dans le premier point, une file d'attente ne peut inclure qu'une seule exécution lorsque le mode d'exécution remplacé est utilisé.

En utilisant le flux de travail [Figure 1](#) comme guide, l'application d'un mode d'exécution remplacé à ce flux de travail entraînerait les résultats suivants :

- Run-7g777 remplacerait les deux autres exécutions de sa file d'attente et serait la seule exécution restante dans la file #1. Run-6f666 et Run-5e555 seraient annulés.
- Run-3c333 remplacerait Run-2b222 et serait le seul run restant dans la file d'attente #2. Run-2b222 serait annulé.

Utilisez le mode d'exécution remplacé si vous souhaitez :

- meilleur débit qu'en mode file d'attente
- encore moins de demandes adressées à des services tiers qu'en mode file d'attente ; cela est avantageux si le service tiers impose des limites de débit, comme Docker Hub

### À propos du mode d'exécution parallèle

En mode d'exécution parallèle, les exécutions sont indépendantes les unes des autres et n'attendent pas la fin des autres exécutions pour démarrer. Il n'y a pas de files d'attente et le débit d'exécution n'est limité que par la rapidité avec laquelle les actions du flux de travail sont effectuées.

Utilisez le mode d'exécution parallèle dans les environnements de développement où chaque utilisateur possède sa propre branche de fonctionnalités et effectue des déploiements sur des cibles qui ne sont pas partagées par d'autres utilisateurs.

### Important

Si vous avez une cible partagée vers laquelle plusieurs utilisateurs peuvent effectuer un déploiement, telle qu'une fonction Lambda dans un environnement de production, n'utilisez pas le mode parallèle, car cela pourrait entraîner des conditions de course. Une situation de course se produit lorsque des workflows parallèles tentent de modifier une ressource partagée en même temps, ce qui entraîne des résultats imprévisibles.

Nombre maximum de courses en parallèle : 1000 par CodeCatalyst espace

### Modification du mode d'exécution

Vous pouvez définir le mode d'exécution sur `queued`, `superseded` ou `parallel`. La valeur par défaut est mise en file d'attente.

Lorsque vous passez du mode d'exécution en file d'attente ou remplacé au mode parallèle CodeCatalyst, vous annulez les exécutions mises en file d'attente et autorisez les exécutions en cours de traitement par une action à se terminer avant de les annuler.

Lorsque vous passez du mode d'exécution de `parallel` à `queued` ou `remplacé`, CodeCatalyst toutes les exécutions parallèles en cours d'exécution sont terminées. Toutes les exécutions que vous démarrez après avoir changé le mode d'exécution en mode file d'attente ou remplacé utilisent le nouveau mode.

### Visual

Pour modifier le mode d'exécution à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. En haut à droite, sélectionnez Propriétés du flux de travail.
7. Développez le mode Avancé, puis sous Mode d'exécution, choisissez l'une des options suivantes :

- a. En file d'attente — voir [À propos du mode d'exécution en file d'attente](#)
  - b. Remplacé — voir [À propos du mode d'exécution remplacé](#)
  - c. Parallèle — voir [À propos du mode d'exécution parallèle](#)
8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
  9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour modifier le mode d'exécution à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. Ajoutez la RunMode propriété, comme suit :

```
Name: Workflow_6d39
SchemaVersion: "1.0"
RunMode: QUEUED|SUPERSEDED|PARALLEL
```

Pour plus d'informations, consultez la description de la RunMode propriété dans la [Propriétés de haut niveau](#) section du [Référence de définition du flux de travail](#).

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Travailler avec des secrets

Il peut arriver que vous deviez utiliser des données sensibles, telles que des informations d'authentification, dans vos flux de travail. Il faut éviter de stocker ces valeurs en texte clair n'importe où dans votre dépôt, car toute personne ayant accès au référentiel contenant le secret peut les

voir. De même, ces valeurs ne doivent pas être utilisées directement dans les définitions de flux de travail, car elles seront visibles sous forme de fichiers dans votre référentiel. Avec CodeCatalyst, vous pouvez protéger ces valeurs en ajoutant un secret à votre projet, puis en le référençant dans le fichier de définition de votre flux de travail. Notez que vous pouvez avoir un maximum de cinq secrets par action.

#### Note

Les secrets ne peuvent être utilisés que pour remplacer les mots de passe et les informations sensibles dans le fichier de définition du flux de travail.

## Rubriques

- [Création d'un secret](#)
- [Modifier un secret](#)
- [Utiliser un secret](#)
- [Supprimer un secret](#)

## Création d'un secret

Pour créer un secret, procédez comme suit. Le secret contient les informations sensibles que vous souhaitez masquer.

#### Note

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

## Pour créer un secret

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans le volet de navigation, choisissez CI/CD, puis Secrets.
3. Choisissez Create secret (Créer un secret).
4. Entrez les informations suivantes :



## Nom

Entrez un nom pour votre secret.

## Valeur

Entrez la valeur du secret. Il s'agit des informations sensibles que vous souhaitez masquer.

Par défaut, la valeur n'est pas affichée. Pour afficher la valeur, choisissez Afficher la valeur.

## Description

(Facultatif) Entrez une description de votre secret.

5. Choisissez Créer.

## Modifier un secret

Pour modifier un secret, procédez comme suit.

Pour modifier un secret

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans le volet de navigation, choisissez CI/CD, puis Secrets.
3. Dans la liste des secrets, choisissez le secret que vous souhaitez modifier.
4. Choisissez Modifier.
5. Modifiez les propriétés suivantes :

## Valeur

Entrez la valeur du secret. Il s'agit de la valeur que vous souhaitez masquer. Par défaut, la valeur n'est pas affichée.

## Description

(Facultatif) Entrez une description de votre secret.

6. Choisissez Enregistrer.

## Utiliser un secret

Pour utiliser un secret dans une action de flux de travail, vous devez obtenir l'identifiant de référence du secret et utiliser cet identifiant dans l'action de flux de travail.

## Rubriques

- [Obtenir l'identifiant d'un secret](#)
- [Référencer un secret dans un flux de travail](#)

### Obtenir l'identifiant d'un secret

Pour obtenir l'identifiant de référence du secret, procédez comme suit. Vous allez ajouter cet identifiant à votre flux de travail.

Pour obtenir l'identifiant de référence du secret

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans le volet de navigation, choisissez CI/CD, puis Secrets.
3. Dans la liste des secrets, trouvez le secret que vous souhaitez utiliser.
4. Dans la colonne Numéro de référence, copiez l'identifiant du secret. La syntaxe de l'ID de référence est la suivante :

```
`${Secrets.<name>}
```

### Référencer un secret dans un flux de travail

Utilisez la procédure suivante pour référencer un secret dans un flux de travail.

Pour faire référence à un secret

1. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
2. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
3. Choisissez Modifier.
4. Choisissez YAML.
5. Modifiez le code YAML pour utiliser l'identifiant du secret. Par exemple, pour utiliser un nom d'utilisateur et un mot de passe enregistrés sous forme de secrets avec la `curl` commande, vous devez utiliser une Run commande similaire à la suivante :

```
- Run: curl -u <username-secret-identifrier>:<password-secret-identifrier> https://  
example.com
```

6. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
7. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Supprimer un secret

Pour supprimer un secret et son identifiant de référence, procédez comme suit.

### Note

Avant de supprimer un secret, nous vous recommandons de supprimer son identifiant de référence pour toutes les actions du flux de travail. Si vous supprimez le secret sans supprimer l'identifiant de référence, l'action échouera lors de sa prochaine exécution.

Pour supprimer l'identifiant de référence d'un secret d'un flux de travail

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
3. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
4. Choisissez Modifier.
5. Choisissez YAML.
6. Recherchez la chaîne suivante dans le flux de travail :

```
`${Secrets}.
```

Cela permet de trouver tous les identifiants de référence de tous les secrets.

7. Supprimez l'identifiant de référence du secret choisi ou remplacez-le par une valeur en texte brut.
8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

Pour supprimer le secret

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).

2. Dans le volet de navigation, choisissez CI/CD, puis Secrets.
3. Dans la liste des secrets, choisissez le secret que vous souhaitez supprimer.
4. Sélectionnez Delete (Supprimer).
5. Entrez **delete** pour confirmer la suppression.
6. Sélectionnez Supprimer.

## Travailler avec des sources

Une source, également appelée source d'entrée, est un référentiel de sources auquel une [action de flux](#) de travail doit accéder pour exécuter ses tâches. Par exemple, une action de flux de travail peut avoir besoin d'accéder à une source pour obtenir des tests unitaires et les exécuter sur les fichiers source de votre application.

CodeCatalyst les flux de travail prennent en charge les sources suivantes :

- CodeCatalyst référentiels de sources — Pour plus d'informations, consultez [Référentiels de sources dans CodeCatalyst](#).
- GitHub référentiels de sources — Pour plus d'informations, consultez [Utilisation de GitHub référentiels dans CodeCatalyst](#).

### Rubriques

- [Spécification de la source qui stockera le fichier de définition du flux de travail](#)
- [Spécification de la source qu'une action de flux de travail utilisera](#)
- [Référencement de fichiers dans un référentiel source](#)
- [Variables produites par la source](#)

### Spécification de la source qui stockera le fichier de définition du flux de travail

Suivez les instructions ci-dessous pour spécifier le référentiel CodeCatalyst source dans lequel vous souhaitez stocker le fichier de définition du flux de travail. Si vous préférez spécifier un dépôt GitHub source, consultez plutôt [Utilisation de GitHub référentiels dans CodeCatalyst](#).

Le référentiel source dans lequel réside votre fichier de définition de flux de travail est identifié par l'étiquette, `WorkflowSource`.

**Note**

Vous spécifiez le référentiel source dans lequel se trouve votre fichier de définition de flux de travail lorsque vous validez votre fichier de définition de flux de travail pour la première fois. Après cette validation, le référentiel et le fichier de définition du flux de travail sont liés ensemble de manière permanente. La seule façon de modifier le référentiel après la validation initiale est de recréer le flux de travail dans un autre référentiel.

Pour spécifier le référentiel source qui stockera le fichier de définition du flux de travail

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez Créer un flux de travail et créez le flux de travail. Pour en savoir plus, consultez [Pour créer un flux de travail à l'aide de l'éditeur visuel](#).

Au cours du processus de création du flux de travail, il vous est demandé de spécifier le CodeCatalyst référentiel dans lequel vous souhaitez stocker le fichier de définition du flux de travail.

## Spécification de la source qu'une action de flux de travail utilisera

Suivez les instructions ci-dessous pour spécifier le référentiel source à utiliser avec une action de flux de travail. Au démarrage, l'action regroupe les fichiers du référentiel source configuré dans un artefact, télécharge l'artefact sur l'[image Docker de l'environnement d'exécution](#) dans lequel l'action est exécutée, puis termine son traitement à l'aide des fichiers téléchargés.

**Note**

Actuellement, dans le cadre d'une action de flux de travail, vous ne pouvez spécifier qu'un seul référentiel source, à savoir le référentiel source dans lequel réside le fichier de définition du flux de travail (dans le `.codecatalyst/workflows/` répertoire). Ce référentiel source est représenté par l'étiquette `WorkflowSource`.

## Visual

Pour spécifier le référentiel source qu'une action utilisera (éditeur visuel)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez Visual.
7. Dans le diagramme du flux de travail, choisissez l'action pour laquelle vous souhaitez spécifier la source.
8. Choisissez Entrées.
9. Dans Sources (facultatif), procédez comme suit :

Spécifiez les étiquettes qui représentent les référentiels sources qui seront nécessaires à l'action. Actuellement, la seule étiquette prise en charge est `WorkflowSource`, qui représente le référentiel source dans lequel votre fichier de définition de flux de travail est stocké.

Si vous omettez une source, vous devez spécifier au moins un artefact d'entrée sous.

`action-name/Inputs/Artifacts`

Pour plus d'informations sur les sources, consultez [Travailler avec des sources](#).

10. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
11. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour spécifier le référentiel source qu'une action utilisera (éditeur YAML)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.

4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. Dans une action, ajoutez un code similaire au suivant :

```
action-name:  
  Inputs:  
    Sources:  
      - WorkflowSource
```

Pour plus d'informations, consultez la description de la Sources propriété dans [Référence de définition du flux de travail](#) Pour votre action.

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Référencement de fichiers dans un référentiel source

Si des fichiers se trouvent dans un référentiel source et que vous devez y faire référence dans l'une des actions de votre flux de travail, procédez comme suit.

### Note

Voir aussi [Référencement de fichiers dans un artefact](#).

## Pour référencer un fichier dans un référentiel source

- Dans l'action dans laquelle vous souhaitez référencer un fichier, ajoutez un code similaire au suivant :

```
Actions:  
  My-action:  
    Inputs:  
      Sources:  
        - WorkflowSource  
    Configuration:
```

```
Steps:  
- run: cd my-app && cat file1.jar
```

Dans le code précédent, l'action recherche dans le my-app répertoire situé à la racine du référentiel WorkflowSource source pour rechercher et afficher le file1.jar fichier.

## Variables produites par la source

Lorsqu'un flux de travail s'exécute, sa source produit des variables que vous pouvez utiliser dans les actions de flux de travail suivantes. Pour plus de détails, voir [variables «BranchName" et CommitId "»](#) dans le [Liste des variables prédéfinies](#).

## Utilisation de déclencheurs

Un déclencheur de flux de travail, ou simplement un déclencheur, vous permet de démarrer automatiquement un flux de travail lorsque certains événements se produisent, comme un envoi de code. Vous souhaitez peut-être configurer des déclencheurs pour éviter à vos développeurs de logiciels d'avoir à lancer des exécutions de flux de travail manuellement via la CodeCatalyst console.

Vous pouvez utiliser trois types de déclencheurs :

- Push : un déclencheur de code push déclenche l'exécution d'un flux de travail chaque fois qu'un commit est envoyé.
- Demande d'extraction : un déclencheur de demande d'extraction entraîne le démarrage d'un flux de travail chaque fois qu'une demande d'extraction est créée, révisée ou fermée.
- Planification : un déclencheur de planification entraîne le démarrage d'un flux de travail selon un calendrier que vous définissez. Envisagez d'utiliser un déclencheur de planification pour exécuter des versions nocturnes de votre logiciel afin que la dernière version soit prête à être utilisée par vos développeurs le lendemain matin.

Vous pouvez utiliser les déclencheurs push, pull request et schedule seuls ou en combinaison dans le même flux de travail.

### Tip

Pour voir un déclencheur en action, lancez un projet avec un plan. La plupart des plans contiennent un flux de travail avec un déclencheur. Recherchez la Trigger propriété dans



le fichier de définition du flux de travail du plan. Pour plus d'informations sur les modèles de présentation, consultez [Création d'un projet à l'aide d'un plan](#).

## Rubriques

- [Une configuration de déclencheur commune](#)
- [Considérations relatives aux déclencheurs lors du branchement](#)
- [Ajouter un déclencheur push, pull ou Schedule](#)
- [Exemples de déclencheurs](#)

## Une configuration de déclencheur commune

Cette section décrit comment configurer des déclencheurs pour une version logicielle commune et une stratégie de branchement.

Stratégie de lancement du logiciel et de branche :

- Le code de l'application se trouve dans un référentiel source.
- Votre main branche contient du code finalisé qui est toujours prêt à être publié.
- Vos développeurs de logiciels apportent leurs modifications dans les branches de fonctionnalités situées à l'extérieur de la main branche.
- Vos développeurs de logiciels [créent une pull request](#) demandant de fusionner leur branche de fonctionnalités main lorsque leur fonctionnalité sera prête.

Vous souhaitez que cette pull request lance automatiquement un flux de travail qui crée et teste, mais ne déploie pas, l'application à l'aide des fichiers de la branche des fonctionnalités du développeur du logiciel.

- Vous, les développeurs de logiciels, vérifiez le build et les tests pour vous assurer que tout est en bon état. Ils [fusionnent ensuite la pull request](#) dans la main branche.

Vous souhaitez que la fusion lance automatiquement un flux de travail qui génère et déploie le code de votre application.

Configuration du flux de travail/du déclencheur proposée :

Compte tenu de la stratégie de branchement logicielle décrite précédemment, vous souhaitez peut-être utiliser deux flux de travail :

- Workflow 1 crée et teste votre application lorsqu'une pull request est créée ou révisée.
- Workflow 2 crée et déploie votre application lorsqu'une pull request est fusionnée.

Le flux de travail 1 ressemblerait à ceci :

```
Triggers:
  - Type: PULLREQUEST
  Branches:
    - main
  Events:
    - OPEN
    - REVISION
Actions:
  BuildAction:
    instructions-for-building-the-app
  TestAction:
    instructions-for-test-the-app
```

Le code de déclenchement précédent lance automatiquement un flux de travail chaque fois qu'un développeur de logiciel crée une pull request (ou en [modifie une](#)) demandant de fusionner sa branche de fonctionnalités avec la main branche. CodeCatalyst lance un flux de travail en utilisant le code de la branche source (c'est-à-dire la branche des fonctionnalités du développeur). Le flux de travail crée et déploie l'application.

Le flux de travail 2 ressemblerait à ceci :

```
Triggers:
  - Type: PUSH
  Branches:
    - main
Actions:
  BuildAction:
    instructions-for-building-the-app
  DeployAction:
    instructions-for-deploying-the-app
```

Dans le code de déclenchement précédent, lorsqu'une fusion a main lieu, le PUSH déclencheur est activé. CodeCatalyst lance un flux de travail en utilisant le code de la main branche (qui inclut désormais le code de la pull request). Le flux de travail crée et déploie l'application.

Pour obtenir des instructions sur l'ajout de déclencheurs à un fichier de définition de flux de travail, consultez [Ajouter un déclencheur push, pull ou Schedule](#).

Pour d'autres exemples de déclencheurs et des explications supplémentaires, voir [Exemples de déclencheurs](#).

## Considérations relatives aux déclencheurs lors du branchement

Cette section décrit certaines des principales considérations à prendre en compte lors de la configuration de déclencheurs incluant des branches.

- **Considération 1 :** Pour les déclencheurs de requêtes push et pull, si vous voulez spécifier une branche, vous devez spécifier la branche de destination (ou « vers ») dans la configuration du déclencheur. Ne spécifiez jamais la branche source (ou « de »).

Dans l'exemple suivant, une commande push depuis n'importe quelle branche main active le flux de travail.

```
Triggers:  
- Type: PUSH  
  Branches:  
    - main
```

Dans l'exemple suivant, une pull request provenant de n'importe quelle branche main active le flux de travail.

```
Triggers:  
- Type: PULLREQUEST  
  Branches:  
    - main  
  Events:  
    - OPEN  
    - REVISION
```

- **Considération 2 :** Pour les déclencheurs push, une fois le flux de travail activé, celui-ci s'exécute à l'aide du fichier de définition du flux de travail et des fichiers source de la branche de destination.

- **Considération 3** : Pour les déclencheurs de pull request, une fois le flux de travail activé, le flux de travail s'exécute à l'aide du fichier de définition du flux de travail et des fichiers source de la branche source (même si vous avez spécifié la branche de destination dans la configuration du déclencheur).
- **Considération 4** : Le même déclencheur dans une branche peut ne pas s'exécuter dans une autre branche.

Considérez le déclencheur suivant :

```
Triggers:  
- Type: PUSH  
  Branches:  
    - main
```

Si le fichier de définition du flux de travail contenant ce déclencheur existe dans `main` et y est cloné `test`, le flux de travail ne démarrera jamais automatiquement en utilisant les fichiers dans `test` (bien que vous puissiez démarrer le flux de travail manuellement pour savoir si vous utilisez les fichiers dans `test`). Passez en revue les considérations 1 et 2 pour comprendre pourquoi le flux de travail ne s'exécutera jamais automatiquement à l'aide des fichiers contenus dans `test`.

Pensez également au déclencheur de pull request suivant :

```
Triggers:  
- Type: PULLREQUEST  
  Branches:  
    - main  
  Events:  
    - OPEN  
    - REVISION
```

Si le fichier de définition du flux de travail contenant ce déclencheur existe dans `main`, le flux de travail ne s'exécutera jamais à l'aide des fichiers contenus dans `main`. (Toutefois, si vous créez une `test` branche à partir de `main`, le flux de travail s'exécutera en utilisant les fichiers contenus dans `test`.) Passez en revue les considérations 1 et 3 pour comprendre pourquoi.

## Ajouter un déclencheur push, pull ou Schedule

Suivez les instructions ci-dessous pour ajouter un déclencheur push, pull ou planifié à votre flux de travail.

### Visual

Pour ajouter un déclencheur (éditeur visuel)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez Visual.
7. Dans le diagramme du flux de travail, sélectionnez la case Source et déclencheurs.
8. Dans le volet de configuration, choisissez Ajouter un déclencheur.
9. Dans la boîte de dialogue Ajouter un déclencheur, entrez les informations dans les champs, comme suit.

### Type de déclencheur

Spécifiez le type de déclencheur. Vous pouvez utiliser l'une des valeurs suivantes :

- Push (éditeur visuel) ou PUSH (éditeur YAML)

Un déclencheur push lance l'exécution d'un flux de travail lorsqu'une modification est transmise à votre référentiel source. Le flux de travail exécuté utilisera les fichiers de la branche vers laquelle vous transférez (c'est-à-dire la branche de destination).

- Pull request (éditeur visuel) ou PULLREQUEST (éditeur YAML)

Un déclencheur de pull request lance un flux de travail lorsqu'une pull request est ouverte, mise à jour ou fermée dans votre référentiel source. Le flux de travail exécuté utilisera les fichiers de la branche d'où vous les extrayez (c'est-à-dire la branche source).

- Calendrier (éditeur visuel) ou SCHEDULE (éditeur YAML)

Un déclencheur de planification lance des exécutions de flux de travail selon un calendrier défini par une expression cron que vous spécifiez. Un flux de travail distinct démarrera pour chaque branche de votre référentiel source à l'aide des fichiers de la branche. (Pour limiter les branches sur lesquelles le déclencheur est activé, utilisez le champ Branches (éditeur visuel) ou la Branches propriété (éditeur YAML).)

Lorsque vous configurez un déclencheur de planification, suivez les instructions suivantes :

- N'utilisez qu'un seul déclencheur de planification par flux de travail.
- Si vous avez défini plusieurs flux de travail dans votre CodeCatalyst espace, nous vous recommandons de ne pas en planifier plus de 10 pour qu'ils démarrent simultanément.
- Assurez-vous de configurer l'expression cron du déclencheur avec un délai suffisant entre les exécutions. Pour de plus amples informations, veuillez consulter [Expression](#).

Pour obtenir des exemples, consultez [Exemples de déclencheurs](#).

### Événements pour la pull request

Ce champ n'apparaît que si vous avez sélectionné le type de déclencheur de demande Pull.

Spécifiez le type d'événements de pull request qui lanceront l'exécution d'un flux de travail.

Les valeurs valides sont les suivantes :

- Une pull request est créée (éditeur visuel) ou OPEN (éditeur YAML)

L'exécution du flux de travail démarre lorsqu'une pull request est créée.

- La pull request est fermée (éditeur visuel) ou CLOSED (éditeur YAML)

L'exécution du flux de travail démarre lorsqu'une pull request est fermée. Le comportement de l'`CLOSED` événement est délicat et il est préférable de le comprendre à l'aide d'un exemple. Pour plus d'informations, consultez [Exemple : un déclencheur avec un pull, des branches et un événement « CLOSED »](#).

- Une nouvelle révision est apportée à la pull request (éditeur visuel) ou REVISION (éditeur YAML)

L'exécution du flux de travail démarre lorsqu'une révision d'une pull request est créée. La première révision est créée lors de la création de la pull request. Ensuite, une nouvelle révision est créée chaque fois que quelqu'un envoie un nouveau commit vers la branche

source spécifiée dans la pull request. Si vous incluez l'événement REVISION dans votre déclencheur de pull request, vous pouvez l'omettre, puisqu'il s'agit d'un sur-ensemble de OPEN.

Vous pouvez spécifier plusieurs événements dans le même déclencheur de pull request.

Pour obtenir des exemples, consultez [Exemples de déclencheurs](#).

## Schedule

Ce champ n'apparaît que si vous avez sélectionné le type de déclencheur Schedule.

Spécifiez l'expression cron qui décrit le moment où vous souhaitez que votre flux de travail planifié ait lieu.

Les expressions Cron CodeCatalyst utilisent la syntaxe à six champs suivante, où chaque champ est séparé par un espace :

*minutes, heures, days-of-month, days-of-week, année*

## Exemples d'expressions cron

| Minutes | Heures | Jours du mois | Mois | Jours de la semaine | Année | Signification  |
|---------|--------|---------------|------|---------------------|-------|--|
| 0       | 0      | ?             | *    | MON-FRI             | *     | Exécute un flux de travail à minuit (UTC +0) du lundi au vendredi. |

| Minutes | Heures | Jours du mois | Mois | Jours de la semaine | Année | Signification   |
|---------|--------|---------------|------|---------------------|-------|---|
| 0       | 2      | *             | *    | ?                   | *     | Exécute un flux de travail à 2h00 (UTC+0) tous les jours.   |
| 15      | 22     | *             | *    | ?                   | *     | Exécute un flux de travail à 22 h 15 (UTC+0) tous les jours.  |
| 0/30    | 22-2   | ?             | *    | SAT-SUN             | *     | Exécute un flux de travail toutes les 30 minutes du samedi au dimanche entre 22 h 00 le jour de départ et 2 h 00 le jour suivant (UTC+0). |



| Minutes | Heures | Jours du mois | Mois | Jours de la semaine | Année     | Signification   |
|---------|--------|---------------|------|---------------------|-----------|---|
| 45      | 13     | L             | *    | ?                   | 2023-2027 | Exécute un flux de travail à 13 h 45 (UTC+0) le dernier jour du mois entre 2023 et 2027 inclus. |

Lorsque vous spécifiez des expressions cron dans CodeCatalyst, assurez-vous de suivre les instructions suivantes :

- Spécifiez une seule expression cron par SCHEDULE déclencheur.
- Mettez l'expression cron entre guillemets (") dans l'éditeur YAML.
- Spécifiez l'heure en temps universel coordonné (UTC). Les autres fuseaux horaires ne sont pas pris en charge.
- Configurez au moins 30 minutes entre les exécutions. Une cadence plus rapide n'est pas prise en charge.
- Spécifiez le *days-of-week* champ *days-of-month* ou, mais pas les deux. Si vous spécifiez une valeur ou un astérisque (\*) dans l'un des champs, vous devez utiliser un point d'interrogation (?) dans l'autre. L'astérisque signifie « tout » et le point d'interrogation signifie « tout ».

Pour plus d'exemples d'expressions cron et des informations sur les caractères génériques tels que ?, et \*L, consultez la [référence des expressions Cron](#) dans le guide de l'utilisateur Amazon EventBridge . Les expressions Cron entre EventBridge et CodeCatalyst fonctionnent exactement de la même manière.

Pour des exemples de déclencheurs de planification, voir [Exemples de déclencheurs](#).

## Branches et modèle de branches

(Facultatif)

Spécifiez les branches de votre référentiel source que le déclencheur surveille afin de savoir quand démarrer une exécution de flux de travail. Vous pouvez utiliser des modèles regex pour définir les noms de vos branches. Par exemple, utilisez `main.*` pour faire correspondre toutes les branches commençant par `main`.

Les branches à spécifier sont différentes selon le type de déclencheur :

- Pour un déclencheur, spécifiez les branches vers lesquelles vous poussez, c'est-à-dire les branches de destination. Une exécution de flux de travail démarrera par branche correspondante, en utilisant les fichiers de la branche correspondante.

Exemples : `main.*`, `mainline`

- Pour un déclencheur de pull request, spécifiez les branches vers lesquelles vous pointez, c'est-à-dire les branches de destination. Une exécution de flux de travail démarrera par branche correspondante, en utilisant le fichier de définition du flux de travail et les fichiers source de la branche source (et non de la branche correspondante).

Exemples : `main.*`, `mainline`, `v1\-.*` (correspond aux branches commençant par `v1-`)

- Pour un déclencheur de planification, spécifiez les branches contenant les fichiers que vous souhaitez que votre exécution planifiée utilise. Une exécution de flux de travail démarrera par branche correspondante, en utilisant le fichier de définition du flux de travail et les fichiers source de la branche correspondante.

Exemples : `main.*`, `version\-1\.*`

### Note

Si vous ne spécifiez pas de branches, le déclencheur surveille toutes les branches de votre référentiel source et lancera l'exécution d'un flux de travail à l'aide du fichier de définition du flux de travail et des fichiers source dans :

- La branche vers laquelle vous poussez (pour les déclencheurs). Pour de plus amples informations, veuillez consulter [Exemple : un simple déclencheur push à code](#).
- La branche d'où vous effectuez l'extraction (pour les déclencheurs de pull request). Pour de plus amples informations, veuillez consulter [Exemple : un simple déclencheur de pull request](#).
- Toutes les succursales (pour les déclencheurs de calendrier). Une exécution de flux de travail commencera par branche de votre référentiel source. Pour de plus amples informations, veuillez consulter [Exemple : un simple déclencheur de planification](#).

Pour plus d'informations sur les branches et les déclencheurs, consultez [Considérations relatives aux déclencheurs lors du branchement](#).

Pour obtenir plus d'exemples, consultez [Exemples de déclencheurs](#).

### Fichiers modifiés

Ce champ n'apparaît que si vous avez sélectionné le type de déclencheur de demande Push ou Pull.

Spécifiez les fichiers ou les dossiers de votre référentiel source que le déclencheur surveille afin de savoir quand démarrer une exécution de flux de travail. Vous pouvez utiliser des expressions régulières pour faire correspondre les noms de fichiers ou les chemins.

Pour obtenir des exemples, consultez [Exemples de déclencheurs](#).

10. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
11. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour ajouter un déclencheur (éditeur YAML)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.

4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez YAML.
7. Ajoutez une `Triggers` section et les propriétés sous-jacentes en utilisant l'exemple suivant comme guide. Pour en savoir plus, consultez [Triggers](#) dans la rubrique [Référence de définition du flux de travail](#).

Un déclencheur de code push peut ressembler à ceci :

```
Triggers:  
  - Type: PUSH  
    Branches:  
      - main
```

Un déclencheur de pull request peut ressembler à ceci :

```
Triggers:  
  - Type: PULLREQUEST  
    Branches:  
      - main.*  
    Events:  
      - OPEN  
      - REVISION  
      - CLOSED
```

Un déclencheur de planification peut ressembler à ceci :

```
Triggers:  
  - Type: SCHEDULE  
    Branches:  
      - main.*  
    # Run the workflow at 1:15 am (UTC+0) every Friday until the end of 2023  
    Expression: "15 1 ? * FRI 2022-2023"
```

Pour plus d'exemples d'expressions cron que vous pouvez utiliser dans la `Expression` propriété, consultez [Expression](#).

Pour plus d'exemples de déclencheurs push, pull request et schedule, consultez [Exemples de déclencheurs](#).

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Exemples de déclencheurs

Les exemples suivants montrent comment ajouter différents types de déclencheurs dans le fichier de définition du flux de travail.

### Rubriques

- [Exemple : un simple déclencheur push à code](#)
- [Exemple : un simple déclencheur « push to main »](#)
- [Exemple : un simple déclencheur de pull request](#)
- [Exemple : un simple déclencheur de planification](#)
- [Exemple : un déclencheur avec un calendrier et des branches](#)
- [Exemple : un déclencheur avec un calendrier, un push et des branches](#)
- [Exemple : une gâchette avec une traction et des branches](#)
- [Exemple : un déclencheur avec un pull, des branches et un événement « CLOSED »](#)
- [Exemple : un déclencheur avec un push, des branches et des fichiers](#)

### Exemple : un simple déclencheur push à code

L'exemple suivant montre un déclencheur qui lance l'exécution d'un flux de travail chaque fois que du code est transféré vers une branche de votre référentiel source.

Lorsque ce déclencheur est activé, CodeCatalyst lance un flux de travail exécuté à l'aide des fichiers de la branche vers laquelle vous pointez (c'est-à-dire la branche de destination).

Par exemple, si vous envoyez un commit à `main`, CodeCatalyst lance un flux de travail exécuté à l'aide du fichier de définition du flux de travail et d'autres fichiers source. `main`

Autre exemple, si vous envoyez un commit à `feature-branch-123`, CodeCatalyst lance un flux de travail exécuté à l'aide du fichier de définition du flux de travail et d'autres fichiers source. `feature-branch-123`

**Triggers:**

- Type: PUSH

**Note**

Si vous souhaitez qu'un flux de travail ne démarre que lorsque vous appuyez dessus `main`, consultez [Exemple : un simple déclencheur « push to main »](#).

**Exemple : un simple déclencheur « push to main »**

L'exemple suivant montre un déclencheur qui lance l'exécution d'un flux de travail chaque fois que du code est transmis à la `main` branche (et uniquement à la `main` branche) de votre référentiel source.

**Triggers:**

- Type: PUSH

**Branches:**

- `main`

**Exemple : un simple déclencheur de pull request**

L'exemple suivant montre un déclencheur qui lance l'exécution d'un flux de travail chaque fois qu'une pull request est créée ou révisée dans votre référentiel source.

Lorsque ce déclencheur est activé, CodeCatalyst lance un flux de travail exécuté à l'aide du fichier de définition du flux de travail et des autres fichiers source de la branche d'où vous effectuez l'extraction (c'est-à-dire la branche source).

Par exemple, si vous créez une pull request avec une branche source appelée `feature-123` et une branche de destination appelée `main`, CodeCatalyst lance un flux de travail exécuté à l'aide du fichier de définition du flux de travail et d'autres fichiers source. `feature-123`

**Triggers:**

- Type: PULLREQUEST

**Events:**

- OPEN
- REVISION

## Exemple : un simple déclencheur de planification

L'exemple suivant montre un déclencheur qui lance un flux de travail exécuté à minuit (UTC+0) du lundi au vendredi.

Lorsque ce déclencheur est activé, CodeCatalyst lance une seule exécution de flux de travail pour chaque branche de votre référentiel source contenant un fichier de définition de flux de travail avec ce déclencheur.

Par exemple, si vous avez trois branches dans votre référentiel source, `main`, `release-v1feature-123`, et que chacune de ces branches contient un fichier de définition de flux de travail avec le déclencheur suivant, CodeCatalyst lance trois exécutions de flux de travail : une utilisant les fichiers dans `main`, une autre utilisant les fichiers dans `release-v1` et une autre utilisant les fichiers dans `feature-123`.

### Triggers:

- Type: SCHEDULE
- Expression: "`0 0 ? * MON-FRI *`"

Pour plus d'exemples d'expressions cron que vous pouvez utiliser dans la Expression propriété, consultez [Expression](#).

## Exemple : un déclencheur avec un calendrier et des branches

L'exemple suivant montre un déclencheur qui lance un flux de travail exécuté à 18h15 (UTC+0) tous les jours.

Lorsque ce déclencheur est activé, CodeCatalyst lance une exécution de flux de travail à l'aide des fichiers de la `main` branche et lance des exécutions supplémentaires pour chaque branche commençant par `release-`.

Par exemple, si vous avez des branches nommées `main`, `release-v1bugfix-1`, `bugfix-2` dans votre référentiel source, CodeCatalyst démarre deux exécutions de flux de travail : l'une utilisant les fichiers dans `main`, l'autre utilisant les fichiers dans `release-v1`. Il ne lance pas les exécutions de flux de travail pour les `bugfix-1` branches `bugfix-1` et.

### Triggers:

- Type: SCHEDULE
- Expression: "`15 18 * * ? *`"
- Branches:
  - `main`

```
- release\-.*
```

Pour plus d'exemples d'expressions cron que vous pouvez utiliser dans la Expression propriété, consultez [Expression](#).

Exemple : un déclencheur avec un calendrier, un push et des branches

L'exemple suivant montre un déclencheur qui lance un flux de travail exécuté à minuit (UTC+0) tous les jours, et chaque fois que le code est envoyé à la branche. main

Dans cet exemple :

- L'exécution d'un flux de travail commence tous les jours à minuit. Le flux de travail exécuté utilise le fichier de définition du flux de travail et les autres fichiers source de la main branche.
- Une exécution de flux de travail démarre également chaque fois que vous envoyez un commit à la main branche. Le flux de travail exécuté utilise le fichier de définition du flux de travail et les autres fichiers source de la branche de destination (main).

Triggers:

```
- Type: SCHEDULE
  Expression: "0 0 * * ? *"
  Branches:
    - main
- Type: PUSH
  Branches:
    - main
```

Pour plus d'exemples d'expressions cron que vous pouvez utiliser dans la Expression propriété, consultez [Expression](#).

Exemple : une gâchette avec une traction et des branches

L'exemple suivant montre un déclencheur qui lance l'exécution d'un flux de travail chaque fois que quelqu'un ouvre ou modifie une pull request avec une branche de destination appelé main. Bien que la branche spécifiée dans la Triggers configuration le soit main, le flux de travail exécuté utilisera le fichier de définition du flux de travail et les autres fichiers source de la branche source (qui est la branche dont vous extrayez).

Triggers:

```
- Type: PULLREQUEST
```



```
Branches:  
- main  
Events:  
- OPEN  
- REVISION
```

Exemple : un déclencheur avec un pull, des branches et un événement « CLOSED »

L'exemple suivant montre un déclencheur qui lance l'exécution d'un flux de travail chaque fois qu'une pull request est fermée sur une branche commençant par `main`.

Dans cet exemple :

- Lorsque vous fermez une pull request dont la branche de destination commence par `main`, l'exécution du flux de travail démarre automatiquement en utilisant le fichier de définition du flux de travail et les autres fichiers source de la branche source (désormais fermée).
- Si vous avez configuré votre dépôt source pour supprimer automatiquement les branches après la fusion d'une pull request, ces branches n'auront jamais la possibilité d'entrer dans l'`CLOSED` état. Cela signifie que les branches fusionnées n'activeront pas le `CLOSED` déclencheur de pull request. Dans ce scénario, le seul moyen d'activer le `CLOSED` déclencheur est de fermer la pull request sans la fusionner.

```
Triggers:  
- Type: PULLREQUEST  
Branches:  
- main.*  
Events:  
- CLOSED
```

Exemple : un déclencheur avec un push, des branches et des fichiers

L'exemple suivant montre un déclencheur qui lance l'exécution d'un flux de travail chaque fois qu'une modification est apportée au `filename.txt` fichier, ou à tout autre fichier du `src` répertoire, sur la `main` branche.

Lorsque ce déclencheur est activé, CodeCatalyst lance un flux de travail exécuté à l'aide du fichier de définition du flux de travail et des autres fichiers source de la `main` branche.

```
Triggers:  
- Type: PUSH
```

```
Branches:  
- main  
FilesChanged:  
- filename.txt  
- src\*.*
```

## Utilisation des variables

Une variable est une paire clé-valeur qui contient des informations auxquelles vous pouvez faire référence dans votre CodeCatalyst flux de travail.

Il existe deux types de variables que vous pouvez utiliser dans un flux de travail :

- Variables définies par l'utilisateur : il s'agit de paires clé-valeur que vous définissez.
- Variables prédéfinies : il s'agit de paires clé-valeur émises automatiquement par un flux de travail. Il n'est pas nécessaire que vous les définissiez.

### Note

CodeCatalyst prend également en charge [les paramètres de GitHub sortie](#), qui se comportent comme des variables et peuvent être référencés dans d'autres actions. Pour plus d'informations, consultez [Utilisation des paramètres de sortie des GitHub actions](#).

## Rubriques

- [Utilisation de variables définies par l'utilisateur](#)
- [Utilisation de variables prédéfinies](#)

## Utilisation de variables définies par l'utilisateur

Les variables définies par l'utilisateur sont des paires clé-valeur que vous définissez. Il en existe deux types :

- Variables en texte brut, ou simplement variables : il s'agit de paires clé-valeur que vous définissez en texte clair dans le fichier de définition du flux de travail.
- Secrets : il s'agit de paires clé-valeur que vous définissez sur une page Secrets distincte de la console Amazon CodeCatalyst . La clé (nom) est une étiquette publique et la valeur contient les

informations que vous souhaitez garder privées. Vous spécifiez uniquement la clé dans le fichier de définition du flux de travail. Utilisez des secrets à la place des mots de passe et autres informations sensibles dans le fichier de définition du flux de travail.

### Note

Par souci de concision, ce guide utilise le terme `variable` pour désigner une variable en texte brut.

## Rubriques

- [Définition d'une variable](#)
- [Définition d'un secret](#)
- [Exporter une variable afin que d'autres actions puissent l'utiliser](#)
- [Référencer une variable dans l'action qui la définit](#)
- [Référencer une variable produite par une autre action](#)
- [Référencer un secret](#)
- [Exemples](#)

## Définition d'une variable

Vous pouvez définir des variables de deux manières :

- Dans la `Inputs` section d'une action de flux de travail, voir [Pour définir une variable dans la section « Entrées »](#)
- Dans la `Steps` section d'une action de flux de travail, voir [Pour définir une variable dans la section « Étapes »](#)

### Note

La `Steps` méthode ne fonctionne qu'avec les actions de CodeCatalyst construction, de test et d'GitHub actions, car ce sont les seules actions qui incluent une `Steps` section.

Pour obtenir des exemples, consultez [Exemples](#).

## Visual

Pour définir une variable dans la section « Entrées » (éditeur visuel)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez Visual.
7. Dans le diagramme du flux de travail, choisissez l'action pour laquelle vous souhaitez définir la variable.
8. Choisissez Entrées.
9. Dans Variables - facultatif, choisissez Ajouter une variable, puis procédez comme suit :

Spécifiez une séquence de paires nom/valeur qui définissent les variables d'entrée que vous souhaitez mettre à la disposition de l'action. Les noms de variables sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms de variables.

Pour plus d'informations sur les variables, y compris des exemples, consultez [Utilisation des variables](#).

10. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
11. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour définir une variable dans la section « Entrées » (éditeur YAML)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.

5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. Dans une action de flux de travail, ajoutez un code similaire au suivant :

```
action-name:
  Inputs:
    Variables:
      - Name: variable-name
        Value: variable-value
```

Pour obtenir plus d'exemples, consultez [Exemples](#). Pour plus d'informations, consultez le [Référence de définition du flux de travail](#) correspondant à votre action.

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Visual

Pour définir une variable dans la section « Étapes » (éditeur visuel)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez Visual.
7. Dans le diagramme du flux de travail, choisissez l'action pour laquelle vous souhaitez définir la variable.
8. Choisissez Configuration.
9. Dans les commandes Shell ou GitHubActions YAML, selon les options disponibles, définissez une variable dans les actionsSteps, de manière explicite ou implicite.
  - Pour définir la variable de manière explicite, incluez-la dans une commande bash directement dans la Steps section.

- Pour définir implicitement une variable, spécifiez-la dans un fichier référencé dans la Steps section de l'action.

Pour obtenir des exemples, consultez [Exemples](#). Pour plus d'informations, reportez-vous à la section [Référence de définition du flux de travail](#) dédiée à l'action.

10. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
11. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour définir une variable dans la section « Étapes » (éditeur YAML)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. Dans une action de flux de travail, définissez une variable dans la Steps section de l'action, de manière explicite ou implicite.
  - Pour définir la variable de manière explicite, incluez-la dans une commande bash directement dans la Steps section.
  - Pour définir implicitement une variable, spécifiez-la dans un fichier référencé dans la Steps section de l'action.

Pour obtenir des exemples, consultez [Exemples](#). Pour plus d'informations, reportez-vous à la section [Référence de définition du flux de travail](#) dédiée à l'action.

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Définition d'un secret

Vous définissez un secret sur la page Secrets de la CodeCatalyst console. Pour en savoir plus, consultez [Travailler avec des secrets](#).

Par exemple, vous pouvez définir un secret qui ressemble à ceci :

- Nom (clé) : **my-password**
- Valeur : **^\*H3#!b9**

Une fois le secret défini, vous pouvez spécifier la clé du secret (**my-password**) dans le fichier de définition du flux de travail. Pour obtenir un exemple montrant la façon de procéder, consultez [Exemple : référence à un secret](#).

Exporter une variable afin que d'autres actions puissent l'utiliser

Suivez les instructions ci-dessous pour exporter une variable depuis une action afin de pouvoir la référencer dans d'autres actions.

Avant d'exporter une variable, notez les points suivants :

- Si vous devez uniquement référencer la variable dans l'action où elle est définie, vous n'avez pas besoin de l'exporter.
- Toutes les actions ne prennent pas en charge l'exportation de variables. Pour déterminer si votre action prend en charge cette fonctionnalité, parcourez les instructions de l'éditeur visuel qui suivent et vérifiez si l'action inclut un bouton Variables dans l'onglet Sorties. Dans l'affirmative, l'exportation de variables est prise en charge.
- Pour exporter une variable depuis une GitHub action, consultez [Exporter un paramètre GitHub de sortie afin que d'autres actions puissent l'utiliser](#).

## Prérequis

Assurez-vous d'avoir défini la variable que vous souhaitez exporter. Pour en savoir plus, consultez [Définition d'une variable](#).

## Visual

Pour exporter une variable (éditeur visuel)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.

4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez Visual.
7. Dans le diagramme du flux de travail, choisissez l'action à partir de laquelle vous souhaitez exporter la variable.
8. Choisissez Sorties.
9. Dans Variables - facultatif, choisissez Ajouter une variable, puis procédez comme suit :  
  
Spécifiez le nom de la variable que vous souhaitez que l'action exporte. Cette variable doit déjà être définie dans la Steps section Inputs ou de la même action.
10. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
11. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## YAML

Pour exporter une variable (éditeur YAML)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. Dans l'action à partir de laquelle vous souhaitez exporter la variable, ajoutez un code similaire au suivant :

```
action-name:  
  Outputs:  
    Variables:  
      - Name: variable-name
```

Pour obtenir plus d'exemples, consultez [Exemples](#).

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.



9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Référencer une variable dans l'action qui la définit

Suivez les instructions ci-dessous pour référencer une variable dans l'action qui la définit.

### Note

Pour référencer une variable générée par une GitHub action, consultez [Référencement d'un paramètre GitHub de sortie](#).

## Prérequis

Assurez-vous d'avoir défini la variable à laquelle vous souhaitez faire référence. Pour en savoir plus, consultez [Définition d'une variable](#).

## Visual

Non disponible Choisissez YAML pour afficher les instructions YAML.

## YAML

Pour référencer une variable dans l'action qui la définit

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. Dans l' CodeCatalyst action qui définit la variable à laquelle vous souhaitez faire référence, ajoutez-la en utilisant la syntaxe bash suivante :

```
$variable-name
```

Par exemple :

```
MyAction:
  Configuration:
    Steps:
      - Run: $variable-name
```

Pour obtenir plus d'exemples, consultez [Exemples](#). Pour plus d'informations, consultez les informations de référence relatives à votre action dans le [Référence de définition du flux de travail](#).

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Référencer une variable produite par une autre action

Utilisez les instructions suivantes pour référencer les variables produites par d'autres actions.

### Note

Pour référencer la sortie d'une variable à partir GitHub d'une action, consultez [Référencement d'un paramètre GitHub de sortie](#).

## Prérequis

Assurez-vous d'avoir exporté la variable à laquelle vous souhaitez faire référence. Pour en savoir plus, consultez [Exporter une variable afin que d'autres actions puissent l'utiliser](#).

## Visual

Non disponible Choisissez YAML pour afficher les instructions YAML.

## YAML

Pour référencer une variable produite par une autre action (éditeur YAML)


1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.

4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. Dans l' CodeCatalyst action, ajoutez une référence à la variable en utilisant la syntaxe suivante :

```
${action-group-name.action-name.variable-name}
```

Remplacer :

- *action-group-name* avec le nom du groupe d'actions qui contient l'action qui produit une variable.

 Note

Vous pouvez omettre *action-group-name* s'il n'existe aucun groupe d'actions ou si la variable est produite par une action du même groupe d'actions.

- *action-name* avec le nom de l'action qui génère la variable.
- *variable-name* avec le nom de la variable.

Par exemple :

```
MySecondAction:
  Configuration:
    Steps:
      - Run: ${MyFirstAction.TIMESTAMP}
```

Pour obtenir plus d'exemples, consultez [Exemples](#). Pour plus d'informations, consultez le [Référence de définition du flux de travail](#) correspondant à votre action.

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

## Référencer un secret

Pour obtenir des instructions sur le référencement d'un secret dans le fichier de définition du flux de travail, reportez-vous [Utiliser un secret](#) à.

Pour voir un exemple, consultez [Exemple : référence à un secret](#).

### Exemples

Les exemples suivants montrent comment définir et référencer des variables dans le fichier de définition du flux de travail.

### Exemples

- [Exemple : définition d'une variable à l'aide de la propriété Inputs](#)
- [Exemple : définition d'une variable à l'aide de la propriété Steps](#)
- [Exemple : exportation d'une variable à l'aide de la propriété Outputs](#)
- [Exemple : Référencement d'une variable définie dans la même action](#)
- [Exemple : Référencement d'une variable définie dans une autre action](#)
- [Exemple : référence à un secret](#)

### Exemple : définition d'une variable à l'aide de la propriété Inputs

L'exemple suivant montre comment définir deux variables, VAR1 et VAR2, dans une Inputs section.

```
Actions:
  Build:
    Identifiant: aws/build@v1
    Inputs:
      Variables:
        - Name: VAR1
          Value: "My variable 1"
        - Name: VAR2
          Value: "My variable 2"
```

### Exemple : définition d'une variable à l'aide de la propriété Steps

L'exemple suivant montre comment définir explicitement une DATE variable dans la Steps section.

```
Actions:
```

```
Build:
  Identifier: aws/build@v1
  Configuration:
    Steps:
      - Run: DATE=$(date +%m-%d-%y)
```

Exemple : exportation d'une variable à l'aide de la propriété Outputs

L'exemple suivant vous montre comment définir deux variables `TIMESTAMP`, `REPOSITORY-URI` et comment les exporter à l'aide de la `Outputs` section.

```
Actions:
  Build:
    Identifier: aws/build@v1
    Inputs:
      Variables:
        - Name: REPOSITORY-URI
          Value: 111122223333.dkr.ecr.us-east-2.amazonaws.com/codecatalyst-ecs-image-repo
    Configuration:
      Steps:
        - Run: TIMESTAMP=$(date +%m-%d-%y-%H-%m-%s)
    Outputs:
      Variables:
        - REPOSITORY-URI
        - TIMESTAMP
```

Exemple : Référencement d'une variable définie dans la même action

L'exemple suivant montre comment spécifier une `VAR1` variable dans `MyBuildAction`, puis la référencer dans la même action en utilisant `$VAR1`.

```
Actions:
  MyBuildAction:
    Identifier: aws/build@v1
    Inputs:
      Variables:
        - Name: VAR1
          Value: my-value
    Configuration:
      Steps:
        - Run: $VAR1
```

## Exemple : Référencement d'une variable définie dans une autre action

L'exemple suivant montre comment spécifier une `TIMESTAMP` variable dans `BuildActionA`, l'exporter à l'aide de la `Outputs` propriété, puis la référencer dans `BuildActionB` using `${BuildActionA.TIMESTAMP}`.

```
Actions:
  BuildActionA:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        - Run: TIMESTAMP=$(date +%m-%d-%y-%H-%m-%s)
    Outputs:
      Variables:
        - TIMESTAMP
  BuildActionB:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        - Run: docker build -t my-ecr-repo/image-repo:latest .
        - Run: docker tag my-ecr-repo/image-repo:${BuildActionA.TIMESTAMP}

# Specifying just '$TIMESTAMP' here will not work
# because TIMESTAMP is not a variable
# in the BuildActionB action.
```

## Exemple : référence à un secret

L'exemple suivant montre comment faire référence à un `my-password` secret. `my-password` est la clé du secret. La clé de ce secret et la valeur de mot de passe correspondante doivent être spécifiées sur la page `Secrets` de la CodeCatalyst console avant d'être utilisés dans le fichier de définition du flux de travail. Pour plus d'informations, consultez [Travailler avec des secrets](#).

```
Actions:
  BuildActionA:
    Identifier: aws/build@v1
    Configuration:
      Steps:
        - Run: curl -u LiJuan:${Secrets.my-password} https://example.com
```

## Utilisation de variables prédéfinies

Les variables prédéfinies sont des paires clé-valeur émises automatiquement par un flux de travail et mises à votre disposition pour que vous puissiez les utiliser dans des actions de flux de travail.

Vous pouvez utiliser des variables prédéfinies dans n'importe quelle action du flux de travail.

### Rubriques

- [Liste des variables prédéfinies](#)
- [Référencement d'une variable prédéfinie](#)
- [Déterminer les variables prédéfinies émises par votre flux de travail](#)
- [Exemples](#)

### Liste des variables prédéfinies

Consultez les sections suivantes pour visualiser les variables prédéfinies produites automatiquement par CodeCatalyst.

#### Note

Cette liste inclut uniquement les variables prédéfinies émises par les CodeCatalyst flux de travail et [CodeCatalyst les actions](#). Si vous utilisez d'autres types d'actions, telles que GitHub les actions Actions ou CodeCatalyst Labs, consultez plutôt [Déterminer les variables prédéfinies émises par votre flux de travail](#).

### Liste

- [variables «BranchName" et CommitId "»](#)
- [Créez et testez des variables d'action](#)
- [Variables d'action « Amazon S3 publish »](#)
- [variables d'action « AWS CDK bootstrap »](#)
- [variables d'action « AWS CDK déployer »](#)
- [variables d'action « AWS Lambda invoquer »](#)
- [Variables d'action « Déployer la AWS CloudFormation pile »](#)
- [Variables d'action « Déployer sur Amazon ECS »](#)
- [Variables d'action « Déployer vers un cluster Kubernetes »](#)

- [Variables d'action « Afficher la définition de la tâche Amazon ECS »](#)
- [variables GitHub d'action « Actions »](#)

variables «BranchName" et CommitId "»

CodeCatalyst produit des variables « BranchName » et « CommitId » lors de l'exécution de votre flux de travail. Consultez le tableau suivant pour plus d'informations sur ces variables.

| Clé        | Valeur   |
|------------|--|
| CommitId   | <p>L'ID de validation représentant l'état du référentiel au moment du début de l'exécution du flux de travail.</p> <p>Exemple : <code>example3819261db00a3ab59468c8b</code></p> <p>Voir aussi : <a href="#">Exemple : Référencement de la variable prédéfinie CommitId « »</a></p> |
| BranchName | <p>Nom de la branche sur laquelle l'exécution du flux de travail a débuté.</p> <p>Exemples : <code>main</code>, <code>feature/branch</code> , <code>test-LiJuan</code></p> <p>Voir aussi : <a href="#">Exemple : Référencement de la variable prédéfinie BranchName « »</a></p>    |

Créez et testez des variables d'action

Les actions de construction et de test ne produisent pas de variables automatiquement.

Variables d'action « Amazon S3 publish »

L'action de publication Amazon S3 ne produit pas de variables automatiquement.

variables d'action « AWS CDK bootstrap »

L'action AWS CDKbootstrap produit les variables prédéfinies suivantes.



| Clé                        | Valeur  |
|----------------------------|---|
| plate-forme de déploiement | Nom de la plateforme de déploiement.<br><br>Codé en dur pour. <code>AWS:CloudFormation</code>   |
| Région                     | Le code de région dans Région AWS le quel la pile AWS CDK bootstrap a été déployée lors de l'exécution du flux de travail.<br><br>Exemple : <code>us-west-2</code>  |
| identifiant de pile        | Le nom de ressource Amazon (ARN) de la pile AWS CDK bootstrap déployée.<br><br>Exemple : <code>arn:aws:cloudformation:us-west-2:111122223333:stack/codecatalyst-cdk-bootstrap-stack/6aad4380-100a-11ec-a10a-03b8a84d40df</code>   |
| SAUTER LE DÉPLOIEMENT      | La valeur de <code>true</code> indique que le déploiement de votre stack AWS CDK bootstrap a été ignoré lors de l'exécution du flux de travail. Un déploiement de pile sera ignoré s'il n'y a aucun changement dans la pile depuis le dernier déploiement.<br><br>Cette variable n'est produite que si sa valeur est <code>true</code> .<br><br>Codé en dur pour. <code>true</code> |

variables d'action « AWS CDK déployer »

L'action de AWS CDKdéploiement produit les variables suivantes.

| Clé                         | Valeur   |
|-----------------------------|--|
| identifiant de pile         | <p>Le nom de ressource Amazon (ARN) de la pile d'AWS CDKapplications sur laquelle a été déployée lors de l'exécution du flux de travail.</p> <p>Exemple : <code>arn:aws:cloudformation:us-west-2:111122223333:stack/codecatalyst-cdk-app-stack/6aad4380-100a-11ec-a10a-03b8a84d40df</code></p>   |
| plate-forme de déploiement  | <p>Nom de la plateforme de déploiement.</p> <p>Codé en dur pour. <code>AWS:CloudFormation</code></p>   |
| Région                      | <p>Le code de région du Région AWS qui a été déployé lors de l'exécution du flux de travail.</p> <p>Exemple : <code>us-west-2</code></p>   |
| SAUTER LE DÉPLOIEMENT       | <p>La valeur de <code>true</code> indique que le déploiement de votre pile d'AWS CDKapplications a été ignoré lors de l'exécution du flux de travail. Un déploiement de pile sera ignoré s'il n'y a aucun changement dans la pile depuis le dernier déploiement.</p> <p>Cette variable n'est produite que si sa valeur est <code>true</code>.</p> <p>Codé en dur pour. <code>true</code></p> |
| AWS CloudFormationvariables | <p>En plus de générer les variables répertoriées précédemment, l'action de AWS CDKdéploiement expose également les variables de CloudFormationsortie en tant que variables de flux de travail à utiliser dans les actions de flux de travail suivantes. Par défaut, l'action</p>   |

| Clé | Valeur   |
|-----|--|
|     | n'expose que les quatre premières CloudFormation variables (ou moins) détectées. Pour déterminer ceux qui sont exposés, exécutez l'action de AWS CDKdéploiement une seule fois, puis consultez l'onglet Variables de la page des détails de l'exécution. Si les variables répertoriées dans l'onglet Variables ne correspondent pas à vos attentes, vous pouvez en configurer d'autres à l'aide de la propriété <code>CfnOutputVariables</code> YAML. Pour plus d'informations, consultez la description de la <a href="#">CfnOutputVariables</a> propriété dans <a href="#">l'référence d'action « AWS CDK déployer »</a> . |

variables d'action « AWS Lambda invoquer »

Par défaut, l'action d'AWS Lambdaappel produit une variable par clé de niveau supérieur dans la charge utile de la réponse Lambda.


Par exemple, si la charge utile de la réponse ressemble à ceci :

```
responsePayload = {
  "name": "Saanvi",
  "location": "Seattle",
  "department": {
    "company": "Amazon",
    "team": "AWS"
  }
}
```

... alors l'action générerait les variables suivantes.

| Clé      | Valeur  |
|----------|---------|
| name     | Saanvi  |
| location | Seattle |

| Clé        | Valeur  |
|------------|---|
| department | {"entreprise" : « Amazon », « équipe » : "AWS"} |

 Note

Vous pouvez modifier les variables générées à l'aide de la propriété `ResponseFilters` YAML. Pour en savoir plus, consultez [ResponseFilters](#) dans la rubrique [référence d'action « AWS Lambda invoquer »](#).

Variables d'action « Déployer la AWS CloudFormation pile »

L'action Deploy AWS CloudFormation stack produit les variables suivantes.

| Clé                        | Valeur  |
|----------------------------|---|
| plate-forme de déploiement | Nom de la plateforme de déploiement.<br><br>Codé en dur pour. <code>AWS:CloudFormation</code>   |
| Région                     | Le code de région du Région AWS qui a été déployé lors de l'exécution du flux de travail.<br><br>Exemple : <code>us-west-2</code>   |
| identifiant de pile        | Le nom de ressource Amazon (ARN) de la pile déployée.<br><br>Exemple : <code>arn:aws:cloudformation:us-west-2:111122223333:stack/codecatalyst-cfn-stack/6aad4380-100a-11ec-a10a-03b8a84d40df</code> |

## Variables d'action « Déployer sur Amazon ECS »

L'action Deploy to Amazon ECS produit les variables suivantes.

| Clé                        | Valeur   |
|----------------------------|--|
| cluster                    | <p>Nom du cluster Amazon ECS sur lequel le flux de travail a été déployé pendant l'exécution du flux de travail.</p> <p>Exemple : <code>codecatalyst-ecs-cluster</code></p>  |
| plate-forme de déploiement | <p>Nom de la plateforme de déploiement.</p> <p>Codé en dur pour. AWS : ECS</p>   |
| web                        | <p>Nom du service Amazon ECS qui a été déployé lors de l'exécution du flux de travail.</p> <p>Exemple : <code>codecatalyst-ecs-service</code></p>  |
| task-definition-arn        | <p>Le nom de ressource Amazon (ARN) de la définition de tâche enregistrée lors de l'exécution du flux de travail.</p> <p>Exemple : <code>arn:aws:ecs:us-west-2:111122223333:task-definition/codecatalyst-task-def:8</code></p> <p>L'8 exemple précédent indique la révision qui a été enregistrée.</p> |
| URL de déploiement         | <p>Lien vers l'onglet Events de la console Amazon ECS, où vous pouvez consulter les détails du déploiement Amazon ECS associé au flux de travail exécuté.</p> <p>Exemple : <code>https://console.aws.amazon.com/ecs/home?region=us-west-2#/clusters/codec</code></p>                                   |

| Clé    | Valeur  |
|--------|---|
|        | <code>atalyst-ecs-cluster/services/codecatalyst-ecs-service/events</code>   |
| Région | Le code de région du Région AWS qui a été déployé lors de l'exécution du flux de travail.<br><br>Exemple : <code>us-west-2</code> |

### Variables d'action « Déployer vers un cluster Kubernetes »

L'action Déployer vers le cluster Kubernetes produit les variables suivantes.

| Clé                        | Valeur  |
|----------------------------|---|
| cluster                    | Le nom de ressource Amazon.com (ARN) du cluster Amazon EKS sur lequel a été déployé lors de l'exécution du flux de travail.<br><br>Exemple : <code>arn:aws:eks:us-west-2:111122223333:cluster/codecatalyst-eks-cluster</code> |
| plate-forme de déploiement | Nom de la plateforme de déploiement.<br><br>Codé en dur pour. <code>AWS:EKS</code>  |
| métadonnées                | Instances réservées. Métadonnées au format JSON relatives au cluster déployé lors de l'exécution du flux de travail.  |
| espace de nom              | L'espace de noms Kubernetes dans lequel le cluster a été déployé.<br><br>Exemple : <code>default</code>   |

| Clé       | Valeur  |
|-----------|---|
| resources | Instances réservées. Métadonnées au format JSON relatives aux ressources déployées lors de l'exécution du flux de travail.  |
| serveur   | <p>Nom du point de terminaison du serveur d'API que vous pouvez utiliser pour communiquer avec votre cluster à l'aide d'outils de gestion tels que <code>kubectl</code>.</p> <p>Pour plus d'informations sur le point de terminaison du service API, consultez la section <a href="#">Contrôle d'accès des points de terminaison du cluster Amazon EKS</a> dans le guide de l'utilisateur Amazon EKS.</p> <p>Exemple : <code>https://random-string.gr7.us-west-2.eks.amazonaws.com</code></p> |

Variables d'action « Afficher la définition de la tâche Amazon ECS »

L'action de définition de tâche Render Amazon ECS produit les variables suivantes.

| Clé                   | Valeur   |
|-----------------------|--|
| définition des tâches | <p>Nom donné au fichier de définition de tâche mis à jour par l'action de définition de tâche Render Amazon ECS. Le nom utilise le format suivant : <code>task-definition-random-string.json</code>.</p> <p>Exemple : <code>task-definition--259-0a2r7gx1TF5Xr.json</code></p> |

## variables GitHub d'action « Actions »

L'action « GitHub Actions » ne produit pas de variables automatiquement.

### Référencement d'une variable prédéfinie

Suivez les instructions ci-dessous pour référencer une variable prédéfinie.

#### Prérequis

Déterminez le nom de la variable prédéfinie à laquelle vous souhaitez faire référence, par exemple `CommitId`. Pour en savoir plus, consultez [Déterminer les variables prédéfinies émises par votre flux de travail](#).

#### Visual

Non disponible Choisissez YAML pour afficher les instructions YAML.

#### YAML

Pour référencer une variable prédéfinie (éditeur YAML)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Edit (Modifier).
6. Choisissez YAML.
7. Dans une CodeCatalyst action, ajoutez la référence de variable prédéfinie en utilisant la syntaxe suivante :

```
${action-group-name.action-name-or-WorkflowSource.variable-name}
```

Remplacer :

- *action-group-name* avec le nom du groupe d'action.



**Note**

Vous pouvez omettre *action-group-names* s'il n'existe aucun groupe d'actions ou si la variable est produite par une action du même groupe d'actions.

- *action-name-or- WorkflowSource* avec :

Nom de l'action qui génère la variable.

or

WorkflowSource, si la variable est la CommitId variable BranchName ou.

- *variable-name* avec le nom de la variable.

Par exemple :

```
MySecondAction:
  Configuration:
    Steps:
      - Run: echo ${MyFirstECSAction.cluster}
```

Voici un autre exemple :

```
MySecondAction:
  Configuration:
    Steps:
      - Run: echo ${WorkflowSource.CommitId}
```

Pour obtenir plus d'exemples, consultez [Exemples](#). Pour plus d'informations, consultez le [Référence de définition du flux de travail](#) correspondant à votre action.

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

Déterminer les variables prédéfinies émises par votre flux de travail

Vous pouvez déterminer les variables prédéfinies émises par votre flux de travail de deux manières :

- Exécutez le flux de travail une seule fois. Une fois l'exécution terminée, les variables émises par le flux de travail sont affichées dans l'onglet Variables de la page des détails de l'exécution. Pour en savoir plus, consultez [Afficher le statut et les détails de l'exécution du flux de travail](#).
- Consultez le [Liste des variables prédéfinies](#). Cette référence répertorie le nom (clé) et la valeur de chaque variable prédéfinie.

### Note

La taille totale maximale des variables d'un flux de travail est répertoriée dans [Quotas pour les flux de travail dans CodeCatalyst](#). Si la taille totale dépasse le maximum, l'action qui se produit une fois le maximum atteint peut échouer.

## Exemples

Les exemples suivants montrent comment référencer des variables prédéfinies dans le fichier de définition du flux de travail.

### Exemples

- [Exemple : Référencement de la variable prédéfinie CommitId « »](#)
- [Exemple : Référencement de la variable prédéfinie BranchName « »](#)

Exemple : Référencement de la variable prédéfinie CommitId « »

L'exemple suivant montre comment faire référence à la variable CommitId prédéfinie dans l'MyBuildAction. La CommitId variable est sortie automatiquement par CodeCatalyst.

Bien que l'exemple montre la variable utilisée dans l'action de génération, vous pouvez l'utiliser CommitId dans n'importe quelle action.

```
MyBuildAction:
  Identifieur: aws/build@v1
  Inputs:
    Sources:
      - WorkflowSource
  Configuration:
    Steps:
      #Build Docker image and tag it with a commit ID
```

- Run: `docker build -t image-repo/my-docker-image:latest .`
- Run: `docker tag image-repo/my-docker-image:${WorkflowSource.CommitId}`

Exemple : Référencement de la variable prédéfinie BranchName « »

L'exemple suivant montre comment faire référence à la variable BranchName prédéfinie dans l'CDKDeploy action. La BranchName variable est sortie automatiquement par CodeCatalyst.

Bien que l'exemple montre la variable utilisée dans l'action de AWS CDKdéploiement, vous pouvez l'utiliser BranchName dans n'importe quelle action.

```
CDKDeploy:
  Identifiant: aws/cdk-deploy@v1
  Inputs:
    Sources:
      - WorkflowSource
  Configuration:
    StackName: app-stack-${WorkflowSource.BranchName}
```

## Référence de définition du flux de travail

La documentation de référence relative au fichier de définition du flux de travail est la suivante.

Un fichier de définition de flux de travail est un fichier YAML qui décrit votre flux de travail. Le fichier est stocké dans un `~/codecatalyst/workflows/` dossier à la racine de votre [dépôt source](#). Le fichier peut avoir une extension `.yml` ou `.yaml`.

Pour créer et modifier le fichier de définition du flux de travail, vous pouvez utiliser un éditeur tel que vim, ou vous pouvez utiliser l'éditeur visuel ou l'éditeur YAML de la CodeCatalyst console. Pour plus d'informations, consultez [Utilisation des éditeurs visuels et YAML de la CodeCatalyst console](#).

### Note

La plupart des propriétés YAML suivantes ont des éléments d'interface utilisateur correspondants dans l'éditeur visuel. Pour rechercher un élément de l'interface utilisateur, utilisez Ctrl+F. L'élément sera répertorié avec sa propriété YAML associée.

## Rubriques

- [Exemple de fichier de définition de flux de travail](#)
- [Consignes et conventions de syntaxe](#)
- [Propriétés de haut niveau](#)
- [Créez et testez une référence d'actions](#)
- [Référence d'action « Amazon S3 publish »](#)
- [référence d'action « AWS CDK bootstrap »](#)
- [référence d'action « AWS CDK déployer »](#)
- [référence d'action « AWS Lambda invoquer »](#)
- [Référence d'action « Deploy AWS CloudFormation Stack »](#)
- [Référence d'action « Déployer sur Amazon ECS »](#)
- [Référence d'action « Déployer vers un cluster Kubernetes »](#)
- [Référence GitHub d'action « Actions »](#)
- [Référence d'action « Render la définition de la tâche Amazon ECS »](#)

## Exemple de fichier de définition de flux de travail

Voici un exemple de fichier de définition de flux de travail simple. Il inclut quelques propriétés de haut niveau, une Triggers section et une Actions section comportant deux actions : Build et Test. Pour plus d'informations, consultez [À propos du fichier de définition du flux de travail](#).

```
Name: MyWorkflow
SchemaVersion: 1.0
RunMode: QUEUED
Triggers:
  - Type: PUSH
    Branches:
      - main
Actions:
  Build:
    Identifiant: aws/build@v1
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      Steps:
        - Run: docker build -t MyApp:latest .
  Test:
```

```
Identifiant: aws/managed-test@v1
DependsOn:
  - Build
Inputs:
  Sources:
    - WorkflowSource
Configuration:
  Steps:
    - Run: npm install
    - Run: npm run test
```

## Consignes et conventions de syntaxe

Cette section décrit les règles de syntaxe du fichier de définition du flux de travail, ainsi que les conventions de dénomination utilisées dans cette documentation de référence.

### Consignes relatives à la syntaxe YAML

Le fichier de définition du flux de travail est écrit en YAML et suit la [spécification YAML 1.1](#), de sorte que tout ce qui est autorisé dans cette spécification est également autorisé dans le flux de travail YAML. Si vous débutez avec le langage YAML, voici quelques instructions rapides pour vous assurer que vous fournissez un code YAML valide.

- distinction majuscules/majuscules : le fichier de définition du flux de travail distingue les majuscules et minuscules. Assurez-vous donc d'utiliser le boîtier présenté dans cette documentation.
- Caractères spéciaux : nous vous recommandons d'utiliser des guillemets ou des guillemets doubles autour des valeurs de propriété qui incluent l'un des caractères spéciaux suivants : { } [ ] , \* # ? , | , - , , < , > = , ! , % , , @ : , ` et ,

Si vous n'incluez pas les guillemets, les caractères spéciaux répertoriés précédemment risquent d'être interprétés de manière inattendue.

- Noms des propriétés : les noms des propriétés (par opposition aux valeurs des propriétés) sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux traits d'union (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets ou de guillemets doubles pour activer les caractères spéciaux et les espaces dans les noms de propriétés.

Non autorisé :

```
'My#Build@action'
```

```
My#Build@action
```

```
My Build Action
```

Autorisé :

```
My-Build-Action_1
```

- Codes d'échappement : si la valeur de votre propriété inclut des codes d'échappement (par exemple, `\n` ou `\t`), suivez les instructions suivantes :
  - Utilisez des guillemets simples pour renvoyer le code d'échappement sous forme de chaîne. Par exemple `'my string \n my string'`, renvoie la chaîne `my string \n my string`.
  - Utilisez des guillemets pour analyser le code d'échappement. Par exemple `"my string \n my new line"`, renvoie :

```
my string
my new line
```

- Commentaires : Préfacez les commentaires par `#`.

Exemple :

```
Name: MyWorkflow
# This is a comment.
SchemaVersion: 1.0
```

- Triple tiret (`---`) : Ne l'utilisez pas `---` dans votre code YAML. CodeCatalyst ignore tout ce qui se trouve après le `---`

## Convention d'appellation

Dans ce guide, nous utilisons les termes propriété et section pour désigner les principaux éléments d'un fichier de définition de flux de travail.

- Une propriété est un élément qui inclut deux points (`:`). Par exemple, dans l'extrait de code suivant, toutes les propriétés suivantes sont des propriétés `:Name`, `SchemaVersion`, `RunMode`, `TriggersType`, et `Branches`
- Une section est une propriété qui possède des sous-propriétés. Dans l'extrait de code suivant, il y a une `Triggers` section.

**Note**

Dans ce guide, les « sections » sont parfois appelées « propriétés », et vice versa, selon le contexte.

```
Name: MyWorkflow
SchemaVersion: 1.0
RunMode: QUEUED
Triggers:
  - Type: PUSH
  Branches:
    - main
```

## Propriétés de haut niveau

Vous trouverez ci-dessous la documentation de référence pour les propriétés de haut niveau du fichier de définition du flux de travail.

```
# Name
Name: workflow-name

# Schema version
SchemaVersion: 1.0

# Run mode
RunMode: QUEUED|SUPERSEDED|PARALLEL

# Compute
Compute:
...

# Triggers
Triggers:
...

# Actions
Actions:
...
```

## Name

(Obligatoire)

Nom du flux de travail. Le nom du flux de travail est affiché dans la liste des flux de travail et mentionné dans les notifications et les journaux. Le nom du flux de travail et le nom du fichier de définition du flux de travail peuvent correspondre, ou vous pouvez les nommer différemment. Les noms des flux de travail ne doivent pas nécessairement être uniques. Les noms des flux de travail sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms des flux de travail.

Interface utilisateur correspondante : éditeur visuel/propriétés du flux de travail/nom du flux de travail

## SchemaVersion

(Obligatoire)

Version du schéma de la définition du flux de travail. Actuellement, la seule valeur valide est 1.0.

Interface utilisateur correspondante : aucune

## RunMode

(Facultatif)

Comment CodeCatalyst gère plusieurs essais. Vous pouvez utiliser l'une des valeurs suivantes :

- **QUEUED**— Plusieurs exécutions sont mises en file d'attente et exécutées les unes après les autres. Vous pouvez avoir jusqu'à 50 essais dans une file d'attente.
- **SUPERSEDED**— Plusieurs exécutions sont mises en file d'attente et exécutées les unes après les autres. Une file d'attente ne peut comporter qu'une seule exécution. Ainsi, si deux séries se retrouvent ensemble dans la même file, la dernière remplace (prend le relais) de la précédente et l'exécution précédente est annulée.
- **PARALLEL**— Plusieurs exécutions ont lieu simultanément.

Si cette propriété est omise, la valeur par défaut est **QUEUED**.



Pour plus d'informations, consultez [Configuration des exécutions en file d'attente, remplacées et parallèles](#).

Interface utilisateur correspondante : éditeur visuel/propriétés du flux de travail/mode avancé/exécution

## Compute

(Facultatif)

Le moteur informatique utilisé pour exécuter les actions de votre flux de travail. Vous pouvez spécifier le calcul au niveau du flux de travail ou au niveau de l'action, mais pas les deux. Lorsqu'elle est spécifiée au niveau du flux de travail, la configuration de calcul s'applique à toutes les actions définies dans le flux de travail. Au niveau du flux de travail, vous pouvez également exécuter plusieurs actions sur la même instance. Pour plus d'informations, consultez [Partage du calcul entre les actions](#).

Pour plus d'informations sur le calcul, consultez [Utilisation des images Docker de l'environnement de calcul et d'exécution](#).

Interface utilisateur correspondante : aucune

```
Name: MyWorkflow
SchemaVersion: 1.0
...
Compute:
  Type: EC2 | Lambda
  Fleet: fleet-name
  SharedInstance: true | false
```

## Type

(Compute/Type)

(Obligatoire s'Computeil est défini)

Type de moteur de calcul. Vous pouvez utiliser l'une des valeurs suivantes :

- EC2 (éditeur visuel) ou EC2 (éditeur YAML)  
Optimisé pour la flexibilité lors des courses d'action.
- Lambda (éditeur visuel) ou Lambda (éditeur YAML)

Vitesses de démarrage des actions optimisées.

Pour plus d'informations sur les types de calcul, veuillez consulter [À propos des types de calcul](#).

Interface utilisateur correspondante : éditeur visuel/propriétés du flux de travail/avancé/type de calcul

Fleet

(Compute/Fleet)

(Facultatif)

Spécifiez la machine ou le parc qui exécutera votre flux de travail ou vos actions de flux de travail. Dans le cas des flottes à la demande, lorsqu'une action démarre, le flux de travail fournit les ressources dont il a besoin et les machines sont détruites à la fin de l'action. Exemples de flottes à la demande : `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Pour plus d'informations sur les flottes à la demande, consultez [Propriétés de flotte à la demande](#).

Avec les flottes provisionnées, vous configurez un ensemble de machines dédiées pour exécuter les actions de votre flux de travail. Ces machines restent inactives, prêtes à exécuter des actions immédiatement. Pour plus d'informations sur les flottes provisionnées, consultez [Propriétés du parc provisionné](#)

S'il Fleet est omis, la valeur par défaut est `Linux.x86-64.Large`.

Pour plus d'informations sur les flottes de calcul, consultez [À propos des flottes informatiques](#).

Interface utilisateur correspondante : éditeur visuel/propriétés du flux de travail/Advanced/ Compute fleet

SharedInstance

(Compute/SharedInstance)

(Facultatif)

Spécifiez la capacité de partage de calcul pour vos actions. Avec le partage de calcul, les actions d'un flux de travail s'exécutent sur la même instance (image de l'environnement d'exécution). Vous pouvez utiliser l'une des valeurs suivantes :

- TRUE signifie que l'image de l'environnement d'exécution est partagée entre les actions du flux de travail.
- FALSE signifie qu'une image d'environnement d'exécution distincte est démarrée et utilisée pour chaque action d'un flux de travail. Vous ne pouvez donc pas partager de ressources telles que des artefacts et des variables sans configuration supplémentaire.

Pour plus d'informations sur le partage de calcul, consultez [Partage du calcul entre les actions](#).

Interface utilisateur correspondante : aucune

## Triggers

(Facultatif)

Séquence d'un ou de plusieurs déclencheurs pour ce flux de travail. Si aucun déclencheur n'est spécifié, vous devez démarrer manuellement votre flux de travail.

Pour plus d'informations sur les déclencheurs, consultez [Utilisation de déclencheurs](#).

Interface utilisateur correspondante : éditeur visuel/diagramme de flux de travail/déclencheurs

```
Name: MyWorkflow
SchemaVersion: 1.0
...
Triggers:
- Type: PUSH
  Branches:
    - branch-name
  FilesChanged:
    - folder1/file
    - folder2/

- Type: PULLREQUEST
  Events:
    - OPEN
    - CLOSED
    - REVISION
  Branches:
    - branch-name
  FilesChanged:
    - file1.txt
```

```
- Type: SCHEDULE
# Run the workflow at 10:15 am (UTC+0) every Saturday
Expression: "15 10 ? * 7 *"
Branches:
  - branch-name
```

## Type

(Triggers/Type)

(Obligatoire s'il est défini)

Spécifiez le type de déclencheur. Vous pouvez utiliser l'une des valeurs suivantes :

- Push (éditeur visuel) ou PUSH (éditeur YAML)

Un déclencheur push lance l'exécution d'un flux de travail lorsqu'une modification est transmise à votre référentiel source. Le flux de travail exécuté utilisera les fichiers de la branche vers laquelle vous transférez (c'est-à-dire la branche de destination).

- Pull request (éditeur visuel) ou PULLREQUEST (éditeur YAML)

Un déclencheur de pull request lance un flux de travail lorsqu'une pull request est ouverte, mise à jour ou fermée dans votre référentiel source. Le flux de travail exécuté utilisera les fichiers de la branche d'où vous les extrayez (c'est-à-dire la branche source).

- Calendrier (éditeur visuel) ou SCHEDULE (éditeur YAML)

Un déclencheur de planification lance des exécutions de flux de travail selon un calendrier défini par une expression cron que vous spécifiez. Un flux de travail distinct démarrera pour chaque branche de votre référentiel source à l'aide des fichiers de la branche. (Pour limiter les branches sur lesquelles le déclencheur est activé, utilisez le champ Branches (éditeur visuel) ou la Branches propriété (éditeur YAML).)

Lorsque vous configurez un déclencheur de planification, suivez les instructions suivantes :

- N'utilisez qu'un seul déclencheur de planification par flux de travail.
- Si vous avez défini plusieurs flux de travail dans votre CodeCatalyst espace, nous vous recommandons de ne pas en planifier plus de 10 pour qu'ils démarrent simultanément.
- Assurez-vous de configurer l'expression cron du déclencheur avec un délai suffisant entre les exécutions. Pour plus d'informations, consultez [Expression](#).

Pour obtenir des exemples, consultez [Exemples de déclencheurs](#).

Interface utilisateur correspondante : éditeur visuel/diagramme de flux de travail/déclencheurs/type de déclencheur

## Events

(Triggers/Events)

(Obligatoire si le déclencheur Type est réglé sur PULLREQUEST)

Spécifiez le type d'événements de pull request qui lanceront l'exécution d'un flux de travail. Les valeurs valides sont les suivantes :

- Une pull request est créée (éditeur visuel) ou OPEN (éditeur YAML)

L'exécution du flux de travail démarre lorsqu'une pull request est créée.

- La pull request est fermée (éditeur visuel) ou CLOSED (éditeur YAML)

L'exécution du flux de travail démarre lorsqu'une pull request est fermée. Le comportement de l'CLOSED événement est délicat et il est préférable de le comprendre à l'aide d'un exemple.

Pour plus d'informations, consultez [Exemple : un déclencheur avec un pull, des branches et un événement « CLOSED »](#).

- Une nouvelle révision est apportée à la pull request (éditeur visuel) ou REVISION (éditeur YAML)

L'exécution du flux de travail démarre lorsqu'une révision d'une pull request est créée. La première révision est créée lors de la création de la pull request. Ensuite, une nouvelle révision est créée chaque fois que quelqu'un envoie un nouveau commit vers la branche source spécifiée dans la pull request. Si vous incluez l'REVISION événement dans votre déclencheur de pull request, vous pouvez l'OPEN mettre, puisqu'il s'REVISION agit d'un sur-ensemble de. OPEN

Vous pouvez spécifier plusieurs événements dans le même déclencheur de pull request.

Pour obtenir des exemples, consultez [Exemples de déclencheurs](#).

Interface utilisateur correspondante : éditeur visuel/diagramme de flux de travail/déclencheurs/événements pour la pull request

## Branches

(Triggers/Branches)

## (Facultatif)

Spécifiez les branches de votre référentiel source que le déclencheur surveille afin de savoir quand démarrer une exécution de flux de travail. Vous pouvez utiliser des modèles regex pour définir les noms de vos branches. Par exemple, utilisez `main.*` pour faire correspondre toutes les branches commençant par `main`.

Les branches à spécifier sont différentes selon le type de déclencheur :

- Pour un déclencheur, spécifiez les branches vers lesquelles vous poussez, c'est-à-dire les branches de destination. Une exécution de flux de travail démarrera par branche correspondante, en utilisant les fichiers de la branche correspondante.

Exemples : `main.*`, `mainline`

- Pour un déclencheur de pull request, spécifiez les branches vers lesquelles vous pointez, c'est-à-dire les branches de destination. Une exécution de flux de travail démarrera par branche correspondante, en utilisant le fichier de définition du flux de travail et les fichiers source de la branche source (et non de la branche correspondante).

Exemples : `main.*`, `mainline`, `v1\-.*` (correspond aux branches commençant par `v1-`)

- Pour un déclencheur de planification, spécifiez les branches contenant les fichiers que vous souhaitez que votre exécution planifiée utilise. Une exécution de flux de travail démarrera par branche correspondante, en utilisant le fichier de définition du flux de travail et les fichiers source de la branche correspondante.

Exemples : `main.*`, `version\-1\.*`

### Note

Si vous ne spécifiez pas de branches, le déclencheur surveille toutes les branches de votre référentiel source et lancera l'exécution d'un flux de travail à l'aide du fichier de définition du flux de travail et des fichiers source dans :

- La branche vers laquelle vous poussez (pour les déclencheurs). Pour plus d'informations, consultez [Exemple : un simple déclencheur push à code](#).
- La branche d'où vous effectuez l'extraction (pour les déclencheurs de pull request). Pour plus d'informations, consultez [Exemple : un simple déclencheur de pull request](#).

- Toutes les succursales (pour les déclencheurs de calendrier). Une exécution de flux de travail démarrera par branche de votre référentiel source. Pour plus d'informations, consultez [Exemple : un simple déclencheur de planification](#).

Pour plus d'informations sur les branches et les déclencheurs, consultez [Considérations relatives aux déclencheurs lors du branchement](#).

Pour obtenir plus d'exemples, consultez [Exemples de déclencheurs](#).

Interface utilisateur correspondante : éditeur visuel/diagramme de flux de travail/déclencheurs/branches

FilesChanged

(Triggers/FilesChanged)

(Facultatif si le déclencheur Type est réglé sur PUSH, ou PULLREQUEST. Non pris en charge si le déclencheur Type est réglé sur SCHEDULE.)

Spécifiez les fichiers ou dossiers de votre référentiel source que le déclencheur surveille afin de savoir quand démarrer une exécution de flux de travail. Vous pouvez utiliser des expressions régulières pour faire correspondre les noms de fichiers ou les chemins.

Pour obtenir des exemples, consultez [Exemples de déclencheurs](#).

Interface utilisateur correspondante : éditeur visuel/diagramme de flux de travail/déclencheurs/fichiers modifiés

Expression

(Triggers/Expression)

(Obligatoire si le déclencheur Type est réglé sur SCHEDULE)

Spécifiez l'expression cron qui décrit le moment où vous souhaitez que votre flux de travail planifié ait lieu.

Les expressions Cron CodeCatalyst utilisent la syntaxe à six champs suivante, où chaque champ est séparé par un espace :

*minutes, heures, days-of-monthmois, days-of-weekannée*

## Exemples d'expressions cron

| Minutes | Heures | Jours du mois | Mois | Jours de la semaine | Année | Signification   |
|---------|--------|---------------|------|---------------------|-------|---|
| 0       | 0      | ?             | *    | MON-FRI             | *     | Exécute un flux de travail à minuit (UTC+0) du lundi au vendredi.                 |
| 0       | 2      | *             | *    | ?                   | *     | Exécute un flux de travail à 2 h 00 (UTC +0) tous les jours.                      |
| 15      | 22     | *             | *    | ?                   | *     | Exécute un flux de travail à 22 h 15 (UTC +0) tous les jours.                     |
| 0/30    | 22-2   | ?             | *    | SAT-SUN             | *     | Exécute un flux de travail toutes les 30 minutes du samedi au dimanche entre 22 h |



| Minutes | Heures | Jours du mois | Mois | Jours de la semaine | Année     | Signification   |
|---------|--------|---------------|------|---------------------|-----------|---|
|         |        |               |      |                     |           | 00 le jour de départ et 2 h 00 le jour suivant (UTC+0).   |
| 45      | 13     | L             | *    | ?                   | 2023-2027 | Exécute un flux de travail à 13 h 45 (UTC+0) le dernier jour du mois entre 2023 et 2027 inclus. |

Lorsque vous spécifiez des expressions cron dans CodeCatalyst, assurez-vous de suivre les instructions suivantes :

- Spécifiez une seule expression cron par SCHEDULE déclencheur.
- Mettez l'expression cron entre guillemets (") dans l'éditeur YAML.
- Spécifiez l'heure en temps universel coordonné (UTC). Les autres fuseaux horaires ne sont pas pris en charge.
- Configurez au moins 30 minutes entre les exécutions. Une cadence plus rapide n'est pas prise en charge.
- Spécifiez le *days-of-week* champ *days-of-month* ou, mais pas les deux. Si vous spécifiez une valeur ou un astérisque (\*) dans l'un des champs, vous devez utiliser un point d'interrogation (?) dans l'autre. L'astérisque signifie « tout » et le point d'interrogation signifie « tout ».

Pour plus d'exemples d'expressions cron et des informations sur les caractères génériques tels que ?, et \*L, consultez la [référence des expressions Cron](#) dans le guide de l'utilisateur Amazon EventBridge

. Les expressions Cron entrent EventBridge et CodeCatalyst fonctionnent exactement de la même manière.

Pour des exemples de déclencheurs de planification, voir [Exemples de déclencheurs](#).

Interface utilisateur correspondante : éditeur visuel/diagramme de flux de travail/déclencheurs/calendrier

## Actions

Séquence d'une ou plusieurs actions pour ce flux de travail. CodeCatalyst prend en charge plusieurs types d'actions, tels que les actions de création et de test, qui offrent différents types de fonctionnalités. Chaque type d'action comporte les éléments suivants :

- une `Identifier` propriété qui indique l'identifiant unique et codé en dur de l'action. Par exemple, `aws/build@v1` identifie l'action de génération.
- une `Configuration` section qui contient les propriétés spécifiques à l'action.

Pour plus d'informations sur chaque type d'action, consultez sa documentation de référence, disponible dans le [Référence de définition du flux de travail](#).

Voici la référence YAML pour les actions et les groupes d'actions dans le fichier de définition du flux de travail.

Pour plus d'informations sur les actions, consultez [Utilisation des actions](#).

```
Name: MyWorkflow
SchemaVersion: 1.0
...
Actions:
  action-name:
    Identifier: action-identifiant
    Configuration:
      ...
#Action groups
action-group-name:
  Actions:
    ...
```

## nom-action

(nom de Actions/ *l'action*)

(Obligatoire)

Remplacez *action-name* par *le* nom que vous souhaitez attribuer à l'action. Les noms d'action doivent être uniques dans le flux de travail et ne doivent inclure que des caractères alphanumériques, des traits d'union et des traits de soulignement. Pour plus d'informations sur les règles de syntaxe, consultez [Consignes relatives à la syntaxe YAML](#).

Pour plus d'informations sur les pratiques de dénomination des actions, y compris les restrictions, consultez le [nom-action](#).

Interface utilisateur correspondante : éditeur visuel/nom de *l'action/onglet de configuration/nom de* l'action ou nom d'affichage de l'action

## action-group-name

(Actions/*action-group-name*)

(Facultatif)

Un groupe d'actions contient une ou plusieurs actions. Le regroupement des actions dans des groupes d'actions vous aide à organiser votre flux de travail et vous permet également de configurer les dépendances entre les différents groupes d'actions.

*action-group-name* Remplacez-le par le nom que vous souhaitez attribuer au groupe d'actions. Les noms des groupes d'actions doivent être uniques dans le flux de travail et ne doivent inclure que des caractères alphanumériques, des traits d'union et des traits de soulignement. Pour plus d'informations sur les règles de syntaxe, consultez [Consignes relatives à la syntaxe YAML](#).

Pour plus d'informations sur les groupes d'actions, consultez [Regrouper les actions dans des groupes d'action](#).

Interface utilisateur correspondante : aucune

## Créez et testez une référence d'actions

Vous trouverez ci-dessous la référence YAML de définition d'action pour les actions de construction et de test. Il existe une référence pour deux actions car leurs propriétés YAML sont très similaires.

Choisissez une propriété YAML dans le code suivant pour en voir la description.

### Note

La plupart des propriétés YAML suivantes ont des éléments d'interface utilisateur correspondants dans l'éditeur visuel. Pour rechercher un élément de l'interface utilisateur, utilisez Ctrl+F. L'élément sera répertorié avec sa propriété YAML associée.

```
# The workflow definition starts here.
# See Propriétés de haut niveau for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
action-name:
  Identifier: aws/build@v1 | aws/managed-test@v1
  DependsOn:
    - dependent-action-name-1
  Compute:
    Type: EC2 | Lambda
    Fleet: fleet-name
    Timeout: timeout-minutes
  Environment:
    Name: environment-name
  Connections:
    - Name: account-connection-name
      Role: iam-role-name
  Caching:
    FileCaching:
      key-name-1:
        Path: file1.txt
      RestoreKeys:
        - restore-key-1
  Inputs:
    Sources:
      - source-name-1
      - source-name-2
  Artifacts:
    - artifact-name
```

Variables:

- Name: *variable-name-1*  
Value: *variable-value-1*
- Name: *variable-name-2*  
Value: *variable-value-2*

Outputs:Artifacts:

- Name: *output-artifact-1*

Files:

- build-output/artifact-1.jar
- "build-output/build\*"

- Name: *output-artifact-2*

Files:

- build-output/artifact-2.1.jar
- build-output/artifact-2.2.jar

Variables:

- *variable-name-1*
- *variable-name-2*

AutoDiscoverReports:

Enabled: *true | false*

ReportNamePrefix: *AutoDiscovered*

IncludePaths:

- *\*\*/\**

ExcludePaths:

- *node\_modules/cdk/junit.xml*

SuccessCriteria:

PassRate: *percent*

LineCoverage: *percent*

BranchCoverage: *percent*

Vulnerabilities:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

StaticAnalysisBug:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

StaticAnalysisSecurity:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

StaticAnalysisQuality:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

StaticAnalysisFinding:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

## Reports:

### report-name-1:

Format: *format*

### IncludePaths:

- "\*.xml"

### ExcludePaths:

- report2.xml
- report3.xml

### SuccessCriteria:

PassRate: *percent*

LineCoverage: *percent*

BranchCoverage: *percent*

### Vulnerabilities:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

### StaticAnalysisBug:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

### StaticAnalysisSecurity:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

### StaticAnalysisQuality:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

### StaticAnalysisFinding:

Severity: *CRITICAL | HIGH | MEDIUM | LOW | INFORMATIONAL*

Number: *whole-number*

## Configuration:

### Container:

Registry: *registry*

Image: *image*

### Steps:

- Run: *"step 1"*
- Run: *"step 2"*

### Packages:

#### NpmConfiguration:

#### PackageRegistries:

- PackageRepository: *package-repository*

#### Scopes:

- *"@scope"*

## nom-action

(Obligatoire)

Spécifiez le nom de l'action. Tous les noms d'action doivent être uniques dans le flux de travail. Les noms d'action sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms d'action.

Interface utilisateur correspondante : onglet Configuration/nom de l'action

## Identifiant

*(nom-action/)* Identifiant

Identifie l'action. Ne modifiez pas cette propriété, sauf si vous souhaitez modifier la version. Pour plus d'informations, consultez [Utilisation des versions d'action](#).

À utiliser aws/build@v1 pour créer des actions.

À utiliser aws/managed-test@v1 pour les actions de test.

*Interface utilisateur correspondante : diagramme de flux de travail/Action-Name/ aws/build @v1 |aws/managed-test @v1 label*

## DependsOn

*(nom-action/)* DependsOn

(Facultatif)

Spécifiez une action ou un groupe d'actions qui doit être exécuté correctement pour que cette action soit exécutée.

Pour plus d'informations sur la fonctionnalité « dépend », consultez. [Configuration des actions pour qu'elles dépendent d'autres actions](#)

Interface utilisateur correspondante : onglet Entrées/dépend de - facultatif

## Compute

*(nom-action/)* Compute

## (Facultatif)

Le moteur informatique utilisé pour exécuter les actions de votre flux de travail. Vous pouvez spécifier le calcul au niveau du flux de travail ou au niveau de l'action, mais pas les deux. Lorsqu'elle est spécifiée au niveau du flux de travail, la configuration de calcul s'applique à toutes les actions définies dans le flux de travail. Au niveau du flux de travail, vous pouvez également exécuter plusieurs actions sur la même instance. Pour plus d'informations, consultez [Partage du calcul entre les actions](#).

Interface utilisateur correspondante : aucune

## Type

(type de *nom d'action*/Compute/)

(Obligatoire s'[Compute](#) est inclus)

Type de moteur de calcul. Vous pouvez utiliser l'une des valeurs suivantes :

- EC2 (éditeur visuel) ou EC2 (éditeur YAML)  
Optimisé pour la flexibilité lors des courses d'action.
- Lambda (éditeur visuel) ou Lambda (éditeur YAML)  
Vitesses de démarrage des actions optimisées.

Pour plus d'informations sur les types de calcul, veuillez consulter [À propos des types de calcul](#).

Interface utilisateur correspondante : onglet Configuration/Type de calcul

## Fleet

(nom de *l'action* /Compute/ **Fleet**)

(Facultatif)

Spécifiez la machine ou le parc qui exécutera votre flux de travail ou vos actions de flux de travail. Dans le cas des flottes à la demande, lorsqu'une action démarre, le flux de travail fournit les ressources dont il a besoin et les machines sont détruites à la fin de l'action. Exemples de flottes à la demande :Linux.x86-64.Large,Linux.x86-64.XLarge. Pour plus d'informations sur les flottes à la demande, consultez[Propriétés de flotte à la demande](#).



Avec les flottes provisionnées, vous configurez un ensemble de machines dédiées pour exécuter les actions de votre flux de travail. Ces machines restent inactives, prêtes à exécuter des actions immédiatement. Pour plus d'informations sur les flottes provisionnées, consultez. [Propriétés du parc provisionné](#)

S'il Fleet est omis, la valeur par défaut est `Linux.x86-64.Large`.

Interface utilisateur correspondante : onglet Configuration/parc de calcul

## Timeout

*(nom-action/)* Timeout

(Facultatif)

Spécifiez la durée en minutes (éditeur YAML) ou en heures et minutes (éditeur visuel) pendant laquelle l'action peut être exécutée avant la CodeCatalyst fin de l'action. Le minimum est de 5 minutes et le maximum est décrit dans [Quotas pour les flux de travail dans CodeCatalyst](#). Le délai d'expiration par défaut est le même que le délai d'expiration maximal.

Interface utilisateur correspondante : onglet Configuration/Délai d'expiration - facultatif

## Environnement

*(nom-action/)* Environnement

(Facultatif)

Spécifiez l' CodeCatalyst environnement à utiliser avec l'action.

Pour plus d'informations sur les environnements, reportez-vous [Utilisation des environnements](#) aux sections et [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/Environnement

## Name

(nom de *l'action /Environment/ Name*)

(Facultatif)

Spécifiez le nom de l'environnement existant que vous souhaitez associer à l'action.

Interface utilisateur correspondante : onglet Configuration/nom de l'environnement

## Connections

(nom de *l'action /Environment/ Connections*)

(Facultatif)

Spécifiez la connexion au compte à associer à l'action. Vous pouvez spécifier un maximum d'une connexion à un compte sous `Environment`.

Pour plus d'informations sur les connexions aux comptes, consultez [Administration Comptes AWS d'un espace](#). Pour plus d'informations sur la façon d'associer une connexion à un compte à votre environnement, consultez [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/

### Name

(nom de *l'action /Environment/Connections/ Name*)

(Facultatif)

Spécifiez le nom de la connexion au compte.

Interface utilisateur correspondante : onglet Configuration/

### Role

(nom de *l'action /Environment/Connections/ Role*)

(Facultatif)

Spécifiez le nom du rôle IAM que cette action utilise pour accéder et opérer dans des AWS services tels qu'Amazon S3 et Amazon ECR. Assurez-vous que ce rôle est ajouté à la connexion de votre compte. Pour ajouter un rôle IAM à une connexion à un compte, consultez [Ajouter des rôles IAM aux connexions aux comptes](#).

#### Note

Vous pouvez peut-être spécifier le nom du `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle ici, à condition qu'il dispose des autorisations suffisantes. Pour plus d'informations sur ce rôle, consultez [Création du CodeCatalystWorkflowDevelopmentRole-\*spaceName\* rôle pour votre compte et votre](#)

[espace](#). Sachez que le `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle dispose d'autorisations très étendues, ce qui peut présenter un risque de sécurité. Nous vous recommandons de n'utiliser ce rôle que dans les didacticiels et les scénarios où la sécurité est moins préoccupante.

#### Warning

Limitez les autorisations à celles requises par les actions de génération et de test. L'utilisation d'un rôle doté d'autorisations plus étendues peut présenter un risque de sécurité.

Interface utilisateur correspondante : onglet Configuration/

## Caching

(*nom-action/*) Caching

(Facultatif)

Section dans laquelle vous pouvez spécifier un cache pour enregistrer les fichiers sur disque et les restaurer à partir de ce cache lors des exécutions de flux de travail suivantes.

Pour plus d'informations sur la mise en cache de fichiers, consultez [Utilisation de la mise en cache de fichiers](#).

Interface utilisateur correspondante : onglet Configuration/mise en cache de fichiers - facultatif

## FileCaching

(nom de *l'action /Caching/ FileCaching*)

(Facultatif)

Section qui spécifie la configuration d'une séquence de caches.

Interface utilisateur correspondante : onglet Configuration/Mise en cache de fichiers - facultatif/

Ajouter un cache

nom-clé-1

(*nom-action-clé-nom-1/Caching/FileCaching/*)

## (Facultatif)

Spécifiez le nom de votre propriété de cache principale. Les noms des propriétés du cache doivent être uniques dans votre flux de travail. Chaque action peut comporter jusqu'à cinq entrées `FileCaching`.

Interface utilisateur correspondante : onglet Configuration/Mise en cache des fichiers - Facultatif/Ajouter un cache/une clé

## Path

*(/Caching/FileCaching/nom-action-nom-clé-1/)* Path

## (Facultatif)

Spécifiez le chemin associé à votre cache.

Interface utilisateur correspondante : onglet Configuration/mise en cache des fichiers - Facultatif/Ajouter un cache/un chemin

## RestoreKeys

*(/Caching/FileCaching/nom-action-nom-clé-1/)* RestoreKeys

## (Facultatif)

Spécifiez la clé de restauration à utiliser comme solution de secours lorsque la propriété de cache principale est introuvable. Les noms des clés de restauration doivent être uniques dans votre flux de travail. Chaque cache peut contenir jusqu'à cinq entrées `RestoreKeys`.

Interface utilisateur correspondante : onglet Configuration/Mise en cache des fichiers - Facultatif/Ajout de clés de cache/restauration - facultatif

## Inputs

*(nom-action/)* Inputs

## (Facultatif)

La `Inputs` section définit les données dont une action a besoin lors de l'exécution d'un flux de travail.

**Note**

Un maximum de quatre entrées (une source et trois artefacts) sont autorisées par action de construction ou action de test. Les variables ne sont pas prises en compte dans ce total.

Si vous devez faire référence à des fichiers résidant dans différentes entrées (par exemple, une source et un artefact), l'entrée source est l'entrée principale et l'artefact est l'entrée secondaire. Les références aux fichiers dans les entrées secondaires utilisent un préfixe spécial pour les distinguer du fichier principal. Pour plus de détails, consultez [Exemple : Référencement de fichiers dans plusieurs artefacts](#).

Interface utilisateur correspondante : onglet Entrées

## Sources

(nom de *l'action /Inputs/ Sources*)

(Facultatif)

Spécifiez les étiquettes qui représentent les référentiels sources qui seront nécessaires à l'action. Actuellement, la seule étiquette prise en charge est `WorkflowSource`, qui représente le référentiel source dans lequel votre fichier de définition de flux de travail est stocké.

Si vous omettez une source, vous devez spécifier au moins un artefact d'entrée sous. *action-name/Inputs/Artifacts*

Pour plus d'informations sur les sources, consultez [Travailler avec des sources](#).

Interface utilisateur correspondante : aucune

## Artifacts - input

(nom de *l'action /Inputs/ Artifacts*)

(Facultatif)

Spécifiez les artefacts des actions précédentes que vous souhaitez fournir en entrée pour cette action. Ces artefacts doivent déjà être définis en tant qu'artefacts de sortie dans les actions précédentes.

Si vous ne spécifiez aucun artefact d'entrée, vous devez spécifier au moins un référentiel source sous *action-name*/Inputs/Sources.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

#### Note

Si la liste déroulante Artéfacts - optionnelle n'est pas disponible (éditeur visuel), ou si vous recevez des erreurs lors de la validation de votre YAML (éditeur YAML), c'est peut-être parce que l'action ne prend en charge qu'une seule entrée. Dans ce cas, essayez de supprimer l'entrée source.

Interface utilisateur correspondante : onglet Entrées/Artefacts - facultatif

## Variables - input

(nom de *l'action* /Inputs/ **Variables**)

(Facultatif)

Spécifiez une séquence de paires nom/valeur qui définissent les variables d'entrée que vous souhaitez mettre à la disposition de l'action. Les noms de variables sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms de variables.

Pour plus d'informations sur les variables, y compris des exemples, consultez [Utilisation des variables](#).

Interface utilisateur correspondante : onglet Entrées/Variables - facultatif

## Outputs

(*nom-action*/) Outputs

(Facultatif)

Définit les données produites par l'action lors de l'exécution d'un flux de travail.

Interface utilisateur correspondante : onglet Sorties

## Artifacts - output

(nom de *l'action /Outputs/ Artifacts*)

(Facultatif)

Spécifiez le nom d'un artefact généré par l'action. Les noms d'artefact doivent être uniques dans un flux de travail et sont limités aux caractères alphanumériques (a-z, A-Z, 0-9) et aux traits de soulignement (\_). Les espaces, les tirets (-) et les autres caractères spéciaux ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les espaces, les tirets et autres caractères spéciaux dans les noms d'artefacts en sortie.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

Interface utilisateur correspondante : onglet Sorties/Artefacts

## Name

(nom de *l'action /Outputs/Artifacts/ Name*)

(Obligatoire s'[Artifacts - output](#) est inclus)

Spécifiez le nom d'un artefact généré par l'action. Les noms d'artefact doivent être uniques dans un flux de travail et sont limités aux caractères alphanumériques (a-z, A-Z, 0-9) et aux traits de soulignement (\_). Les espaces, les tirets (-) et les autres caractères spéciaux ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les espaces, les tirets et autres caractères spéciaux dans les noms d'artefacts en sortie.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

Interface utilisateur correspondante : onglet Sorties/Artifacts/Nouvelle sortie/Nom de l'artefact de construction

## Files

(nom de *l'action /Outputs/Artifacts/ Files*)

(Obligatoire s'[Artifacts - output](#) est inclus)

Spécifiez les fichiers CodeCatalyst inclus dans l'artefact généré par l'action. Ces fichiers sont générés par l'action du flux de travail lorsqu'elle s'exécute et sont également disponibles dans votre référentiel source. Les chemins de fichiers peuvent résider dans un référentiel source ou dans un artefact issu d'une action précédente, et sont relatifs au référentiel source ou à la racine de l'artefact. Vous pouvez utiliser des modèles globulaires pour définir des chemins. Exemples :

- Pour spécifier un seul fichier situé à la racine de votre emplacement de compilation ou de l'emplacement de votre référentiel source, utilisez `my-file.jar`.
- Pour spécifier un seul fichier dans un sous-répertoire, utilisez `directory/my-file.jar` ou `directory/subdirectory/my-file.jar`.
- Pour spécifier tous les fichiers, utilisez `**/*`. Le modèle `**` glob indique qu'il doit correspondre à un nombre quelconque de sous-répertoires.
- Pour spécifier tous les fichiers et répertoires d'un répertoire nommé `directory`, utilisez `directory/**/*`. Le modèle `**` glob indique qu'il doit correspondre à un nombre quelconque de sous-répertoires.
- Pour spécifier tous les fichiers d'un répertoire nommé `directory`, mais aucun de ses sous-répertoires, utilisez `directory/*`.

#### Note

Si le chemin de votre fichier comporte un ou plusieurs astérisques (\*) ou autres caractères spéciaux, mettez-le entre guillemets (). Pour plus d'informations sur les caractères spéciaux, consultez [Consignes et conventions de syntaxe](#).

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

#### Note

Vous devrez peut-être ajouter un préfixe au chemin du fichier pour indiquer dans quel artefact ou dans quelle source le trouver. Pour plus d'informations, consultez [Référencement de fichiers dans un référentiel source](#) et [Référencement de fichiers dans un artefact](#).



Interface utilisateur correspondante : onglet Sorties/Artefacts/Nouvelle sortie/Fichiers produits par build

## Variables - output

(nom de *l'action /Outputs/ Variables*)

(Facultatif)

Spécifiez les variables que vous souhaitez que l'action exporte afin qu'elles puissent être utilisées par les actions suivantes.

Pour plus d'informations sur les variables, y compris des exemples, consultez [Utilisation des variables](#).

Interface utilisateur correspondante : onglet Sorties/Variables/Ajouter une variable

nom-variable-1

(*nom-action-nom-variable-1/Outputs/Variables/*)

(Facultatif)

Spécifiez le nom de la variable que vous souhaitez que l'action exporte. Cette variable doit déjà être définie dans la Steps section Inputs ou de la même action.

Pour plus d'informations sur les variables, y compris des exemples, consultez [Utilisation des variables](#).

Interface utilisateur correspondante : onglet Sorties/Variables/Ajouter une variable/Nom

## AutoDiscoverReports


(nom de *l'action /Outputs/ AutoDiscoverReports*)

(Facultatif)

Définit la configuration de la fonctionnalité de découverte automatique.

Lorsque vous activez la découverte automatique, les CodeCatalyst recherches Inputs sont toutes transmises à l'action ainsi que tous les fichiers générés par l'action elle-même, à la recherche de rapports de test, de couverture de code et d'analyse de la composition logicielle (SCA). Pour chaque rapport trouvé, le CodeCatalyst transforme en CodeCatalyst rapport. Un CodeCatalyst rapport est un

rapport qui est entièrement intégré au CodeCatalyst service et qui peut être consulté et manipulé via la CodeCatalyst console.

 Note

Par défaut, la fonction de découverte automatique inspecte tous les fichiers. Vous pouvez limiter les fichiers inspectés à l'aide des [ExcludePaths](#) propriétés [IncludePaths](#) or.

Interface utilisateur correspondante : onglet Sorties/Rapports/Rapports de découverte automatique

## Enabled

(nom de *l'action /Outputs/AutoDiscoverReports/ Enabled*)

(Facultatif)

Activez ou désactivez la fonction de découverte automatique.

Les valeurs valides sont true ou false.

S'il Enabled est omis, la valeur par défaut est true.

Interface utilisateur correspondante : onglet Sorties/Rapports/Rapports de découverte automatique

## ReportNamePrefix

(nom de *l'action /Outputs/AutoDiscoverReports/ ReportNamePrefix*)

(Obligatoire s'[AutoDiscoverReports](#) est inclus et activé)

Spécifiez un préfixe qui CodeCatalyst précède tous les rapports trouvés afin de nommer les rapports associés. CodeCatalyst Par exemple, si vous spécifiez le préfixe deAutoDiscovered, et que vous CodeCatalyst découvrez automatiquement deux rapports de testTestSuiteTwo.xml, TestSuiteOne.xml et, les CodeCatalyst rapports associés seront nommés AutoDiscoveredTestSuiteOne et. AutoDiscoveredTestSuiteTwo

Interface utilisateur correspondante : onglet Sorties/Rapports/Nom du préfixe

## IncludePaths

(nom de *l'action /Outputs/AutoDiscoverReports/ IncludePaths*)

Ou

*(/Outputs/Reports/nom-action-nom-rapport-1/)* IncludePaths

(Obligatoire s'[AutoDiscoverReports](#) est inclus et activé, ou s'il [Reports](#) est inclus)

Spécifiez les fichiers et les chemins de fichiers CodeCatalyst inclus lors de la recherche de rapports bruts. Par exemple, si vous le spécifiez `/test/report/*`, CodeCatalyst recherche l'intégralité de [l'image de construction](#) utilisée par l'action à la recherche du `/test/report/*` répertoire. Lorsqu'il trouve ce répertoire, CodeCatalyst il recherche des rapports dans ce répertoire.

#### Note

Si le chemin de votre fichier comporte un ou plusieurs astérisques (\*) ou autres caractères spéciaux, mettez-le entre guillemets (). "" Pour plus d'informations sur les caractères spéciaux, consultez [Consignes et conventions de syntaxe](#).

Si cette propriété est omise, la valeur par défaut est `**/*`, ce qui signifie que la recherche inclut tous les fichiers, quel que soit leur chemin.

#### Note

Pour les rapports configurés manuellement, `IncludePaths` il doit s'agir d'un modèle global correspondant à un seul fichier.

Interface utilisateur correspondante :

- Onglet Sorties/Rapports/Rapports de découverte automatique/Inclure/Exclure des chemins/Inclure des chemins
- ***Onglet Sorties/Rapports/Configuration manuelle des rapports/nom-du-rapport-1 /Inclure/Exclure les chemins/ Inclure les chemins***

ExcludePaths

(nom de *l'action /Outputs/AutoDiscoverReports/ ExcludePaths*)

Ou

*(/Outputs/Reports/nom-action-nom-rapport-1/)* ExcludePaths

## (Facultatif)

Spécifiez les fichiers et les chemins de fichiers à CodeCatalyst exclure lors de la recherche de rapports bruts. Par exemple, si vous le spécifiez `/test/my-reports/**/*`, ne CodeCatalyst recherchera pas de fichiers dans le `/test/my-reports/` répertoire. Pour ignorer tous les fichiers d'un répertoire, utilisez le modèle `**/*` glob.

### Note

Si le chemin de votre fichier comporte un ou plusieurs astérisques (\*) ou autres caractères spéciaux, mettez-le entre guillemets (). Pour plus d'informations sur les caractères spéciaux, consultez [Consignes et conventions de syntaxe](#).

Interface utilisateur correspondante :

- Onglet Sorties/Rapports/Rapports de découverte automatique/Inclure/Exclure des chemins/Exclure des chemins
- **Onglet Sorties/Rapports/Configuration manuelle des rapports/nom-du-rapport-1 /Inclure/Exclure des chemins/Exclure des chemins**

## SuccessCriteria

(nom de *l'action /Outputs/AutoDiscoverReports/ SuccessCriteria*)

Ou

*(/Outputs/Reports/nom-action-nom-rapport-1/)* SuccessCriteria

## (Facultatif)

Spécifiez les critères de réussite pour les rapports de test, de couverture du code, d'analyse de la composition logicielle (SCA) et d'analyse statique (SA).

Pour plus d'informations, consultez [Configuration des critères de réussite pour les rapports](#).

Interface utilisateur correspondante : onglet Sortie/Rapports/Critères de réussite

## PassRate

(nom de *l'action /Outputs/AutoDiscoverReports/SuccessCriteria/ PassRate*)

Ou

*(nom-action-nom-rapport-1/Outputs/Reports/) /SuccessCriteria/ **PassRate***

(Facultatif)

Spécifiez le pourcentage de tests dans un rapport de test qui doivent être réussis pour que le CodeCatalyst rapport associé soit marqué comme réussi. Les valeurs valides incluent les nombres décimaux. Par exemple, 50, 60.5. Les critères de taux de réussite ne sont appliqués qu'aux rapports de test. Pour plus d'informations sur les rapports de test, consultez [Rapports d'essais](#).

Interface utilisateur correspondante : onglet Sortie/Rapports/Critères de succès/Taux de réussite

## LineCoverage

(nom de l'action */Outputs/AutoDiscoverReports/SuccessCriteria/ **LineCoverage***)

Ou

*(nom-action-nom-rapport-1/Outputs/Reports/) /SuccessCriteria/ **LineCoverage***

(Facultatif)

Spécifiez le pourcentage de lignes d'un rapport de couverture de code qui doivent être couvertes pour que le CodeCatalyst rapport associé soit marqué comme transmis. Les valeurs valides incluent les nombres décimaux. Par exemple, 50, 60.5. Les critères de couverture de ligne sont appliqués uniquement aux rapports de couverture de code. Pour plus d'informations sur les rapports de couverture du code, consultez [Rapports sur la couverture du code](#).

Interface utilisateur correspondante : onglet Sortie/Rapports/Critères de succès/Couverture des lignes

## BranchCoverage

(nom de l'action */Outputs/AutoDiscoverReports/SuccessCriteria/ **BranchCoverage***)

Ou

*(nom-action-nom-rapport-1/Outputs/Reports/) /SuccessCriteria/ **BranchCoverage***

(Facultatif)

Spécifiez le pourcentage de branches dans un rapport de couverture de code qui doivent être couvertes pour que le CodeCatalyst rapport associé soit marqué comme transmis. Les valeurs valides incluent les nombres décimaux. Par exemple, 50, 60.5. Les critères de couverture des succursales sont appliqués uniquement aux rapports de couverture par code. Pour plus d'informations sur les rapports de couverture du code, consultez [Rapports sur la couverture du code](#).

Interface utilisateur correspondante : onglet de sortie/Rapports/Critères de succès/Couverture des branches

## Vulnerabilities

(nom de *l'action /Outputs/AutoDiscoverReports/SuccessCriteria/Vulnerabilities*)

Ou

(*nom-action-nom-rapport-1/Outputs/Reports/*) */SuccessCriteria/Vulnerabilities*

(Facultatif)

Spécifiez le nombre maximum et la gravité des vulnérabilités autorisées dans le rapport SCA pour que le CodeCatalyst rapport associé soit marqué comme transmis. Pour définir les vulnérabilités, vous devez spécifier :

- La gravité minimale des vulnérabilités que vous souhaitez inclure dans le décompte. Les valeurs valides, de la plus sévère à la moins sévère, sont les suivantes : CRITICALHIGH,MEDIUM,LOW,,INFORMATIONAL.

Par exemple, si vous le souhaitezHIGH, HIGH les CRITICAL vulnérabilités seront comptabilisées.

- Le nombre maximum de vulnérabilités de la gravité spécifiée que vous souhaitez autoriser. Le dépassement de ce nombre entraîne le marquage CodeCatalyst du rapport comme ayant échoué. Les valeurs valides sont des nombres entiers.

Les critères de vulnérabilité sont appliqués uniquement aux rapports SCA. Pour plus d'informations sur les rapports SCA, consultez [Rapports d'analyse de la composition des logiciels](#).

Pour spécifier la sévérité minimale, utilisez la `Severity` propriété. Pour spécifier le nombre maximum de vulnérabilités, utilisez la `Number` propriété.

Interface utilisateur correspondante : onglet Sortie/Rapports/Critères de succès/Vulnérabilités

## StaticAnalysisBug

(nom de l'action */Outputs/AutoDiscoverReports/SuccessCriteria/StaticAnalysisBug*)

Ou

(*nom-action-nom-rapport-1/Outputs/Reports/*) */SuccessCriteria/StaticAnalysisBug*

(Facultatif)

Spécifiez le nombre maximum et la gravité des bogues autorisés dans le rapport SA pour que le CodeCatalyst rapport associé soit marqué comme passé. Pour spécifier les bogues, vous devez spécifier :

- La gravité minimale des bogues que vous souhaitez inclure dans le décompte. Les valeurs valides, de la plus sévère à la moins sévère, sont les suivantes : CRITICALHIGH,MEDIUM,LOW,,INFORMATIONAL.  
Par exemple, si vous le souhaitezHIGH, HIGH les CRITICAL bogues seront comptabilisés.
- Le nombre maximum de bogues de la gravité spécifiée que vous souhaitez autoriser. Le dépassement de ce nombre entraîne le marquage CodeCatalyst du rapport comme ayant échoué. Les valeurs valides sont des nombres entiers.

Les critères de bogues ne sont appliqués qu' PyLint aux rapports ESLint SA. Pour plus d'informations sur les rapports SA, consultez [Rapports d'analyse statiques](#).

Pour spécifier la sévérité minimale, utilisez la Severity propriété. Pour spécifier le nombre maximum de vulnérabilités, utilisez la Number propriété.

Interface utilisateur correspondante : onglet de sortie/Rapports/Critères de succès/Bugs

## StaticAnalysisSecurity

(nom de l'action */Outputs/AutoDiscoverReports/SuccessCriteria/StaticAnalysisSecurity*)

Ou

*(nom-action-nom-rapport-1/Outputs/Reports/) /SuccessCriteria/  
**StaticAnalysisSecurity***

(Facultatif)

Spécifiez le nombre maximum et la gravité des vulnérabilités de sécurité autorisées dans le rapport SA pour que le CodeCatalyst rapport associé soit marqué comme transmis. Pour définir les failles de sécurité, vous devez spécifier :

- La gravité minimale des failles de sécurité que vous souhaitez inclure dans le décompte. Les valeurs valides, de la plus sévère à la moins sévère, sont les suivantes : CRITICALHIGH,MEDIUM,LOW,,INFORMATIONAL.

Par exemple, si vous le souhaitezHIGH, HIGH les failles CRITICAL de sécurité seront prises en compte.

- Le nombre maximum de failles de sécurité de la gravité spécifiée que vous souhaitez autoriser. Le dépassement de ce nombre entraîne le marquage CodeCatalyst du rapport comme ayant échoué. Les valeurs valides sont des nombres entiers.

Les critères de vulnérabilité de sécurité ne s'appliquent qu' PyLint aux rapports ESLint SA. Pour plus d'informations sur les rapports SA, consultez[Rapports d'analyse statiques](#).

Pour spécifier la sévérité minimale, utilisez la Severity propriété. Pour spécifier le nombre maximum de vulnérabilités, utilisez la Number propriété.

Interface utilisateur correspondante : onglet de sortie/rapports/critères de succès/vulnérabilités de sécurité

## StaticAnalysisQuality

*(nom de l'action /Outputs/AutoDiscoverReports/SuccessCriteria/  
**StaticAnalysisQuality**)*

Ou

*(nom-action-nom-rapport-1/Outputs/Reports/) /SuccessCriteria/  
**StaticAnalysisQuality***



## (Facultatif)

Spécifiez le nombre maximum et la gravité des problèmes de qualité autorisés dans le rapport SA pour que le CodeCatalyst rapport associé soit marqué comme réussi. Pour définir les problèmes de qualité, vous devez spécifier :

- La gravité minimale des problèmes de qualité que vous souhaitez inclure dans le décompte. Les valeurs valides, de la plus sévère à la moins sévère, sont les suivantes : CRITICALHIGH,MEDIUM,LOW,,INFORMATIONAL.

Par exemple, si vous le souhaitezHIGH, HIGH les problèmes CRITICAL de qualité seront pris en compte.

- Le nombre maximum de problèmes de qualité de la gravité spécifiée que vous souhaitez autoriser. Le dépassement de ce nombre entraîne le marquage CodeCatalyst du rapport comme ayant échoué. Les valeurs valides sont des nombres entiers.

Les critères relatifs aux problèmes de qualité ne s'appliquent qu' PyLint aux rapports ESLint SA. Pour plus d'informations sur les rapports SA, consultez [Rapports d'analyse statiques](#).

Pour spécifier la sévérité minimale, utilisez la Severity propriété. Pour spécifier le nombre maximum de vulnérabilités, utilisez la Number propriété.

Interface utilisateur correspondante : onglet de sortie/Rapports/Critères de succès/Problèmes de qualité

## StaticAnalysisFinding

(nom de *l'action /Outputs/AutoDiscoverReports/SuccessCriteria/**StaticAnalysisFinding***)

Ou

(*nom-action-nom-rapport-1/Outputs/Reports/*) */SuccessCriteria/**StaticAnalysisFinding***

## (Facultatif)

Spécifiez le nombre maximum et la gravité des résultats autorisés dans le rapport SA pour que le CodeCatalyst rapport associé soit marqué comme adopté. Pour spécifier les résultats, vous devez spécifier :

- La gravité minimale des résultats que vous souhaitez inclure dans le décompte. Les valeurs valides, de la plus sévère à la moins sévère, sont les suivantes : CRITICALHIGH,MEDIUM,LOW,,INFORMATIONAL.  
Par exemple, si vous le souhaitezHIGH, HIGH les CRITICAL résultats seront comptabilisés.
- Le nombre maximum de résultats de la gravité spécifiée que vous souhaitez autoriser. Le dépassement de ce nombre entraîne le marquage CodeCatalyst du rapport comme ayant échoué. Les valeurs valides sont des nombres entiers.

Les résultats ne sont appliqués qu'aux rapports de la SARIF SA. Pour plus d'informations sur les rapports SA, consultez [Rapports d'analyse statiques](#).

Pour spécifier la sévérité minimale, utilisez la `Severity` propriété. Pour spécifier le nombre maximum de vulnérabilités, utilisez la `Number` propriété.

Interface utilisateur correspondante : onglet Produit/Rapports/Critères de succès/Conclusions

## Reports

(nom de *l'action /Outputs/ Reports*)

(Facultatif)

Section qui spécifie la configuration des rapports de test.

Interface utilisateur correspondante : onglet Sorties/Rapports

nom-du-rapport-1

(*nom-action-nom-rapport-1/Outputs/Reports/*)

(Obligatoire s'[Reports](#) est inclus)

Le nom que vous souhaitez donner au CodeCatalyst rapport qui sera généré à partir de vos rapports bruts.

Interface utilisateur correspondante : onglet Sorties/Rapports/Configuration manuelle des rapports/

Nom du rapport

## Format

(*/Outputs/Reports/nom-action-nom-rapport-1/*) Format

(Obligatoire s'[Reports](#)il est inclus)

Spécifiez le format de fichier que vous utilisez pour vos rapports. Les valeurs possibles sont les suivantes.

- Pour les rapports de test :
  - Pour Cucumber JSON, spécifiez Cucumber (éditeur visuel) ou CUCUMBERJSON (éditeur YAML).
  - Pour JUnit XML, spécifiez JUnit (éditeur visuel) ou JUNITXML (éditeur YAML).
  - Pour NUnit XML, spécifiez NUnit (éditeur visuel) ou NUNITXML (éditeur YAML).
  - Pour NUnit 3 XML, spécifiez NUnit3 (éditeur visuel) ou NUNIT3XML (éditeur YAML).
  - Pour Visual Studio TRX, spécifiez Visual Studio TRX (éditeur visuel) ou VISUALSTUDIOTRX (éditeur YAML).
  - Pour TestNG XML, spécifiez TestNG (éditeur visuel) ou TESTNGXML (éditeur YAML).
- Pour les rapports sur la couverture du code :
  - Pour Clover XML, spécifiez Clover (éditeur visuel) ou CLOVERXML (éditeur YAML).
  - Pour Cobertura XML, spécifiez Cobertura (éditeur visuel) ou COBERTURAXML (éditeur YAML).
  - Pour le JaCoCo XML, spécifiez JaCoCo(éditeur visuel) ou JACOCOXML (éditeur YAML).
  - Pour le SimpleCov JSON généré par [simplecov](#), et non par [simplecov-json](#), spécifiez Simplecov (éditeur visuel) ou (éditeur YAML). SIMPLECOV
- Pour les rapports d'analyse de composition logicielle (SCA) :
  - Pour SARIF, spécifiez SARIF (éditeur visuel) ou SARIFSCA (éditeur YAML).

***Interface utilisateur correspondante : onglet Sorties/Rapports/  
Configuration manuelle des rapports/Ajouter/configurer des rapports/nom-du-  
rapport-1 /Type de rapport et format de rapport***

## Configuration

*(nom-action/)* Configuration

(Obligatoire) Section dans laquelle vous pouvez définir les propriétés de configuration de l'action.

Interface utilisateur correspondante : onglet Configuration

## Container

(nom de *l'action /Configuration/ Container*)

## (Facultatif)

Spécifiez l'image Docker, ou le conteneur, que l'action utilise pour terminer son traitement. Vous pouvez spécifier l'une [des images actives](#) fournies avec CodeCatalyst ou utiliser votre propre image. Si vous choisissez d'utiliser votre propre image, elle peut résider dans Amazon ECR, Docker Hub ou dans un autre registre. Si vous ne spécifiez aucune image Docker, l'action utilise l'une des images actives pour son traitement. Pour plus d'informations sur l'image active utilisée par défaut, consultez [Images actives](#).

Pour plus d'informations sur la spécification de votre propre image Docker, consultez [Affectation d'une image Docker d'environnement d'exécution personnalisée à une action](#).

Interface utilisateur correspondante : image Docker de l'environnement d'exécution - facultatif

## Registry

(nom de *l'action /Configuration/Container/ Registry*)

(Obligatoire si Container est inclus)

Spécifiez le registre dans lequel votre image est stockée. Les valeurs valides sont les suivantes :

- CODECATALYST (éditeur YAML)

L'image est enregistrée dans le CodeCatalyst registre.

- Docker Hub (éditeur visuel) ou DockerHub (éditeur YAML)

L'image est stockée dans le registre d'images de Docker Hub.

- Autre registre (éditeur visuel) ou Other (éditeur YAML)

L'image est stockée dans un registre d'images personnalisé. N'importe quel registre accessible au public peut être utilisé.

- Amazon Elastic Container Registry (éditeur visuel) ou ECR (éditeur YAML)

L'image est stockée dans un référentiel d'images Amazon Elastic Container Registry. Pour utiliser une image dans un référentiel Amazon ECR, cette action nécessite l'accès à Amazon ECR. Pour activer cet accès, vous devez créer un [rôle IAM](#) incluant les autorisations suivantes et une politique de confiance personnalisée. (Vous pouvez modifier un rôle existant pour inclure les autorisations et la politique, si vous le souhaitez.)

Le rôle IAM doit inclure les autorisations suivantes dans sa politique de rôle :

- `ecr:BatchCheckLayerAvailability`
- `ecr:BatchGetImage`
- `ecr:GetAuthorizationToken`
- `ecr:GetDownloadUrlForLayer`

Le rôle IAM doit inclure la politique de confiance personnalisée suivante :

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

Pour plus d'informations sur la création de rôles IAM, consultez la section [Création d'un rôle à l'aide de politiques de confiance personnalisées \(console\)](#) dans le guide de l'utilisateur IAM.

Une fois le rôle créé, vous devez l'attribuer à l'action par le biais d'un environnement. Pour plus d'informations, consultez [Associer un environnement, une connexion à un compte et un rôle IAM à une action de flux de travail](#).

Interface utilisateur correspondante : Amazon Elastic Container Registry, Docker Hub et autres options de registre

## Image

(nom de *l'action /Configuration/Container/ Image*)

(Obligatoire si Containeril est inclus)

Spécifiez l'un des éléments suivants :

- Si vous utilisez un CODECATALYST registre, définissez l'image sur l'une des [images actives](#) suivantes :
  - CodeCatalystLinux\_x86\_64:2024\_03
  - CodeCatalystLinux\_x86\_64:2022\_11
  - CodeCatalystLinux\_Arm64:2024\_03
  - CodeCatalystLinux\_Arm64:2022\_11
  - CodeCatalystLinuxLambda\_x86\_64:2024\_03
  - CodeCatalystLinuxLambda\_x86\_64:2022\_11
  - CodeCatalystLinuxLambda\_Arm64:2024\_03
  - CodeCatalystLinuxLambda\_Arm64:2022\_11
  - CodeCatalystWindows\_x86\_64:2022\_11
- Si vous utilisez un registre Docker Hub, attribuez à l'image le nom de l'image Docker Hub et la balise facultative.

Exemple : postgres:latest

- Si vous utilisez un registre Amazon ECR, définissez l'image sur l'URI du registre Amazon ECR.

Exemple : 111122223333.dkr.ecr.us-west-2.amazonaws.com/codecatalyst-ecs-image-repo

- Si vous utilisez un registre personnalisé, définissez l'image sur la valeur attendue par le registre personnalisé.

Interface utilisateur correspondante : image docker de l'environnement d'exécution (si le registre l'est **CODECATALYST**), image Docker Hub (si le registre est Docker Hub), URL de l'image ECR (si le registre est Amazon Elastic Container Registry) et URL de l'image (si le registre est Autre registre).

## Steps

(nom de *l'action /Configuration/ Steps*)

(Obligatoire)

Spécifiez les commandes shell que vous souhaitez exécuter pendant l'action d'installation, de configuration et d'exécution de vos outils de génération.

Voici un exemple de création d'un projet npm :

Steps:

- Run: npm install
- Run: npm run build

Voici un exemple de la manière de spécifier des chemins de fichiers :

Steps:

- Run: cd \$ACTION\_BUILD\_SOURCE\_PATH\_WorkflowSource/app && cat file2.txt
- Run: cd \$ACTION\_BUILD\_SOURCE\_PATH\_MyBuildArtifact/build-output/ && cat file.txt

Pour plus d'informations sur la spécification des chemins de fichiers, reportez-vous [Référencement de fichiers dans un référentiel source](#) aux sections et [Référencement de fichiers dans un artefact](#).

Interface utilisateur correspondante : onglet Configuration/commandes Shell

## Packages

(nom de *l'action /Configuration/ Packages*)

(Facultatif)

Section dans laquelle vous pouvez spécifier un référentiel de packages que l'action utilise pour résoudre les dépendances. Les packages vous permettent de stocker et de partager en toute sécurité les packages logiciels utilisés pour le développement d'applications.

Pour plus d'informations sur les packages, consultez [Packages en CodeCatalyst](#).

Interface utilisateur correspondante : onglet Configuration/Packages

## NpmConfiguration

(nom de *l'action /Configuration/Packages/ NpmConfiguration*)

(Obligatoire s'[Packages](#) est inclus)

Une section qui définit la configuration du format de package npm. Cette configuration est utilisée par une action lors de l'exécution d'un flux de travail.

Pour plus d'informations sur la configuration du package npm, consultez [Utilisation de npm](#).

Interface utilisateur correspondante : onglet Configuration/Packages/Ajouter une configuration/npm

## PackageRegistries

(nom de l'action */Configuration/Packages/NpmConfiguration/PackageRegistries*)

(Obligatoire s'[Packages](#) est inclus)

Section dans laquelle vous pouvez définir les propriétés de configuration d'une séquence de référentiels de packages.

Interface utilisateur correspondante : onglet Configuration/Packages/Ajouter une configuration/npm/Ajouter un référentiel de packages

## PackagesRepository

(nom de l'action */Configuration/Packages/NpmConfiguration/PackageRegistries/PackagesRepository*)

(Obligatoire s'[Packages](#) est inclus)

Spécifiez le nom du référentiel de CodeCatalyst packages que vous souhaitez utiliser par l'action.

Si vous spécifiez plusieurs référentiels par défaut, le dernier sera prioritaire.

Pour plus d'informations sur les référentiels de packages, consultez [Référentiels de packages](#).

Interface utilisateur correspondante : onglet Configuration/Packages/Ajouter une configuration/NPM/Ajouter un référentiel de paquets/Référentiel de packages

## Scopes

(nom de l'action */Configuration/Packages/NpmConfiguration/PackageRegistries/Scopes*)

(Facultatif)

Spécifiez une séquence de portées que vous souhaitez définir dans votre registre de packages. Lors de la définition des étendues, le référentiel de packages spécifié est configuré comme registre pour toutes les étendues répertoriées. Si un package avec cette portée est demandé via le client npm, il



utilisera ce référentiel au lieu du dépôt par défaut. Le nom de chaque étendue doit être préfixé par « @ ».

Si vous incluez des étendues de remplacement, le dernier référentiel aura la priorité.

Si Scopes ce paramètre est omis, le référentiel de packages spécifié est configuré comme registre par défaut pour tous les packages utilisés par l'action.

Pour plus d'informations sur les étendues, reportez-vous à la section [Espaces de noms de packages et Packages délimités](#).

Interface utilisateur correspondante : onglet Configuration/Packages/Ajouter une configuration/NPM/Ajouter un référentiel de paquets/Scopes - facultatif

## Référence d'action « Amazon S3 publish »

Vous trouverez ci-dessous la référence YAML de définition d'action pour l'action de publication Amazon S3. Pour savoir comment utiliser cette action, consultez [Ajout de l'action « Amazon S3 publish »](#).

### Note

La plupart des propriétés YAML suivantes ont des éléments d'interface utilisateur correspondants dans l'éditeur visuel. Pour rechercher un élément de l'interface utilisateur, utilisez Ctrl+F. L'élément sera répertorié avec sa propriété YAML associée.

```
# The workflow definition starts here.
# See Propriétés de haut niveau for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
S3Publish\_nn:
  Identifiant: aws/s3-publish@v1
  DependsOn:
    - build-action
  Compute:
    Type: EC2 | Lambda
```

```
Fleet: fleet-name  
Timeout: timeout-minutes  
Inputs:  
  Sources:  
    - source-name-1  
  Artifacts:  
    - artifact-name  
  Variables:  
    - Name: variable-name-1  
      Value: variable-value-1  
    - Name: variable-name-2  
      Value: variable-value-2  
Environment:  
  Name: environment-name  
  Connections:  
    - Name: account-connection-name  
      Role: iam-role-name  
Configuration:  
  SourcePath: my/source  
  DestinationBucketName: s3-bucket-name  
  TargetPath: my/target
```

## S3Publish

(Obligatoire)

Spécifiez le nom de l'action. Tous les noms d'action doivent être uniques dans le flux de travail. Les noms d'action sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms d'action.

Par défaut: S3Publish\_nn.

Interface utilisateur correspondante : onglet Configuration/nom de l'action

## Identifiant

(*S3Publish*/Identifiant)

(Obligatoire)

Identifie l'action. Ne modifiez pas cette propriété, sauf si vous souhaitez modifier la version. Pour de plus amples informations, veuillez consulter [Utilisation des versions d'action](#).

Par défaut: `aws/s3-publish@v1`.

Interface utilisateur correspondante : diagramme de flux de travail/ `S3Publish _nn/ aws/s3-publish @v1` label

## DependsOn

(*S3Publish*/DependsOn)

(Facultatif)

Spécifiez une action ou un groupe d'actions qui doit être exécuté correctement pour que cette action soit exécutée.

Pour plus d'informations sur la fonctionnalité « dépend », consultez. [Configuration des actions pour qu'elles dépendent d'autres actions](#)

Interface utilisateur correspondante : onglet Entrées/dépend de - facultatif

## Compute

(*S3Publish*/Compute)

(Facultatif)

Le moteur informatique utilisé pour exécuter les actions de votre flux de travail. Vous pouvez spécifier le calcul au niveau du flux de travail ou au niveau de l'action, mais pas les deux. Lorsqu'elle est spécifiée au niveau du flux de travail, la configuration de calcul s'applique à toutes les actions définies dans le flux de travail. Au niveau du flux de travail, vous pouvez également exécuter plusieurs actions sur la même instance. Pour de plus amples informations, veuillez consulter [Partage du calcul entre les actions](#).

Interface utilisateur correspondante : aucune

## Type

(*S3Publish*/Compute/Type)

(Obligatoire s'[Compute](#) est inclus)

Type de moteur de calcul. Vous pouvez utiliser l'une des valeurs suivantes :

- EC2 (éditeur visuel) ou EC2 (éditeur YAML)

Optimisé pour la flexibilité lors des courses d'action.

- Lambda (éditeur visuel) ou Lambda (éditeur YAML)

Vitesses de démarrage des actions optimisées.

Pour plus d'informations sur les types de calcul, veuillez consulter [À propos des types de calcul](#).

Interface utilisateur correspondante : onglet Configuration/Type de calcul

## Fleet

(*S3Publish*/Compute/Fleet)

(Facultatif)

Spécifiez la machine ou le parc qui exécutera votre flux de travail ou vos actions de flux de travail. Dans le cas des flottes à la demande, lorsqu'une action démarre, le flux de travail fournit les ressources dont il a besoin et les machines sont détruites à la fin de l'action. Exemples de flottes à la demande : `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Pour plus d'informations sur les flottes à la demande, consultez [Propriétés de flotte à la demande](#).

Avec les flottes provisionnées, vous configurez un ensemble de machines dédiées pour exécuter les actions de votre flux de travail. Ces machines restent inactives, prêtes à exécuter des actions immédiatement. Pour plus d'informations sur les flottes provisionnées, consultez [Propriétés du parc provisionné](#)

S'il Fleet est omis, la valeur par défaut est `Linux.x86-64.Large`.

Interface utilisateur correspondante : onglet Configuration/parc de calcul

## Timeout

(*S3Publish*/Timeout)

(Obligatoire)

Spécifiez la durée en minutes (éditeur YAML) ou en heures et minutes (éditeur visuel) pendant laquelle l'action peut être exécutée avant la CodeCatalyst fin de l'action. Le minimum est de 5 minutes et le maximum est décrit dans [Quotas pour les flux de travail dans CodeCatalyst](#). Le délai d'expiration par défaut est le même que le délai d'expiration maximal.

Interface utilisateur correspondante : onglet Configuration/Délai d'expiration - facultatif

## Inputs

(*S3Publish*/Inputs)

(Facultatif)

La Inputs section définit les données dont ils ont S3Publish besoin lors de l'exécution d'un flux de travail.

### Note

Un maximum de quatre entrées (une source et trois artefacts) sont autorisées pour chaque action de AWS CDK déploiement. Les variables ne sont pas prises en compte dans ce total.

Si vous devez faire référence à des fichiers résidant dans différentes entrées (par exemple, une source et un artefact), l'entrée source est l'entrée principale et l'artefact est l'entrée secondaire. Les références aux fichiers dans les entrées secondaires utilisent un préfixe spécial pour les distinguer du fichier principal. Pour plus de détails, consultez [Exemple : Référencement de fichiers dans plusieurs artefacts](#).

Interface utilisateur correspondante : onglet Entrées

## Sources

(*S3Publish*/Inputs/Sources)

(Obligatoire si les fichiers que vous souhaitez publier sur Amazon S3 sont stockés dans un référentiel source)

Si les fichiers que vous souhaitez publier sur Amazon S3 sont stockés dans un référentiel source, spécifiez l'étiquette de ce référentiel source. Actuellement, la seule étiquette prise en charge est `estWorkflowSource`.

Si les fichiers que vous souhaitez publier sur Amazon S3 ne figurent pas dans un référentiel source, ils doivent résider dans un artefact généré par une autre action.

Pour plus d'informations sur les sources, consultez [Travailler avec des sources](#).

Interface utilisateur correspondante : onglet Entrées/Sources - facultatif

## Artifacts - input

(*S3Publish*/Inputs/Artifacts)

(Obligatoire si les fichiers que vous souhaitez publier sur Amazon S3 sont stockés dans un [artefact de sortie](#) issu d'une action précédente)

Si les fichiers que vous souhaitez publier sur Amazon S3 sont contenus dans un artefact généré par une action précédente, spécifiez cet artefact ici. Si vos fichiers ne sont pas contenus dans un artefact, ils doivent se trouver dans votre référentiel source.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

Interface utilisateur correspondante : onglet Configuration/Artefacts - facultatif

## Variables - input

(*S3Publish*/Inputs/Variables)

(Facultatif)

Spécifiez une séquence de paires nom/valeur qui définissent les variables d'entrée que vous souhaitez mettre à la disposition de l'action. Les noms de variables sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms de variables.

Pour plus d'informations sur les variables, y compris des exemples, consultez [Utilisation des variables](#).

Interface utilisateur correspondante : onglet Entrées/Variables - facultatif

## Environnement

(*S3Publish*/Environment)

(Obligatoire)

Spécifiez l' CodeCatalyst environnement à utiliser avec l'action.

Pour plus d'informations sur les environnements, reportez-vous [Utilisation des environnements](#) aux sections et [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/'Environnement/Connection/Role'/  
Environnement

## Name

(*S3Publish*/Environment/Name)

(Obligatoire s'[Environnement](#)il est inclus)

Spécifiez le nom de l'environnement existant que vous souhaitez associer à l'action.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Connection/Role'/  
Environnement

## Connections

(*S3Publish*/Environment/Connections)

(Obligatoire s'[Environnement](#)il est inclus)

Spécifiez la connexion au compte à associer à l'action. Vous pouvez spécifier un maximum d'une connexion à un compte sous `Environment`.

Pour plus d'informations sur les connexions aux comptes, consultez [Administration Comptes AWS d'un espace](#). Pour plus d'informations sur la façon d'associer une connexion à un compte à votre environnement, consultez [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/'Environnement/Connection/Role'/  
Connection

## Name

(*S3Publish*/Environment/Connections/Name)

(Obligatoire)

Spécifiez le nom de la connexion au compte.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Connection/Role'/  
Connection

## Role

(*S3Publish*/Environment/Connections/Role)

(Obligatoire)

Spécifiez le nom du rôle IAM que l'action de publication Amazon S3 utilise pour accéder aux fichiers AWS et les copier sur Amazon S3. Assurez-vous que ce rôle inclut :

- La politique d'autorisation suivante :

### Warning

Limitez les autorisations à celles indiquées dans la politique suivante. L'utilisation d'un rôle doté d'autorisations plus étendues peut présenter un risque de sécurité.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:ListBucket",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

- La politique de confiance personnalisée suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```



```
    "Sid": "",
    "Effect": "Allow",
    "Principal": {
      "Service": [
        "codecatalyst-runner.amazonaws.com",
        "codecatalyst.amazonaws.com"
      ]
    },
    "Action": "sts:AssumeRole"
  }
]
```

Assurez-vous que ce rôle est associé à la connexion de votre compte. Pour en savoir plus sur l'association d'un rôle IAM à une connexion à un compte, consultez [Ajouter des rôles IAM aux connexions aux comptes](#).

#### Note

Vous pouvez indiquer le nom du `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle ici, si vous le souhaitez. Pour plus d'informations sur ce rôle, consultez [Création du `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle pour votre compte et votre espace](#). Sachez que le `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle dispose d'autorisations très étendues, ce qui peut présenter un risque de sécurité. Nous vous recommandons de n'utiliser ce rôle que dans les didacticiels et les scénarios où la sécurité est moins préoccupante.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Connection/Rôle'/ Rôle

## Configuration

(*S3Publish*/Configuration)

(Obligatoire)

Section dans laquelle vous pouvez définir les propriétés de configuration de l'action.

Interface utilisateur correspondante : onglet Configuration

## SourcePath

(*S3Publish*/Configuration/SourcePath)

(Obligatoire)

Spécifiez le nom et le chemin d'un répertoire ou d'un fichier que vous souhaitez publier sur Amazon S3. Le répertoire ou le fichier peut résider dans un référentiel source ou dans un artefact issu d'une action précédente, et est relatif au référentiel source ou à la racine de l'artefact.

Exemples :

La spécification `./myFolder/` copie le contenu de `/myFolder` vers Amazon S3 et préserve la structure de répertoire sous-jacente.

Spécifier `./myFolder/myfile.txt` des copies uniquement `myfile.txt` pour Amazon S3. (La structure du répertoire est supprimée.)

Vous ne pouvez pas utiliser de caractères génériques.

### Note

Vous devrez peut-être ajouter un préfixe au répertoire ou au chemin du fichier pour indiquer dans quel artefact ou dans quelle source le trouver. Pour plus d'informations, consultez [Référencement de fichiers dans un référentiel source](#) et [Référencement de fichiers dans un artefact](#).

Interface utilisateur correspondante : onglet Configuration/chemin source

## DestinationBucketName

(*S3Publish*/Configuration/DestinationBucketName)

(Obligatoire)

Spécifiez le nom du compartiment Amazon S3 dans lequel vous souhaitez publier les fichiers.

Interface utilisateur correspondante : onglet Configuration/compartiment de destination - facultatif

## TargetPath

(*S3Publish*/Configuration/TargetPath)

## (Facultatif)

Spécifiez le nom et le chemin du répertoire dans Amazon S3 dans lequel vous souhaitez publier vos fichiers. Si le répertoire n'existe pas, il sera créé. Le chemin du répertoire ne doit pas inclure le nom du bucket.

Exemples :

```
myS3Folder
```

```
./myS3Folder/myS3Subfolder
```

Interface utilisateur correspondante : onglet Configuration/répertoire de destination - facultatif

## référence d'action « AWS CDK bootstrap »

Voici la référence YAML de définition d'action pour l'action AWS CDKbootstrap. Pour savoir comment utiliser cette action, consultez [Ajout de l'action « AWS CDK bootstrap »](#).

### Note

La plupart des propriétés YAML suivantes ont des éléments d'interface utilisateur correspondants dans l'éditeur visuel. Pour rechercher un élément de l'interface utilisateur, utilisez Ctrl+F. L'élément sera répertorié avec sa propriété YAML associée.

```
# The workflow definition starts here.
# See Propriétés de haut niveau for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
CDKBootstrapAction\_nn:
  Identifiant: aws/cdk-bootstrap@v1
  DependsOn:
    - action-name
  Compute:
    Type: EC2 | Lambda
    Fleet: fleet-name
    Timeout: timeout-minutes
```

```
Inputs:  
# Specify a source or an artifact, but not both.  
Sources:  
- source-name-1  
Artifacts:  
- artifact-name  
Outputs:  
Artifacts:  
- Name: cdk_bootstrap_artifacts  
Files:  
- "cdk.out/**/*"  
Environment:  
Name: environment-name  
Connections:  
- Name: account-connection-name  
Role: iam-role-name  
Configuration:  
Region: us-west-2  
CdkCliVersion: version
```

## CDKBootstrapAction

(Obligatoire)

Spécifiez le nom de l'action. Tous les noms d'action doivent être uniques dans le flux de travail. Les noms d'action sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms d'action.

Par défaut: CDKBootstrapAction\_nn.

Interface utilisateur correspondante : onglet Configuration/nom d'affichage de l'action

### Identifiant

(*CDKBootstrapAction*/Identifiant)

(Obligatoire)

Identifie l'action. Ne modifiez pas cette propriété, sauf si vous souhaitez modifier la version. Pour de plus amples informations, veuillez consulter [Utilisation des versions d'action](#).

Par défaut: aws/cdk-bootstrap@v1.

Interface utilisateur correspondante : diagramme de flux de travail/ CDKBootstrapAction \_nn/ aws/ cdk-bootstrap @v1 label

## DependsOn

(*CDKBootstrapAction*/DependsOn)

(Facultatif)

Spécifiez une action ou un groupe d'actions qui doit être exécuté correctement pour que cette action soit exécutée.

Pour plus d'informations sur la fonctionnalité « dépend », consultez. [Configuration des actions pour qu'elles dépendent d'autres actions](#)

Interface utilisateur correspondante : onglet Entrées/dépend de - facultatif

## Compute

(*CDKBootstrapAction*/Compute)

(Facultatif)

Le moteur informatique utilisé pour exécuter les actions de votre flux de travail. Vous pouvez spécifier le calcul au niveau du flux de travail ou au niveau de l'action, mais pas les deux. Lorsqu'elle est spécifiée au niveau du flux de travail, la configuration de calcul s'applique à toutes les actions définies dans le flux de travail. Au niveau du flux de travail, vous pouvez également exécuter plusieurs actions sur la même instance. Pour de plus amples informations, veuillez consulter [Partage du calcul entre les actions](#).

Interface utilisateur correspondante : aucune

## Type

(*CDKBootstrapAction*/Compute/Type)

(Obligatoire s'[Compute](#) est inclus)

Type de moteur de calcul. Vous pouvez utiliser l'une des valeurs suivantes :

- EC2 (éditeur visuel) ou EC2 (éditeur YAML)

Optimisé pour la flexibilité lors des courses d'action.

- Lambda (éditeur visuel) ou Lambda (éditeur YAML)

Vitesses de démarrage des actions optimisées.

Pour plus d'informations sur les types de calcul, veuillez consulter [À propos des types de calcul](#).

Interface utilisateur correspondante : onglet Configuration/Avancé - facultatif/Type de calcul

## Fleet

(*CDKBootstrapAction*/Compute/Fleet)

(Facultatif)

Spécifiez la machine ou le parc qui exécutera votre flux de travail ou vos actions de flux de travail. Dans le cas des flottes à la demande, lorsqu'une action démarre, le flux de travail fournit les ressources dont il a besoin et les machines sont détruites à la fin de l'action. Exemples de flottes à la demande : `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Pour plus d'informations sur les flottes à la demande, consultez [Propriétés de flotte à la demande](#).

Avec les flottes provisionnées, vous configurez un ensemble de machines dédiées pour exécuter les actions de votre flux de travail. Ces machines restent inactives, prêtes à exécuter des actions immédiatement. Pour plus d'informations sur les flottes provisionnées, consultez [Propriétés du parc provisionné](#)

S'il Fleet est omis, la valeur par défaut est `Linux.x86-64.Large`.

Interface utilisateur correspondante : onglet Configuration/Avancé - facultatif/Parc de calcul

## Timeout

(*CDKBootstrapAction*/Timeout)

(Obligatoire)

Spécifiez la durée en minutes (éditeur YAML) ou en heures et minutes (éditeur visuel) pendant laquelle l'action peut être exécutée avant la CodeCatalyst fin de l'action. Le minimum est de 5 minutes et le maximum est décrit dans [Quotas pour les flux de travail dans CodeCatalyst](#). Le délai d'expiration par défaut est le même que le délai d'expiration maximal.

Interface utilisateur correspondante : onglet Configuration/Délai d'expiration - facultatif

## Inputs

(*CDKBootstrapAction*/Inputs)

(Facultatif)

La Inputs section définit les données dont l'action AWS CDKbootstrap a besoin lors de l'exécution d'un flux de travail.

Interface utilisateur correspondante : onglet Entrées

### Note

Une seule entrée (source ou artefact) est autorisée pour chaque action d'AWS CDKamorçage.

## Sources

(*CDKBootstrapAction*/Inputs/Sources)

(Obligatoire si votre AWS CDK application est stockée dans un référentiel source)

Si votre AWS CDK application est stockée dans un référentiel source, spécifiez l'étiquette de ce référentiel source. L'action AWS CDKbootstrap synthétise l'application dans ce référentiel avant de démarrer le processus de démarrage. Actuellement, la seule étiquette de référentiel prise en charge est `WorkflowSource`.

Si votre AWS CDK application n'est pas contenue dans un référentiel source, elle doit résider dans un artefact généré par une autre action.

Pour plus d'informations sur les sources, consultez [Travailler avec des sources](#).

Interface utilisateur correspondante : onglet Entrées/Sources - facultatif

## Artifacts - input

(*CDKBootstrapAction*/Inputs/Artifacts)

(Obligatoire si votre AWS CDK application est stockée dans un [artefact de sortie](#) d'une action précédente)

Si votre AWS CDK application est contenue dans un artefact généré par une action précédente, spécifiez cet artefact ici. L'action AWS CDKbootstrap synthétise l'application contenue dans l'artefact spécifié dans un CloudFormation modèle avant de démarrer le processus d'amorçage. Si votre AWS CDK application n'est pas contenue dans un artefact, elle doit résider dans votre référentiel source.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

Interface utilisateur correspondante : onglet Entrées/Artefacts - facultatif

## Outputs

(*CDKBootstrapAction*/Outputs)

(Facultatif)

Définit les données produites par l'action lors de l'exécution d'un flux de travail.

Interface utilisateur correspondante : onglet Sorties

## Artifacts - output

(*CDKBootstrapAction*/Outputs/Artifacts)

(Facultatif)

Spécifiez les artefacts générés par l'action. Vous pouvez faire référence à ces artefacts en tant qu'entrée dans d'autres actions.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

Interface utilisateur correspondante : onglet Sorties/Artefacts

## Name

(*CDKBootstrapAction*/Outputs/Artifacts/Name)

(Obligatoire s'[Artifacts - output](#) est inclus)

Spécifiez le nom de l'artefact qui contiendra le AWS CloudFormation modèle synthétisé par l'action AWS CDKbootstrap lors de l'exécution. La valeur par défaut est `cdk_bootstrap_artifacts`. Si vous ne spécifiez aucun artefact, l'action synthétise le modèle, mais ne l'enregistre pas dans un



artefact. Envisagez d'enregistrer le modèle synthétisé dans un artefact afin d'en conserver une trace à des fins de test ou de résolution des problèmes.

Interface utilisateur correspondante : onglet Sorties/Artifacts/Ajouter un artefact/Créer un nom d'artefact

## Files

(*CDKBootstrapAction*/Outputs/Artifacts/Files)

(Obligatoire s'[Artifacts - output](#) est inclus)

Spécifiez les fichiers à inclure dans l'artefact. Vous devez spécifier "cdk.out/\*\*/\*" d'inclure le AWS CloudFormation modèle synthétisé de votre AWS CDK application.

### Note

cdk.out est le répertoire par défaut dans lequel les fichiers synthétisés sont enregistrés. Si vous avez spécifié un répertoire de sortie autre que celui cdk.out de votre cdk.json fichier, spécifiez ce répertoire ici au lieu de cdk.out.

Interface utilisateur correspondante : onglet Sorties/Artifacts/Ajouter un artefact/Fichiers produits par build

## Environment

(*CDKBootstrapAction*/Environment)

(Obligatoire)

Spécifiez l' CodeCatalyst environnement à utiliser avec l'action.

Pour plus d'informations sur les environnements, reportez-vous [Utilisation des environnements](#) aux sections et [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/Environnement

## Name

(*CDKBootstrapAction*/Environment/Name)

(Obligatoire s'[Environnement](#)il est inclus)

Spécifiez le nom de l'environnement existant que vous souhaitez associer à l'action.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/  
Environnement

## Connections

(*CDKBootstrapAction*/Environment/Connections)

(Obligatoire s'[Environnement](#)il est inclus)

Spécifiez la connexion au compte à associer à l'action. Vous pouvez spécifier un maximum d'une connexion à un compte sous `Environment`.

Pour plus d'informations sur les connexions aux comptes, consultez [Administration Comptes AWS d'un espace](#). Pour plus d'informations sur la façon d'associer une connexion à un compte à votre environnement, consultez [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/ connexion  
AWS au compte

## Name

(*CDKBootstrapAction*/Environment/Connections/Name)

(Obligatoire)

Spécifiez le nom de la connexion au compte.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/ connexion  
AWS au compte

## Role

(*CDKBootstrapAction*/Environment/Connections/Role)

(Obligatoire)

Spécifiez le nom du rôle IAM que l'action AWS CDKbootstrap utilise pour accéder à la pile bootstrap AWS et l'ajouter. Assurez-vous que ce rôle inclut les politiques suivantes :

**Note**

Les autorisations indiquées dans la politique d'autorisation suivante sont celles requises par la `cdk bootstrap` commande pour effectuer son amorçage. Ces autorisations peuvent changer si la commande `bootstrap` est AWS CDK modifiée.

**Warning**

Utilisez ce rôle uniquement avec l'action `AWS CDKbootstrap`. Il est très permissif et son utilisation avec d'autres actions peut présenter un risque de sécurité.

- La politique d'autorisation suivante :

**Warning**

Limitez les autorisations à celles indiquées dans la politique suivante. L'utilisation d'un rôle doté d'autorisations plus étendues peut présenter un risque de sécurité.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "ssm:GetParameterHistory",
        "ecr:PutImageScanningConfiguration",
        "cloudformation:*",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "ssm:GetParameters",
        "iam:PutRolePolicy",
        "ssm:GetParameter",
        "ssm>DeleteParameters",
        "ecr>DeleteRepository",
        "ssm:PutParameter",

```

```

        "ssm:DeleteParameter",
        "iam:PassRole",
        "ecr:SetRepositoryPolicy",
        "ssm:GetParametersByPath",
        "ecr:DescribeRepositories",
        "ecr:GetLifecyclePolicy"
    ],
    "Resource": [
        "arn:aws:ssm:aws-region:aws-account:parameter/cdk-bootstrap/*",
        "arn:aws:cloudformation:aws-region:aws-account:stack/CDKToolkit/*",
        "arn:aws:ecr:aws-region:aws-account:repository/cdk-*",
        "arn:aws:iam::aws-account:role/cdk-*"
    ]
},
{
    "Sid": "VisualEditor1",
    "Effect": "Allow",
    "Action": [
        "cloudformation:RegisterType",
        "cloudformation:CreateUploadBucket",
        "cloudformation:ListExports",
        "cloudformation:DescribeStackDriftDetectionStatus",
        "cloudformation:SetTypeDefaultVersion",
        "cloudformation:RegisterPublisher",
        "cloudformation:ActivateType",
        "cloudformation:ListTypes",
        "cloudformation:DeactivateType",
        "cloudformation:SetTypeConfiguration",
        "cloudformation:DeregisterType",
        "cloudformation:ListTypeRegistrations",
        "cloudformation:EstimateTemplateCost",
        "cloudformation:DescribeAccountLimits",
        "cloudformation:BatchDescribeTypeConfigurations",
        "cloudformation:CreateStackSet",
        "cloudformation:ListStacks",
        "cloudformation:DescribeType",
        "cloudformation:ListImports",
        "s3:*",
        "cloudformation:PublishType",
        "ecr:CreateRepository",
        "cloudformation:DescribePublisher",
        "cloudformation:DescribeTypeRegistration",
        "cloudformation:TestType",
        "cloudformation:ValidateTemplate",
    ]
}

```

```

        "cloudformation:ListTypeVersions"
    ],
    "Resource": "*"
}
]
}

```

### Note

La première fois que le rôle est utilisé, utilisez le caractère générique suivant dans la déclaration de politique de ressources, puis délimitez la politique avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"

```

- La politique de confiance personnalisée suivante :

```

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

```

Assurez-vous que ce rôle est ajouté à la connexion de votre compte. Pour en savoir plus sur l'ajout d'un rôle IAM à une connexion à un compte, consultez [Ajouter des rôles IAM aux connexions aux comptes](#).

**Note**

Vous pouvez indiquer le nom du `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle ici, si vous le souhaitez. Pour plus d'informations sur ce rôle, consultez [Création du `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle pour votre compte et votre espace](#). Sachez que le `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle dispose d'autorisations très étendues, ce qui peut présenter un risque de sécurité. Nous vous recommandons de n'utiliser ce rôle que dans les didacticiels et les scénarios où la sécurité est moins préoccupante.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Rôle'/ Rôle

## Configuration

(*CDKBootstrapAction*/Configuration)

(Obligatoire)

Section dans laquelle vous pouvez définir les propriétés de configuration de l'action.

Interface utilisateur correspondante : onglet Configuration

## Region

(*CDKBootstrapAction*/Configuration/Region)

(Obligatoire)

Spécifiez le Région AWS lieu dans lequel la pile bootstrap sera déployée. Cette région doit correspondre à celle dans laquelle votre AWS CDK application est déployée. Pour obtenir la liste des codes de région, consultez la section [Points de terminaison régionaux](#).

Interface utilisateur correspondante : onglet Configuration/région

## CdkCliVersion

(*CDKBootstrapAction*/Configuration/CdkCliVersion)

## (Facultatif)

Cette propriété est disponible avec la version 1.0.13 ou ultérieure de l'action de AWS CDKdéploiement, et la version 1.0.8 ou ultérieure de l'action AWS CDKbootstrap.

Spécifiez l'un des éléments suivants :

- Version complète de l'interface de ligne de AWS Cloud Development Kit (AWS CDK) commande (CLI) (également appelée AWS CDK boîte à outils) que vous souhaitez utiliser pour cette action. Exemple: 2.102.1. Pensez à spécifier une version complète pour garantir la cohérence et la stabilité lors de la création et du déploiement de votre application.

Ou

- latest. Envisagez latest de spécifier pour tirer parti des dernières fonctionnalités et correctifs de la CLI CDK.

L'action téléchargera la version spécifiée (ou la dernière version) de la AWS CDK CLI sur l'[image de CodeCatalyst construction](#), puis utilisera cette version pour exécuter les commandes nécessaires au déploiement de votre application CDK ou au démarrage de votre AWS environnement.

Pour obtenir la liste des versions de CLI CDK prises en charge que vous pouvez utiliser, consultez la section [AWS CDK Versions](#).

Si vous omettez cette propriété, l'action utilise une version de AWS CDK CLI par défaut décrite dans l'une des rubriques suivantes :

- [Versions de la CLI CDK utilisées par l'action « AWS CDK déploiement »](#)
- [Versions de la CLI CDK utilisées par l'action « AWS CDK bootstrap »](#)

Interface utilisateur correspondante : onglet Configuration/version de la AWS CDKCLI

## référence d'action « AWS CDK déployer »

Voici la référence YAML de définition d'action pour l'action de AWS CDK déploiement. Pour savoir comment utiliser cette action, consultez [Ajout de l'action « AWS CDK déployer »](#).

### Note

La plupart des propriétés YAML suivantes ont des éléments d'interface utilisateur correspondants dans l'éditeur visuel. Pour rechercher un élément de l'interface utilisateur, utilisez Ctrl+F. L'élément sera répertorié avec sa propriété YAML associée.

```
# The workflow definition starts here.
# See Propriétés de haut niveau for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
CDKDeploy\_nn:
  Identifiant: aws/cdk-deploy@v1
  DependsOn:
    - CDKBootstrap
  Compute:
    Type: EC2 | Lambda
    Fleet: fleet-name
    Timeout: timeout-minutes
  Inputs:
    # Specify a source or an artifact, but not both.
    Sources:
      - source-name-1
    Artifacts:
      - artifact-name
  Outputs:
    Artifacts:
      - Name: cdk_artifact
    Files:
      - "cdk.out/**/*"
  Environment:
    Name: environment-name
  Connections:
    - Name: account-connection-name
      Role: iam-role-name
  Configuration:
    StackName: my-cdk-stack
    Region: us-west-2
```



```
Tags: '{"key1": "value1", "key2": "value2"}'  
Context: '{"key1": "value1", "key2": "value2"}'  
CdkCliVersion: version  
CdkRootPath: directory/cdk.json  
CfnOutputVariables: '["CfnOutputKey1", "CfnOutputKey2", "CfnOutputKey3"]'  
CloudAssemblyRootPath: path-to-cdk.out
```

## CDKDeploy

(Obligatoire)

Spécifiez le nom de l'action. Tous les noms d'action doivent être uniques dans le flux de travail. Les noms d'action sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms d'action.

Par défaut: CDKDeploy\_nn.

Interface utilisateur correspondante : onglet Configuration/nom de l'action

### Identifiant

(*CDKDeploy*/Identifiant)

(Obligatoire)

Identifie l'action. Ne modifiez pas cette propriété, sauf si vous souhaitez modifier la version. Pour plus d'informations, consultez [Utilisation des versions d'action](#).

Par défaut: aws/cdk-deploy@v1.

Interface utilisateur correspondante : diagramme de flux de travail/ CDKDeploy\_nn/ aws/cdk-deploy @v1 label

### DependsOn

(*CDKDeploy*/DependsOn)

(Facultatif)

Spécifiez une action ou un groupe d'actions qui doit s'exécuter correctement pour que l'action de AWS CDK déploiement soit exécutée. Nous vous recommandons de spécifier l'action AWS CDK bootstrap dans la DependsOn propriété, comme suit :

```
CDKDeploy:
  Identifier: aws/cdk-deploy@v1
  DependsOn:
    - CDKBootstrap
```

### Note

Le [bootstrap](#) est une condition préalable obligatoire au déploiement d'une AWS CDK application. Si vous n'incluez pas l'action AWS CDK Bootstrap dans votre flux de travail, vous devez trouver un autre moyen de déployer la pile AWS CDK Bootstrap avant d'exécuter votre action de AWS CDK déploiement. Pour plus d'informations, consultez [Prérequis](#) dans [Ajout de l'action « AWS CDK déployer »](#).

Pour plus d'informations sur la fonctionnalité « dépend », consultez. [Configuration des actions pour qu'elles dépendent d'autres actions](#)

Interface utilisateur correspondante : onglet Entrées/dépend de - facultatif

## Compute

(*CDKDeploy*/Compute)

(Facultatif)

Le moteur informatique utilisé pour exécuter les actions de votre flux de travail. Vous pouvez spécifier le calcul au niveau du flux de travail ou au niveau de l'action, mais pas les deux. Lorsqu'elle est spécifiée au niveau du flux de travail, la configuration de calcul s'applique à toutes les actions définies dans le flux de travail. Au niveau du flux de travail, vous pouvez également exécuter plusieurs actions sur la même instance. Pour plus d'informations, consultez [Partage du calcul entre les actions](#).

Interface utilisateur correspondante : aucune

## Type

(*CDKDeploy*/Compute/Type)

(Obligatoire s'[Compute](#) est inclus)

Type de moteur de calcul. Vous pouvez utiliser l'une des valeurs suivantes :

- EC2 (éditeur visuel) ou EC2 (éditeur YAML)

Optimisé pour la flexibilité lors des courses d'action.

- Lambda (éditeur visuel) ou Lambda (éditeur YAML)

Vitesses de démarrage des actions optimisées.

Pour plus d'informations sur les types de calcul, veuillez consulter [À propos des types de calcul](#).

Interface utilisateur correspondante : onglet Configuration/Avancé - facultatif/Type de calcul

## Fleet

(*CDKDeploy*/Compute/Fleet)

(Facultatif)

Spécifiez la machine ou le parc qui exécutera votre flux de travail ou vos actions de flux de travail. Dans le cas des flottes à la demande, lorsqu'une action démarre, le flux de travail fournit les ressources dont il a besoin et les machines sont détruites à la fin de l'action. Exemples de flottes à la demande :Linux.x86-64.Large,Linux.x86-64.XLarge. Pour plus d'informations sur les flottes à la demande, consultez [Propriétés de flotte à la demande](#).

Avec les flottes provisionnées, vous configurez un ensemble de machines dédiées pour exécuter les actions de votre flux de travail. Ces machines restent inactives, prêtes à exécuter des actions immédiatement. Pour plus d'informations sur les flottes provisionnées, consultez. [Propriétés du parc provisionné](#)

S'il Fleet est omis, la valeur par défaut estLinux.x86-64.Large.

Interface utilisateur correspondante : onglet Configuration/Avancé - facultatif/Parc de calcul

## Timeout

(*CDKDeploy*/Timeout)

(Obligatoire)

Spécifiez la durée en minutes (éditeur YAML) ou en heures et minutes (éditeur visuel) pendant laquelle l'action peut être exécutée avant la CodeCatalyst fin de l'action. Le minimum est de 5

minutes et le maximum est décrit dans [Quotas pour les flux de travail dans CodeCatalyst](#). Le délai d'expiration par défaut est le même que le délai d'expiration maximal.

Interface utilisateur correspondante : onglet Configuration/Délai d'expiration - facultatif

## Inputs

(*CDKDeploy*/Inputs)

(Facultatif)

La `Inputs` section définit les données dont ils ont `CDKDeploy` besoin lors de l'exécution d'un flux de travail.

### Note

Une seule entrée (source ou artefact) est autorisée pour chaque action de AWS CDK déploiement.

Interface utilisateur correspondante : onglet Entrées

## Sources

(*CDKDeploy*/Inputs/Sources)

(Obligatoire si l' AWS CDK application que vous souhaitez déployer est stockée dans un référentiel source)

Si votre AWS CDK application est stockée dans un référentiel source, spécifiez l'étiquette de ce référentiel source. L'action de AWS CDK déploiement synthétise l'application dans ce référentiel avant de démarrer le processus de déploiement. Actuellement, la seule étiquette prise en charge est `WorkflowSource`.

Si votre AWS CDK application n'est pas contenue dans un référentiel source, elle doit résider dans un artefact généré par une autre action.

Pour plus d'informations sur les sources, consultez [Travailler avec des sources](#).

Interface utilisateur correspondante : onglet Entrées/Sources - facultatif

## Artifacts - input

(*CDKDeploy*/Inputs/Artifacts)

(Obligatoire si l' AWS CDK application que vous souhaitez déployer est stockée dans un [artefact de sortie](#) d'une action précédente)

Si votre AWS CDK application est contenue dans un artefact généré par une action précédente, spécifiez cet artefact ici. L'action de AWS CDK déploiement synthétise l'application dans l'artefact spécifié dans un CloudFormation modèle avant de démarrer le processus de déploiement. Si votre AWS CDK application n'est pas contenue dans un artefact, elle doit résider dans votre référentiel source.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

Interface utilisateur correspondante : onglet Entrées/Artifacts - facultatif

## Outputs

(*CDKDeploy*/Outputs)

(Facultatif)

Définit les données produites par l'action lors de l'exécution d'un flux de travail.

Interface utilisateur correspondante : onglet Sorties

## Artifacts - output

(*CDKDeploy*/Outputs/Artifacts)

(Facultatif)

Spécifiez les artefacts générés par l'action. Vous pouvez faire référence à ces artefacts en tant qu'entrée dans d'autres actions.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

Interface utilisateur correspondante : onglet Sorties/Artifacts

## Name

(*CDKDeploy*/Outputs/Artifacts/Name)

(Obligatoire s'[Artifacts - output](#) est inclus)

Spécifiez le nom de l'artefact qui contiendra le AWS CloudFormation modèle synthétisé par l'action de AWS CDK déploiement lors de l'exécution. La valeur par défaut est `cdk_artifact`. Si vous ne spécifiez aucun artefact, l'action synthétise le modèle mais ne l'enregistre pas dans un artefact. Envisagez d'enregistrer le modèle synthétisé dans un artefact pour en conserver une trace à des fins de test ou de débogage.

Interface utilisateur correspondante : onglet Sorties/Artifacts/Ajouter un artefact/Créer un nom d'artefact

## Files

(*CDKDeploy*/Outputs/Artifacts/Files)

(Obligatoire s'[Artifacts - output](#) est inclus)

Spécifiez les fichiers à inclure dans l'artefact. Vous devez spécifier "`cdk.out/**/*`" d'inclure le AWS CloudFormation modèle synthétisé de votre AWS CDK application.

### Note

`cdk.out` est le répertoire par défaut dans lequel les fichiers synthétisés sont enregistrés. Si vous avez spécifié un répertoire de sortie autre que celui `cdk.out` de votre `cdk.json` fichier, spécifiez ce répertoire ici au lieu de `cdk.out`.

Interface utilisateur correspondante : onglet Sorties/Artifacts/Ajouter un artefact/Fichiers produits par build

## Environment

(*CDKDeploy*/Environment)

(Obligatoire)

Spécifiez l' environnement CodeCatalyst à utiliser avec l'action.

Pour plus d'informations sur les environnements, reportez-vous [Utilisation des environnements](#) aux sections et [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Rôle'/ Environnement

## Name

(*CDKDeploy*/Environment/Name)

(Obligatoire s'[Environnement](#)il est inclus)

Spécifiez le nom de l'environnement existant que vous souhaitez associer à l'action.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Rôle'/ Environnement

## Connections

(*CDKDeploy*/Environment/Connections)

(Obligatoire s'[Environnement](#)il est inclus)

Spécifiez la connexion au compte à associer à l'action. Vous pouvez spécifier un maximum d'une connexion à un compte sous `Environment`.

Pour plus d'informations sur les connexions aux comptes, consultez [Administration Comptes AWS d'un espace](#). Pour plus d'informations sur la façon d'associer une connexion à un compte à votre environnement, consultez [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Rôle'/ connexion AWS au compte

## Name

(*CDKDeploy*/Environment/Connections/Name)

(Obligatoire)

Spécifiez le nom de la connexion au compte.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/ connexion AWS au compte

## Role

(*CDKDeploy*/Environment/Connections/Role)

(Obligatoire)

Spécifiez le nom du rôle IAM que l'action de AWS CDK déploiement utilise pour accéder à la pile d' AWS CDK applications AWS et la déployer. Assurez-vous que ce rôle inclut :

- La politique d'autorisation suivante :

### Warning

Limitez les autorisations à celles indiquées dans la politique suivante. L'utilisation d'un rôle doté d'autorisations plus étendues peut présenter un risque de sécurité.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeChangeSet",
        "cloudformation:DescribeStacks",
        "cloudformation:ListStackResources"
      ],
      "Resource": "*"
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::aws-account:role/cdk-*"
    }
  ]
}
```



- La politique de confiance personnalisée suivante :

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

Assurez-vous que ce rôle est ajouté à la connexion de votre compte. Pour en savoir plus sur l'ajout d'un rôle IAM à une connexion à un compte, consultez [Ajouter des rôles IAM aux connexions aux comptes](#).

#### Note

Vous pouvez indiquer le nom du `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle ici, si vous le souhaitez. Pour plus d'informations sur ce rôle, consultez [Création du `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle pour votre compte et votre espace](#). Sachez que le `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle dispose d'autorisations très étendues, ce qui peut présenter un risque de sécurité. Nous vous recommandons de n'utiliser ce rôle que dans les didacticiels et les scénarios où la sécurité est moins préoccupante.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Rôle'/ Rôle

## Configuration

(`CDKDeploy`/Configuration)

(Obligatoire)

Section dans laquelle vous pouvez définir les propriétés de configuration de l'action.

Interface utilisateur correspondante : onglet Configuration

StackName

(*CDKDeploy*/Configuration/StackName)

(Obligatoire)

Le nom de votre pile d' AWS CDK applications, tel qu'il apparaît dans le fichier de point d'entrée du répertoire de bin votre AWS CDK application. L'exemple suivant montre le contenu d'un fichier de TypeScript point d'entrée, le nom de la pile étant *surligné en italique rouge*. Si votre fichier de point d'entrée est rédigé dans une autre langue, il aura un aspect similaire.

```
import * as cdk from 'aws-cdk-lib';
import { CdkWorkshopTypescriptStack } from '../lib/cdk_workshop_typescript-stack';

const app = new cdk.App();
new CdkWorkshopTypescriptStack(app, 'CdkWorkshopTypescriptStack');
```

Vous ne pouvez spécifier qu'une seule pile.

 Tip

Si vous avez plusieurs piles, vous pouvez créer une pile parent avec des piles imbriquées. Vous pouvez ensuite spécifier la pile parent dans cette action pour déployer toutes les piles.

Interface utilisateur correspondante : onglet Configuration/nom de la pile

Region

(*CDKDeploy*/Configuration/Region)

(Obligatoire)

Spécifiez le Région AWS lieu dans lequel la pile AWS CDK d'applications sera déployée. Pour obtenir la liste des codes de région, consultez la section [Points de terminaison régionaux](#).

Interface utilisateur correspondante : onglet Configuration/région

## Tags

(*CDKDeploy*/Configuration/Tags)

(Facultatif)

Spécifiez les balises que vous souhaitez appliquer aux AWS ressources de la pile AWS CDK d'applications. Les balises sont appliquées à la pile elle-même ainsi qu'aux ressources individuelles de la pile. Pour plus d'informations sur le balisage, consultez la section [Balisage](#) dans le guide du AWS Cloud Development Kit (AWS CDK) développeur.

Interface utilisateur correspondante : onglet Configuration/Avancé - facultatif/Tags

## Context

(*CDKDeploy*/Configuration/Context)

(Facultatif)

Spécifiez les contextes, sous forme de paires clé-valeur, à associer à la pile d' AWS CDK applications. Pour plus d'informations sur les contextes, consultez la section [Contextes d'exécution](#) dans le Guide du AWS Cloud Development Kit (AWS CDK) développeur.

Interface utilisateur correspondante : onglet Configuration/Avancé - facultatif/Contexte

## CdkCliVersion

(*CDKDeploy*/Configuration/CdkCliVersion)

(Facultatif)

Cette propriété est disponible avec la version 1.0.13 ou ultérieure de l'action de AWS CDK déploiement, et la version 1.0.8 ou ultérieure de l'action AWS CDK bootstrap.

Spécifiez l'un des éléments suivants :

- Version complète de l'interface de ligne de AWS Cloud Development Kit (AWS CDK) commande (CLI) (également appelée AWS CDK boîte à outils) que vous souhaitez utiliser pour cette action.

Exemple: 2.102.1. Pensez à spécifier une version complète pour garantir la cohérence et la stabilité lors de la création et du déploiement de votre application.

Ou

- latest. Envisagez latest de spécifier pour tirer parti des dernières fonctionnalités et correctifs de la CLI CDK.

L'action téléchargera la version spécifiée (ou la dernière version) de la AWS CDK CLI sur l'[image de CodeCatalyst construction](#), puis utilisera cette version pour exécuter les commandes nécessaires au déploiement de votre application CDK ou au démarrage de votre AWS environnement.

Pour obtenir la liste des versions de CLI CDK prises en charge que vous pouvez utiliser, consultez la section [AWS CDK Versions](#).

Si vous omettez cette propriété, l'action utilise une version de AWS CDK CLI par défaut décrite dans l'une des rubriques suivantes :

- [Versions de la CLI CDK utilisées par l'action « AWS CDK déploiement »](#)
- [Versions de la CLI CDK utilisées par l'action « AWS CDK bootstrap »](#)

Interface utilisateur correspondante : onglet Configuration/version de la AWS CDK CLI

## CdkRootPath

(*CDKDeploy*/Configuration/CdkRootPath)

(Facultatif)

Le chemin d'accès au répertoire qui contient le cdk.json fichier de votre AWS CDK projet. L'action de AWS CDK déploiement s'exécute à partir de ce dossier, et toutes les sorties créées par l'action seront ajoutées à ce répertoire. Si elle n'est pas spécifiée, l'action de AWS CDK déploiement suppose que le cdk.json fichier se trouve à la racine de votre AWS CDK projet.

Interface utilisateur correspondante : onglet de configuration/répertoire où réside le fichier cdk.json

## CfnOutputVariables

(*CDKDeploy*/Configuration/CfnOutputVariables)

## (Facultatif)

Spécifiez `CfnOutput` les constructions du code de votre AWS CDK application que vous souhaitez exposer en tant que variables de sortie du flux de travail. Vous pouvez ensuite référencer les variables de sortie du flux de travail dans les actions suivantes de votre flux de travail. Pour plus d'informations sur les variables dans CodeCatalyst, consultez [Utilisation des variables](#).

Par exemple, si le code de votre AWS CDK application ressemble à ceci :

```
import { Duration, Stack, StackProps, CfnOutput, RemovalPolicy } from 'aws-cdk-lib';
import * as dynamodb from 'aws-cdk-lib/aws-dynamodb';
import * as s3 from 'aws-cdk-lib/aws-s3';
import { Construct } from 'constructs';
import * as cdk from 'aws-cdk-lib';
export class HelloCdkStack extends Stack {
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);
    const bucket = new s3.Bucket(this, 'my-bucket', {
      removalPolicy: RemovalPolicy.DESTROY,
    });
    new CfnOutput(this, 'bucketName', {
      value: bucket.bucketName,
      description: 'The name of the s3 bucket',
      exportName: 'myBucket',
    });
    const table = new dynamodb.Table(this, 'todos-table', {
      partitionKey: { name: 'todoId', type: dynamodb.AttributeType.NUMBER },
      billingMode: dynamodb.BillingMode.PAY_PER_REQUEST,
      removalPolicy: RemovalPolicy.DESTROY,
    })
    new CfnOutput(this, 'tableName', {
      value: table.tableName,
      description: 'The name of the dynamodb table',
      exportName: 'myDynamoDbTable',
    });
    ...
  }
}
```

... et votre `CfnOutputVariables` propriété ressemble à ceci :

Configuration:

...

```
CfnOutputVariables: ['bucketName','tableName']
```

... puis l'action génère les variables de sortie du flux de travail suivantes :

| Clé        | Valeur            |
|------------|-------------------|
| bucketName | bucket.bucketName |
| tableName  | table.tableName   |

Vous pouvez ensuite référencer les `tableName` variables `bucketName` et dans les actions suivantes. Pour savoir comment référencer les variables de sortie du flux de travail dans les actions suivantes, voir [Référencement d'une variable prédéfinie](#).

Si vous ne spécifiez aucune `CfnOutput` construction dans la `CfnOutputVariables` propriété, l'action expose les quatre premières variables de sortie (ou moins) qu'elle trouve en tant que variables de CloudFormation sortie du flux de travail. Pour plus d'informations, consultez [variables d'action « AWS CDK déployer »](#).

#### Tip

Pour obtenir une liste de toutes les variables de CloudFormation sortie produites par l'action, exécutez le flux de travail contenant l'action de AWS CDK déploiement une fois, puis consultez l'onglet Logs de l'action. Les journaux contiennent une liste de toutes les variables CloudFormation de sortie associées à votre AWS CDK application. Une fois que vous connaissez toutes les CloudFormation variables, vous pouvez spécifier celles que vous souhaitez convertir en variables de sortie de flux de travail à l'aide de la `CfnOutputVariables` propriété.

Pour plus d'informations sur les variables de AWS CloudFormation sortie, consultez la documentation de la `CfnOutput` construction, disponible sous la rubrique [class CfnOutput \(construct\)](#) dans la référence de l'AWS Cloud Development Kit (AWS CDK) API.

Interface utilisateur correspondante : onglet Configuration/variables AWS CloudFormation de sortie

CloudAssemblyRootPath

(*CDKDeploy*/Configuration/CloudAssemblyRootPath)

## (Facultatif)

Si vous avez déjà synthétisé la pile de votre AWS CDK application dans un assemblage cloud (à l'aide de `cdk synth`), spécifiez le chemin racine du répertoire d'assemblage cloud (`cdk.out`). Le AWS CloudFormation modèle situé dans le répertoire d'assemblage cloud spécifié sera déployé par l'action de AWS CDK déployement dans votre ordinateur à Compte AWS l'aide de la `cdk deploy --app` commande. Lorsque l'`--app` option est présente, l'`cdk synth` opération ne se produit pas.

Si vous ne spécifiez aucun répertoire d'assemblage cloud, l'action de AWS CDK déployement exécutera la `cdk deploy` commande sans l'`--app` option. Sans `--app` cette option, l'`cdk deploy` opération synthétisera (`cdk synth`) et déploiera votre AWS CDK application dans votre Compte AWS.

Pourquoi spécifier un assemblage cloud synthétisé existant alors que l'action « AWS CDK déployer » peut effectuer la synthèse au moment de l'exécution ?

Vous souhaitez peut-être spécifier un assemblage cloud synthétisé existant pour :

- Assurez-vous que le même ensemble de ressources est déployé à chaque fois que l'action « AWS CDK déployer » est exécutée

Si vous ne spécifiez pas d'assemblage cloud, il est possible que l'action de AWS CDK déployement synthétise et déploie différents fichiers en fonction du moment où elle est exécutée. Par exemple, l'action de AWS CDK déployement peut synthétiser un assemblage cloud avec un ensemble de dépendances pendant une phase de test et un autre ensemble de dépendances pendant une phase de production (si ces dépendances ont changé entre les étapes). Pour garantir une parité exacte entre ce qui est testé et ce qui est déployé, nous recommandons de synthétiser une seule fois, puis d'utiliser le champ `Path to cloud assembly directory` (éditeur visuel) ou la `CloudAssemblyRootPath` propriété (éditeur YAML) pour spécifier l'assembly cloud déjà synthétisé.

- Utilisez des gestionnaires de packages et des outils non standard avec l'application AWS CDK

Au cours d'une `synth` opération, l'action de AWS CDK déployement tente d'exécuter votre application à l'aide d'outils standard tels que `npm` ou `pip`. Si l'action ne parvient pas à exécuter correctement votre application à l'aide de ces outils, la synthèse n'aura pas lieu et l'action échouera. Pour contourner ce problème, vous pouvez spécifier les commandes exactes nécessaires pour exécuter correctement votre application dans le `cdk.json` fichier de l'AWS CDK application, puis synthétiser votre application à l'aide d'une méthode qui n'implique pas l'action de

AWS CDK déploiement. Une fois l'assembly cloud généré, vous pouvez le spécifier dans le champ Path to cloud assembly directory (éditeur visuel) ou dans la CloudAssemblyRootPath propriété (éditeur YAML) de l'action de AWS CDK déploiement.

Pour plus d'informations sur la configuration du `cdk.json` fichier afin d'inclure les commandes d'installation et d'exécution de votre AWS CDK application, consultez la section [Spécification de la commande de l'application](#).

Pour plus d'informations sur les `cdk synth` commandes `cdk deploy` et, ainsi que sur l'option, voir [Déployer des piles](#), [synthétiser des piles](#) et [ignorer la synthèse dans le guide](#) du développeur.AWS Cloud Development Kit (AWS CDK)

Pour plus d'informations sur les assemblages cloud, consultez [Cloud Assembly](#) dans la référence des AWS Cloud Development Kit (AWS CDK) API.

Interface utilisateur correspondante : onglet Configuration/Chemin d'accès au répertoire d'assemblage du cloud

## référence d'action « AWS Lambda invoquer »

Voici la référence YAML de définition d'action pour l'action d'AWS Lambda appel. Pour savoir comment utiliser cette action, consultez [Ajouter l'action « AWS Lambda invoquer »](#).

### Note

La plupart des propriétés YAML suivantes ont des éléments d'interface utilisateur correspondants dans l'éditeur visuel. Pour rechercher un élément de l'interface utilisateur, utilisez Ctrl+F. L'élément sera répertorié avec sa propriété YAML associée.

```
# The workflow definition starts here.  
# See Propriétés de haut niveau for details.
```

```
Name: MyWorkflow  
SchemaVersion: 1.0  
Actions:
```

```
# The action definition starts here.  
LambdaInvoke\_nn:
```



```

Identifiant: aws/lambda-invoke@v1
DependsOn:
  - dependent-action
Compute:
  Type: EC2 | Lambda
  Fleet: fleet-name
Timeout: timeout-minutes
Inputs:
  # Specify a source or an artifact, but not both.
  Sources:
    - source-name-1
  Artifacts:
    - request-payload
  Variables:
    - Name: variable-name-1
      Value: variable-value-1
    - Name: variable-name-2
      Value: variable-value-2
Environment:
  Name: environment-name
Connections:
  - Name: account-connection-name
    Role: iam-role-name
Configuration:
  Function: my-function/function-arn
  AWSRegion: us-west-2
  # Specify RequestPayload or RequestPayloadFile, but not both.
  RequestPayload: '{"firstname": "Li", lastname: "Jean", "company": "ExampleCo",
"team": "Development"}'
  RequestPayloadFile: my/request-payload.json
  ContinueOnError: true/false
  LogType: Tail/None
  ResponseFilters: '{"name": ".name", "company": ".department.company"}'
Outputs:
  Artifacts:
    - Name: lambda_artifacts
      Files:
        - "lambda-response.json"

```

## LambdaInvoke

(Obligatoire)

Spécifiez le nom de l'action. Tous les noms d'action doivent être uniques dans le flux de travail. Les noms d'action sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms d'action.

Par défaut: `Lambda_Invoke_Action_Workflow_nn`.

Interface utilisateur correspondante : onglet Configuration/nom de l'action

## Identifieur

(*LambdaInvoke*/Identifieur)

(Obligatoire)

Identifie l'action. Ne modifiez pas cette propriété, sauf si vous souhaitez modifier la version. Pour de plus amples informations, veuillez consulter [Utilisation des versions d'action](#).

Par défaut: `aws/lambda-invoke@v1`.

Interface utilisateur correspondante : diagramme de flux de travail/ `LambdaInvoke_nn`/ `aws/lambda-invoke @v1` label

## DependsOn

(*LambdaInvoke*/DependsOn)

(Facultatif)

Spécifiez une action ou un groupe d'actions qui doit être exécuté correctement pour que cette action soit exécutée.

Pour plus d'informations sur la fonctionnalité « dépend », consultez. [Configuration des actions pour qu'elles dépendent d'autres actions](#)

Interface utilisateur correspondante : onglet Entrées/dépend de - facultatif

## Compute

(*LambdaInvoke*/Compute)

## (Facultatif)

Le moteur informatique utilisé pour exécuter les actions de votre flux de travail. Vous pouvez spécifier le calcul au niveau du flux de travail ou au niveau de l'action, mais pas les deux. Lorsqu'elle est spécifiée au niveau du flux de travail, la configuration de calcul s'applique à toutes les actions définies dans le flux de travail. Au niveau du flux de travail, vous pouvez également exécuter plusieurs actions sur la même instance. Pour de plus amples informations, veuillez consulter [Partage du calcul entre les actions](#).

Interface utilisateur correspondante : aucune

## Type

(*LambdaInvoke*/Compute/Type)

(Obligatoire s'[Compute](#) est inclus)

Type de moteur de calcul. Vous pouvez utiliser l'une des valeurs suivantes :

- EC2 (éditeur visuel) ou EC2 (éditeur YAML)  
Optimisé pour la flexibilité lors des courses d'action.
- Lambda (éditeur visuel) ou Lambda (éditeur YAML)  
Vitesses de démarrage des actions optimisées.

Pour plus d'informations sur les types de calcul, veuillez consulter [À propos des types de calcul](#).

Interface utilisateur correspondante : onglet Configuration/Type de calcul

## Fleet

(*LambdaInvoke*/Compute/Fleet)

(Facultatif)

Spécifiez la machine ou le parc qui exécutera votre flux de travail ou vos actions de flux de travail. Dans le cas des flottes à la demande, lorsqu'une action démarre, le flux de travail fournit les ressources dont il a besoin et les machines sont détruites à la fin de l'action. Exemples de flottes à la demande :Linux.x86-64.Large,Linux.x86-64.XLarge. Pour plus d'informations sur les flottes à la demande, consultez[Propriétés de flotte à la demande](#).

Avec les flottes provisionnées, vous configurez un ensemble de machines dédiées pour exécuter les actions de votre flux de travail. Ces machines restent inactives, prêtes à exécuter des actions immédiatement. Pour plus d'informations sur les flottes provisionnées, consultez. [Propriétés du parc provisionné](#)

S'il Fleet est omis, la valeur par défaut est `Linux.x86-64.Large`.

Interface utilisateur correspondante : onglet Configuration/parc de calcul

## Timeout

(*LambdaInvoke*/Timeout)

(Obligatoire)

Spécifiez la durée en minutes (éditeur YAML) ou en heures et minutes (éditeur visuel) pendant laquelle l'action peut être exécutée avant la CodeCatalyst fin de l'action. Le minimum est de 5 minutes et le maximum est décrit dans [Quotas pour les flux de travail dans CodeCatalyst](#). Le délai d'expiration par défaut est le même que le délai d'expiration maximal.

Interface utilisateur correspondante : onglet Configuration/Délai d'expiration - facultatif

## Inputs

(*LambdaInvoke*/Inputs)

(Obligatoire)

La `Inputs` section définit les données dont l'action d'AWS Lambda appelle a besoin lors de l'exécution d'un flux de travail.

### Note

Une seule entrée (source ou artefact) est autorisée par action d'AWS Lambda appel. Les variables ne sont pas prises en compte dans ce total.

Interface utilisateur correspondante : onglet Entrées

## Sources

(*LambdaInvoke*/Inputs/Sources)

(Obligatoire s'[RequestPayloadFile](#)il est fourni)

Si vous souhaitez transmettre un fichier JSON de charge utile de demande à l'action AWS Lambda d'appel et que ce fichier de charge utile est stocké dans un référentiel source, spécifiez l'étiquette de ce référentiel source. Actuellement, la seule étiquette prise en charge est `WorkflowSource`.

Si le fichier de charge utile de votre demande n'est pas contenu dans un référentiel source, il doit résider dans un artefact généré par une autre action.

Pour plus d'informations sur le fichier de charge utile, consultez [RequestPayloadFile](#).

#### Note

Au lieu de spécifier un fichier de charge utile, vous pouvez ajouter le code JSON de la charge utile directement à l'action à l'aide de la `RequestPayload` propriété. Pour de plus amples informations, veuillez consulter [RequestPayload](#).

Pour plus d'informations sur les sources, consultez [Travailler avec des sources](#).

Interface utilisateur correspondante : onglet Entrées/Sources - facultatif

## Artifacts - input

(*LambdaInvoke*/Inputs/Artifacts)

(Obligatoire s'[RequestPayloadFile](#)il est fourni)

Si vous souhaitez transmettre un fichier JSON de charge utile de demande à l'action d'AWS Lambda appel et que ce fichier de charge utile est contenu dans un [artefact de sortie](#) d'une action précédente, spécifiez cet artefact ici.

Pour plus d'informations sur le fichier de charge utile, consultez [RequestPayloadFile](#).

#### Note

Au lieu de spécifier un fichier de charge utile, vous pouvez ajouter le code JSON de la charge utile directement à l'action à l'aide de la `RequestPayload` propriété. Pour de plus amples informations, veuillez consulter [RequestPayload](#).

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

Interface utilisateur correspondante : onglet Configuration/Artefacts - facultatif

## Variables - input

(*LambdaInvoke*/Inputs/Variables)

(Facultatif)

Spécifiez une séquence de paires nom/valeur qui définissent les variables d'entrée que vous souhaitez mettre à la disposition de l'action. Les noms de variables sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms de variables.

Pour plus d'informations sur les variables, y compris des exemples, consultez [Utilisation des variables](#).

Interface utilisateur correspondante : onglet Entrées/Variables - facultatif

## Environnement

(*LambdaInvoke*/Environnement)

(Obligatoire)

Spécifiez l' CodeCatalyst environnement à utiliser avec l'action.

Pour plus d'informations sur les environnements, reportez-vous [Utilisation des environnements](#) aux sections et [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/  
Environnement

## Name

(*LambdaInvoke*/Environnement/Name)

(Obligatoire s'[Environnement](#)il est inclus)

Spécifiez le nom de l'environnement existant que vous souhaitez associer à l'action.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Connection/Role'/  
Environnement

## Connections

(*LambdaInvoke*/Environment/Connections)

(Obligatoire s'[Environnement](#) est inclus)

Spécifiez la connexion au compte à associer à l'action. Vous pouvez spécifier un maximum d'une connexion à un compte sous `Environment`.

Pour plus d'informations sur les connexions aux comptes, consultez [Administration Comptes AWS d'un espace](#). Pour plus d'informations sur la façon d'associer une connexion à un compte à votre environnement, consultez [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/'Environnement/Connection/Role'/  
Connection

## Name

(*LambdaInvoke*/Environment/Connections/Name)

(Obligatoire)

Spécifiez le nom de la connexion au compte.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Connection/Role'/  
Connection

## Role

(*LambdaInvoke*/Environment/Connections/Role)

(Obligatoire)

Spécifiez le nom du rôle IAM que l'action d'AWS Lambda appel utilise pour accéder à votre fonction Lambda AWS et l'appeler. Assurez-vous que ce rôle inclut :

- La politique d'autorisation suivante :

**⚠ Warning**

Limitez les autorisations à celles indiquées dans la politique suivante. L'utilisation d'un rôle doté d'autorisations plus étendues peut présenter un risque de sécurité.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:aws-region:aws-account:function:function-name"
    }
  ]
}
```

- La politique de confiance personnalisée suivante :

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



Assurez-vous que ce rôle est associé à la connexion de votre compte. Pour en savoir plus sur l'association d'un rôle IAM à une connexion à un compte, consultez [Ajouter des rôles IAM aux connexions aux comptes](#).

#### Note

Vous pouvez indiquer le nom du CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle ici, si vous le souhaitez. Pour plus d'informations sur ce rôle, consultez [Création du CodeCatalystWorkflowDevelopmentRole-\*spaceName\* rôle pour votre compte et votre espace](#). Sachez que le CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle dispose d'autorisations très étendues, ce qui peut présenter un risque de sécurité. Nous vous recommandons de n'utiliser ce rôle que dans les didacticiels et les scénarios où la sécurité est moins préoccupante.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Connexion/Rôle'/ Rôle

## Configuration

(*LambdaInvoke*/Configuration)

(Obligatoire)

Section dans laquelle vous pouvez définir les propriétés de configuration de l'action.

Interface utilisateur correspondante : onglet Configuration

## Function

(*LambdaInvoke*/Configuration/Function)

(Obligatoire)

Spécifiez la AWS Lambda fonction que cette action invoquera. Vous pouvez spécifier le nom de la fonction ou son Amazon Resource Name (ARN). Vous pouvez trouver le nom ou l'ARN dans la console Lambda.

**Note**

Le AWS compte sur lequel réside la fonction Lambda peut être différent du compte spécifié ci-dessous. **Connections :**

Interface utilisateur correspondante : onglet de configuration/fonction

## AWSRegion

(*LambdaInvoke*/Configuration/AWSRegion)

(Obligatoire)

Spécifiez la AWS région dans laquelle réside votre AWS Lambda fonction. Pour obtenir la liste des codes de région, consultez la section [Points de terminaison régionaux](#) dans le Références générales AWS.

Interface utilisateur correspondante : onglet Configuration/compartiment de destination - facultatif

## RequestPayload

(*LambdaInvoke*/Configuration/RequestPayload)

(Facultatif)

Si vous souhaitez transmettre une charge utile de demande à l'action d'AWS Lambda appel, spécifiez la charge utile de la demande ici, au format JSON.

Exemple de charge utile de demande :

```
'{ "key": "value" }'
```

Si vous ne souhaitez pas transmettre de charge utile de requête à votre fonction Lambda, omettez cette propriété.

**Note**

Vous pouvez préciser RequestPayload ou RequestPayloadFile, mais pas les deux.

Pour plus d'informations sur la charge utile des demandes, consultez la rubrique [Invoke](#) dans la référence de l'AWS Lambda API.

Interface utilisateur correspondante : onglet Configuration/Charge utile de demande - facultatif

## RequestPayloadFile

(*LambdaInvoke*/Configuration/RequestPayloadFile)

(Facultatif)

Si vous souhaitez transmettre une charge utile de demande à l'action d'AWS Lambda appel, spécifiez le chemin d'accès à ce fichier de charge utile de demande ici. Le fichier doit être au format JSON.

Le fichier de charge utile de la demande peut résider dans un référentiel source ou dans un artefact issu d'une action précédente. Le chemin du fichier est relatif au référentiel source ou à la racine de l'artefact.

Si vous ne souhaitez pas transmettre de charge utile de requête à votre fonction Lambda, omettez cette propriété.

### Note

Vous pouvez préciser RequestPayload ou RequestPayloadFile, mais pas les deux.

Pour plus d'informations sur le fichier de charge utile de la demande, consultez la rubrique [Invoke](#) dans la référence de l'AWS Lambda API.

Interface utilisateur correspondante : onglet Configuration/fichier de charge utile de demande - facultatif

## ContinueOnError

(*LambdaInvoke*/Configuration/RequestPayloadFile)

(Facultatif)

Spécifiez si vous souhaitez marquer l'action d'AWS Lambda appel comme réussie même si la AWS Lambda fonction invoquée échoue. Envisagez de définir cette propriété sur `true` pour permettre aux actions suivantes de votre flux de travail de démarrer malgré l'échec de Lambda.

Par défaut, l'action échoue en cas d'échec de la fonction Lambda (« off » dans l'éditeur visuel ou `false` dans l'éditeur YAML).

Interface utilisateur correspondante : onglet Configuration/Continuer en cas d'erreur

## LogType

(*LambdaInvoke*/Configuration/LogType)

(Facultatif)

Spécifiez si vous souhaitez inclure les journaux d'erreurs dans la réponse de la fonction Lambda après son appel. Vous pouvez consulter ces journaux dans l'onglet Logs de l'action Lambda Invoke de la CodeCatalyst console. Les valeurs possibles sont :

- `Tail`— journaux des retours
- `None`— ne pas renvoyer les journaux

La valeur par défaut est `Tail`.

Pour plus d'informations sur le type de journal, consultez la rubrique [Invoke](#) dans la référence de l'AWS Lambda API.

Pour de plus amples informations sur l'affichage des journaux, veuillez consulter [Afficher le statut et les détails de l'exécution du flux de travail](#).

Interface utilisateur correspondante : onglet Configuration/type de journal

## ResponseFilters

(*LambdaInvoke*/Configuration/ResponseFilters)

(Facultatif)

Spécifiez les clés de la charge utile de réponse Lambda que vous souhaitez convertir en variables de sortie. Vous pouvez ensuite référencer les variables de sortie dans les actions suivantes de votre flux de travail. Pour plus d'informations sur les variables dans CodeCatalyst, voir [Utilisation des variables](#).

Par exemple, si la charge utile de votre réponse ressemble à ceci :

```
responsePayload = {
```

```

"name": "Saanvi",
"location": "Seattle",
"department": {
  "company": "Amazon",
  "team": "AWS"
}
}

```

... et vos filtres de réponse ressemblent à ceci :

Configuration:

```

...
ResponseFilters: '{"name": ".name", "company": ".department.company"}'

```

... puis l'action génère les variables de sortie suivantes :

| Clé     | Valeur |
|---------|--------|
| name    | Saanvi |
| company | Amazon |

Vous pouvez ensuite référencer les company variables name et dans les actions suivantes.

Si vous ne spécifiez aucune cléResponseFilters, l'action convertit chaque clé de niveau supérieur de la réponse Lambda en variable de sortie. Pour de plus amples informations, veuillez consulter [variables d'action « AWS Lambda invoquer »](#).

Envisagez d'utiliser des filtres de réponse pour limiter les variables de sortie générées à celles que vous souhaitez réellement utiliser.

Interface utilisateur correspondante : onglet Configuration/filtres de réponse - facultatif

## Outputs

(*LambdaInvoke*/Outputs)

(Facultatif)

Définit les données produites par l'action lors de l'exécution d'un flux de travail.

Interface utilisateur correspondante : onglet Sorties

## Artifacts

(*LambdaInvoke*/Outputs/Artifacts)

(Facultatif)

Spécifiez les artefacts générés par l'action. Vous pouvez faire référence à ces artefacts en tant qu'entrée dans d'autres actions.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

Interface utilisateur correspondante : onglet Sorties/Artifacts/ Nom de l'artefact de construction

## Name

(*LambdaInvoke*/Outputs/Artifacts/Name)

(Facultatif)

Spécifiez le nom de l'artefact qui contiendra la charge utile de réponse Lambda renvoyée par la fonction Lambda. La valeur par défaut est `lambda_artifacts`. Si vous ne spécifiez aucun artefact, la charge utile de la réponse Lambda peut être consultée dans les journaux de l'action, qui sont disponibles dans l'onglet Logs de l'action dans la console. CodeCatalyst Pour de plus amples informations sur l'affichage des journaux, veuillez consulter [Afficher le statut et les détails de l'exécution du flux de travail](#).

Interface utilisateur correspondante : onglet Sorties/Artifacts/ Nom de l'artefact de construction

## Files

(*LambdaInvoke*/Outputs/Artifacts/Files)

(Facultatif)

Spécifiez les fichiers à inclure dans l'artefact. Vous devez spécifier de `lambda-response.json` telle sorte que le fichier de charge utile de la réponse Lambda soit inclus.

Interface utilisateur correspondante : onglet Sorties/Artifacts/Fichiers produits par build

## Référence d'action « Deploy AWS CloudFormation Stack »

Voici la référence YAML de définition d'action pour l'action Deploy AWS CloudFormation stack. Pour savoir comment utiliser cette action, consultez [Ajout de l'action « Deploy AWS CloudFormation stack »](#).

### Note

La plupart des propriétés YAML suivantes ont des éléments d'interface utilisateur correspondants dans l'éditeur visuel. Pour rechercher un élément de l'interface utilisateur, utilisez Ctrl+F. L'élément sera répertorié avec sa propriété YAML associée.

```
# The workflow definition starts here.
# See Propriétés de haut niveau for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
DeployCloudFormationStack:
  Identifiant: aws/cfn-deploy@v1
  DependsOn:
    - build-action
  Compute:
    Type: EC2 | Lambda
    Fleet: fleet-name
    Timeout: timeout-minutes
  Environment:
    Name: environment-name
  Connections:
    - Name: account-connection-name
      Role: DeployRole
  Inputs:
    Sources:
      - source-name-1
    Artifacts:
      - CloudFormation-artifact
  Configuration:
    name: stack-name
    region: us-west-2
```

```
template: template-path  
role-arn: arn:aws:iam::123456789012:role/StackRole  
capabilities: CAPABILITY_IAM,CAPABILITY_NAMED_IAM,CAPABILITY_AUTO_EXPAND  
parameter-overrides: KeyOne=ValueOne,KeyTwo=ValueTwo | path-to-JSON-file  
no-execute-changeset: 1|0  
fail-on-empty-changeset: 1|0  
disable-rollback: 1|0  
termination-protection: 1|0  
timeout-in-minutes: minutes  
notification-arns: arn:aws:sns:us-east-1:123456789012:MyTopic,arn:aws:sns:us-east-1:123456789012:MyOtherTopic  
monitor-alarm-arns: arn:aws:cloudwatch::123456789012:alarm/MyAlarm,arn:aws:cloudwatch::123456789012:alarm/MyOtherAlarm  
monitor-timeout-in-minutes: minutes  
tags: '[{"Key": "MyKey1", "Value": "MyValue1"}, {"Key": "MyKey2", "Value": "MyValue2"}]'
```

## DeployCloudFormationStack

(Obligatoire)

Spécifiez le nom de l'action. Tous les noms d'action doivent être uniques dans le flux de travail. Les noms d'action sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms d'action.

Par défaut: DeployCloudFormationStack\_nn.

Interface utilisateur correspondante : onglet Configuration/nom d'affichage de l'action

### Identifiant

(*DeployCloudFormationStack/Identifiant*)

(Obligatoire)

Identifie l'action. Ne modifiez pas cette propriété, sauf si vous souhaitez modifier la version. Pour de plus amples informations, veuillez consulter [Utilisation des versions d'action](#).

Par défaut: aws/cfn-deploy@v1.

Interface utilisateur correspondante : diagramme de flux de travail/ DeployCloudFormationStack\_nn/ aws/cfn-deploy @v1 label



## DependsOn

(*DeployCloudFormationStack*/DependsOn)

(Facultatif)

Spécifiez une action ou un groupe d'actions qui doit être exécuté correctement pour que cette action soit exécutée.

Pour plus d'informations sur la fonctionnalité « dépend », consultez. [Configuration des actions pour qu'elles dépendent d'autres actions](#)

Interface utilisateur correspondante : onglet Entrées/dépend de - facultatif

## Compute

(*DeployCloudFormationStack*/Compute)

(Facultatif)

Le moteur informatique utilisé pour exécuter les actions de votre flux de travail. Vous pouvez spécifier le calcul au niveau du flux de travail ou au niveau de l'action, mais pas les deux. Lorsqu'elle est spécifiée au niveau du flux de travail, la configuration de calcul s'applique à toutes les actions définies dans le flux de travail. Au niveau du flux de travail, vous pouvez également exécuter plusieurs actions sur la même instance. Pour de plus amples informations, veuillez consulter [Partage du calcul entre les actions](#).

Interface utilisateur correspondante : aucune

## Type

(*DeployCloudFormationStack*/Compute/Type)

(Obligatoire s'[Compute](#) est inclus)

Type de moteur de calcul. Vous pouvez utiliser l'une des valeurs suivantes :

- EC2 (éditeur visuel) ou EC2 (éditeur YAML)  
Optimisé pour la flexibilité lors des courses d'action.
- Lambda (éditeur visuel) ou Lambda (éditeur YAML)

Vitesses de démarrage des actions optimisées.

Pour plus d'informations sur les types de calcul, veuillez consulter [À propos des types de calcul](#).

Interface utilisateur correspondante : onglet Configuration/Avancé - facultatif/Type de calcul

## Fleet

*(DeployCloudFormationStack/Compute/Fleet)*

(Facultatif)

Spécifiez la machine ou le parc qui exécutera votre flux de travail ou vos actions de flux de travail. Dans le cas des flottes à la demande, lorsqu'une action démarre, le flux de travail fournit les ressources dont il a besoin et les machines sont détruites à la fin de l'action. Exemples de flottes à la demande : `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Pour plus d'informations sur les flottes à la demande, consultez [Propriétés de flotte à la demande](#).

Avec les flottes provisionnées, vous configurez un ensemble de machines dédiées pour exécuter les actions de votre flux de travail. Ces machines restent inactives, prêtes à exécuter des actions immédiatement. Pour plus d'informations sur les flottes provisionnées, consultez [Propriétés du parc provisionné](#)

S'il Fleet est omis, la valeur par défaut est `Linux.x86-64.Large`.

Interface utilisateur correspondante : onglet Configuration/Avancé - facultatif/Parc de calcul

## Timeout

*(DeployCloudFormationStack/Timeout)*

(Facultatif)

Spécifiez la durée en minutes (éditeur YAML) ou en heures et minutes (éditeur visuel) pendant laquelle l'action peut être exécutée avant la CodeCatalyst fin de l'action. Le minimum est de 5 minutes et le maximum est décrit dans [Quotas pour les flux de travail dans CodeCatalyst](#). Le délai d'expiration par défaut est le même que le délai d'expiration maximal.

Interface utilisateur correspondante : onglet Configuration/Délai d'expiration en minutes - facultatif

## Environment

*(DeployCloudFormationStack/Environment)*

(Obligatoire)

Spécifiez l' CodeCatalyst environnement à utiliser avec l'action.

Pour plus d'informations sur les environnements, reportez-vous [Utilisation des environnements](#) aux sections et [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/  
Environnement

### Name

*(DeployCloudFormationStack/Environment/Name)*

(Obligatoire s'[Environnement](#)il est inclus)

Spécifiez le nom de l'environnement existant que vous souhaitez associer à l'action.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/  
Environnement

### Connections

*(DeployCloudFormationStack/Environment/Connections)*

(Obligatoire s'[Environnement](#)il est inclus)

Spécifiez la connexion au compte à associer à l'action. Vous pouvez spécifier un maximum d'une connexion à un compte sous `Environment`.

Pour plus d'informations sur les connexions aux comptes, consultez [Administration Comptes AWS d'un espace](#). Pour plus d'informations sur la façon d'associer une connexion à un compte à votre environnement, consultez [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/ connexion  
AWS au compte

## Name

(*DeployCloudFormationStack*/Environment/Connections/Name)

(Obligatoire)

Spécifiez le nom de la connexion au compte.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/ connexion AWS au compte

## Role

(*DeployCloudFormationStack*/Environment/Connections/Role)

(Obligatoire)

Spécifiez le nom du rôle IAM que l'action Deploy AWS CloudFormation stack utilise pour accéder au AWS CloudFormation service AWS et à celui-ci.

Assurez-vous que ce rôle inclut les politiques suivantes :

- La politique d'autorisation suivante :

### Warning

Limitez les autorisations à celles indiquées dans la politique suivante. L'utilisation d'un rôle doté d'autorisations plus étendues peut présenter un risque de sécurité.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "cloudformation:CreateStack",
      "cloudformation>DeleteStack",
      "cloudformation:Describe*",
      "cloudformation:UpdateStack",
      "cloudformation:CreateChangeSet",
      "cloudformation>DeleteChangeSet",
      "cloudformation:ExecuteChangeSet",
```

```

        "cloudformation:SetStackPolicy",
        "cloudformation:ValidateTemplate",
        "cloudformation:List*",
        "iam:PassRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
}]
}

```

### Note

La première fois que le rôle est utilisé, utilisez le caractère générique suivant dans la déclaration de politique de ressources, puis délimitez la politique avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

- La politique de confiance personnalisée suivante :

```

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

```

Assurez-vous que ce rôle est ajouté à la connexion au compte dans votre espace. Pour en savoir plus sur l'ajout d'un rôle IAM à une connexion à un compte, consultez [Ajouter des rôles IAM aux connexions aux comptes](#).

**Note**

Vous pouvez indiquer le nom du `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle ici, si vous le souhaitez. Pour plus d'informations sur ce rôle, consultez [Création du `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle pour votre compte et votre espace](#). Sachez que le `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle dispose d'autorisations très étendues, ce qui peut présenter un risque de sécurité. Nous vous recommandons de n'utiliser ce rôle que dans les didacticiels et les scénarios où la sécurité est moins préoccupante.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Rôle'/ Rôle

## Inputs

(*DeployCloudFormationStack/Inputs*)

(Facultatif)

La `Inputs` section définit les données dont ils ont `DeployCloudFormationStack` besoin lors de l'exécution d'un flux de travail.

**Note**

Un maximum de quatre entrées (une source et trois artefacts) sont autorisées par action de `AWS CloudFormationstack Deploy`.

Si vous devez faire référence à des fichiers résidant dans différentes entrées (par exemple, une source et un artefact), l'entrée source est l'entrée principale et l'artefact est l'entrée secondaire. Les références aux fichiers dans les entrées secondaires utilisent un préfixe spécial pour les distinguer du fichier principal. Pour plus de détails, consultez [Exemple : Référencement de fichiers dans plusieurs artefacts](#).

Interface utilisateur correspondante : onglet Entrées

## Sources

(*DeployCloudFormationStack/Inputs/Sources*)

(Obligatoire si votre AWS SAM modèle CloudFormation ou votre modèle est stocké dans un référentiel source)

Si votre AWS SAM modèle CloudFormation ou votre modèle est stocké dans un référentiel source, spécifiez l'étiquette de ce référentiel source. Actuellement, la seule étiquette prise en charge est `WorkflowSource`.

Si votre AWS SAM modèle CloudFormation ou votre modèle ne figure pas dans un référentiel source, il doit résider dans un artefact généré par une autre action ou dans un compartiment Amazon S3.

Pour plus d'informations sur les sources, consultez [Travailler avec des sources](#).

Interface utilisateur correspondante : onglet Entrées/Sources - facultatif

## Artifacts - input

*(DeployCloudFormationStack/Inputs/Artifacts)*

(Obligatoire si votre AWS SAM modèle CloudFormation ou votre modèle est stocké dans un [artefact de sortie](#) d'une action précédente)

Si le AWS SAM modèle CloudFormation ou que vous souhaitez déployer est contenu dans un artefact généré par une action précédente, spécifiez cet artefact ici. Si votre CloudFormation modèle n'est pas contenu dans un artefact, il doit résider dans votre référentiel source ou dans un compartiment Amazon S3.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

Interface utilisateur correspondante : onglet Configuration/Artefacts - facultatif

## Configuration

*(DeployCloudFormationStack/Configuration)*

(Obligatoire)

Section dans laquelle vous pouvez définir les propriétés de configuration de l'action.

Interface utilisateur correspondante : onglet Configuration

## name

*(DeployCloudFormationStack/Configuration/name)*

(Obligatoire)

Spécifiez le nom de la CloudFormation pile que l'action Deploy AWS CloudFormation stack crée ou met à jour.

Interface utilisateur correspondante : onglet Configuration/nom de la pile

## region

*(DeployCloudFormationStack/Configuration/region)*

(Obligatoire)

Spécifiez le Région AWS lieu dans lequel la pile sera déployée. Pour obtenir la liste des codes de région, consultez la section [Points de terminaison régionaux](#).

Interface utilisateur correspondante : onglet Configuration/région de pile

## template

*(DeployCloudFormationStack/Configuration/template)*

(Obligatoire)

Spécifiez le nom et le chemin d'accès à votre fichier CloudFormation ou à votre fichier AWS SAM modèle. Le modèle peut être au format JSON ou YAML et peut résider dans un référentiel source, un artefact issu d'une action précédente ou un compartiment Amazon S3. Si le fichier modèle se trouve dans un référentiel source ou un artefact, le chemin est relatif à la source ou à la racine de l'artefact. Si le modèle se trouve dans un compartiment Amazon S3, le chemin est la valeur de l'URL de l'objet du modèle.

Exemples :

```
./MyFolder/MyTemplate.json
```

```
MyFolder/MyTemplate.yml
```

```
https://MyBucket.s3.us-west-2.amazonaws.com/MyTemplate.yml
```



**Note**

Vous devrez peut-être ajouter un préfixe au chemin du fichier du modèle pour indiquer dans quel artefact ou dans quelle source le trouver. Pour plus d'informations, consultez [Référencement de fichiers dans un référentiel source](#) et [Référencement de fichiers dans un artefact](#).

Interface utilisateur correspondante : onglet/modèle de configuration

role-arn

(*DeployCloudFormationStack*/Configuration/role-arn)

(Obligatoire)

Spécifiez le nom de ressource Amazon (ARN) du rôle de pile. CloudFormation utilise ce rôle pour accéder aux ressources de votre stack et les modifier. Par exemple :  
`arn:aws:iam::123456789012:role/StackRole`.

Assurez-vous que le rôle de pile inclut :

- Une ou plusieurs politiques d'autorisation. Les politiques dépendent des ressources dont vous disposez dans votre pile. Par exemple, si votre pile inclut une AWS Lambda fonction, vous devez ajouter des autorisations qui accordent l'accès à Lambda. Si vous avez suivi le didacticiel décrit dans [Tutoriel : Déployer une application sans serveur à l'aide de AWS CloudFormation](#), il inclut une procédure intitulée, [Pour créer un rôle de pile](#) qui répertorie les autorisations dont le rôle de pile a besoin si vous déployez une pile d'applications sans serveur typique.

**Warning**

Limitez les autorisations à celles requises par le CloudFormation service pour accéder aux ressources de votre pile. L'utilisation d'un rôle doté d'autorisations plus étendues peut présenter un risque de sécurité.

- La politique de confiance suivante :

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

```
{
  "Sid": "",
  "Effect": "Allow",
  "Principal": {
    "Service": "cloudformation.amazonaws.com"
  },
  "Action": "sts:AssumeRole"
}
]
```

Associez éventuellement ce rôle à la connexion de votre compte. Pour en savoir plus sur l'association d'un rôle IAM à une connexion à un compte, consultez [Ajouter des rôles IAM aux connexions aux comptes](#). Si vous n'associez pas le rôle de pile à la connexion au compte, le rôle de pile n'apparaîtra pas dans la liste déroulante des rôles de pile de l'éditeur visuel ; toutefois, l'ARN du rôle peut toujours être spécifié dans le `role-arn` champ à l'aide de l'éditeur YAML.

#### Note

Vous pouvez indiquer le nom du `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle ici, si vous le souhaitez. Pour plus d'informations sur ce rôle, consultez [Création du CodeCatalystWorkflowDevelopmentRole-\*spaceName\* rôle pour votre compte et votre espace](#). Sachez que le `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle dispose d'autorisations très étendues, ce qui peut présenter un risque de sécurité. Nous vous recommandons de n'utiliser ce rôle que dans les didacticiels et les scénarios où la sécurité est moins préoccupante.

Interface utilisateur correspondante : onglet Configuration/rôle Stack - facultatif

## capabilities

(*DeployCloudFormationStack*/Configuration/capabilities)

(Obligatoire)

Spécifiez une liste des fonctionnalités IAM requises pour autoriser la création AWS CloudFormation de certaines piles. Dans la plupart des cas, vous pouvez repartir `capabilities` avec la valeur par défaut de `CAPABILITY_IAM`, `CAPABILITY_NAMED_IAM`, `CAPABILITY_AUTO_EXPAND`.

Si vous voyez `##[error] requires capabilities: [capability-name]` dans les journaux des actions de votre AWS CloudFormationstack Deploy, consultez [Comment corriger les erreurs liées aux fonctionnalités IAM ?](#) pour plus d'informations sur la manière de résoudre le problème.

Pour plus d'informations sur les fonctionnalités IAM, consultez la section [Reconnaissance des ressources IAM dans les AWS CloudFormation modèles](#) du Guide de l'utilisateur IAM.

Interface utilisateur correspondante : onglet Configuration/Avancé/Capacités

parameter-overrides

(*DeployCloudFormationStack*/Configuration/parameter-overrides)

(Facultatif)

Spécifiez dans votre AWS SAM modèle AWS CloudFormation ou dans votre modèle des paramètres qui n'ont pas de valeurs par défaut ou pour lesquels vous souhaitez spécifier des valeurs autres que celles par défaut. Pour plus d'informations sur les paramètres, consultez la section [Paramètres](#) du guide de AWS CloudFormation l'utilisateur.

L'`parameter-override`sétablissement accepte :

- Un fichier JSON contenant les paramètres et les valeurs.
- Liste de paramètres et de valeurs séparée par des virgules.

Pour spécifier un fichier JSON

1. Assurez-vous que le fichier JSON utilise l'une des syntaxes suivantes :

```
{
  "Parameters": {
    "Param1": "Value1",
    "Param2": "Value2",
    ...
  }
}
```

Ou...

```
[
```

```
{
  "ParameterKey": "Param1",
  "ParameterValue": "Value1"
},
...
]
```

(Il existe d'autres syntaxes, mais elles ne sont pas prises en charge CodeCatalyst au moment de la rédaction.) Pour plus d'informations sur la spécification de CloudFormation paramètres dans un fichier JSON, consultez la section [Syntaxe JSON prise en charge](#) dans le manuel de référence des AWS CLI commandes.

2. Spécifiez le chemin d'accès au fichier JSON en utilisant l'un des formats suivants :

- Si votre fichier JSON réside dans un artefact de sortie d'une action précédente, utilisez :

```
file:///artifacts/current-action-name/output-artifact-name/path-to-json-file
```

Voir l'exemple 1 pour plus de détails.

- Si votre fichier JSON se trouve dans votre référentiel source, utilisez :

```
file:///sources/WorkflowSource/path-to-json-file
```

Voir l'exemple 2 pour plus de détails.

Exemple 1 — Le fichier JSON réside dans un artefact de sortie

```
##My workflow YAML
...
Actions:
  MyBuildAction:
    Identifieur: aws/build@v1
    Outputs:
      Artifacts:
        - Name: ParamArtifact
          Files:
            - params.json
    Configuration:
      ...
  MyDeployCFNStackAction:
    Identifieur: aws/cfn-deploy@v1
```

```

Configuration:
  parameter-overrides: file:///artifacts/MyDeployCFNStackAction/
  ParamArtifact/params.json

```

Exemple 2 — Le fichier JSON se trouve dans votre référentiel source, dans un dossier appelé `my/folder`

```

##My workflow YAML
...
Actions:
  MyDeployCloudFormationStack:
    Identifier: aws/cfn-deploy@v1
    Inputs:
      Sources:
        - WorkflowSource
    Configuration:
      parameter-overrides: file:///sources/WorkflowSource/my/folder/params.json

```

Pour utiliser une liste de paramètres séparée par des virgules

- Ajoutez des paires nom-valeur de paramètre dans la `parameter-overrides` propriété en utilisant le format suivant :

*param-1=value-1,param-2=value-2*

Par exemple, en supposant le AWS CloudFormation modèle suivant :

```

##My CloudFormation template

Description: My AWS CloudFormation template

Parameters:
  InstanceType:
    Description: Defines the Amazon EC2 compute for the production server.
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - t2.small
      - t3.medium

```

Resources:

...

... vous pouvez définir la `parameter-overrides` propriété comme suit :

```
##My workflow YAML
...
Actions:
...
  DeployCloudFormationStack:
    Identifier: aws/cfn-deploy@v1
    Configuration:
      parameter-overrides: InstanceType=t3.medium,UseVPC=true
```

#### Note

Vous pouvez spécifier un nom de paramètre sans valeur correspondante en l'utilisant `undefined` comme valeur. Par exemple :

```
parameter-overrides: MyParameter=undefined
```

L'effet est que lors d'une mise à jour de la pile, CloudFormation utilise la valeur de paramètre existante pour le nom de paramètre donné.

Interface utilisateur correspondante :

- Onglet Configuration/Avancé/Remplacements de paramètres
- Onglet Configuration/Avancé/Remplacements de paramètres/Spécifier les remplacements à l'aide d'un fichier
- Onglet Configuration/Avancé/Remplacements de paramètres/Spécifier les remplacements à l'aide d'un ensemble de valeurs

`no-execute-changeset`

(*DeployCloudFormationStack*/Configuration/no-execute-changeset)

(Facultatif)

Spécifiez si vous souhaitez CodeCatalyst créer l'ensemble de CloudFormation modifications, puis arrêtez-le avant de l'exécuter. Cela vous donne la possibilité de passer en revue les modifications définies dans la CloudFormation console. Si vous estimez que l'ensemble de modifications semble correct, désactivez cette option, puis réexécutez le flux de travail afin de CodeCatalyst pouvoir créer et exécuter l'ensemble de modifications sans vous arrêter. Par défaut, l'ensemble de modifications est créé et exécuté sans arrêt. Pour plus d'informations, consultez le paramètre de AWS CloudFormation [déploiement](#) dans la référence des AWS CLI commandes. Pour plus d'informations sur l'affichage d'un ensemble de modifications, consultez la section [Affichage d'un ensemble de modifications](#) dans le Guide de AWS CloudFormation l'utilisateur.

Interface utilisateur correspondante : onglet Configuration/Avancé/Aucune exécution définie

fail-on-empty-changeset

*(DeployCloudFormationStack/Configuration/fail-on-empty-changeset)*

(Facultatif)

Spécifiez si vous souhaitez échouer CodeCatalyst à l'action Deploy AWS CloudFormation stack si l'ensemble de CloudFormation modifications est vide. (Si un ensemble de modifications est vide, cela signifie qu'aucune modification n'a été apportée à la pile lors du dernier déploiement.) Par défaut, l'action est autorisée si l'ensemble de modifications est vide et un UPDATE\_COMPLETE message est renvoyé même si la pile n'a pas été mise à jour.

Pour plus d'informations sur ce paramètre, consultez le paramètre de AWS CloudFormation [déploiement](#) dans le manuel de référence des AWS CLI commandes. Pour plus d'informations sur les ensembles de modifications, consultez la section [Mise à jour des piles à l'aide d'ensembles de modifications](#) dans le Guide de AWS CloudFormation l'utilisateur.

Interface utilisateur correspondante : onglet Configuration/Avancé/Échec si le changeset est vide

disable-rollback

*(DeployCloudFormationStack/Configuration/disable-rollback)*

(Facultatif)

Spécifiez si vous CodeCatalyst souhaitez annuler le déploiement de la pile en cas d'échec. Le rollback ramène la pile au dernier état stable connu. Par défaut, les annulations sont activées. Pour

plus d'informations sur ce paramètre, consultez le paramètre de AWS CloudFormation [déploiement](#) dans le manuel de référence des AWS CLI commandes.

Pour plus d'informations sur la façon dont l'action Deploy AWS CloudFormation stack gère les annulations, consultez [Configuration des annulations](#).

Pour plus d'informations sur l'annulation d'une pile, consultez la section [Options d'échec d'une pile](#) dans le Guide de AWS CloudFormation l'utilisateur.

Interface utilisateur correspondante : onglet Configuration/Avancé/Désactiver la restauration

termination-protection

*(DeployCloudFormationStack/Configuration/termination-protection)*

(Facultatif)

Spécifiez si vous souhaitez que la AWS CloudFormation pile Deploy ajoute une protection de terminaison à la pile qu'elle déploie. 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. Par défaut, la protection contre la résiliation est désactivée. Pour plus d'informations, consultez [la section Protection d'une pile contre la suppression](#) dans le Guide de AWS CloudFormation l'utilisateur.

Interface utilisateur correspondante : onglet Configuration/Avancé/Protection de terminaison

timeout-in-minutes

*(DeployCloudFormationStack/Configuration/timeout-in-minutes)*

(Facultatif)

Spécifiez le temps, en minutes, à CloudFormation allouer avant de chronométrer les opérations de création de pile et de définir le statut de la pile sur CREATE\_FAILED. Si CloudFormation ne peut pas créer la pile complète dans le délai imparti, il échoue à créer la pile en raison du délai d'expiration et procède à la restauration de 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.



Interface utilisateur correspondante : onglet Configuration/avancé/délai CloudFormation d'expiration notification-arns

(*DeployCloudFormationStack*/Configuration/notification-arns)

(Facultatif)


Spécifiez l'ARN d'une rubrique Amazon SNS à laquelle vous souhaitez CodeCatalyst envoyer des messages de notification. Par exemple, `arn:aws:sns:us-east-1:111222333:MyTopic`. Lorsque l'action Deploy AWS CloudFormation stack s'exécute, CodeCatalyst coordonne avec CloudFormation pour envoyer une notification par AWS CloudFormation événement qui se produit pendant le processus de création ou de mise à jour de la pile. (Les événements sont visibles dans l'onglet Événements de la AWS CloudFormation console pour la pile.) Vous pouvez définir jusqu'à cinq sujets. Pour plus d'informations, consultez la section [Qu'est-ce qu'Amazon SNS ?](#).

Interface utilisateur correspondante : onglet Configuration/Avancé/ARN de notification monitor-alarm-arns

(*DeployCloudFormationStack*/Configuration/monitor-alarm-arns)

(Facultatif)

Spécifiez le nom de ressource Amazon (ARN) d'une CloudWatch alarme Amazon à utiliser comme déclencheur d'annulation. Par exemple, `arn:aws:cloudwatch::123456789012:alarm/MyAlarm`. Vous pouvez disposer d'un maximum de cinq déclencheurs de restauration.

 Note

Si vous spécifiez un ARN CloudWatch d'alarme, vous devez également configurer des autorisations supplémentaires pour permettre à l'action d'y accéder CloudWatch. Pour de plus amples informations, veuillez consulter [Configuration des annulations](#).

Interface utilisateur correspondante : onglet Configuration/Avancé/Surveiller les ARN d'alarme monitor-timeout-in-minutes

(*DeployCloudFormationStack*/Configuration/monitor-timeout-in-minutes)

## (Facultatif)

Spécifiez une durée, comprise entre 0 et 180 minutes, pendant laquelle CloudFormation les alarmes spécifiées seront surveillées. La surveillance commence une fois que toutes les ressources de la pile ont été déployées. Si l'alarme se produit dans le délai de surveillance spécifié, le déploiement échoue et CloudFormation annule l'ensemble du fonctionnement de la pile.

Par défaut : 0. CloudFormation surveille uniquement les alarmes pendant le déploiement des ressources de la pile, et non après.

Interface utilisateur correspondante : onglet Configuration/Avancé/Temps de surveillance

## tags

(*DeployCloudFormationStack*/Configuration/tags)

## (Facultatif)

Spécifiez les balises à attacher à votre CloudFormation pile. Les balises sont des paires clé-valeur arbitraires que vous pouvez utiliser pour identifier votre pile à des fins telles que la répartition des coûts. Pour plus d'informations sur ce que sont les balises et comment elles peuvent être utilisées, consultez la section [Marquage de vos ressources](#) dans le guide de l'utilisateur Amazon EC2 pour les instances Linux. Pour plus d'informations sur le balisage CloudFormation, consultez la section [Configuration des options de AWS CloudFormation pile](#) dans le Guide de l'AWS CloudFormation utilisateur.

Une clé peut comporter des caractères alphanumériques ou des espaces, et peut comporter jusqu'à 127 caractères. Une valeur peut comporter des caractères alphanumériques ou des espaces, et peut comporter jusqu'à 255 caractères.

Vous pouvez ajouter jusqu'à 50 balises uniques pour chaque pile.

Interface utilisateur correspondante : onglet Configuration/Avancé/Tags

## Référence d'action « Déployer sur Amazon ECS »

Vous trouverez ci-dessous la référence YAML de définition d'action pour l'action Deploy to Amazon ECS. Pour savoir comment utiliser cette action, consultez [Ajout de l'action « Déployer sur Amazon ECS »](#).

**Note**

La plupart des propriétés YAML suivantes ont des éléments d'interface utilisateur correspondants dans l'éditeur visuel. Pour rechercher un élément de l'interface utilisateur, utilisez Ctrl+F. L'élément sera répertorié avec sa propriété YAML associée.

```
# The workflow definition starts here.
# See Propriétés de haut niveau for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
ECSDeployAction\_nn:
  Identifiant: aws/ecs-deploy@v1
  DependsOn:
    - build-action
  Compute:
    Type: EC2 | Lambda
    Fleet: fleet-name
  Timeout: timeout-minutes
  Environment:
    Name: environment-name
  Connections:
    - Name: account-connection-name
      Role: DeployToECS
  Inputs:
    # Specify a source or an artifact, but not both.
  Sources:
    - source-name-1
  Artifacts:
    - task-definition-artifact
  Configuration:
    region: us-east-1
    cluster: ecs-cluster
    service: ecs-service
    task-definition: task-definition-path
    force-new-deployment: false|true
    codedeploy-appspec: app-spec-file-path
    codedeploy-application: application-name
```

```
codedeploy-deployment-group: deployment-group-name  
codedeploy-deployment-description: deployment-description
```

## ECSDeployAction

(Obligatoire)

Spécifiez le nom de l'action. Tous les noms d'action doivent être uniques dans le flux de travail. Les noms d'action sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms d'action.

Par défaut: ECSDeployAction\_nn.

Interface utilisateur correspondante : onglet Configuration/nom d'affichage de l'action

### Identifiant

(*ECSDeployAction*/Identifiant)

(Obligatoire)

Identifie l'action. Ne modifiez pas cette propriété, sauf si vous souhaitez modifier la version. Pour de plus amples informations, veuillez consulter [Utilisation des versions d'action](#).

Par défaut: aws/ecs-deploy@v1.

Interface utilisateur correspondante : diagramme de flux de travail/étiquette ECS DeployAction\_nn/  
aws/ecs-deploy @v1

### DependsOn

(*ECSDeployAction*/DependsOn)

(Facultatif)

Spécifiez une action ou un groupe d'actions qui doit être exécuté correctement pour que cette action soit exécutée.

Pour plus d'informations sur la fonctionnalité « dépend », consultez. [Configuration des actions pour qu'elles dépendent d'autres actions](#)

Interface utilisateur correspondante : onglet Entrées/dépend de - facultatif

## Compute

(*ECSDeployAction*/Compute)

(Facultatif)

Le moteur informatique utilisé pour exécuter les actions de votre flux de travail. Vous pouvez spécifier le calcul au niveau du flux de travail ou au niveau de l'action, mais pas les deux. Lorsqu'elle est spécifiée au niveau du flux de travail, la configuration de calcul s'applique à toutes les actions définies dans le flux de travail. Au niveau du flux de travail, vous pouvez également exécuter plusieurs actions sur la même instance. Pour de plus amples informations, veuillez consulter [Partage du calcul entre les actions](#).

Interface utilisateur correspondante : aucune

## Type

(*ECSDeployAction*/Compute/Type)

(Obligatoire s'[Compute](#) est inclus)

Type de moteur de calcul. Vous pouvez utiliser l'une des valeurs suivantes :

- EC2 (éditeur visuel) ou EC2 (éditeur YAML)  
Optimisé pour la flexibilité lors des courses d'action.
- Lambda (éditeur visuel) ou Lambda (éditeur YAML)  
Vitesses de démarrage des actions optimisées.

Pour plus d'informations sur les types de calcul, veuillez consulter [À propos des types de calcul](#).

Interface utilisateur correspondante : onglet Configuration/Avancé - facultatif/Type de calcul

## Fleet

(*ECSDeployAction*/Compute/Fleet)

(Facultatif)

Spécifiez la machine ou le parc qui exécutera votre flux de travail ou vos actions de flux de travail. Dans le cas des flottes à la demande, lorsqu'une action démarre, le flux de travail fournit les ressources dont il a besoin et les machines sont détruites à la fin de l'action. Exemples de flottes à la demande : `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Pour plus d'informations sur les flottes à la demande, consultez [Propriétés de flotte à la demande](#).

Avec les flottes provisionnées, vous configurez un ensemble de machines dédiées pour exécuter les actions de votre flux de travail. Ces machines restent inactives, prêtes à exécuter des actions immédiatement. Pour plus d'informations sur les flottes provisionnées, consultez [Propriétés du parc provisionné](#)

S'il `Fleet` est omis, la valeur par défaut est `Linux.x86-64.Large`.

Interface utilisateur correspondante : onglet Configuration/Avancé - facultatif/Parc de calcul

## Timeout

(*ECSDeployAction*/Timeout)

(Facultatif)

Spécifiez la durée en minutes (éditeur YAML) ou en heures et minutes (éditeur visuel) pendant laquelle l'action peut être exécutée avant la CodeCatalyst fin de l'action. Le minimum est de 5 minutes et le maximum est décrit dans [Quotas pour les flux de travail dans CodeCatalyst](#). Le délai d'expiration par défaut est le même que le délai d'expiration maximal.

Interface utilisateur correspondante : onglet Configuration/Délai d'expiration - facultatif

## Environnement

(*ECSDeployAction*/Environnement)

(Obligatoire)

Spécifiez l' CodeCatalyst environnement à utiliser avec l'action.

Pour plus d'informations sur les environnements, reportez-vous [Utilisation des environnements](#) aux sections et [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role/'

Environnement

## Name

(*ECSDeployAction*/Environment/Name)

(Obligatoire s'[Environment](#) est inclus)

Spécifiez le nom de l'environnement existant que vous souhaitez associer à l'action.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/  
Environnement

## Connections

(*ECSDeployAction*/Environment/Connections)

(Obligatoire s'[Environment](#) est inclus)

Spécifiez la connexion au compte à associer à l'action. Vous pouvez spécifier un maximum d'une connexion à un compte sous `Environment`.

Pour plus d'informations sur les connexions aux comptes, consultez [Administration Comptes AWS d'un espace](#). Pour plus d'informations sur la façon d'associer une connexion à un compte à votre environnement, consultez [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/ connexion  
AWS au compte

## Name

(*ECSDeployAction*/Environment/Connections/Name)

(Obligatoire)

Spécifiez le nom de la connexion au compte.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/ connexion  
AWS au compte

## Role

(*ECSDeployAction*/Environment/Connections/Role)

## (Obligatoire)

Spécifiez le nom du rôle IAM auquel l'action Deploy to Amazon ECS permet d'accéder AWS. Assurez-vous que ce rôle inclut les politiques suivantes :

- La politique d'autorisation suivante :

### Warning

Limitez les autorisations à celles indiquées dans la politique suivante. L'utilisation d'un rôle doté d'autorisations plus étendues peut présenter un risque de sécurité.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "ecs:DescribeServices",
      "ecs:CreateTaskSet",
      "ecs>DeleteTaskSet",
      "ecs:ListClusters",
      "ecs:RegisterTaskDefinition",
      "ecs:UpdateServicePrimaryTaskSet",
      "ecs:UpdateService",
      "elasticloadbalancing:DescribeTargetGroups",
      "elasticloadbalancing:DescribeListeners",
      "elasticloadbalancing:ModifyListener",
      "elasticloadbalancing:DescribeRules",
      "elasticloadbalancing:ModifyRule",
      "lambda:InvokeFunction",
      "lambda:ListFunctions",
      "cloudwatch:DescribeAlarms",
      "sns:Publish",
      "sns:ListTopics",
      "s3:GetObject",
      "s3:GetObjectVersion",
      "codedeploy:CreateApplication",
      "codedeploy:CreateDeployment",
      "codedeploy:CreateDeploymentGroup",
      "codedeploy:GetApplication",
      "codedeploy:GetDeployment",
      "codedeploy:GetDeploymentGroup",
```



```

    "codedeploy:ListApplications",
    "codedeploy:ListDeploymentGroups",
    "codedeploy:ListDeployments",
    "codedeploy:StopDeployment",
    "codedeploy:GetDeploymentTarget",
    "codedeploy:ListDeploymentTargets",
    "codedeploy:GetDeploymentConfig",
    "codedeploy:GetApplicationRevision",
    "codedeploy:RegisterApplicationRevision",
    "codedeploy:BatchGetApplicationRevisions",
    "codedeploy:BatchGetDeploymentGroups",
    "codedeploy:BatchGetDeployments",
    "codedeploy:BatchGetApplications",
    "codedeploy:ListApplicationRevisions",
    "codedeploy:ListDeploymentConfigs",
    "codedeploy:ContinueDeployment"
  ],
  "Resource": "*",
  "Effect": "Allow"
}, {"Action": [
  "iam:PassRole"
],
  "Effect": "Allow",
  "Resource": "*",
  "Condition": {"StringLike": {"iam:PassedToService": [
    "ecs-tasks.amazonaws.com",
    "codedeploy.amazonaws.com"
  ]
  }
}
}]
}

```

### Note

La première fois que le rôle est utilisé, utilisez le caractère générique suivant dans la déclaration de politique de ressources, puis délimitez la politique avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

- La politique de confiance personnalisée suivante :

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

Assurez-vous que ce rôle est ajouté à la connexion de votre compte. Pour en savoir plus sur l'ajout d'un rôle IAM à une connexion à un compte, consultez [Ajouter des rôles IAM aux connexions aux comptes](#).

#### Note

Vous pouvez indiquer le nom du CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle ici, si vous le souhaitez. Pour plus d'informations sur ce rôle, consultez [Création du CodeCatalystWorkflowDevelopmentRole-\*spaceName\* rôle pour votre compte et votre espace](#). Sachez que le CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle dispose d'autorisations très étendues, ce qui peut présenter un risque de sécurité. Nous vous recommandons de n'utiliser ce rôle que dans les didacticiels et les scénarios où la sécurité est moins préoccupante.

Interface utilisateur correspondante : onglet Configuration/Environnement/Compte/Rôle/ Rôle

## Inputs

(*ECSDeployAction*/Inputs)

## (Facultatif)

La `Inputs` section définit les données dont ils ont `ECSDeployAction` besoin lors de l'exécution d'un flux de travail.

### Note

Une seule entrée (source ou artefact) est autorisée par action `Deploy to Amazon ECS`.

Interface utilisateur correspondante : onglet Entrées

## Sources

(*`ECSDeployAction`*/Inputs/Sources)

(Obligatoire si votre fichier de définition de tâche est stocké dans un référentiel source)

Si votre fichier de définition de tâche est stocké dans un référentiel source, spécifiez l'étiquette de ce référentiel source. Actuellement, la seule étiquette prise en charge est `WorkflowSource`.

Si votre fichier de définition de tâche n'est pas contenu dans un référentiel source, il doit résider dans un artefact généré par une autre action.

Pour plus d'informations sur les sources, consultez [Travailler avec des sources](#).

Interface utilisateur correspondante : onglet Entrées/Sources - facultatif

## Artifacts - input

(*`ECSDeployAction`*/Inputs/Artifacts)

(Obligatoire si votre fichier de définition de tâche est stocké dans un [artefact de sortie](#) d'une action précédente)

Si le fichier de définition de tâche que vous souhaitez déployer est contenu dans un artefact généré par une action précédente, spécifiez cet artefact ici. Si votre fichier de définition de tâche n'est pas contenu dans un artefact, il doit résider dans votre référentiel source.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

Interface utilisateur correspondante : onglet Configuration/Artefacts - facultatif

## Configuration

(*ECSDeployAction*/Configuration)

(Obligatoire)

Section dans laquelle vous pouvez définir les propriétés de configuration de l'action.

Interface utilisateur correspondante : onglet Configuration

## region

(Configuration/region)

(Obligatoire)

Spécifiez la AWS région dans laquelle se trouvent votre cluster et votre service Amazon ECS. Pour obtenir la liste des codes de région, consultez la section [Points de terminaison régionaux](#) dans le Références générales AWS.

Interface utilisateur correspondante : onglet Configuration/région

## cluster

(*ECSDeployAction*/Configuration/cluster)

(Obligatoire)

Spécifiez le nom d'un cluster Amazon ECS existant. L'action Deploy to Amazon ECS déploiera votre application conteneurisée en tant que tâche dans ce cluster. Pour plus d'informations sur les clusters Amazon ECS, consultez la section [Clusters](#) du manuel Amazon Elastic Container Service Developer Guide.

Interface utilisateur correspondante : onglet Configuration/Cluster

## service

(*ECSDeployAction*/Configuration/service)

(Obligatoire)

Spécifiez le nom d'un service Amazon ECS existant quiinstanciera le fichier de définition de tâche. Ce service doit résider sous le cluster spécifié dans le `cluster` champ. Pour plus d'informations sur les services Amazon ECS, consultez les [services Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.

Interface utilisateur correspondante : onglet Configuration/Service

## task-definition

(*ECSDeployAction*/Configuration/task-definition)

(Obligatoire)

Spécifiez le chemin d'accès à un fichier de définition de tâche existant. Si le fichier se trouve dans votre dépôt source, le chemin est relatif au dossier racine du dépôt source. Si votre fichier réside dans un artefact issu d'une action de flux de travail précédente, le chemin est relatif au dossier racine de l'artefact. Pour plus d'informations sur les fichiers de définition de tâches, consultez la section [Définitions de tâches](#) dans le manuel Amazon Elastic Container Service Developer Guide.

Interface utilisateur correspondante : onglet Configuration/Définition des tâches

## force-new-deployment

(*ECSDeployAction*/Configuration/force-new-deployment)

(Obligatoire)

S'il est activé, le service Amazon ECS peut démarrer de nouveaux déploiements sans modifier la définition du service. Le déploiement forcé entraîne l'arrêt par le service de toutes les tâches en cours d'exécution et le lancement de nouvelles tâches. Pour plus d'informations sur le forçage de nouveaux déploiements, consultez la section [Mettre à jour un service](#) dans le manuel Amazon Elastic Container Service Developer Guide.

Par défaut : `false`

Interface utilisateur correspondante : onglet Configuration/Forcer un nouveau déploiement du service

## codedeploy-appspec

(*ECSDeployAction*/Configuration/codedeploy-appspec)

(Obligatoire si vous avez configuré votre service Amazon ECS pour utiliser des déploiements bleu/vert, sinon, omettez)

Spécifiez le nom et le chemin d'accès à un fichier de spécification CodeDeploy d'application (AppSpec) existant. Ce fichier doit se trouver à la racine de votre dépôt CodeCatalyst source. Pour plus d'informations sur AppSpec les fichiers, consultez les [fichiers de spécification de CodeDeploy l'application \(AppSpec\)](#) dans le guide de AWS CodeDeploy l'utilisateur.

**Note**

Ne fournissez CodeDeploy des informations que si vous avez configuré votre service Amazon ECS pour effectuer des déploiements bleu/vert. Pour les déploiements de mises à jour continues (par défaut), omettez CodeDeploy les informations. Pour plus d'informations sur les déploiements Amazon ECS, consultez les [types de déploiement Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.

**Note**

Les CodeDeploy champs peuvent être masqués dans l'éditeur visuel. Pour les faire apparaître, voir [Pourquoi les CodeDeploy champs sont-ils absents de l'éditeur visuel ?](#).

Interface utilisateur correspondante : onglet Configuration/ CodeDeploy AppSpec

codedeploy-application

(*ECSDeployAction*/Configuration/codedeploy-application)

(Obligatoire s'codedeploy-appspecil est inclus)

Spécifiez le nom d'une CodeDeploy application existante. Pour plus d'informations sur CodeDeploy les applications, consultez la section [Utilisation des applications CodeDeploy dans](#) le Guide de AWS CodeDeploy l'utilisateur.

Interface utilisateur correspondante : onglet de configuration/application CodeDeploy

codedeploy-deployment-group

(*ECSDeployAction*/Configuration/codedeploy-deployment-group)

(Obligatoire s'codeDeploy-appspecil est inclus)

Spécifiez le nom d'un groupe CodeDeploy de déploiement existant. Pour plus d'informations sur les groupes de CodeDeploy déploiement, consultez la section [Utilisation des groupes de déploiement CodeDeploy dans](#) le Guide de AWS CodeDeploy l'utilisateur.

Interface utilisateur correspondante : onglet Configuration/groupe CodeDeploy de déploiement  
codeDeploy-deployment-description

(*ECSDeployAction*/Configuration/codeDeploy-deployment-description)

(Facultatif)

Spécifiez une description du déploiement que cette action créera. Pour plus d'informations, consultez la section [Utilisation des déploiements CodeDeploy dans](#) le Guide de l'AWS CodeDeployutilisateur.

Interface utilisateur correspondante : onglet Configuration/description CodeDeploy du déploiement

## Référence d'action « Déployer vers un cluster Kubernetes »

Voici la référence YAML de définition d'action pour l'action Deploy to Kubernetes cluster. Pour savoir comment utiliser cette action, consultez [Ajout de l'action « Déployer vers le cluster Kubernetes »](#).

### Note

La plupart des propriétés YAML suivantes ont des éléments d'interface utilisateur correspondants dans l'éditeur visuel. Pour rechercher un élément de l'interface utilisateur, utilisez Ctrl+F. L'élément sera répertorié avec sa propriété YAML associée.

```
# The workflow definition starts here.
# See Propriétés de haut niveau for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
DeployToKubernetesCluster\_nn:
  Identifiant: aws/kubernetes-deploy@v1
  DependsOn:
```

```
- build-action
Compute:
  - Type: EC2 | Lambda
  - Fleet: fleet-name
Timeout: timeout-minutes
Environment:
  Name: environment-name
Connections:
  - Name: account-connection-name
  Role: DeployToEKS
Inputs:
  # Specify a source or an artifact, but not both.
Sources:
  - source-name-1
Artifacts:
  - manifest-artifact
Configuration:
  Namespace: namespace
  Region: us-east-1
  Cluster: eks-cluster
  Manifests: manifest-path
```

## DeployToKubernetesCluster

(Obligatoire)

Spécifiez le nom de l'action. Tous les noms d'action doivent être uniques dans le flux de travail. Les noms d'action sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms d'action.

Par défaut: `DeployToKubernetesCluster_nn`.

Interface utilisateur correspondante : onglet Configuration/nom d'affichage de l'action

### Identifiant

*(DeployToKubernetesCluster/Identifiant)*

(Obligatoire)

Identifie l'action. Ne modifiez pas cette propriété, sauf si vous souhaitez modifier la version. Pour de plus amples informations, veuillez consulter [Utilisation des versions d'action](#).



Par défaut: `aws/kubernetes-deploy@v1`.

Interface utilisateur correspondante : diagramme de flux de travail/ `DeployToKubernetesCluster _nn/ aws/kubernetes-deploy @v1` label

## DependsOn

*(DeployToKubernetesCluster/DependsOn)*

(Facultatif)

Spécifiez une action ou un groupe d'actions qui doit être exécuté correctement pour que cette action soit exécutée.

Pour plus d'informations sur la fonctionnalité « dépend », consultez. [Configuration des actions pour qu'elles dépendent d'autres actions](#)

Interface utilisateur correspondante : onglet Entrées/dépend de - facultatif

## Compute

*(DeployToKubernetesCluster/Compute)*

(Facultatif)

Le moteur informatique utilisé pour exécuter les actions de votre flux de travail. Vous pouvez spécifier le calcul au niveau du flux de travail ou au niveau de l'action, mais pas les deux. Lorsqu'elle est spécifiée au niveau du flux de travail, la configuration de calcul s'applique à toutes les actions définies dans le flux de travail. Au niveau du flux de travail, vous pouvez également exécuter plusieurs actions sur la même instance. Pour de plus amples informations, veuillez consulter [Partage du calcul entre les actions](#).

Interface utilisateur correspondante : aucune

## Type

*(DeployToKubernetesCluster/Compute/Type)*

(Obligatoire s'[Compute](#) est inclus)

Type de moteur de calcul. Vous pouvez utiliser l'une des valeurs suivantes :

- EC2 (éditeur visuel) ou EC2 (éditeur YAML)

Optimisé pour la flexibilité lors des courses d'action.

- Lambda (éditeur visuel) ou Lambda (éditeur YAML)

Vitesses de démarrage des actions optimisées.

Pour plus d'informations sur les types de calcul, veuillez consulter [À propos des types de calcul](#).

Interface utilisateur correspondante : onglet Configuration/Avancé - facultatif/Type de calcul

## Fleet

*(DeployToKubernetesCluster/Compute/Fleet)*

(Facultatif)

Spécifiez la machine ou le parc qui exécutera votre flux de travail ou vos actions de flux de travail. Dans le cas des flottes à la demande, lorsqu'une action démarre, le flux de travail fournit les ressources dont il a besoin et les machines sont détruites à la fin de l'action. Exemples de flottes à la demande :Linux.x86-64.Large,Linux.x86-64.XLarge. Pour plus d'informations sur les flottes à la demande, consultez[Propriétés de flotte à la demande](#).

Avec les flottes provisionnées, vous configurez un ensemble de machines dédiées pour exécuter les actions de votre flux de travail. Ces machines restent inactives, prêtes à exécuter des actions immédiatement. Pour plus d'informations sur les flottes provisionnées, consultez. [Propriétés du parc provisionné](#)

S'il Fleet est omis, la valeur par défaut estLinux.x86-64.Large.

Interface utilisateur correspondante : onglet Configuration/Avancé - facultatif/Parc de calcul

## Timeout

*(DeployToKubernetesCluster/Timeout)*

(Facultatif)

Spécifiez la durée en minutes (éditeur YAML) ou en heures et minutes (éditeur visuel) pendant laquelle l'action peut être exécutée avant la CodeCatalyst fin de l'action. Le minimum est de 5 minutes et le maximum est décrit dans[Quotas pour les flux de travail dans CodeCatalyst](#). Le délai d'expiration par défaut est le même que le délai d'expiration maximal.

Interface utilisateur correspondante : onglet Configuration/Délai d'expiration - facultatif

## Environment

(*DeployToKubernetesCluster*/Environment)

(Obligatoire)

Spécifiez l' CodeCatalyst environnement à utiliser avec l'action.

Pour plus d'informations sur les environnements, reportez-vous [Utilisation des environnements](#) aux sections et [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/  
Environnement

## Name

(*DeployToKubernetesCluster*/Environment/Name)

(Obligatoire s'[Environnement](#)il est inclus)

Spécifiez le nom de l'environnement existant que vous souhaitez associer à l'action.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/  
Environnement

## Connections

(*DeployToKubernetesCluster*/Environment/Connections)

(Obligatoire s'[Environnement](#)il est inclus)

Spécifiez la connexion au compte à associer à l'action. Vous pouvez spécifier un maximum d'une connexion à un compte sous `Environment`.

Pour plus d'informations sur les connexions aux comptes, consultez [Administration Comptes AWS d'un espace](#). Pour plus d'informations sur la façon d'associer une connexion à un compte à votre environnement, consultez [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/ connexion  
AWS au compte

## Name

(*DeployToKubernetesCluster*/Environment/Connections/Name)

(Obligatoire)

Spécifiez le nom de la connexion au compte.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/ connexion AWS au compte

## Role

(*DeployToKubernetesCluster*/Environment/Connections/Role)

(Obligatoire)

Spécifiez le nom du rôle IAM auquel l'action de cluster Deploy to Kubernetes utilise pour accéder. AWS Assurez-vous que ce rôle inclut les politiques suivantes :

- La politique d'autorisation suivante :

### Warning

Limitez les autorisations à celles indiquées dans la politique suivante. L'utilisation d'un rôle doté d'autorisations plus étendues peut présenter un risque de sécurité.

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

**Note**

La première fois que le rôle est utilisé, utilisez le caractère générique suivant dans la déclaration de politique de ressources, puis délimitez la politique avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

- La politique de confiance personnalisée suivante :

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

Assurez-vous que ce rôle est ajouté à :

- Connexion à votre compte. Pour en savoir plus sur l'ajout d'un rôle IAM à une connexion à un compte, consultez [Ajouter des rôles IAM aux connexions aux comptes](#).
- Vos Kubernetes ConfigMap. Pour en savoir plus sur l'ajout d'un rôle IAM à un ConfigMap, consultez la section [Gérer les utilisateurs et les rôles IAM](#) dans la eksctl documentation.

**i** Tip

Consultez également [Tutoriel : Déploiement d'une application sur Amazon EKS](#) les instructions relatives à l'ajout d'un rôle IAM à une connexion à un compte et ConfigMap.

**i** Note

Vous pouvez indiquer le nom du `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle ici, si vous le souhaitez. Pour plus d'informations sur ce rôle, consultez [Création du `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle pour votre compte et votre espace](#). Sachez que le `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle dispose d'autorisations très étendues, ce qui peut présenter un risque de sécurité. Nous vous recommandons de n'utiliser ce rôle que dans les didacticiels et les scénarios où la sécurité est moins préoccupante. Si vous décidez d'utiliser le `CodeCatalystWorkflowDevelopmentRole-spaceName` rôle, assurez-vous de l'ajouter à votre ConfigMap fichier, en suivant les instructions de la section [Gérer les utilisateurs et les rôles IAM](#) de la `eksctl` documentation.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/ Rôle

## Inputs

(*DeployToKubernetesCluster*/Inputs)

(Facultatif)

La `Inputs` section définit les données dont ils ont `DeployToKubernetesCluster` besoin lors de l'exécution d'un flux de travail.

**i** Note

Une seule entrée (source ou artefact) est autorisée par action Deploy to Amazon EKS.

Interface utilisateur correspondante : onglet Entrées

## Sources

*(DeployToKubernetesCluster/Inputs/Sources)*

(Obligatoire si votre fichier manifeste est stocké dans un référentiel source)

Si votre ou vos fichiers manifestes Kubernetes sont stockés dans un référentiel source, spécifiez l'étiquette de ce référentiel source. Actuellement, la seule étiquette prise en charge est `WorkflowSource`.

Si vos fichiers manifestes ne sont pas contenus dans un référentiel source, ils doivent résider dans un artefact généré par une autre action.

Pour plus d'informations sur les sources, consultez [Travailler avec des sources](#).

Interface utilisateur correspondante : onglet Entrées/Sources - facultatif

## Artifacts - input

*(DeployToKubernetesCluster/Inputs/Artifacts)*

(Obligatoire si votre fichier manifeste est stocké dans un [artefact de sortie](#) d'une action précédente)

Si le ou les fichiers manifestes Kubernetes sont contenus dans un artefact généré par une action précédente, spécifiez cet artefact ici. Si vos fichiers manifestes ne sont pas contenus dans un artefact, ils doivent résider dans votre référentiel source.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

Interface utilisateur correspondante : onglet Configuration/Artefacts - facultatif

## Configuration

*(DeployToKubernetesCluster/Configuration)*

(Obligatoire)

Section dans laquelle vous pouvez définir les propriétés de configuration de l'action.

Interface utilisateur correspondante : onglet Configuration

## Namespace

*(DeployToKubernetesCluster/Configuration/namespace)*

(Facultatif)

Spécifiez l'espace de noms Kubernetes dans lequel votre application Kubernetes sera déployée. À utiliser de default si vous n'utilisez pas d'espaces de noms avec votre cluster. Pour plus d'informations sur les espaces de noms, consultez la section [Subdivision de votre cluster à l'aide des espaces de noms Kubernetes dans la documentation de Kubernetes](#).

Si vous omettez l'espace de noms, une valeur de default est utilisée.

Interface utilisateur correspondante : onglet Configuration/espace de noms

## Region

*(DeployToKubernetesCluster/Configuration/Region)*

(Obligatoire)

Spécifiez la AWS région dans laquelle se trouvent votre cluster et votre service Amazon EKS. Pour obtenir la liste des codes de région, consultez la section [Points de terminaison régionaux](#) dans le Références générales AWS.

Interface utilisateur correspondante : onglet Configuration/région

## Cluster

*(DeployToKubernetesCluster/Configuration/Cluster)*

(Obligatoire)

Spécifiez le nom d'un cluster Amazon EKS existant. L'action Déployer vers le cluster Kubernetes déploiera votre application conteneurisée dans ce cluster. Pour plus d'informations sur les clusters Amazon EKS, consultez la section [Clusters](#) du guide de l'utilisateur Amazon EKS.

Interface utilisateur correspondante : onglet Configuration/Cluster

## Manifests

*(DeployToKubernetesCluster/Configuration/Manifests)*



## (Obligatoire)

Spécifiez le chemin d'accès à vos fichiers manifestes Kubernetes au format YAML, appelés fichiers de configuration, fichiers de configuration ou simplement configurations dans la documentation de Kubernetes.

Si vous utilisez plusieurs fichiers manifestes, placez-les dans un seul dossier et référez ce dossier. Les fichiers manifestes sont traités de manière alphanumérique par Kubernetes. Veillez donc à préfixer les noms de fichiers par des chiffres ou des lettres croissants pour contrôler l'ordre de traitement. Par exemple :

```
00-namespace.yaml
```

```
01-deployment.yaml
```

Si vos fichiers manifestes se trouvent dans votre dépôt source, le chemin est relatif au dossier racine du dépôt source. Si les fichiers se trouvent dans un artefact issu d'une action de flux de travail précédente, le chemin est relatif au dossier racine de l'artefact.

Exemples :

```
Manifests/
```

```
deployment.yaml
```

```
my-deployment.yaml
```

N'utilisez pas de caractères génériques (\*).

### Note

[Les diagrammes Helm](#) et les [fichiers de personnalisation](#) ne sont pas pris en charge.


Pour plus d'informations sur les fichiers manifestes, consultez la section [Organisation des configurations de ressources](#) dans la documentation de Kubernetes.

Interface utilisateur correspondante : onglet Configuration/Manifestes

## Référence GitHub d'action « Actions »

Vous trouverez ci-dessous la référence YAML de définition d'action pour l'action GitHubActions.

Choisissez une propriété YAML dans le code suivant pour en voir la description.

 Note

La plupart des propriétés YAML suivantes ont des éléments d'interface utilisateur correspondants dans l'éditeur visuel. Pour rechercher un élément de l'interface utilisateur, utilisez Ctrl+F. L'élément sera répertorié avec sa propriété YAML associée.

```
# The workflow definition starts here.
# See Propriétés de haut niveau for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
action-name:
  Identifier: aws/github-actions-runner@v1
  DependsOn:
    - dependent-action-name-1
  Compute:
    Fleet: fleet-name
  Timeout: timeout-minutes
  Environment:
    Name: environment-name
  Connections:
    - Name: account-connection-name
      Role: iam-role-name
  Inputs:
    Sources:
      - source-name-1
      - source-name-2
    Artifacts:
      - artifact-name
  Variables:
    - Name: variable-name-1
      Value: variable-value-1
    - Name: variable-name-2
      Value: variable-value-2
  Outputs:
    Artifacts:
```

- **Name:** *output-artifact-1*  
**Files:**
  - github-output/artifact-1.jar
  - "github-output/build\*"
- **Name:** *output-artifact-2*  
**Files:**
  - github-output/artifact-2.1.jar
  - github-output/artifact-2.2.jar

**Variables:**

- *variable-name-1*
- *variable-name-2*

**AutoDiscoverReports:****Enabled:** *true | false***ReportNamePrefix:** *AutoDiscovered***IncludePaths:**

- *\*\*/\**

**ExcludePaths:**

- *node\_modules/cdk/junit.xml*

**SuccessCriteria:****PassRate:** *percent***LineCoverage:** *percent***BranchCoverage:** *percent***Vulnerabilities:****Severity:** *CRITICAL|HIGH|MEDIUM|LOW|INFORMATIONAL***Number:** *whole-number***Reports:*****report-name-1:*****Format:** *format***IncludePaths:**

- *\*.xml*

**ExcludePaths:**

- *report2.xml*
- *report3.xml*

**SuccessCriteria:****PassRate:** *percent***LineCoverage:** *percent***BranchCoverage:** *percent***Vulnerabilities:****Severity:** *CRITICAL|HIGH|MEDIUM|LOW|INFORMATIONAL***Number:** *whole-number***Configuration****Steps:**

- *github-actions-code*

## nom-action

(Obligatoire)

Spécifiez le nom de l'action. Tous les noms d'action doivent être uniques dans le flux de travail. Les noms d'action sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms d'action.

*Interface utilisateur correspondante : onglet de configuration/nom d'action*

## Identifiant

*(nom-action/)* Identifiant

Identifie l'action. Ne modifiez pas cette propriété, sauf si vous souhaitez modifier la version. Pour de plus amples informations, veuillez consulter [Utilisation des versions d'action](#).

À utiliser `aws/github-actions-runner@v1` pour GitHub Actions Actions.

Interface utilisateur correspondante : diagramme de flux de *travail/nom d'action/étiquette aws/ @v1 github-actions-runner*

## DependsOn

*(nom-action/)* DependsOn

(Facultatif)

Spécifiez une action ou un groupe d'actions qui doit être exécuté correctement pour que cette action soit exécutée.

Pour plus d'informations sur la fonctionnalité « dépend », consultez. [Configuration des actions pour qu'elles dépendent d'autres actions](#)

Interface utilisateur correspondante : onglet Entrées/dépend de - facultatif

## Compute

*(nom-action/)* Compute

## (Facultatif)

Le moteur informatique utilisé pour exécuter les actions de votre flux de travail. Vous pouvez spécifier le calcul au niveau du flux de travail ou au niveau de l'action, mais pas les deux. Lorsqu'elle est spécifiée au niveau du flux de travail, la configuration de calcul s'applique à toutes les actions définies dans le flux de travail. Au niveau du flux de travail, vous pouvez également exécuter plusieurs actions sur la même instance. Pour de plus amples informations, veuillez consulter [Partage du calcul entre les actions](#).

Interface utilisateur correspondante : aucune

## Fleet

(nom de *l'action /Compute/ Fleet*)

### (Facultatif)

Spécifiez la machine ou le parc qui exécutera votre flux de travail ou vos actions de flux de travail. Dans le cas des flottes à la demande, lorsqu'une action démarre, le flux de travail fournit les ressources dont il a besoin et les machines sont détruites à la fin de l'action. Exemples de flottes à la demande :Linux.x86-64.Large,Linux.x86-64.XLarge. Pour plus d'informations sur les flottes à la demande, consultez[Propriétés de flotte à la demande](#).

Avec les flottes provisionnées, vous configurez un ensemble de machines dédiées pour exécuter les actions de votre flux de travail. Ces machines restent inactives, prêtes à exécuter des actions immédiatement. Pour plus d'informations sur les flottes provisionnées, consultez. [Propriétés du parc provisionné](#)

S'il Fleet est omis, la valeur par défaut estLinux.x86-64.Large.

Interface utilisateur correspondante : onglet Configuration/parc de calcul - facultatif

## Timeout

(*nom-action/*) Timeout

### (Facultatif)

Spécifiez la durée en minutes (éditeur YAML) ou en heures et minutes (éditeur visuel) pendant laquelle l'action peut être exécutée avant la CodeCatalyst fin de l'action. Le minimum est de 5

minutes et le maximum est décrit dans [Quotas pour les flux de travail dans CodeCatalyst](#). Le délai d'expiration par défaut est le même que le délai d'expiration maximal.

Interface utilisateur correspondante : onglet Configuration/Délai d'expiration - facultatif

## Environment

(*nom-action/*) Environment

(Facultatif)

Spécifiez l' CodeCatalyst environnement à utiliser avec l'action.

Pour plus d'informations sur les environnements, reportez-vous [Utilisation des environnements](#) aux sections et [Création d'un environnement](#).

Interface utilisateur correspondante : onglet Configuration/Environnement/Compte/Rôle

## Name

(nom de *l'action /Environment/ Name*)

(Obligatoire s'[Environment](#)il est inclus)

Spécifiez le nom de l'environnement existant que vous souhaitez associer à l'action.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role/'  
Environnement

## Connections

(nom de *l'action /Environment/ Connections*)

(Obligatoire s'[Environment](#)il est inclus)

Spécifiez la connexion au compte à associer à l'action. Vous pouvez spécifier un maximum d'une connexion à un compte sous `Environment`.

Pour plus d'informations sur les connexions aux comptes, consultez [Administration Comptes AWS d'un espace](#). Pour plus d'informations sur la façon d'associer une connexion à un compte à votre environnement, consultez [Création d'un environnement](#).

Interface utilisateur correspondante : aucune

## Name

(nom de *l'action /Environment/Connections/ Name*)

(Facultatif)

Spécifiez le nom de la connexion au compte.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Role'/ connexion AWS au compte

## Role

(nom de *l'action /Environment/Connections/ Role*)

(Facultatif)

Spécifiez le nom du rôle IAM que cette action utilise pour accéder et opérer dans des AWS services tels qu'Amazon S3 et Amazon ECR. Assurez-vous que ce rôle est ajouté à la connexion de votre compte. Pour ajouter un rôle IAM à une connexion à un compte, consultez [Ajouter des rôles IAM aux connexions aux comptes](#).

### Note

Vous pouvez peut-être spécifier le nom du CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle ici, à condition qu'il dispose des autorisations suffisantes. Pour plus d'informations sur ce rôle, consultez [Création du CodeCatalystWorkflowDevelopmentRole-\*spaceName\* rôle pour votre compte et votre espace](#). Sachez que le CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle dispose d'autorisations très étendues, ce qui peut présenter un risque de sécurité. Nous vous recommandons de n'utiliser ce rôle que dans les didacticiels et les scénarios où la sécurité est moins préoccupante.

### Warning

Limitez les autorisations à celles requises par l'GitHub action Action. L'utilisation d'un rôle doté d'autorisations étendues peut présenter un risque de sécurité.

Interface utilisateur correspondante : onglet Configuration/'Environnement/Compte/Rôle'/ Rôle

## Inputs

(*nom-action/*) Inputs

(Facultatif)

La Inputs section définit les données dont une action a besoin lors de l'exécution d'un flux de travail.

### Note

Un maximum de quatre entrées (une source et trois artefacts) sont autorisées par action GitHub Actions. Les variables ne sont pas prises en compte dans ce total.

Si vous devez faire référence à des fichiers résidant dans différentes entrées (par exemple, une source et un artefact), l'entrée source est l'entrée principale et l'artefact est l'entrée secondaire. Les références aux fichiers dans les entrées secondaires utilisent un préfixe spécial pour les distinguer du fichier principal. Pour plus de détails, consultez [Exemple : Référencement de fichiers dans plusieurs artefacts](#).

Interface utilisateur correspondante : onglet Entrées

## Sources

(nom de *l'action /Inputs/ Sources*)

(Facultatif)

Spécifiez les étiquettes qui représentent les référentiels sources qui seront nécessaires à l'action. Actuellement, la seule étiquette prise en charge est `WorkflowSource`, qui représente le référentiel source dans lequel votre fichier de définition de flux de travail est stocké.

Si vous omettez une source, vous devez spécifier au moins un artefact d'entrée sous. *action-name/Inputs/Artifacts*

Pour plus d'informations sur les sources, consultez [Travailler avec des sources](#).



Interface utilisateur correspondante : onglet Entrées/Sources - facultatif

## Artifacts - input

(nom de *l'action /Inputs/ **Artifacts***)

(Facultatif)

Spécifiez les artefacts des actions précédentes que vous souhaitez fournir en entrée pour cette action. Ces artefacts doivent déjà être définis en tant qu'artefacts de sortie dans les actions précédentes.

Si vous ne spécifiez aucun artefact d'entrée, vous devez spécifier au moins un référentiel source sous *action-name/Inputs/Sources*.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

### Note

Si la liste déroulante Artéfacts - optionnelle n'est pas disponible (éditeur visuel), ou si vous recevez des erreurs lors de la validation de votre YAML (éditeur YAML), c'est peut-être parce que l'action ne prend en charge qu'une seule entrée. Dans ce cas, essayez de supprimer l'entrée source.

Interface utilisateur correspondante : onglet Entrées/Artefacts - facultatif

## Variables - input

(nom de *l'action /Inputs/ **Variables***)

(Facultatif)

Spécifiez une séquence de paires nom/valeur qui définissent les variables d'entrée que vous souhaitez mettre à la disposition de l'action. Les noms de variables sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms de variables.

Pour plus d'informations sur les variables, y compris des exemples, consultez [Utilisation des variables](#).

Interface utilisateur correspondante : onglet Entrées/Variables - facultatif

## Outputs

(*nom-action/*) Outputs

(Facultatif)

Définit les données produites par l'action lors de l'exécution d'un flux de travail.

Interface utilisateur correspondante : onglet Sorties

## Artifacts - output

(nom de *l'action /Outputs/ Artifacts*)

(Facultatif)

Spécifiez le nom d'un artefact généré par l'action. Les noms d'artefact doivent être uniques dans un flux de travail et sont limités aux caractères alphanumériques (a-z, A-Z, 0-9) et aux traits de soulignement (\_). Les espaces, les tirets (-) et les autres caractères spéciaux ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les espaces, les tirets et autres caractères spéciaux dans les noms d'artefacts en sortie.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

Interface utilisateur correspondante : onglet Sorties/Artefacts

## Name

(nom de *l'action /Outputs/Artifacts/ Name*)

(Obligatoire s'[Artifacts - output](#) est inclus)

Spécifiez le nom d'un artefact généré par l'action. Les noms d'artefact doivent être uniques dans un flux de travail et sont limités aux caractères alphanumériques (a-z, A-Z, 0-9) et aux traits de soulignement (\_). Les espaces, les tirets (-) et les autres caractères spéciaux ne sont pas autorisés.

Vous ne pouvez pas utiliser de guillemets pour activer les espaces, les tirets et autres caractères spéciaux dans les noms d'artefacts en sortie.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

Interface utilisateur correspondante : onglet Sorties/Artifacts/Ajouter un artefact/Créer un nom d'artefact

## Files

(nom de *l'action /Outputs/Artifacts/ Files*)

(Obligatoire s'[Artifacts - output](#) est inclus)


Spécifiez les fichiers CodeCatalyst inclus dans l'artefact généré par l'action. Ces fichiers sont générés par l'action du flux de travail lorsqu'elle s'exécute et sont également disponibles dans votre référentiel source. Les chemins de fichiers peuvent résider dans un référentiel source ou dans un artefact issu d'une action précédente, et sont relatifs au référentiel source ou à la racine de l'artefact. Vous pouvez utiliser des modèles globulaires pour définir des chemins. Exemples :

- Pour spécifier un seul fichier situé à la racine de votre emplacement de compilation ou de l'emplacement de votre référentiel source, utilisez `my-file.jar`.
- Pour spécifier un seul fichier dans un sous-répertoire, utilisez `directory/my-file.jar` ou `directory/subdirectory/my-file.jar`.
- Pour spécifier tous les fichiers, utilisez `**/*`. Le modèle `**` glob indique qu'il doit correspondre à un nombre quelconque de sous-répertoires.
- Pour spécifier tous les fichiers et répertoires d'un répertoire nommé `directory`, utilisez `directory/**/*`. Le modèle `**` glob indique qu'il doit correspondre à un nombre quelconque de sous-répertoires.
- Pour spécifier tous les fichiers d'un répertoire nommé `directory`, mais aucun de ses sous-répertoires, utilisez `directory/*`.

### Note

Si le chemin de votre fichier comporte un ou plusieurs astérisques (\*) ou autres caractères spéciaux, mettez-le entre guillemets (). Pour plus d'informations sur les caractères spéciaux, consultez [Consignes et conventions de syntaxe](#).

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

 Note

Vous devrez peut-être ajouter un préfixe au chemin du fichier pour indiquer dans quel artefact ou dans quelle source le trouver. Pour plus d'informations, consultez [Référencement de fichiers dans un référentiel source](#) et [Référencement de fichiers dans un artefact](#).

Interface utilisateur correspondante : onglet Sorties/Artifacts/Ajouter un artefact/Fichiers produits par build

## Variables - output

(nom de *l'action /Outputs/ Variables*)

(Facultatif)

Spécifiez les variables que vous souhaitez que l'action exporte afin qu'elles puissent être utilisées par les actions suivantes.

Pour plus d'informations sur les variables, y compris des exemples, consultez [Utilisation des variables](#).

Interface utilisateur correspondante : onglet Sorties/Variables/Ajouter une variable

nom-variable-1

(*nom-action-nom-variable-1*/Outputs/Variables)

(Facultatif)

Spécifiez le nom de la variable que vous souhaitez que l'action exporte. Cette variable doit déjà être définie dans la Steps section Inputs ou de la même action.

Pour plus d'informations sur les variables, y compris des exemples, consultez [Utilisation des variables](#).

Interface utilisateur correspondante : onglet Sorties/Variables/Ajouter une variable/Nom

## AutoDiscoverReports

(nom de *l'action /Outputs/ **AutoDiscoverReports***)

(Facultatif)

Définit la configuration de la fonctionnalité de découverte automatique.

Lorsque vous activez la découverte automatique, les CodeCatalyst recherches Inputs sont toutes transmises à l'action ainsi que tous les fichiers générés par l'action elle-même, à la recherche de rapports de test, de couverture de code et d'analyse de la composition logicielle (SCA). Pour chaque rapport trouvé, le CodeCatalyst transforme en CodeCatalyst rapport. Un CodeCatalyst rapport est un rapport qui est entièrement intégré au CodeCatalyst service et qui peut être consulté et manipulé via la CodeCatalyst console.

### Note

Par défaut, la fonction de découverte automatique inspecte tous les fichiers. Vous pouvez limiter les fichiers inspectés à l'aide des [ExcludePaths](#) propriétés [IncludePaths](#) or.

Interface utilisateur correspondante : aucune

## Enabled

(nom de *l'action /Outputs/AutoDiscoverReports/ **Enabled***)

(Facultatif)

Activez ou désactivez la fonction de découverte automatique.

Les valeurs valides sont true ou false.

S'il Enabled est omis, la valeur par défaut est true.

Interface utilisateur correspondante : onglet Sorties/Rapports/Découverte automatique des rapports

## ReportNamePrefix

(nom de *l'action /Outputs/AutoDiscoverReports/ **ReportNamePrefix***)

(Obligatoire s'[AutoDiscoverReports](#) est inclus et activé)

Spécifiez un préfixe qui CodeCatalyst précède tous les rapports trouvés afin de nommer les rapports associés. CodeCatalyst Par exemple, si vous spécifiez le préfixe `deAutoDiscovered`, et que vous CodeCatalyst découvrez automatiquement deux rapports de `testTestSuiteTwo.xml`, `TestSuiteOne.xml` et, les CodeCatalyst rapports associés seront nommés `AutoDiscoveredTestSuiteOne` et `AutoDiscoveredTestSuiteTwo`

Interface utilisateur correspondante : onglet Sorties/Rapports/Découverte automatique des rapports/  
Préfixe du rapport

## IncludePaths

(nom de *l'action* `/Outputs/AutoDiscoverReports/ IncludePaths`)

Ou

*(/Outputs/Reports/nom-action-nom-rapport-1/)* `IncludePaths`

(Obligatoire s'[AutoDiscoverReports](#) est inclus et activé, ou s'il [Reports](#) est inclus)

Spécifiez les fichiers et les chemins de fichiers CodeCatalyst inclus lors de la recherche de rapports bruts. Par exemple, si vous le spécifiez `"/test/report/*"`, CodeCatalyst recherche l'intégralité de [l'image de construction](#) utilisée par l'action à la recherche du `/test/report/*` répertoire. Lorsqu'il trouve ce répertoire, CodeCatalyst il recherche des rapports dans ce répertoire.

### Note

Si le chemin de votre fichier comporte un ou plusieurs astérisques (\*) ou autres caractères spéciaux, mettez-le entre guillemets (). "" Pour plus d'informations sur les caractères spéciaux, consultez [Consignes et conventions de syntaxe](#).

Si cette propriété est omise, la valeur par défaut est `**/*`, ce qui signifie que la recherche inclut tous les fichiers, quel que soit leur chemin.

### Note

Pour les rapports configurés manuellement, `IncludePaths` il doit s'agir d'un modèle global correspondant à un seul fichier.

Interface utilisateur correspondante :

- Onglet Sorties/Rapports/Découverte automatique des rapports/« Inclure/Exclure les chemins »/ Inclure les chemins
- **Onglet Sorties/Rapports/Configuration manuelle des rapports/nom-du-rapport-1 /'Inclure/Exclure les chemins'/Inclure les chemins**

## ExcludePaths

(nom de l'action /Outputs/AutoDiscoverReports/ **ExcludePaths**)

Ou

(/Outputs/Reports/nom-action-nom-rapport-1/) ExcludePaths

(Facultatif)

Spécifiez les fichiers et les chemins de fichiers à CodeCatalyst exclure lors de la recherche de rapports bruts. Par exemple, si vous le spécifiez "/test/my-reports/\*\*/\*", ne CodeCatalyst recherchera pas de fichiers dans le /test/my-reports/ répertoire. Pour ignorer tous les fichiers d'un répertoire, utilisez le modèle \*/\* glob.

### Note

Si le chemin de votre fichier comporte un ou plusieurs astérisques (\*) ou autres caractères spéciaux, mettez-le entre guillemets (). "" Pour plus d'informations sur les caractères spéciaux, consultez [Consignes et conventions de syntaxe](#).

Interface utilisateur correspondante :

- Onglet Sorties/Rapports/Découverte automatique des rapports/« Inclure/Exclure des chemins »/ Exclure des chemins
- **Onglet Sorties/Rapports/Configuration manuelle des rapports/nom-de-rapport-1 /'Inclure/Exclure des chemins'/Exclure des chemins**

## SuccessCriteria

(nom de l'action /Outputs/AutoDiscoverReports/ **SuccessCriteria**)

Ou

*(/Outputs/Reports/nom-action-nom-rapport-1/)* SuccessCriteria

(Facultatif)

Spécifiez les critères de réussite pour les rapports de test, de couverture du code, d'analyse de la composition logicielle (SCA) et d'analyse statique (SA).

Pour de plus amples informations, veuillez consulter [Configuration des critères de réussite pour les rapports](#).

Interface utilisateur correspondante :

- Onglet Produits/Rapports/Découverte automatique des rapports/Critères de réussite
- **Onglet Sorties/Rapports/Configuration manuelle des rapports/nom-du-rapport-1//Critères de réussite**

PassRate

(nom de *l'action /Outputs/AutoDiscoverReports/SuccessCriteria/ PassRate*)

Ou

*(nom-action-nom-rapport-1/Outputs/Reports/) /SuccessCriteria/ PassRate*

(Facultatif)

Spécifiez le pourcentage de tests dans un rapport de test qui doivent être réussis pour que le CodeCatalyst rapport associé soit marqué comme réussi. Les valeurs valides incluent les nombres décimaux. Par exemple, 50, 60.5. Les critères de taux de réussite ne sont appliqués qu'aux rapports de test. Pour plus d'informations sur les rapports de test, consultez [Rapports d'essais](#).

Interface utilisateur correspondante :

- Onglet Produits/Rapports/Découverte automatique des rapports/Critères de succès/Taux de réussite
- **Onglet Sorties/Rapports/Configuration manuelle des rapports/nom-du-rapport-1 /Critères de succès/Taux de réussite**



## LineCoverage

(nom de l'action */Outputs/AutoDiscoverReports/SuccessCriteria/ **LineCoverage***)

Ou

*(nom-action-nom-rapport-1/Outputs/Reports/) /SuccessCriteria/ **LineCoverage***

(Facultatif)

Spécifiez le pourcentage de lignes d'un rapport de couverture de code qui doivent être couvertes pour que le CodeCatalyst rapport associé soit marqué comme transmis. Les valeurs valides incluent les nombres décimaux. Par exemple, 50, 60.5. Les critères de couverture de ligne sont appliqués uniquement aux rapports de couverture de code. Pour plus d'informations sur les rapports de couverture du code, consultez [Rapports sur la couverture du code](#).

Interface utilisateur correspondante :

- Onglet Produits/Rapports/Découverte automatique des rapports/Critères de succès/Couverture des lignes
- **Onglet Sorties/Rapports/Configuration manuelle des rapports/nom-du-rapport-1 /Critères de succès/Couverture des lignes**

## BranchCoverage

(nom de l'action */Outputs/AutoDiscoverReports/SuccessCriteria/ **BranchCoverage***)

Ou

*(nom-action-nom-rapport-1/Outputs/Reports/) /SuccessCriteria/ **BranchCoverage***

(Facultatif)

Spécifiez le pourcentage de branches dans un rapport de couverture de code qui doivent être couvertes pour que le CodeCatalyst rapport associé soit marqué comme transmis. Les valeurs valides incluent les nombres décimaux. Par exemple, 50, 60.5. Les critères de couverture des succursales sont appliqués uniquement aux rapports de couverture par code. Pour plus d'informations sur les rapports de couverture du code, consultez [Rapports sur la couverture du code](#).

Interface utilisateur correspondante :

- Onglet Produits/Rapports/Découverte automatique des rapports/Critères de succès/Couverture des succursales
- **Onglet Sorties/Rapports/Configuration manuelle des rapports/nom-du-rapport-1 /Critères de succès/Couverture des succursales**

## Vulnerabilities

(nom de *l'action /Outputs/AutoDiscoverReports/SuccessCriteria/**Vulnerabilities***)

Ou

(*nom-action-nom-rapport-1/Outputs/Reports/*) */SuccessCriteria/**Vulnerabilities***

(Facultatif)

Spécifiez le nombre maximum et la gravité des vulnérabilités autorisées dans le rapport SCA pour que le CodeCatalyst rapport associé soit marqué comme transmis. Pour définir les vulnérabilités, vous devez spécifier :

- La gravité minimale des vulnérabilités que vous souhaitez inclure dans le décompte. Les valeurs valides, de la plus sévère à la moins sévère, sont les suivantes : CRITICALHIGH,MEDIUM,LOW,,INFORMATIONAL.

Par exemple, si vous le souhaitezHIGH, HIGH les CRITICAL vulnérabilités seront comptabilisées.

- Le nombre maximum de vulnérabilités de la gravité spécifiée que vous souhaitez autoriser. Le dépassement de ce nombre entraîne le marquage CodeCatalyst du rapport comme ayant échoué. Les valeurs valides sont des nombres entiers.

Les critères de vulnérabilité ne sont appliqués qu'aux rapports SCA. Pour plus d'informations sur les rapports SCA, consultez[Rapports d'analyse de la composition des logiciels](#).

Pour spécifier la sévérité minimale, utilisez la `Severity` propriété. Pour spécifier le nombre maximum de vulnérabilités, utilisez la `Number` propriété.

Pour plus d'informations sur les rapports SCA, consultez[Types de rapports de test](#).

Interface utilisateur correspondante :

- Onglet Produits/Rapports/Découverte automatique des rapports/Critères de succès/Vulnérabilités
- ***Onglet Sorties/Rapports/Configuration manuelle des rapports/nom-du-rapport-1 /Critères de succès/Vulnérabilités***

## Reports

(nom de *l'action /Outputs/ Reports*)

(Facultatif)

Section qui spécifie la configuration des rapports de test.

Interface utilisateur correspondante : onglet Sorties/Rapports

nom-rapport-1

***(nom-action-nom-rapport-1/Outputs/Reports/)***

(Obligatoire s'[Reports](#)il est inclus)

Le nom que vous souhaitez donner au CodeCatalyst rapport qui sera généré à partir de vos rapports bruts.

Interface utilisateur correspondante : onglet Sorties/Rapports/Configuration manuelle des rapports/

Nom du rapport

## Format

***(/Outputs/Reports/nom-action-nom-rapport-1/)*** Format

(Obligatoire s'[Reports](#)il est inclus)

Spécifiez le format de fichier que vous utilisez pour vos rapports. Les valeurs possibles sont les suivantes.

- Pour les rapports de test :
  - Pour Cucumber JSON, spécifiez Cucumber (éditeur visuel) ou CUCUMBERJSON (éditeur YAML).
  - Pour JUnit XML, spécifiez JUnit (éditeur visuel) ou JUNITXML (éditeur YAML).
  - Pour NUnit XML, spécifiez NUnit (éditeur visuel) ou NUNITXML (éditeur YAML).

- Pour NUnit 3 XML, spécifiez NUnit3 (éditeur visuel) ou NUNIT3XML (éditeur YAML).
- Pour Visual Studio TRX, spécifiez Visual Studio TRX (éditeur visuel) ou VISUALSTUDIOTRX (éditeur YAML).
- Pour TestNG XML, spécifiez TestNG (éditeur visuel) ou TESTNGXML (éditeur YAML).
- Pour les rapports sur la couverture du code :
  - Pour Clover XML, spécifiez Clover (éditeur visuel) ou CLOVERXML (éditeur YAML).
  - Pour Cobertura XML, spécifiez Cobertura (éditeur visuel) ou COBERTURAXML (éditeur YAML).
  - Pour le JaCoCo XML, spécifiez JaCoCo(éditeur visuel) ou JACOCOXML (éditeur YAML).
  - Pour le SimpleCov JSON généré par [simplecov](#), et non par [simplecov-json](#), spécifiez Simplecov (éditeur visuel) ou (éditeur YAML). SIMPLECOV
- Pour les rapports d'analyse de composition logicielle (SCA) :
  - Pour SARIF, spécifiez SARIF (éditeur visuel) ou SARIFSCA (éditeur YAML).

***Interface utilisateur correspondante : onglet Sorties/Rapports/ Configuration manuelle des rapports/Ajouter un rapport/nom-du-rapport-1 / Type de rapport et format de rapport***

## Configuration

*(nom-action/)* Configuration

(Obligatoire) Section dans laquelle vous pouvez définir les propriétés de configuration de l'action.

Interface utilisateur correspondante : onglet Configuration

## Steps

(nom de *l'action /Configuration/ Steps*)

(Obligatoire)

Spécifiez votre code GitHub d'action tel qu'il apparaît sur la page de détails de l'action dans [GitHub Marketplace](#). Ajoutez le code en suivant ces instructions :

1. Collez le code de la steps : section GitHub Action dans la Steps : section du CodeCatalyst flux de travail. Le code commence par un tiret (-) et ressemble à ce qui suit.

GitHub code à coller :

```
- name: Lint Code Base
  uses: github/super-linter@v4
  env:
    VALIDATE_ALL_CODEBASE: false
    DEFAULT_BRANCH: master
    GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN }
```

2. Passez en revue le code que vous venez de coller et modifiez-le si nécessaire afin qu'il soit conforme aux CodeCatalyst normes. Par exemple, avec le bloc de code précédent, vous pouvez supprimer le code en *italique rouge* et ajouter le code en gras.

CodeCatalyst flux de travail yaml :

```
Steps:
  - name: Lint Code Base
    uses: github/super-linter@v4
    env:
      VALIDATE_ALL_CODEBASE: false
      DEFAULT_BRANCH: mastermain
      GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN }
```

3. Pour le code supplémentaire inclus dans l' GitHub action mais qui n'existe pas dans la steps : section, ajoutez-le au CodeCatalyst flux de travail à l'aide d'un code CodeCatalyst équivalent. Vous pouvez les consulter [Référence de définition du flux de travail](#) pour avoir un aperçu de la manière dont vous pourriez porter votre GitHub code CodeCatalyst. Les étapes de migration détaillées ne sont pas abordées dans ce guide.

Voici un exemple de la manière de spécifier des chemins de fichiers dans une action GitHub Actions :

```
Steps:
  - name: Lint Code Base
    uses: github/super-linter@v4
    ...
  - run: cd /sources/WorkflowSource/MyFolder/ && cat file.txt
  - run: cd /artifacts/MyGitHubAction/MyArtifact/MyFolder/ && cat file2.txt
```

Pour plus d'informations sur la spécification des chemins de fichiers, reportez-vous [Référencement de fichiers dans un référentiel source](#) aux sections et [Référencement de fichiers dans un artefact](#).

Interface utilisateur correspondante : onglet GitHub Configuration/Actions YAML

## Référence d'action « Render la définition de la tâche Amazon ECS »

Vous trouverez ci-dessous la référence YAML de définition d'action pour l'action de définition de tâche Render Amazon ECS. Pour savoir comment utiliser cette action, consultez [Ajout de l'action « Render la définition de tâche Amazon ECS »](#).

### Note

La plupart des propriétés YAML suivantes ont des éléments d'interface utilisateur correspondants dans l'éditeur visuel. Pour rechercher un élément de l'interface utilisateur, utilisez Ctrl+F. L'élément sera répertorié avec sa propriété YAML associée.

```
# The workflow definition starts here.
# See Propriétés de haut niveau for details.

Name: MyWorkflow
SchemaVersion: 1.0
Actions:

# The action definition starts here.
ECSRenderTaskDefinition\_nn:
  Identifiant: aws/ecs-render-task-definition@v1
  DependsOn:
    - build-action
  Compute:
    Type: EC2 | Lambda
    Fleet: fleet-name
    Timeout: timeout-minutes
  Inputs:
    # Specify a source or an artifact, but not both.
    Sources:
      - source-name-1
    Artifacts:
      - task-definition-artifact
    Variables:
      - Name: variable-name-1
        Value: variable-value-1
      - Name: variable-name-2
        Value: variable-value-2
  Configuration
    task-definition: task-definition-path
```

```

container-definition-name: container-definition-name
image: docker-image-name
environment-variables:
  - variable-name-1=variable-value-1
  - variable-name-2=variable-value-2
Outputs:
Artifacts:
  - Name: TaskDefArtifact
    Files: "task-definition*"
Variables:
  - task-definition

```

## ECSRenderTaskDefinition

(Obligatoire)

Spécifiez le nom de l'action. Tous les noms d'action doivent être uniques dans le flux de travail. Les noms d'action sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms d'action.

Par défaut: ECSRenderTaskDefinition\_nn.

Interface utilisateur correspondante : onglet Configuration/nom de l'action

### Identifiant

(*ECSRenderTaskDefinition*/Identifiant)

(Obligatoire)

Identifie l'action. Ne modifiez pas cette propriété, sauf si vous souhaitez modifier la version. Pour de plus amples informations, veuillez consulter [Utilisation des versions d'action](#).

Par défaut: aws/ecs-render-task-definition@v1.

Interface utilisateur correspondante : diagramme de flux de travail/ ECSRenderTaskDefinition\_nn/ aws/ @v1 label ecs-render-task-definition

### DependsOn

(*ECSRenderTaskDefinition*/DependsOn)

## (Facultatif)

Spécifiez une action ou un groupe d'actions qui doit être exécuté correctement pour que cette action soit exécutée.

Pour plus d'informations sur la fonctionnalité « dépend », consultez. [Configuration des actions pour qu'elles dépendent d'autres actions](#)

Interface utilisateur correspondante : onglet Entrées/dépend de - facultatif

## Compute

(*ECSRenderTaskDefinition*/Compute)

### (Facultatif)

Le moteur informatique utilisé pour exécuter les actions de votre flux de travail. Vous pouvez spécifier le calcul au niveau du flux de travail ou au niveau de l'action, mais pas les deux. Lorsqu'elle est spécifiée au niveau du flux de travail, la configuration de calcul s'applique à toutes les actions définies dans le flux de travail. Au niveau du flux de travail, vous pouvez également exécuter plusieurs actions sur la même instance. Pour de plus amples informations, veuillez consulter [Partage du calcul entre les actions](#).

Interface utilisateur correspondante : aucune

## Type

(*ECSRenderTaskDefinition*/Compute/Type)

(Obligatoire s'[Compute](#) est inclus)

Type de moteur de calcul. Vous pouvez utiliser l'une des valeurs suivantes :

- EC2 (éditeur visuel) ou EC2 (éditeur YAML)  
Optimisé pour la flexibilité lors des courses d'action.
- Lambda (éditeur visuel) ou Lambda (éditeur YAML)  
Vitesses de démarrage des actions optimisées.

Pour plus d'informations sur les types de calcul, veuillez consulter [À propos des types de calcul](#).



Interface utilisateur correspondante : onglet Configuration/Type de calcul

## Fleet

(*ECSRenderTaskDefinition*/Compute/Fleet)

(Facultatif)

Spécifiez la machine ou le parc qui exécutera votre flux de travail ou vos actions de flux de travail. Dans le cas des flottes à la demande, lorsqu'une action démarre, le flux de travail fournit les ressources dont il a besoin et les machines sont détruites à la fin de l'action. Exemples de flottes à la demande : `Linux.x86-64.Large`, `Linux.x86-64.XLarge`. Pour plus d'informations sur les flottes à la demande, consultez [Propriétés de flotte à la demande](#).

Avec les flottes provisionnées, vous configurez un ensemble de machines dédiées pour exécuter les actions de votre flux de travail. Ces machines restent inactives, prêtes à exécuter des actions immédiatement. Pour plus d'informations sur les flottes provisionnées, consultez [Propriétés du parc provisionné](#)

S'il Fleet est omis, la valeur par défaut est `Linux.x86-64.Large`.

Interface utilisateur correspondante : onglet Configuration/parc de calcul

## Timeout

(*ECSRenderTaskDefinition*/Timeout)

(Facultatif)

Spécifiez la durée en minutes (éditeur YAML) ou en heures et minutes (éditeur visuel) pendant laquelle l'action peut être exécutée avant la CodeCatalyst fin de l'action. Le minimum est de 5 minutes et le maximum est décrit dans [Quotas pour les flux de travail dans CodeCatalyst](#). Le délai d'expiration par défaut est le même que le délai d'expiration maximal.

Interface utilisateur correspondante : onglet Configuration/Délai d'expiration - facultatif

## Inputs

(*ECSRenderTaskDefinition*/Inputs)

(Facultatif)

La `Inputs` section définit les données dont ils ont `ECSRenderTaskDefinition` besoin lors de l'exécution d'un flux de travail.

#### Note

Une seule entrée (source ou artefact) est autorisée par action de définition de tâche Render Amazon ECS. Les variables ne sont pas prises en compte dans ce total.

Interface utilisateur correspondante : onglet Entrées

## Sources

(*ECSRenderTaskDefinition*/Inputs/Sources)

(Obligatoire si votre fichier de définition de tâche est stocké dans un référentiel source)

Si votre fichier de définition de tâche est stocké dans un référentiel source, spécifiez l'étiquette de ce référentiel source. Actuellement, la seule étiquette prise en charge est `WorkflowSource`.

Si votre fichier de définition de tâche n'est pas contenu dans un référentiel source, il doit résider dans un artefact généré par une autre action.

Pour plus d'informations sur les sources, consultez [Travailler avec des sources](#).

Interface utilisateur correspondante : onglet Entrées/Sources - facultatif

## Artifacts - input

(*ECSRenderTaskDefinition*/Inputs/Artifacts)

(Obligatoire si votre fichier de définition de tâche est stocké dans un [artefact de sortie](#) d'une action précédente)

Si le fichier de définition de tâche que vous souhaitez déployer est contenu dans un artefact généré par une action précédente, spécifiez cet artefact ici. Si votre fichier de définition de tâche n'est pas contenu dans un artefact, il doit résider dans votre référentiel source.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

Interface utilisateur correspondante : onglet Configuration/Artefacts - facultatif

## Variables - input

(*ECSRenderTaskDefinition*/Inputs/Variables)

(Obligatoire)

Spécifiez une séquence de paires nom/valeur qui définissent les variables d'entrée que vous souhaitez mettre à la disposition de l'action. Les noms de variables sont limités aux caractères alphanumériques (a-z, A-Z, 0-9), aux tirets (-) et aux traits de soulignement (\_). Les espaces ne sont pas autorisés. Vous ne pouvez pas utiliser de guillemets pour activer les caractères spéciaux et les espaces dans les noms de variables.

Pour plus d'informations sur les variables, y compris des exemples, consultez [Utilisation des variables](#).

Interface utilisateur correspondante : onglet Entrées/Variables - facultatif

## Configuration

(*ECSRenderTaskDefinition*/Configuration)

(Obligatoire)

Section dans laquelle vous pouvez définir les propriétés de configuration de l'action.

Interface utilisateur correspondante : onglet Configuration

## task-definition

(*ECSRenderTaskDefinition*/Configuration/task-definition)

(Obligatoire)

Spécifiez le chemin d'accès à un fichier de définition de tâche existant. Si le fichier se trouve dans votre dépôt source, le chemin est relatif au dossier racine du dépôt source. Si votre fichier réside dans un artefact issu d'une action de flux de travail précédente, le chemin est relatif au dossier racine de l'artefact. Pour plus d'informations sur les fichiers de définition de tâches, consultez la section [Définitions de tâches](#) dans le manuel Amazon Elastic Container Service Developer Guide.

Interface utilisateur correspondante : onglet Configuration/Définition des tâches

## container-definition-name

(*ECSRenderTaskDefinition*/Configuration/container-definition-name)

(Obligatoire)

Spécifiez le nom du conteneur dans lequel votre image Docker sera exécutée. Vous pouvez trouver ce nom dans le `name` champ `containerDefinitions`, de votre fichier de définition de tâche. Pour plus d'informations, consultez la section [Nom](#) dans le manuel Amazon Elastic Container Service Developer Guide.

Interface utilisateur correspondante : onglet Configuration/nom du conteneur

image

(*ECSRenderTaskDefinition*/Configuration/image)

(Obligatoire)

Spécifiez le nom de l'image Docker que l'action de définition de tâche Render Amazon ECS doit ajouter à votre fichier de définition de tâche. L'action ajoute ce nom au `image` champ `containerDefinitions`, de votre fichier de définition de tâche. Si une valeur existe déjà dans le `image` champ, l'action la remplace. Vous pouvez inclure des variables dans le nom de l'image.

Exemples :

Si vous le spécifiez `MyDockerImage:${WorkflowSource.CommitId}`, l'action est ajoutée `MyDockerImage:commit-id` au fichier de définition de tâche, où *commit-id est un identifiant* de validation généré lors de l'exécution par le flux de travail.

*Si vous le spécifiez `my-ecr-repo/image-repo:${date +%m-%d-%y-%H-%m-%s}`, l'action ajoute `my-ecr-repo/image-repo : date +%m-%d-%y-%h-%m-%s` au fichier de définition de la tâche, où `my-ecr-repo` est l'URI d'un Amazon Elastic Container Registry (ECR) et la date `+%m-%d-%d-%y-%h-%m-%s` est un horodatage au format généré lors de l'exécution par le flux de travail. month-day-year-hour-minute-second*

Pour plus d'informations sur ce `image` champ, consultez l'[image](#) dans le manuel Amazon Elastic Container Service Developer Guide. Pour plus d'informations sur les variables, consultez [Utilisation des variables](#).

Interface utilisateur correspondante : onglet Configuration/nom de l'image

## environment-variables

(*ECSRenderTaskDefinition*/Configuration/environment-variables)

(Obligatoire)

Spécifiez les variables d'environnement que l'action Render Amazon ECS doit ajouter à votre fichier de définition de tâche. L'action ajoute les variables au `environment` champ `containerDefinitions`, de votre fichier de définition de tâche. Si des variables existent déjà dans le fichier, l'action remplace les valeurs des variables existantes et ajoute de nouvelles variables. Pour plus d'informations sur les variables d'environnement Amazon ECS, consultez la section [Spécification des variables d'environnement](#) dans le manuel Amazon Elastic Container Service Developer Guide.

Interface utilisateur correspondante : onglet Configuration/variables d'environnement - facultatif

## Outputs

(*ECSRenderTaskDefinition*/Outputs)

(Obligatoire)

Définit les données produites par l'action lors de l'exécution d'un flux de travail.

Interface utilisateur correspondante : onglet Sorties

## Artifacts

(*ECSRenderTaskDefinition*/Outputs/Artifacts)

(Obligatoire)

Spécifiez les artefacts générés par l'action. Vous pouvez faire référence à ces artefacts en tant qu'entrée dans d'autres actions.

Pour plus d'informations sur les artefacts, y compris des exemples, consultez [Travailler avec des artefacts](#).

Interface utilisateur correspondante : onglet Sorties/Artefacts

## Name

(*ECSRenderTaskDefinition*/Outputs/Artifacts/Name)

### (Obligatoire)

Spécifiez le nom de l'artefact qui contiendra le fichier de définition de tâche mis à jour. La valeur par défaut est `MyTaskDefinitionArtifact`. Vous devez ensuite spécifier cet artefact comme entrée dans l'action Deploy to Amazon ECS. Pour comprendre comment ajouter cet artefact en tant qu'entrée à l'action Deploy to Amazon ECS, consultez [Exemple de flux de travail](#).

Interface utilisateur correspondante : onglet Sorties/Artefacts/Nom

## Files

(*ECSRenderTaskDefinition*/Outputs/Artefacts/Files)

### (Obligatoire)

Spécifiez les fichiers à inclure dans l'artefact. Vous devez spécifier `task-definition-*` que le fichier de définition de tâche mis à jour, qui commence par `task-definition-`, sera inclus.

Interface utilisateur correspondante : onglet Sorties/Artefacts/Fichiers

## Variables

(*ECSRenderTaskDefinition*/Outputs/Variables)

### (Obligatoire)

Spécifiez le nom d'une variable à définir par l'action de rendu. L'action de rendu définira la valeur de cette variable sur le nom du fichier de définition de tâche mis à jour (par exemple, `task-definition-random-string.json`). Vous devez ensuite spécifier cette variable dans la propriété de définition de tâche (éditeur visuel) ou **task-definition** (éditeur yaml) de l'action Deploy to Amazon ECS. Pour comprendre comment ajouter cette variable à l'action Deploy to Amazon ECS, consultez [Exemple de flux de travail](#).


Par défaut : `task-definition`

Interface utilisateur correspondante : onglet Sorties/Variables/Champ de nom

## Quotas pour les flux de travail dans CodeCatalyst

Le tableau suivant décrit les quotas et les limites des flux de travail sur Amazon CodeCatalyst.

Pour plus d'informations sur les quotas sur Amazon CodeCatalyst, consultez [Quotas pour CodeCatalyst](#).

|  |  |
|--|--|
| Nombre maximum de flux de travail par espace   | 800  |
| Taille maximale du fichier de définition du flux de travail  | 256 Ko   |
| Nombre maximal de fichiers de flux de travail traités lors d'un événement source unique                                | 50   |
| Nombre maximal de fichiers traités lors d'un événement source unique   | 4 000  |
| Nombre maximum de flottes actives par espace   | 10   |
| Nombre maximal d'instances de calcul actives par parc  | 20   |
| Nombre maximum d'artefacts d'entrée par action   | 10   |
| Nombre maximum d'artefacts de sortie par action  | 10   |
| Taille totale maximale des variables de sortie d'une seule action  | 120 KO   |
| Longueur maximale d'une valeur variable en sortie  | 500 caractères ou plus, selon l'action qui émet la valeur.   |
|  | <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> <b>Note</b><br/>Les valeurs peuvent être tronquées si elles dépassent la limite de l'action.</p> </div> |
| Nombre maximal de jours pendant lesquels les artefacts générés pendant l'exécution d'un flux de travail sont conservés | 30   |

|  |   |
|--|---|
| Nombre maximum de rapports par action  | 50  |
| Nombre maximum de cas de test par rapport de test                              | 20 000  |
| Nombre maximum de fichiers par rapport de couverture de code                   | 20 000  |
| Nombre maximum de résultats d'analyse de la composition logicielle par rapport | 20 000  |
| Nombre maximum de fichiers par rapport d'analyse statique                      | 20 000  |
| Nombre maximal d'exécutions de flux de travail simultanées par espace          | 100   |
| Nombre maximum d'actions par flux de travail                                   | 50  |
| Nombre maximum d'actions exécutées simultanément par flux de travail           | 50  |
| Nombre maximum d'actions exécutées simultanément par espace                    | 200   |
| Durée maximale pendant laquelle une action peut être exécutée                  | <p>Pour les actions de génération et de test, le délai d'expiration est de 8 heures.</p> <p>Pour toutes les autres actions, le délai d'expiration est de 1 heure.</p> |
| Nombre maximum d'environnements associés à un Compte AWS par espace            | 5 000   |
| Nombre maximum de secrets par action   | 5   |
| Nombre maximum de secrets par espace   | 500 000   |



# Problèmes liés à CodeCatalyst

Dans CodeCatalyst, vous pouvez surveiller les fonctionnalités, les bogues et tout autre travail impliqué dans votre projet. Chaque œuvre est conservée dans un dossier distinct appelé numéro. Vous pouvez diviser un problème en objectifs plus petits en y ajoutant une liste de tâches. Chaque problème peut avoir une description, un responsable, un statut et d'autres propriétés, que vous pouvez rechercher, regrouper et filtrer. Vous pouvez consulter vos problèmes à l'aide des vues par défaut ou créer vos propres vues avec un filtrage, un tri ou un regroupement personnalisés. Pour plus d'informations sur les concepts liés aux problèmes, consultez [Concepts des problèmes](#). Pour savoir comment créer votre premier numéro, consultez [Création d'un problème dans CodeCatalyst](#).

Voici un flux de travail possible pour une équipe confrontée à des problèmes :

Jorge Souza est un développeur travaillant sur un projet. Lui et les autres membres du projet Li Juan, Mateo Jackson et Wang Xiulan collaborent pour déterminer le travail à effectuer. Chaque jour, lui et ses collègues développeurs tiennent une réunion de synchronisation, dirigée par Wang Xiulan. Ils ouvrent le tableau en accédant à la vue du tableau de l'une de leurs équipes. En créant des vues, les utilisateurs et les équipes peuvent enregistrer des filtres, des regroupements et un tri des problèmes afin de visualiser facilement les problèmes qui répondent à leurs critères spécifiés. Leur affichage contient les problèmes regroupés par personne assignée et triés par priorité pour afficher les problèmes les plus importants et l'état des problèmes pour chaque développeur. Lorsque des tâches sont assignées à Jorge, il planifie son travail en créant un problème pour chaque tâche. Lors de la création de problèmes, Jorge peut choisir le statut, la priorité et l'effort d'estimation du travail appropriés. Pour les problèmes plus importants, Jorge ajoute des tâches au problème, afin de diviser le travail en objectifs plus petits. Jorge crée ses problèmes avec un statut de brouillon, tel qu'un arriéré, car il n'a pas l'intention de s'y atteler immédiatement. Les problèmes dont le statut est une ébauche apparaissent dans la vue Brouillons où ils doivent être planifiés et priorisés. Une fois que Jorge est prêt à commencer le travail, il déplace le problème correspondant vers le tableau en mettant son statut à un statut dans une autre catégorie (Non démarré, démarré ou terminé). Au fur et à mesure que chaque tâche est en cours d'élaboration, l'équipe peut filtrer par titre, statut, destinataire, étiquette, priorité et estimation pour trouver un problème spécifique ou des problèmes similaires correspondant au paramètre spécifié. À l'aide du tableau, Jorge et son équipe peuvent voir le nombre de tâches accomplies pour chaque problème et suivre la day-to-day progression en faisant glisser chaque problème d'un statut à l'autre jusqu'à ce que la tâche soit terminée. Au fur et à mesure que le projet progresse, les problèmes terminés s'accumulent dans le statut Terminé. Wang Xiulan décide de les supprimer de la vue en les archivant à l'aide du bouton d'archivage rapide, afin que les développeurs puissent se concentrer sur les problèmes liés aux travaux en cours et à venir.

Lors de la planification de leur travail, les développeurs travaillant sur le projet choisissent Trier par et Grouper par pour trouver les problèmes qu'ils souhaitent transférer du backlog au tableau de bord. Ils peuvent choisir d'ajouter des problèmes au tableau en fonction des demandes des clients les plus prioritaires. Ils regroupent donc le tableau par étiquette de demande client et trient par priorité. Ils peuvent également trier par estimation pour s'assurer qu'ils prennent en charge un volume de travail qu'ils peuvent accomplir. Le chef de projet, Saanvi Sarkar, passe régulièrement en revue et résorbe l'arriéré afin de s'assurer que la priorité reflète fidèlement l'importance de chaque problème pour la réussite du projet.

## Rubriques

- [Concepts des problèmes](#)
- [Création d'un problème dans CodeCatalyst](#)
- [Rédaction et collaboration sur des problèmes dans CodeCatalyst](#)
- [Recherche et affichage des problèmes](#)
- [Problèmes liés à l'exportation](#)
- [Configuration des paramètres des problèmes](#)
- [Quotas pour les numéros parus dans CodeCatalyst](#)

## Concepts des problèmes

La création d'un problème est un moyen rapide et efficace de suivre le travail effectué dans le cadre d'un projet. Vous pouvez utiliser les problèmes pour discuter du travail lors de réunions de synchronisation quotidiennes, hiérarchiser les tâches, etc.

Cette page inclut une liste de concepts qui vous aideront à utiliser efficacement les problèmes dans CodeCatalyst.

## Problèmes actifs

Les problèmes actifs sont des problèmes qui n'ont pas le statut de brouillon ou qui ne sont pas archivés. En d'autres termes, les problèmes actifs sont des problèmes dont le statut correspond à l'une des catégories de statut suivantes : Non démarré, Commencé et Terminé. Pour plus d'informations sur les statuts et les catégories de statut, consultez [Statut et catégories de statut](#).

Vous pouvez consulter tous les problèmes actifs de votre projet à partir de la vue Problèmes actifs par défaut.

## Numéros archivés

Un numéro archivé est un problème qui n'est plus pertinent pour votre projet. Par exemple, vous pouvez [archiver un problème](#) s'il est résolu et que vous n'avez plus besoin de le voir dans la colonne Terminé, ou s'il a été créé par erreur. Les numéros archivés peuvent être désarchivés si nécessaire.

## Cessionnaire

Le cessionnaire est la personne à laquelle le problème est attribué. Si la personne n'apparaît pas dans la liste lorsque vous la recherchez, cela signifie qu'elle n'a pas été ajoutée à votre projet. Pour les ajouter, consultez [Inviter un utilisateur à rejoindre votre projet](#). Pour autoriser plusieurs personnes à participer à un problème, consultez [Activation ou désactivation de plusieurs destinataires](#). Les problèmes impliquant plusieurs destinataires apparaîtront sur votre tableau avec des avatars de couleurs différentes, chacun représentant l'un des destinataires.

## Champs personnalisés

Les champs personnalisés vous permettent de personnaliser les différents attributs d'un problème en fonction de vos besoins en matière de suivi et de gestion des problèmes au sein d'un projet. Par exemple, vous pouvez ajouter un champ pour la feuille de route, une date d'échéance spécifique ou un champ pour le demandeur.

## Estimation

Dans le développement agile, l'estimation est connue sous le nom de story points. Utilisez l'estimation pour inclure la quantité de travail requise, en plus de l'ambiguïté et de la complexité du problème. Utilisez des estimations plus élevées pour les problèmes présentant un risque, une difficulté et des inconnues plus élevés. Vous pouvez modifier le type d'estimation pour votre projet. Pour plus d'informations sur les types d'estimation et sur leur configuration, consultez [Configuration de l'estimation des efforts liés aux problèmes](#).

## Problème

Un problème est un enregistrement qui permet de suivre le travail lié à votre projet. Vous pouvez créer un problème lié à une fonctionnalité, à une tâche, à un bogue ou à toute autre tâche liée à votre projet. Si vous utilisez le développement agile, un problème peut également décrire une histoire épique ou une histoire utilisateur.

## Étiquette

L'étiquette est utilisée pour regrouper, trier et filtrer les problèmes. Vous pouvez saisir un nouveau nom d'étiquette ou choisir l'une des étiquettes dans la liste remplie. Cette liste comprend les étiquettes récemment utilisées dans le projet. Un numéro peut avoir plusieurs étiquettes, et une étiquette peut être supprimée d'un numéro. Pour personnaliser les étiquettes, voir [Étiquettes](#).

## Priorité

La priorité fait référence au niveau d'importance du problème. Il existe quatre options : faible, moyenne, élevée et aucune priorité.

## Statut et catégories de statut

Le statut est l'état actuel du problème et est utilisé pour vérifier rapidement l'évolution d'un problème tout au long de son cycle de vie, du début à la fin. Tous les problèmes doivent avoir un statut, et chaque statut appartient à une catégorie de statut. Les catégories de statut sont utilisées pour vous aider à organiser vos statuts et à remplir les affichages des problèmes par défaut.

Il existe cinq statuts par défaut et quatre catégories de statut dans CodeCatalyst. Vous pouvez créer d'autres statuts, mais vous ne pouvez pas créer d'autres catégories de statut. La liste suivante contient les statuts par défaut et leurs catégories de statut entre parenthèses : Backlog (Brouillon), À faire (Non commencé), En cours (Commencé), En cours de révision (Commencé) et Terminé (Terminé).

Pour plus d'informations sur l'utilisation des statuts, consultez [Statutes](#).

## Tâches

Des tâches peuvent être ajoutées aux problèmes pour mieux décomposer et organiser le travail lié à ce problème. Vous pouvez ajouter des tâches à un problème lors de sa création ou ajouter des tâches à un problème existant. Lorsque vous consultez un numéro, vous pouvez le réorganiser, le supprimer ou marquer ses tâches comme terminées.

## Vues

Les problèmes de votre CodeCatalyst projet sont affichés dans les vues. Les vues peuvent être soit des vues en grille qui présentent les problèmes sous forme de liste, soit des vues de tableau qui présentent les problèmes sous forme de vignettes dans des colonnes organisées par statut de problème. Il existe quatre vues par défaut, et vous pouvez [créer vos propres vues grâce au](#)

[regroupement, au filtrage et au tri personnalisés](#). La liste suivante contient des informations sur les quatre vues par défaut.

- La vue Brouillons est une vue en grille qui montre les problèmes sur lesquels nous ne travaillons pas actuellement. Tout problème créé avec un statut dans la catégorie État provisoire apparaît dans cette vue. Cette vue peut être utilisée par les équipes pour voir quels problèmes sont toujours en cours de définition ou attendent d'être assignés et résolus.
- La vue Problèmes actifs est une vue d'ensemble de tous les problèmes en cours de résolution. Tout problème dont le statut se situe dans les catégories Non démarré, Commencé ou Terminé s'affichera dans cette vue.
- La vue Tous les problèmes est une vue en grille qui montre tous les problèmes du projet, qu'il s'agisse de brouillons ou de problèmes actifs.
- La vue Archivé affiche tous les numéros archivés.

## Création d'un problème dans CodeCatalyst

Les fonctionnalités d'intelligence artificielle génératives d'Amazon CodeCatalyst sont en version préliminaire et sont susceptibles d'être modifiées. Ils ne sont disponibles que dans la région ouest des États-Unis (Oregon). L'accès aux fonctionnalités génératives de l'IA varie selon le niveau. Pour plus d'informations, consultez [Tarification d'](#).

Les équipes de développement créent des problèmes pour faciliter le suivi et la gestion de leur travail. Vous pouvez créer des problèmes dans le cadre d'un projet en fonction de vos besoins. Par exemple, vous pouvez créer un problème pour suivre la mise à jour d'une variable dans votre code. Vous pouvez attribuer des problèmes à d'autres utilisateurs du projet, utiliser des étiquettes pour vous aider à suivre votre travail, etc.

Suivez ces instructions pour créer un problème dans CodeCatalyst.

Pour créer un problème

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez au projet dans lequel vous souhaitez créer un problème.
3. Sur la page d'accueil du projet, choisissez Créer un problème. Sinon, dans le volet de navigation, sélectionnez Problèmes.

#### 4. Choisissez Créer un problème.

##### Note

Vous pouvez également ajouter des problèmes en ligne lorsque vous utilisez une vue en grille.

5. Entrez le titre du numéro.
6. (Facultatif) Entrez une description. Vous pouvez utiliser Markdown pour ajouter une mise en forme.
7. (Facultatif) Choisissez un statut, une priorité ou une estimation pour le problème. Si le paramètre d'estimation du projet est défini sur Masquer les estimations, il n'y aura pas de champ Estimation.
8. (Facultatif) Ajoutez des tâches au problème. Les tâches peuvent être utilisées pour décomposer le travail lié à un problème en objectifs plus petits. Pour ajouter une tâche, choisissez + Ajouter des tâches. Entrez ensuite le nom de la tâche dans le champ de texte et appuyez sur Entrée. Après avoir ajouté des tâches, vous pouvez les marquer comme terminées en cochant la case ou les réorganiser en choisissant et en faisant glisser la tâche depuis le côté gauche de la case à cocher.
9. (Facultatif) Ajoutez une étiquette existante ou créez-en une nouvelle et ajoutez-la en choisissant + Ajouter une étiquette.
  - a. Pour ajouter une étiquette existante, choisissez-la dans la liste. Vous pouvez saisir un terme de recherche dans le champ pour rechercher toutes les étiquettes contenant ce terme dans le projet.
  - b. Pour créer une nouvelle étiquette et l'ajouter, entrez le nom de l'étiquette que vous souhaitez créer dans le champ de recherche et appuyez sur Entrée.
10. (Facultatif) Ajoutez un destinataire en choisissant + Ajouter un destinataire. Vous pouvez rapidement vous ajouter en tant que cessionnaire en choisissant + M'ajouter.

##### Tip

Vous pouvez choisir d'attribuer un problème à Amazon Q pour qu'Amazon Q essaie de le résoudre. Pour plus d'informations, consultez [Tutoriel : Utiliser les fonctionnalités CodeCatalyst génératives de l'IA pour accélérer votre travail de développement](#).

Cette fonctionnalité nécessite que les fonctionnalités d'IA génératives soient activées pour l'espace. Pour plus d'informations, consultez la section [Gestion des fonctionnalités génératives de l'IA](#).

11. (Facultatif) Ajoutez un champ personnalisé existant ou créez-en un nouveau. Les problèmes peuvent comporter plusieurs champs personnalisés.
  - a. Pour ajouter un champ personnalisé existant, choisissez-le dans la liste. Vous pouvez saisir un terme de recherche dans le champ pour rechercher tous les champs personnalisés contenant ce terme dans le projet.
  - b. Pour créer un nouveau champ personnalisé et l'ajouter, entrez le nom du champ personnalisé que vous souhaitez créer dans le champ de recherche et appuyez sur Entrée. Choisissez ensuite le type de champ personnalisé que vous souhaitez créer et définissez une valeur.
12. Choisissez Créer un problème. Une notification apparaît dans le coin inférieur droit : si le problème a été créé avec succès, un message de confirmation apparaît indiquant que le problème a été créé avec succès. Si le problème n'a pas été créé correctement, un message d'erreur indiquant la raison de l'échec apparaît. Vous pouvez ensuite choisir Réessayer pour modifier et réessayer de créer le problème, ou choisir Ignorer pour ignorer le problème. Les deux options annuleront la notification.

#### Note

Vous ne pouvez pas lier une pull request à un problème lorsque vous la créez. Cependant, vous pouvez le [modifier après l'avoir créé](#) pour ajouter des liens vers des pull requests.

## Bonnes pratiques lors de la création et du traitement des problèmes assignés à Amazon Q

Lorsque vous créez des problèmes, certains d'entre eux persistent parfois. Les causes peuvent être complexes et variables. Parfois, c'est parce qu'on ne sait pas exactement qui devrait y travailler. D'autres fois, le problème nécessite des recherches ou une expertise sur une partie particulière de la base de code et les meilleurs candidats pour le travail sont occupés à d'autres problèmes. Souvent, il faut d'abord s'occuper d'autres travaux urgents. L'une ou l'ensemble de ces causes peuvent entraîner des problèmes qui ne sont pas résolus. CodeCatalyst inclut l'intégration avec

un assistant d'intelligence artificielle génératif appelé Amazon Q qui peut analyser un problème en fonction de son titre et de sa description. Si vous attribuez le problème à Amazon Q, celui-ci tentera de créer un projet de solution que vous pourrez évaluer. Cela peut vous aider, vous et votre équipe, à vous concentrer et à optimiser le travail sur les problèmes qui nécessitent votre attention, tandis qu'Amazon Q travaille sur une solution aux problèmes que vous ne disposez pas des ressources nécessaires pour résoudre immédiatement.

### Note

Propulsé par Amazon Bedrock : AWS implémente la [détection automatique des abus](#). La fonctionnalité de développement de fonctionnalités Assign issues to Amazon Q étant basée sur Amazon Bedrock, les utilisateurs peuvent tirer pleinement parti des contrôles mis en œuvre dans Amazon Bedrock pour renforcer la sûreté, la sécurité et l'utilisation responsable de l'intelligence artificielle (IA).

Amazon Q est le plus performant pour résoudre les problèmes simples et directs. Pour de meilleurs résultats, utilisez un langage simple pour expliquer clairement ce que vous voulez faire. Voici quelques bonnes pratiques qui vous aideront à créer des problèmes optimisés pour qu'Amazon Q puisse les résoudre.

- Restez simple. Amazon Q fait de son mieux avec des modifications de code simples et des correctifs qui peuvent être expliqués dans le titre et la description du problème. N'attribuez pas de problèmes à des titres vagues ou à des descriptions trop fleuries ou contradictoires.
- Soyez précis. Plus vous fournirez d'informations sur les modifications exactes nécessaires pour résoudre le problème, plus Amazon Q sera en mesure de créer une solution capable de résoudre le problème. Si possible, incluez des détails spécifiques tels que le nom des API que vous souhaitez modifier, les méthodes que vous souhaitez mettre à jour, les tests nécessitant des modifications et tout autre détail auquel vous pouvez penser.
- Assurez-vous d'avoir inclus tous les détails dans le titre et la description du problème avant de l'attribuer à Amazon Q. Vous ne pouvez pas modifier le titre ou la description d'un problème une fois que vous l'avez attribué à Amazon Q. Assurez-vous donc de disposer de toutes les informations requises dans un numéro avant de l'attribuer à Amazon Q.
- N'attribuez que les problèmes nécessitant des modifications de code dans un seul référentiel source. Amazon Q ne peut travailler que sur le code d'un seul référentiel source dans CodeCatalyst. Les référentiels liés ne sont pas pris en charge. Assurez-vous que le problème ne nécessite que des modifications dans un seul référentiel source avant de l'attribuer à Amazon Q.



- Utilisez la valeur par défaut suggérée par Amazon Q pour approuver chaque étape. Par défaut, Amazon Q aura besoin de votre approbation pour chaque étape. Cela vous permet d'interagir avec Amazon Q dans des commentaires non seulement sur le problème, mais également sur toute pull request qu'il crée. Cela fournit une expérience plus interactive avec Amazon Q qui vous permet d'ajuster son approche et d'affiner le code qu'il crée pour résoudre le problème.

#### Note

Amazon Q ne répond pas aux commentaires individuels relatifs aux problèmes ou aux pull requests, mais il les examinera lorsqu'il lui sera demandé de reconsidérer son approche ou de créer une révision.

- Lisez toujours attentivement l'approche suggérée par Amazon Q. Une fois que vous aurez approuvé son approche, Amazon Q commencera à générer du code en fonction de cette approche. Assurez-vous que l'approche semble correcte et inclut tous les détails que vous attendez avant de demander à Amazon Q de continuer.
- Assurez-vous de n'autoriser Amazon Q à travailler sur les flux de travail que si vous ne disposez pas de flux de travail existants susceptibles de les déployer avant leur révision. Les flux de travail de votre projet sont peut-être configurés pour démarrer des exécutions lors d'événements de pull request. Dans ce cas, toute pull request créée par Amazon Q qui inclut la création ou la mise à jour du flux de travail YAML peut démarrer une exécution des flux de travail inclus dans la pull request. Il est recommandé de ne pas autoriser Amazon Q à travailler sur des fichiers de flux de travail, sauf si vous êtes certain qu'aucun flux de travail de votre projet n'exécutera automatiquement ces flux de travail avant de consulter et d'approuver la pull request qu'il crée.

Pour plus d'informations, consultez la section [Gestion Tutoriel : Utiliser les fonctionnalités CodeCatalyst génératives de l'IA pour accélérer votre travail de développement des fonctionnalités d'IA générative](#).

## Rédaction et collaboration sur des problèmes dans CodeCatalyst

### Table des matières

- [Modification d'un numéro](#)
- [Utilisation de pièces jointes](#)
- [Gestion des tâches liées aux problèmes](#)
- [Marquer un problème comme bloqué ou débloqué](#)

- [Ajouter, modifier ou supprimer des commentaires](#)
  - [Utiliser des mentions dans un commentaire](#)
- [Progression d'un problème](#)
  - [Entre le carnet de commandes et le conseil d'administration](#)
  - [Progressez un problème tout au long des étapes du cycle de vie au tableau](#)
  - [Transférer les problèmes entre les groupes](#)
- [Archivage d'un problème](#)

## Modification d'un numéro

Suivez ces étapes pour modifier le titre, la description, le statut, le destinataire, la priorité, l'estimation ou les libellés d'un problème.

Pour modifier un problème

1. Choisissez le problème que vous souhaitez modifier pour afficher les détails du problème. Pour obtenir de l'aide pour identifier votre problème, consultez [Recherche et affichage des problèmes](#).
2. Pour modifier le titre du numéro, choisissez-le, entrez un nouveau titre et appuyez sur Entrée.
3. Pour modifier la description, choisissez-la, entrez une nouvelle description et appuyez sur Entrée. Vous pouvez utiliser Markdown pour ajouter du formatage.
4. Dans Tâches, vous pouvez consulter et gérer les tâches associées au problème. Pour plus d'informations, consultez [Gestion des tâches liées aux problèmes](#).
5. Pour modifier le statut, l'estimation ou la priorité, choisissez une option dans les menus déroulants correspondants.
6. Dans Étiquettes, vous pouvez ajouter une étiquette existante, en créer une nouvelle ou en supprimer une.
  - a. Pour ajouter une étiquette existante, choisissez + Ajouter une étiquette et choisissez l'étiquette dans la liste. Vous pouvez saisir un terme de recherche dans le champ pour rechercher toutes les étiquettes contenant ce terme dans le projet.
  - b. Pour créer une nouvelle étiquette et l'ajouter, choisissez + Ajouter une étiquette, entrez le nom de l'étiquette que vous souhaitez créer dans le champ de recherche et appuyez sur Entrée.
  - c. Pour supprimer une étiquette, cliquez sur l'icône X à côté de l'étiquette que vous souhaitez supprimer. Si vous supprimez une étiquette de tous les numéros, elle apparaîtra dans la

section Étiquettes non utilisées de la section Étiquettes des paramètres du numéro. Les libellés non utilisés apparaissent à la fin de la liste des libellés lorsque vous utilisez des filtres ou que vous ajoutez des libellés à un problème. Vous trouverez un aperçu de toutes les étiquettes (utilisées et non utilisées) et des problèmes qui les contiennent dans les paramètres du problème.

7. Pour attribuer un problème, choisissez + Ajouter un destinataire dans la section Destinataire, puis recherchez et choisissez le destinataire dans la liste. Vous pouvez choisir + M'ajouter pour vous ajouter rapidement en tant que cessionnaire.
8. Dans Pièces jointes, vous pouvez ajouter, télécharger ou supprimer des pièces jointes. Pour plus d'informations, consultez [Utilisation de pièces jointes](#).
9. Pour lier une pull request, choisissez Link pull request, puis choisissez une pull request dans la liste ou entrez son URL ou son ID. Pour dissocier une pull request, cliquez sur l'icône de dissociation.

 Tip

Après avoir ajouté un lien vers une pull request à un problème, vous pouvez accéder rapidement à celle-ci en choisissant son identifiant dans la liste des pull requests liées. Vous pouvez utiliser l'URL d'une pull request pour lier des pull requests appartenant à des projets différents de ceux du forum de discussion, mais seuls les utilisateurs membres de ce projet pourront consulter cette pull request ou y accéder.

10. (Facultatif) Ajoutez et définissez un champ personnalisé existant, créez un nouveau champ personnalisé ou supprimez un champ personnalisé. Les problèmes peuvent comporter plusieurs champs personnalisés.
  - a. Pour ajouter un champ personnalisé existant, choisissez-le dans la liste. Vous pouvez saisir un terme de recherche dans le champ pour rechercher tous les champs personnalisés contenant ce terme dans le projet.
  - b. Pour créer un nouveau champ personnalisé et l'ajouter, entrez le nom du champ personnalisé que vous souhaitez créer dans le champ de recherche et appuyez sur Entrée. Choisissez ensuite le type de champ personnalisé que vous souhaitez créer et définissez une valeur.
  - c. Pour supprimer un champ personnalisé, cliquez sur l'icône X à côté du champ personnalisé que vous souhaitez supprimer. Si vous supprimez un champ personnalisé de tous les numéros, le champ personnalisé sera supprimé et vous ne le verrez plus lors du filtrage.

## Utilisation de pièces jointes

Vous pouvez ajouter des pièces jointes aux numéros CodeCatalyst afin de rendre les fichiers associés facilement accessibles. Utilisez la procédure suivante pour gérer les pièces jointes associées à un problème.

La taille des pièces jointes ajoutées aux numéros est prise en compte dans les quotas de stockage de votre espace. Pour plus d'informations sur l'affichage et la gestion des pièces jointes de votre projet, consultez [Afficher et gérer les pièces jointes](#).

### Important

Les pièces jointes aux numéros ne sont ni scannées ni analysées par Amazon CodeCatalyst. Tout utilisateur peut ajouter une pièce jointe à un problème susceptible de contenir du code ou du contenu malveillants. Assurez-vous que les utilisateurs connaissent les meilleures pratiques en matière de gestion des pièces jointes et de protection contre le code, le contenu ou les virus malveillants.

Pour ajouter, télécharger ou supprimer des pièces jointes

1. Choisissez le problème pour lequel vous souhaitez gérer les pièces jointes. Pour obtenir de l'aide pour identifier votre problème, consultez [Recherche et affichage des problèmes](#).
2. Pour ajouter une pièce jointe, choisissez Charger un fichier. Accédez au fichier dans l'explorateur de fichiers de votre système d'exploitation et sélectionnez-le. Choisissez Ouvrir pour l'ajouter en tant que pièce jointe. Pour obtenir des informations sur les quotas, telles que la taille maximale des pièces jointes, consultez [Quotas pour les numéros parus dans CodeCatalyst](#).

Notez les restrictions suivantes concernant les noms de fichiers joints et les types de contenu :

- Les caractères suivants ne sont pas autorisés dans les noms de fichiers :
  - Personnages de contrôle : `0x00-0x1f` et `0x80-0x9f`
  - Caractères réservés : `/?,<,>,\,,:,*,|`, et `"`
  - Noms de fichiers réservés Unix : et . . .
  - Périodes et espaces de suivi

- Noms de fichiers réservés à Windows : CON, PRN, AUX, NUL, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, and LPT9
- Le type de contenu de la pièce jointe doit respecter le modèle de type de support suivant :

```
media-type = type "/" [tree "."] subtype ["+" suffix]* [";" parameter];
```

Par exemple, `text/html; charset=UTF-8`.

3. Pour télécharger une pièce jointe, choisissez le menu représentant des points de suspension à côté de la pièce jointe que vous souhaitez télécharger, puis sélectionnez Télécharger.
4. Pour copier l'URL d'une pièce jointe, choisissez le menu représentant des points de suspension à côté de la pièce jointe dont vous souhaitez copier l'URL, puis choisissez Copier l'URL.
5. Pour supprimer une pièce jointe, choisissez le menu représentant des points de suspension à côté de la pièce jointe que vous souhaitez supprimer, puis choisissez Supprimer.

## Gestion des tâches liées aux problèmes

Des tâches peuvent être ajoutées aux problèmes pour mieux décomposer, organiser et suivre le travail lié à ce problème.

Pour gérer les tâches liées à un problème

1. Choisissez le problème pour lequel vous souhaitez gérer les tâches. Pour obtenir de l'aide pour identifier votre problème, consultez [Recherche et affichage des problèmes](#).
2. Dans Tâches, vous pouvez consulter et gérer les tâches associées au problème.
  1. Pour ajouter une tâche, saisissez le nom de la tâche dans le champ de texte et appuyez sur Entrée.
  2. Pour marquer une tâche comme terminée, cochez la case correspondante.
  3. Pour afficher ou mettre à jour les détails d'une tâche, sélectionnez-la dans la liste.
  4. Pour réorganiser les tâches, choisissez-les et faites-les glisser depuis le côté gauche de la case à cocher.
  5. Pour supprimer une tâche, choisissez le menu représentant des points de suspension de la tâche, puis sélectionnez Supprimer.

## Marquer un problème comme bloqué ou débloqué

Si quelque chose vous empêche de travailler sur un problème, vous pouvez le marquer comme bloqué. Par exemple, votre problème peut être bloqué s'il repose sur une modification apportée à une autre partie de votre base de code qui n'a pas encore été fusionnée.

Lorsque vous marquez un problème comme bloqué, vous CodeCatalyst ajoutez une étiquette rouge « bloqué » au problème, ce qui le rend très visible dans votre backlog, vos archives ou sur votre forum.

Vous pouvez débloquent le problème lorsque des circonstances extérieures sont résolues.

Pour marquer un problème comme bloqué

1. Ouvrez le problème que vous souhaitez marquer comme bloqué. Pour obtenir de l'aide pour identifier votre problème, consultez [Recherche et affichage des problèmes](#).
2. Choisissez Actions, puis sélectionnez Marquer comme bloqué.

Pour débloquent un problème

1. Ouvrez le problème que vous souhaitez débloquent. Pour obtenir de l'aide pour identifier votre problème, consultez [Recherche et affichage des problèmes](#).
2. Choisissez Actions, puis Marquer comme débloquent.

## Ajouter, modifier ou supprimer des commentaires

Vous pouvez laisser un commentaire sur un problème. Dans les commentaires, vous pouvez étiqueter les autres membres de l'espace, les autres projets de l'espace, les problèmes connexes et le code.

Pour ajouter un commentaire à un problème

1. Accédez à votre projet.
2. Dans la barre de navigation, sélectionnez Problèmes.
3. Choisissez le problème pour lequel vous souhaitez ajouter le commentaire. Pour obtenir de l'aide pour identifier votre problème, consultez [Recherche et affichage des problèmes](#).
4. Entrez le commentaire dans le champ Commentaires. Vous pouvez utiliser Markdown pour ajouter du formatage.

## 5. Sélectionnez Send (Envoyer).

### Pour modifier un commentaire

Vous pouvez modifier les commentaires que vous faites sur les problèmes. Vous ne pouvez modifier que le commentaire dont vous êtes l'auteur.

1. Accédez à votre projet.
2. Dans la barre de navigation, sélectionnez Problèmes.
3. Choisissez le numéro pour lequel vous souhaitez modifier un commentaire. Pour obtenir de l'aide pour identifier votre problème, consultez [Recherche et affichage des problèmes](#).
4. Pour modifier un commentaire, recherchez le commentaire que vous souhaitez modifier.

#### Tip

Vous pouvez d'abord trier les commentaires en fonction des plus anciens ou des plus récents. Les commentaires sont chargés 10 à la fois.

5. Choisissez l'icône représentant des points de suspension, puis sélectionnez Modifier.
6. Modifiez le commentaire. Vous pouvez utiliser Markdown pour ajouter du formatage.
7. Choisissez Enregistrer. Le commentaire est maintenant mis à jour.

### Pour supprimer un commentaire

Vous pouvez supprimer les commentaires que vous faites sur des problèmes. Vous ne pouvez supprimer que les commentaires dont vous êtes l'auteur. Lorsqu'un commentaire est supprimé, votre nom d'utilisateur s'affiche, mais avec les mots « Ce commentaire a été supprimé » à la place du texte original du commentaire.

1. Accédez à votre projet.
2. Dans la barre de navigation, sélectionnez Problèmes.
3. Choisissez le problème pour lequel vous souhaitez supprimer un commentaire. Pour obtenir de l'aide pour identifier votre problème, consultez [Recherche et affichage des problèmes](#).
4. Choisissez l'icône représentant des points de suspension, sélectionnez Supprimer, puis cliquez sur Confirmer.

## Utiliser des mentions dans un commentaire

Vous pouvez mentionner les membres de l'espace, les autres projets de l'espace, les problèmes connexes et le code dans les commentaires. Cela crée un lien rapide vers l'utilisateur ou la ressource que vous mentionnez.

### À @mention dans un commentaire

1. Accédez à votre projet.
2. Dans la barre de navigation, sélectionnez Problèmes.
3. Choisissez le problème que vous souhaitez modifier pour afficher les détails du problème. Pour obtenir de l'aide pour identifier votre problème, consultez [Recherche et affichage des problèmes](#).
4. Choisissez la zone de texte Ajouter un commentaire.
5. Tapez `@user_name` pour mentionner un autre utilisateur.
6. Tapez `@project_name` pour mentionner un projet.
7. Tapez `@issue_name` ou `@issue_number` pour mentionner un autre problème.
8. Tapez `@file_name` pour mentionner des fichiers ou du code spécifiques dans un référentiel source.

#### Note

Une liste des 5 principaux éléments (utilisateurs, référentiels de sources, projets, etc.) contenant les termes qui correspondent aux @mention vôtres sera renseignée au fur et à mesure que vous tapez.

9. Choisissez l'article que vous souhaitez mentionner. Le chemin indiquant où se trouve l'élément sera renseigné dans la zone de texte des commentaires.
10. Terminez votre commentaire et choisissez Envoyer.

## Progression d'un problème

Chaque problème a un cycle de vie. Dans CodeCatalyst, les problèmes commencent généralement sous forme de brouillon dans le backlog. Lorsque le traitement de ce problème doit commencer, il est transféré dans une autre catégorie de statut et passe par différents statuts jusqu'à ce qu'il soit terminé, puis il est archivé. Vous pouvez déplacer ou faire progresser un problème tout au long de son cycle de vie en procédant comme suit :



- Vous pouvez déplacer un problème entre le backlog et le forum.
- Vous pouvez faire passer le problème en cours par différentes étapes de finalisation.
- Vous pouvez archiver un problème terminé.

## Entre le carnet de commandes et le conseil d'administration

Vous pouvez déplacer un problème du backlog vers le tableau une fois que vous avez commencé à travailler sur le problème. Vous pouvez également remettre un problème dans le backlog si le travail est reporté.

Pour transférer un problème entre le backlog et le forum

1. Accédez à votre projet.
2. Dans le volet de navigation, sélectionnez Problèmes. La vue par défaut est le tableau.
3. Pour transférer un problème du tableau vers le backlog, procédez comme suit :
  - a. Choisissez le problème que vous souhaitez déplacer. Pour obtenir de l'aide pour identifier votre problème, consultez [Recherche et affichage des problèmes](#).
  - b. Choisissez Backlog dans le menu déroulant État.
4. Pour transférer un problème du backlog vers le tableau, procédez comme suit :
  - a. Pour accéder au backlog, choisissez Board, puis Backlog.
  - b. Choisissez le problème que vous souhaitez déplacer. Pour obtenir de l'aide pour identifier votre problème, consultez [Recherche et affichage des problèmes](#).
  - c. Choisissez Ajouter au tableau ou choisissez un statut autre que Backlog.

## Progresser un problème tout au long des étapes du cycle de vie au tableau

Vous pouvez déplacer un problème au sein d'un tableau en passant par différents statuts jusqu'à ce qu'il soit terminé.

Pour déplacer un problème au sein du tableau

1. Dans le volet de navigation, sélectionnez Problèmes. La vue par défaut est le tableau.
2. Effectuez l'une des actions suivantes :
  - Glissez et déposez un problème vers un autre statut.

- Choisissez un problème, puis choisissez un statut dans le menu déroulant État.
- Choisissez un problème, puis choisissez Move to : **next-status**.

Pour plus d'informations sur l'archivage d'un problème, consultez [Archivage d'un problème](#).

## Transférer les problèmes entre les groupes

Vous pouvez [regrouper les problèmes](#) dans les vues Tous les numéros et Tableau selon différents paramètres. Si les problèmes sont groupés, vous pouvez les déplacer d'un groupe à l'autre. Le déplacement d'un problème d'un groupe à un autre modifiera automatiquement le champ dans lequel les problèmes sont regroupés pour correspondre au groupe cible.

À titre d'exemple de scénario, supposons CodeCatalyst qu'une entreprise utilise des solutions dont les problèmes sont attribués à deux personnes, Wang Xiulan et Saanvi Sarkar. Le tableau est regroupé par groupe Assignée, et il existe deux groupes, un pour chaque personne assignée. Le transfert d'un problème du groupe Wang Xiulan vers le groupe Saanvi Sarkar mettra à jour le responsable du problème à Saanvi Sarkar.

## Archivage d'un problème

### Note

Les problèmes ne sont pas supprimés dans un projet, ils sont archivés. Pour supprimer des problèmes, vous devez supprimer le projet.

Vous pouvez archiver un problème lorsqu'il n'est plus nécessaire dans votre projet. Lorsque vous archivez un numéro, il CodeCatalyst est supprimé de toutes les vues qui filtrent les problèmes archivés. Les numéros archivés peuvent être consultés dans la vue par défaut des numéros archivés, où ils peuvent être désarchivés si nécessaire.

Vous archivez un problème si :

- Vous avez terminé le problème et vous n'en avez plus besoin dans la colonne Terminé.
- Vous n'avez pas l'intention d'y travailler.
- Vous l'avez créé par erreur.
- Vous avez atteint le nombre maximum de problèmes actifs.

## Pour archiver un problème

1. Ouvrez le numéro que vous souhaitez archiver. Pour obtenir de l'aide pour identifier votre problème, consultez [Recherche et affichage des problèmes](#).
2. Choisissez Actions, puis choisissez Déplacer vers l'archive.
3. (Facultatif) Pour archiver rapidement plusieurs numéros ayant le statut Terminé, choisissez les points de suspension verticaux en haut de chaque statut Terminé sur le tableau, puis choisissez Archiver les problèmes.

## Pour désarchiver un problème

1. Ouvrez le dossier que vous souhaitez désarchiver. Vous pouvez consulter la liste des numéros archivés en ouvrant la vue des problèmes archivés dans le menu déroulant du sélecteur d'affichage des problèmes. Pour obtenir de l'aide pour identifier votre problème, consultez [Recherche et affichage des problèmes](#).
2. Choisissez Désarchiver.

# Recherche et affichage des problèmes

Les sections suivantes décrivent comment rechercher et visualiser efficacement les problèmes au sein d'un CodeCatalyst projet.

Les problèmes de votre CodeCatalyst projet sont affichés dans les vues. Les vues peuvent être soit des vues en grille qui présentent les problèmes sous forme de liste, soit des vues de tableau qui présentent les problèmes sous forme de vignettes dans des colonnes organisées par statut de problème. Il existe quatre vues par défaut, et vous pouvez [créer vos propres vues grâce au regroupement, au filtrage et au tri personnalisés](#). La liste suivante contient des informations sur les quatre vues par défaut.

- La vue Brouillons est une vue en grille qui montre les problèmes non résolus actuellement. Tout problème créé avec un statut dans la catégorie État provisoire apparaît dans cette vue. Cette vue peut être utilisée par les équipes pour voir quels problèmes sont toujours en cours de définition ou attendent d'être assignés et résolus.
- La vue Problèmes actifs est une vue d'ensemble de tous les problèmes en cours de résolution. Tout problème dont le statut se situe dans les catégories Non démarré, Commencé ou Terminé apparaîtra dans cette vue.

- La vue Tous les problèmes est une vue en grille qui montre tous les problèmes du projet, qu'il s'agisse de brouillons ou de problèmes actifs.
- La vue Archivé affiche tous les numéros archivés.

## Recherche d'un problème

Vous pouvez détecter un problème en recherchant des paramètres spécifiques. Pour plus d'informations sur l'affinement de votre recherche, consultez [Rechercher dans CodeCatalyst](#).

Pour rechercher un problème

1. Accédez à votre projet.
2. Utilisez la barre de recherche pour rechercher des problèmes ou des informations liées à des problèmes. Vous pouvez utiliser les paramètres de requête pour affiner votre recherche. Pour plus d'informations, consultez [Rechercher dans CodeCatalyst](#).

## Problèmes de tri

Par défaut, les problèmes CodeCatalyst sont triés par ordre manuel. La commande manuelle affiche les problèmes dans l'ordre dans lequel ils sont déplacés par les utilisateurs. Vous pouvez glisser-déposer les problèmes lorsqu'ils sont triés dans l'ordre manuel pour modifier leur ordre. Cette option de tri est utile pour résorber l'arriéré des problèmes et les classer par ordre de priorité.

Le tableau suivant montre comment les problèmes peuvent être triés à la fois en mode grille et en mode tableau.

| Options de tri de la vue en grille | Options de tri de la vue du tableau |
|------------------------------------|-------------------------------------|
| Commande manuelle                  | Commande manuelle                   |
| Dernière mise à jour               | Dernière mise à jour                |
| Priorité                           | Priorité                            |
| Estimation                         | Estimation                          |
| Title                              | Title                               |

| Options de tri de la vue en grille | Options de tri de la vue du tableau |
|------------------------------------|-------------------------------------|
| ID                                 |                                     |
| Statut                             |                                     |
| Bloqué                             |                                     |
| Champs personnalisés               |                                     |

Utilisez la procédure suivante pour modifier le mode de tri des problèmes.

Pour trier les problèmes

1. Accédez à votre projet.
2. Dans le volet de navigation, sélectionnez Problèmes. La vue par défaut est le tableau.
3. (Facultatif) Choisissez Problèmes actifs pour ouvrir le menu déroulant du sélecteur d'affichage des problèmes afin d'accéder à un autre affichage des problèmes.
4. Pour trier une vue en grille, deux options s'offrent à vous :
  - a. Choisissez l'en-tête du champ selon lequel vous souhaitez trier. Le choix de l'en-tête passera de l'ordre croissant à l'ordre décroissant.
  - b. Choisissez le menu déroulant Trier par et choisissez un paramètre de tri. Les problèmes seront triés par ordre croissant.
5. Pour trier une vue du tableau, choisissez le menu déroulant Trier par et choisissez un paramètre de tri. Les problèmes seront triés par ordre croissant.

## Problèmes de regroupement

Le regroupement est utilisé pour organiser les problèmes sur le tableau en fonction de plusieurs paramètres, tels que le destinataire, les étiquettes et la priorité.

Pour regrouper les problèmes

1. Accédez à votre projet.
2. Dans le volet de navigation, sélectionnez Problèmes. La vue par défaut est le tableau.

3. (Facultatif) Choisissez Problèmes actifs pour ouvrir le menu déroulant du sélecteur d'affichage des problèmes afin d'accéder à un autre affichage des problèmes.
4. Choisissez le groupe.
5. Dans Grouper par, choisissez un paramètre à regrouper par :
  - Si vous choisissez Destinataire ou Priorité, choisissez l'ordre de groupe.
  - Si vous choisissez Label, choisissez les labels, puis choisissez Grouper order.
6. (Facultatif) Cliquez sur le bouton Afficher les groupes vides pour afficher ou masquer les groupes auxquels aucun problème n'est actuellement attribué.
7. La vue est mise à jour au fur et à mesure que vous faites vos choix. Un problème apparaît uniquement dans le groupe correspondant aux paramètres configurés.

## Problèmes de filtrage

Utilisez le filtrage pour rechercher les problèmes contenant un nom, une priorité, une étiquette, des champs personnalisés ou un responsable spécifique.

### Pour filtrer les problèmes

1. Accédez à votre projet.
2. Dans le volet de navigation, sélectionnez Problèmes.
3. (Facultatif) Choisissez Problèmes actifs pour ouvrir le menu déroulant du sélecteur d'affichage des problèmes afin d'accéder à un autre affichage des problèmes.

#### Note

Pour filtrer en fonction d'une chaîne dans le nom ou la description du problème, saisissez-la dans la barre de recherche du problème.

4. Choisissez Filtre, puis choisissez + Ajouter un filtre.
5. Choisissez les paramètres à filtrer. Vous pouvez choisir plusieurs filtres et paramètres. Vous pouvez configurer des filtres pour afficher les problèmes correspondant à chaque filtre ou à un filtre individuel en sélectionnant et/ou ou. La vue sera mise à jour pour afficher les problèmes correspondant au filtre.

## Création d'une vue des problèmes

Vous pouvez créer des [vues](#) pour visualiser rapidement les problèmes correspondant à un ensemble de filtres particulier. Cela peut vous aider à gagner du temps et à consulter rapidement les problèmes que vous avez précédemment filtrés, regroupés ou triés.

Pour créer une vue des problèmes

1. Dans le volet de navigation, sélectionnez Problèmes.
2. (Facultatif) Selon votre cas d'utilisation, vous souhaitez peut-être créer une vue à partir d'une vue existante. Pour accéder à un autre affichage, choisissez Problèmes actifs pour ouvrir le menu déroulant du sélecteur d'affichage des problèmes et choisissez l'affichage.
3. (Facultatif) Configurez les filtres, le regroupement et le tri avant de créer votre vue. Vous pouvez les ajouter lors de la création d'une vue, mais si vous le faites auparavant, vous pouvez prévisualiser ce qui s'affiche dans la vue avant de la créer.
4. Ouvrez le menu déroulant du sélecteur d'affichage des problèmes depuis la barre d'en-tête. Pour créer une vue du tableau dans laquelle les problèmes sont affichés dans des colonnes en fonction de leur statut, choisissez le signe + dans la colonne Tableau. Pour créer une vue en grille dans laquelle les problèmes sont affichés sous forme de liste, choisissez le signe + dans la colonne Grille. Vous pouvez modifier le type de vue avant sa création si vous changez d'avis.
5. Dans la boîte de dialogue Créer une vue, entrez le nom de la vue.
6. Les champs Filtres, Regrouper les numéros par et Trier les problèmes par sont remplis en fonction des paramètres de l'affichage actuel. Mettez-les à jour si nécessaire.
7. Choisissez Créer une vue pour créer la vue et passer à celle-ci.

## Problèmes liés à l'exportation

Vous pouvez exporter les problèmes de votre affichage actuel dans un fichier .xlsx. Pour exporter les problèmes, effectuez les opérations suivantes.

Pour exporter les problèmes

1. Accédez à votre projet.
2. Dans la barre de navigation, sélectionnez Problèmes.

3. Choisissez Problèmes actifs pour ouvrir le menu déroulant du sélecteur d'affichage des problèmes et accédez à la vue contenant les problèmes que vous souhaitez exporter. Seuls les problèmes affichés dans la vue seront exportés.
4. Choisissez le menu Ellipses, puis sélectionnez Exporter vers Excel.
5. Le fichier .xlsx est téléchargé. Par défaut, il porte le nom du projet et la date à laquelle l'exportation a été terminée.

## Configuration des paramètres des problèmes

Les rubriques suivantes expliquent comment configurer les paramètres en cas de problème dans CodeCatalyst.

### Rubriques

- [Activation ou désactivation de plusieurs destinataires](#)
- [Configuration de l'estimation des efforts liés aux problèmes](#)
- [Statutes](#)
- [Étiquettes](#)
- [Champs personnalisés](#)
- [Afficher et gérer les pièces jointes](#)

## Activation ou désactivation de plusieurs destinataires

Procédez comme suit pour configurer le paramètre pour plusieurs destinataires en cas de problème dans CodeCatalyst

Pour activer ou désactiver plusieurs destinataires

1. Dans le volet de navigation, sélectionnez Problèmes.
2. Choisissez Problèmes actifs pour ouvrir le menu déroulant du sélecteur d'affichage des problèmes et choisissez Paramètres.
3. Dans Destinataire, dans la section Paramètres de base, activez l'indicateur pour permettre à plusieurs personnes d'être affectées au même problème. Un problème peut avoir jusqu'à 10 personnes assignées. Si vous n'activez pas cette option, vous ne pourrez affecter qu'un seul responsable à un problème.



## Configuration de l'estimation des efforts liés aux problèmes

Procédez comme suit pour configurer le paramètre d'estimation de l'effort en cas de problème dans CodeCatalyst.

Pour configurer l'estimation de l'effort en cas de problème

1. Dans le volet de navigation, sélectionnez Problèmes.
2. Choisissez Problèmes actifs pour ouvrir le menu déroulant du sélecteur d'affichage des problèmes et choisissez Paramètres.
3. Dans Estimation, dans la section Paramètres de base, choisissez le mode d'affichage des valeurs d'estimation. Les types d'estimations disponibles sont la taille des t-shirts, le séquençage de Fibonacci ou les estimations masquées. Lorsque le type d'estimation est mis à jour, aucune donnée n'est perdue et la valeur d'estimation de tous les numéros est automatiquement convertie. Le mappage de conversion est présenté dans le tableau suivant.

| Taille du t-shirt | Séquence de Fibonacci |
|-------------------|-----------------------|
| XS                | 1                     |
| XS                | 2                     |
| S                 | 3                     |
| M                 | 5                     |
| L                 | 8                     |
| XL                | 13                    |

## Statuses

Vous pouvez ajouter des statuts personnalisés sur votre tableau. Chaque statut personnalisé doit appartenir à l'une des catégories suivantes : Brouillon, Non commencé, Commencé ou Terminé. Les catégories de statut sont utilisées pour aider à organiser les statuts et à renseigner les vues par défaut. Pour plus d'informations sur les statuts et les catégories de statut, voir [Statut et catégories de statut](#) et pour plus d'informations sur les vues, voir [Recherche et affichage des problèmes](#).

## Pour créer un statut

1. Dans le volet de navigation, sélectionnez Problèmes.
2. Choisissez Problèmes actifs pour ouvrir le menu déroulant du sélecteur d'affichage des problèmes et choisissez Paramètres.
3. Dans Statuts, cliquez sur l'icône plus à côté de la catégorie dans laquelle vous souhaitez attribuer le statut.
4. Nommez le statut, puis cliquez sur l'icône en forme de coche.

### Note

Cliquez sur l'icône X pour annuler l'ajout d'un statut.

Le statut personnalisé est désormais visible sur votre tableau et apparaît sous forme d'option lors de la création d'un problème.

## Pour modifier un statut

1. Dans le volet de navigation, sélectionnez Problèmes.
2. Choisissez Problèmes actifs pour ouvrir le menu déroulant du sélecteur d'affichage des problèmes et choisissez Paramètres.
3. Dans Statuts, cliquez sur l'icône de modification à côté du statut que vous souhaitez modifier ou modifier.
4. Modifiez le statut, puis cliquez sur l'icône en forme de coche.

Le statut modifié est désormais visible sur votre tableau.

## Pour déplacer un statut

1. Dans le volet de navigation, sélectionnez Problèmes.
2. Cliquez sur l'icône représentant des points de suspension, puis sur Paramètres.
3. Dans Statuts, choisissez le statut que vous souhaitez déplacer.
4. Glissez et déposez le statut là où vous voulez qu'il soit.


 Note

Vous ne pouvez déplacer un statut que dans la catégorie qui lui est assignée.

Les statuts sont désormais réorganisés sur votre tableau.


## Pour désactiver un statut

1. Dans le volet de navigation, sélectionnez Problèmes.
2. Choisissez Problèmes actifs pour ouvrir le menu déroulant du sélecteur d'affichage des problèmes et choisissez Paramètres.
3. Dans Statuts, choisissez le statut que vous souhaitez désactiver.
4. Sur le statut que vous souhaitez désactiver, cliquez sur le bouton d'activation du statut. Le statut est désormais grisé.

 Note

Le statut désactivé apparaît sur le tableau jusqu'à ce que tous les problèmes en soient retirés. Les problèmes ne peuvent pas être ajoutés à un statut désactivé.

5. Pour réactiver un statut désactivé, cliquez sur le bouton correspondant. Le statut n'est plus grisé.

 Note

Il doit y avoir au moins un statut actif dans chaque catégorie. S'il n'existe qu'un seul statut dans la catégorie, vous ne pouvez pas le désactiver.

## Étiquettes

Vous pouvez personnaliser les étiquettes en fonction des problèmes. Cela inclut la modification de l'étiquette et le changement de couleur. Les étiquettes peuvent vous aider à classer et à organiser votre travail.

## Création d'une étiquette

Dans CodeCatalyst, vous créez des étiquettes en les ajoutant soit lorsque vous créez un nouveau numéro, soit lorsque vous modifiez un numéro existant. Pour plus d'informations, consultez [Création d'un problème dans CodeCatalyst](#) et [Rédaction et collaboration sur des problèmes dans CodeCatalyst](#).

## Modification d'une étiquette

Pour modifier le nom ou la couleur d'une étiquette existante, procédez comme suit.

Pour modifier une étiquette

1. Dans le volet de navigation, sélectionnez Problèmes.
2. Choisissez Problèmes actifs pour ouvrir le menu déroulant du sélecteur d'affichage des problèmes et choisissez Paramètres.
3. La vignette Étiquettes contient la liste des étiquettes utilisées dans le projet. Cliquez sur l'icône d'édition à côté de l'étiquette que vous souhaitez modifier. Effectuez une ou plusieurs des actions suivantes :
  - a. Modifiez le nom de l'étiquette.
  - b. Pour changer de couleur, choisissez la roue chromatique. Utilisez le sélecteur pour choisir une nouvelle couleur.
4. Pour enregistrer les modifications que vous avez apportées à l'étiquette, cliquez sur l'icône en forme de coche.
5. L'étiquette modifiée est désormais visible dans la liste des étiquettes disponibles. Vous pouvez également voir combien de numéros utilisent cette étiquette.

### Note

Vous pouvez choisir le numéro affiché à côté de chaque étiquette pour accéder à la page Tous les numéros et voir tous les numéros contenant cette étiquette.

## Supprimer une étiquette

Il est actuellement impossible de supprimer une étiquette de problème dans CodeCatalyst. Si vous supprimez une étiquette de tous les numéros, elle apparaîtra dans la section Étiquettes non utilisées

de la section Étiquettes des paramètres du numéro. Les libellés non utilisés apparaissent à la fin de la liste des libellés lorsque vous utilisez des filtres ou que vous ajoutez des libellés à un problème. Vous trouverez un aperçu de toutes les étiquettes (utilisées et non utilisées) et des problèmes qui les contiennent dans les paramètres du problème.

## Champs personnalisés

Vous pouvez créer des champs personnalisés pour vous aider à organiser et à visualiser le travail relatif à votre projet. Les champs personnalisés sont ajoutés à la liste des filtres disponibles dans Filter afin que vous puissiez filtrer les problèmes par champs personnalisés. Les champs personnalisés sont des paires de nom et de valeur. Vous filtrez en fonction du nom du champ personnalisé, puis de la valeur de ce champ personnalisé.

Un problème peut comporter plusieurs champs personnalisés.

### Création d'un champ personnalisé

Dans CodeCatalyst, vous créez des champs personnalisés en les ajoutant soit lorsque vous créez un problème, soit lorsque vous modifiez un problème existant. Pour plus d'informations, consultez [Création d'un problème dans CodeCatalyst](#) et [Rédaction et collaboration sur des problèmes dans CodeCatalyst](#).

### Supprimer un champ personnalisé

Pour supprimer un champ personnalisé, vous devez supprimer le champ personnalisé pour chaque numéro auquel il est ajouté. Lorsqu'un champ personnalisé est supprimé, il ne s'affiche plus dans Filter. Vous pouvez utiliser des filtres pour afficher tous les problèmes dans un champ personnalisé et les supprimer en les modifiant. Pour plus d'informations, consultez [Recherche et affichage des problèmes](#) et [Modification d'un numéro](#)

## Afficher et gérer les pièces jointes

Vous pouvez consulter un tableau avec chaque pièce jointe ajoutée aux problèmes de votre projet dans les paramètres des problèmes. Ce tableau contient des informations détaillées sur chaque pièce jointe, notamment des informations telles que le type de contenu, la date à laquelle elle a été ajoutée, le problème auquel elle a été ajoutée, son statut et la taille du fichier.

Ce tableau peut être utilisé pour identifier facilement les pièces jointes volumineuses relatives à des problèmes terminés ou archivés afin de les supprimer afin de libérer de l'espace de stockage.

**⚠ Important**

Les pièces jointes aux numéros ne sont ni scannées ni analysées par Amazon CodeCatalyst. Tout utilisateur peut ajouter une pièce jointe à un problème susceptible de contenir du code ou du contenu malveillants. Assurez-vous que les utilisateurs connaissent les meilleures pratiques en matière de gestion des pièces jointes et de protection contre le code, le contenu ou les virus malveillants.

Pour afficher et gérer toutes les pièces jointes à un problème dans un projet

1. Dans le volet de navigation, sélectionnez Problèmes.
2. Cliquez sur l'icône représentant des points de suspension, puis sur Paramètres.
3. Choisissez l'onglet Pièces jointes.

## Quotas pour les numéros parus dans CodeCatalyst

Le tableau suivant décrit les quotas et les limites applicables aux problèmes sur Amazon CodeCatalyst. Pour plus d'informations sur les quotas sur Amazon CodeCatalyst, consultez [Quotas pour CodeCatalyst](#).

| Ressource                                    | Quota par défaut   |
|--|--|
| Problèmes actifs                             | Maximum de 1 000 par projet.   |
| Taille de la pièce jointe                    | Maximum de 500 Mo par pièce jointe.<br><br>Le stockage total maximal des pièces jointes dépend des limites de stockage globales de votre espace. Pour plus d'informations, consultez <a href="#">Tarification d'</a> . |
| Nombre total de numéros (actifs et archivés) | Maximum de 100 000 par projet.   |
| Vues enregistrées                            | Maximum de 50 vues de problèmes enregistrées par projet.   |

| Ressource  | Quota par défaut                        |
|--|---|
| Nombre de pull requests que vous pouvez associer à un problème | Maximum de 50 pull requests par numéro. |
| Statuts (par projet)   | Maximum de 50 par projet.               |
| Statuts (par numéro)   | Maximum de 50 par numéro.               |
| Étiquettes (par projet)  | Maximum de 200 par projet.              |
| Étiquettes (par numéro)  | Maximum de 50 par numéro.               |
| Champs personnalisés (par numéro)                              | Maximum de 50 par numéro.               |
| Cessionnaires  | Maximum de 10 par numéro.               |
| Commentaires   | Maximum de 1 000 par numéro.            |
| Tâches   | Maximum de 100 par numéro.              |

# Identité, autorisations et accès dans CodeCatalyst

Lorsque vous vous connectez à Amazon CodeCatalyst pour la première fois, vous créez un AWS Builder ID. AWS Les identifiants de constructeur n'existent pas dans AWS Identity and Access Management. Le nom d'utilisateur que vous choisissez lors de votre première connexion devient votre identifiant unique pour votre identité.

Dans CodeCatalyst, vous pouvez vous connecter pour la première fois de deux manières :

- Dans le cadre de la création d'un espace.
- Dans le cadre de l'acceptation d'une invitation à un projet ou à un espace dans CodeCatalyst.

Le ou les rôles associés à votre identité déterminent les actions que vous pouvez effectuer dans ce cadre CodeCatalyst. Les rôles de projet, tels qu'administrateur de projet et contributeur, sont spécifiques à un projet. Vous pouvez donc avoir un rôle dans un projet et un rôle différent dans un autre projet. Si vous créez un espace, le rôle d'administrateur de l'espace vous est CodeCatalyst automatiquement attribué. Lorsque les utilisateurs acceptent des invitations à un projet, CodeCatalyst ajoutent ces identités à l'espace et leur attribuent le rôle d'accès limité. Lorsque vous invitez des utilisateurs à rejoindre des projets, vous choisissez le rôle que vous souhaitez leur attribuer dans le projet, qui détermine les actions qu'ils peuvent et ne peuvent pas entreprendre dans le cadre du projet. La plupart des utilisateurs travaillant sur un projet n'ont besoin que du rôle de contributeur pour effectuer leurs tâches. Pour plus d'informations, consultez [Utilisation des rôles sur Amazon CodeCatalyst](#).

Outre le rôle de projet, les utilisateurs d'un projet ont besoin d'un jeton d'accès personnel (PAT) pour accéder aux référentiels sources d'un projet lorsqu'ils utilisent des clients Git ou des environnements de développement intégrés (IDE). Les membres du projet peuvent utiliser ce PAT avec des applications tierces en tant que mot de passe spécifique à l'application associé à leur CodeCatalyst identité. Par exemple, lorsque vous clonez un dépôt source sur un ordinateur local, vous devez fournir un PAT ainsi que votre nom CodeCatalyst d'utilisateur.

Vous pouvez configurer l'accès entre CodeCatalyst et les AWS ressources en utilisant un [rôle de service](#) pour effectuer des actions telles que l'accès aux AWS CloudFormation piles et aux ressources lorsque vous déployez des actions dans des flux de travail. Vous devez configurer l'accès CodeCatalyst et les AWS ressources pour les actions de flux de travail incluses dans les modèles de projet à exécuter.



## Rubriques

- [Utilisation des rôles sur Amazon CodeCatalyst](#)
- [Gestion des jetons d'accès personnels sur Amazon CodeCatalyst](#)
- [Authentification multifactorielle \(MFA\) sur Amazon CodeCatalyst](#)
- [Sécurité sur Amazon CodeCatalyst](#)
- [Surveillance sur Amazon CodeCatalyst](#)
- [Quotas d'identité, d'autorisation et d'accès dans CodeCatalyst](#)
- [Résolution des problèmes](#)

## Utilisation des rôles sur Amazon CodeCatalyst

Dans Amazon CodeCatalyst, vous pouvez attribuer des rôles aux utilisateurs à la fois au niveau du projet et au niveau de l'espace. Dans un projet, un rôle indique ce qu'un utilisateur est autorisé à faire dans un projet avec les ressources nécessaires à ce projet. Les utilisateurs deviennent membres d'un espace lorsqu'ils rejoignent un projet. Vous pouvez ajouter ou supprimer des utilisateurs en tant qu'administrateurs d'un espace. Le rôle d'administrateur de l'espace dispose des autorisations les plus étendues de tous les rôles dans CodeCatalyst. Il est recommandé d'attribuer aux utilisateurs les autorisations les plus restreintes nécessaires à l'exécution de leurs tâches.

Vous pouvez attribuer des rôles aux utilisateurs de l'espace. Vous pouvez également attribuer des rôles aux utilisateurs dans les projets dont ils sont membres. Chaque utilisateur ne peut avoir qu'un seul rôle dans un projet ou un espace, mais les utilisateurs peuvent avoir des rôles différents dans chaque projet et dans chaque espace. Par exemple, un utilisateur peut avoir le rôle d'administrateur de projet dans un projet et le rôle de contributeur dans un autre projet.

## Rubriques

- [Types de rôles](#)
- [Autorisations disponibles pour chaque rôle](#)
- [Afficher et modifier les rôles des utilisateurs](#)

## Types de rôles

Trois rôles sont disponibles pour un espace :

- Administrateur de l'espace

- Utilisateur avancé
- Accès limité

Les utilisateurs qui acceptent une invitation à un projet se voient automatiquement attribuer le rôle d'accès limité dans l'espace contenant le projet.

Quatre rôles sont disponibles pour les membres d'un projet :

- Administrateur du projet
- Participant
- Réviseur
- Lecture seule

Lorsque vous ajoutez un utilisateur à un projet, le rôle Accès limité lui est CodeCatalyst automatiquement attribué. Si vous supprimez un utilisateur de tous les projets, le rôle d'accès limité est CodeCatalyst automatiquement supprimé de cet utilisateur.

## Rôle d'administrateur de l'espace

Le rôle d'administrateur de l'espace est le rôle le plus puissant de CodeCatalyst. Attribuez le rôle d'administrateur de l'espace uniquement aux utilisateurs qui doivent administrer tous les aspects d'un espace, car ce rôle dispose de toutes les autorisations CodeCatalyst. Les utilisateurs dotés du rôle d'administrateur de l'espace sont les seuls à pouvoir ajouter ou supprimer d'autres utilisateurs du rôle d'administrateur de l'espace et supprimer l'espace.

Lorsque vous créez un espace, le rôle d'administrateur de l'espace vous est CodeCatalyst automatiquement attribué. Il est recommandé d'ajouter ce rôle à au moins un autre utilisateur qui peut jouer ce rôle au cas où le créateur de l'espace d'origine ne serait pas disponible.

## Rôle d'utilisateur avancé

Le rôle d'utilisateur avancé est le deuxième rôle le plus puissant dans CodeCatalyst les espaces, mais il ne permet pas d'accéder aux projets d'un espace. Il est conçu pour les utilisateurs qui doivent être en mesure de créer des projets dans un espace et d'aider à gérer les utilisateurs et les ressources de l'espace. Attribuez le rôle d'utilisateur avancé aux utilisateurs qui sont des chefs d'équipe ou des responsables qui doivent être capables de créer des projets et de gérer les utilisateurs dans l'espace dans le cadre de leur travail.

## Rôle à accès limité

Le rôle d'accès limité est le rôle que la plupart des utilisateurs auront dans CodeCatalyst les espaces. Il s'agit du rôle attribué automatiquement aux utilisateurs lorsqu'ils acceptent une invitation à participer à un projet dans un espace. Il fournit les autorisations limitées dont ils ont besoin pour travailler dans l'espace qui contient ce projet. Attribuez le rôle d'accès limité aux utilisateurs que vous invitez directement dans l'espace, sauf si leur travail les oblige à gérer certains aspects de l'espace.

## Rôle d'administrateur de projet

Le rôle d'administrateur de projet est le rôle le plus puissant d'un CodeCatalyst projet. Attribuez ce rôle uniquement aux utilisateurs qui doivent administrer tous les aspects d'un projet, y compris la modification des paramètres du projet, la gestion des autorisations du projet et la suppression de projets.

Les rôles du projet ne disposent d'aucune autorisation au niveau de l'espace. Par conséquent, les utilisateurs dotés du rôle d'administrateur de projet ne peuvent pas créer de projets supplémentaires. Seuls les utilisateurs dotés du rôle d'administrateur de l'espace ou d'utilisateur avancé peuvent créer des projets.

### Note

Le rôle d'administrateur de l'espace dispose de toutes les autorisations nécessaires CodeCatalyst.

## Rôle du contributeur

Le rôle de contributeur est destiné à la majorité des membres d'un CodeCatalyst projet. Attribuez ce rôle aux utilisateurs qui doivent être en mesure de travailler avec du code, des flux de travail, des problèmes et des actions dans le cadre d'un projet.

## Rôle du réviseur

Le rôle de réviseur est destiné aux utilisateurs qui doivent être en mesure d'interagir avec les ressources d'un projet, telles que les pull requests et les problèmes, mais pas de créer et de fusionner du code, de créer des flux de travail ou de démarrer ou d'arrêter des exécutions de flux de travail dans un CodeCatalyst projet. Attribuez le rôle de réviseur aux utilisateurs qui doivent être en mesure d'approuver et de commenter les pull requests, de créer, de mettre à jour, de résoudre et de commenter les problèmes, ainsi que de consulter le code et les flux de travail d'un projet.

## Rôle en lecture seule

Le rôle en lecture seule est destiné aux utilisateurs qui ont besoin de consulter les ressources et le statut des ressources, mais qui ne doivent pas interagir avec elles ni contribuer directement au projet. Les utilisateurs dotés de ce rôle ne peuvent pas créer de ressources dans CodeCatalyst, mais ils peuvent les afficher et les copier, par exemple en clonant des référentiels ou en téléchargeant des pièces jointes à des problèmes sur un ordinateur local. Attribuez le rôle Lecture seule aux utilisateurs qui ont besoin de consulter les ressources et l'état du projet, mais qui ne doivent pas interagir directement avec celui-ci.

## Autorisations disponibles pour chaque rôle
























Le tableau suivant indique les autorisations disponibles pour chaque CodeCatalyst rôle. Utilisez les liens pour accéder à l'ensemble d'autorisations approprié.

- [Space permissions](#)
- [Extensions permissions](#)
- [Project permissions](#)
- [Source repository permissions](#)
- [Dev Environment permissions](#)
- [Package repository and package permissions](#)
- [Workflow permissions](#)
- [Issues permissions](#)
- [Custom blueprint permissions](#)
- [Notifications permissions](#)
- [Search permissions](#)






|              |                                   |                           |                     |                                 |                      |                  |                       |
|--------------|-----------------------------------|---------------------------|---------------------|---------------------------------|----------------------|------------------|-----------------------|
| Autorisation | Rôle d'administrateur de l'espace | Rôle d'utilisateur avancé | Rôle à accès limité | Rôle d'administrateur de projet | Rôle du contributeur | Rôle du réviseur | Rôle en lecture seule |
|--------------|-----------------------------------|---------------------------|---------------------|---------------------------------|----------------------|------------------|-----------------------|

### Autorisations relatives aux espaces

| Autorisation  | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Créer de l'espace   |    |    |    |    |    |    |    |
| Modifier les informations de facturation de l'espace        |    |    |    |    |    |    |    |
| Configurer et activer l'authentification unique             |    |    |    |    |    |    |    |
| Supprimer l'authentification unique                         |  |  |  |  |  |  |  |
| Activer les fonctionnalités d'IA génératives pour un espace |  |  |  |  |  |  |  |








| Autorisation   | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Désactiver les fonctionnalités d'IA génératives pour un espace       |    |    |    |    |    |    |    |
| Supprimer un espace  |    |    |    |    |    |    |    |
| Ajouter d'autres utilisateurs au rôle d'administrateur de l'espace   |   |   |   |   |   |   |   |
| Supprimer d'autres utilisateurs du rôle d'administrateur de l'espace |  |  |  |  |  |  |  |

































| Autorisation  | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Créer une équipe                                      |    |    |    |    |    |    |    |
| Supprimer l'équipe                                    |    |    |    |    |    |    |    |
| Équipe de mise à jour                                 |    |    |    |    |    |    |    |
| Désactiver les ressources de la machine pour l'espace |    |    |    |    |    |    |    |
| Activez les ressources de la machine pour l'espace    |  |  |  |  |  |  |  |
| Création d'un projet                                  |  |  |  |  |  |  |  |




| Autorisation  | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Associer une connexion à un compte AWS à l'espace                 |    |    |    |    |    |    |    |
| Mettre à jour une connexion à un compte AWS                       |    |    |    |    |    |    |    |
| Dissocier une connexion à un compte AWS de l'espace               |  |  |  |  |  |  |  |
| Supprimer une connexion à un compte AWS et la retirer de l'espace |  |  |  |  |  |  |  |
























































| Autorisation                                   | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Invitez d'autres personnes dans l'espace       |    |    |    |    |    |    |    |
| Création d'une connexion VPC                   |    |    |    |    |    |    |    |
| Modifier la connexion VPC                      |    |    |    |    |    |    |    |
| Supprimer la connexion VPC                     |  |  |  |  |  |  |  |
| Afficher les journaux d'activité dans l'espace |  |  |  |  |  |  |  |

| Autorisation                             | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Afficher les connexions aux comptes AWS  |    |    |    |    |    |    |    |
| Afficher les incidents pour CodeCatalyst |    |    |    |    |    |    |    |
| Afficher l'espace                        |  |  |  |  |  |  |  |
| Afficher les équipes                     |  |  |  |  |  |  |  |
| Afficher les connexions VPC              |  |  |  |  |  |  |  |
| Autorisations relatives aux extensions   | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |











































| Autorisation                 | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|------------------------------|---|---|---|---|---|---|---|
| Installer des extensions     |    |    |    |    |    |    |    |
| Mettre à jour les extensions |    |    |    |    |    |    |    |
| Supprimer des extensions     |    |    |    |    |    |    |    |
| Connecter un GitHub compte   |  |  |  |  |  |  |  |
| Déconnecter un GitHub compte |  |  |  |  |  |  |  |
| Connecter un site Jira       |  |  |  |  |  |  |  |
| Déconnecter un site Jira     |  |  |  |  |  |  |  |

| Autorisation  | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Afficher les détails de configuration des extensions installées |    |    |    |    |    |    |    |
| Afficher les extensions   |    |    |    |    |    |    |    |
| Permissions du projet   | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
| Modifier les paramètres du projet                               |  |  |  |  |  |  |  |
| Désactiver les ressources de la machine pour le projet          |  |  |  |  |  |  |  |

| Autorisation  | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Activer les ressources de la machine pour le projet |    |    |    |    |    |    |    |
| Supprimer le projet                                 |    |    |    |    |    |    |    |
| Inviter des utilisateurs à rejoindre un projet      |    |    |    |    |    |    |    |
| Modifier les rôles des utilisateurs dans un projet  |  |  |  |  |  |  |  |
| Supprimer des utilisateurs d'un projet              |  |  |  |  |  |  |  |




















































| Autorisation                                | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Ajouter une équipe à un projet              |    |    |    |    |    |    |    |
| Supprimer une équipe d'un projet            |    |    |    |    |    |    |    |
| Modifier le rôle de l'équipe dans le projet |    |    |    |    |    |    |    |
| Voir le projet                              |  |  |  |  |  |  |  |
| Afficher l'activité du projet               |  |  |  |  |  |  |  |
| Afficher les équipes du projet              |  |  |  |  |  |  |  |
| Afficher les plans                          |  |  |  |  |  |  |  |







| Autorisation                           | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Autorisations du référentiel source    |   |   |   |   |   |   |   |
| Création de référentiels               |    |    |    |    |    |    |    |
| Référentiels de liens                  |    |    |    |    |    |    |    |
| Dissocier les référentiels             |   |   |   |   |   |   |   |
| Supprimer des référentiels             |  |  |  |  |  |  |  |
| Modifier les paramètres du référentiel |  |  |  |  |  |  |  |



































| Autorisation                           | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Afficher les référentiels              |    |    |    |    |    |    |    |
| Afficher les paramètres du référentiel |    |    |    |    |    |    |    |
| Référentiels de clonage                |    |    |    |    |    |    |    |
| Création de branches                   |  |  |  |  |  |  |  |
| Création de règles de branche          |  |  |  |  |  |  |  |
| Modifier la branche par défaut         |  |  |  |  |  |  |  |













| Autorisation                         | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--------------------------------------|---|---|---|---|---|---|---|
| Supprimer des branches               |    |    |    |    |    |    |    |
| Fusionner des branches               |    |    |    |    |    |    |    |
| Mettre à jour les règles de branche  |    |    |    |    |    |    |    |
| Afficher les succursales             |   |   |   |   |   |   |   |
| Afficher les règles de la succursale |  |  |  |  |  |  |  |
| Création de dossiers                 |  |  |  |  |  |  |  |
| Supprimer des dossiers               |  |  |  |  |  |  |  |





















| Autorisation                 | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|------------------------------|---|---|---|---|---|---|---|
| Modifier des dossiers        |    |    |    |    |    |    |    |
| Afficher les dossiers        |    |    |    |    |    |    |    |
| Création de fichiers         |    |    |    |    |    |    |    |
| Supprimer les fichiers       |    |    |    |    |    |    |    |
| Modifier des fichiers        |  |  |  |  |  |  |  |
| Afficher les fichiers        |  |  |  |  |  |  |  |
| Créer et publier des commits |  |  |  |  |  |  |  |
| Afficher les validations     |  |  |  |  |  |  |  |






| Autorisation  | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Créer des pull requests                                       |    |    |    |    |    |    |    |
| Créer des règles d'approbation pour les pull requests         |    |    |    |    |    |    |    |
| Remplacer les exigences de fusion pour les pull requests      |    |    |    |    |    |    |    |
| Mettre à jour les pull requests                               |  |  |  |  |  |  |  |
| Mettre à jour les règles d'approbation pour les pull requests |  |  |  |  |  |  |  |

| Autorisation   | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Afficher des demandes d'extraction                       |    |    |    |    |    |    |    |
| Afficher les règles d'approbation pour les pull requests |    |    |    |    |    |    |    |
| Clôturer les pull requests                               |   |   |   |   |   |   |   |
| Approuver les pull requests                              |  |  |  |  |  |  |  |
| Commentez les pull requests                              |  |  |  |  |  |  |  |

| Autorisation   | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Interagissez avec Amazon Q dans les commentaires sur les pull requests |    |    |    |    |    |    |    |
| Création d'une révision pour une pull request créée par Amazon Q       |    |    |    |    |    |    |    |
| Lier un problème à une pull request                                    |  |  |  |  |  |  |  |
| Dissocier un problème d'une pull request                               |  |  |  |  |  |  |  |















| Autorisation  | Rôle d'administrateur de l'espace | Rôle d'utilisateur avancé | Rôle à accès limité | Rôle d'administrateur de projet | Rôle du contributeur | Rôle du réviseur | Rôle en lecture seule |
|---|-----------------------------------|---------------------------|---------------------|---------------------------------|----------------------|------------------|-----------------------|
| Autorisations de l'environnement de développement       |                                   |                           |                     |                                 |                      |                  |                       |
| Créez votre propre environnement de développement       |                                   |                           |                     |                                 |                      |                  |                       |
| Arrêtez votre propre environnement de développement     |                                   |                           |                     |                                 |                      |                  |                       |
| Stop Dev Environnements créés par d'autres utilisateurs |                                   |                           |                     |                                 |                      |                  |                       |



























| Autorisation   | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Reprenez votre propre environnement de développement                         |    |    |    |    |    |    |    |
| Visualisez vos propres environnements de développement                       |    |    |    |    |    |    |    |
| Afficher les environnements de développement créés par d'autres utilisateurs |  |  |  |  |  |  |  |





























| Autorisation   | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Modifiez votre propre environnement de développement                         |    |    |    |    |    |    |    |
| Modifier les environnements de développement créés par d'autres utilisateurs |    |    |    |    |    |    |    |
| Supprimer votre propre environnement de développement                        |  |  |  |  |  |  |  |






































| Autorisation  | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Supprimer les environnements de développement créés par d'autres utilisateurs   |    |    |    |    |    |    |    |
| Création d'un fichier de développement pour les environnements de développement |    |    |    |    |    |    |    |
| Modifier un fichier de développement pour les environnements de développement   |  |  |  |  |  |  |  |























| Autorisation   | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Supprimer un fichier de développement pour les environnements de développement |  |  |  |  |  |  |  |
| Afficher un fichier de développement pour les environnements de développement  |  |  |  |  |  |  |  |
| Référentiel de packages et autorisations de packages                           |   |   |   |   |   |   |   |







| Autorisation  | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Création d'un référentiel de packages               |    |    |    |    |    |    |    |
| Afficher les référentiels de packages               |    |    |    |    |    |    |    |
| Modifier le référentiel de packages                 |    |    |    |    |    |    |    |
| Supprimer le référentiel de packages                |  |  |  |  |  |  |  |
| Création d'un référentiel de packages de passerelle |  |  |  |  |  |  |  |

| Autorisation  | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Afficher les référentiels de packages Gateway           |    |    |    |    |    |    |    |
| Supprimer le référentiel de packages de passerelle      |    |    |    |    |    |    |    |
| Ajouter un référentiel de packages en amont             |  |  |  |  |  |  |  |
| Modifier l'ordre de recherche des référentiels en amont |  |  |  |  |  |  |  |

| Autorisation  | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Supprimer le référentiel de packages en amont           |    |    |    |    |    |    |    |
| Se connecter à un référentiel de packages               |    |    |    |    |    |    |    |
| Lire des packages à partir d'un référentiel de packages |  |  |  |  |  |  |  |
| Publier des packages dans un référentiel de packages    |  |  |  |  |  |  |  |

| Autorisation  | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Lire et conserver les packages depuis un référentiel en amont |    |    |    |    |    |    |    |
| Afficher les packages   |    |    |    |    |    |    |    |
| Afficher les versions des packages                            |   |   |   |   |   |   |   |
| Afficher les ressources relatives à la version du package     |  |  |  |  |  |  |  |

| Autorisation   | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Lister les dépendances entre les versions des packages |    |    |    |    |    |    |    |
| État de version du package de mise à jour              |    |    |    |    |    |    |    |
| Mettre à jour la configuration d'origine du package    |  |  |  |  |  |  |  |
| Supprimer la version du package                        |  |  |  |  |  |  |  |
| Autorisations relatives au flux                        | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |

| Autorisation                          | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---------------------------------------|---|---|---|---|---|---|---|
| Créer un flux de travail              |    |    |    |    |    |    |    |
| Mettre à jour le workflow             |    |    |    |    |    |    |    |
| Supprimer le flux de travail          |    |    |    |    |    |    |    |
| Démarrer le flux de travail           |    |    |    |    |    |    |    |
| Arrêter le workflow                   |  |  |  |  |  |  |  |
| Créer des secrets de flux de travail  |  |  |  |  |  |  |  |
| Mettre à jour les secrets du workflow |  |  |  |  |  |  |  |
































































| Autorisation                             | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Supprimer les secrets du flux de travail |    |    |    |    |    |    |    |
| Créer des environnements                 |    |    |    |    |    |    |    |
| Supprimer des environnements             |    |    |    |    |    |    |    |
| Création d'une flotte                    |  |  |  |  |  |  |  |
| Mettre à jour la flotte                  |  |  |  |  |  |  |  |
| Supprimer la flotte                      |  |  |  |  |  |  |  |








































| Autorisation   | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Gérer les ressources du flux de travail dans d'autres comptes    |    |    |    |    |    |    |    |
| Associer une connexion VPC à un environnement                    |    |    |    |    |    |    |    |
| Dissocier une connexion VPC à un environnement                   |  |  |  |  |  |  |  |
| Associer un environnement connecté à un VPC à un flux de travail |  |  |  |  |  |  |  |

























| Autorisation  | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Dissocier un environnement connecté à un VPC d'un flux de travail |    |    |    |    |    |    |    |
| Suivi d'un commit dans un flux de travail                         |    |    |    |    |    |    |    |
| Affichage des environnements                                      |  |  |  |  |  |  |  |
| Afficher les journaux des actions de construction                 |  |  |  |  |  |  |  |
| Afficher les flottes  |  |  |  |  |  |  |  |

| Autorisation  | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Afficher les journaux des actions de test             |    |    |    |    |    |    |    |
| Afficher le flux de travail                           |    |    |    |    |    |    |    |
| Afficher les exécutions de flux de travail            |    |    |    |    |    |    |    |
| Afficher les résultats d'exécution du flux de travail |  |  |  |  |  |  |  |
| Afficher les secrets du flux de travail               |  |  |  |  |  |  |  |






























| Autorisation                     | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|----------------------------------|---|---|---|---|---|---|---|
| Délivrer des autorisations       | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
| Créer un problème                |    |    |    |    |    |    |    |
| Problème de mise à jour          |    |    |    |    |    |    |    |
| Afficher les numéros             |    |    |    |    |    |    |    |
| Archiver un problème             |  |  |  |  |  |  |  |
| Attribuer un problème à Amazon Q |  |  |  |  |  |  |  |





























| Autorisation   | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Interagissez avec Amazon Q dans les commentaires sur un problème |    |    |    |    |    |    |    |
| Annuler l'attribution d'Amazon Q à un problème                   |    |    |    |    |    |    |    |
| Problèmes de mise à jour créés par d'autres utilisateurs         |  |  |  |  |  |  |  |
| Afficher les commentaires sur un problème                        |  |  |  |  |  |  |  |

| Autorisation                                 | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Créer un commentaire sur un problème         |    |    |    |    |    |    |    |
| Mettre à jour un commentaire sur un problème |    |    |    |    |    |    |    |
| Création d'une étiquette                     |    |    |    |    |    |    |    |
| Mettre à jour une étiquette                  |  |  |  |  |  |  |  |
| Afficher les étiquettes                      |  |  |  |  |  |  |  |
| Ajouter une étiquette à un numéro            |  |  |  |  |  |  |  |
| Supprimer l'étiquette d'un numéro            |  |  |  |  |  |  |  |
























| Autorisation   | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Création d'un statut personnalisé pour les problèmes |    |    |    |    |    |    |    |
| Mettre à jour un statut personnalisé                 |    |    |    |    |    |    |    |
| Afficher un statut personnalisé                      |  |  |  |  |  |  |  |
| Déplacer un statut personnalisé                      |  |  |  |  |  |  |  |
| Désactiver un statut personnalisé                    |  |  |  |  |  |  |  |



















| Autorisation                             | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Ajouter une pièce jointe à un numéro     |    |    |    |    |    |    |    |
| Afficher une pièce jointe à un problème  |    |    |    |    |    |    |    |
| Supprimer une pièce jointe d'un problème |    |    |    |    |    |    |    |
| Associer une pull request à un problème  |  |  |  |  |  |  |  |
| Dissocier une pull request d'un problème |  |  |  |  |  |  |  |
| Associer un projet Jira                  |  |  |  |  |  |  |  |

| Autorisation                             | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Dissocier un projet Jira                 |    |    |    |    |    |    |    |
| Autorisations de plan personnalisées     |   |   |   |   |   |   |   |
| Créer un projet personnalisé             |    |    |    |    |    |    |    |
| Publier un aperçu du plan personnalisé   |  |  |  |  |  |  |  |
| Dépublier un aperçu du plan personnalisé |  |  |  |  |  |  |  |

| Autorisation  | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Publier un plan personnalisé                                    |    |    |    |    |    |    |    |
| Dépublier un plan personnalisé                                  |    |    |    |    |    |    |    |
| Ajouter un plan personnalisé à un catalogue de plans spatiaux   |    |    |    |    |    |    |    |
| Supprimer un plan personnalisé d'un catalogue de plans spatiaux |  |  |  |  |  |  |  |

| Autorisation   | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Gérer les autorisations de publication pour un plan personnalisé |    |    |    |    |    |    |    |
| Gérez la version du catalogue pour un plan personnalisé          |    |    |    |    |    |    |    |
| Mettre à jour un plan personnalisé                               |  |  |  |  |  |  |  |
| Supprimer une version personnalisée du plan                      |  |  |  |  |  |  |  |

| Autorisation   | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|--|---|---|---|---|---|---|---|
| Supprimer un plan personnalisé                           |    |    |    |    |    |    |    |
| Appliquer un plan personnalisé à un projet               |    |    |    |    |    |    |    |
| Dissocier un plan personnalisé d'un projet               |    |    |    |    |    |    |    |
| Mettre à jour la version d'un plan personnalisé appliqué |  |  |  |  |  |  |  |
| Modifier l'alias d'un plan personnalisé                  |  |  |  |  |  |  |  |

| Autorisation                              | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Afficher les plans personnalisés publiés  |    |    |    |    |    |    |    |
| Autorisations relatives aux notifications |   |   |   |   |   |   |   |
| Configurer le canal de notification       |   |   |   |   |   |   |   |
| Supprimer le canal de notification        |  |  |  |  |  |  |  |
| Modifier les paramètres de notification   |  |  |  |  |  |  |  |

| Autorisation  | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---|---|---|---|---|---|---|---|
| Afficher les paramètres de notification   |    |    |    |    |    |    |    |
| Recevez automatiquement des notifications concernant les CodeCatalyst incidents |    |    |    |    |    |    |    |
| Configurer les notifications par e-mail pour votre compte de messagerie associé |  |  |  |  |  |  |  |

| Autorisation                          | Rôle d'administrateur de l'espace   | Rôle d'utilisateur avancé   | Rôle à accès limité   | Rôle d'administrateur de projet   | Rôle du contributeur  | Rôle du réviseur  | Rôle en lecture seule   |
|---------------------------------------|---|---|---|---|---|---|---|
| Rechercher dans un projet             |  |  |  |  |  |  |  |
| Effectuez une recherche dans l'espace |  |  |  |  |  |  |  |

## Afficher et modifier les rôles des utilisateurs

Vous pouvez consulter le rôle attribué à un utilisateur. Cela vous permet de comprendre les actions qu'ils peuvent entreprendre dans le cadre d'un projet. Vous pouvez également modifier leur rôle s'ils ont besoin d'autorisations supplémentaires.

Pour afficher le rôle d'un utilisateur dans un projet

1. Accédez au projet dans lequel vous souhaitez afficher les rôles associés à chaque membre du projet.

### Tip

Vous pouvez choisir le projet à afficher dans la barre de navigation supérieure.

2. Dans le volet de navigation, sélectionnez Paramètres du projet.
3. Dans l'onglet Membres, le rôle de chaque membre du projet est affiché dans Rôle.



## Pour modifier les rôles des utilisateurs dans un projet

1. Accédez au projet dans lequel vous souhaitez modifier les rôles associés aux membres du projet.

### Tip

Vous pouvez choisir le projet à afficher dans la barre de navigation supérieure.

2. Dans le volet de navigation, sélectionnez Paramètres du projet.
3. Dans l'onglet Membres, dans Membres du projet, choisissez les utilisateurs dont vous souhaitez modifier les rôles. Choisissez Action, puis Modifier le rôle.
4. Dans Rôle, choisissez le rôle du projet, puis cliquez sur Confirmer.

## Affichage et modification des rôles dans l'espace

Tous les utilisateurs qui acceptent des invitations à un projet CodeCatalyst deviennent membres de l'espace du projet. Vous pouvez consulter la liste des membres de l'espace. Vous pouvez modifier les rôles des utilisateurs en passant d'un accès limité à un rôle d'administrateur de l'espace afin de mieux gérer votre espace et ses ressources. Le rôle d'administrateur de l'espace est le seul dans lequel les utilisateurs peuvent créer des projets CodeCatalyst.

### Warning

Le rôle d'administrateur de l'espace est le rôle le plus puissant de CodeCatalyst. Les utilisateurs dotés de ce rôle peuvent effectuer n'importe quelle action CodeCatalyst, y compris supprimer l'espace. Attribuez ce rôle uniquement aux utilisateurs qui ont besoin de ce niveau d'accès à votre espace. Pour plus d'informations, consultez [Rôle d'administrateur de l'espace](#).

## Pour modifier le rôle d'un utilisateur dans l'espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Naviguez vers l'espace.

**i** Tip

Si vous appartenez à plusieurs espaces, vous pouvez choisir l'espace à afficher dans la barre de navigation supérieure.

3. Choisissez l'onglet Membres.
4. Choisissez l'utilisateur dont vous souhaitez modifier le rôle, puis choisissez Changer de rôle.
5. Dans Modifier le rôle, choisissez le rôle que vous souhaitez attribuer, puis cliquez sur Confirmer.

## Gestion des jetons d'accès personnels sur Amazon CodeCatalyst

Pour accéder à certaines CodeCatalyst ressources, telles que les référentiels de sources, sur un ordinateur local doté d'un client Git ou d'un environnement de développement intégré (IDE), vous devez entrer un mot de passe spécifique à l'application. Vous pouvez créer un jeton d'accès personnel (PAT) à utiliser à cette fin. Les PAT que vous créez sont associés à votre identité d'utilisateur dans tous les espaces et projets qu'ils contiennent. CodeCatalyst Vous pouvez créer plusieurs PAT pour votre CodeCatalyst identité.

Vous pouvez consulter les noms et les dates d'expiration des PAT que vous avez créés, et vous pouvez supprimer ceux dont vous n'avez plus besoin. Vous ne pouvez copier le secret PAT qu'au moment de sa création.

**i** Note

Par défaut, les PAT expirent au bout d'un an.

### Création de PAT

Les PAT sont associés à votre identité d'utilisateur dans CodeCatalyst. Vous ne pouvez copier un secret PAT qu'au moment de sa création.

#### Création de PAT (console)

Vous pouvez utiliser la console pour créer des PAT dans CodeCatalyst.

## Pour créer un jeton d'accès personnel (console)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la barre de menu supérieure, choisissez votre badge de profil, puis sélectionnez Mes paramètres. La page CodeCatalyst Mes paramètres s'ouvre.

### Tip

Vous pouvez également accéder à votre profil utilisateur en vous rendant sur la page des membres d'un projet ou d'un espace et en choisissant votre nom dans la liste des membres.

3. Sous Jetons d'accès personnels, choisissez Créer.

La page Create PAT s'affiche.

4. Dans Nom du PAT, entrez un nom descriptif pour votre PAT.
5. Dans Date d'expiration, conservez la date par défaut ou cliquez sur l'icône du calendrier pour sélectionner une date personnalisée. La date d'expiration par défaut est de 1 an à compter de la date actuelle.
6. Choisissez Créer.

### Tip

Vous pouvez également créer ce jeton lorsque vous choisissez Clone un référentiel pour un référentiel source.

7. Pour copier le secret PAT, choisissez Copier. Conservez le secret PAT dans un endroit où vous pourrez le récupérer.

### Important

Le secret PAT ne s'affiche qu'une seule fois. Vous ne pouvez pas le récupérer après avoir fermé la fenêtre. Si vous n'avez pas enregistré le secret PAT dans un emplacement sécurisé, vous pouvez en créer un autre.

## Création de PAT (CLI)

Vous pouvez utiliser la CLI pour créer des PAT dans CodeCatalyst.

Pour créer un jeton d'accès personnel (AWS CLI)

1. Sur le terminal ou sur la ligne de commande, exécutez la `create-access-token` commande comme suit.

```
aws codecatalyst create-access-token
```

En cas de succès, la commande renvoie des informations sur le PAT créé, comme dans l'exemple suivant.

```
{
  "secret": "value",
  "name": "marymajor-22222EXAMPLE",
  "expiresTime": "2024-02-04T01:56:04.402000+00:00"
}
```

2.

Vous ne pouvez consulter le secret PAT qu'une seule fois, lorsque vous créez le PAT. Si vous avez égaré un secret PAT ou si vous craignez qu'il ne soit pas stocké de manière sécurisée, vous pouvez en créer un autre.

Vous pouvez consulter les PAT associés à votre compte utilisateur en utilisant le AWS CLI. Vous ne pouvez consulter que les informations relatives au PAT, et non la valeur du secret PAT lui-même.

### Note

Assurez-vous que vous utilisez une version récente du AWS CLI pour travailler avec CodeCatalyst. Les versions antérieures peuvent ne pas contenir les CodeCatalyst commandes. Vous devez configurer votre AWS CLI profil avant de pouvoir l'utiliser avec CodeCatalyst. Pour plus d'informations, consultez [Configuration pour utiliser AWS CLI le CodeCatalyst](#).

## Affichage des PAT

Vous pouvez consulter les PAT dans CodeCatalyst. La liste indique tous les PAT que vous avez associés à votre identité d'utilisateur. Votre PAT est associé à votre profil utilisateur dans tous les espaces et projets de CodeCatalyst. Les PAT expirés ne s'affichent pas car ils sont supprimés après leur expiration.

### Affichage des PAT (console)

Vous pouvez utiliser la console pour afficher les PAT associés à votre identité d'utilisateur dans CodeCatalyst.

Pour consulter vos jetons d'accès personnels (console)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la barre de menu supérieure, choisissez votre badge de profil, puis sélectionnez Mes paramètres. La page CodeCatalyst Mes paramètres s'ouvre.

#### Tip

Vous pouvez également accéder à votre profil utilisateur en vous rendant sur la page des membres d'un projet ou d'un espace et en choisissant votre nom dans la liste des membres.

3. Sous Jetons d'accès personnels, consultez les noms et les dates d'expiration de vos PAT actuels.

### Affichage des PAT (CLI)

Vous pouvez utiliser la CLI pour afficher les PAT associés à votre identité d'utilisateur dans CodeCatalyst.

Pour consulter vos jetons d'accès personnels (AWS CLI)

- Sur le terminal ou sur la ligne de commande, exécutez la `list-access-tokens` commande comme suit.

```
aws codecatalyst list-access-tokens
```

En cas de succès, la commande renvoie des informations sur les PAT associés à votre compte utilisateur, comme dans l'exemple suivant.

```
{
  "items": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEEaaaaa",
      "name": "marymajor-22222EXAMPLE",
      "expiresTime": "2024-02-04T01:56:04.402000+00:00"
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEebbbb",
      "name": "marymajor-11111EXAMPLE",
      "expiresTime": "2023-03-12T01:58:40.694000+00:00"
    }
  ]
}
```

## Supprimer des PAT

Vous pouvez supprimer les PAT associés à votre identité d'utilisateur dans CodeCatalyst.

### Supprimer des PAT (console)

Vous pouvez utiliser la console pour supprimer des PAT. CodeCatalyst

Pour supprimer un jeton d'accès personnel (console)

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans la barre de menu supérieure, choisissez votre badge de profil, puis sélectionnez Mes paramètres. La page CodeCatalyst Mes paramètres s'ouvre.

#### Tip

Vous pouvez également accéder à votre profil utilisateur en vous rendant sur la page des membres d'un projet ou d'un espace et en choisissant votre nom dans la liste des membres.

3. Sous Jetons d'accès personnels, choisissez le sélecteur à côté du PAT que vous souhaitez supprimer, puis choisissez Supprimer.

Sur le PAT de suppression : ? <name> page, pour confirmer la suppression, tapez supprimer dans le champ de texte. Sélectionnez Delete (Supprimer).

## Supprimer des PAT (CLI)

Vous pouvez supprimer un PAT associé à votre identité d'utilisateur en utilisant le AWS CLI. Pour ce faire, vous devez fournir l'ID du PAT, que vous pouvez consulter à l'aide de la delete-access-token commande.

### Note

Assurez-vous que vous utilisez une version récente du AWS CLI pour travailler avec CodeCatalyst. Les versions antérieures peuvent ne pas contenir les CodeCatalyst commandes. Pour plus d'informations sur l'utilisation du AWS CLI with CodeCatalyst, consultez [Configuration pour utiliser AWS CLI le CodeCatalyst](#).

### Pour supprimer un jeton d'accès personnel (AWS CLI)

- Sur le terminal ou sur la ligne de commande, exécutez la delete-access-token commande en fournissant l'ID du PAT que vous souhaitez supprimer. Par exemple, exécutez la commande suivante pour supprimer un PAT dont l'ID est **123EXAMPLE**.

```
aws codecatalyst delete-access-token --id a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbb
```

En cas de succès, cette commande ne renvoie aucune réponse.

## Authentification multifactorielle (MFA) sur Amazon CodeCatalyst

Que vous ayez créé votre profil AWS Builder ID pour un usage personnel ou professionnel, nous vous recommandons de configurer l'authentification multifactorielle (MFA) comme couche de sécurité supplémentaire. Nous vous recommandons particulièrement de configurer la MFA si vous êtes membre d'un espace et que vous collaborez avec d'autres personnes sur des projets. Étant donné

que plusieurs personnes peuvent avoir accès à un projet, les risques de violations de sécurité augmentent.

Lorsque vous activez le MFA, vous devez vous connecter à Amazon à l'aide de votre adresse e-mail et CodeCatalyst de votre mot de passe. Cette partie de la connexion est le premier facteur, lorsque vous utilisez quelque chose que vous connaissez. Vous vous connectez ensuite à l'aide d'un code ou d'une clé de sécurité. C'est le deuxième facteur, qui est quelque chose que vous avez. Le deuxième facteur peut être un code d'authentification généré par votre appareil mobile ou en appuyant sur une touche de sécurité connectée à votre ordinateur. Pris ensemble, ces multiples facteurs renforcent la sécurité en empêchant les accès non autorisés.

## Comment enregistrer un appareil pour une utilisation avec l'authentification multifactorielle

Utilisez la procédure suivante sur Mon profil > Authentification multifactorielle pour enregistrer votre nouvel appareil pour l'authentification multifactorielle (MFA).

### Note


Nous vous recommandons de télécharger d'abord l'application d'authentification appropriée sur votre appareil avant de commencer les étapes de cette procédure. Pour obtenir la liste des applications que vous pouvez utiliser pour les appareils MFA, consultez. [Applications d'authentification](#)

Pour enregistrer votre appareil afin de l'utiliser avec le MFA

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. En haut à droite, cliquez sur la flèche à côté de l'icône portant votre première initiale, puis sélectionnez Profil utilisateur. La page CodeCatalyst de profil s'ouvre.
3. Sur la page de profil, choisissez Gérer le profil et la sécurité. La page de profil AWS Builder ID s'ouvre.
4. Sur le côté gauche de la page, sélectionnez Sécurité.
5. Sur la page Authentification multifactorielle, choisissez Enregistrer un appareil.
6. Sur la page Enregistrer un appareil MFA, choisissez l'un des types de périphériques MFA suivants et suivez les instructions :



- Clé de sécurité ou authentificateur intégré
  1. Sur la page Enregistrer la clé de sécurité de votre utilisateur, suivez les instructions fournies par votre navigateur ou votre plateforme.

 Note

Cette expérience varie en fonction de votre système d'exploitation et de votre navigateur. Suivez donc les instructions affichées par votre navigateur ou votre plateforme. Une fois que votre appareil aura été enregistré avec succès, vous aurez la possibilité d'associer un nom d'affichage convivial à votre appareil nouvellement inscrit. Si vous souhaitez modifier cela, choisissez Renommer, entrez le nouveau nom, puis cliquez sur Enregistrer.

- Application Authenticator
  1. Sur la page Configurer l'application d'authentification, vous remarquerez peut-être des informations de configuration pour le nouveau dispositif MFA, notamment un code QR graphique. Le graphique est une représentation de la clé secrète qui peut être saisie manuellement sur les appareils qui ne prennent pas en charge les codes QR.
  2. À l'aide du périphérique MFA physique, procédez comme suit :
    - a. Ouvrez une application d'authentification MFA compatible. Pour obtenir la liste des applications testées que vous pouvez utiliser avec les appareils MFA, consultez [Applications d'authentification testées](#). Si l'application MFA prend en charge plusieurs appareils, choisissez l'option permettant de créer un nouvel appareil MFA.
    - b. Déterminez si l'application MFA prend en charge les codes QR, puis effectuez l'une des opérations suivantes sur la page Configurer l'application d'authentification :
      - i. Choisissez Afficher le code QR, puis utilisez l'application pour scanner le code QR. Par exemple, vous pouvez choisir l'icône de l'appareil photo ou une option similaire à Scanner le code. Utilisez ensuite l'appareil photo de l'appareil pour scanner le code.
      - ii. Choisissez Afficher la clé secrète, puis entrez cette clé secrète dans votre application MFA.

**⚠ Important**

Lorsque vous configurez un dispositif MFA pour AWS Builder ID, enregistrez une copie du code QR ou de la clé secrète dans un endroit sûr. Cela peut être utile si vous perdez le téléphone ou si vous devez réinstaller l'application d'authentification MFA. Si l'une de ces situations se produit, vous pouvez rapidement reconfigurer l'application pour qu'elle utilise la même configuration MFA.

3. Sur la page Configurer l'application d'authentification, sous Code d'authentification, entrez le mot de passe à usage unique qui apparaît actuellement sur le dispositif MFA physique.

**⚠ Important**

Envoyez votre demande immédiatement après avoir généré le code. Si vous générez le code puis attendez trop longtemps pour envoyer la demande, le dispositif MFA est correctement associé à votre profil AWS Builder ID, mais le dispositif MFA n'est pas synchronisé. En effet, les TOTP (Time-based One-Time Passwords ou mots de passe à usage unique à durée limitée) expirent après une courte période. Dans ce cas, vous pouvez resynchroniser le dispositif.

4. Choisissez Assign MFA (Affecter le MFA). Le dispositif MFA peut désormais commencer à générer des mots de passe à usage unique et est maintenant prêt à être utilisé.

## Applications d'authentification

Les applications d'authentification sont des authentificateurs tiers basés sur un mot de passe à usage unique (OTP). Les utilisateurs peuvent utiliser une application d'authentification installée sur leur appareil mobile ou leur tablette en tant que dispositif MFA autorisé. L'application d'authentification tierce doit être conforme à la RFC 6238, qui est un algorithme TOTP (mot de passe à usage unique basé sur le temps) basé sur des normes capable de générer des codes d'authentification à six chiffres.

Lorsqu'ils sont invités à saisir le MFA, les utilisateurs doivent saisir un code valide provenant de leur application d'authentification dans la zone de saisie présentée. Chaque dispositif MFA attribué à un utilisateur doit être unique. Deux applications d'authentification peuvent être enregistrées pour un utilisateur donné.

## Applications d'authentification testées

Bien que toute application compatible TOTP fonctionne avec le MFA d'IAM Identity Center, le tableau suivant répertorie les applications d'authentification tierces les plus connues parmi lesquelles choisir.

| Système d'exploitation | Application d'authentification testée   |
|------------------------|---|
| Android                | <a href="#">Authy</a> , <a href="#">Duo Mobile</a> , <a href="#">LastPass authentificateur</a> , <a href="#">authentificateur Microsoft</a> , <a href="#">authentificateur Google</a> |
| iOS                    | <a href="#">Authy</a> , <a href="#">Duo Mobile</a> , <a href="#">LastPass authentificateur</a> , <a href="#">authentificateur Microsoft</a> , <a href="#">authentificateur Google</a> |

## Modification de vos appareils MFA

Après avoir enregistré un appareil MFA, vous pouvez modifier son nom ou le supprimer. Nous vous recommandons de toujours activer au moins un appareil MFA pour un niveau de sécurité supplémentaire. Vous pouvez enregistrer jusqu'à cinq appareils. Pour savoir comment en ajouter d'autres, consultez [Comment enregistrer un appareil pour une utilisation avec l'authentification multifactorielle](#).

### Modification du nom d'un appareil MFA

Pour renommer votre appareil MFA

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. En haut à droite, cliquez sur la flèche à côté de l'icône portant votre première initiale, puis sélectionnez Profil utilisateur. La page CodeCatalyst de profil s'ouvre.
3. Sur la page de profil, choisissez Gérer le profil et la sécurité. La page de profil AWS Builder ID s'ouvre.
4. Choisissez Authentification multifactorielle sur le côté gauche de la page. Vous verrez que Rename est grisé lorsque vous arrivez sur la page.
5. Sélectionnez le périphérique MFA que vous souhaitez modifier. Choisissez Rename (Renommer). Puis un modal apparaît.

6. Dans l'invite qui s'affiche, entrez le nouveau nom dans le champ Nom du périphérique MFA, puis choisissez Renommer. L'appareil renommé apparaît sous Dispositifs d'authentification multifactorielle (MFA).

## Supprimer un appareil MFA

Pour supprimer un appareil MFA

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. En haut à droite, cliquez sur la flèche à côté de l'icône portant votre première initiale, puis sélectionnez Profil utilisateur. La page CodeCatalyst de profil s'ouvre.
3. Sur la page de profil, choisissez Gérer le profil et la sécurité. La page de profil AWS Builder ID s'ouvre.
4. Choisissez Authentification multifactorielle sur le côté gauche de la page. Vous verrez que Supprimer est grisé lorsque vous arrivez sur la page.
5. Sélectionnez le périphérique MFA que vous souhaitez modifier. Sélectionnez Delete (Supprimer). Un modal apparaît indiquant Supprimer le périphérique MFA ? . Suivez les instructions pour supprimer votre appareil.
6. Sélectionnez Delete (Supprimer). L'appareil supprimé n'apparaît plus sous Appareils d'authentification multifactoriels (MFA).

## Sécurité sur Amazon CodeCatalyst

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des espaces 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 CodeCatalyst, voir [AWS Services concernés par programme de conformitéAWS](#) .

- Sécurité dans le cloud — Votre responsabilité est déterminée par les AWS services que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre entreprise et la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation d'Amazon CodeCatalyst. Il vous explique comment procéder à la configuration CodeCatalyst 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 CodeCatalyst ressources.

## Table des matières

- [Protection des données sur Amazon CodeCatalyst](#)
- [Identity and Access Management et Amazon CodeCatalyst](#)
- [Validation de conformité pour Amazon CodeCatalyst](#)
- [Résilience chez Amazon CodeCatalyst](#)
- [Sécurité de l'infrastructure sur Amazon CodeCatalyst](#)
- [Analyse de configuration et de vulnérabilité sur Amazon CodeCatalyst](#)
- [Vos données et votre confidentialité sur Amazon CodeCatalyst](#)
- [Bonnes pratiques pour les actions de flux de travail sur Amazon CodeCatalyst](#)
- [CodeCatalyst modèle de confiance](#)

## Protection des données sur Amazon CodeCatalyst

La sécurité et la conformité sont une responsabilité partagée entre Amazon CodeCatalyst et le client, tout comme le AWS [modèle](#) de s'applique à votre utilisation des AWS ressources utilisées dans un flux de travail. Comme décrit dans ce modèle, CodeCatalyst est responsable de la protection de l'infrastructure globale du service. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Ce modèle de responsabilité partagée s'applique à la protection des données dans CodeCatalyst.

Pour des raisons de protection des données, nous vous recommandons de protéger les informations d'identification de votre compte et de configurer l'authentification multifactorielle lors de la connexion. Pour en savoir plus, consultez [Authentification multifactorielle \(MFA\) sur Amazon CodeCatalyst](#).

Ne saisissez pas d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou dans des champs de forme libre tels que le champ Nom. Cela inclut les noms des ressources et tous les autres identifiants que vous entrez CodeCatalyst en plus de ceux qui sont connectés Comptes AWS. Par exemple, ne saisissez pas d'informations confidentielles ou sensibles dans les noms des espaces, des projets ou des flottes de déploiement. Toutes les données que vous entrez dans les balises, les noms ou les champs de forme libre utilisés pour les noms peuvent être utilisées pour les journaux de facturation ou de diagnostic ou peuvent être incluses dans les chemins d'URL. Cela s'applique à l'utilisation de la console, de l'API AWS CLI, de l' CodeCatalyst Action Development Kit ou de tout autre AWS SDK.

Si vous fournissez une URL vers un serveur externe, nous vous recommandons vivement de ne pas inclure d'informations d'identification de sécurité dans l'URL pour valider votre demande auprès de ce serveur.

CodeCatalyst les référentiels sources sont automatiquement chiffrés au repos. Aucune action du client n'est requise. CodeCatalyst chiffre également les données du référentiel en transit à l'aide du protocole HTTPS.

CodeCatalyst prend en charge le MFA. Pour en savoir plus, consultez [Authentification multifactorielle \(MFA\) sur Amazon CodeCatalyst](#).

## Chiffrement des données

CodeCatalyst stocke et transfère les données en toute sécurité au sein du service. Toutes les données sont cryptées en transit et au repos. Toutes les données créées ou stockées par le service, y compris les métadonnées du service, sont stockées nativement dans le service et cryptées.

### Note

Bien que les informations relatives aux problèmes soient stockées en toute sécurité au sein du service, les informations relatives aux problèmes en suspens sont également stockées dans le cache local du navigateur où vous avez consulté les forums de discussion, les arriérés et les problèmes individuels. Pour une sécurité optimale, veuillez à vider le cache de votre navigateur pour supprimer ces informations.

Si vous utilisez des ressources liées à CodeCatalyst, telles qu'une connexion de compte à un référentiel lié Compte AWS ou un référentiel lié GitHub, les données en transit depuis CodeCatalyst cette ressource liée sont cryptées, mais le traitement des données dans cette ressource liée est

géré par ce service lié. Pour plus d'informations, consultez la documentation du service lié et [Bonnes pratiques pour les actions de flux de travail sur Amazon CodeCatalyst](#).

## Gestion des clés

CodeCatalyst ne prend pas en charge la gestion des clés.

## Confidentialité du trafic inter-réseaux

Lorsque vous créez un espace dans CodeCatalyst, vous choisissez l'Région AWS endroit où les données et les ressources seront stockées pour cet espace. Les données et les métadonnées du projet ne laissent jamais cela de côté Région AWS. Toutefois, pour faciliter la navigation au sein de la partition CodeCatalyst, un ensemble limité de métadonnées relatives à l'espace, au projet et à l'utilisateur est répliqué sur l'ensemble Régions AWS de la [partition](#). Il ne sera pas répliqué en Régions AWS dehors de cette partition. Par exemple, si vous choisissez USA West (Oregon) comme espace Région AWS lorsque vous créez votre espace, vos données ne seront pas répliquées dans les régions de Chine ou AWS GovCloud (US). Pour plus d'informations, consultez [la section Gestion Régions AWS](#), [infrastructure AWS globale](#) et [points de terminaison AWS de service](#).

Les données répliquées à Régions AWS l'intérieur d'une partition incluent :

- Une valeur de hachage cryptée qui représente le nom de l'espace afin de garantir l'unicité des noms d'espace. Cette valeur n'est pas lisible par l'homme et n'expose pas les noms réels des espaces
- L'identifiant unique de l'espace
- Métadonnées pour l'espace qui facilitent la navigation dans les espaces
- L'Région AWS endroit où se trouve l'espace
- Les identifiants uniques de tous les projets de l'espace
- L'ID de rôle qui indique le rôle d'un utilisateur dans un espace ou un projet
- Lors de l'inscription CodeCatalyst, les données et métadonnées relatives au processus d'inscription, notamment :
  - L'identifiant unique du ID de constructeur AWS
  - Le nom d'affichage de l'utilisateur dans son ID de constructeur AWS
  - L'alias de l'utilisateur dans son ID de constructeur AWS
  - L'adresse e-mail utilisée lorsque l'utilisateur s'est inscrit à son ID de constructeur AWS
  - L'état d'avancement du processus d'inscription

- Si vous créez un espace dans le cadre du processus d'inscription, l'Compte AWSidentifiant utilisé comme compte de facturation pour l'espace

Les noms d'espaces sont uniques d'un bout à l'autre CodeCatalyst. Veillez à ne pas inclure de données sensibles dans le nom de l'espace.

Lorsque vous travaillez avec des ressources liées et des comptes connectés, tels qu'une connexion à un référentiel Compte AWS ou à un GitHub référentiel, nous vous recommandons de configurer vos emplacements source et de destination avec le niveau de sécurité le plus élevé que chacun d'entre eux prend en charge. CodeCatalyst sécurise la connexion entre Comptes AWS Régions AWS, et les zones de disponibilité en utilisant le protocole TLS (Transport Layer Security) 1.2.

## Identity and Access Management et Amazon CodeCatalyst

Sur Amazon CodeCatalyst, vous créez et utilisez un AWS Builder ID pour vous connecter et accéder à vos espaces et à vos projets. Un AWS Builder ID n'est pas une identité dans AWS Identity and Access Management (IAM) et n'existe pas dans un Compte AWS. Cependant, il CodeCatalyst s'intègre à IAM lors de la vérification d'un espace à des fins de facturation, et lorsqu'il est connecté Compte AWS à un espace pour y créer et utiliser des Compte AWS ressources.

AWS Identity and Access Management (IAM) est un Service AWS qui aide un administrateur à contrôler en toute sécurité l'accès aux ressources AWS. Des administrateurs IAM contrôlent les personnes qui s'authentifient (sont connectées) et sont autorisées (disposent d'autorisations) à utiliser des ressources . IAM est un Service AWS que vous pouvez utiliser sans frais supplémentaires.

Lorsque vous créez un espace sur Amazon CodeCatalyst, vous devez en connecter un Compte AWS comme compte de facturation pour votre espace. Vous devez disposer des autorisations d'administrateur Compte AWS pour vérifier l' CodeCatalyst espace, ou vous devez avoir l'autorisation. Vous avez également la possibilité d'ajouter un rôle IAM à votre espace, qui CodeCatalyst peut être utilisé pour créer et accéder aux ressources de l'espace connectéCompte AWS. C'est ce qu'on appelle un [rôle de service](#). Vous pouvez choisir de créer des connexions avec plusieurs comptes Compte AWS et de créer des rôles de service pour CodeCatalyst chacun de ces comptes.

### Note

La facturation CodeCatalyst a lieu sur le compte Compte AWS désigné comme étant le compte de facturation. Toutefois, si vous créez un rôle de CodeCatalyst service dans



ce rôle Compte AWS ou dans tout autre service connectéCompte AWS, les ressources créées et utilisées par le rôle de CodeCatalyst service seront facturées dans ce rôle connectéCompte AWS. Pour plus d'informations, consultez [la section Gestion de la facturation](#) dans le manuel Amazon CodeCatalyst Administrator Guide.

## Rubriques

- [Politiques basées sur l'identité dans IAM](#)
- [Actions politiques dans IAM](#)
- [Ressources relatives aux politiques dans IAM](#)
- [Clés de conditions de politique dans IAM](#)
- [Exemples de politiques basées sur l'identité pour les connexions CodeCatalyst](#)
- [Utilisation de balises pour contrôler l'accès aux ressources de connexion au compte](#)
- [CodeCatalyst référence aux autorisations](#)
- [Utilisation des rôles liés aux services pour CodeCatalyst](#)
- [AWSpolitiques gérées pour Amazon CodeCatalyst](#)
- [Rôles IAM pour l' CodeCatalyst accès d'Amazon aux ressources AWS](#)

## Politiques basées sur l'identité dans IAM

Les politiques basées sur l'identité sont des documents de politique d'autorisation JSON que vous pouvez joindre à une identité. Cette identité peut être un utilisateur, un groupe d'utilisateurs ou un rôle. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez [Création de politiques IAM](#) dans le Guide de l'utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité car celle-ci s'applique à l'utilisateur ou au rôle auquel elle est attachée. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez [Références des éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.

## Exemples de politiques basées sur l'identité pour CodeCatalyst

Pour consulter des exemples de politiques CodeCatalyst basées sur l'identité, consultez. [Exemples de politiques basées sur l'identité pour les connexions CodeCatalyst](#)

## Actions politiques dans IAM

Les administrateurs peuvent utiliser les politiques JSON AWS pour spécifier qui a accès à quoi. C'est-à-dire quel principal peut effectuer quelles actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de politique possèdent généralement le même nom que l'opération d'API AWS associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
  "prefix:action1",
  "prefix:action2"
]
```

## Ressources relatives aux politiques dans IAM

Les administrateurs peuvent utiliser les politiques JSON AWS pour spécifier qui a accès à quoi. C'est-à-dire quel principal peut effectuer quelles actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets pour lesquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son [Amazon Resource Name \(ARN\)](#). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (\*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*" 
```

## Clés de conditions de politique dans IAM

Les administrateurs peuvent utiliser les politiques JSON AWS pour spécifier qui a accès à quoi. C'est-à-dire quel principal peut effectuer quelles actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des [opérateurs de condition](#), tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération AND logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une opération OR logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Pour plus d'informations, consultez [Éléments des politiques IAM : variables et balises](#) dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques à un service. Pour afficher toutes les clés de condition globales AWS, consultez [Clés de contexte de condition globale AWS](#) dans le Guide de l'utilisateur IAM.

## Exemples de politiques basées sur l'identité pour les connexions CodeCatalyst

Dans CodeCatalyst, Comptes AWS sont nécessaires pour gérer la facturation d'un espace et pour accéder aux ressources dans les flux de travail du projet. Une connexion au compte est utilisée pour autoriser l'ajout Comptes AWS à un espace. Des politiques basées sur l'identité sont utilisées dans le connecté. Comptes AWS

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou modifier les ressources CodeCatalyst. Ils ne peuvent pas non plus exécuter des tâches à l'aide de la AWS Management Console, de l'AWS Command Line Interface (AWS CLI) ou de l'API AWS. Un administrateur IAM doit créer des politiques IAM autorisant les utilisateurs et les rôles à exécuter des actions sur les ressources dont ils ont besoin. L'administrateur doit ensuite attacher ces stratégies aux utilisateurs qui en ont besoin.

Les exemples de politiques IAM suivants accordent des autorisations pour les actions liées aux connexions aux comptes. Utilisez-les pour limiter l'accès à des comptes connectés à CodeCatalyst.

Exemple 1 : Autoriser un utilisateur à accepter des demandes de connexion en une seule fois Région AWS

La politique d'autorisation suivante permet uniquement aux utilisateurs de consulter et d'accepter les demandes de connexion entre CodeCatalyst et Comptes AWS. En outre, la politique utilise une condition pour autoriser uniquement les actions dans la région us-west-2 et non depuis une autre région. Régions AWS Pour consulter et approuver la demande, l'utilisateur se connecte AWS Management Console avec le même compte que celui spécifié dans la demande.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecatalyst:AcceptConnection",
        "codecatalyst:GetPendingConnection"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": "us-west-2"
        }
      }
    }
  ]
}
```

Exemple 2 : Autoriser la gestion des connexions dans la console pour un Région AWS

La politique d'autorisation suivante permet aux utilisateurs de gérer les connexions entre CodeCatalyst et Comptes AWS dans une même région. La politique utilise une condition pour autoriser uniquement les actions dans la région us-west-2 et non depuis une autre région. Régions AWS Après avoir créé une connexion, vous pouvez créer le CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle en choisissant l'option dans le AWS Management Console. Dans l'exemple de politique, la condition de l'iam:PassRole action inclut les principes de service pour CodeCatalyst. Seuls les rôles dotés de cet accès seront créés dans le AWS Management Console.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecatalyst:*"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestedRegion": "us-west-2"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:CreateRole",
        "iam:CreatePolicy",
        "iam:AttachRolePolicy",
        "iam:ListRoles"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "codecatalyst.amazonaws.com",
            "codecatalyst-runner.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

### Exemple 3 : Refuser la gestion des connexions

La politique d'autorisation suivante refuse aux utilisateurs toute possibilité de gérer les connexions entre CodeCatalyst et Comptes AWS.

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

### Utilisation de balises pour contrôler l'accès aux ressources de connexion au compte

Les balises peuvent être attachées à la ressource ou transmises dans la demande aux services qui prennent en charge le balisage. Les ressources des politiques peuvent comporter des balises, et certaines actions des politiques peuvent inclure des balises. Les clés de condition de balisage incluent les clés `aws:RequestTag` de `aws:ResourceTag` condition et. Lorsque vous créez une stratégie IAM, vous pouvez utiliser des clés de condition de balise pour contrôler les éléments suivants :

- Quels utilisateurs peuvent effectuer des actions sur une ressource de connexion, en fonction des balises qu'elle possède déjà.
- quelles balises peuvent être transmises dans une demande d'action ;
- si des clés de balise spécifiques peuvent être utilisées dans une demande.

Les exemples suivants montrent comment spécifier des conditions de balise dans les politiques relatives aux connexions de CodeCatalyst compte pour les utilisateurs. Pour de plus amples informations sur l'utilisation des clés de condition, consultez la section [Clés de conditions de politique dans IAM](#).

### Exemple 1 : Autoriser les actions en fonction des balises contenues dans la demande

La politique suivante autorise les utilisateurs à approuver les connexions aux comptes.

Pour ce faire, elle autorise les actions `AcceptConnection` et `TagResource` si la demande spécifie une balise nommée `Project` avec la valeur `ProjectA`. (La clé de condition `aws:RequestTag` est utilisée pour contrôler les balises qui peuvent être transmises dans une demande IAM.) La condition `aws:TagKeys` garantit que la clé de balise est sensible à la casse.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codecatalyst:AcceptConnection",
        "codecatalyst:TagResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/Project": "ProjectA"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": ["Project"]
        }
      }
    }
  ]
}
```

## Exemple 2 : Autoriser les actions en fonction des balises de ressources

La politique suivante autorise les utilisateurs à effectuer des actions sur les ressources de connexion au compte et à obtenir des informations sur celles-ci.

Pour ce faire, il autorise des actions spécifiques si la connexion possède une balise nommée `Project` avec la valeur `ProjectA`. (La clé de condition `aws:ResourceTag` est utilisée pour contrôler les balises qui peuvent être transmises dans une demande IAM.)

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

```

    "codecatalyst:GetConnection",
    "codecatalyst>DeleteConnection",
    "codecatalyst:AssociateIamRoleToConnection",
    "codecatalyst:DisassociateIamRoleFromConnection",
    "codecatalyst:ListIamRolesForConnection",
    "codecatalyst:PutBillingAuthorization"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceTag/Project": "ProjectA"
    }
  }
}
]
}

```

## CodeCatalyst référence aux autorisations

Cette section fournit une référence d'autorisations pour les actions utilisées avec la ressource de connexion au compte à Comptes AWS laquelle vous êtes connecté CodeCatalyst. La section suivante décrit les actions relatives aux autorisations uniquement liées à la connexion de comptes.

### Autorisations requises pour les connexions aux comptes

Les autorisations suivantes sont requises pour utiliser les connexions aux comptes.

| CodeCatalyst autorisations pour les connexions aux comptes | Autorisations nécessaires  | Ressources  |
|--|--|---|
| AcceptConnection   | Nécessaire pour accepter une demande de connexion de ce compte à un CodeCatalyst espace. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API. | Prend en charge uniquement un caractère générique (*) dans l'élément Resource de stratégie. |
| AssociateIamRoleToConnection                               | Nécessaire pour associer un rôle IAM à une connexion   | arn:aws:codecatalyst:region: <i>account_ID</i>  |



| CodeCatalyst autorisations pour les connexions aux comptes | Autorisations nécessaires  | Ressources   |
|--|--|--|
|  | à un compte. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.   | <i>D</i> :/connections/ <i>connection_ID</i>                                       |
| DeleteConnection   | Nécessaire pour supprimer une connexion au compte. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.                                     | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| DisassociateIamRoleFromConnection                          | Nécessaire pour dissocier un rôle IAM d'une connexion à un compte. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.                     | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| GetBillingAuthorization                                    | Obligatoire pour décrire l'autorisation de facturation pour une connexion à un compte. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API. | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |
| GetConnection  | Nécessaire pour établir une connexion à un compte. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.                                     | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i> |

| CodeCatalyst autorisations pour les connexions aux comptes | Autorisations nécessaires   | Ressources  |
|--|---|---|
| GetPendingConnection                                       | Nécessaire pour obtenir une demande en attente pour connecter ce compte à un CodeCatalyst espace. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API. | Prend en charge uniquement un caractère générique (*) dans l'élément Resource de stratégie. |
| ListConnections  | Obligatoire pour répertorier les connexions aux comptes qui ne sont pas en attente. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.               | Prend en charge uniquement un caractère générique (*) dans l'élément Resource de stratégie. |
| ListIamRolesForConnection                                  | Obligatoire pour répertorier les rôles IAM associés à une connexion à un compte. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.                  | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>          |
| ListTagsForResource  | Obligatoire pour répertorier les tags associés à une connexion à un compte. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.                       | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>          |

| CodeCatalyst autorisations pour les connexions aux comptes | Autorisations nécessaires  | Ressources  |
|--|--|---|
| PutBillingAuthorization                                    | Nécessaire pour créer ou mettre à jour l'autorisation de facturation pour une connexion à un compte. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API. | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>          |
| RejectConnection   | Nécessaire pour rejeter une demande de connexion de ce compte à un CodeCatalyst espace. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.              | Prend en charge uniquement un caractère générique (*) dans l'élément Resource de stratégie. |
| TagResource  | Nécessaire pour créer ou modifier des balises associées à une connexion à un compte. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.                 | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>          |
| UntagResource  | Nécessaire pour supprimer les tags associés à une connexion à un compte. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.                             | arn:aws:codecatalyst:region: <i>account_ID</i> :/connections/ <i>connection_ID</i>          |

## Autorisations requises pour les applications IAM Identity Center

Les autorisations suivantes sont requises pour utiliser les applications IAM Identity Center.

| CodeCatalyst autorisations pour les applications IAM Identity Center | Autorisations nécessaires  | Ressources   |
|--|--|--|
| AssociateIdentityCenterApplicationToSpace                            | Nécessaire pour associer une application IAM Identity Center à un CodeCatalyst espace. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.                           | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| AssociateIdentityToIdentityCenterApplication                         | Nécessaire pour associer une identité à une application IAM Identity Center pour un CodeCatalyst espace. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.         | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| BatchAssociateIdentitiesToIdentityCenterApplication                  | Nécessaire pour associer plusieurs identités à une application IAM Identity Center pour un CodeCatalyst espace. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.  | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| BatchDisassociateIdentitiesFromIdentityCenterApplication             | Nécessaire pour dissocier plusieurs identités d'une application IAM Identity Center pour un CodeCatalyst espace. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API. | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| CreateIdentityCenterApplication                                      | Nécessaire pour créer une application IAM Identity   | arn:aws:codecatalyst:region: <i>account_ID</i>   |

| CodeCatalyst autorisations pour les applications IAM Identity Center | Autorisations nécessaires  | Ressources   |
|--|--|--|
|  | Center. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.  | <i>D</i> :/identity-center-applications/ <i>identity-center-application_ID</i>                                       |
| CreateSpaceAdminRoleAssignment                                       | Nécessaire pour créer une attribution de rôle d'administrateur pour un CodeCatalyst espace donné et une application IAM Identity Center. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API. | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| DeleteIdentityCenterApplication                                      | Nécessaire pour supprimer une application IAM Identity Center. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.   | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| DisassociateIdentityCenterApplicationFromSpace                       | Nécessaire pour dissocier une application IAM Identity Center d'un CodeCatalyst espace. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.  | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |

| CodeCatalyst autorisations pour les applications IAM Identity Center | Autorisations nécessaires   | Ressources   |
|--|---|--|
| DisassociateIdentityFromIdentityCenterApplication                    | Nécessaire pour dissocier une identité d'une application IAM Identity Center pour un CodeCatalyst espace. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API. | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| GetIdentityCenterApplication   | Nécessaire pour obtenir des informations sur une application IAM Identity Center. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.                         | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| ListIdentityCenterApplications                                       | Nécessaire pour consulter la liste des applications IAM Identity Center présentes dans le compte. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.         | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| ListIdentityCenterApplicationsForSpace                               | Nécessaire pour afficher la liste des applications IAM Identity Center par CodeCatalyst espace. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.           | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |

| CodeCatalyst autorisations pour les applications IAM Identity Center | Autorisations nécessaires   | Ressources   |
|--|---|--|
| ListSpacesForIdentityCenterApplication                               | Nécessaire pour afficher la liste des CodeCatalyst espaces par application IAM Identity Center. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.           | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| SynchronizIdentityCenterApplication                                  | Nécessaire pour synchroniser une application IAM Identity Center avec le magasin d'identités sous-jacent. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API. | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |
| UpdateIdentityCenterApplication                                      | Nécessaire pour mettre à jour une application IAM Identity Center. Il s'agit d'une autorisation de stratégie IAM uniquement, pas d'une action d'API.  | arn:aws:codecatalyst:region: <i>account_ID</i> :/identity-center-applications/ <i>identity-center-application_ID</i> |

## Utilisation des rôles liés aux services pour CodeCatalyst

Amazon CodeCatalyst utilise des rôles AWS Identity and Access Management liés à un [service](#) (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié à CodeCatalyst. Les rôles liés au service sont prédéfinis par CodeCatalyst et incluent toutes les autorisations dont le service a besoin pour appeler d'autres AWS services en votre nom.

Un rôle lié à un service facilite la configuration CodeCatalyst car vous n'avez pas à ajouter manuellement les autorisations nécessaires. CodeCatalyst définit les autorisations associées à ses rôles liés aux services et, sauf indication contraire, seul CodeCatalyst peut assumer ses rôles. Les

autorisations définies comprennent la politique d'approbation et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous pouvez supprimer un rôle lié à un service uniquement après la suppression préalable de ses ressources connexes. Cela protège vos CodeCatalyst ressources car vous ne pouvez pas supprimer par inadvertance l'autorisation d'accès aux ressources.

Pour plus d'informations sur les autres services prenant en charge les rôles liés à un service, consultez les [AWS services opérationnels avec IAM](#) et recherchez les services présentant la mention Yes (Oui) dans la colonne Service-linked roles (Rôles liés à un service). Sélectionnez un Oui ayant un lien pour consulter la documentation du rôle lié à un service, pour ce service.

### Autorisations de rôle liées à un service pour CodeCatalyst

CodeCatalyst utilise le rôle lié au service nommé `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`— Autorise Amazon à accéder en CodeCatalyst lecture seule aux profils d'instance d'application et aux utilisateurs et groupes de l'annuaire associés en votre nom.

#### Le rôle lié à un service

`AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` approuve les services suivants pour endosser le rôle :

- `codecatalyst.amazonaws.com`

#### La politique d'autorisations de rôle nommée

`AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy` CodeCatalyst permet d'effectuer les actions suivantes sur les ressources spécifiées :

- Action: `View application instance profiles and associated directory users and groups` pour CodeCatalyst spaces that support identity federation and SSO users and groups

Vous devez configurer les autorisations de manière à permettre à vos utilisateurs, groupes ou rôles de créer, modifier ou supprimer un rôle lié à un service. Pour plus d'informations, consultez [Autorisations de rôles liés à un service](#) dans le Guide de l'utilisateur IAM.



## Création d'un rôle lié à un service pour CodeCatalyst

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous créez un espace dans le AWS Management Console, le ou l'AWS CLI, vous CodeCatalyst créez le rôle lié au service pour vous.

### Important

Ce rôle lié à un service peut apparaître dans votre compte si vous avez effectué une action dans un autre service qui utilise les fonctions prises en charge par ce rôle. De plus, si vous utilisiez le CodeCatalyst service avant le 17 novembre 2023, date à laquelle il a commencé à prendre en charge les rôles liés au service, vous avez CodeCatalyst créé le `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` rôle dans votre compte. Pour en savoir plus, consultez la section [Un nouveau rôle est apparu dans mon compte Compte AWS](#).

Si vous supprimez ce rôle lié à un service et que vous avez ensuite besoin de le recréer, vous pouvez utiliser la même procédure pour recréer le rôle dans votre compte. Lorsque vous créez un espace, il CodeCatalyst crée à nouveau le rôle lié au service pour vous.

Vous pouvez également utiliser la console IAM pour créer un rôle lié à un service avec le cas d'utilisation des profils d'instance d'application View et des utilisateurs et groupes d'annuaires associés. Dans l'interface AWS CLI ou l'API AWS, créez un rôle lié à un service avec le nom de service `codecatalyst.amazonaws.com`. Pour de plus amples informations, consultez [Création d'un rôle lié à un service](#) dans le Guide de l'utilisateur IAM. Si vous supprimez ce rôle lié à un service, vous pouvez utiliser ce même processus pour créer le rôle à nouveau.

## Modification d'un rôle lié à un service pour CodeCatalyst

CodeCatalyst ne vous permet pas de modifier le rôle `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` lié au service. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas changer le nom du rôle, car plusieurs entités peuvent faire référence à ce rôle. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour en savoir plus, consultez [Modification d'un rôle lié à un service](#) dans le guide de l'utilisateur IAM.

## Supprimer un rôle lié à un service pour CodeCatalyst

Vous n'avez pas besoin de supprimer manuellement le rôle `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`. Lorsque vous supprimez un espace dans l'APIAWS Management Console, le AWS CLI ou l'AWSAPI, les ressources CodeCatalyst sont nettoyées et le rôle lié au service est supprimé pour vous.

Vous pouvez également utiliser la console IAM, l'AWS CLI ou l'API AWS pour supprimer manuellement le rôle lié à un service. Pour cela, vous devez commencer par nettoyer les ressources de votre rôle lié à un service. Vous pouvez ensuite supprimer ce rôle manuellement.

### Note

Si le CodeCatalyst service utilise le rôle lorsque vous essayez de supprimer les ressources, la suppression risque d'échouer. Si cela se produit, patientez quelques minutes et réessayez.

Pour supprimer CodeCatalyst les ressources utilisées par `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`

- [Supprimez l'espace.](#)

Pour supprimer manuellement le rôle lié à un service à l'aide d'IAM

Utilisez la console IAM, l'AWS CLI ou l'API AWS pour supprimer le rôle lié à un service `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization`. Pour plus d'informations, consultez [Suppression d'un rôle lié à un service](#) dans le Guide de l'utilisateur IAM.

Régions prises en charge pour les rôles CodeCatalyst liés à un service

CodeCatalyst prend en charge l'utilisation de rôles liés au service dans toutes les régions où le service est disponible. Pour plus d'informations, consultez [AWSRégions et points de terminaison.](#)

CodeCatalyst ne prend pas en charge l'utilisation de rôles liés à un service dans toutes les régions où le service est disponible. Vous pouvez utiliser le rôle `AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronization` dans les régions suivantes :

| Nom de la région               | Identité de la région | Support dans CodeCatalyst |
|--------------------------------|-----------------------|---------------------------|
| US East (Virginie du Nord)     | us-east-1             | Non                       |
| USA Est (Ohio)                 | us-east-2             | Non                       |
| USA Ouest (Californie du Nord) | us-west-1             | Non                       |
| USA Ouest (Oregon)             | us-west-2             | Oui                       |
| Afrique (Le Cap)               | af-south-1            | Non                       |
| Asie-Pacifique (Hong Kong)     | ap-east-1             | Non                       |
| Asie-Pacifique (Jakarta)       | ap-southeast-3        | Non                       |
| Asie-Pacifique (Mumbai)        | ap-south-1            | Non                       |
| Asie-Pacifique (Osaka)         | ap-northeast-3        | Non                       |
| Asie-Pacifique (Séoul)         | ap-northeast-2        | Non                       |
| Asie-Pacifique (Singapour)     | ap-southeast-1        | Non                       |
| Asie-Pacifique (Sydney)        | ap-southeast-2        | Non                       |
| Asie-Pacifique (Tokyo)         | ap-northeast-1        | Non                       |
| Canada (Centre)                | ca-central-1          | Non                       |
| Europe (Francfort)             | eu-central-1          | Non                       |
| Europe (Irlande)               | eu-west-1             | Oui                       |
| Europe (Londres)               | eu-west-2             | Non                       |
| Europe (Milan)                 | eu-south-1            | Non                       |
| Europe (Paris)                 | eu-west-3             | Non                       |
| Europe (Stockholm)             | eu-north-1            | Non                       |

| Nom de la région            | Identité de la région | Support dans CodeCatalyst |
|-----------------------------|-----------------------|---------------------------|
| Moyen-Orient (Bahreïn)      | me-south-1            | Non                       |
| Moyen-Orient (EAU)          | me-central-1          | Non                       |
| Amérique du Sud (São Paulo) | sa-east-1             | Non                       |
| AWS GovCloud (USA Est)      | us-gov-east-1         | Non                       |
| AWS GovCloud (US-Ouest)     | us-gov-west-1         | Non                       |

## AWSpolitiques gérées pour Amazon CodeCatalyst

Une politique gérée par AWS est une politique autonome créée et administrée par AWS. Les politiques gérées par AWS 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.

Gardez à l'esprit que les politiques gérées par AWS peuvent ne pas accorder les autorisations de moindre privilège pour vos cas d'utilisation spécifiques, car elles sont disponibles pour tous les clients AWS. Nous vous recommandons de réduire encore les autorisations en définissant des [politiques gérées par le client](#) qui sont spécifiques à vos cas d'utilisation.

Vous ne pouvez pas modifier les autorisations définies dans les politiques gérées par AWS. Si AWS met à jour les autorisations définies dans une politique gérée par AWS, la mise à jour affecte toutes les identités de principal (utilisateurs, groupes et rôles) auxquelles la politique est associée. AWS est plus susceptible de mettre à jour une politique gérée par AWS lorsqu'un nouveau Service AWS est lancé ou que de nouvelles opérations API deviennent accessibles pour les services existants.

Pour plus d'informations, consultez la section [Politiques gérées par AWS](#) dans le Guide de l'utilisateur IAM.

## Politique gérée par AWS :AmazonCodeCatalystSupportAccess

Il s'agit d'une politique qui autorise tous les administrateurs de l'espace et les membres de l'espace à utiliser le plan de support Premium Business ou Enterprise associé au compte de facturation de l'espace. Ces autorisations permettent aux administrateurs de l'espace et aux membres d'utiliser le plan d'assistance premium pour les ressources auxquelles ils sont autorisés dans le cadre des politiques d' autorisation CodeCatalyst.

### Détails des autorisations

Cette politique inclut les autorisations suivantes.

- **support**— Accorde des autorisations permettant aux utilisateurs de rechercher, de créer et de résoudre des demandes de AWS support. Accorde également des autorisations pour décrire les communications, les niveaux de gravité, les pièces jointes et les détails du dossier d'assistance associés.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "support:DescribeAttachment",
        "support:DescribeCaseAttributes",
        "support:DescribeCases",
        "support:DescribeCommunications",
        "support:DescribeIssueTypes",
        "support:DescribeServices",
        "support:DescribeSeverityLevels",
        "support:DescribeSupportLevel",
        "support:SearchForCases",
        "support:AddAttachmentsToSet",
        "support:AddCommunicationToCase",
        "support:CreateCase",
        "support:InitiateCallForCase",
```

```

        "support:InitiateChatForCase",
        "support:PutCaseAttributes",
        "support:RateCaseCommunication",
        "support:ResolveCase"
    ],
    "Resource": "*"
}
]
}

```

## Stratégie AWS gérée : AmazonCodeCatalystFullAccess

Il s'agit d'une politique qui accorde des autorisations pour gérer votre CodeCatalyst espace et les comptes connectés sur la page Amazon CodeCatalyst Spaces duAWS Management Console. Cette application est utilisée pour configurer Comptes AWS ceux qui sont connectés à votre espace dans CodeCatalyst.

### Détails des autorisations

Cette politique inclut les autorisations suivantes.

- `codecatalyst`— Accorde des autorisations complètes à la page Amazon CodeCatalyst Spaces dans leAWS Management Console.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeCatalystResourceAccess"
      "Effect": "Allow",
      "Action": [
        "codecatalyst:*",
        "iam:ListRoles"
      ],
      "Resource": "*"
    },
    {

```

```
"Sid": "CodeCatalystAssociateIAMRole"
"Effect": "Allow",
"Action": [
    "iam:PassRole"
],
"Resource": "*",
"Condition": {
    "StringEquals": {
        "iam:PassedToService": [
            "codecatalyst.amazonaws.com",
            "codecatalyst-runner.amazonaws.com"
        ]
    }
}
}
```

### Stratégie AWS gérée : AmazonCodeCatalystReadOnlyAccess

Il s'agit d'une politique qui accorde l'autorisation de consulter et de répertorier les informations relatives aux espaces et aux comptes connectés sur la page Amazon CodeCatalyst Spaces du AWS Management Console. Cette application est utilisée pour configurer Comptes AWS ceux qui sont connectés à votre espace dans CodeCatalyst.

#### Détails des autorisations

Cette politique inclut les autorisations suivantes.

- `codecatalyst`— Accorde des autorisations en lecture seule à la page Amazon CodeCatalyst Spaces dans le AWS Management Console

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

```
        "Action": [
            "codecatalyst:Get*",
            "codecatalyst:List*",
        ],
        "Resource": "*"
    }
]
```

Stratégie AWS gérée :

AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy

Vous ne pouvez pas attacher

AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy ; à vos entités IAM. Cette politique est associée à un rôle lié à un service qui permet d' CodeCatalyst effectuer des actions en votre nom. Pour de plus amples informations, veuillez consulter [Utilisation des rôles liés aux services pour CodeCatalyst](#).

Cette politique permet aux clients de consulter les profils des instances d'application ainsi que les utilisateurs et groupes de répertoires associés lors de la gestion des espaces dans CodeCatalyst. Les clients consulteront ces ressources lorsqu'ils géreront des espaces prenant en charge la fédération d'identités et les utilisateurs et groupes SSO.

### Détails des autorisations

Cette politique inclut les autorisations suivantes.

- `sso`— Accorde des autorisations permettant aux utilisateurs de consulter les profils d'instances d'applications gérés dans IAM Identity Center pour les espaces associés dans CodeCatalyst.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid":
      "AmazonCodeCatalystServiceRoleForIdentityCenterApplicationSynchronizationPolicy",
```



```

"Effect": "Allow",
"Action": [
  "sso:ListInstances",
  "sso:ListApplications",
  "sso:ListApplicationAssignments",
  "sso:DescribeInstance",
  "sso:DescribeApplication"
],
"Resource": "*"
}
]
}

```

## CodeCatalyst mises à jour des politiques AWS gérées

Consultez les détails des mises à jour des politiques AWS gérées CodeCatalyst 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 CodeCatalyst document](#).

| Modification   | Description   | Date             |
|--|---|------------------|
| <a href="#">AmazonCodeCatalyst ServiceRoleForIdentityCenterApplicationSynchronizationPolicy</a> – Nouvelle politique | CodeCatalyst a ajouté la politique.<br><br>Accorde des autorisations permettant aux CodeCatalyst utilisateurs de consulter les profils des instances d'applications ainsi que les utilisateurs et groupes d'annuaires associés. | 17 novembre 2023 |
| <a href="#">AmazonCodeCatalyst SupportAccess</a> – Nouvelle politique  | CodeCatalyst a ajouté la politique.<br><br>Accorde des autorisations permettant aux CodeCatalyst utilisateurs de recherche  | 20 avril 2023    |

| Modification   | Description  | Date          |
|--|--|---------------|
|  | r, de créer et de résoudre des demandes d'assistance, ainsi que de consulter les communications et les détails connexes. |               |
| <a href="#">AmazonCodeCatalyst FullAccess</a> – Nouvelle politique     | CodeCatalyst a ajouté la politique.<br>Accorde un accès complet à CodeCatalyst.  | 20 avril 2023 |
| <a href="#">AmazonCodeCatalyst ReadOnlyAccess</a> – Nouvelle politique | CodeCatalyst a ajouté la politique.<br>Accorde un accès en lecture seule à CodeCatalyst                                  | 20 avril 2023 |
| CodeCatalyst a commencé à suivre les modifications                     | CodeCatalyst a commencé à suivre les modifications apportées AWS à ses politiques gérées.                                | 20 avril 2023 |

## Rôles IAM pour l'accès de CodeCatalyst à Amazon aux ressources AWS

CodeCatalyst peut accéder aux ressources AWS en vous connectant un compte AWS à un espace CodeCatalyst. Vous pouvez ensuite créer les rôles de service suivants et les associer lorsque vous connectez votre compte.

Pour plus d'informations sur les éléments que vous utilisez dans une politique JSON, consultez la [référence des éléments de stratégie JSON IAM](#) dans le guide de l'utilisateur IAM.

- Pour accéder aux ressources dans un compte AWS pour vos projets CodeCatalyst et flux de travail, vous devez d'abord autoriser l'accès CodeCatalyst à ces ressources en votre nom. Pour ce faire, vous devez créer un rôle de service dans un compte AWS connecté qui CodeCatalyst peut assumer la responsabilité des utilisateurs et des projets de l'espace. Vous pouvez soit choisir de

créer et d'utiliser le rôle de CodeCatalystWorkflowDevelopmentRole-*spaceName*service, soit créer des rôles de service personnalisés et configurer ces politiques et rôles IAM manuellement. Il est recommandé d'attribuer à ces rôles le moins d'autorisations nécessaires.

#### Note

Pour les rôles de service personnalisés, le principal CodeCatalyst de service est requis. Pour plus d'informations sur le principal CodeCatalyst de service et le modèle de confiance, consultez [CodeCatalyst modèle de confiance](#).

- Pour gérer le support d'un espace via le connectedCompte AWS, vous pouvez choisir de créer et d'utiliser le rôle de AWSRoleForCodeCatalystSupportservice qui permet aux CodeCatalyst utilisateurs d'accéder au support. Pour plus d'informations sur la prise en charge d'un CodeCatalyst espace, consultez [AWS Supportpour Amazon CodeCatalyst](#).

## Comprendre le rôle CodeCatalystWorkflowDevelopmentRole-*spaceName*du service

Vous pouvez ajouter un rôle IAM à votre espace qui CodeCatalyst peut être utilisé pour créer des ressources et y accéder dans un espace connectéCompte AWS. C'est ce que l'on appelle un [rôle de service](#). Le moyen le plus simple de créer un rôle de service est d'en ajouter un lorsque vous créez l'espace et de choisir l'CodeCatalystWorkflowDevelopmentRole-*spaceName*option pour ce rôle. Cela crée non seulement le rôle de service avec les AdministratorAccess éléments attachés, mais également la politique de confiance qui permet CodeCatalyst d'assumer le rôle au nom des utilisateurs dans les projets de l'espace. Le rôle de service est limité à l'espace, et non à des projets individuels. Pour créer ce rôle, consultez [Création du CodeCatalystWorkflowDevelopmentRole-\*spaceName\*rôle pour votre compte et votre espace](#). Vous ne pouvez créer qu'un seul rôle pour chaque espace de chaque compte.

#### Note

Ce rôle est uniquement recommandé pour les comptes de développement et utilise la politique AdministratorAccess AWS gérée, ce qui lui donne un accès complet pour créer de nouvelles politiques et ressources dans ce cadreCompte AWS.

La politique associée au CodeCatalystWorkflowDevelopmentRole-*spaceName*rôle est conçue pour fonctionner avec des projets créés avec des plans dans l'espace. Il permet aux utilisateurs de ces

projets de développer, de créer, de tester et de déployer du code en utilisant les ressources du ConnectedCompte AWS. Pour plus d'informations, consultez [la section Création d'un rôle pour un AWS service](#).

La politique attachée au CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle est la stratégie AdministratorAccess gérée dans AWS. Il s'agit d'une politique qui accorde un accès complet à toutes les AWS actions et ressources. Pour consulter le document de politique JSON dans la console IAM, consultez [AdministratorAccess](#).

La politique de confiance suivante permet CodeCatalyst d'assumer le CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle. Pour plus d'informations sur le modèle de CodeCatalyst confiance, consultez [CodeCatalyst modèle de confiance](#).

```
"Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:codecatalyst:::space/spaceId/project/*"
        }
      }
    }
  ]
```

### Création du CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle pour votre compte et votre espace

Suivez ces étapes pour créer le CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle qui sera utilisé pour les flux de travail de votre espace. Pour chaque compte pour lequel vous souhaitez attribuer des rôles IAM à utiliser dans des projets, vous devez ajouter à votre espace un rôle tel que le rôle de développeur.


Avant de commencer, vous devez disposer de privilèges administratifs pour votre administrateur Compte AWS ou être en mesure de travailler avec celui-ci. Pour plus d'informations sur la manière dont Comptes AWS les rôles IAM et IAM sont utilisés dans CodeCatalyst, consultez [Administration Comptes AWS d'un espace](#).

Pour créer et ajouter le CodeCatalyst CodeCatalystWorkflowDevelopmentRole-*spaceName*

1. Avant de commencer dans la CodeCatalyst console, ouvrez leAWS Management Console, puis assurez-vous que vous êtes connecté avec le même identifiant Compte AWS pour votre espace.
2. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
3. Accédez à votre CodeCatalyst espace. Choisissez Settings (Paramètres), puis Comptes AWS.
4. Choisissez le lien correspondant à l'Compte AWSendroit où vous souhaitez créer le rôle. La page Compte AWSde détails s'affiche.
5. Choisissez Gérer les rôles à partir de AWS Management Console.

La page Ajouter un rôle IAM à Amazon CodeCatalyst Space s'ouvre dans leAWS Management Console. Voici la page Amazon CodeCatalyst Spaces. Il se peut que vous deviez vous connecter pour accéder à la page.

6. Choisissez Créer un rôle d'administrateur de CodeCatalyst développement dans IAM. Cette option crée un rôle de service qui contient la politique d'autorisation et la politique de confiance pour le rôle de développement. Le rôle aura un nomCodeCatalystWorkflowDevelopmentRole-*spaceName*. Pour plus d'informations sur le rôle et la politique de rôle, consultez [Comprendre le rôle CodeCatalystWorkflowDevelopmentRole-\*spaceName\*du service](#).

 Note

Ce rôle est uniquement recommandé pour les comptes de développeur et utilise la politique AdministratorAccess AWS gérée, ce qui lui donne un accès complet pour créer de nouvelles politiques et ressources dans ce cadreCompte AWS.

7. Choisissez Créer un rôle de développement.
8. Sur la page des connexions, sous Rôles IAM disponibles pour CodeCatalyst, consultez le CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle dans la liste des rôles IAM ajoutés à votre compte.
9. Pour retourner dans votre espace, choisissez Go to Amazon CodeCatalyst.

## Comprendre le rôle AWSRoleForCodeCatalystSupport du service

Vous pouvez ajouter un rôle IAM à votre espace que CodeCatalyst les utilisateurs d'un espace peuvent utiliser pour créer des dossiers d'assistance et y accéder. C'est ce qu'on appelle un [rôle de service](#) pour le support. Le moyen le plus simple de créer un rôle de service pour le support est d'en ajouter un lorsque vous créez l'espace et de choisir l'`AWSRoleForCodeCatalystSupport` option pour ce rôle. Cela crée non seulement la politique et le rôle, mais également la politique de confiance qui permet à CodeCatalyst d'assumer le rôle au nom des utilisateurs dans les projets de l'espace. Le rôle de service est limité à l'espace, et non à des projets individuels. Pour créer ce rôle, consultez [Création du AWSRoleForCodeCatalystSupport rôle pour votre compte et votre espace](#).

La politique attachée au `AWSRoleForCodeCatalystSupport` rôle est une politique gérée qui donne accès aux autorisations de support. Pour de plus amples informations, veuillez consulter [Politique gérée par AWS :AmazonCodeCatalystSupportAccess](#).

Le rôle de confiance de la politique permet CodeCatalyst d'assumer le rôle.

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

## Création du AWSRoleForCodeCatalystSupport rôle pour votre compte et votre espace

Suivez ces étapes pour créer le `AWSRoleForCodeCatalystSupport` rôle qui sera utilisé pour les demandes d'assistance dans votre espace. Le rôle doit être ajouté au compte de facturation désigné pour l'espace.

Avant de commencer, vous devez disposer de privilèges administratifs pour votre administrateur Compte AWS ou être en mesure de travailler avec celui-ci. Pour plus d'informations sur la manière

dont Comptes AWS les rôles IAM et IAM sont utilisés dans CodeCatalyst, consultez [Administration Comptes AWS d'un espace](#).

Pour créer et ajouter CodeCatalyst AWSRoleForCodeCatalystSupport

1. Avant de commencer dans la CodeCatalyst console, ouvrez leAWS Management Console, puis assurez-vous que vous êtes connecté avec le même identifiant Compte AWS pour votre espace.
2. Accédez à votre CodeCatalyst espace. Choisissez Settings (Paramètres), puis Comptes AWS.
3. Choisissez le lien correspondant à l'Compte AWSendroit où vous souhaitez créer le rôle. La page Compte AWSde détails s'affiche.
4. Choisissez Gérer les rôles à partir de AWS Management Console.

La page Ajouter un rôle IAM à Amazon CodeCatalyst Space s'ouvre dans leAWS Management Console. Voici la page Amazon CodeCatalyst Spaces. Il se peut que vous deviez vous connecter pour accéder à la page.

5. Sous Détails de CodeCatalyst l'espace, choisissez Ajouter un rôle de CodeCatalyst support. Cette option crée un rôle de service qui contient la politique d'autorisation et la politique de confiance pour le rôle de développement préliminaire. Le rôle portera un nom AWSRoleForCodeCatalystSupportavec un identifiant unique ajouté. Pour plus d'informations sur le rôle et la politique de rôle, consultez [Comprendre le rôle AWSRoleForCodeCatalystSupportdu service](#).
6. Sur la page Ajouter un rôle pour le CodeCatalyst support, laissez la valeur par défaut sélectionnée, puis choisissez Créer un rôle.
7. Sous Rôles IAM disponibles pour CodeCatalyst, consultez le CodeCatalystWorkflowDevelopmentRole-*spaceName* rôle dans la liste des rôles IAM ajoutés à votre compte.
8. Pour retourner dans votre espace, choisissez Go to Amazon CodeCatalyst.

Configuration des rôles IAM pour les actions de flux de travail dans CodeCatalyst

Cette section décrit les rôles et les politiques IAM que vous pouvez créer pour les utiliser avec votre CodeCatalyst compte. Pour obtenir des instructions sur la création d'exemples de rôles, consultez [Création manuelle de rôles pour les actions de flux de travail](#). Après avoir créé votre rôle IAM, copiez l'ARN du rôle pour ajouter le rôle IAM à la connexion de votre compte et associez-le à l'environnement de votre projet. Pour en savoir plus, veuillez consulter la section [Ajouter des rôles IAM aux connexions aux comptes](#).

## CodeCatalyst rôle de construction pour l'accès à Amazon S3

Pour les actions de création de CodeCatalyst flux de travail, vous pouvez utiliser le rôle de CodeCatalystWorkflowDevelopmentRole-*spaceName*service par défaut ou créer un rôle IAM nommé CodeCatalystBuildRoleforS3Access. Ce rôle utilise une politique avec des autorisations délimitées qui CodeCatalyst doit exécuter des tâches sur les AWS CloudFormation ressources de votreCompte AWS.

Ce rôle autorise les utilisateurs à effectuer les opérations suivantes :

- Écrivez dans des compartiments Amazon S3.
- Support à la création de ressources avecAWS CloudFormation. Cela nécessite un accès à Amazon S3.

Ce rôle utilise la politique suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "s3:PutObject",
      "iam:PassRole"
    ],
    "Resource": "resource_ARN",
    "Effect": "Allow"
  }]
}
```

### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"

```



## CodeCatalyst créer un rôle pour AWS CloudFormation

Pour les actions de création de CodeCatalyst flux de travail, vous pouvez utiliser le rôle de CodeCatalystWorkflowDevelopmentRole-*spaceName*service par défaut ou créer un rôle IAM avec les autorisations nécessaires. Ce rôle utilise une politique avec des autorisations délimitées qui CodeCatalyst doit exécuter des tâches sur les AWS CloudFormation ressources de votreCompte AWS.

Ce rôle autorise les utilisateurs à effectuer les opérations suivantes :

- Support à la création de ressources avecAWS CloudFormation. Cela est requis, de même que le rôle de CodeCatalyst création pour l'accès à Amazon S3 et le rôle de CodeCatalyst déploiement pourAWS CloudFormation.

Les politiques AWS gérées suivantes doivent être associées à ce rôle :

- AWSCloudFormationFullAccess
- IAM FullAccess
- Amazon S3 FullAccess
- API Amazon GatewayAdministrator
- AWSLambdaFullAccess

## CodeCatalyst rôle de construction pour CDK

Pour les CodeCatalyst flux de travail qui exécutent des actions de génération du CDK, tels que les applications Web modernes à trois niveaux, vous pouvez utiliser le rôle de CodeCatalystWorkflowDevelopmentRole-*spaceName*service par défaut ou créer un rôle IAM avec les autorisations nécessaires. Ce rôle utilise une politique avec des autorisations délimitées qui CodeCatalyst doit démarrer et exécuter les commandes de génération du CDK pour les AWS CloudFormation ressources de votre. Compte AWS

Ce rôle autorise les utilisateurs à effectuer les opérations suivantes :

- Écrivez dans des compartiments Amazon S3.
- Support à la création de structures CDK et de piles de AWS CloudFormation ressources. Cela nécessite l'accès à Amazon S3 pour le stockage des artefacts, à Amazon ECR pour le support des référentiels d'images et à SSM pour la gouvernance et la surveillance du système pour les instances virtuelles.

Ce rôle utilise la politique suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*",
        "ecr:*",
        "ssm:*",
        "s3:*",
        "iam:PassRole",
        "iam:GetRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "iam:PutRolePolicy"
      ],
      "Resource": "*"
    }
  ]
}
```

#### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

## CodeCatalyst rôle de déploiement pour AWS CloudFormation

Pour les actions de déploiement de CodeCatalyst flux de travail qui utilisent AWS CloudFormation, vous pouvez utiliser le rôle de CodeCatalystWorkflowDevelopmentRole-*spaceName*service par défaut ou vous pouvez utiliser une politique avec des autorisations étendues qui CodeCatalyst doit exécuter des tâches sur les AWS CloudFormation ressources de votre Compte AWS.

Ce rôle autorise les utilisateurs à effectuer les opérations suivantes :

- Permet CodeCatalyst d'invoquer une fonction  $\lambda$  pour effectuer un déploiement bleu/vert via. AWS CloudFormation
- Permet de CodeCatalyst créer et de mettre à jour des piles et des ensembles de modifications dans. AWS CloudFormation

Ce rôle utilise la politique suivante :

```
{"Action": [  
  "cloudformation:CreateStack",  
  "cloudformation>DeleteStack",  
  "cloudformation:Describe*",  
  "cloudformation:UpdateStack",  
  "cloudformation:CreateChangeSet",  
  "cloudformation>DeleteChangeSet",  
  "cloudformation:ExecuteChangeSet",  
  "cloudformation:SetStackPolicy",  
  "cloudformation:ValidateTemplate",  
  "cloudformation:List*",  
  "iam:PassRole"  
],  
  "Resource": "resource_ARN",  
  "Effect": "Allow"  
}
```

### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

## CodeCatalyst rôle de déploiement pour Amazon EC2

CodeCatalyst les actions de déploiement du flux de travail utilisent un rôle IAM doté des autorisations nécessaires. Ce rôle utilise une politique avec des autorisations délimitées qui CodeCatalyst doit exécuter des tâches sur les ressources Amazon EC2 de votre. Compte AWS La politique par défaut

pour le CodeCatalystWorkflowDevelopmentRole-*spaceName*rôle n'inclut pas les autorisations pour Amazon EC2 ou Amazon EC2 Auto Scaling.

Ce rôle autorise les utilisateurs à effectuer les opérations suivantes :

- Créez des déploiements Amazon EC2.
- Lisez les balises d'une instance ou identifiez une instance Amazon EC2 à l'aide des noms de groupes Auto Scaling.
- Lisez, créez, mettez à jour et supprimez les groupes Amazon EC2 Auto Scaling, les hooks de cycle de vie et les politiques de dimensionnement.
- Publiez des informations dans les rubriques Amazon SNS.
- Récupérez des informations sur les CloudWatch alarmes.
- Lisez et mettez à jour Elastic Load Balancing.

Ce rôle utilise la politique suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:CompleteLifecycleAction",
        "autoscaling>DeleteLifecycleHook",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeLifecycleHooks",
        "autoscaling:PutLifecycleHook",
        "autoscaling:RecordLifecycleActionHeartbeat",
        "autoscaling>CreateAutoScalingGroup",
        "autoscaling:UpdateAutoScalingGroup",
        "autoscaling:EnableMetricsCollection",
        "autoscaling:DescribePolicies",
        "autoscaling:DescribeScheduledActions",
        "autoscaling:DescribeNotificationConfigurations",
        "autoscaling:SuspendProcesses",
        "autoscaling:ResumeProcesses",
        "autoscaling:AttachLoadBalancers",
        "autoscaling:AttachLoadBalancerTargetGroups",
        "autoscaling:PutScalingPolicy",
        "autoscaling:PutScheduledUpdateGroupAction",
```

```

"autoscaling:PutNotificationConfiguration",
"autoscaling:PutWarmPool",
"autoscaling:DescribeScalingActivities",
"autoscaling>DeleteAutoScalingGroup",
"ec2:DescribeInstances",
"ec2:DescribeInstanceStatus",
"ec2:TerminateInstances",
"tag:GetResources",
"sns:Publish",
"cloudwatch:DescribeAlarms",
"cloudwatch:PutMetricAlarm",
"elasticloadbalancing:DescribeLoadBalancers",
"elasticloadbalancing:DescribeInstanceHealth",
"elasticloadbalancing:RegisterInstancesWithLoadBalancer",
"elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
"elasticloadbalancing:DescribeTargetGroups",
"elasticloadbalancing:DescribeTargetHealth",
"elasticloadbalancing:RegisterTargets",
"elasticloadbalancing:DeregisterTargets"
],
"Resource": "resource_ARN"
  }
]
}

```

### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"

```

## CodeCatalyst rôle de déploiement pour Amazon ECS

Pour les actions CodeCatalyst de flux de travail, vous pouvez créer un rôle IAM doté des autorisations nécessaires. Vous pouvez utiliser le rôle de CodeCatalystWorkflowDevelopmentRole-*spaceName*service par défaut ou créer un rôle IAM pour les actions de déploiement à utiliser pour les CodeCatalyst déploiements Lambda. Ce rôle utilise

une politique avec des autorisations délimitées qui CodeCatalyst doit exécuter des tâches sur les ressources Amazon ECS de votre Compte AWS.

Ce rôle autorise les utilisateurs à effectuer les opérations suivantes :

- Lancez le déploiement progressif d'Amazon ECS pour le compte d'un CodeCatalyst utilisateur, sur un compte spécifié dans la CodeCatalyst connexion.
- Lisez, mettez à jour et supprimez des ensembles de tâches Amazon ECS.
- Mettez à jour les groupes cibles, les auditeurs et les règles d'Elastic Load Balancing.
- Appelez les fonctions Lambda.
- Accédez aux fichiers de révision dans les compartiments Amazon S3.
- Récupérez des informations sur les CloudWatch alarmes.
- Publiez des informations dans les rubriques Amazon SNS.

Ce rôle utilise la politique suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "ecs:DescribeServices",
      "ecs:CreateTaskSet",
      "ecs>DeleteTaskSet",
      "ecs:ListClusters",
      "ecs:RegisterTaskDefinition",
      "ecs:UpdateServicePrimaryTaskSet",
      "ecs:UpdateService",
      "elasticloadbalancing:DescribeTargetGroups",
      "elasticloadbalancing:DescribeListeners",
      "elasticloadbalancing:ModifyListener",
      "elasticloadbalancing:DescribeRules",
      "elasticloadbalancing:ModifyRule",
      "lambda:InvokeFunction",
      "lambda:ListFunctions",
      "cloudwatch:DescribeAlarms",
      "sns:Publish",
      "sns:ListTopics",
      "s3:GetObject",
      "s3:GetObjectVersion",
      "codedeploy:CreateApplication",
```

```

    "codedeploy:CreateDeployment",
    "codedeploy:CreateDeploymentGroup",
    "codedeploy:GetApplication",
    "codedeploy:GetDeployment",
    "codedeploy:GetDeploymentGroup",
    "codedeploy:ListApplications",
    "codedeploy:ListDeploymentGroups",
    "codedeploy:ListDeployments",
    "codedeploy:StopDeployment",
    "codedeploy:GetDeploymentTarget",
    "codedeploy:ListDeploymentTargets",
    "codedeploy:GetDeploymentConfig",
    "codedeploy:GetApplicationRevision",
    "codedeploy:RegisterApplicationRevision",
    "codedeploy:BatchGetApplicationRevisions",
    "codedeploy:BatchGetDeploymentGroups",
    "codedeploy:BatchGetDeployments",
    "codedeploy:BatchGetApplications",
    "codedeploy:ListApplicationRevisions",
    "codedeploy:ListDeploymentConfigs",
    "codedeploy:ContinueDeployment"
  ],
  "Resource": "*",
  "Effect": "Allow"
}, {"Action": [
  "iam:PassRole"
],
  "Effect": "Allow",
  "Resource": "*",
  "Condition": {"StringLike": {"iam:PassedToService": [
    "ecs-tasks.amazonaws.com",
    "codedeploy.amazonaws.com"
  ]
  }
}
}]
}

```

### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

## CodeCatalyst rôle de déploiement pour Lambda

Pour les actions CodeCatalyst de flux de travail, vous pouvez créer un rôle IAM doté des autorisations nécessaires. Vous pouvez utiliser le rôle de CodeCatalystWorkflowDevelopmentRole-*spaceName*service par défaut ou créer un rôle IAM pour les actions de déploiement à utiliser pour les CodeCatalyst déploiements Lambda. Ce rôle utilise une politique avec des autorisations délimitées qui CodeCatalyst doit exécuter des tâches sur les ressources Lambda de votre. Compte AWS

Ce rôle autorise les utilisateurs à effectuer les opérations suivantes :

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

Ce rôle utilise la politique suivante :

```
*{*
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:DescribeAlarms",
        "lambda:UpdateAlias",
        "lambda:GetAlias",
        "lambda:GetProvisionedConcurrencyConfig",
        "sns:Publish"
      ],
      "Resource": "resource_ARN",
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:GetObject",
```



```

        "s3:GetObjectVersion"
    ],
    "Resource": "arn:aws:s3:::/CodeDeploy/",
    "Effect": "Allow"
  },
  {
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion"
    ],
    "Resource": "",
    "Condition": {
      "StringEquals": {
        "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
      }
    },
    "Effect": "Allow"
  },
  {
    "Action": [
      "lambda:InvokeFunction"
    ],
    "Resource": "arn:aws:lambda::function:CodeDeployHook_*",
    "Effect": "Allow"
  }
]
}

```

### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"

```

## CodeCatalyst rôle de déploiement pour Lambda

Pour les actions de CodeCatalyst flux de travail, vous pouvez utiliser le rôle de CodeCatalystWorkflowDevelopmentRole-*spaceName*service par défaut ou créer un rôle IAM avec

les autorisations nécessaires. Ce rôle utilise une politique avec des autorisations délimitées qui CodeCatalyst doit exécuter des tâches sur les ressources Lambda de votre. Compte AWS

Ce rôle autorise les utilisateurs à effectuer les opérations suivantes :

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

Ce rôle utilise la politique suivante :

```
*{*
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:DescribeAlarms",
        "lambda:UpdateAlias",
        "lambda:GetAlias",
        "lambda:GetProvisionedConcurrencyConfig",
        "sns:Publish"
      ],
      "Resource": "resource_ARN",
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "arn:aws:s3:::/CodeDeploy/",
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "",
      "Condition": {
```

```
        "StringEquals": {
            "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
        }
    },
    "Effect": "Allow"
},
{
    "Action": [
        "lambda:InvokeFunction"
    ],
    "Resource": "arn:aws:lambda::function:CodeDeployHook_*",
    "Effect": "Allow"
}
]
```

### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"

```

## CodeCatalyst rôle de déploiement pour AWS SAM

Pour les actions de CodeCatalyst flux de travail, vous pouvez utiliser le rôle de CodeCatalystWorkflowDevelopmentRole-*spaceName*service par défaut ou créer un rôle IAM avec les autorisations nécessaires. Ce rôle utilise une politique avec des autorisations délimitées qui CodeCatalyst doit exécuter des tâches AWS SAM et AWS CloudFormation des ressources sur votreCompte AWS.

Ce rôle autorise les utilisateurs à effectuer les opérations suivantes :

- Permet CodeCatalyst d'invoquer une fonction Lambda pour effectuer le déploiement d'applications sans serveur et d'applications CLI AWS SAM.
- Permet de CodeCatalyst créer et de mettre à jour des piles et des ensembles de modifications dans. AWS CloudFormation

Ce rôle utilise la politique suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "iam:PassRole",
        "iam>DeleteRole",
        "iam:GetRole",
        "iam:TagRole",
        "iam>CreateRole",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "cloudformation:*",
        "lambda:*",
        "apigateway:*"
      ],
      "Resource": "*"
    }
  ]
}
```

#### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

## CodeCatalyst rôle en lecture seule pour Amazon EC2

Pour les actions CodeCatalyst de flux de travail, vous pouvez créer un rôle IAM doté des autorisations nécessaires. Ce rôle utilise une politique avec des autorisations délimitées qui CodeCatalyst doit exécuter des tâches sur les ressources Amazon EC2 de votre. Compte AWS Le

rôle CodeCatalystWorkflowDevelopmentRole-*spaceName* de service n'inclut pas les autorisations pour Amazon EC2 ni les actions décrites pour Amazon. CloudWatch

Ce rôle autorise les utilisateurs à effectuer les opérations suivantes :

- Obtenez le statut des instances Amazon EC2.
- Obtenez CloudWatch des statistiques pour les instances Amazon EC2.

Ce rôle utilise la politique suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:Describe",
      "Resource": "resource_ARN"
    },
    {
      "Effect": "Allow",
      "Action": "elasticloadbalancing:Describe",
      "Resource": "resource_ARN"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:ListMetrics",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:Describe"
      ],
      "Resource": "resource_ARN"
    },
    {
      "Effect": "Allow",
      "Action": "autoscaling:Describe",
      "Resource": "resource_ARN"
    }
  ]
}
```

**Note**

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

## CodeCatalyst rôle en lecture seule pour Amazon ECS

Pour les actions CodeCatalyst de flux de travail, vous pouvez créer un rôle IAM doté des autorisations nécessaires. Ce rôle utilise une politique avec des autorisations délimitées qui CodeCatalyst doit exécuter des tâches sur les ressources Amazon ECS de votre Compte AWS.

Ce rôle autorise les utilisateurs à effectuer les opérations suivantes :

- Lisez les ensembles de tâches Amazon ECS.
- Récupérez des informations sur les CloudWatch alarmes.

Ce rôle utilise la politique suivante :

```
*{*
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ecs:DescribeServices",
        "cloudwatch:DescribeAlarms"
      ],
      "Resource": "resource_ARN",
      "Effect": "Allow"
    },
    {
      "Action": [
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeListeners",
        "elasticloadbalancing:DescribeRules"
      ],
      "Resource": "resource_ARN",
```

```

    "Effect": "Allow"
  },
  {
    "Action": [
      "s3:GetObject",
      "s3:GetObjectVersion"
    ],
    "Resource": "",
    "Condition": {
      "StringEquals": {
        "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
      }
    },
    "Effect": "Allow"
  },
  {
    "Action": [
      "iam:PassRole"
    ],
    "Effect": "Allow",
    "Resource": [
      "arn:aws:iam:::role/ecsTaskExecutionRole",
      "arn:aws:iam:::role/ECSTaskExecution"
    ],
    "Condition": {
      "StringLike": {
        "iam:PassedToService": [
          "ecs-tasks.amazonaws.com"
        ]
      }
    }
  }
]
}

```

### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"

```

## CodeCatalyst rôle en lecture seule pour Lambda

Pour les actions CodeCatalyst de flux de travail, vous pouvez créer un rôle IAM doté des autorisations nécessaires. Ce rôle utilise une politique avec des autorisations délimitées qui CodeCatalyst doit exécuter des tâches sur les ressources Lambda de votre. Compte AWS

Ce rôle donne des autorisations pour ce qui suit :

- Lisez les fonctions Lambda et les alias.
- Accédez aux fichiers de révision dans les compartiments Amazon S3.
- Récupérez des informations sur les CloudWatch alarmes.

Le rôle utilise la stratégie suivante.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:DescribeAlarms",
        "lambda:GetAlias",
        "lambda:GetProvisionedConcurrencyConfig"
      ],
      "Resource": "resource_ARN",
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "arn:aws:s3:::/CodeDeploy/",
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:GetObject",

```



```
        "s3:GetObjectVersion"
    ],
    "Resource": "",
    "Condition": {
        "StringEquals": {
            "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
        }
    },
    "Effect": "Allow"
}
]
```

### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"

```

## Création manuelle de rôles pour les actions de flux de travail

CodeCatalyst les actions de flux de travail utilisent les rôles IAM que vous créez, appelés rôle de construction, rôle de déploiement et rôle de pile.


Suivez ces étapes pour créer ces rôles dans IAM.

### Pour créer un rôle de déploiement

1. Créez une politique pour le rôle, comme suit :
  - a. Connectez-vous à AWS.
  - b. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
  - c. Dans le panneau de navigation, choisissez Politiques (Politiques).
  - d. Sélectionnez Create policy (Créer une politique).
  - e. Choisissez l'onglet JSON.
  - f. Supprimez le code existant.

g. Collez le code suivant :

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "cloudformation:CreateStack",
      "cloudformation>DeleteStack",
      "cloudformation:Describe*",
      "cloudformation:UpdateStack",
      "cloudformation:CreateChangeSet",
      "cloudformation>DeleteChangeSet",
      "cloudformation:ExecuteChangeSet",
      "cloudformation:SetStackPolicy",
      "cloudformation:ValidateTemplate",
      "cloudformation:List*",
      "iam:PassRole"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }]
}
```

 Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

- h. Choisissez Suivant : Balises.
- i. Choisissez Suivant : Vérification.
- j. Dans Nom, entrez :

```
codecatalyst-deploy-policy
```

- k. Choisissez Créer une politique.

Vous venez de créer une politique d'autorisation.

## 2. Créez le rôle de déploiement comme suit :

- a. Dans le volet de navigation, sélectionnez Rôles, puis Créer un rôle.
- b. Choisissez Politique de confiance personnalisée.
- c. Supprimez la politique de confiance personnalisée existante.
- d. Ajoutez la politique de confiance personnalisée suivante :

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

- e. Choisissez Suivant.
- f. Dans Politiques d'autorisations, recherchez `codecatalyst-deploy-policy` et cochez sa case.
- g. Choisissez Suivant.
- h. Dans Nom du rôle, entrez :

**codecatalyst-deploy-role**

- i. Pour la description du rôle, entrez :

**CodeCatalyst deploy role**

- j. Sélectionnez Créer un rôle.

Vous venez de créer un rôle de déploiement avec une politique de confiance et une politique d'autorisations.

3. Obtenez l'ARN du rôle de déploiement, comme suit :
  - a. Dans le panneau de navigation, choisissez Roles (Rôles).
  - b. Dans le champ de recherche, entrez le nom du rôle que vous venez de créer (codecatalyst-deploy-role).
  - c. Choisissez le rôle dans la liste.

La page Résumé du rôle s'affiche.

- d. En haut, copiez la valeur de l'ARN.

Vous avez maintenant créé le rôle de déploiement avec les autorisations appropriées et obtenu son ARN.

#### Pour créer un rôle de build

1. Créez une politique pour le rôle, comme suit :
  - a. Connectez-vous à AWS.
  - b. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
  - c. Dans le panneau de navigation, choisissez Politiques (Politiques).
  - d. Sélectionnez Create policy (Créer une politique).
  - e. Choisissez l'onglet JSON.
  - f. Supprimez le code existant.
  - g. Collez le code suivant :

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "s3:PutObject",
      "iam:PassRole"
    ],
    "Resource": "*"
  }
]
```

```
"Effect": "Allow"
}]
}
```

### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
}
```

- h. Choisissez Suivant : Balises.
- i. Choisissez Suivant : Vérification.
- j. Dans Nom, entrez :

```
codecatalyst-build-policy
```

- k. Choisissez Créer une politique.

Vous venez de créer une politique d'autorisation.

2. Créez le rôle de build, comme suit :
  - a. Dans le volet de navigation, sélectionnez Rôles, puis Créer un rôle.
  - b. Choisissez Politique de confiance personnalisée.
  - c. Supprimez la politique de confiance personnalisée existante.
  - d. Ajoutez la politique de confiance personnalisée suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      }
    }
  ]
}
```

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

- e. Choisissez Suivant.
- f. Dans Politiques d'autorisations, recherchez `codecatalyst-build-policy` et cochez sa case.
- g. Choisissez Suivant.
- h. Dans Nom du rôle, entrez :

**codecatalyst-build-role**

- i. Pour la description du rôle, entrez :

**CodeCatalyst build role**

- j. Sélectionnez Créer un rôle.

Vous avez maintenant créé un rôle de build avec une politique de confiance et une politique d'autorisations.

3. Obtenez l'ARN du rôle de build, comme suit :
  - a. Dans le panneau de navigation, choisissez Roles (Rôles).
  - b. Dans le champ de recherche, entrez le nom du rôle que vous venez de créer (`codecatalyst-build-role`).
  - c. Choisissez le rôle dans la liste.

La page Résumé du rôle s'affiche.

- d. En haut, copiez la valeur de l'ARN.

Vous avez maintenant créé le rôle de build avec les autorisations appropriées et obtenu son ARN.

## Pour créer un rôle de pile

### Note

Il n'est pas nécessaire de créer un rôle de pile, bien que cela soit recommandé pour des raisons de sécurité. Si vous ne créez pas le rôle de pile, vous devrez ajouter les politiques d'autorisation décrites plus loin dans cette procédure au rôle de déploiement.

1. Connectez-vous à AWS l'aide du compte sur lequel vous souhaitez déployer votre stack.
2. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
3. Dans le volet de navigation, sélectionnez Rôles, puis sélectionnez Créer un rôle.
4. En haut de la page, sélectionnez le AWSservice.
5. Dans la liste des services, sélectionnez CloudFormation.
6. Sélectionnez Next: Permissions (Étape suivante : autorisations).
7. Dans le champ de recherche, ajoutez les politiques nécessaires pour accéder aux ressources de votre pile. Par exemple, si votre pile inclut une AWS Lambda fonction, vous devez ajouter une politique qui accorde l'accès à Lambda.

### Tip

Si vous ne savez pas quelles politiques ajouter, vous pouvez les omettre pour le moment. Lorsque vous testez l'action, si vous ne disposez pas des autorisations appropriées AWS CloudFormation, des erreurs indiquent les autorisations que vous devez ajouter.

8. Choisissez Suivant : Balises.
9. Choisissez Suivant : Vérification.
10. Dans Nom du rôle, entrez :

**codecatalyst-stack-role**

11. Sélectionnez Créer un rôle.
12. Pour obtenir l'ARN du rôle de pile, procédez comme suit :
  - a. Dans le panneau de navigation, choisissez Roles (Rôles).

- b. Dans le champ de recherche, entrez le nom du rôle que vous venez de créer (codecatalyst-stack-role).
- c. Choisissez le rôle dans la liste.
- d. Sur la page Résumé, copiez la valeur ARN du rôle.

## Utilisation AWS CloudFormation pour créer des politiques et des rôles dans IAM

Vous pouvez choisir de créer et d'utiliser des AWS CloudFormation modèles pour créer les politiques et les rôles dont vous avez besoin pour accéder aux ressources dans et Compte AWS pour vos CodeCatalyst projets et flux de travail. AWS 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 sur AWS. Si vous avez l'intention de créer plusieurs rôles Comptes AWS, la création d'un modèle peut vous aider à effectuer cette tâche plus rapidement.

L'exemple de modèle suivant crée un rôle et une politique d'action de déploiement.

```
Parameters:
  CodeCatalystAccountId:
    Type: String
    Description: Account ID from the connections page
  ExternalId:
    Type: String
    Description: External ID from the connections page
Resources:
  CrossAccountRole:
    Type: 'AWS::IAM::Role'
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          - Effect: Allow
            Principal:
              AWS:
                - !Ref CodeCatalystAccountId
            Action:
              - 'sts:AssumeRole'
            Condition:
              StringEquals:
                sts:ExternalId: !Ref ExternalId
    Path: /
```



```
Policies:
- PolicyName: CodeCatalyst-CloudFormation-action-policy
  PolicyDocument:
    Version: "2012-10-17"
    Statement:
      - Effect: Allow
        Action:
          - 'cloudformation:CreateStack'
          - 'cloudformation>DeleteStack'
          - 'cloudformation:Describe*'
          - 'cloudformation:UpdateStack'
          - 'cloudformation:CreateChangeSet'
          - 'cloudformation>DeleteChangeSet'
          - 'cloudformation:ExecuteChangeSet'
          - 'cloudformation:SetStackPolicy'
          - 'cloudformation:ValidateTemplate'
          - 'cloudformation:List*'
          - 'iam:PassRole'
        Resource: '*'
```

## Création manuelle du rôle pour le plan de l'application Web

Le plan d'application CodeCatalyst Web utilise les rôles IAM que vous créez, appelés rôle de construction pour CDK, rôle de déploiement et rôle de pile.

Procédez comme suit pour créer le rôle dans IAM.

Pour créer un rôle de build

1. Créez une politique pour le rôle, comme suit :
  - a. Connectez-vous à AWS.
  - b. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
  - c. Dans le panneau de navigation, choisissez Politiques (Politiques).
  - d. Choisissez Create Policy (Créer une politique).
  - e. Choisissez l'onglet JSON.
  - f. Supprimez le code existant.
  - g. Collez le code suivant :

```
{
  "Version": "2012-10-17",
```

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "cloudformation:*",  
      "ecr:*",  
      "ssm:*",  
      "s3:*",  
      "iam:PassRole",  
      "iam:GetRole",  
      "iam:CreateRole",  
      "iam:AttachRolePolicy",  
      "iam:PutRolePolicy"  
    ],  
    "Resource": "*"   
  }  
]
```

#### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"   

```

- h. Choisissez Suivant : Balises.
- i. Choisissez Suivant : Vérification.
- j. Dans Nom, entrez :

```
codecatalyst-webapp-build-policy
```

- k. Choisissez Créer une politique.

Vous venez de créer une politique d'autorisation.

2. Créez le rôle de build, comme suit :
  - a. Dans le volet de navigation, sélectionnez Rôles, puis Créer un rôle.
  - b. Choisissez Politique de confiance personnalisée.

- c. Supprimez la politique de confiance personnalisée existante.
- d. Ajoutez la politique de confiance personnalisée suivante :

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

- e. Choisissez Suivant.
- f. Associez la politique d'autorisations au rôle de build. Sur la page Ajouter des autorisations, dans la section Politiques d'autorisations, recherchez `codecatalyst-webapp-build-policy` et cochez la case correspondante.
- g. Choisissez Suivant.
- h. Dans Nom du rôle, entrez :

**codecatalyst-webapp-build-role**

- i. Pour la description du rôle, entrez :

**CodeCatalyst Web app build role**

- j. Sélectionnez Créer un rôle.

Vous avez maintenant créé un rôle de build avec une politique de confiance et une politique d'autorisations.

3. Associez la politique d'autorisations au rôle de build, comme suit :

- a. Dans le volet de navigation, choisissez Rôles, puis recherchez `codecatalyst-webapp-build-role`.
- b. Choisissez `codecatalyst-webapp-build-role` d'en afficher les détails.
- c. Dans l'onglet Autorisations, choisissez Ajouter des autorisations, puis choisissez Joindre des politiques.
- d. Recherchez `codecatalyst-webapp-build-policy`, cochez sa case, puis choisissez Joindre des politiques.

Vous avez maintenant associé la politique d'autorisations au rôle de build. Le rôle de création dispose désormais de deux politiques : une politique d'autorisation et une politique de confiance.

4. Obtenez l'ARN du rôle de build, comme suit :
  - a. Dans le panneau de navigation, choisissez Roles (Rôles).
  - b. Dans le champ de recherche, entrez le nom du rôle que vous venez de créer (`codecatalyst-webapp-build-role`).
  - c. Choisissez le rôle dans la liste.

La page Résumé du rôle s'affiche.

- d. En haut, copiez la valeur de l'ARN.

Vous avez maintenant créé le rôle de build avec les autorisations appropriées et obtenu son ARN.

## Création manuelle de rôles pour le plan SAM

Le plan CodeCatalyst SAM utilise les rôles IAM que vous créez, appelés rôle de création CloudFormation et rôle de déploiement pour SAM.


Suivez ces étapes pour créer les rôles dans IAM.

Pour créer un rôle de build pour CloudFormation

1. Créez une politique pour le rôle, comme suit :
  - a. Connectez-vous à AWS.

- b. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
- c. Dans le panneau de navigation, choisissez Politiques (Politiques).
- d. Choisissez Create Policy (Créer une politique).
- e. Choisissez l'onglet JSON.
- f. Supprimez le code existant.
- g. Collez le code suivant :

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

 Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

- h. Choisissez Suivant : Balises.
- i. Choisissez Suivant : Vérification.
- j. Dans Nom, entrez :

```
codecatalyst-SAM-build-policy
```

- k. Choisissez Créer une politique.

**Vous venez de créer une politique d'autorisation.**

2. Créez le rôle de build, comme suit :
  - a. Dans le volet de navigation, sélectionnez Rôles, puis Créer un rôle.
  - b. Choisissez Politique de confiance personnalisée.
  - c. Supprimez la politique de confiance personnalisée existante.
  - d. Ajoutez la politique de confiance personnalisée suivante :

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

- e. Choisissez Suivant.
- f. Associez la politique d'autorisations au rôle de build. Sur la page Ajouter des autorisations, dans la section Politiques d'autorisations, recherchez `codecatalyst-SAM-build-policy` et cochez la case correspondante.
- g. Choisissez Suivant.
- h. Dans Nom du rôle, entrez :

**codecatalyst-SAM-build-role**

- i. Pour la description du rôle, entrez :

**CodeCatalyst SAM build role**

- j. Sélectionnez Créer un rôle.

Vous avez maintenant créé un rôle de build avec une politique de confiance et une politique d'autorisations.

3. Associez la politique d'autorisations au rôle de build, comme suit :
  - a. Dans le volet de navigation, choisissez Rôles, puis recherchez `codecatalyst-SAM-build-role`.
  - b. Choisissez `codecatalyst-SAM-build-role` d'en afficher les détails.
  - c. Dans l'onglet Autorisations, choisissez Ajouter des autorisations, puis choisissez Joindre des politiques.
  - d. Recherchez `codecatalyst-SAM-build-policy`, cochez sa case, puis choisissez Joindre des politiques.

Vous avez maintenant associé la politique d'autorisations au rôle de build. Le rôle de création dispose désormais de deux politiques : une politique d'autorisation et une politique de confiance.

4. Obtenez l'ARN du rôle de build, comme suit :
  - a. Dans le panneau de navigation, choisissez Roles (Rôles).
  - b. Dans le champ de recherche, entrez le nom du rôle que vous venez de créer (`codecatalyst-SAM-build-role`).
  - c. Choisissez le rôle dans la liste.

La page Résumé du rôle s'affiche.

- d. En haut, copiez la valeur de l'ARN.

Vous avez maintenant créé le rôle de build avec les autorisations appropriées et obtenu son ARN.

## Pour créer un rôle de déploiement pour SAM

1. Créez une politique pour le rôle, comme suit :
  - a. Connectez-vous à AWS.
  - b. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.

- c. Dans le panneau de navigation, choisissez Politiques (Politiques).
- d. Choisissez Create Policy (Créer une politique).
- e. Choisissez l'onglet JSON.
- f. Supprimez le code existant.
- g. Collez le code suivant :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "iam:PassRole",
        "iam>DeleteRole",
        "iam:GetRole",
        "iam:TagRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "cloudformation:*",
        "lambda:*",
        "apigateway:*"
      ],
      "Resource": "*"
    }
  ]
}
```

#### Note

La première fois que le rôle est utilisé pour exécuter des actions de flux de travail, utilisez le caractère générique dans la déclaration de politique de ressources, puis définissez la stratégie avec le nom de la ressource une fois celle-ci disponible.

```
"Resource": "*"
```

- h. Choisissez Suivant : Balises.



- i. Choisissez Suivant : Vérification.
- j. Dans Nom, entrez :

```
codecatalyst-SAM-deploy-policy
```

- k. Choisissez Créer une politique.

Vous venez de créer une politique d'autorisation.

2. Créez le rôle de build, comme suit :

- a. Dans le volet de navigation, sélectionnez Rôles, puis Créer un rôle.
- b. Choisissez Politique de confiance personnalisée.
- c. Supprimez la politique de confiance personnalisée existante.
- d. Ajoutez la politique de confiance personnalisée suivante :

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

- e. Choisissez Suivant.
- f. Associez la politique d'autorisations au rôle de build. Sur la page Ajouter des autorisations, dans la section Politiques d'autorisations, recherchez codecatalyst-SAM-deploy-policy et cochez la case correspondante.
- g. Choisissez Suivant.
- h. Dans Nom du rôle, entrez :

`codecatalyst-SAM-deploy-role`

- i. Pour la description du rôle, entrez :

`CodeCatalyst SAM deploy role`

- j. Sélectionnez Créer un rôle.

Vous avez maintenant créé un rôle de build avec une politique de confiance et une politique d'autorisations.

3. Associez la politique d'autorisations au rôle de build, comme suit :
  - a. Dans le volet de navigation, choisissez Rôles, puis recherchez `codecatalyst-SAM-deploy-role`.
  - b. Choisissez `codecatalyst-SAM-deploy-role` d'en afficher les détails.
  - c. Dans l'onglet Autorisations, choisissez Ajouter des autorisations, puis choisissez Joindre des politiques.
  - d. Recherchez `codecatalyst-SAM-deploy-policy`, cochez sa case, puis choisissez Joindre des politiques.

Vous avez maintenant associé la politique d'autorisations au rôle de build. Le rôle de création dispose désormais de deux politiques : une politique d'autorisation et une politique de confiance.

4. Obtenez l'ARN du rôle de build, comme suit :
  - a. Dans le panneau de navigation, choisissez Roles (Rôles).
  - b. Dans le champ de recherche, entrez le nom du rôle que vous venez de créer (`codecatalyst-SAM-deploy-role`).
  - c. Choisissez le rôle dans la liste.

La page Résumé du rôle s'affiche.

- d. En haut, copiez la valeur de l'ARN.

Vous avez maintenant créé le rôle de build avec les autorisations appropriées et obtenu son ARN.

## Validation de conformité pour Amazon CodeCatalyst

Pour savoir si un Service AWS fait partie du champ d'application de programmes de conformité spécifiques, veuillez consulter [Services AWS dans le champ d'application par programme de conformité](#) et choisissez le programme de conformité qui vous intéresse. Pour obtenir des renseignements généraux, veuillez consulter [AWS Compliance Programs](#) (français non garanti).

Vous pouvez télécharger les rapports de l'audit externe avec AWS Artifact. Pour plus d'informations, veuillez consulter [Downloading Reports dans AWS Artifact](#) (français non garanti).

Votre responsabilité de conformité lors de l'utilisation de Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise, ainsi que par la législation et la réglementation applicables. AWS fournit les ressources suivantes pour faciliter le respect de la conformité :

- [Guides Quick Start de la sécurité et de la conformité](#) : ces guides de déploiement traitent de considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de référence dans AWS centrés sur la sécurité et la conformité.
- [Architecture pour la sécurité et la conformité HIPAA sur Amazon Web Services](#) : ce livre blanc décrit comment les entreprises peuvent utiliser AWS pour créer des applications éligibles à la loi HIPAA.

### Note

Tous les Services AWS ne sont pas éligibles à HIPAA. Pour plus d'informations, veuillez consulter [HIPAA Eligible Services Reference](#) (français non garanti).

- [Ressources de conformité AWS](#) : cet ensemble de manuels et de guides peut s'appliquer à votre secteur d'activité et à votre emplacement.
- [AWS Guides de conformité destinés aux clients](#) — Comprenez le modèle de responsabilité partagée sous l'angle de la conformité. Les guides résument les meilleures pratiques en matière de sécurisation Services AWS et décrivent les directives relatives aux contrôles de sécurité dans de nombreux cadres (notamment le National Institute of Standards and Technology (NIST), le Payment Card Industry Security Standards Council (PCI) et l'Organisation internationale de normalisation (ISO)).
- [Évaluation des ressources à l'aide de règles](#) dans le Guide du développeur AWS Config : le service AWS Config évalue dans quelle mesure vos configurations de ressources sont conformes aux pratiques internes, aux directives sectorielles et aux réglementations.

- [AWS Security Hub](#) : ce Service AWS fournit une vue complète de votre état de sécurité dans AWS. Security Hub utilise des contrôles de sécurité pour évaluer vos ressources AWS et vérifier votre conformité par rapport aux normes et aux bonnes pratiques du secteur de la sécurité. Pour obtenir la liste des services et des contrôles pris en charge, veuillez consulter [Security Hub controls reference](#) (français non garanti).
- [AWS Audit Manager](#) – Ce service Service AWS vous aide à auditer en continu votre utilisation d'AWS pour simplifier la gestion des risques et la conformité aux réglementations et aux normes du secteur.

## Résilience chez Amazon CodeCatalyst

L'infrastructure mondiale d'AWS repose sur les Régions AWS et les zones de disponibilité. Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, reliées par un réseau à latence faible, à débit élevé et à forte redondance. 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 les Régions AWS et les zones de disponibilité, consultez [Infrastructure mondiale d'AWS](#). Pour en savoir plus sur CodeCatalyst les données qui sont répliquées Régions AWS, consultez [Protection des données sur Amazon CodeCatalyst](#).

## Sécurité de l'infrastructure sur Amazon CodeCatalyst

En tant que service géré, Amazon CodeCatalyst est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services de sécurité AWS et la manière dont AWS protège l'infrastructure, consultez la section [Sécurité du cloud AWS](#). Pour concevoir votre environnement AWS en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section [Protection de l'infrastructure](#) dans le Security Pillar AWS Well-Architected Framework (Pilier de sécurité de l'infrastructure Well-Architected Framework).

Vous utilisez des appels d'API AWS publiés pour accéder CodeCatalyst via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et nous recommandons TLS 1.3.

- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser [AWS Security Token Service](#) (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

## Analyse de configuration et de vulnérabilité sur Amazon CodeCatalyst

La configuration et les contrôles informatiques sont une responsabilité partagée entre AWS et vous, notre client. Pour de plus amples informations, veuillez consulter [Modèle de responsabilité partagée AWS](#).

## Vos données et votre confidentialité sur Amazon CodeCatalyst

Amazon CodeCatalyst prend le respect de votre vie privée très au sérieux, et la sécurité de vos informations est notre priorité absolue. Vous pouvez en savoir plus sur la manière dont nous traitons vos informations dans la [AWSPolitique de confidentialité](#).

Pour demander et consulter vos données, consultez la section [Demande de vos données](#) dans leRéférences générales AWS.

## Supprimer votre profil AWS Builder ID

La suppression de votre profil est une action permanente irréversible. Le processus de suppression commence immédiatement après que vous avez sélectionné Supprimer. Amazon CodeCatalyst commence à supprimer votre profil et toutes les informations personnelles associées. Ce processus peut prendre jusqu'à 90 jours.

Lorsque votre profil est supprimé, vous ne pouvez pas accéder à vos données ou les récupérer sur Amazon CodeCatalyst. Cela inclut les jetons d'accès personnels, les rôles, les adhésions d'utilisateurs et tous CodeCatalyst les espaces Amazon dont vous êtes le seul membre. Vous ne pouvez plus vous connecter à Amazon CodeCatalyst.

Pour plus d'informations sur la façon de supprimer votre profil AWS Builder ID, consultez [Supprimer votre AWS Builder ID](#) dans leRéférences générales AWS.

# Bonnes pratiques pour les actions de flux de travail sur Amazon CodeCatalyst

Il existe un certain nombre de bonnes pratiques de sécurité à prendre en compte lorsque vous développez vos flux de travail dans CodeCatalyst. Les directives suivantes sont générales et ne constituent 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

- [Informations sensibles](#)
- [Termes de licence](#)
- [Code non fiable](#)
- [GitHub Les actions](#)

## Informations sensibles

N'intégrez pas d'informations sensibles dans votre fichier YAML. Plutôt que d'intégrer des informations d'identification, des clés ou des jetons dans votre fichier YAML, nous vous recommandons d'utiliser CodeCatalyst des secrets. Les secrets permettent de stocker et de référencer facilement des informations sensibles depuis votre YAML.

## Termes de licence

Assurez-vous de prêter attention aux conditions de licence de l'action que vous choisissez d'utiliser.

## Code non fiable

Les actions sont généralement des modules autonomes à usage unique qui peuvent être partagés au sein d'un projet, d'un espace ou de la communauté au sens large. L'utilisation de code provenant d'autres utilisateurs peut apporter un gain de commodité et d'efficacité considérable, mais elle introduit également un nouveau vecteur de menace. Consultez les sections suivantes pour vous assurer que vous suivez les meilleures pratiques afin de sécuriser vos flux de travail CI/CD.

## GitHub Les actions

GitHub Les actions sont open source, créées et maintenues par la communauté. Nous suivons le [modèle de responsabilité partagée](#) et considérons le code source d' GitHub Actions comme des

données clients dont vous êtes responsable. GitHub Les actions peuvent être autorisées à accéder aux secrets, aux jetons du référentiel, au code source, aux liens de compte et à votre temps de calcul. Assurez-vous d'avoir confiance en la fiabilité et en la sécurité des GitHub actions que vous comptez exécuter.

Conseils plus spécifiques et meilleures pratiques de sécurité pour les GitHub actions :

- [Renforcement de la sécurité](#)
- [Empêcher ses propres demandes](#)
- [Entrée non fiable](#)
- [Comment faire confiance à vos éléments de base](#)

## CodeCatalyst modèle de confiance

Le modèle de CodeCatalyst confiance d'Amazon CodeCatalyst permet d'assumer le rôle de service dans le connectéCompte AWS. Le modèle relie le rôle IAM, les principaux CodeCatalyst de service et l' CodeCatalyst espace. La politique de confiance utilise la clé de `aws:SourceArn` condition pour accorder des autorisations à l' CodeCatalyst espace spécifié dans la clé de condition. Pour plus d'informations sur cette clé de condition, consultez [aws:SourceArn](#) dans le guide de l'utilisateur IAM.

Une politique de confiance est un document de politique JSON dans lequel vous définissez les mandataires auxquels vous faites confiance pour assumer le rôle. Une politique d'approbation de rôle est une politique basée sur les ressources requise qui est attachée à un rôle dans IAM. Pour plus d'informations, consultez la section [Termes et concepts](#) du guide de l'utilisateur IAM. Pour plus de détails sur les principes de service pour CodeCatalyst, voir [Principes de service pour CodeCatalyst](#).

Dans la politique de confiance suivante, les principaux de service répertoriés dans l'Principalélément reçoivent des autorisations issues de la politique basée sur les ressources, et le Condition bloc est utilisé pour limiter l'accès à la ressource délimitée.

```
"Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "codecatalyst-runner.amazonaws.com",
          "codecatalyst.amazonaws.com"
        ]
      }
    }
  ]
```

```
    ]
  },
  "Action": "sts:AssumeRole",
  "Condition": {
    "ArnLike": {
      "aws:SourceArn": "arn:aws:codecatalyst:::space/spaceId/project/*"
    }
  }
}
```

Dans la politique de confiance, les responsables du CodeCatalyst service ont accès via la clé de `aws:SourceArn` condition, qui contient le nom de ressource Amazon (ARN) pour l'identifiant de l'espace CodeCatalyst. L'ARN utilise le format suivant :

```
arn:aws:codecatalyst:::space/spaceId/project/*
```

#### Important

Utilisez l'identifiant d'espace uniquement dans les clés de condition, telles que `aws:SourceArn`. N'utilisez pas l'ID d'espace dans les déclarations de politique IAM comme ARN de ressource.

Il est recommandé de limiter autant que possible les autorisations dans la politique.

- Vous pouvez utiliser le caractère générique (\*) dans la clé de `aws:SourceArn` condition pour spécifier tous les projets dans l'espace avec `project/*`.
- Vous pouvez spécifier des autorisations au niveau des ressources dans la clé de `aws:SourceArn` condition pour un projet spécifique dans l'espace avec `project/projectId`

## Principes de service pour CodeCatalyst

Vous utilisez l'Principalélément dans une politique JSON basée sur les ressources pour spécifier le principal auquel l'accès à une ressource est autorisé ou refusé. Les principaux que vous pouvez spécifier dans la politique d'approbation comprennent les utilisateurs, les rôles, les comptes et les services. Vous ne pouvez pas utiliser l'Principalélément dans une stratégie basée sur l'identité ; de même, vous ne pouvez pas identifier un groupe d'utilisateurs en tant que principal dans une



stratégie (telle qu'une stratégie basée sur les ressources) car les groupes sont liés aux autorisations et non à l'authentification, et les principaux sont des entités IAM authentifiées.

Dans la politique de confiance, vous pouvez spécifier Services AWS dans l'Principalélément d'une politique basée sur les ressources ou dans des clés de condition qui prennent en charge les principaux. Les principes du service sont définis par le service. Les principes de service définis pour CodeCatalyst :

- `codecatalyst.amazonaws.com` - Ce principal de service est utilisé pour un rôle qui accordera l'accès à CodeCatalyst AWS
- `codecatalyst-runner.amazonaws.com` - Ce principal de service est utilisé pour un rôle qui accordera l'accès aux ressources lors des déploiements pour les flux de travail. CodeCatalyst AWS CodeCatalyst

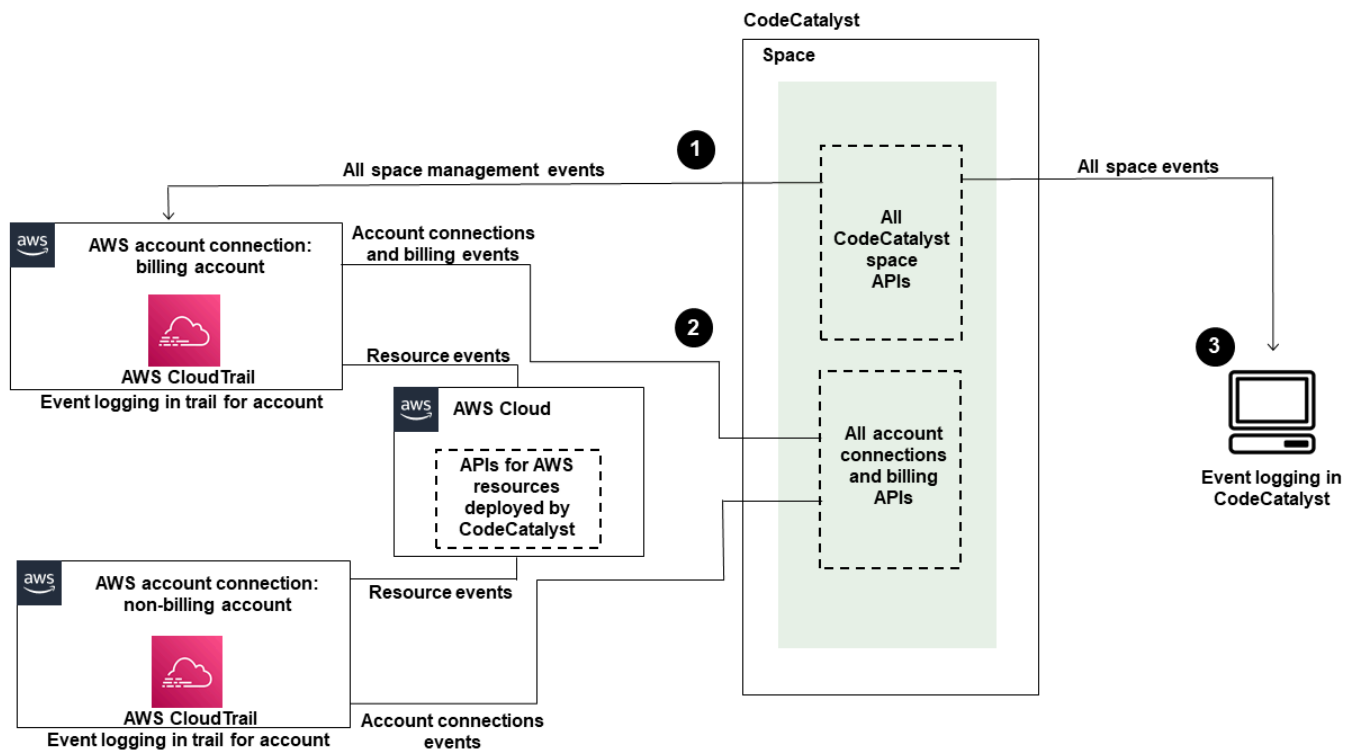
Pour plus d'informations, voir [Éléments de politique AWS JSON : Principal](#) dans le guide de l'utilisateur IAM.

## Surveillance sur Amazon CodeCatalyst

Sur Amazon CodeCatalyst, les événements de gestion de l'espace sont collectés par le compte de facturation de l'espace AWS CloudTrail et sont enregistrés dans le journal de suivi de celui-ci. CloudTrail la journalisation est la principale méthode de gestion de la journalisation des CodeCatalyst événements, et une méthode secondaire consiste à visualiser la connexion aux événements CodeCatalyst.

Les événements du compte sont enregistrés avec le parcours et le compartiment désigné configurés pour le Compte AWS.

Le schéma suivant montre comment tous les événements de gestion de l'espace sont enregistrés CloudTrail pour le compte de facturation, tandis que les connexions au compte/les événements de facturation et les événements liés aux AWS ressources sont enregistrés CloudTrail pour le compte correspondant.



Le diagramme suivant illustre les étapes suivantes :

1. Lorsqu'un espace est créé, un Compte AWS est connecté à l'espace et est désigné comme compte de facturation. Le parcours utilisé est celui qui a été créé CloudTrail pour le compte de facturation, où les événements spatiaux sont enregistrés. CloudTrail capture les appels d'API et les événements associés effectués par ou au nom d'un CodeCatalyst espace et envoie les fichiers journaux dans un compartiment S3 que vous spécifiez. Si le compte de facturation est remplacé par un autre compte AWS, les événements spatiaux sont enregistrés dans le dossier de suivi et dans le compartiment correspondant à ce compte. Pour plus d'informations sur les événements de CodeCatalyst gestion enregistrés CloudTrail, consultez [CodeCatalyst informations dans CloudTrail](#).
2. Les autres comptes connectés à l'espace, y compris le compte de facturation, enregistrent un sous-ensemble d'événements relatifs aux connexions aux comptes et aux événements de facturation. CodeCatalyst les flux de travail qui génèrent des événements de compte pour les AWS ressources déployées pour ce compte sont également enregistrés dans le journal et le compartiment du Compte AWS. CloudTrail capture les appels d'API et les événements associés effectués par ou au nom d'un CodeCatalyst espace et envoie les fichiers journaux dans un compartiment S3 que vous spécifiez. Pour plus d'informations sur les événements de CodeCatalyst gestion enregistrés CloudTrail, consultez [Accès aux événements enregistrés dans CodeCatalyst](#).

- Vous pouvez également surveiller CodeCatalyst les actions effectuées dans votre espace pendant un laps de temps précis à l'aide de la [list-event-logs](#) commande AWS CLI. Pour plus d'informations, consultez [le guide de référence des CodeCatalyst API Amazon](#). Vous devez avoir le rôle d'administrateur de l'espace pour consulter la liste des événements afin d'effectuer CodeCatalyst des actions dans votre espace. Pour plus d'informations, consultez [Accès aux événements enregistrés dans CodeCatalyst](#).

#### Note

ListEventLogsgarantit les événements des 30 derniers jours dans un espace donné. Vous pouvez également consulter et récupérer une liste des événements de gestion survenus au cours des 90 derniers jours CodeCatalyst dans la AWS CloudTrail console en consultant l'historique des événements ou en créant un suivi pour créer et maintenir un enregistrement des événements qui se sont étendus sur les 90 derniers jours. Pour plus d'informations, consultez les [sections Utilisation de l'historique des CloudTrail événements](#) et [Utilisation des CloudTrail traces](#).

#### Note

AWS les ressources déployées dans des comptes connectés pour les CodeCatalyst flux de travail ne sont pas enregistrées dans le cadre de la CloudTrail journalisation de l'CodeCatalyst espace. Par exemple, CodeCatalyst les ressources incluent un espace ou un projet. AWS les ressources incluent un service Amazon ECS ou une fonction Lambda. Vous devez configurer la CloudTrail journalisation séparément pour chaque Compte AWS endroit où les ressources sont déployées.

Voici un flux possible pour la surveillance des événements dans CodeCatalyst.

Mary Major est administratrice d'un CodeCatalyst espace et consulte tous les événements de gestion relatifs CodeCatalyst aux ressources au niveau de l'espace et au niveau du projet dans l'espace qui sont connectées. CloudTrail Voir par [CodeCatalyst informations dans CloudTrail](#) exemple les événements connectés CloudTrail.

Pour les ressources créées dans CodeCatalyst, telles que les environnements de développement, Mary consulte l'historique des événements dans le compte de facturation de l'espace et étudie les événements dans CodeCatalyst lesquels les environnements de développement ont été créés

par les membres du projet. L'événement fournit le type d'identité IAM du magasin d'identités et les informations d'identification pour l'ID de AWS générateur de l'utilisateur qui a créé l'environnement de développement. Pour les ressources créées AWS lors du déploiement par des flux de travail CodeCatalyst, telles qu'une fonction Lambda pour un déploiement sans serveur, le Compte AWS propriétaire peut consulter l'historique des événements du journal associé au compte distinct Compte AWS (qui est également un compte connecté à CodeCatalyst) pour l'action de déploiement du flux de travail.

Pour approfondir ses recherches, Mary peut également consulter les événements de toutes les CodeCatalyst API de l'espace à l'aide de la [list-event-logs](#) commande dans le AWS CLI.

## Rubriques

- [Enregistrement des appels CodeCatalyst d'API en mode connecté à Comptes AWS l'aide de AWS CloudTrail](#)
- [Accès aux événements enregistrés dans CodeCatalyst](#)

## Enregistrement des appels CodeCatalyst d'API en mode connecté à Comptes AWS l'aide de AWS CloudTrail

Amazon CodeCatalyst est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou unService AWS. CloudTrail capture les appels d'API effectués pour le compte d' CodeCatalyst in connected Comptes AWS sous forme d'événements. Si vous créez un suivi, vous pouvez activer la diffusion continue des CloudTrail événements vers un compartiment S3, y compris les événements pour CodeCatalyst. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements.

CodeCatalyst prend en charge la journalisation des actions suivantes sous forme d'événements dans les fichiers CloudTrail journaux :

- Les événements de gestion CodeCatalyst des espaces seront enregistrés sur Compte AWS le compte de facturation désigné pour l'espace. Pour en savoir plus, consultez [CodeCatalyst événements spatiaux](#).

**Note**

Les événements de données relatifs aux CodeCatalyst espaces sont accessibles à l'aide de la CLI, comme indiqué dans [Accès aux événements enregistrés dans CodeCatalyst](#).

- Les événements relatifs aux ressources utilisées dans les actions de CodeCatalyst flux de travail qui se produisent dans un environnement connecté Compte AWS seront enregistrés en tant qu'événements dans ce dernier Compte AWS. Pour en savoir plus, consultez [CodeCatalyst connexions aux comptes et événements de facturation](#).

**Important**

Plusieurs comptes peuvent être associés à un espace, mais la CloudTrail journalisation des événements dans les CodeCatalyst espaces et les projets ne s'applique qu'au compte de facturation.

Le compte de facturation de l'espace est le vôtre, Compte AWS qui est débité pour les CodeCatalyst ressources au-delà du niveau AWS gratuit. Plusieurs comptes peuvent être connectés à un espace, tandis qu'un seul compte peut être désigné comme compte de facturation. Le compte de facturation ou les comptes connectés supplémentaires pour l'espace peuvent comporter des rôles IAM utilisés pour déployer des AWS ressources et une infrastructure, comme un cluster Amazon ECS ou un compartiment S3, à partir de CodeCatalyst flux de travail. Vous pouvez utiliser le flux de travail YAML pour identifier Compte AWS celui vers lequel vous avez effectué le déploiement.

**Note**

AWS Les ressources déployées dans des comptes connectés pour les CodeCatalyst flux de travail ne sont pas enregistrées dans le cadre de la CloudTrail journalisation de l' CodeCatalyst espace. Par exemple, CodeCatalyst les ressources incluent un espace ou un projet. AWS Les ressources incluent un service Amazon ECS ou une fonction Lambda. CloudTrail la journalisation doit être configurée séparément pour chaque Compte AWS endroit où les ressources sont déployées.

CodeCatalyst la connexion aux comptes connectés inclut les considérations suivantes :

- L'accès aux CloudTrail événements est géré avec IAM dans le compte connecté et non dans CodeCatalyst.
- Les connexions tierces, telles que les liens vers un GitHub référentiel, entraîneront l'enregistrement des noms de ressources tierces dans CloudTrail les journaux.

#### Note

CloudTrail la journalisation des CodeCatalyst événements se fait au niveau de l'espace et n'isole pas les événements en fonction des limites du projet.

Pour plus d'informations CloudTrail, consultez le [guide de AWS CloudTrail l'utilisateur](#).

#### Note

Cette section décrit la CloudTrail journalisation de tous les événements enregistrés dans un CodeCatalyst espace connecté et ceux Comptes AWS auxquels il est connecté CodeCatalyst. En outre, pour consulter tous les événements enregistrés dans un CodeCatalyst espace, vous pouvez également utiliser la `aws codecatalyst list-event-logs` commande AWS CLI et. Pour en savoir plus, consultez [Accès aux événements enregistrés dans CodeCatalyst](#).

## CodeCatalyst événements spatiaux

Les actions CodeCatalyst relatives à la gestion des ressources au niveau de l'espace et au niveau du projet sont enregistrées dans le compte de facturation de l'espace. Pour la CloudTrail journalisation d'un CodeCatalyst espace, les événements sont enregistrés en tenant compte des considérations suivantes.

- CloudTrail les événements s'appliquent à l'ensemble de l'espace et ne sont pas limités à un seul projet.
- Lorsque vous vous connectez Compte AWS à un CodeCatalyst espace, les événements enregistrables pour les connexions au compte y sont enregistrés. Compte AWS Une fois cette connexion activée, vous ne pouvez pas la désactiver.

- Lorsque vous connectez un Compte AWS à un CodeCatalyst espace et que vous le désignez comme compte de facturation pour l'espace, les événements y sont enregistrés. Une fois cette connexion activée, vous ne pouvez pas la désactiver.

Les événements relatifs aux ressources au niveau de l'espace et au niveau du projet sont enregistrés uniquement dans le compte de facturation. Pour modifier le compte de CloudTrail destination, mettez à jour le compte de facturation dans CodeCatalyst. Au début du cycle de facturation mensuel suivant, la modification prend effet pour le nouveau compte de facturation dans CodeCatalyst. Ensuite, le compte de CloudTrail destination est mis à jour.

Vous trouverez ci-dessous des exemples d'événements liés à AWS des actions de gestion des ressources au niveau CodeCatalyst de l'espace et au niveau du projet. Les API suivantes sont publiées via le SDK et la CLI. Les événements seront enregistrés sur le compte de facturation Compte AWS indiqué pour l' CodeCatalyst espace.

- [CreateDevEnvironment](#)
- [CreateProject](#)
- [DeleteDevEnvironment](#)
- [GetDevEnvironment](#)
- [GetProject](#)
- [GetSpace](#)
- [GetSubscription](#)
- [ListDevEnvironments](#)
- [ListDevEnvironmentSessions](#)
- [ListEventLogs](#)
- [ListProjects](#)
- [ListSourceRepositories](#)
- [StartDevEnvironment](#)
- [StartDevEnvironmentSession](#)
- [StopDevEnvironment](#)
- [StopDevEnvironmentSession](#)
- [UpdateDevEnvironment](#)

## CodeCatalyst connexions aux comptes et événements de facturation

Voici des exemples d'événements liés à AWS des actions liées à des connexions CodeCatalyst de comptes ou à la facturation :

- `AcceptConnection`
- `AssociateIAMRoletoConnection`
- `DeleteConnection`
- `DissassociateIAMRolefromConnection`
- `GetBillingAuthorization`
- `GetConnection`
- `GetPendingConnection`
- `ListConnections`
- `ListIAMRolesforConnection`
- `PutBillingAuthorization`
- `RejectConnection`

## CodeCatalyst informations dans CloudTrail

CloudTrail est activé sur et Compte AWS lorsque vous créez ce compte. Lorsque vous le connectez Compte AWS à un CodeCatalyst espace, les événements qui se produisent dans cet espace et qui Compte AWS sont connectés se CloudTrail connectent à ce compte AWS. Les événements enregistrables CodeCatalyst sont enregistrés sous forme d' CloudTrail événements dans CloudTrail les journaux du compte connecté et dans l'historique des événements de la CloudTrail console, ainsi que d'autres AWS événements enregistrables dans ce compte.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer :

- Si la demande a été faite par un utilisateur avec son identifiant AWS Builder.
- Si la demande a été effectuée avec les informations d'identification utilisateur racine ou AWS Identity and Access Management (IAM).
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.



- Si la requête a été effectuée par un autre service AWS.

Pour de plus amples informations, veuillez consulter l'[élément userIdentity CloudTrail](#) .

## Accès aux CloudTrail événements

Pour un enregistrement continu des événements survenus dans votre régionCompte AWS, y compris des événements liés aux CodeCatalyst activités qui s'y déroulentCompte AWS, créez un parcours. Un journal de suivi permet à CloudTrail de livrer les fichiers journaux dans un compartiment S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions Régions AWS. Le journal d'activité consigne les événements de toutes les régions dans la partition AWS et livre les fichiers journaux dans le compartiment S3 de votre choix. 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 les ressources suivantes :

- [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](#)

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

## Exemple d'événement de connexion à un CodeCatalyst compte dans AWS

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`ListConnections` action. Pour un Compte AWS utilisateur connecté à l'espace, `ListConnections` il est utilisé pour afficher toutes les connexions au compte CodeCatalyst à cet effetCompte AWS. L'événement sera enregistré à l'Compte AWSadresse spécifiée`accountId`, et la valeur du `arn` sera le Amazon Resource Name (ARN) du rôle utilisé pour l'action.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AKIAI44QH8DHBEXAMPLE",
    "arn": "role-ARN",
    "accountId": "account-ID",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AKIAI44QH8DHBEXAMPLE",
        "arn": "role-ARN",
        "accountId": "account-ID",
        "userName": "user-name"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-09-06T15:04:31Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2022-09-06T15:08:43Z",
  "eventSource": "account-ID",
  "eventName": "ListConnections",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.168.0.1",
  "userAgent": "aws-cli/1.18.147 Python/2.7.18 Linux/5.4.207-126.363.amzn2int.x86_64
botocore/1.18.6",
  "requestParameters": null,
  "responseElements": null,
  "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 ",
  "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 ",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "account-ID",
  "eventCategory": "Management"
}
```

## Exemple d'événement CodeCatalyst relatif aux ressources d'un projet dans AWS

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'CreateDevEnvironmentaction. Un Compte AWS compte connecté à l'espace et constituant le compte de facturation désigné pour l'espace est utilisé pour les événements au niveau du projet dans l'espace, tels que la création d'un environnement de développement.

SoususerIdentity, dans le accountId champ, il s'agit de l'ID de compte IAM Identity Center (432677196278) qui héberge le pool d'identités pour toutes les identités AWS Builder ID. Cet identifiant de compte contient les informations suivantes sur l' CodeCatalyst utilisateur associé à l'événement.

- Le type champ indique le type d'entité IAM pour la demande. Pour les CodeCatalyst événements relatifs à l'espace et aux ressources du projet, cette valeur estIdentityCenterUser. Le accountId champ indique le compte propriétaire de l'entité utilisée pour obtenir les informations d'identification.
- Le userId champ contient l'identifiant AWS Builder ID de l'utilisateur.
- Le identityStoreArn champ contient l'ARN du rôle pour le compte et l'utilisateur de la banque d'identités.

Le recipientAccountId champ contient l'identifiant du compte de facturation de l'espace, avec une valeur d'exemple ici de 111122223333.

Pour de plus amples informations, veuillez consulter l'[élément userIdentity CloudTrail](#).

```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "IdentityCenterUser",
    "accountId": "432677196278",
    "onBehalfOf": {
      "userId": "user-ID",
      "identityStoreArn": "arn:aws:identitystore::432677196278:identitystore/d-9067642ac7"
    }
  },
  "credentialId": "ABCDefGhiJKLMn11Lmn_1AbCDEFGHijk-AaBCdEFGHIjKLMnOPqrs11abEXAMPLE",
  "eventTime": "2023-05-18T17:10:50Z",
  "eventSource": "codecatalyst.amazonaws.com",
  "eventName": "CreateDevEnvironment",
  "awsRegion": "us-west-2",
```

```
"sourceIPAddress": "192.168.0.1",
"userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:102.0) Gecko/20100101
Firefox/102.0",
"requestParameters": {
  "spaceName": "MySpace",
  "projectName": "MyProject",
  "ides": [{
    "runtime": "public.ecr.aws/q6e8p2q0/cloud9-ide-runtime:2.5.1",
    "name": "Cloud9"
  }],
  "instanceType": "dev.standard1.small",
  "inactivityTimeoutMinutes": 15,
  "persistentStorage": {
    "sizeInGiB": 16
  }
},
"responseElements": {
  "spaceName": "MySpace",
  "projectName": "MyProject",
  "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 "
},
"requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"sharedEventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"eventCategory": "Management"
}
```

### Note

Dans certains cas, l'agent utilisateur peut ne pas être connu. Dans ce cas, CodeCatalyst fournira une valeur de Unknown dans le userAgent champ de l' CloudTrail événement.

## Consultation des parcours de vos CodeCatalyst événements

Vous pouvez créer et gérer des requêtes pour vos CloudTrail journaux à l'aide d'une table de requêtes dans Amazon Athena. Pour plus d'informations sur la création d'une requête, consultez la section [Interrogation des AWS CloudTrail journaux](#) dans le guide de l'utilisateur d'Amazon Athena.

## Accès aux événements enregistrés dans CodeCatalyst

Lorsque les utilisateurs effectuent des actions sur Amazon CodeCatalyst, celles-ci sont enregistrées en tant qu'événements. Vous pouvez utiliser le AWS CLI pour consulter les journaux des événements survenus dans un espace au cours d'une période spécifiée. Vous pouvez consulter ces événements pour consulter les actions entreprises dans l'espace, notamment la date et l'heure des actions, le nom de l'utilisateur qui a effectué l'action et l'adresse IP à laquelle l'utilisateur a fait la demande.

### Note

Les événements de gestion d'un CodeCatalyst espace sont enregistrés CloudTrail pour le compte de facturation connecté. Pour plus d'informations sur les événements de CodeCatalyst gestion enregistrés CloudTrail, consultez [CodeCatalyst informations dans CloudTrail](#).

Pour consulter le journal des événements d'un espace, vous devez l'avoir installé et configuré AWS CLI avec un profil pour CodeCatalyst, et vous devez avoir le rôle d'administrateur de l'espace pour cet espace. Pour plus d'informations, consultez [Configuration pour utiliser AWS CLI le CodeCatalyst](#) et [Rôle d'administrateur de l'espace](#).

### Note

Pour consulter la journalisation des événements qui se produisent pour le compte de CodeCatalyst in connected Comptes AWS, ou pour consulter la journalisation des événements concernant l'espace ou les ressources du projet dans le compte de facturation connecté, vous pouvez utiliser AWS CloudTrail. Pour plus d'informations, consultez [Enregistrement des appels CodeCatalyst d'API en mode connecté à Comptes AWS l'aide de AWS CloudTrail](#).

1. Ouvrez un terminal ou une ligne de commande et exécutez la `aws codecatalyst list-event-logs` commande en spécifiant :
  - Le nom de l'espace avec l'option `--space-name`.
  - Date et heure auxquelles vous souhaitez commencer à consulter les événements, au format d'horodatage en temps universel coordonné (UTC) tel que spécifié dans la [RFC 3339](#), avec l'option `--start-time`.

- La date et l'heure auxquelles vous souhaitez arrêter de consulter les événements, au format d'horodatage en temps universel coordonné (UTC) tel que spécifié dans la [RFC 3339](#), avec l'option. **--end-time**
- (Facultatif) Le nombre maximum de résultats à renvoyer dans une seule réponse, avec l'**--max-results** option. Si le nombre de résultats est supérieur au nombre que vous spécifiez, la réponse comprendra un `nextToken` élément que vous pourrez utiliser pour renvoyer les résultats suivants.
- (Facultatif) Limitez les résultats à un type d'événement spécifique que vous souhaitez renvoyer, avec l'**--event-name** option.

Cet exemple renvoie les événements enregistrés dans l'espace nommé *ExampleCorp* entre le *30 novembre et le 1er décembre* et *indique* qu'un maximum de *2* événements doivent être renvoyés dans la réponse.

```
aws codecatalyst list-event-logs --space-name ExampleCorp --start-time 2022-11-30
--end-time 2022-12-01 --event-name list-event-logs --max-results 2
```

2. Si des événements se sont produits pendant cette période, la commande renvoie des résultats similaires aux suivants :

```
{
  "nextToken": "EXAMPLE",
  "items": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "eventName": "listEventLogs",
      "eventType": "AwsApiCall",
      "eventCategory": "MANAGEMENT",
      "eventSource": "manage",
      "eventTime": "2022-12-01T22:47:24.605000+00:00",
      "operationType": "READONLY",
      "userIdentity": {
        "userType": "USER",
        "principalId": "a1b2c3d4e5-678fgh90-1a2b-3c4d-e5f6-EXAMPLE11111"
        "userName": "MaryMajor"
      },
      "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "requestPayload": {
        "contentType": "application/json",
```

```

        "data": "{\"spaceName\":\"ExampleCorp\",\"startTime\":
\"2022-12-01T00:00:00Z\",\"endTime\":\"2022-12-10T00:00:00Z\",\"maxResults\":
\"2\"}"
    },
    "sourceIpAddress": "127.0.0.1",
    "userAgent": "aws-cli/2.9.0 Python/3.9.11 Darwin/21.3.0 exe/x86_64
prompt/off command/codecatalyst.list-event-logs"
  },
  {
    "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaa",
    "eventName": "createProject",
    "eventType": "AwsApiCall",
    "eventCategory": "MANAGEMENT",
    "eventSource": "manage",
    "eventTime": "2022-12-01T09:15:32.068000+00:00",
    "operationType": "MUTATION",
    "userIdentity": {
      "userType": "USER",
      "principalId": "a1b2c3d4e5-678fgh90-1a2b-3c4d-e5f6-EXAMPLE11111",
      "userName": "MaryMajor"
    },
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "requestPayload": {
      "contentType": "application/json",
      "data": "{\"spaceName\":\"ExampleCorp\",\"name\":\"MyFirstProject
\",\"displayName\":\"MyFirstProject\"}"
    },
    "responsePayload": {
      "contentType": "application/json",
      "data": "{\"spaceName\":\"ExampleCorp\",\"name\":\"MyFirstProject
\",\"displayName\":\"MyFirstProject\",\"id\":\"a1b2c3d4-5678-90ab-cdef-
EXAMPLE4444\"}"
    },
    "sourceIpAddress": "192.0.2.23",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:102.0)
Gecko/20100101 Firefox/102.0"
  }
]
}

```

3. Exécutez à nouveau la `list-event-logs` commande avec l'`--next-token` option et la valeur du jeton renvoyé pour récupérer le prochain ensemble d'événements enregistrés correspondant à la demande.

## Quotas d'identité, d'autorisation et d'accès dans CodeCatalyst

Le tableau suivant décrit les quotas et les limites d'identité, d'autorisation et d'accès sur Amazon CodeCatalyst. Pour plus d'informations sur les quotas sur Amazon CodeCatalyst, consultez [Quotas pour CodeCatalyst](#).

| Ressource   | Informations   |
|---|--|
| Alias dans CodeCatalyst   | <p>Toute combinaison de caractères autorisés comprise entre 3 et 100 caractères doit commencer par une lettre. Caractères valides : A-Z, a-z et 0-9. Les alias ne peuvent pas :</p> <ul style="list-style-type: none"> <li>contenir moins de 3 caractères</li> <li>contenir des espaces ou l'un des caractères suivants : ? ^ * [ \ ~ :</li> </ul> |
| Nombre maximum d'invitations envoyées par un utilisateur par jour   | 500  |
| Nombre maximum d'invitations envoyées par jour à une adresse e-mail | 25   |
| Nombre maximum de jetons d'accès personnels (PAT) par utilisateur   | 100  |
| Mots de passe dans CodeCatalyst                                     | <p>Toute combinaison de caractères autorisés d'une longueur comprise entre 8 et 64 caractères. Caractères valides : A-Z, a-z et 0-9. Votre mot de passe peut inclure les caractères non alphanumériques suivants : ( ~ ! @ # \$ % ^ &amp; * _ - + = `   \ { } [ ] : ; " ' &lt; &gt; , . ? / )</p>  |
| Les noms PAT sont CodeCatalyst                                      | Toute combinaison de caractères autorisés comprise entre 1 et 100 caractères   |



| Ressource  | Informations                    |
|--|---------------------------------|
| Délai d'expiration de l'invitation d'un membre du projet         | Expire au bout de 24 heures     |
| Durée d'expiration de l'invitation d'un membre de l'espace       | Expire au bout de 24 heures     |
| Délai avant l'expiration de la vérification d'une adresse e-mail | Expire 10 minutes après l'envoi |

## Résolution des problèmes

Cette section peut vous aider à résoudre certains problèmes courants que vous pourriez rencontrer lors de l'accès à votre CodeCatalyst profil Amazon.

### Problèmes lors de l'inscription

Il se peut que vous rencontriez des problèmes lors de votre inscription. Nous avons des solutions.

#### Mon adresse e-mail est déjà utilisée

Si l'e-mail que vous avez saisi est déjà utilisé et que vous le reconnaissez comme étant le vôtre, il se peut que vous ayez déjà un profil chez nous. Connectez-vous avec cette identité existante. Si vous ne possédez pas l'e-mail existant, inscrivez-vous avec un autre e-mail non utilisé.

#### Je ne parviens pas à terminer la vérification par e-mail

Si vous n'avez pas reçu votre e-mail de vérification

1. Vérifiez votre dossier de spam, de courrier indésirable et d'éléments supprimés.

#### Note

Cet e-mail de vérification provient soit de l'adresse, `no-reply@signin.aws` soit `deno-reply@login.awsapps.com`. Nous vous recommandons de configurer votre système de messagerie de manière à ce qu'il accepte les e-mails provenant de ces adresses d'expéditeur et qu'il ne les traite pas comme du courrier indésirable ou du spam.

2. Patientez 5 minutes et actualisez votre boîte de réception. Vérifiez à nouveau votre dossier de spam, de courrier indésirable et d'éléments supprimés.
3. Si vous ne voyez toujours pas votre e-mail de vérification, choisissez Renvoyer le code. Si vous avez déjà quitté cette page, redémarrez votre flux de travail pour vous inscrire à [Amazon CodeCatalyst](#).

## Mon mot de passe ne répond pas aux exigences minimales

Pour votre sécurité, votre mot de passe doit comprendre 8 à 20 caractères, majuscules et minuscules, ainsi que des chiffres.

## Problèmes de connexion

### J'ai oublié mon mot de passe

Suivez les étapes de [J'ai oublié mon mot de passe](#).

### Mon mot de passe ne fonctionne pas

Vous devez respecter ces exigences chaque fois que vous définissez ou modifiez votre mot de passe :

- Les mots de passe distinguent majuscules et minuscules.
- Les mots de passe doivent comporter entre 8 et 64 caractères, avec des lettres majuscules et minuscules, des chiffres et au moins un caractère non alphanumérique.
- Vous ne pouvez pas réutiliser les trois derniers mots de passe .

### Je ne parviens pas à activer le MFA

Pour activer le MFA, ajoutez un ou plusieurs appareils MFA à votre profil en suivant les étapes décrites dans. [Authentification multifactorielle \(MFA\) sur Amazon CodeCatalyst](#)

### Je ne parviens pas à ajouter un appareil MFA

Si vous ne parvenez pas à ajouter un autre appareil MFA, il se peut que le nombre maximal d'appareils MFA que vous puissiez enregistrer soit atteint. Il se peut que vous deviez supprimer un appareil MFA existant avant d'en ajouter un nouveau.

## Je ne parviens pas à supprimer un appareil MFA

Si vous avez l'intention de désactiver le MFA, procédez au retrait de votre appareil MFA en suivant les étapes décrites dans [Supprimer un appareil MFA](#). Toutefois, si vous souhaitez que l'authentification MFA reste activée, vous devez ajouter un autre périphérique MFA avant de tenter de supprimer un périphérique MFA existant. Pour plus d'informations sur l'ajout d'un autre périphérique MFA, consultez [Comment enregistrer un appareil pour une utilisation avec l'authentification multifactorielle](#).

## Problèmes de déconnexion

### Je ne trouve pas où me déconnecter

Dans le coin supérieur droit de la page, choisissez **Se déconnecter**.

### Me déconnecter ne me déconnecte pas complètement

Le système est conçu pour vous déconnecter immédiatement, mais la déconnexion complète peut prendre jusqu'à une heure.

### Je reçois une erreur indiquant qu'un rôle n'existe pas suite à un échec du flux de travail

Problème : Après avoir créé un projet à partir de l'application Web ou du plan sans serveur, le flux de travail échoue avec l'erreur suivante :

CLIENT\_ERROR : Le rôle n'existe pas

Solution possible : Une fois que vous avez configuré un rôle IAM avec les autorisations nécessaires pour exécuter votre flux de travail et que vous avez ajouté le rôle IAM à votre flux de travail YAML, le flux de travail échoue toujours car le rôle IAM devra peut-être être ajouté à la connexion de votre compte. Ajoutez le rôle IAM à la connexion au compte pour votre espace, comme indiqué dans [Ajouter des rôles IAM aux connexions aux comptes](#).

### Je reçois une erreur de rôle suite à l'échec d'un flux de travail

Problème : Après avoir créé un projet à partir de l'application Web ou du plan sans serveur, le flux de travail échoue avec l'erreur suivante :

CLIENT\_ERROR : Le rôle n'est pas correctement configuré ou n'existe pas

Solution possible : L'espace dans lequel le projet a été créé peut avoir besoin de configurer une Compte AWS connexion ou de compléter une demande de connexion à un compte. Si votre espace dispose déjà d'une Compte AWS connexion active, créez et ajoutez un rôle IAM autorisé à exécuter des actions de flux de travail. Ajoutez le rôle IAM à la connexion de votre compte, comme indiqué dans [Ajouter des rôles IAM aux connexions aux comptes](#).

Solution possible : si le projet a été créé sans spécifier de connexion, la connexion au compte doit être associée à l'environnement de déploiement. Si votre espace dispose déjà d'une Compte AWS connexion active et qu'un rôle IAM a été ajouté, vous devez ajouter la connexion au compte avec le rôle IAM à votre environnement de déploiement, comme indiqué dans [Ajout de la connexion au compte et des rôles IAM à votre environnement de déploiement](#)

## Je dois mettre à jour le rôle IAM dans le flux de travail d'un projet

Si la Compte AWS connexion est complètement configurée et que le rôle IAM est créé et ajouté à la connexion au compte, vous pouvez mettre à jour le rôle IAM dans le flux de travail de votre projet.

1. Choisissez l'option CI/CD et choisissez votre flux de travail. Cliquez sur le bouton YAML.
2. Choisissez Modifier.
3. Dans le `ActionRoleArn` : champ, remplacez l'ARN du rôle IAM par l'ARN du rôle IAM mis à jour. Choisissez Valider.
4. Choisissez Commit (Valider).

Le flux de travail démarre automatiquement s'il s'agit de la branche principale. Sinon, pour réexécuter le flux de travail, choisissez Exécuter.

## Comment remplir un formulaire d'assistance ?

Vous pouvez vous rendre [sur Amazon CodeCatalyst](#) ou remplir un [formulaire de commentaires de support](#). Dans la section Demande d'informations, sous Comment pouvons-nous vous aider, indiquez que vous êtes un CodeCatalyst client Amazon. Fournissez le plus de détails possible afin que nous puissions résoudre votre problème le plus efficacement possible.

# Extensions dans CodeCatalyst

Amazon CodeCatalyst inclut des extensions qui vous aident à ajouter des fonctionnalités et à les intégrer à des produits extérieurs CodeCatalyst. Grâce aux extensions du CodeCatalyst catalogue, les équipes peuvent personnaliser leurs expériences dans CodeCatalyst.

## Rubriques

- [Concepts d'extensions](#)
- [Installation et désinstallation d'extensions dans CodeCatalyst](#)
- [Utilisation de GitHub référentiels dans CodeCatalyst](#)
- [Problèmes liés à l'utilisation de Jira dans CodeCatalyst](#)

## Concepts d'extensions

Voici quelques concepts et termes à connaître lorsque vous travaillez avec des extensions dans CodeCatalyst.

## Extensions

Une extension est un module complémentaire que vous pouvez installer dans votre CodeCatalyst espace afin d'ajouter de nouvelles fonctionnalités à vos projets et de les intégrer à des services extérieurs CodeCatalyst. Les extensions peuvent être consultées et installées à partir du CodeCatalyst catalogue.

## CodeCatalyst catalogue

Le CodeCatalyst catalogue est une liste centralisée de toutes les extensions disponibles dans CodeCatalyst. Vous pouvez parcourir le CodeCatalyst catalogue pour trouver des extensions susceptibles d'améliorer l'expérience de votre équipe dans des domaines CodeCatalyst tels que les sources, les flux de travail, etc.

## Installation et désinstallation d'extensions dans CodeCatalyst

Une extension est un module complémentaire que vous pouvez installer dans votre CodeCatalyst espace afin d'ajouter de nouvelles fonctionnalités à vos projets et de les intégrer à des services

extérieurs CodeCatalyst. Les extensions peuvent être consultées et installées à partir du CodeCatalyst catalogue.

## Rubriques

- [Installation d'une extension](#)
- [Désinstaller une extension](#)

## Installation d'une extension

Vous pouvez installer des extensions pour votre CodeCatalyst espace afin d'ajouter des fonctionnalités aux projets de cet espace. Vous pouvez consulter le CodeCatalyst catalogue en cliquant sur l'icône

Catalogue 

### Important

Pour installer une extension, vous devez être connecté avec un compte doté du rôle d'administrateur de l'espace dans l'espace.

Pour installer une extension depuis le CodeCatalyst catalogue

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.
3. Accédez au CodeCatalyst catalogue en cliquant sur l'icône Catalogue



dans le menu supérieur. Vous pouvez rechercher des extensions ou filtrer les extensions en fonction des catégories.

4. (Facultatif) Choisissez le nom de l'extension pour obtenir plus de détails sur l'extension, tels que les autorisations dont elle bénéficiera.
5. Choisissez Installer. Vérifiez les autorisations requises par l'extension, et si vous souhaitez continuer, sélectionnez à nouveau Installer.

Après avoir installé une extension, vous verrez la page de détails de l'extension installée. Parcourez les onglets pour plus d'informations sur l'extension. La page de détails est également l'endroit où vous pourrez effectuer une configuration supplémentaire de l'extension si nécessaire.


## Désinstaller une extension

Vous pouvez désinstaller les extensions précédemment installées dans votre CodeCatalyst espace. La désinstallation d'une extension peut supprimer des ressources associées à cette extension de votre CodeCatalyst espace ou de vos projets.

### Important

Pour désinstaller une extension, vous devez être connecté avec un compte doté du rôle d'administrateur de l'espace dans l'espace.

Pour désinstaller une extension de votre CodeCatalyst espace

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.
3. Procédez de l'une des manières suivantes pour afficher la liste des extensions installées pour votre espace :
  - a. Choisissez Paramètres, puis Extensions installées.
  - b. Cliquez sur l'icône Catalogue  dans le menu supérieur.
4. Choisissez Configurer sur l'extension que vous souhaitez désinstaller.
5. Choisissez Désinstaller sur la page de détails de l'extension.
6. Consultez les informations de la boîte de dialogue de désinstallation de l'extension. Suivez les instructions, puis choisissez Désinstaller pour désinstaller l'extension.

## Utilisation de GitHub référentiels dans CodeCatalyst

GitHub est un service basé sur le cloud qui aide les développeurs à stocker et à gérer leur code. L'extension GitHub Repositories vous permet d'utiliser des GitHub référentiels liés dans des projets

Amazon CodeCatalyst . Vous pouvez également lier des GitHub référentiels lors de la création d'un nouveau CodeCatalyst projet. Pour plus d'informations, consultez [Création d'un projet avec un GitHub référentiel lié](#).

#### Note

Vous ne pouvez pas utiliser de GitHub référentiels vides ou archivés avec des CodeCatalyst projets. L'extension GitHub Repositories n'est pas compatible avec GitHub Enterprise Server.

Une fois que vous aurez installé et configuré l'extension de GitHub référentiels, vous pourrez :

- Consultez vos GitHub référentiels dans la liste des référentiels sources dans CodeCatalyst
- Stockez et gérez les fichiers de définition des flux de travail dans vos GitHub référentiels.
- Création, lecture, mise à jour et suppression de fichiers stockés dans des GitHub référentiels liés à partir d'environnements de CodeCatalyst développement
- Le CodeCatalyst flux de travail de démarrage s'exécute automatiquement lorsque le code est transféré vers vos GitHub référentiels
- Utiliser les fichiers source de votre GitHub référentiel liés dans les CodeCatalyst flux de travail
- Lire et exécuter des GitHub actions dans les CodeCatalyst flux de travail

Les rubriques suivantes expliquent comment installer et configurer l'extension GitHub Repositories et comment utiliser vos GitHub référentiels une fois qu'ils sont liés à vos CodeCatalyst projets.

#### Rubriques

- [Démarrage rapide : utilisation de GitHub référentiels dans CodeCatalyst](#)
- [Gestion des GitHub comptes dans CodeCatalyst](#)
- [Gestion des GitHub référentiels dans CodeCatalyst](#)
- [Affichage des GitHub référentiels liés dans CodeCatalyst](#)
- [Utilisation de GitHub référentiels dans les flux de travail CodeCatalyst](#)
- [Utilisation des restrictions d'accès aux adresses IP dans GitHub Enterprise Cloud](#)
- [Blocage des fusions de GitHub pull request en cas d'échec des flux de travail](#)



## Démarrage rapide : utilisation de GitHub référentiels dans CodeCatalyst

Procédez comme suit pour installer l'extension GitHub Repositories, vous connecter à votre GitHub compte et lier vos GitHub référentiels à vos projets existants CodeCatalyst .

Vous pouvez également installer l'extension GitHub Repositories, vous connecter à votre GitHub compte, lier des GitHub référentiels lors de la création d'un nouveau CodeCatalyst projet. Pour plus d'informations, consultez [Création d'un projet avec un GitHub référentiel lié](#).

### Table des matières

- [Étape 1 : Installation de l' GitHub extension depuis le CodeCatalyst catalogue](#)
- [Étape 2 : Connectez votre GitHub compte à votre CodeCatalyst espace](#)
- [Étape 3 : lier votre GitHub dépôt à votre CodeCatalyst projet](#)
- [Étapes suivantes](#)

### Étape 1 : Installation de l' GitHub extension depuis le CodeCatalyst catalogue

La première étape de l'utilisation GitHub des référentiels dans CodeCatalyst consiste à installer l'extension des GitHub référentiels à partir du CodeCatalyst catalogue. Pour installer l'extension, effectuez les étapes suivantes en choisissant l'extension GitHub des référentiels.

#### Important

Dans le cadre de l'installation et de la configuration de l'extension GitHub Repositories, vous devez installer une extension dans votre GitHub compte. Pour cela, vous devez être à la fois administrateur du GitHub compte et administrateur de l' CodeCatalyst espace.

Pour installer une extension depuis le CodeCatalyst catalogue

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.
3. Accédez au CodeCatalyst CodeCatalyst catalogue en cliquant sur l'icône Catalogue



dans le menu supérieur. Vous pouvez rechercher des GitHub référentiels ou filtrer des extensions en fonction des catégories.

4. (Facultatif) Pour obtenir plus de détails sur l'extension, tels que les autorisations dont elle bénéficiera, choisissez le nom de l'extension GitHub des référentiels.
5. Choisissez Installer. Vérifiez les autorisations requises par l'extension, et si vous souhaitez continuer, sélectionnez à nouveau Installer.

Après avoir installé l'extension de GitHub référentiels, vous êtes redirigé vers la page de détails de l'extension de GitHub référentiels, où vous pouvez consulter et gérer les GitHub comptes connectés et les référentiels liés GitHub .

## Étape 2 : Connectez votre GitHub compte à votre CodeCatalyst espace

Après avoir installé l'extension GitHub Repositories, l'étape suivante consiste à connecter votre GitHub compte à votre CodeCatalyst espace.

### Important

Pour connecter votre GitHub compte à votre CodeCatalyst espace, vous devez être à la fois administrateur du GitHub compte et administrateur de l' CodeCatalyst espace.

Pour connecter votre GitHub compte à CodeCatalyst

1. Dans l'onglet GitHub Comptes connectés, choisissez Connect GitHub account pour accéder au site externe pour lequel GitHub.
2. Connectez-vous à votre GitHub compte à l'aide de vos GitHub informations d'identification, puis choisissez le compte sur lequel vous souhaitez installer Amazon CodeCatalyst.

### Tip

Si vous avez déjà connecté votre GitHub compte à un autre espace, il ne vous sera pas demandé de procéder à une nouvelle autorisation. Vous verrez plutôt une boîte de dialogue vous demandant où vous souhaitez installer l'extension si vous êtes membre ou collaborateur de plusieurs GitHub comptes, ou la page de configuration de l' CodeCatalyst application Amazon si vous n'appartenez qu'à un seul GitHub compte. Configurez l'application pour l'accès au référentiel que vous souhaitez autoriser, puis choisissez Enregistrer. Si le bouton Enregistrer n'est pas actif dans GitHub, modifiez la configuration, puis réessayez.

3. Choisissez si vous souhaitez autoriser l'accès CodeCatalyst à tous les référentiels actuels et futurs, ou choisissez les GitHub référentiels spécifiques dans lesquels vous souhaitez les utiliser. CodeCatalyst L'option par défaut est tous les GitHub référentiels du GitHub compte.
4. Vérifiez les autorisations accordées à CodeCatalyst, puis choisissez Installer.

Une fois connecté à votre GitHub compte à CodeCatalyst, vous pouvez consulter le compte connecté dans l'onglet GitHub Comptes de la page de détails de l'extension GitHub des référentiels.

### Étape 3 : lier votre GitHub dépôt à votre CodeCatalyst projet

La troisième et dernière étape de l'utilisation de vos GitHub référentiels CodeCatalyst consiste à lier le référentiel au CodeCatalyst projet dans lequel vous souhaitez l'utiliser.

#### Important

Bien que vous puissiez lier un GitHub dépôt en tant que contributeur, vous ne pouvez dissocier un GitHub référentiel qu'en tant qu'administrateur de l'espace ou administrateur du projet. Pour plus d'informations, consultez [Dissocier un GitHub dépôt d'un projet CodeCatalyst](#).

Pour lier un GitHub dépôt à un CodeCatalyst projet depuis la page de détails de l'extension du GitHub référentiel

1. Dans l'onglet GitHub Référentiels liés, choisissez Lier le GitHub référentiel.
2. Dans le menu déroulant du GitHub compte, choisissez le GitHub compte qui contient le référentiel que vous souhaitez associer.
3. Dans le menu déroulant du GitHub référentiel, choisissez le référentiel que vous souhaitez lier à un CodeCatalyst projet.

#### Tip

Si le nom du dépôt est grisé, vous ne pouvez pas lier ce dépôt car il a déjà été lié à un autre projet dans l'espace.

4. Dans le menu déroulant du CodeCatalyst projet, choisissez le CodeCatalyst projet auquel vous souhaitez lier le GitHub référentiel.
5. Choisissez Lier.

Vous pouvez également lier votre GitHub dépôt à un projet à partir des référentiels source dans Code. Pour plus d'informations, consultez [Gestion des GitHub référentiels dans CodeCatalyst](#).

## Étapes suivantes

Après avoir installé l'extension GitHub Repositories, connecté votre GitHub compte et lié vos GitHub référentiels à vos CodeCatalyst projets, vous pouvez l'utiliser dans les CodeCatalyst flux de travail et les environnements de développement. Pour plus d'informations, consultez [Utilisation de GitHub référentiels dans les flux de travail CodeCatalyst](#) et [Création d'un environnement de développement](#).

## Gestion des GitHub comptes dans CodeCatalyst

Pour utiliser GitHub les référentiels dans CodeCatalyst, vous devez connecter votre GitHub compte à votre CodeCatalyst espace. Si vous ne souhaitez plus utiliser les référentiels d'un compte dans CodeCatalyst, vous pouvez le GitHub déconnecter. Lorsqu'un compte est déconnecté, les événements présents dans les référentiels du compte ne démarrent pas les flux de travail et vous ne pourrez pas utiliser ces référentiels avec les environnements de CodeCatalyst développement.

### Important

Pour connecter ou déconnecter votre GitHub compte et votre CodeCatalyst espace, vous devez être à la fois administrateur du GitHub compte et administrateur de l' CodeCatalyst espace.

## Table des matières


- [Connecter un GitHub compte à un CodeCatalyst espace](#)
- [Connectez un GitHub compte à un CodeCatalyst espace lors de la création du projet](#)
- [Déconnecter un GitHub compte d'un CodeCatalyst espace](#)

## Connecter un GitHub compte à un CodeCatalyst espace

Procédez comme suit pour connecter un GitHub compte à votre CodeCatalyst espace.

Pour connecter votre GitHub compte à CodeCatalyst

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).

2. Accédez à votre CodeCatalyst espace.
3. Procédez de l'une des manières suivantes pour afficher l'extension de GitHub référentiels installée pour votre espace :
  - a. Choisissez Paramètres, puis Extensions installées.
  - b. Cliquez sur l'icône Catalogue  dans le menu supérieur.
4. Dans les GitHub référentiels, choisissez Configurer.
5. Dans l'onglet GitHub Comptes connectés, choisissez Connect GitHub account pour accéder au site externe pour lequel GitHub.
6. Connectez-vous à votre GitHub compte à l'aide de vos GitHub informations d'identification, puis choisissez le compte sur lequel vous souhaitez installer Amazon CodeCatalyst.

 Tip

Si vous avez déjà connecté un GitHub compte à l'espace, il ne vous sera pas demandé de procéder à une nouvelle autorisation. Vous verrez plutôt une boîte de dialogue vous demandant où vous souhaitez installer l'extension si vous êtes membre ou collaborateur de plusieurs GitHub espaces, ou la page de configuration de l' CodeCatalyst application Amazon si vous n'appartenez qu'à un seul GitHub espace. Configurez l'application pour l'accès au référentiel que vous souhaitez autoriser, puis choisissez Enregistrer. Si le bouton Enregistrer n'est pas actif, modifiez la configuration, puis réessayez.

7. Choisissez si vous souhaitez autoriser l'accès CodeCatalyst à tous les référentiels actuels et futurs, ou choisissez les GitHub référentiels spécifiques dans lesquels vous souhaitez les utiliser. CodeCatalyst L'option par défaut est tous les GitHub référentiels de l' GitHub espace.
8. Vérifiez les autorisations accordées à CodeCatalyst, puis choisissez Installer.

Après avoir installé l'extension de GitHub référentiels, vous êtes redirigé vers la page de détails de l'extension de GitHub référentiels, où vous pouvez consulter et gérer les GitHub comptes connectés et les référentiels liés GitHub .

## Connectez un GitHub compte à un CodeCatalyst espace lors de la création du projet

Vous pouvez connecter un GitHub compte à votre CodeCatalyst espace lors de la création d'un nouveau CodeCatalyst projet. Pour plus d'informations, consultez [Création d'un projet avec un GitHub référentiel lié](#).


## Déconnecter un GitHub compte d'un CodeCatalyst espace

Procédez comme suit pour déconnecter un GitHub compte de votre CodeCatalyst espace. Une fois qu'un compte est déconnecté, les événements présents dans les référentiels du compte ne démarrent pas les flux de travail et vous ne pourrez pas utiliser ces référentiels avec les environnements de CodeCatalyst développement.

### Note

Pour déconnecter un GitHub compte, vous devez d'abord dissocier tous les GitHub référentiels liés à ce compte. Pour plus d'informations, consultez [Dissocier un GitHub dépôt d'un projet CodeCatalyst](#).

Pour déconnecter un GitHub compte

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.
3. Procédez de l'une des manières suivantes pour afficher l'extension de GitHub référentiels installée pour votre CodeCatalyst espace :
  - a. Choisissez Paramètres, puis Extensions installées.
  - b. Cliquez sur l'icône Catalogue  dans le menu supérieur.
4. Dans les GitHub référentiels, choisissez Configurer.
5. Dans l'onglet GitHub des comptes, choisissez le GitHub compte que vous souhaitez déconnecter.
6. Choisissez Déconnecter le GitHub compte.
7. Dans la boîte de dialogue de déconnexion, examinez les effets de la déconnexion du compte.
8. Entrez déconnecter dans le champ de saisie de texte, puis choisissez Déconnecter.

## Gestion des GitHub référentiels dans CodeCatalyst

Pour utiliser GitHub des référentiels dans CodeCatalyst, vous devez d'abord lier les référentiels à votre CodeCatalyst projet. Avant de pouvoir lier un GitHub dépôt, vous devez connecter le GitHub compte auquel appartient le dépôt à votre CodeCatalyst espace. Pour plus d'informations, consultez [Connecter un GitHub compte à un CodeCatalyst espace](#).

Si vous ne souhaitez plus utiliser le GitHub référentiel dans CodeCatalyst, vous pouvez le dissocier d'un CodeCatalyst projet. Lorsqu'un référentiel est dissocié, les événements de ce référentiel ne démarrent pas les exécutions de flux de travail et vous ne pourrez pas utiliser ce référentiel avec les environnements de CodeCatalyst développement.

### Table des matières

- [Lier un GitHub référentiel à un CodeCatalyst projet](#)
- [Lier un GitHub référentiel à un CodeCatalyst projet lors de la création du projet](#)
- [Dissocier un GitHub dépôt d'un projet CodeCatalyst](#)

### Lier un GitHub référentiel à un CodeCatalyst projet


Vous pouvez utiliser GitHub des référentiels liés dans les flux de travail, où les événements du GitHub référentiel lié démarrent des flux de travail susceptibles de générer, de tester ou de déployer du code, en fonction de la configuration du flux de travail. Les fichiers de configuration des flux de travail qui utilisent GitHub des référentiels liés sont stockés dans le GitHub référentiel lié. GitHub Les référentiels liés peuvent également être utilisés avec les environnements de développement pour créer, mettre à jour et supprimer des fichiers dans le GitHub référentiel lié. Vous pouvez lier un GitHub référentiel à un CodeCatalyst projet depuis la page de détails de l'extension des GitHub référentiels ou depuis la vue Référentiels source dans Code dans le projet lui-même.

#### Note

Un GitHub référentiel ne peut être lié qu'à un seul CodeCatalyst projet dans un espace. Vous ne pouvez pas lier un dépôt portant le même nom qu'un autre référentiel du projet.

Pour lier un GitHub dépôt à un CodeCatalyst projet depuis la page de détails de l'extension du GitHub référentiel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).

2. Accédez à votre CodeCatalyst espace.
3. Procédez de l'une des manières suivantes pour afficher l'extension de GitHub référentiels installée pour votre espace :
  - a. Choisissez Paramètres, puis Extensions installées.
  - b. Cliquez sur l'icône Catalogue  dans le menu supérieur.
4. Dans les GitHub référentiels, choisissez Configurer.
5. Dans l'onglet GitHub Référentiels liés, choisissez Lier le GitHub référentiel.
6. Dans la liste déroulante des GitHub comptes, choisissez le GitHub compte qui contient le référentiel que vous souhaitez associer.
7. Dans le menu déroulant du GitHub référentiel, choisissez le référentiel que vous souhaitez lier à un CodeCatalyst projet.


 Tip

Si le nom du dépôt est grisé, vous ne pouvez pas lier ce dépôt car il a déjà été lié à un autre projet dans l'espace.

8. (Facultatif) Si aucun GitHub référentiel ne figure dans la liste des référentiels, il se peut qu'il n'ait pas été configuré pour l'accès au référentiel dans l' CodeCatalyst application Amazon dans GitHub. Vous pouvez configurer les GitHub référentiels dans CodeCatalyst lesquels le compte connecté peut être utilisé.
  - a. Accédez à votre [GitHub](#) compte, sélectionnez Paramètres, puis Applications.
  - b. Dans l'onglet GitHub Applications installées, choisissez Configurer pour l' CodeCatalyst application Amazon.
  - c. Procédez de l'une des manières suivantes pour configurer l'accès aux GitHub référentiels auxquels vous souhaitez créer un lien : CodeCatalyst
    - Pour fournir un accès à tous les référentiels actuels et futurs, choisissez Tous les référentiels.



- Pour fournir un accès à des référentiels spécifiques, choisissez Ne sélectionner que les référentiels, choisissez le menu déroulant Sélectionner les référentiels, puis choisissez un référentiel que vous souhaitez autoriser à créer des liens. CodeCatalyst

 Note

- Vous ne pouvez pas utiliser de GitHub référentiels vides ou archivés avec des CodeCatalyst projets.
- Vous ne pouvez pas lier un GitHub dépôt portant le même nom qu'un dépôt d'un CodeCatalyst projet.
- L'extension GitHub Repositories n'est pas compatible avec les référentiels GitHub Enterprise Server.

9. Dans le menu déroulant CodeCatalyst du projet, choisissez le CodeCatalyst projet auquel vous souhaitez lier le GitHub référentiel.
10. Choisissez Lier.

Pour lier un GitHub référentiel à un CodeCatalyst projet depuis la page des référentiels sources d'un projet

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst projet.
3. Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources.
4. Choisissez Ajouter un référentiel, puis choisissez Lier le référentiel.
5. Dans le menu déroulant du fournisseur de référentiel, sélectionnez GitHub.
6. Dans le menu déroulant du GitHub compte, choisissez le GitHub compte qui contient le référentiel que vous souhaitez associer.
7. Dans le menu déroulant du GitHub référentiel, choisissez le GitHub référentiel auquel vous souhaitez associer votre CodeCatalyst projet.

**i** Tip

Si le nom du dépôt est grisé, vous ne pouvez pas lier ce dépôt car il a déjà été lié à un autre projet sur Amazon CodeCatalyst.

8. (Facultatif) Si aucun GitHub référentiel n'apparaît dans la liste déroulante des GitHub référentiels, il se peut qu'il n'ait pas été configuré pour l'accès au référentiel dans l'application spatiale dans GitHub. Vous pouvez configurer les GitHub référentiels dans CodeCatalyst lesquels le compte connecté peut être utilisé.
  - a. Accédez à votre [GitHub](#) compte, sélectionnez Paramètres, puis Applications.
  - b. Dans l'onglet GitHub Applications installées, choisissez Configurer pour l' application Amazon.
  - c. Procédez de l'une des manières suivantes pour configurer l'accès aux GitHub référentiels auxquels vous souhaitez créer un lien : CodeCatalyst
    - Pour fournir un accès à tous les référentiels actuels et futurs, choisissez Tous les référentiels.
    - Pour fournir un accès à des référentiels spécifiques, choisissez Ne sélectionner que les référentiels, choisissez le menu déroulant Sélectionner les référentiels, puis choisissez un référentiel que vous souhaitez autoriser à créer des liens. CodeCatalyst

**i** Note

- Vous ne pouvez pas utiliser de GitHub référentiels vides ou archivés avec des CodeCatalyst projets.
- Vous ne pouvez pas lier un GitHub dépôt portant le même nom qu'un dépôt d'un CodeCatalyst projet.
- L'extension GitHub Repositories n'est pas compatible avec les référentiels GitHub Enterprise Server.

9. Choisissez Lier.

## Lier un GitHub référentiel à un CodeCatalyst projet lors de la création du projet

Vous pouvez lier un GitHub référentiel à un nouveau CodeCatalyst projet lors de la création du nouveau CodeCatalyst projet. Pour plus d'informations, consultez [Création d'un projet avec un GitHub référentiel lié](#).


## Dissocier un GitHub dépôt d'un projet CodeCatalyst

La dissociation d'un dépôt ne supprime pas le GitHub dépôt et n'y apporte aucune modification. Il ne supprime aucun fichier de configuration de flux de travail stocké dans ce référentiel lié. Cependant, une fois que vous avez dissocié un GitHub référentiel, les événements de ce référentiel ne démarrent plus les exécutions de flux de travail et vous ne pouvez pas utiliser le référentiel avec des environnements de développement.

### Important

Pour dissocier votre GitHub dépôt de votre CodeCatalyst projet, vous devez être l'administrateur de l'espace ou l'administrateur du projet.

Pour dissocier un dépôt GitHub

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.
3. Procédez de l'une des manières suivantes pour afficher l'extension de GitHub référentiels installée pour votre espace :
  - a. Choisissez Paramètres, puis Extensions installées.
  - b. Cliquez sur l'icône Catalogue  dans le menu supérieur.
4. Dans les GitHub référentiels, choisissez Configurer.
5. Dans l'onglet GitHub Référentiels liés, choisissez le GitHub référentiel que vous souhaitez dissocier.
6. Choisissez Dissocier le GitHub référentiel.
7. Dans la boîte de dialogue Dissocier, examinez les effets de la dissociation du référentiel.
8. Entrez Dissocier dans le champ de saisie de texte et choisissez Dissocier.

## Affichage des GitHub référentiels liés dans CodeCatalyst

Vous pouvez consulter les GitHub référentiels liés dans la liste des référentiels sources de votre projet ou sur la page de détails de l'extension GitHub des référentiels. Le fait de les sélectionner dans la liste des référentiels ne les ouvre pas. CodeCatalyst Au lieu de cela GitHub, ils s'ouvrent, où vous pouvez consulter et travailler sur le code dans le référentiel lié.

Pour afficher les GitHub référentiels liés dans CodeCatalyst

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst projet.
3. Dans le volet de navigation, choisissez Code, puis sélectionnez Référentiels sources.

Pour afficher les GitHub référentiels liés depuis vous, rendez-vous sur la page de détails de l'extension

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace, puis choisissez l'onglet Extensions installées.
3. Dans les GitHub référentiels, choisissez Configurer.
4. Choisissez l'onglet GitHub Référentiels liés pour afficher tous les GitHub référentiels connectés aux CodeCatalyst projets de votre CodeCatalyst espace.

Les GitHub référentiels liés à votre projet apparaissent dans la liste. Choisissez le GitHub référentiel dans lequel vous souhaitez afficher et modifier les fichiers GitHub.

### Note

Si un flux de travail utilise un GitHub référentiel dans une action source, les modifications que vous apportez au flux de travail YAML dans l'éditeur visuel ou dans l'éditeur YAML CodeCatalyst seront automatiquement validées et transmises au GitHub référentiel.

## Utilisation de GitHub référentiels dans les flux de travail CodeCatalyst

Vous pouvez utiliser un GitHub référentiel lié comme source d'un flux de travail, dans lequel les modifications apportées à une branche spécifiée d'un GitHub référentiel lié démarrent automatiquement l'exécution d'un flux de travail.

Un flux de travail est une procédure automatisée qui décrit comment créer, tester et déployer votre code dans le cadre d'un système d'intégration et de livraison continues (CI/CD). Un flux de travail définit une série d'étapes, ou d'actions, à effectuer lors de l'exécution d'un flux de travail. Un flux de travail définit également les événements, ou déclencheurs, qui déclenchent le démarrage du flux de travail. Pour configurer un flux de travail, vous devez créer un fichier de définition de flux de travail à l'aide de l'[éditeur visuel ou YAML](#) de la CodeCatalyst console.

### Tip

Pour un aperçu rapide de la manière dont vous pouvez utiliser les flux de travail dans un projet, [créez un projet avec un plan](#). Chaque plan déploie un flux de travail fonctionnel que vous pouvez examiner, exécuter et tester.

Lorsque vous configurez un flux de travail pour utiliser un GitHub référentiel lié, le fichier de configuration du flux de travail est stocké dans ce GitHub référentiel. La configuration du flux de travail est un fichier YAML qui définit le nom du flux de travail, les déclencheurs, les ressources, les artefacts et les actions. Pour plus d'informations sur le fichier de configuration du flux de travail, consultez [Référence de définition du flux de travail](#).

Le fichier de configuration du flux de travail doit se trouver dans le `./codecatalyst/workflows/` répertoire de votre GitHub référentiel.

Vous pouvez utiliser l'éditeur de flux de travail pour créer et configurer des flux de travail. Pour plus d'informations, consultez [Commencer à utiliser les flux de travail dans CodeCatalyst](#) et [Travailler avec des sources](#).

## Lancer automatiquement l'exécution d'un flux de travail après les événements GitHub du référentiel

Vous pouvez configurer un CodeCatalyst flux de travail pour démarrer automatiquement une exécution lorsque le code est transféré vers la branche spécifiée de votre GitHub référentiel. Pour démarrer l'exécution automatique d'un flux de travail, ajoutez un déclencheur dans la `Triggers` section du fichier de configuration du flux de travail.

Exemple : un simple déclencheur push à code

L'exemple suivant montre un déclencheur qui lance l'exécution d'un flux de travail chaque fois que du code est envoyé à une branche de votre référentiel source.

```
Triggers:  
- Type: PUSH
```

Exemple : un simple déclencheur de pull request

L'exemple suivant montre un déclencheur qui lance l'exécution d'un flux de travail chaque fois qu'une pull request est créée contre une branche de votre référentiel source.

```
Triggers:  
- Type: PULLREQUEST  
Events:  
- OPEN
```

Pour plus d'informations, voir [Utilisation de déclencheurs](#).

## Utilisation des restrictions d'accès aux adresses IP dans GitHub Enterprise Cloud

L'extension Amazon CodeCatalyst GitHub Repositories est compatible avec les [restrictions d'accès IP GitHub d'Enterprise Cloud](#). Lorsque vous configurez une organisation GitHub Enterprise Cloud pour restreindre l'accès à des adresses IP spécifiques, vous pouvez également [autoriser les GitHub applications à configurer la liste d'autorisation, qui permettra](#) d' CodeCatalyst enregistrer automatiquement ses adresses IP auprès de celle-ci GitHub. Vous pouvez également [ajouter les adresses CodeCatalyst IP manuellement](#). Pour plus d'informations, consultez [Adresses IP utilisées par l'extension GitHub des référentiels](#).

Si les adresses CodeCatalyst IP ne figurent pas dans la liste des adresses autorisées d'une GitHub organisation, l' CodeCatalyst GitHub application Amazon ne pourra pas accéder à vos GitHub référentiels.

### Adresses IP utilisées par l'extension GitHub des référentiels

Les adresses IP suivantes sont utilisées par l'extension GitHub Repositories pour accéder aux ressources de votre GitHub organisation :

```
us-west-2  
52.32.242.246  
54.148.176.49
```

```
35.164.118.94
eu-west-1
34.241.64.10
34.246.255.80
3.248.38.7
```

## Blocage des fusions de GitHub pull request en cas d'échec des flux de travail

Après avoir lié un GitHub dépôt à CodeCatalyst, vous pouvez ajouter des CodeCatalyst flux de travail pour les pull requests. Une ou plusieurs exécutions de flux de travail peuvent avoir lieu lors d'un commit spécifique, et le statut d'exécution de chaque flux de travail CodeCatalyst est également reflété dans le statut de validation dans GitHub. Lorsqu'un nouveau commit est envoyé, les nouveaux [statuts d'exécution](#) du flux de travail sont reflétés GitHub pour ce nouveau commit. Si vous réexécutez un flux de travail pour une validation, le nouveau statut d'exécution du flux de travail remplace le statut précédent de cette validation et de ce flux de travail.

Vous pouvez définir des règles de protection des branches GitHub pour bloquer une fusion par pull request lorsque le dernier commit présente le statut d'échec de l'exécution du flux de travail. Avec les règles de protection des branches, le statut du dernier commit affecte la capacité à fusionner une pull request GitHub. Pour plus d'informations, consultez GitHub la documentation [À propos des vérifications de statut](#) et [À propos des branches protégées](#). Pour en savoir plus sur les flux de travail, consultez [Travailler avec des courses](#) et [Utilisation de déclencheurs](#).

## Problèmes liés à l'utilisation de Jira dans CodeCatalyst

Jira est une application logicielle qui aide les équipes de développement agiles à planifier, attribuer, suivre, signaler et gérer le travail. L'extension Jira Software vous permet d'utiliser des projets Jira dans des projets Amazon CodeCatalyst .

### Note

CodeCatalyst est uniquement compatible avec Jira Software Cloud.

Une fois que vous aurez installé et configuré l'extension Jira Software pour un CodeCatalyst projet Amazon, vous pourrez :

1. Accédez aux projets Jira depuis CodeCatalyst en les liant à des CodeCatalyst projets
2. Problèmes liés à la mise à jour de Jira avec les CodeCatalyst pull requests
3. Afficher le statut et le flux de travail des CodeCatalyst pull requests liées dans les problèmes liés à Jira

Les rubriques suivantes expliquent comment installer et configurer l'extension Jira Software pour utiliser Jira dans CodeCatalyst

## Rubriques

- [Démarrage rapide : problèmes liés à l'utilisation de Jira dans CodeCatalyst](#)
- [Gestion des sites Jira dans CodeCatalyst](#)
- [Gérer les projets Jira dans CodeCatalyst](#)
- [Lier un problème Jira à une CodeCatalyst pull request](#)
- [Afficher CodeCatalyst des événements dans Jira](#)
- [Recherche de problèmes liés à Jira dans CodeCatalyst](#)

## Démarrage rapide : problèmes liés à l'utilisation de Jira dans CodeCatalyst

Procédez comme suit pour installer l'extension Jira Software, connecter vos sites Jira à vos CodeCatalyst espaces et lier vos projets Jira à vos projets CodeCatalyst

### Table des matières

- [Étape 1 : Installation de l'extension logicielle Jira depuis le catalogue CodeCatalyst](#)
- [Étape 2 : Connectez vos sites Jira à votre espace CodeCatalyst](#)
- [Étape 3 : lier votre projet Jira à votre CodeCatalyst projet](#)
- [Étape 4 : Lier vos problèmes Jira à une CodeCatalyst pull request](#)
- [Étapes suivantes](#)

## Étape 1 : Installation de l'extension logicielle Jira depuis le catalogue CodeCatalyst


La première étape pour gérer Jira in CodeCatalyst consiste à installer l'extension Jira Software depuis le CodeCatalyst catalogue. Pour installer l'extension, effectuez les étapes suivantes en choisissant l'extension Jira Software.



**⚠ Important**

Pour installer et configurer l'extension Jira Software, vous devez disposer d'un compte Atlassian et d'un site Jira existant, mais un projet Jira peut être créé ultérieurement. Vous devez être l'administrateur du site Jira et l'administrateur de CodeCatalyst Space.

Pour installer une extension depuis le CodeCatalyst catalogue

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.
3. Accédez au CodeCatalyst CodeCatalyst catalogue en cliquant sur l'icône Catalogue  dans le menu supérieur. Vous pouvez rechercher Jira Software ou filtrer les extensions en fonction des catégories.
4. (Facultatif) Pour obtenir plus de détails sur l'extension, tels que les autorisations dont elle bénéficiera, choisissez le nom de l'extension Jira Software.
5. Choisissez Installer. Vérifiez les autorisations requises par l'extension, et si vous souhaitez continuer, sélectionnez à nouveau Installer.

Après avoir installé l'extension Jira Software, vous êtes redirigé vers la page de détails de l'extension Jira Software, où vous pouvez consulter et gérer les sites Jira connectés et les projets Jira liés.

## Étape 2 : Connectez vos sites Jira à votre espace CodeCatalyst

Après avoir installé l'extension Jira Software, l'étape suivante consiste à connecter votre site Jira à votre CodeCatalyst espace.

**⚠ Important**

Pour connecter votre site Jira à votre CodeCatalyst espace, vous devez être à la fois administrateur du site Jira et administrateur de l' CodeCatalyst espace.

## Pour connecter votre site Jira à CodeCatalyst

1. Dans l'onglet Connected Jira sites, choisissez Connect Jira site pour accéder au site externe d'Atlassian Marketplace.
2. Choisissez Télécharger maintenant pour commencer l'installation CodeCatalyst sur votre site Jira.
3. En fonction de votre rôle, effectuez l'une des opérations suivantes :
  1. Si vous êtes administrateur de site Jira, dans le menu déroulant du site, choisissez le site Jira pour installer l' CodeCatalyst application, puis choisissez Installer l'application.

### Note

Si vous avez un site Jira, cette étape n'apparaîtra pas et vous serez automatiquement redirigé vers l'étape suivante.

2. a. Si vous n'êtes pas administrateur Jira, dans le menu déroulant du site, choisissez le site Jira sur lequel installer l' CodeCatalyst application, puis sélectionnez Request app. Pour plus d'informations sur l'installation des applications Jira, voir [Qui peut installer des applications ?](#) .
    - b. Entrez la raison pour laquelle vous devez effectuer l'installation CodeCatalyst dans le champ de saisie ou conserver le texte par défaut, puis choisissez Soumettre la demande.
4. Passez en revue les actions effectuées au CodeCatalyst moment de l'installation de l'application, puis choisissez Get it now.
5. Une fois l'application installée, choisissez Return CodeCatalyst to pour revenir à CodeCatalyst.

Après avoir connecté votre site Jira à CodeCatalyst, vous pouvez consulter le site connecté dans l'onglet Sites Jira connectés de la page de détails de l'extension Jira Software.

## Étape 3 : lier votre projet Jira à votre CodeCatalyst projet

La troisième étape de la gestion de votre projet Jira CodeCatalyst consiste à lier le projet Jira au CodeCatalyst projet dans lequel vous souhaitez l'utiliser.

**Note**

Un CodeCatalyst projet ne peut être lié qu'à un seul projet Jira. Un projet Jira peut être lié à plusieurs CodeCatalyst projets.

**Important**

Pour lier votre projet Jira à votre CodeCatalyst projet, vous devez être l'administrateur de l'CodeCatalyst espace ou l'administrateur CodeCatalyst du projet.

Pour lier un projet Jira à un CodeCatalyst projet depuis la page de détails de l'extension Jira Software

1. Dans l'onglet Projets Jira liés, choisissez Lier un projet Jira.
2. Dans le menu déroulant du site Jira, choisissez le site Jira contenant le projet que vous souhaitez lier.
3. Dans le menu déroulant du projet Jira, choisissez le projet que vous souhaitez associer à un CodeCatalyst projet.
4. Dans le menu déroulant CodeCatalyst du projet, choisissez le CodeCatalyst projet que vous souhaitez associer à un projet Jira.
5. Choisissez Lier.

Une fois qu'un projet Jira est lié à un CodeCatalyst projet, l'accès aux CodeCatalyst problèmes est complètement désactivé et les problèmes du volet de CodeCatalyst navigation sont remplacés par un élément relatif aux problèmes Jira renvoyant au projet Jira.

## Étape 4 : Lier vos problèmes Jira à une CodeCatalyst pull request

Après avoir lié un projet Jira à un CodeCatalyst projet, vous pouvez lier CodeCatalyst les problèmes en créant une pull request et en les affichant en tant que propriété de la pull request.

**Note**

Vous ne pouvez pas créer de pull request sans un référentiel source comportant deux branches dans votre CodeCatalyst projet. Pour plus d'informations sur les pull requests, consultez la section [Utilisation des pull requests dans CodeCatalyst](#).

## Pour associer les problèmes liés à Jira à une CodeCatalyst pull request

1. Dans le volet de navigation, choisissez Code, puis choisissez Pull requests.
2. Choisissez Créer une pull request pour saisir les détails de la pull request.
3. Dans le menu déroulant Référentiel source, choisissez le référentiel source dans lequel vous souhaitez lier une pull request.
4. Dans le menu déroulant Branche source, choisissez la branche qui contient les modifications que vous souhaitez vérifier.
5. Dans le menu déroulant Branche de destination, choisissez la branche dans laquelle vous souhaitez fusionner les modifications révisées.
6. Dans le champ de saisie du texte du titre de la pull request, saisissez le titre de vos pull requests.
7. Choisissez Lier les problèmes pour les problèmes Jira - champ facultatif, choisissez le menu déroulant et recherchez les problèmes Jira que vous souhaitez ajouter à partir du projet Jira associé.
8. Sélectionnez les problèmes Jira que vous souhaitez ajouter à la pull request.
9. Choisissez Create pour créer la pull request.

Le statut récapitulatif de la pull request et le statut des événements de CodeCatalyst flux de travail associés sont reflétés dans votre problème Jira.

## Étapes suivantes

Après avoir installé l'extension Jira Software, connecté votre site Jira, lié vos projets Jira à votre CodeCatalyst projet et lié une pull request, les mises à jour CodeCatalyst sont reflétées dans votre projet Jira. Pour plus d'informations sur l'affichage CodeCatalyst des événements dans Jira, consultez [Afficher CodeCatalyst des événements dans Jira](#).

## Gestion des sites Jira dans CodeCatalyst

Pour gérer les projets Jira dans CodeCatalyst, vous devez connecter vos sites Jira à votre CodeCatalyst espace. Si vous ne souhaitez plus utiliser les projets d'un site Jira dans CodeCatalyst, vous pouvez le déconnecter. Lorsqu'un site Jira est déconnecté, les problèmes de Jira issus des projets du site ne seront pas disponibles dans les CodeCatalyst projets, et CodeCatalyst Issues redeviendra le fournisseur de problèmes.

**⚠ Important**

Pour connecter ou déconnecter votre site Jira et votre CodeCatalyst espace, vous devez être à la fois administrateur du site Jira et administrateur de CodeCatalyst Space.


**Table des matières**

- [Connecter un site Jira à un espace CodeCatalyst](#)
- [Déconnexion d'un site Jira d'un espace CodeCatalyst](#)

**Connecter un site Jira à un espace CodeCatalyst**


Procédez comme suit pour connecter un site Jira à votre CodeCatalyst espace.

Pour connecter votre site Jira à CodeCatalyst

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.
3. Procédez de l'une des manières suivantes pour afficher l'extension Jira Software installée pour votre espace :
  - a. Choisissez Paramètres, puis Extensions installées.
  - b. Cliquez sur l'icône Catalogue  dans le menu supérieur.
4. Dans Jira Software, choisissez Configurer.
5. Dans l'onglet Connected Jira sites, choisissez Connect Jira site pour accéder au site externe d'Atlassian Marketplace.
6. Choisissez Télécharger maintenant pour commencer l'installation CodeCatalyst sur votre site Jira.

**📘 Note**

Si vous l'avez déjà installé sur CodeCatalyst votre site Jira, vous en serez averti. Choisissez Commencer pour passer à l'étape finale.

7. En fonction de votre rôle, effectuez l'une des opérations suivantes :
    1. Si vous êtes administrateur de site Jira, dans le menu déroulant du site, choisissez le site Jira pour installer l' CodeCatalyst application, puis choisissez Installer l'application.
-  **Note**

Si vous avez un site Jira, cette étape n'apparaîtra pas et vous serez automatiquement redirigé vers l'étape suivante.
2. a. Si vous n'êtes pas administrateur Jira, dans le menu déroulant du site, choisissez le site Jira sur lequel installer l' CodeCatalyst application, puis sélectionnez Request app. Pour plus d'informations sur l'installation des applications Jira, voir [Qui peut installer des applications ?](#) .
    - b. Entrez la raison pour laquelle vous devez effectuer l'installation CodeCatalyst dans le champ de saisie ou conserver le texte par défaut, puis choisissez Soumettre la demande.
  8. Passez en revue les actions effectuées au CodeCatalyst moment de l'installation de l'application, puis choisissez Get it now.
  9. Une fois l'application installée, choisissez Return CodeCatalyst to pour revenir à CodeCatalyst.

Après avoir connecté votre site Jira à CodeCatalyst, vous pouvez consulter le site connecté dans l'onglet Sites Jira connectés de la page de détails de l'extension Jira Software.

## Déconnexion d'un site Jira d'un espace CodeCatalyst


Procédez comme suit pour déconnecter un site Jira de votre CodeCatalyst espace.

### Note

Pour déconnecter un site Jira, vous devez d'abord dissocier tous les projets Jira liés à ce compte. Pour plus d'informations, consultez [Dissocier un projet Jira d'un projet CodeCatalyst](#).

Pour déconnecter un site Jira

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.

3. Procédez de l'une des manières suivantes pour afficher l'extension Jira Software installée pour votre espace :
  - a. Choisissez Paramètres, puis Extensions installées.
  - b. Cliquez sur l'icône Catalogue  dans le menu supérieur.
4. Dans Jira Software, choisissez Configurer.
5. Dans l'onglet Sites Jira connectés, sélectionnez le site Jira que vous souhaitez déconnecter.
6. Choisissez Déconnecter le site Jira.
7. Dans la boîte de dialogue Déconnexion, examinez les effets de la déconnexion du site.
8. Entrez déconnecter dans le champ de saisie de texte, puis choisissez Déconnecter.

## Gérer les projets Jira dans CodeCatalyst

Pour gérer un projet Jira dans CodeCatalyst, vous devez d'abord lier le projet Jira à votre CodeCatalyst projet. Vous pouvez ensuite ajouter et utiliser des problèmes Jira à une CodeCatalyst pull request dans un CodeCatalyst projet. Avant de pouvoir lier un projet Jira, vous devez connecter le site Jira auquel appartient le projet à votre CodeCatalyst espace. Pour plus d'informations, consultez [Connecter un site Jira à un espace CodeCatalyst](#).

Si vous ne souhaitez plus utiliser un projet Jira CodeCatalyst, vous pouvez le dissocier de votre CodeCatalyst projet. Lorsqu'un projet Jira est dissocié, les problèmes Jira ne seront pas disponibles dans le CodeCatalyst projet et Issues redeviendra le fournisseur CodeCatalyst des problèmes.

### Important

Pour lier votre projet Jira à votre CodeCatalyst projet, vous devez être l'administrateur de l'CodeCatalyst espace ou l'administrateur CodeCatalyst du projet.

## Table des matières

- [Lier un projet Jira à un CodeCatalyst projet](#)
- [Dissocier un projet Jira d'un projet CodeCatalyst](#)


## Lier un projet Jira à un CodeCatalyst projet

Vous pouvez utiliser des projets Jira liés pour gérer les problèmes et lier les CodeCatalyst pull requests à un problème Jira. Le statut récapitulatif d'une pull request et le statut des événements de CodeCatalyst flux de travail associés sont reflétés dans votre problème Jira.

### Note

Un CodeCatalyst projet ne peut être lié qu'à un seul projet Jira. Un projet Jira peut être lié à plusieurs CodeCatalyst projets.

Pour lier un projet Jira à un CodeCatalyst projet depuis la page de détails de l'extension Jira Software

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.
3. Procédez de l'une des manières suivantes pour afficher l'extension Jira Software installée pour votre espace :
  - a. Choisissez Paramètres, puis Extensions installées.
  - b. Cliquez sur l'icône Catalogue  dans le menu supérieur.
4. Dans Jira Software, choisissez Configurer.
5. Dans l'onglet Projets Jira liés, choisissez Lier un projet Jira.
6. Dans le menu déroulant du site Jira, choisissez le site Jira contenant le projet que vous souhaitez lier.
7. Dans le menu déroulant du projet Jira, choisissez le projet que vous souhaitez associer à un CodeCatalyst projet.
8. Dans le menu déroulant CodeCatalyst du projet, choisissez le CodeCatalyst projet que vous souhaitez associer à un projet Jira.
9. Choisissez Lier.




Une fois qu'un projet Jira est lié à un CodeCatalyst projet, l'accès aux CodeCatalyst problèmes est complètement désactivé et les problèmes du volet de CodeCatalyst navigation sont remplacés par un élément relatif aux problèmes Jira renvoyant au projet Jira.

## Dissocier un projet Jira d'un projet CodeCatalyst

La dissociation d'un projet ne supprime pas le projet Jira, y compris les éléments de planification ou les informations de développement, et n'y apporte aucune modification. Cependant, une fois que vous avez dissocié un projet Jira, les problèmes Jira du projet ne pourront plus être liés au projet CodeCatalyst


Pour dissocier un projet Jira

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.
3. Procédez de l'une des manières suivantes pour afficher l'extension Jira Software installée pour votre espace :
  - a. Choisissez Paramètres, puis Extensions installées.
  - b. Cliquez sur l'icône Catalogue  dans le menu supérieur.
4. Dans l'onglet Projets Jira liés, sélectionnez le projet Jira que vous souhaitez dissocier.
5. Choisissez Dissocier le projet Jira.
6. Dans la boîte de dialogue Dissocier, examinez les effets de la dissociation du référentiel.
7. Entrez dissocier dans le champ de saisie de texte et choisissez Dissocier.

## Lier un problème Jira à une CodeCatalyst pull request

Vous pouvez lier les pull requests créées dans un référentiel CodeCatalyst source aux problèmes liés à Jira. Après avoir lié un problème Jira, celui-ci est affiché en tant que propriété de la pull request. Par conséquent, les événements de pull request, les événements de flux de travail et les événements de déploiement sont envoyés à Jira et ajoutés au problème Jira. Les pull requests peuvent être liées à un ou plusieurs problèmes liés à Jira. Vous ne pouvez lier que les pull requests qui se trouvent dans un référentiel CodeCatalyst source, pas celles d'un référentiel tiers comme celui-ci GitHub. Avant de pouvoir lier les problèmes de Jira à une pull request, votre projet Jira doit être lié au

CodeCatalyst projet. Pour plus d'informations sur l'association d'un projet Jira à un CodeCatalyst projet, consultez [Gestion des sites Jira dans CodeCatalyst](#).

 Note

Vous ne pouvez pas créer de pull request sans un référentiel source comportant deux branches dans votre CodeCatalyst projet. Pour plus d'informations sur les pull requests, consultez la section [Utilisation des pull requests dans CodeCatalyst](#).

Pour associer les problèmes liés à Jira à une CodeCatalyst pull request

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst projet.
3. Dans le volet de navigation, choisissez Code, puis choisissez Pull requests.
4. Choisissez Créer une pull request pour saisir les détails de la pull request.
5. Dans le menu déroulant Référentiel source, choisissez le référentiel source dans lequel vous souhaitez lier une pull request.
6. Dans le menu déroulant Branche source, choisissez la branche qui contient les modifications que vous souhaitez vérifier.
7. Dans le menu déroulant Branche de destination, choisissez la branche dans laquelle vous souhaitez fusionner les modifications révisées.
8. Dans le champ de saisie du texte du titre de la pull request, saisissez le titre de vos pull requests.
9. Choisissez Lier les problèmes pour les problèmes Jira - champ facultatif, choisissez le menu déroulant et recherchez les problèmes Jira que vous souhaitez ajouter à partir du projet Jira associé.
10. Sélectionnez les problèmes Jira que vous souhaitez ajouter à la pull request.
11. Choisissez Create pour créer la pull request.

Une fois que vous avez lié les problèmes de Jira à une CodeCatalyst pull request, un résumé de la pull request est disponible. Le résumé inclut les exécutions du flux de travail, les problèmes liés, les réviseurs requis, les réviseurs facultatifs et l'auteur.

**Note**

Personne assignée et créée par les informations associées au problème Jira ne sont pas disponibles dans. CodeCatalyst

Après avoir lié une pull request, le CodeCatalyst projet synchronisé et le projet Jira permettent aux mises à jour de CodeCatalyst se refléter dans votre projet Jira. Le statut de la pull request liée et tous les événements du flux de travail liés à la pull request apparaîtront dans le numéro de Jira lors de sa consultation dans Jira. Pour plus d'informations sur l'affichage CodeCatalyst des événements dans Jira, consultez [Afficher CodeCatalyst des événements dans Jira](#).

## Afficher CodeCatalyst des événements dans Jira

Si vos CodeCatalyst projets et les projets Jira sont liés, le statut récapitulatif de la pull request et le statut des événements de CodeCatalyst flux de travail associés sont reflétés dans votre numéro Jira. Par exemple, si vous fermez ou fusionnez une pull request CodeCatalyst, la mise à jour du statut se reflète dans le numéro de Jira. CodeCatalyst les événements CI/CD du flux de travail liés à une CodeCatalyst pull request sont synchronisés, de sorte qu'une exécution réussie du flux de travail sera également envoyée au problème Jira.

Pour consulter les CodeCatalyst événements d'un numéro de Jira

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst projet.
3. Dans le volet de CodeCatalyst navigation, choisissez Code, choisissez Pull requests, puis choisissez la pull request contenant le problème Jira que vous souhaitez afficher dans votre projet Jira.
4. Dans le volet Informations supplémentaires, choisissez le problème Jira que vous souhaitez consulter dans votre projet Jira.
5. Dans le volet Détails du projet Jira, choisissez la pull request répertoriée pour Développement pour voir les détails de la pull request.
6. (Facultatif) Pour voir les dernières versions, choisissez l'onglet Builds.
7. (Facultatif) Pour voir l'état du développement, cliquez sur l'onglet Déploiements.

## Recherche de problèmes liés à Jira dans CodeCatalyst

Après avoir lié un projet Jira, vous pouvez rechercher des problèmes dans le projet Jira lié à l'aide de la barre de recherche CodeCatalyst globale. Vous pouvez également rechercher des problèmes liés à Jira en CodeCatalyst créant un lien vers les problèmes à partir d'une pull request. Pour plus d'informations sur le lien entre les problèmes Jira et une CodeCatalyst pull request, consultez [Lier un problème Jira à une CodeCatalyst pull request](#).

Pour rechercher des problèmes liés à Jira dans des projets Jira liés

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst projet.
3. Dans la barre de recherche globale, recherchez les problèmes d'un projet Jira associé ou les problèmes Jira que vous souhaitez associer à une pull request.

# Rechercher dans CodeCatalyst

Utilisez la barre de recherche ou une fenêtre de résultats de recherche dédiée CodeCatalyst pour effectuer une recherche dans le code, les problèmes, les projets et les utilisateurs CodeCatalyst.

Vous pouvez trouver des ressources dans votre espace et dans vos projets en saisissant des requêtes telles que le nom, la description et le statut dans la barre de recherche. Vous pouvez également affiner vos requêtes de recherche à l'aide de la langue des requêtes de recherche.

## Rubriques

- [Affiner votre requête de recherche](#)
- [Considérations relatives à l'utilisation de la recherche](#)
- [Référence aux champs consultables](#)

## Pour effectuer une recherche

1. Dans la barre de recherche située dans la barre de navigation supérieure, entrez une requête de recherche.
2. (Facultatif) Affinez votre requête de recherche à l'aide CodeCatalyst de la langue de requête de recherche. Pour en savoir plus, consultez [Affiner votre requête de recherche](#).
3. Effectuez l'une des actions suivantes :
  - Pour rechercher des ressources dans le projet dans lequel vous vous trouvez actuellement, choisissez Ce projet.
  - Pour rechercher des ressources dans tous les projets de l'espace dans lequel vous vous trouvez actuellement, choisissez Cet espace.
4. Affichez les résultats de recherche dans une fenêtre de résultats de recherche dédiée en effectuant l'une des opérations suivantes :
  - Au bas de la fenêtre des résultats de recherche rapide, choisissez Afficher tous les résultats dans nom-projet | nom-espace pour afficher tous les résultats de recherche.
  - Appuyez sur Entrée pour afficher tous les résultats de recherche.

 Tip

Mentionnez les autres utilisateurs du projet dans un commentaire ou une description d'une pull request, ou dans un commentaire ou une description d'un problème, en utilisant le signe @ suivi de leur nom d'affichage ou de leur nom d'utilisateur. Vous pouvez également créer un lien vers des ressources telles que des problèmes ou des fichiers de code en utilisant le signe @ suivi du nom du problème ou du fichier de code.

## Affiner votre requête de recherche

Si vous ne trouvez pas ce que vous recherchez après avoir effectué une recherche, vous pouvez affiner votre recherche à CodeCatalyst l'aide du langage de requête spécialisé. Les champs individuels n'ont pas de limite de caractères, mais la requête globale est limitée à 1 024 caractères.

### Rubriques

- [Affinage par type](#)
- [Affinage par domaine](#)
- [Affinage à l'aide d'opérateurs booléens](#)
- [Affinage par projet](#)

## Affinage par type

Pour affiner la portée de votre recherche en fonction d'un type d'information spécifique, incluez *type:result-type* dans votre recherche, où le *type de résultat* est `code`, `issueproject`, ou `user`.

### Exemples :

- `type:code AND java`— Affiche les résultats du code dans les champs liés au code contenant « java ».

Pour en savoir plus, consultez [Champs de code](#).

- `type:issue AND Bug`— Afficher les résultats du problème dans les champs relatifs au problème contenant le mot « Bug ».

Pour en savoir plus, consultez [Champs du problème](#).

- `type:user AND MaryMajor`— Affiche les résultats de l'utilisateur dans les champs relatifs à l'utilisateur qui contiennent « MaryMajor ».

Pour en savoir plus, consultez [Champs utilisateur](#).

- `type:project AND Datafeeder`— Affiche les résultats du projet contenant « Datafeeder ».

Pour en savoir plus, consultez [Domaines du projet](#).

## Affinage par domaine

Pour affiner la portée de votre recherche à un champ spécifique, incluez *field-name:query* dans votre recherche le texte que vous recherchez où le *nom du champ* est `title` `username` `projectdescription`,, etc., et la *requête* est le texte que vous recherchez. Pour obtenir la liste des champs, voir [Référence aux champs consultables](#). Vous pouvez rechercher plusieurs requêtes à l'aide de parenthèses.

Exemples :

- `title:bug`— Affiche les résultats dont le titre contient « bug ».
- `username:John`— Affiche les résultats dont le nom d'utilisateur contient « John ».
- `project:DataFeeder`— Afficher les résultats dans le projet « DataFeeder ». La requête ne distingue pas les majuscules et minuscules.
- `description:overview`— Afficher les résultats dont la description contient une « vue d'ensemble ».

## Affinage à l'aide d'opérateurs booléens

Pour définir des contraintes sur les phrases de recherche, vous pouvez utiliser les opérateurs booléens AND OR, et NOT Si vous listez plusieurs phrases, CodeCatalyst joignez-les OR par défaut. Vous pouvez regrouper les phrases de recherche à l'aide de parenthèses.

- `exception AND type:code`— Affiche uniquement les résultats du code pour « exception ».
- `path:README.md AND repo:ServerlessAPI`— Affiche les résultats pour les chemins avec « README.md » où le dépôt est nommé « ServerLessAPI ».
- `buildspec.yml AND (repo:ServerlessAPI OR ServerlessWebApp)`— Affiche les résultats pour « buildspec.yml » où le référentiel est « ServerLessAPI » ou « ». ServerlessWebApp

- `path:java NOT (path:py OR path:ts)`— Affiche les résultats lorsque le chemin contient « java » mais pas « py » ou « ts ».

## Affinage par projet

Pour affiner la portée de votre recherche en fonction d'un projet spécifique, incluez *project:name* *AND query* dans votre recherche le *nom* du projet dans lequel vous recherchez et la *requête* le contenu que vous recherchez.

- `project:name AND query`— Affiche les résultats dont le chemin contient la requête et le nom du projet.

## Considérations relatives à l'utilisation de la recherche

Mises à jour différées du contenu : plusieurs minutes peuvent être nécessaires pour que les mises à jour du contenu, telles que les changements de nom ou les réassignations de problèmes, apparaissent dans les résultats de recherche. Les mises à jour importantes, telles qu'une migration de base de code, peuvent mettre plus de temps à apparaître dans les résultats de recherche.

Échapper aux caractères spéciaux — Les caractères spéciaux suivants nécessitent une attention particulière dans vos requêtes de recherche : `+ - & & | | ! ( ) { } [ ] ^ " ~ * ? : \`. Les caractères spéciaux n'influenceront pas la requête, et vous devez soit les supprimer, soit y échapper. Pour échapper à un caractère, ajoutez une barre oblique inverse (`\`) devant celui-ci. Par exemple, la requête de recherche `[Feature]` doit être `Feature` ou `\ [Feature \]`.

Affiner la recherche : la recherche ne fait pas la distinction majuscules/minuscules. La recherche en minuscules empêche vos requêtes de diviser les mots en cas de changement de majuscules. Par exemple, pour effectuer une requête pour `MyService` et uniquement `MyService`, pensez à effectuer une requête `myservice` afin d'éviter les résultats contenant uniquement `my ouservice`.

Par défaut, la recherche associe des mots et des parties de mots avec une conjonction par OR. Par exemple, `new function` pourrait renvoyer des résultats contenant les deux `new function` et également des résultats contenant uniquement `new` ou `function`. Pour éviter ce dernier cas, combinez plusieurs mots avec AND. Par exemple, vous pouvez effectuer une recherche `new AND function`.

Branches par défaut : la recherche renvoie uniquement les résultats du code issu de la dernière validation sur la branche par défaut d'un dépôt source. Pour trouver du code sur d'autres branches



ou validations, pensez à [cloner le référentiel localement](#), à [ouvrir la branche dans un environnement de développement](#) ou à [consulter les branches et les détails dans l' CodeCatalyst interface utilisateur](#).

La modification de la branche par défaut entraîne la mise à jour des fichiers détectables par la recherche. Pour en savoir plus, consultez [Afficher et modifier la branche par défaut d'un référentiel](#).

## Référence aux champs consultables

CodeCatalyst effectue des recherches dans les champs suivants lorsque vous saisissez des requêtes de recherche. Les alias sont un autre nom que vous pouvez utiliser pour référencer le champ dans le langage de requête avancé.

### Champs de code

| Champ                | Alias   | Description  |
|----------------------|---------|--|
| Nom de la succursale | branche | Nom de la branche sur laquelle se trouve le fichier de code.   |
| code                 | N/A     | Informations sur le contenu du code sous forme d'extraits de code indiquant les parties du code source correspondant à la recherche.   |
| ID de commission     | N/A     | ID de validation du commit dans lequel le fichier de code renvoyé a été mis à jour pour la dernière fois. Il peut s'agir ou non de l'ID de validation situé en haut du nom de branche spécifié dans. <code>branchName</code> |
| Message d'engagement | N/A     | Message de validation du commit dans lequel le fichier de code a été mis à jour pour la dernière fois. Il peut   |

| Champ                        | Alias             | Description  |
|------------------------------|-------------------|--|
|                              |                   | s'agir ou non du message de validation situé au bout du nom de branche spécifié dans <code>branchName</code> . Si aucun message de validation n'a été fourni, cette valeur sera une chaîne vide.   |
| <code>filePath</code>        | <code>path</code> | Chemin de fichier de ce fichier de code.   |
| <code>lastUpdatedBy</code>   | N/A               | CodeCatalyst utilisateur ayant mis à jour le fichier de code pour la dernière fois. Si le nom d'utilisateur n'est pas disponible, cette valeur sera l'adresse e-mail de l'utilisateur telle que configurée dans le fichier de configuration Git. |
| <code>lastUpdatedById</code> | N/A               | ID unique généré par le système de l'utilisateur qui a mis à jour le fichier de code pour la dernière fois. Si le nom d'utilisateur n'est pas disponible, cette valeur peut être l'adresse e-mail de l'utilisateur.                              |
| <code>lastUpdatedTime</code> | N/A               | Heure à laquelle les données de recherche ont été mises à jour pour la dernière fois avec le commit contenant le fichier de code (horodatage en temps universel coordonné (UTC)).  |

| Champ              | Alias                    | Description   |
|--------------------|--------------------------|---|
| projectId          | N/A                      | ID unique du projet généré par le système.  |
| projectName        | Noms des projets, projet | Nom d'affichage du projet qui contient le référentiel source dans lequel le fichier de code a été validé. |
| ID du référentiel  | Repoid                   | ID unique du référentiel source généré par le système.  |
| Nom du référentiel | référentiel, dépôt       | Afficher le nom du référentiel source dans lequel le fichier de code a été validé.                        |

## Champs du problème

| Champ                          | Alias                       | Description   |
|--------------------------------|-----------------------------|---|
| Identifiants des destinataires | Identifiant du destinataire | Identifiants uniques générés par le système pour les utilisateurs affectés au problème. |
| cessionnaires                  | cessionnaire                | Noms d'utilisateur des utilisateurs affectés au problème.                               |
| Créé par                       | N/A                         | Afficher le nom de l'utilisateur qui a créé le problème.                                |
| createdById                    | N/A                         | ID unique généré par le système de l'utilisateur à l'origine du problème.               |

| Champ                    | Alias                      | Description  |
|--------------------------|----------------------------|--|
| Heure de création        | N/A                        | Heure à laquelle le problème a été créé (horodatage en temps universel coordonné (UTC)).                             |
| description              | N/A                        | Description du problème.   |
| est archivé              | archived                   | Valeur booléenne indiquant s'il faut créer le problème dans un état archivé.   |
| est bloqué               | blocked                    | Valeur booléenne qui indique si le problème est marqué comme bloqué.   |
| Identifiants d'étiquette | Identifiant de l'étiquette | Identifiants uniques générés par le système pour les étiquettes associées à un problème.                             |
| lastUpdatedBy            | N/A                        | Afficher le nom de l'utilisateur qui a mis à jour le problème pour la dernière fois.                                 |
| lastUpdatedById          | N/A                        | ID unique généré par le système de l'utilisateur qui a mis à jour le problème pour la dernière fois.                 |
| lastUpdatedTime          | N/A                        | Heure à laquelle le problème a été mis à jour pour la dernière fois (horodatage en temps universel coordonné (UTC)). |
| priority                 | N/A                        | Priorité du problème, si une priorité a été attribuée.   |

| Champ       | Alias                    | Description  |
|-------------|--------------------------|--|
| projectId   | N/A                      | ID unique du projet généré par le système.   |
| projectName | Noms des projets, projet | Projet dans lequel ce numéro peut être trouvé.   |
| ID court    | N/A                      | Identifiant abrégé et auto-incrémenté pour le problème.  |
| status      | N/A                      | État du problème qui indique si le problème est en attente ou s'il est inscrit dans une colonne. |
| ID d'état   | N/A                      | Identifiant système de l'état.   |
| title       | N/A                      | Titre du numéro.   |

## Domaines du projet

| Champ              | Alias   | Description  |
|--------------------|---------|--|
| description        | N/A     | Description du projet.   |
| lastUpdatedTime    | N/A     | Heure à laquelle les métadonnées du projet ont été mises à jour pour la dernière fois (horodatage en temps universel coordonné (UTC)). |
| projectName        | project | Nom du projet dans l'espace.   |
| Parcours du projet | N/A     | Nom URL routable du projet, défini lors de la création du projet. Utilisé dans les URL qui nécessitent le nom du projet.               |

## Champs utilisateur

| Champ           | Alias    | Description   |
|-----------------|----------|---|
| displayName     | N/A      | Nom utilisé pour l'utilisateur dans CodeCatalyst. Les noms d'affichage ne sont pas uniques.   |
| e-mail          | N/A      | Adresse e-mail de l'utilisateur.  |
| lastUpdatedTime | N/A      | Heure à laquelle les métadonnées de l'utilisateur ont été mises à jour pour la dernière fois (horodatage en temps universel coordonné (UTC)).                               |
| userName        | username | Nom d'utilisateur choisi par l'utilisateur lors de son inscription à CodeCatalyst. Contrairement aux noms d'affichage, les noms d'utilisateur ne peuvent pas être modifiés. |

# Résolution des problèmes liés à Amazon CodeCatalyst

Les informations suivantes peuvent vous aider à résoudre les problèmes courants dans CodeCatalyst. Vous pouvez également utiliser le rapport de CodeCatalyst santé d'Amazon pour déterminer s'il existe des problèmes de service susceptibles d'avoir un impact sur votre expérience.

## Rubriques

- [Résolution des problèmes d'accès généraux](#)
- [Résolution des problèmes de support](#)
- [Une partie ou la totalité d'Amazon CodeCatalyst n'est pas disponible](#)
- [Je ne parviens pas à créer un projet dans CodeCatalyst](#)
- [Je souhaite envoyer des commentaires dans CodeCatalyst](#)
- [Résolution des problèmes liés aux référentiels sources](#)
- [Projets et plans de résolution des problèmes](#)
- [Résolution des problèmes liés aux flux de travail](#)
- [Résolution des problèmes liés à la recherche dans CodeCatalyst](#)
- [Résolution des problèmes liés aux comptes associés à votre espace](#)
- [Résolution des problèmes liés aux environnements de développement](#)
- [Résolution des problèmes liés aux problèmes](#)
- [Résolution des problèmes entre Amazon CodeCatalyst et AWS les SDK ou AWS CLI](#)

## Résolution des problèmes d'accès généraux

### J'ai oublié mon mot de passe

Problème : j'ai oublié le mot de passe que j'utilise pour mon AWS Builder ID et Amazon CodeCatalyst.

Solutions possibles : le moyen le plus simple de résoudre ce problème est de réinitialiser votre mot de passe.

1. Ouvrez [Amazon CodeCatalyst](#) et saisissez votre adresse e-mail. Choisissez ensuite Continuer (Continuer).

2. Choisissez Mot de passe oublié ?
3. Nous vous enverrons un e-mail contenant un lien vous permettant de modifier votre mot de passe. Si vous ne voyez pas l'e-mail dans votre boîte de réception, vérifiez votre dossier de courrier indésirable.

## Une partie ou la totalité d'Amazon CodeCatalyst n'est pas disponible

**Problème** : j'ai accédé à la CodeCatalyst console ou j'ai suivi un lien vers celle-ci, mais un message d'erreur s'affiche.

**Solutions possibles** : les raisons les plus courantes à l'origine de ce problème sont soit que vous avez suivi un lien vers un projet ou un espace auquel vous n'avez pas été invité, soit que le service présente un problème général de disponibilité. Consultez le [rapport Health](#) pour voir s'il existe des problèmes connus liés au service. Sinon, contactez la personne qui vous a invité au projet ou à l'espace et demandez une autre invitation. Si vous n'avez été invité à aucun projet ou espace, vous pouvez vous inscrire et [créer votre propre espace et vos propres projets](#).

## Je ne parviens pas à créer un projet dans CodeCatalyst

**Problème** : je souhaite créer un projet, mais le bouton Créer un projet s'affiche comme non disponible ou je reçois un message d'erreur.

**Correctifs possibles** : les raisons les plus courantes à l'origine de ce problème sont que vous êtes connecté à la console avec un AWS Builder ID qui ne possède pas le rôle d'administrateur de l'espace. Vous devez avoir ce rôle pour créer des projets dans un espace.

Si vous détenez ce rôle et que le bouton n'apparaît pas comme disponible, il se peut qu'il y ait un problème temporaire avec le service. Actualisez votre navigateur et réessayez.

## Résolution des problèmes de support

### Je reçois un message d'erreur lorsque j'AWS Support accède à Amazon CodeCatalyst

**Problème** : Lorsque je choisis l' option CodeCatalyst AWS Support pour Amazon, je reçois le message d'erreur suivant :



## Unable to assume role

To access support cases, you must add the role `AWSRoleForCodeCatalystSupport` to the Compte AWS that is the billing account for the space.

Corrections possibles : Ajoutez le Compte AWS rôle requis au compte de facturation de l'espace. Le compte désigné comme compte de facturation pour l'espace utilise le `AWSRoleForCodeCatalystSupport` rôle et la politique `AmazonCodeCatalystSupportAccess` gérée. Pour en savoir plus, consultez [Création du `AWSRoleForCodeCatalystSupport` rôle pour votre compte et votre espace](#).

### Note

Un AWS Builder ID ne peut être pris en charge que pour l'alias avec lequel il est authentifié et uniquement pour les ressources basées sur des CodeCatalyst autorisations. L'assistance relative aux comptes et à la facturation est disponible pour tous les utilisateurs de l'espace. Cependant, les constructeurs ne peuvent obtenir de l'aide que pour les ressources et les informations pour lesquelles ils sont autorisés à accéder CodeCatalyst.

## Je ne peux pas créer de dossiers de support technique pour mon espace

Problème : je n'arrive pas à créer de dossiers de support technique pour mon espace.

Corrections : un plan Business Support ou Enterprise Support doit être ajouté au compte de facturation de l'espace afin que les utilisateurs de l'espace puissent créer des dossiers de support technique. Demandez à votre administrateur d'espace d'ajouter un AWS Support forfait à votre compte de facturation d'espace ou rendez-vous sur <https://repost.aws/> pour demander à la AWS communauté.

## Mon compte pour les demandes d'assistance n'est plus connecté à mon espace dans CodeCatalyst

Problème : Mon compte pour les demandes d'assistance n'est plus connecté à mon espace dans CodeCatalyst.

Corrections : Si un utilisateur ayant le rôle d'administrateur de l'espace change de compte de facturation de l'espace, cela déconnectera le AWS Support plan et tous les dossiers associés de

l'espace. Les AWS Support dossiers associés à l'ancien compte de facturation des espaces ne seront plus visibles AWS Support pour Amazon CodeCatalyst. L'utilisateur root de ce compte de facturation peut consulter et résoudre les anciens dossiers à partir du AWS Management Console et peut configurer des autorisations IAM AWS Support permettant aux autres utilisateurs de consulter et de résoudre les anciens dossiers. Vous ne pourrez pas continuer à bénéficier d'une assistance technique CodeCatalyst depuis l'ancien compte de facturation des espaces via le AWS Management Console, mais vous pouvez bénéficier d'une assistance technique pour d'autres services jusqu'à ce que votre AWS Support plan soit annulé.

Pour plus d'informations, voir [Mettre à jour, résoudre et rouvrir votre dossier](#) dans le Guide de l'AWS Support utilisateur.

## Je ne parviens pas à ouvrir un dossier d'assistance pour un autre Service AWSAWS Support utilisateur sur Amazon CodeCatalyst

**Problème :** je n'arrive pas à ouvrir un dossier de support pour un autre Service AWS dans AWS Support for CodeCatalyst.

**Corrections possibles :** vous ne pouvez ouvrir des dossiers d'assistance CodeCatalyst qu'à partir de AWS Support pour CodeCatalyst. Si vous avez besoin d'assistance pour des services ou des ressources CodeCatalyst déployés depuis un autre service AWS, Amazon ou un autre service tiers, vous devrez créer un dossier par le biais du canal d'assistance du service AWS Management Console ou du service tiers. Pour plus d'informations, consultez les sections [Création de dossiers de support et gestion de dossiers](#) dans le Guide de l'AWS Support utilisateur.

## Une partie ou la totalité d'Amazon CodeCatalyst n'est pas disponible

**Problème :** j'ai accédé à la CodeCatalyst console ou j'ai suivi un lien vers celle-ci, mais un message d'erreur s'affiche.

**Solutions possibles :** les raisons les plus courantes à l'origine de ce problème sont soit que vous avez suivi un lien vers un projet ou un espace auquel vous n'avez pas été invité, soit que le service présente un problème général de disponibilité. Consultez le [rapport Health](#) pour voir s'il existe des problèmes connus liés au service. Sinon, contactez la personne qui vous a invité au projet ou à l'espace et demandez une autre invitation. Si vous n'avez été invité à aucun projet ou espace, vous pouvez vous inscrire et [créer votre propre espace et vos propres projets](#).

## Je ne parviens pas à créer un projet dans CodeCatalyst

Problème : je souhaite créer un projet, mais le bouton Créer un projet s'affiche comme non disponible ou je reçois un message d'erreur.

Correctifs possibles : les raisons les plus courantes à l'origine de ce problème sont que vous êtes connecté à la console avec un AWS Builder ID qui ne possède pas le rôle d'administrateur de l'espace. Vous devez avoir ce rôle pour créer des projets dans un espace.

Si vous détenez ce rôle et que le bouton n'apparaît pas comme disponible, il se peut qu'il y ait un problème temporaire avec le service. Actualisez votre navigateur et réessayez.

## Je souhaite envoyer des commentaires dans CodeCatalyst

Problème : j'ai trouvé un bogue CodeCatalyst et je souhaite envoyer un commentaire.

Corrections possibles : vous pouvez envoyer des commentaires directement dans CodeCatalyst.

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Dans le volet de navigation, choisissez Envoyer des commentaires.
3. Choisissez le type de commentaire dans le menu déroulant et saisissez votre commentaire.

## Résolution des problèmes liés aux référentiels sources

Les informations suivantes peuvent vous aider à résoudre les problèmes courants liés aux référentiels sources dans CodeCatalyst

### Rubriques

- [J'ai atteint la capacité de stockage maximale pour mon espace et je reçois des avertissements ou des erreurs](#)
- [Je reçois un message d'erreur lorsque j'essaie de cloner ou de transférer vers un référentiel CodeCatalyst source Amazon](#)
- [Je reçois un message d'erreur lorsque j'essaie de valider ou de transférer vers un référentiel CodeCatalyst source Amazon](#)
- [J'ai besoin d'un référentiel source pour mon projet](#)

- [Mon dépôt source est tout neuf mais contient un commit](#)
- [Je veux une branche différente comme branche par défaut](#)
- [Je reçois des e-mails concernant l'activité des pull requests](#)
- [J'ai oublié mon jeton d'accès personnel \(PAT\)](#)
- [Une pull request n'affiche pas les modifications que j'attends](#)
- [Une pull request indique le statut Non fusionnable](#)

## J'ai atteint la capacité de stockage maximale pour mon espace et je reçois des avertissements ou des erreurs

Problème : je souhaite valider du code dans un ou plusieurs référentiels sources CodeCatalyst, mais une erreur s'affiche. Dans la console, je vois un message sur la page du référentiel source indiquant que j'ai atteint la limite de stockage pour l'espace.

Corrections possibles : en fonction de votre rôle dans le projet ou dans l'espace, vous pouvez réduire la taille d'un ou de plusieurs de vos référentiels sources, supprimer les référentiels sources inutilisés ou modifier votre niveau de facturation pour un niveau offrant plus d'espace de stockage.

- Pour réduire la taille d'un référentiel source dans un projet, vous pouvez supprimer les branches inutilisées. Pour plus d'informations, consultez [Pour supprimer une branche \(console\)](#) et [Rôle du contributeur](#).
- Pour réduire l'espace de stockage global d'un espace, vous pouvez supprimer les référentiels sources inutilisés. Pour plus d'informations, consultez [Supprimer un référentiel source](#) et [Rôle d'administrateur de projet](#).
- Pour augmenter la quantité de stockage disponible pour votre espace, vous pouvez modifier votre niveau de facturation pour un niveau offrant plus d'espace de stockage. Pour plus d'informations, consultez [Modifier votre niveau CodeCatalyst de facturation](#) dans le manuel Amazon CodeCatalyst Administrator Guide.

## Je reçois un message d'erreur lorsque j'essaie de cloner ou de transférer vers un référentiel CodeCatalyst source Amazon

Problème : Lorsque j'essaie de cloner un référentiel source sur un ordinateur local ou dans un environnement de développement intégré (IDE), je reçois une erreur d'autorisation.

Correctifs possibles : vous n'avez peut-être pas de jeton d'accès personnel (PAT) pour votre identifiant AWS Builder, vous n'avez peut-être pas configuré votre système de gestion des informations d'identification avec votre PAT ou votre PAT a peut-être expiré. Essayez une ou plusieurs des solutions suivantes :

- Créez un jeton d'accès personnel (PAT). Pour plus d'informations, consultez [Gestion des jetons d'accès personnels sur Amazon CodeCatalyst](#).
- Assurez-vous d'avoir accepté une invitation à participer au projet qui contient le référentiel source et que vous êtes toujours membre de ce projet. Vous ne pouvez pas cloner un dépôt source si vous n'êtes pas un membre actif de ce projet. Connectez-vous à la console et essayez d'accéder à l'espace et au projet dans lesquels vous essayez de cloner un référentiel source. Si le projet ne figure pas dans la liste des projets de l'espace, cela signifie soit que vous n'êtes pas membre de ce projet, soit que vous n'avez pas accepté d'invitation à participer à ce projet. Pour plus d'informations, consultez [Acceptation d'une invitation et création de votre identifiant AWS Builder](#).
- Assurez-vous que votre commande de clonage est correctement formatée et inclut votre identifiant AWS Builder. Par exemple :

```
https://LiJuan@git.us-west-2.codecatalyst.aws/  
v1/ExampleCorp/MyExampleProject/MyExampleRepo
```

- Utilisez le AWS CLI pour vous assurer qu'un PAT est associé à votre identifiant AWS Builder et qu'il n'est pas expiré. Si vous n'en avez pas ou si le PAT a expiré, créez-en un. Pour plus d'informations, consultez [Gestion des jetons d'accès personnels sur Amazon CodeCatalyst](#).
- Essayez de créer un environnement de développement pour travailler avec le code du référentiel source au lieu de le cloner dans un dépôt local ou un IDE. Pour plus d'informations, consultez [Création d'un environnement de développement](#).

## Je reçois un message d'erreur lorsque j'essaie de valider ou de transférer vers un référentiel CodeCatalyst source Amazon

Problème : Lorsque j'essaie d'envoyer un message push vers un dépôt source, je reçois une erreur d'autorisation.

Corrections possibles : il se peut que vous n'ayez aucun rôle dans le projet qui vous permette de valider et d'appliquer des modifications de code au projet. Affichez votre rôle dans le projet dans lequel vous essayez d'appliquer des modifications à un référentiel source. Pour plus d'informations, consultez [Afficher les membres d'un projet](#) et [Utilisation des rôles sur Amazon CodeCatalyst](#).

Si vous avez un rôle qui permet de valider et de transférer des modifications, la branche dans laquelle vous essayez de valider les modifications peut avoir une règle de branche configurée pour elle qui vous empêche d'appliquer des modifications de code à cette branche. Essayez plutôt de créer une branche et de transférer votre code vers cette branche. Pour plus d'informations, consultez [Gérer les actions autorisées pour une branche à l'aide de règles de branche](#).

## J'ai besoin d'un référentiel source pour mon projet

Problème : mon projet n'a pas de référentiel source ou j'ai besoin d'un autre référentiel source pour mon projet.

Corrections possibles : Certains projets sont créés sans aucune ressource. Si vous êtes membre du projet, vous pouvez créer des référentiels sources pour ce projet dans CodeCatalyst. Si une personne ayant le rôle d'administrateur de l'espace installe les GitHub référentiels et les connecte à un GitHub compte, vous pouvez créer un lien vers les GitHub référentiels disponibles pour les ajouter à votre projet si vous détenez le rôle d'administrateur de projet. Pour plus d'informations, consultez les [sections Création d'un référentiel source](#) et [Liaison d'un référentiel source](#).

## Mon dépôt source est tout neuf mais contient un commit

Problème : je viens de créer un dépôt de sources. Il doit être vide, mais il contient un commit, une branche et un README .md fichier.

Corrections possibles : ce comportement est attendu. Tous les référentiels sources CodeCatalyst incluent un commit initial qui définit la branche par défaut sur `main` et inclut soit un exemple de code (si le référentiel a été créé pour un projet à l'aide d'un plan incluant un exemple de code), soit un modèle de fichier de balisage pour un fichier README du référentiel. Vous pouvez créer des branches supplémentaires dans la console et dans les clients Git. Vous pouvez créer et modifier des fichiers dans la console, et supprimer des fichiers dans les environnements de développement et les clients Git.

## Je veux une branche différente comme branche par défaut

Problème : mon dépôt source contient une branche par défaut nommée `main`, mais je veux une branche différente comme branche par défaut.

Correctifs possibles : vous ne pouvez pas modifier ou supprimer la branche par défaut dans les référentiels sources dans CodeCatalyst. Vous pouvez créer des branches supplémentaires et les

utiliser dans les actions source des flux de travail. Vous pouvez également choisir de lier GitHub des référentiels et de les utiliser comme référentiels pour votre projet.

## Je reçois des e-mails concernant l'activité des pull requests

Problème : je ne me suis pas inscrit ni configuré de notifications par e-mail concernant l'activité des pull requests, mais je les reçois quand même.

Corrections possibles : des notifications par e-mail sont envoyées automatiquement concernant l'activité des pull requests. Pour plus d'informations, consultez [Utilisation des pull requests sur Amazon CodeCatalyst](#).

## J'ai oublié mon jeton d'accès personnel (PAT)

Problème : j'utilise un PAT pour cloner, transférer et extraire du code pour les référentiels sources, mais j'ai perdu la valeur de mon jeton et je ne le trouve pas dans la CodeCatalyst console.

Corrections possibles : le moyen le plus rapide de résoudre ce problème est de créer un autre PAT et de configurer votre gestionnaire d'informations d'identification ou votre IDE pour utiliser ce nouveau PAT. Nous n'affichons la valeur d'un PAT que lorsque vous le créez. Si vous perdez cette valeur, elle ne peut pas être récupérée. Pour plus d'informations, consultez [Gestion des jetons d'accès personnels sur Amazon CodeCatalyst](#).

## Une pull request n'affiche pas les modifications que j'attends

Problème : j'ai créé une pull request, mais je ne vois pas les modifications que je m'attends à voir entre les branches source et destination.

Correctifs possibles : Cela peut être dû à un certain nombre de problèmes. Essayez une ou plusieurs des solutions suivantes :

- Il se peut que vous examiniez les modifications entre les anciennes révisions ou que vous ne visualisiez pas les dernières modifications. Actualisez votre navigateur et assurez-vous d'avoir choisi la comparaison entre les révisions que vous souhaitez consulter.
- Les modifications apportées à une pull request ne peuvent pas toutes être affichées dans la console. Par exemple, vous ne pouvez pas afficher les sous-modules Git dans la console. Vous ne pouvez donc pas voir les différences entre les sous-modules dans une pull request. Certaines différences sont peut-être trop importantes pour être affichées. Pour plus d'informations, consultez [Quotas pour les référentiels sources dans CodeCatalyst](#) et [Affichage d'un fichier](#).

- Les pull requests affichent les différences entre la base de fusion et la révision que vous choisissez. Lorsque vous créez une pull request, la différence qui s'affiche pour vous est la différence entre l'extrémité de la branche source et la pointe de la branche de destination. Une fois la pull request créée, la différence affichée se situe entre la révision et sa base de fusion. La base de fusion est le commit qui se trouvait à l'extrémité de la branche de destination lorsque la révision a été créée. La base de fusion peut changer entre les révisions. Pour plus d'informations sur les différences et les bases de fusion dans Git, consultez [git-merge-base](#) la documentation Git.

## Une pull request indique le statut Non fusionnable

Problème : je souhaite fusionner une pull request, mais son statut indique « Non fusionnable ».

Corrections possibles : Cela peut être dû à un ou plusieurs problèmes :

- Tous les réviseurs requis pour votre pull request doivent approuver une pull request avant qu'elle ne puisse être fusionnée. Consultez la liste des réviseurs requis pour tous les réviseurs avec une icône en forme d'horloge à côté du nom. Une icône en forme d'horloge indique que le réviseur n'a pas approuvé la pull request.

### Note

Si un réviseur requis a été retiré de votre projet avant d'approuver la pull request, vous ne pouvez pas fusionner la pull request. Fermez la pull request et créez une nouvelle pull request.

- Il peut y avoir un conflit de fusion entre la branche source et la branche de destination. CodeCatalyst ne prend pas en charge toutes les stratégies et options de fusion Git possibles. Vous pouvez évaluer les branches pour détecter les conflits de fusion dans un environnement de développement ou cloner le référentiel et utiliser un IDE ou des outils Git pour rechercher et résoudre les conflits de fusion. Pour plus d'informations, voir [Fusion d'une pull request](#).

## Projets et plans de résolution des problèmes

Cette section peut vous aider à résoudre certains problèmes courants que vous pouvez rencontrer lors de l'utilisation de projets et de plans sur Amazon. CodeCatalyst



## API Java avec AWS Fargate plan de dépendances manquantes pour apache-maven-3.8.6

Problème : Pour un projet créé à partir de l'API Java avec AWS Fargate Blueprint, le flux de travail échoue avec une erreur due à des apache-maven-3.8.6 dépendances manquantes. Le flux de travail échoue avec une sortie similaire à l'exemple suivant :

```
Step 8/25 : RUN wget https://d1cdn.apache.org/maven/maven-3/3.8.6/binaries/apache-
maven-3.8.6-bin.tar.gz -P /tmp
---> Running in 1851ce6f4d1b
[91m--2023-03-10 01:24:55-- https://d1cdn.apache.org/maven/maven-3/3.8.6/binaries/
apache-maven-3.8.6-bin.tar.gz
[0m[91mResolving d1cdn.apache.org (d1cdn.apache.org)...
[0m[91m151.101.2.132, 2a04:4e42::644
Connecting to d1cdn.apache.org (d1cdn.apache.org)|151.101.2.132|:443...
[0m[91mconnected.
[0m[91mHTTP request sent, awaiting response... [0m[91m404 Not Found
2023-03-10 01:24:55 ERROR 404: Not Found.
[0mThe command '/bin/sh -c wget https://d1cdn.apache.org/maven/maven-3/3.8.6/binaries/
apache-maven-3.8.6-bin.tar.gz -P /tmp' returned a non-zero code: 8
[Container] 2023/03/10 01:24:55 Command failed with exit status 8
```

Solution : mettez à jour le fichier Dockerfile du plan en procédant comme suit.

1. Dans la barre de recherche, entrez apache-maven-3.8.6 pour localiser le fichier docker dans le projet créé avec l'API Java avec Blueprint. AWS Fargate
2. Mettez à jour le Dockerfile (/static-assets/app/Dockerfile) à utiliser maven:3.9.0-amazoncorretto-11 comme image de base et supprimez la dépendance au apache-maven-3.8.6 package.
3. (Recommandé) Nous recommandons également de mettre à jour la taille du tas de mémoire Maven à 6 Go.

Vous trouverez ci-dessous un exemple de Dockerfile.

```
FROM maven:3.9.0-amazoncorretto-11 AS builder

COPY ./pom.xml ./pom.xml
COPY src ./src/
```

```
ENV MAVEN_OPTS='-Xmx6g'  
  
RUN mvn -Dmaven.test.skip=true clean package  
  
FROM amazoncorretto:11-alpine  
  
COPY --from=builder target/CustomerService-0.0.1.jar CustomerService-0.0.1.jar  
EXPOSE 80  
CMD ["java","-jar","-Dspring.profiles.active=prod","/CustomerService-0.0.1.jar", "--server.port=80"]
```

## Le flux de travail moderne du plan d'application Web à trois niveaux OnPullRequest échoue en raison d'une erreur d'autorisation pour Amazon CodeGuru

Problème : Lorsque j'essaie d'exécuter un flux de travail pour mon projet, le flux de travail ne s'exécute pas et le message suivant s'affiche :

```
Failed at codeguru_codereview: The action failed during runtime. View the action's logs for more details.
```

Solution : L'une des causes possibles de cet échec d'action peut être due à des autorisations manquantes dans la politique de rôle IAM, lorsque votre version du rôle de service utilisée par CodeCatalyst in the connected ne dispose pas des autorisations requises pour que l'action codeguru\_codereview s'exécute correctement. Compte AWS Pour résoudre ce problème, soit le rôle de service doit être mis à jour avec les autorisations requises, soit vous devez remplacer le rôle de service utilisé pour le flux de travail par un rôle disposant des autorisations requises pour Amazon CodeGuru et Amazon CodeGuru Reviewer. En suivant les étapes suivantes, recherchez votre rôle et mettez à jour les autorisations de la politique de rôle afin de permettre au flux de travail de s'exécuter correctement.

### Note

Ces étapes s'appliquent aux flux de travail suivants dans CodeCatalyst :

- Le OnPullRequest flux de travail prévu pour les projets créés avec le plan d'application Web moderne à trois niveaux dans CodeCatalyst

- Des flux de travail ont été ajoutés aux projets CodeCatalyst avec des actions qui accèdent à Amazon CodeGuru ou Amazon CodeGuru Reviewer.

Chaque projet contient des flux de travail avec des actions qui utilisent un rôle et un environnement fournis par la Compte AWS personne connectée à votre projet dans CodeCatalyst. Le flux de travail avec les actions et leur politique désignée est stocké dans votre référentiel source dans le répertoire `/.codecatalyst/workflows`. Il n'est pas nécessaire de modifier le code YAML du flux de travail, sauf si vous ajoutez un nouvel ID de rôle au flux de travail existant. Pour plus d'informations sur les éléments du modèle YAML et le formatage, consultez [Référence de définition du flux de travail](#).

Voici les étapes de haut niveau à suivre pour modifier votre politique de rôle et vérifier le flux de travail YAML.

Pour référencer le nom de votre rôle dans le flux de travail YAML et mettre à jour la politique

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace. Accédez à votre projet.
3. Choisissez CI/CD, puis Workflows.
4. Choisissez le flux de travail intitulé OnPullRequest. Choisissez l'onglet Définition.
5. Dans le flux de travail YAML, dans le `RoLe` : champ situé sous l'action `codeguru_codereview`, notez le nom du rôle. Il s'agit du rôle associé à la politique que vous allez modifier dans IAM. L'exemple suivant montre le nom du rôle.

The screenshot displays the Amazon CodeCatalyst interface for a workflow named 'OnPullRequest'. The interface includes a navigation bar with 'CI/CD', 'Workflows', 'Environments', 'Compute', 'Secrets', and 'Change tracking'. Below the navigation bar, the workflow name 'OnPullRequest' is displayed along with the repository 'mysfitscvb63' and a test icon. The workflow status is 'Valid' and the latest run is 'Run-9ab13'. The workflow definition is shown in a code editor, detailing triggers, actions, and configuration.

```

1 Name: OnPullRequest
2 SchemaVersion: "1.0"
3 Triggers:
4   - Type: PULLREQUEST
5     Branches:
6       - main
7     Events:
8       - OPEN
9       - REVISION
10 Actions:
11   codeguru_codereview:
12     Identifier: aws/build@v1
13     Inputs:
14       Sources:
15         - WorkflowSource
16     Variables:
17       - Name: AWS_DEFAULT_REGION
18         Value: us-west-2
19     Outputs:
20       Artifacts:
21         - Name: codereview
22           Files:
23             - ./code-guru/*
24 Configuration:
25   Steps:
26     - Run: curl -OL https://github.com/aws/aws-codeguru-c
27     - Run: unzip aws-codeguru-cli.zip
28     - Run: export PATH=$PATH:./aws-codeguru-cli/bin
29     - Run: aws-codeguru-cli --root-dir ./src --no-prompt -
30 Environment:
31   Name: development
32 Connections:
33   - Name: connection-11-30
34     Role: CodeCatalystPreviewDevelopmentAdministrator-j
  
```

## 6. Effectuez l'une des actions suivantes :

- (Recommandé) Mettez à jour le rôle de service associé à votre projet avec les autorisations requises pour Amazon CodeGuru et Amazon CodeGuru Reviewer. Le rôle portera un nom `CodeCatalystWorkflowDevelopmentRole-spaceName` avec un identifiant unique ajouté. Pour plus d'informations sur le rôle et la politique de rôle, consultez [Comprendre le rôle CodeCatalystWorkflowDevelopmentRole-\*spaceName\* du service](#). Passez aux étapes suivantes pour mettre à jour la politique dans IAM.

### Note

Vous devez disposer d'un accès AWS administrateur Compte AWS au rôle et à la politique.

- Modifiez le rôle de service utilisé pour le flux de travail en un rôle disposant des autorisations requises pour Amazon CodeGuru et Amazon CodeGuru Reviewer ou créez un nouveau rôle avec les autorisations requises.
7. Connectez-vous à l'outil AWS Management Console, puis ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.

Dans la console IAM, recherchez le rôle indiqué à l'étape 5, par exemple `CodeCatalystPreviewDevelopmentRole`.

8. Dans le rôle indiqué à l'étape 5, modifiez la politique d'autorisation pour inclure les `codeguru:*` autorisations `codeguru-reviewer:*` et. Une fois ces autorisations ajoutées, la politique d'autorisation doit ressembler à la suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudformation:*",
        "lambda:*",
        "apigateway:*",
        "ecr:*",
        "ecs:*",
        "ssm:*",
        "codedeploy:*",
        "s3:*",
        "iam:DeleteRole",
        "iam:UpdateRole",
        "iam:Get*",
        "iam:TagRole",
        "iam:PassRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "iam:PutRolePolicy",
        "iam:CreatePolicy",
        "iam>DeletePolicy",
        "iam:CreatePolicyVersion",
        "iam>DeletePolicyVersion",
        "iam:PutRolePermissionsBoundary",
        "iam>DeleteRolePermissionsBoundary",
        "sts:AssumeRole",
      ]
    }
  ]
}
```

```
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeListeners",
        "elasticloadbalancing:ModifyListener",
        "elasticloadbalancing:DescribeRules",
        "elasticloadbalancing:ModifyRule",
        "cloudwatch:DescribeAlarms",
        "sns:Publish",
        "sns:ListTopics",
        "codeguru-reviewer:*",
        "codeguru:*"
    ],
    "Resource": "*",
    "Effect": "Allow"
}
]
```

9. Après avoir apporté les corrections de politique, revenez à l'exécution du flux de travail CodeCatalyst et recommencez.

## Vous cherchez toujours à résoudre votre problème ?

Vous pouvez vous rendre [sur Amazon CodeCatalyst](#) ou remplir un [formulaire de commentaires de support](#). Dans la section Demande d'informations, sous Comment pouvons-nous vous aider, indiquez que vous êtes un CodeCatalyst client Amazon. Fournissez le plus de détails possible afin que nous puissions résoudre votre problème le plus efficacement possible.

## Résolution des problèmes liés aux flux de travail

Consultez les sections suivantes pour résoudre les problèmes liés aux flux de travail sur Amazon CodeCatalyst. Pour de plus amples informations sur les flux de travail, veuillez consulter [???](#).

### Rubriques

- [Comment corriger les messages « Le flux de travail est inactif » ?](#)
- [Comment corriger les erreurs « La définition du flux de travail ne comporte aucune erreur » ?](#)
- [Comment corriger les erreurs « Impossible de localiser les informations d'identification » et ExpiredToken « » ?](#)
- [Comment corriger les erreurs « Impossible de se connecter au serveur » ?](#)
- [Pourquoi les CodeDeploy champs sont-ils absents de l'éditeur visuel ?](#)

- [Comment corriger les erreurs liées aux fonctionnalités IAM ?](#)
- [Comment corriger les erreurs « npm install » ?](#)
- [Pourquoi plusieurs flux de travail portent-ils le même nom ?](#)
- [Puis-je stocker les fichiers de définition de mon flux de travail dans un autre dossier ?](#)
- [Comment ajouter des actions en séquence à mon flux de travail ?](#)
- [Pourquoi mon flux de travail est-il correctement validé mais échoue-t-il lors de l'exécution ?](#)
- [La détection automatique ne détecte aucun rapport concernant mon action](#)
- [Mon action échoue sur les rapports découverts automatiquement une fois que j'ai configuré les critères de réussite](#)
- [La découverte automatique génère des rapports dont je ne veux pas](#)
- [La découverte automatique génère de nombreux petits rapports pour une seule infrastructure de test](#)
- [Les flux de travail répertoriés sous CI/CD ne correspondent pas à ceux du référentiel source](#)
- [Je n'arrive pas à créer ou à mettre à jour des flux de travail](#)

## Comment corriger les messages « Le flux de travail est inactif » ?

Problème : Dans la CodeCatalyst console, sous CI/CD, Workflows, votre flux de travail apparaît avec le message suivant :

```
Workflow is inactive.
```

Ce message indique que le fichier de définition du flux de travail contient un déclencheur qui ne s'applique pas à la branche sur laquelle vous vous trouvez actuellement. Par exemple, votre fichier de définition de flux de travail peut contenir un PUSH déclencheur qui fait référence à votre main branche, mais vous êtes sur une branche de fonctionnalités. Étant donné que les modifications que vous apportez à votre branche de fonctionnalités ne s'appliquent pas aux main exécutions de flux de travail et ne démarrent pas main, CodeCatalyst désactive le flux de travail de la branche et le marque comme telInactive.

Correctifs possibles :

Si vous souhaitez démarrer un flux de travail sur votre branche de fonctionnalités, vous pouvez effectuer les opérations suivantes :

- Dans votre branche de fonctionnalités, dans le fichier de définition du flux de travail, supprimez la `Branches` propriété de la `Triggers` section pour qu'elle ressemble à ceci :

```
Triggers:  
- Type: PUSH
```

Cette configuration entraîne l'activation du déclencheur lors d'un push vers n'importe quelle branche, y compris votre branche de fonctionnalités. Si le déclencheur est activé, un flux de travail CodeCatalyst sera lancé à l'aide du fichier de définition du flux de travail et des fichiers source, quelle que soit la branche vers laquelle vous envoyez le message.

- Dans votre branche de fonctionnalités, dans le fichier de définition du flux de travail, supprimez la `Triggers` section et exécutez le flux de travail manuellement.
- Dans votre branche de fonctionnalités, dans le fichier de définition du flux de travail, modifiez la `PUSH` section afin qu'elle fasse référence à votre branche de fonctionnalités plutôt qu'à une autre branche (par exemple). `main`

#### Important

Veillez à ne pas valider ces modifications si vous n'avez pas l'intention de les fusionner pour les réintégrer dans votre `main` succursale.

Pour plus d'informations sur la modification du fichier de définition du flux de travail, consultez [Pour modifier un flux de travail](#).

Pour plus d'informations sur les déclencheurs, consultez [Utilisation de déclencheurs](#).

## Comment corriger les erreurs « La définition du flux de travail **ne** comporte aucune erreur » ?

Problème : l'un des messages d'erreur suivants s'affiche :

Erreur 1 :

Sur la page CI/CD, Workflows, sous le nom de votre flux de travail, vous pouvez voir :

Workflow definition has ***n*** errors



## Erreur 2 :

Lorsque vous modifiez un flux de travail, vous cliquez sur le bouton Valider et le message suivant apparaît en haut de la CodeCatalyst console :

```
The workflow definition has errors. Fix the errors and choose Validate to verify your changes.
```

## Erreur 3 :

Après avoir accédé à la page de détails de votre flux de travail, l'erreur suivante s'affiche dans le champ de définition du flux de travail :

```
n errors
```

Correctifs possibles :

- Choisissez CI/CD, choisissez Workflows, puis choisissez le nom du flux de travail contenant l'erreur. Dans le champ de définition du flux de travail situé en haut, choisissez le lien vers l'erreur. Les détails de l'erreur apparaissent au bas de la page. Suivez les conseils de dépannage indiqués dans le message d'erreur pour résoudre le problème.
- Assurez-vous que le fichier de définition du flux de travail est un fichier YAML.
- Assurez-vous que les propriétés YAML du fichier de définition du flux de travail sont imbriquées au bon niveau. Pour savoir comment les propriétés doivent être imbriquées dans le fichier de définition du flux de travail, reportez-vous à la documentation de votre action [Référence de définition du flux de travail](#), ou consultez la documentation de votre action, qui est liée à from [Ajouter une action](#).
- Assurez-vous que les astérisques (\*) et les autres caractères spéciaux sont correctement masqués. Pour y échapper, ajoutez des guillemets simples ou doubles. Par exemple :

```
Outputs:  
Artifacts:  
  - Name: myartifact  
  Files:  
    - "**/*"
```

Pour plus d'informations sur les caractères spéciaux dans le fichier de définition du flux de travail, consultez [Consignes et conventions de syntaxe](#).

- Assurez-vous que les propriétés YAML du fichier de définition du flux de travail utilisent les bonnes majuscules. Pour plus d'informations sur les règles relatives au boîtier, voir [Consignes](#)

[et conventions de syntaxe](#). Pour déterminer le bon boîtier de chaque propriété, reportez-vous à la documentation de votre action ou consultez la [Référence de définition du flux de travail](#) documentation relative à votre action, dont le lien se trouve à partir de [Ajouter une action](#).

- Assurez-vous que la `SchemaVersion` propriété est présente et définie sur la bonne version dans le fichier de définition du flux de travail. Pour plus d'informations, consultez [SchemaVersion](#).
- Assurez-vous que la `Triggers` section du fichier de définition du flux de travail inclut toutes les propriétés requises. Pour déterminer les propriétés requises, choisissez le déclencheur dans l'[éditeur visuel](#) et recherchez les champs dans lesquels des informations sont manquantes, ou consultez la documentation de référence du déclencheur à l'adresse [Triggers](#).
- Assurez-vous que la `DependsOn` propriété du fichier de définition du flux de travail est correctement configurée et qu'elle n'introduit pas de dépendances circulaires. Pour plus d'informations, consultez [Configuration des actions pour qu'elles dépendent d'autres actions](#).
- Assurez-vous que la `Actions` section du fichier de définition du flux de travail inclut au moins une action. Pour plus d'informations, consultez [Actions](#).
- Assurez-vous que chaque action inclut toutes les propriétés requises. Pour déterminer les propriétés requises, choisissez l'action dans l'[éditeur visuel](#) et recherchez les champs dans lesquels des informations sont manquantes, ou consultez la documentation de votre action, qui est liée à partir de [Ajouter une action](#).
- Assurez-vous que tous les artefacts d'entrée ont des artefacts de sortie correspondants. Pour plus d'informations, consultez [Définition d'un artefact de sortie](#).
- Assurez-vous que les variables définies dans une action sont exportées afin de pouvoir être utilisées dans d'autres actions. Pour plus d'informations, consultez [Exporter une variable afin que d'autres actions puissent l'utiliser](#).

## Comment corriger les erreurs « Impossible de localiser les informations d'identification » et ExpiredToken « » ?

Problème : pendant que vous travaillez [Tutoriel : Déploiement d'une application sur Amazon EKS](#), l'un des messages d'erreur suivants ou les deux s'affichent dans la fenêtre du terminal de votre machine de développement :

```
Unable to locate credentials. You can configure credentials by running "aws configure".
```

```
ExpiredToken: The security token included in the request is expired
```

## Correctifs possibles :

Ces erreurs indiquent que les informations d'identification que vous utilisez pour accéder aux AWS services ont expiré. Dans ce cas, n'exécutez pas la `aws configure` commande. Utilisez plutôt les instructions suivantes pour actualiser votre clé AWS d'accès et votre jeton de session.

Pour actualiser votre clé AWS d'accès et votre jeton de session

1. Assurez-vous de disposer de l'URL, du nom d'utilisateur et du mot de passe du portail d'AWS accès de l'utilisateur que vous utilisez dans l'intégralité du didacticiel Amazon EKS (codecatalyst-eks-user). Vous devriez avoir configuré ces éléments à la fin [Étape 1 : configurer votre machine de développement](#) du didacticiel.

### Note

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

2. Effectuez l'une des actions suivantes :
  - Collez l'URL du portail d'AWS accès dans la barre d'adresse de votre navigateur.

Ou

  - Actualisez la page du portail d'AWS accès si elle est déjà chargée.
3. Connectez-vous avec le nom `codecatalyst-eks-user` d'utilisateur et le mot de passe, si ce n'est pas déjà fait.
4. Choisissez Compte AWS, puis choisissez le nom auquel vous avez attribué l'`codecatalyst-eks-user` utilisateur et le jeu d'autorisations. Compte AWS
5. À côté du nom du jeu d'autorisations (`codecatalyst-eks-permission-set`), choisissez Ligne de commande ou Accès programmatique.
6. Copiez les commandes au milieu de la page. Ils ressemblent à ce qui suit :

```
export AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"  
export AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"  
export AWS_SESSION_TOKEN="session-token"
```

... où le *jeton de session* est une longue chaîne aléatoire.

7. Collez les commandes dans l'invite de votre terminal sur votre machine de développement et appuyez sur Entrée.

Les nouvelles clés et le nouveau jeton de session sont chargés.

Vous avez maintenant actualisé vos informations d'identification. Les `kubectl` commandes `AWS CLI``eksctl`, et devraient maintenant fonctionner.

## Comment corriger les erreurs « Impossible de se connecter au serveur » ?

Problème : pendant que vous suivez le didacticiel décrit dans [Tutoriel : Déploiement d'une application sur Amazon EKS](#), un message d'erreur similaire au suivant s'affiche dans la fenêtre du terminal de votre machine de développement :

```
Unable to connect to the server: dial tcp: lookup long-string.gr7.us-west-2.eks.amazonaws.com on 1.2.3.4:5: no such host
```

Correctifs possibles :

Cette erreur indique généralement que les informations d'identification utilisées par l'outil `kubectl` pour se connecter à votre cluster Amazon EKS ont expiré. Pour résoudre le problème, actualisez les informations d'identification en saisissant la commande suivante à l'invite du terminal :

```
aws eks update-kubeconfig --name codecatalyst-eks-cluster --region us-west-2
```

Où :

- *codecatalyst-eks-cluster* est remplacé par le nom de votre cluster Amazon EKS.
- *us-west-2* est remplacé par la région dans laquelle votre cluster est AWS déployé.

## Pourquoi les CodeDeploy champs sont-ils absents de l'éditeur visuel ?

Problème : vous utilisez une action [Deploy to Amazon ECS](#) et les CodeDeploy champs ne s'affichent pas, par exemple CodeDeploy AppSpec dans l'éditeur visuel du flux de travail. Ce problème peut se produire parce que le service Amazon ECS que vous avez spécifié dans le champ Service n'est pas configuré pour effectuer des déploiements bleu/vert.

## Correctifs possibles :

- Choisissez un autre service Amazon ECS dans l'onglet Configuration de l'action Deploy to Amazon ECS. Pour plus d'informations, consultez [Ajout de l'action « Déployer sur Amazon ECS »](#).
- Configurez le service Amazon ECS sélectionné pour effectuer des déploiements bleu/vert. Pour plus d'informations sur la configuration des déploiements bleu/vert, consultez la section Déploiement [bleu/vert avec dans CodeDeploy](#) le manuel Amazon Elastic Container Service Developer Guide.

## Comment corriger les erreurs liées aux fonctionnalités IAM ?

Problème : vous utilisez une action [Deploy AWS CloudFormation stack](#), et vous pouvez le voir `##[error] requires capabilities: [capability-name]` dans les journaux de votre action Deploy AWS CloudFormation stack.

Correctifs possibles : suivez la procédure suivante pour ajouter la fonctionnalité au fichier de définition du flux de travail. Pour plus d'informations sur les fonctionnalités IAM, consultez la section [Reconnaissance des ressources IAM dans les AWS CloudFormation modèles](#) du Guide de l'utilisateur IAM.

### Visual

Pour ajouter une fonctionnalité IAM à l'aide de l'éditeur visuel

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez Visual.
7. Dans le diagramme du flux de travail, choisissez votre action Deploy AWS CloudFormation stack.
8. Cliquez sur l'onglet Configuration.
9. En bas de la page, sélectionnez Avancé - facultatif.

10. Dans la liste déroulante Fonctionnalités, cochez la case à côté de la fonctionnalité mentionnée dans le message d'erreur. Si la fonctionnalité n'est pas disponible dans la liste, utilisez l'éditeur YAML pour l'ajouter.
11. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
12. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.
13. Si une nouvelle exécution de flux de travail ne démarre pas automatiquement, exécutez-le manuellement pour voir si les modifications corrigent l'erreur. Pour plus d'informations sur l'exécution manuelle d'un flux de travail, consultez [Démarrage d'une exécution de flux de travail](#).

## YAML

Pour ajouter une fonctionnalité IAM à l'aide de l'éditeur YAML

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Choisissez votre projet.
3. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
4. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
5. Choisissez Modifier.
6. Choisissez YAML.
7. Dans l'action Deploy AWS CloudFormation stack, ajoutez une capabilities propriété, comme celle-ci :

```
DeployCloudFormationStack:  
  Configuration:  
    capabilities: capability-name
```

Remplacez *capability-name* par le nom de la fonctionnalité IAM indiquée dans le message d'erreur. Utilisez des virgules et aucun espace pour répertorier plusieurs fonctionnalités. Pour plus d'informations, consultez la description de la capabilities propriété dans le [Référence d'action « Deploy AWS CloudFormation Stack »](#).

8. (Facultatif) Choisissez Valider pour valider le code YAML du flux de travail avant de le valider.
9. Choisissez Valider, entrez un message de validation, puis choisissez à nouveau Valider.

10. Si une nouvelle exécution de flux de travail ne démarre pas automatiquement, exécutez-le manuellement pour voir si les modifications corrigent l'erreur. Pour plus d'informations sur l'exécution manuelle d'un flux de travail, consultez [Démarrage d'une exécution de flux de travail](#).

## Comment corriger les erreurs « npm install » ?

Problème : votre [action de AWS CDK déploiement ou d'AWS CDKamorçage](#) échoue avec une `npm install` erreur. Cette erreur peut se produire parce que vous stockez les dépendances de votre AWS CDK application dans un registre de gestionnaire de packages de nœuds privés (npm) auquel l'action ne peut pas accéder.

Corrections possibles : suivez les instructions suivantes pour mettre à jour le `cdk.json` fichier de votre AWS CDK application avec des informations de registre et d'authentification supplémentaires.

Avant de commencer

1. Créez des secrets pour vos informations d'authentification. Vous allez référencer ces secrets dans le `cdk.json` fichier au lieu de fournir les équivalents en texte clair. Pour créer les secrets :
  - a. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
  - b. Choisissez votre projet.
  - c. Dans le volet de navigation, choisissez CI/CD, puis Secrets.
  - d. Créez deux secrets dotés des propriétés suivantes :

| Premier secret  | Deuxième secret   |
|---|---|
| Nom: <code>npmUsername</code>   | Nom: <code>npmAuthToken</code>  |
| Valeur : <code>npm-username</code> , où <code>npm-username</code> est le nom d'utilisateur utilisé pour vous authentifier auprès de votre registre npm privé. | Valeur : <code>npm-auth-token</code> , où <code>npm-auth-token</code> est le jeton d'accès utilisé pour s'authentifier auprès de votre registre npm privé. Pour plus d'informations sur les jetons d'accès npm, voir <a href="#">À propos des jetons d'accès</a> dans la documentation npm. |
| (Facultatif) Description : The username used to authenticate to the private npm registry.   |   |

| Premier secret | Deuxième secret   |
|----------------|---|
|                | (Facultatif) Description : The access token used to authenticate to the private npm registry. |

Pour plus d'informations sur les secrets, consultez [Travailler avec des secrets](#).

2. Ajoutez les secrets en tant que variables d'environnement à votre AWS CDK action. L'action remplacera les variables par des valeurs réelles lors de son exécution. Pour ajouter les secrets :
  - a. Dans le volet de navigation, choisissez CI/CD, puis Workflows.
  - b. Choisissez le nom de votre flux de travail. Vous pouvez filtrer par le nom du référentiel source ou de la branche où le flux de travail est défini, ou filtrer par nom de flux de travail.
  - c. Choisissez Modifier.
  - d. Choisissez Visual.
  - e. Dans le diagramme du flux de travail, choisissez votre AWS CDK action.
  - f. Sélectionnez l'onglet Entrées.
  - g. Ajoutez deux variables avec les propriétés suivantes :

| Première variable                                     | Deuxième variable                                      |
|---|--|
| Nom: NPMUSER  | Nom: NPMTOKEN  |
| Value (Valeur) : <code>\${Secrets.npmUsername}</code> | Value (Valeur) : <code>\${Secrets.npmAuthToken}</code> |

Vous disposez désormais de deux variables contenant des références à des secrets.

Le code YAML de votre fichier de définition de flux de travail doit ressembler à ce qui suit :

#### Note

L'exemple de code suivant provient d'une action d'AWS CDK amorçage ; une action de AWS CDK déploiement sera similaire.




```
Name: CDK_Bootstrap_Action
SchemaVersion: 1.0
Actions:
  CDKBootstrapAction:
    Identifiant: aws/cdk-bootstrap@v1
    Inputs:
      Variables:
        - Name: NPMUSER
          Value: ${Secrets.npmUsername}
        - Name: NPMTOKEN
          Value: ${Secrets.npmAuthToken}
      Sources:
        - WorkflowSource
    Environment:
      Name: Dev2
    Connections:
      - Name: account-connection
        Role: codecatalystAdmin
    Configuration:
      Parameters:
        Region: "us-east-2"
```

Vous êtes maintenant prêt à utiliser les NPMTOKEN variables NPMUSER et de votre `cdk.json` fichier. Passez à la procédure suivante.

Pour mettre à jour votre fichier `cdk.json`

1. Accédez au répertoire racine de votre AWS CDK projet et ouvrez le `cdk.json` fichier.
2. Recherchez la "app" : propriété et modifiez-la pour inclure le code indiqué en *italique rouge* :

 Note


L'exemple de code suivant provient d'un TypeScript projet. Si vous utilisez un JavaScript projet, le code sera similaire mais pas identique.

```
{
```

```
"app": "npm set registry=https://your-registry/folder/CDK-package/ --
userconfig .npmrc && npm set //your-registry/folder/CDK-package/:always-auth=true
--userconfig .npmrc && npm set //your-registry/folder/CDK-package/:_authToken=
\"${NPMUSER}\" : \"${NPMTOKEN}\" && npm install && npx ts-node --prefer-ts-exts bin/
hello-cdk.ts|js",
"watch": {
  "include": [
    "*"
  ],
  "exclude": [
    "README.md",
    "cdk*.json",
    "**/*.d.ts",
    "**/*.js",
    "tsconfig.json",
    "package*.json",
    ...
  ]
}
```

3. Dans le code *surligné en italique rouge*, remplacez :

- *Your-registry/folder/cdk-package/* avec le chemin d'accès aux dépendances de votre projet dans votre registre privé. AWS CDK
- *hello-cdk.ts|js* avec le nom de votre fichier de point d'entrée. Il peut s'agir d'un fichier .ts .js (TypeScriptJavaScript) ou () selon la langue que vous utilisez.

 Note

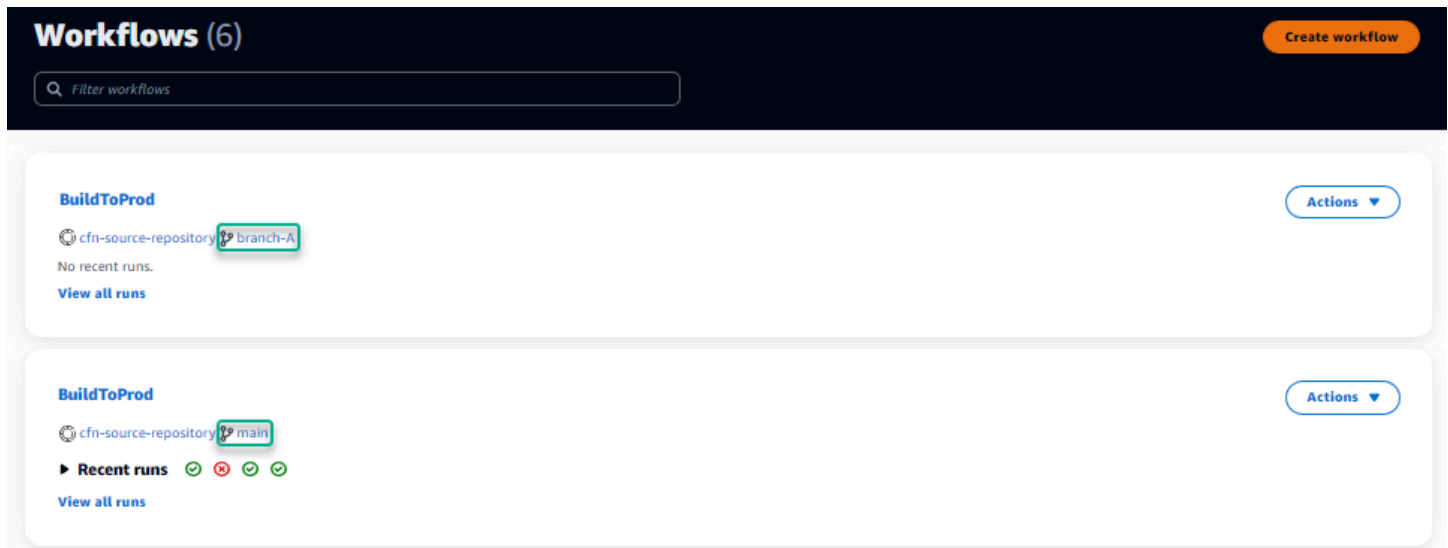
L'action remplacera les variables ***NPMUSER et NPMTOKEN*** par ***le nom d'utilisateur et le jeton*** d'accès npm que vous avez spécifiés dans Secrets.

4. Enregistrez votre fichier cdk.json.
5. Réexécutez l'action manuellement pour voir si les modifications corrigent l'erreur. Pour plus d'informations sur l'exécution manuelle d'actions, consultez [Démarrage d'une exécution de flux de travail](#).

## Pourquoi plusieurs flux de travail portent-ils le même nom ?

Les flux de travail sont stockés par branche et par référentiel. Deux flux de travail différents peuvent porter le même nom s'ils existent dans des branches différentes. Sur la page Flux de travail, vous

pouvez différencier les flux de travail du même nom en consultant le nom de la branche. Pour plus d'informations, consultez [Travailler avec des succursales sur Amazon CodeCatalyst](#).



Puis-je stocker les fichiers de définition de mon flux de travail dans un autre dossier ?

Non, vous devez stocker tous les fichiers de définition du flux de travail dans le `.codecatalyst/workflows` dossier. Si vous utilisez un dépôt mono avec plusieurs projets logiques, placez tous vos fichiers de définition de flux de travail dans le `.codecatalyst/workflows` dossier, puis utilisez la `FilesChanged` propriété dans un déclencheur pour déclencher le flux de travail sur un chemin de projet spécifié. Pour plus d'informations, consultez [Utilisation de déclencheurs](#).

Comment ajouter des actions en séquence à mon flux de travail ?

Par défaut, lorsque vous ajoutez une action à votre flux de travail, celle-ci ne comporte aucune dépendance et s'exécute en parallèle avec les autres actions.

Si vous souhaitez organiser les actions en séquence, vous pouvez définir une dépendance par rapport à une autre action en définissant le `DependsOn` champ. Vous pouvez également configurer une action pour consommer des artefacts ou des variables issus d'autres actions. Pour plus d'informations, consultez [Configuration des actions pour qu'elles dépendent d'autres actions](#).

## Pourquoi mon flux de travail est-il correctement validé mais échoue-t-il lors de l'exécution ?

Si vous avez validé votre flux de travail à l'aide du `Validate` bouton, mais qu'il a quand même échoué, cela peut être dû à une limitation du validateur.

Toute erreur faisant référence à une CodeCatalyst ressource telle que des secrets, des environnements ou des flottes dans la configuration du flux de travail ne sera pas enregistrée lors d'une validation. Si des références non valides sont utilisées, l'erreur ne sera identifiée que lors de l'exécution d'un flux de travail. De même, s'il existe des erreurs dans la configuration de votre action, telles que l'absence d'un champ obligatoire ou des fautes de frappe dans les attributs de l'action, elles ne seront identifiées que lors de l'exécution du flux de travail. Pour plus d'informations, consultez [Création, modification et suppression d'un flux de travail](#).

## La détection automatique ne détecte aucun rapport concernant mon action

Problème : J'ai configuré la détection automatique pour une action qui exécute des tests, mais aucun rapport n'est découvert par CodeCatalyst.

Correctifs possibles : Cela peut être dû à un certain nombre de problèmes. Essayez une ou plusieurs des solutions suivantes :

- Assurez-vous que l'outil utilisé pour exécuter les tests produit des sorties dans l'un des formats CodeCatalyst compatibles. Par exemple, si vous `pytest` souhaitez autoriser la découverte des rapports CodeCatalyst de test et de couverture du code, incluez les arguments suivants :

```
--junitxml=test_results.xml --cov-report xml:test_coverage.xml
```

Pour plus d'informations, consultez [Types de rapports de test](#).

- Assurez-vous que l'extension de fichier pour les sorties correspond au format choisi. Par exemple, lors de la configuration `pytest` pour produire des résultats au JUnitXML format, vérifiez que l'extension du fichier est bien `.xml`. Pour plus d'informations, consultez [Types de rapports de test](#).
- Assurez-vous qu'il `IncludePaths` est configuré pour inclure l'ensemble du système de fichiers (`**/*`), sauf si vous excluez certains dossiers exprès. De même, assurez-vous de `ExcludePaths` ne pas exclure les répertoires dans lesquels vous souhaitez que vos rapports se trouvent.
- Si vous avez configuré manuellement un rapport pour utiliser un fichier de sortie spécifique, il sera exclu de la détection automatique. Pour plus d'informations, consultez [Exemple : Configuration de rapports](#).

- La détection automatique peut ne pas trouver de rapports car l'action a échoué avant que les résultats ne soient générés. Par exemple, il se peut que la compilation ait échoué avant que les tests unitaires aient été exécutés.

## Mon action échoue sur les rapports découverts automatiquement une fois que j'ai configuré les critères de réussite

Problème : lorsque j'active la découverte automatique et que je configure les critères de réussite, certains rapports ne répondent pas aux critères de réussite et l'action échoue par conséquent.

Correctifs possibles : Pour résoudre ce problème, essayez une ou plusieurs des solutions suivantes :

- Modifiez `IncludePaths` ou `ExcludePaths` excluez les rapports qui ne vous intéressent pas.
- Mettez à jour les critères de réussite pour permettre à tous les rapports de passer. Par exemple, si deux rapports ont été découverts, l'un présentant une couverture linéaire de 50 % et l'autre de 70 %, ajustez la couverture linéaire minimale à 50 %. Pour plus d'informations, consultez [Critères de réussite](#).
- Transformez le rapport défaillant en un rapport configuré manuellement. Cela vous permet de configurer différents critères de réussite pour ce rapport spécifique. Pour plus d'informations, consultez [Configuration des critères de réussite pour les rapports](#).

## La découverte automatique génère des rapports dont je ne veux pas

Problème : lorsque j'active la découverte automatique, cela génère des rapports dont je ne veux pas avoir besoin. Par exemple, CodeCatalyst génère des rapports de couverture de code pour les fichiers inclus dans les dépendances de mon application stockées dans `node_modules`.

Corrections possibles : vous pouvez ajuster la `ExcludePaths` configuration pour exclure les fichiers indésirables. Par exemple, pour exclure `node_modules`, ajoutez `node_modules/**/*`. Pour plus d'informations, consultez [Inclure/exclure des chemins](#).

## La découverte automatique génère de nombreux petits rapports pour une seule infrastructure de test

Problème : lorsque j'utilise certains frameworks de reporting relatifs aux tests et à la couverture du code, j'ai remarqué que la découverte automatique génère un grand nombre de rapports. Par

exemple, lorsque vous utilisez le [plug-in Maven Surefire](#), la découverte automatique produit un rapport différent pour chaque classe de test.

Corrections possibles : votre framework peut être en mesure d'agrèger les sorties dans un seul fichier. Par exemple, si vous utilisez le plugin Maven Surefire, vous pouvez l'utiliser `npx junit-merge` pour agrèger les fichiers manuellement. L'expression complète peut ressembler à ceci :

```
mvn test; cd test-package-path/surefire-reports && npx junit-merge -d ./ && rm *Test.xml
```

## Les flux de travail répertoriés sous CI/CD ne correspondent pas à ceux du référentiel source

Problème : Les flux de travail affichés sur la page CI/CD, Workflows ne correspondent pas à ceux du `~/.codecatalyst/workflows/` dossier de votre référentiel [source](#). Vous pouvez constater les incohérences suivantes :

- Un flux de travail apparaît sur la page Flux de travail, mais aucun fichier de définition de flux de travail correspondant n'existe dans votre référentiel source.
- Un fichier de définition de flux de travail existe dans votre référentiel source, mais aucun flux de travail correspondant n'apparaît sur la page Workflows.
- Un flux de travail existe à la fois dans le référentiel source et dans la page Workflows, mais les deux sont différents.

Ce problème peut se produire si la page Workflows n'a pas eu le temps de s'actualiser ou si un quota de flux de travail a été dépassé.

Correctifs possibles :

- Patientez. Vous devez généralement attendre deux ou trois secondes après une validation dans la source avant de voir le changement sur la page Workflows.
- Si vous avez dépassé un quota de flux de travail, effectuez l'une des opérations suivantes :

### Note

Pour déterminer si un quota de flux de travail a été dépassé [Quotas pour les flux de travail dans CodeCatalyst](#), passez en revue et recoupez les quotas documentés par rapport aux

flux de travail de votre référentiel source ou sur la page Flux de travail. Aucun message d'erreur n'indique qu'un quota a été dépassé. Vous devrez donc vérifier par vous-même.

- Si vous avez dépassé le nombre maximal de flux de travail par quota d'espace, [supprimez certains flux](#) de travail, puis effectuez un test de validation par rapport au fichier de définition du flux de travail. Un exemple de validation de test peut consister à ajouter un espace au fichier.
- Si vous avez dépassé le quota de taille maximale du fichier de définition du flux de travail, modifiez le fichier de définition du flux de travail pour réduire sa longueur.
- Si vous avez dépassé le nombre maximal de fichiers de flux de travail traités dans le cadre d'un quota d'événements source unique, effectuez plusieurs validations de test. Modifiez un nombre de flux de travail inférieur au maximum dans chaque validation.
- Augmentez les quotas de flux de travail en activant la facturation des niveaux payants. Pour plus d'informations, consultez [la section Gestion de la facturation](#) dans le manuel Amazon CodeCatalyst Administrator Guide.

## Je n'arrive pas à créer ou à mettre à jour des flux de travail

Problème : je souhaite créer ou mettre à jour un flux de travail, mais un message d'erreur s'affiche lorsque j'essaie de valider la modification.

Corrections possibles : selon votre rôle dans le projet ou dans l'espace, il se peut que vous ne soyez pas autorisé à envoyer du code vers les référentiels sources du projet. Les fichiers YAML pour les flux de travail sont stockés dans des référentiels. Pour plus d'informations, consultez [Fichiers de définition du flux de travail](#). Les rôles d'administrateur d'espace, d'administrateur de projet et de contributeur sont tous autorisés à valider et à envoyer du code vers les référentiels d'un projet.

Si vous avez le rôle de contributeur mais que vous ne pouvez pas créer ou modifier le flux de travail YAML dans une branche spécifique, il se peut qu'une règle de branche configurée pour cette branche empêche les utilisateurs ayant ce rôle de transmettre du code à cette branche en particulier. Essayez de créer un flux de travail dans une autre branche ou de valider vos modifications dans une autre branche. Pour plus d'informations, voir [Gérer les actions autorisées pour une branche à l'aide de règles de branche](#).

# Résolution des problèmes liés à la recherche dans CodeCatalyst

Consultez les sections suivantes pour résoudre les problèmes liés à la recherche. CodeCatalyst Pour de plus amples informations sur les flux de travail, veuillez consulter [Rechercher dans CodeCatalyst](#).

## Rubriques

- [Je ne trouve pas d'utilisateur dans mon projet](#)
- [Je ne vois pas ce que je recherche dans mon projet ou mon espace](#)
- [Le nombre de résultats de recherche ne cesse de changer lorsque je navigue dans les pages](#)
- [Ma requête de recherche n'est pas terminée](#)

## Je ne trouve pas d'utilisateur dans mon projet

Problème : Lorsque j'essaie de consulter les informations d'un utilisateur, je ne les vois pas dans le projet.

Corrections possibles : La recherche ne prend actuellement pas en charge la recherche d'utilisateurs au sein d'un projet. Pour rechercher des utilisateurs ayant accès à votre espace, passez à Cet espace dans QuickSearch ou supprimez les filtres de projet que vous avez éventuellement spécifiés à l'aide du langage de requête avancé.

## Je ne vois pas ce que je recherche dans mon projet ou mon espace

Problème : les résultats ne s'affichent pas lorsque j'essaie de rechercher des informations spécifiques.

Correctifs possibles : les mises à jour du contenu peuvent prendre quelques secondes pour s'afficher dans les résultats de recherche. Les mises à jour importantes peuvent prendre plusieurs minutes.

Pour les ressources qui n'ont pas été mises à jour récemment, vous devrez peut-être affiner votre recherche. Vous pouvez affiner en ajoutant d'autres mots clés ou en utilisant le langage de requête avancé. Pour plus d'informations sur l'affinement de vos requêtes, consultez [Affiner votre requête de recherche](#).



## Le nombre de résultats de recherche ne cesse de changer lorsque je navigue dans les pages

Problème : le nombre de résultats de recherche semble changer lorsque je passe à la page suivante. Le nombre total de résultats n'est donc pas clair.

Corrections possibles : Lorsque vous naviguez dans les pages des résultats de recherche, vous pouvez constater une modification du nombre de résultats de recherche correspondant à votre requête. Le nombre de résultats peut être mis à jour pour refléter un nombre plus précis de correspondances découvertes lorsque vous naviguez dans les pages.

Lorsque vous parcourez les résultats, le message suivant peut s'afficher : Aucun résultat pour « test ». Vous recevrez le message si vous n'avez pas accès aux résultats restants.

## Ma requête de recherche n'est pas terminée

Problème : les résultats de ma recherche ne s'affichent pas et cela semble prendre trop de temps.

Corrections possibles : il est possible que votre recherche ne soit pas terminée lorsque plusieurs recherches sont effectuées simultanément dans l'espace, soit par programmation, soit en raison d'une forte activité de l'équipe. Si vous effectuez des recherches programmatiques, mettez-les en pause ou diminuez-les. Sinon, réessayez dans quelques secondes.

## Résolution des problèmes liés aux comptes associés à votre espace

Dans CodeCatalyst, vous pouvez ajouter un Compte AWS à votre espace pour accorder des autorisations aux ressources et à des fins de facturation. Les informations suivantes peuvent vous aider à résoudre les problèmes courants liés aux comptes associés dans CodeCatalyst.

### Rubriques

- [Ma demande Compte AWS de connexion reçoit une erreur de jeton non valide](#)
- [CodeCatalyst Le flux de travail de mon projet Amazon échoue avec une erreur concernant le compte, l'environnement ou le rôle IAM configuré](#)
- [J'ai besoin d'un compte, d'un rôle et d'un environnement associés pour créer un projet](#)
- [Je ne parviens pas à accéder à la page Amazon CodeCatalyst Spaces dans AWS Management Console](#)

- [Je souhaite utiliser un autre compte comme compte de facturation](#)

## Ma demande Compte AWS de connexion reçoit une erreur de jeton non valide

Problème : lors de la création d'une demande de connexion avec un jeton de connexion, la page n'accepte pas le jeton et affiche une erreur indiquant que le jeton n'est pas valide.

Corrections possibles : Assurez-vous de fournir l'identifiant de compte que vous souhaitez ajouter à votre espace. Vous devez disposer d'autorisations administratives pour votre compte Compte AWS ou être en mesure de travailler avec votre administrateur pour ajouter le compte.

Lorsque vous choisissez de vérifier le compte, une nouvelle fenêtre de navigateur s'ouvre dans le AWS Management Console. Le même compte doit être connecté côté console. Réessayez après avoir vérifié les points suivants :

- Vous êtes connecté AWS Management Console avec le même Compte AWS que celui que vous souhaitez ajouter à votre espace.
- Vous êtes connecté au en sélectionnant AWS Management Console l'ouest des États-Unis (Oregon) Région AWS (us-west-2).
- Si vous êtes arrivé depuis la page de facturation et Compte AWS que vous souhaitez ajouter le compte de facturation spécifié pour votre espace, assurez-vous que le compte n'est pas déjà un compte de facturation pour un autre espace.

## CodeCatalyst Le flux de travail de mon projet Amazon échoue avec une erreur concernant le compte, l'environnement ou le rôle IAM configuré

Problème : lorsque le flux de travail s'exécute et ne trouve aucun compte configuré ou aucun rôle IAM associé à votre espace, vous devez remplir les champs rôle, connexion et environnement manuellement dans le flux de travail YAML. Consultez l'action de flux de travail qui a échoué et notez si les messages d'erreur sont les suivants :

- Le rôle n'est pas disponible pour être utilisé avec la connexion associée à l'environnement.
- L'action n'a pas abouti. État : ÉCHEC ; la valeur fournie pour la connexion au compte ou l'environnement n'est pas valide. Vérifiez que la connexion est associée à votre espace et que l'environnement est associé à votre projet.

- L'action n'a pas abouti. État : ÉCHEC ; la valeur fournie pour le rôle IAM n'est pas valide. Vérifiez que le nom existe, que le rôle IAM est ajouté à la connexion de votre compte et que la connexion est déjà associée à votre espace Amazon CodeCatalyst

Corrections possibles : assurez-vous que les champs YAML du flux de travail contiennent des valeurs précises pour [l'environnement](#), [les connexions](#) et [le rôle](#). Les actions CodeCatalyst de flux de travail qui nécessitent un environnement sont des actions de création ou de déploiement qui exécutent AWS des ressources ou qui génèrent des piles de AWS ressources.

Choisissez le bloc d'action du flux de travail qui a échoué, puis choisissez Visual. Cliquez sur l'onglet Configuration. Si les champs Environnement, Nom de connexion et Nom du rôle ne sont pas renseignés, vous devrez mettre à jour le flux de travail manuellement. Procédez comme suit pour modifier le code YAML de votre flux de travail :

- Développez le `/.codecatalyst` répertoire, puis `/workflows` développez-le. Ouvrez le fichier YAML du flux de travail. Assurez-vous que les rôles IAM et les informations de compte sont spécifiés dans le code YAML que vous avez configuré pour votre flux de travail. Exemple :

```
Actions:
  cdk_bootstrap:
    Identifiant: action-@v1
    Inputs:
      Sources:
        - WorkflowSource
    Environment:
      Name: Staging
    Connections:
      - Name: account-connection
        Role: build-role
```

Les propriétés Environment, Connection et Role sont requises pour exécuter des actions de création et de déploiement de CodeCatalyst flux de travail avec AWS des ressources. Pour un exemple, consultez les paramètres YAML de référence de l'action de CodeCatalyst construction pour [l'environnement](#), [les connexions](#) et [le rôle](#).

- Assurez-vous qu'un compte a été ajouté à votre espace et que le ou les rôles IAM appropriés ont été ajoutés au compte. Vous pouvez ajuster ou ajouter des comptes si vous avez le rôle d'administrateur de l'espace. Pour de plus amples informations, veuillez consulter [Administration Comptes AWS d'un espace](#).

## J'ai besoin d'un compte, d'un rôle et d'un environnement associés pour créer un projet

Problème : Dans les options de création de projet, soit aucun compte supplémentaire n'est disponible dans mon espace pour mon projet, soit j'ai besoin qu'un autre compte soit ajouté à mon espace pour que mon projet puisse l'utiliser.

Corrections possibles : Pour votre espace, vous pouvez ajouter des personnes autorisées Comptes AWS à les ajouter à votre projet si vous avez le rôle d'administrateur de l'espace. Vous devez également disposer d'un Compte AWS endroit où vous disposez d'autorisations administratives ou pouvez travailler avec votre AWS administrateur.

Pour vous assurer qu'un compte et un rôle seront disponibles sur l'écran de création du projet, vous devez d'abord ajouter le compte et les rôles. Pour de plus amples informations, veuillez consulter [Administration Comptes AWS d'un espace](#).

Vous avez la possibilité de choisir de créer un rôle de service avec une politique de rôle appelée politique de CodeCatalystWorkflowDevelopmentRole-*spaceName*rôle. Le rôle portera un nom CodeCatalystWorkflowDevelopmentRole-*spaceName* avec un identifiant unique ajouté. Pour plus d'informations sur le rôle et la politique de rôle, consultez [Comprendre le rôle CodeCatalystWorkflowDevelopmentRole-\*spaceName\*du service](#). Pour connaître les étapes de création du rôle, consultez [Création du CodeCatalystWorkflowDevelopmentRole-\*spaceName\*rôle pour votre compte et votre espace](#). Le rôle est ajouté à votre compte et disponible sur les pages de création de projets de CodeCatalyst.

## Je ne parviens pas à accéder à la page Amazon CodeCatalyst Spaces dans AWS Management Console

Problème : Lorsque j'essaie d'accéder à la CodeCatalyst page Amazon AWS Management Console pour ajouter un compte à mon CodeCatalyst espace ou ajouter des rôles à un compte dans AWS, je reçois une erreur d'autorisation.

Correctifs possibles :

Pour votre espace, vous pouvez ajouter des personnes autorisées Comptes AWS à les ajouter à votre projet si vous avez le rôle d'administrateur de l'espace. Vous devez également disposer d'un Compte AWS endroit où vous disposez d'autorisations administratives ou pouvez travailler avec votre AWS administrateur. Vous devez d'abord vous assurer que vous êtes connecté AWS

Management Console au même compte que celui que vous souhaitez gérer. Une fois connecté au AWS Management Console, vous pouvez ouvrir la console et réessayer.

Ouvrez la CodeCatalyst page Amazon à l'AWS Management Console adresse <https://us-west-2.console.aws.amazon.com/codecatalyst/home?region=us-west-2#/>.

## Je souhaite utiliser un autre compte comme compte de facturation

**Problème :** Lorsque j'ai configuré mon CodeCatalyst identifiant, j'ai effectué plusieurs étapes pour configurer mon espace et associer un utilisateur autorisé Compte AWS. Maintenant, je souhaite autoriser un autre compte pour la facturation.

**Corrections possibles :** Pour votre espace, vous pouvez autoriser les comptes de facturation si vous avez le rôle d'administrateur de l'espace. Vous devez également disposer d'un Compte AWS endroit où vous disposez d'autorisations administratives ou pouvez travailler avec votre AWS administrateur.

Pour plus d'informations, consultez [la section Gestion de la facturation](#) dans le manuel Amazon CodeCatalyst Administrator Guide.

## Résolution des problèmes liés aux environnements de développement

Consultez les sections suivantes pour résoudre les problèmes liés aux environnements de développement. Pour plus d'informations sur les environnements de développement, consultez [Environnements de développement dans CodeCatalyst](#).

### Rubriques

- [La création de mon environnement de développement n'a pas réussi en raison d'un problème de quotas](#)
- [Je ne peux pas transférer les modifications de mon environnement de développement vers une branche spécifique d'un référentiel](#)
- [Mon environnement de développement n'a pas repris](#)
- [Mon environnement de développement est déconnecté](#)
- [Mon environnement de développement connecté au VPC a échoué](#)
- [Je ne trouve pas le répertoire dans lequel se trouve mon projet](#)
- [Je ne parviens pas à me connecter à mon environnement de développement via SSH](#)

- [Je ne parviens pas à me connecter à mon environnement de développement via SSH car ma configuration SSH locale est manquante](#)
- [Je ne parviens pas à me connecter à mon environnement de développement via SSH car je rencontre des problèmes avec mon AWS Config profil codecatalyst](#)
- [Résolution des problèmes liés aux IDE](#)
- [Résolution des problèmes liés aux fichiers de développement](#)

## La création de mon environnement de développement n'a pas réussi en raison d'un problème de quotas

Problème : je souhaite créer un environnement de développement dans CodeCatalyst, mais une erreur s'affiche. Dans la console, je vois un message sur la page Dev Environments indiquant que j'ai atteint la limite de stockage pour l'espace.

Corrections possibles : selon votre rôle dans le projet ou dans l'espace, vous pouvez soit supprimer un ou plusieurs de vos propres environnements de développement, soit, si vous avez le rôle d'administrateur de l'espace, vous pouvez supprimer les environnements de développement non utilisés créés par d'autres utilisateurs. Vous pouvez également décider de modifier le niveau de facturation pour un niveau qui inclut davantage de stockage.

- Pour consulter les limites de stockage, consultez l'onglet Facturation de l' CodeCatalystespace Amazon pour voir si les quotas d'utilisation ont atteint le maximum autorisé. Si les quotas ont atteint le maximum, contactez une personne ayant le rôle d'administrateur de l'espace pour supprimer les environnements de développement inutiles ou pour envisager de modifier le niveau de facturation.
- Pour supprimer les environnements de développement que vous avez créés et dont vous n'avez plus besoin, consultez [Suppression d'un environnement de développement](#).

Si le problème persiste et qu'une erreur s'affiche dans votre IDE, vérifiez que votre CodeCatalyst rôle vous permet de créer un environnement de développement. Les rôles d'administrateur d'espace, d'administrateur de projet et de contributeur sont tous autorisés à créer des environnements de développement. Pour en savoir plus, consultez [Utilisation des rôles sur Amazon CodeCatalyst](#).

## Je ne peux pas transférer les modifications de mon environnement de développement vers une branche spécifique d'un référentiel

Problème : je souhaite valider et transférer les modifications de code de mon environnement de développement vers une branche d'un référentiel source, mais une erreur s'affiche.

Corrections possibles : selon votre rôle dans le projet ou dans l'espace, il se peut que vous ne soyez pas autorisé à envoyer du code vers les référentiels sources du projet. Les rôles d'administrateur d'espace, d'administrateur de projet et de contributeur sont tous autorisés à envoyer du code vers les référentiels du projet.

Si vous avez le rôle de contributeur mais que vous ne pouvez pas envoyer de code à une branche spécifique, une règle de branche configurée pour cette branche spécifique empêche les utilisateurs ayant ce rôle de transmettre du code à cette branche en particulier. Essayez d'appliquer vos modifications à une autre branche, ou créez une branche, puis transférez votre code vers cette branche. Pour en savoir plus, consultez [Gérer les actions autorisées pour une branche à l'aide de règles de branche](#).

## Mon environnement de développement n'a pas repris

Problème : mon environnement de développement n'a pas repris après son arrêt.

Solutions possibles : pour résoudre le problème, consultez l'onglet Facturation de l' CodeCatalyst espace Amazon pour voir si les quotas d'utilisation ont atteint les limites maximales. Si les quotas ont atteint la limite maximale, contactez l'administrateur de votre espace pour augmenter le niveau de facturation.

## Mon environnement de développement est déconnecté

Problème : mon environnement de développement s'est déconnecté pendant que je l'utilisais.

Solutions possibles : Pour résoudre le problème, vérifiez votre connexion Internet. Si vous n'êtes pas connecté à Internet, connectez-vous et reprenez le travail dans votre environnement de développement.

## Mon environnement de développement connecté au VPC a échoué

Problème : j'ai associé une connexion VPC à mon environnement de développement et des erreurs se produisent.

Correctifs possibles : Docker utilise un dispositif de couche de liaison appelé réseau de ponts qui permet aux conteneurs connectés au même réseau de ponts de communiquer. Le pont par défaut utilise généralement le sous-réseau 172.17.0.0/16 pour la mise en réseau des conteneurs. Si le sous-réseau du VPC de l'instance de votre environnement utilise la même plage d'adresses que celle utilisée par Docker, un conflit d'adresse IP peut se produire. Pour résoudre un conflit d'adresses IP provoqué par Amazon VPC et Docker utilisant le même bloc d'adresse CIDR IPv4, configurez un bloc d'adresse CIDR différent de 172.17.0.0/16

### Note

Vous ne pouvez pas modifier la plage d'adresses IP d'un VPC ou d'un sous-réseau existant.

## Je ne trouve pas le répertoire dans lequel se trouve mon projet

Problème : je ne trouve pas le répertoire dans lequel se trouve mon projet.

Corrections possibles : Pour localiser votre projet, remplacez le répertoire par /projects. Il s'agit du répertoire dans lequel vous pouvez trouver vos projets.

## Je ne parviens pas à me connecter à mon environnement de développement via SSH

Pour résoudre votre problème de connexion à votre environnement de développement via SSH, vous pouvez exécuter la ssh commande avec l'-vvv option permettant d'afficher plus d'informations sur la manière de résoudre votre problème :

```
ssh -vvv codecatalyst-dev-env=<space-name>=<project-name>=<dev-environment-id>
```

## Je ne parviens pas à me connecter à mon environnement de développement via SSH car ma configuration SSH locale est manquante

Si votre configuration SSH locale (~/.ssh/config) est manquante ou si le contenu de la Host codecatalyst-dev-env\* section est obsolète, vous ne pourrez pas vous connecter à votre environnement de développement via SSH. Pour résoudre ce problème, supprimez la Host codecatalyst-dev-env\* section et exécutez à nouveau la première commande depuis le modal SSH Access. Pour en savoir plus, consultez [Connexion à un environnement de développement via SSH](#).



# Je ne parviens pas à me connecter à mon environnement de développement via SSH car je rencontre des problèmes avec mon AWS Config profil **codecatalyst**

Assurez-vous que votre AWS Config (~/.aws/config) pour le codecatalyst profil correspond à celui décrit dans [Configuration pour utiliser AWS CLI le CodeCatalyst](#). Si ce n'est pas le cas, supprimez le profil pour codecatalyst et exécutez à nouveau la première commande depuis le modal SSH Access. Pour plus d'informations, consultez [Connexion à un environnement de développement via SSH](#).

## Résolution des problèmes liés aux IDE

Consultez les sections suivantes pour résoudre les problèmes liés aux IDE dans CodeCatalyst. Pour plus d'informations sur les IDE, consultez [Utilisation d'environnements de développement avec un IDE](#).

### Rubriques

- [Les versions des images d'exécution ne correspondent pas dans AWS Cloud9](#)
- [Je ne peux pas accéder à mes fichiers /projects/projects dans AWS Cloud9](#)
- [Je ne parviens pas à lancer mon environnement de développement en AWS Cloud9 utilisant un fichier de développement personnalisé](#)
- [J'ai des problèmes avec AWS Cloud9](#)
- [Dans JetBrains, je ne peux pas me connecter à mes environnements de développement via CodeCatalyst](#)
- [Je n'arrive pas à installer AWS Toolkit pour mon IDE](#)
- [Dans mon IDE, je ne peux pas lancer mes environnements de développement](#)

## Les versions des images d'exécution ne correspondent pas dans AWS Cloud9

AWS Cloud9 utilise différentes versions de l'actif frontal et de l'image d'exécution du backend. L'utilisation de versions différentes peut entraîner le mauvais fonctionnement de AWS Toolkit l'extension Git. Pour résoudre le problème, accédez au tableau de bord de l'environnement de développement, arrêtez votre environnement de développement, puis redémarrez-le. Pour résoudre le problème à l'aide des API, utilisez l'UpdateDevEnvironmentAPI pour mettre à jour

le moteur d'exécution. Pour plus d'informations, consultez [UpdateDevEnvironment](#) la référence des CodeCatalyst API Amazon.

## Je ne peux pas accéder à mes fichiers `/projects/projects` dans AWS Cloud9

L'AWS Cloud9 éditeur n'est pas en mesure d'accéder aux fichiers du répertoire `/projects/projects`. Pour résoudre le problème, utilisez le AWS Cloud9 terminal pour accéder à vos fichiers ou déplacez-les vers un autre répertoire.

## Je ne parviens pas à lancer mon environnement de développement en AWS Cloud9 utilisant un fichier de développement personnalisé

L'image de votre fichier de développement n'est peut-être pas compatible avec AWS Cloud9. Pour résoudre le problème, consultez le fichier de développement de votre référentiel et de l'environnement de développement correspondant, puis créez-en un nouveau pour continuer.

## J'ai des problèmes avec AWS Cloud9

Pour les autres problèmes, consultez la section de résolution des problèmes dans le [guide de AWS Cloud9 l'utilisateur](#).

## Dans JetBrains, je ne peux pas me connecter à mes environnements de développement via CodeCatalyst

Pour résoudre le problème, vérifiez que seule la dernière version de JetBrains est installée. Si vous avez plusieurs versions, désinstallez les anciennes versions et enregistrez à nouveau votre gestionnaire de protocole en fermant l'IDE et le navigateur. Ouvrez JetBrains et enregistrez à nouveau le gestionnaire de protocole.

## Je n'arrive pas à installer AWS Toolkit pour mon IDE

Pour résoudre ce problème pour VS Code, installez manuellement AWS Toolkit for Visual Studio Code à partir de [GitHub](#).

Pour résoudre ce problème JetBrains, installez manuellement AWS Toolkit for JetBrains à partir de [GitHub](#).

## Dans mon IDE, je ne peux pas lancer mes environnements de développement

Pour résoudre ce problème pour VS Code, vérifiez que vous disposez de la dernière version de VS Code et que vous l'AWS Toolkit for Visual Studio Code avez installée. Si vous ne disposez pas

de la dernière version, mettez à jour et lancez votre environnement de développement. Pour plus d'informations, consultez [Amazon CodeCatalyst pour VS Code](#).

Pour résoudre ce problème JetBrains, vérifiez que vous disposez de la dernière version de JetBrains et que vous l'AWS Toolkit for JetBrains avez installée. Si vous ne disposez pas de la dernière version, mettez à jour et lancez votre environnement de développement. Pour plus d'informations, consultez [Amazon CodeCatalyst pour JetBrains](#).

## Résolution des problèmes liés aux fichiers de développement

Consultez les sections suivantes pour résoudre les problèmes liés aux fichiers de développement dans CodeCatalyst. Pour plus d'informations sur les fichiers de développement, consultez [Configuration de votre environnement de développement](#).

### Rubriques

- [Mon environnement de développement utilise le fichier de développement universel par défaut, même si j'ai implémenté une image personnalisée dans un fichier de développement personnalisé.](#)
- [Mon projet ne se construit pas dans mon environnement de développement avec le fichier de développement universel par défaut](#)
- [Je souhaite déplacer un fichier de développement de référentiel pour un environnement de développement](#)
- [Je ne parviens pas à démarrer mon devfile](#)
- [Je ne sais pas comment vérifier le statut de mon fichier de développement](#)
- [Mon fichier de développement n'est pas compatible avec les outils fournis dans la dernière image](#)

Mon environnement de développement utilise le fichier de développement universel par défaut, même si j'ai implémenté une image personnalisée dans un fichier de développement personnalisé.

Si vous CodeCatalyst rencontrez des erreurs lors du démarrage d'un environnement de développement utilisant un fichier de développement personnalisé, l'environnement de développement utilise par défaut le fichier de développement universel par défaut. Pour résoudre le problème, vous pouvez vérifier l'erreur exacte dans les journaux ci-dessous `/aws/mde/logs/devfile.log`. Vous pouvez également vérifier si `postStart` l'exécution a été réussie dans vos journaux `:/aws/mde/logs/devfileCommand.log`.

## Mon projet ne se construit pas dans mon environnement de développement avec le fichier de développement universel par défaut

Pour résoudre le problème, vérifiez que vous n'utilisez pas de fichier de développement personnalisé. Si vous n'utilisez pas de fichier de développement personnalisé, consultez le `devfile.yaml` fichier dans le référentiel source du projet pour localiser et corriger les éventuelles erreurs.

## Je souhaite déplacer un fichier de développement de référentiel pour un environnement de développement

Vous pouvez déplacer le fichier de développement par défaut dans `/projects/devfile.yaml` votre référentiel de code source. Pour mettre à jour l'emplacement du fichier de développement, utilisez la commande suivante : `/aws/mde/mde start --location repository-name/devfile.yaml`.

## Je ne parviens pas à démarrer mon devfile

En cas de problème lors du démarrage de votre fichier de développement, celui-ci passe en mode de restauration afin que vous puissiez toujours vous connecter à votre environnement et corriger votre fichier de développement. En mode de restauration, l'exécution `/aws/mde/mde status` ne contiendra pas l'emplacement de votre fichier de développement.

```
{
  "status": "STABLE"
}
```

Vous pouvez vérifier l'erreur dans les journaux ci-dessous `/aws/mde/logs`, corriger le fichier de développement et réessayer de l'exécuter `/aws/mde/mde start`.

## Je ne sais pas comment vérifier le statut de mon fichier de développement

Vous pouvez vérifier l'état de votre fichier de développement en exécutant `/aws/mde/mde status`. Après avoir exécuté cette commande, l'un des éléments suivants peut s'afficher :

- `{"status": "STABLE", "location": "devfile.yaml" }`

Cela indique que votre fichier de développement est correct.

- `{"status": "STABLE" }`

Cela indique que votre fichier de développement n'a pas pu démarrer et qu'il est entré en mode de restauration.

Vous pouvez vérifier l'erreur exacte dans les journaux ci-dessous `/aws/mde/logs/devfile.log`.

Vous pouvez également vérifier si `postStart` l'exécution a été réussie dans vos journaux `:/aws/mde/logs/devfileCommand.log`.

Pour en savoir plus, consultez [Images de fichier universelles](#).

## Mon fichier de développement n'est pas compatible avec les outils fournis dans la dernière image

Dans votre environnement de développement, `devfile` ou `devfile postStart` peut échouer si l'latest outillage ne dispose pas de l'outillage requis pour un projet spécifique. Pour résoudre le problème, procédez comme suit :

1. Accédez à votre `devfile`.
2. Dans votre fichier de développement, passez à une version d'image granulaire au lieu de `latest`. Cela peut ressembler à ce qui suit :

```
components:
  - container:
      image: public.ecr.aws/amazonlinux/universal-image:1.0
```

3. Créez un nouvel environnement de développement à l'aide du fichier de développement mis à jour.

## Résolution des problèmes liés aux problèmes

Les informations suivantes peuvent vous aider à résoudre les problèmes courants liés aux problèmes liés à CodeCatalyst.

### Rubriques

- [Je ne parviens pas à choisir un destinataire pour mon problème](#)

## Je ne parviens pas à choisir un destinataire pour mon problème

Problème : lors de la création d'un problème, la liste des personnes assignées est vide.

Corrections possibles : La liste des destinataires est directement liée aux CodeCatalyst utilisateurs répertoriés en tant que membres du projet. Pour vérifier que l'accès au profil utilisateur fonctionne correctement, cliquez sur l'icône du profil, puis sur Profil utilisateur. Si les informations du profil utilisateur ne sont pas renseignées, consultez le rapport de santé pour détecter tout incident. S'il est renseigné, déposez un ticket de service.

## Résolution des problèmes entre Amazon CodeCatalyst et AWS les SDK ou AWS CLI

Les informations suivantes peuvent vous aider à résoudre les problèmes courants liés à l'utilisation AWS CLI ou à CodeCatalyst l'utilisation des AWS SDK.

### Rubriques

- [Je reçois un message d'erreur lorsque j'entre aws codecatalyst sur une ligne de commande ou un terminal indiquant que le choix n'est pas valide](#)
- [Je reçois une erreur d'identification lorsque j'exécute aws codecatalyst des commandes](#)

## Je reçois un message d'erreur lorsque j'entre aws codecatalyst sur une ligne de commande ou un terminal indiquant que le choix n'est pas valide

Problème : Lorsque j'essaie d'utiliser le AWS CLI with CodeCatalyst, une ou plusieurs aws codecatalyst commandes ne sont pas reconnues comme valides.

Solution : La cause la plus courante de ce problème est que vous utilisez une version du AWS CLI qui ne contient pas les mises à jour les plus récentes pour les derniers services et commandes. Mettez à jour votre installation du, AWS CLI puis réessayez. Pour de plus amples informations, veuillez consulter [Configuration pour utiliser AWS CLI le CodeCatalyst](#).

## Je reçois une erreur d'identification lorsque j'exécute aws codecatalyst des commandes

Problème : Lorsque j'essaie d'utiliser le AWS CLI with CodeCatalyst, je reçois un message indiquant `You can configure credentials by running "aws configure". ou Unable to locate authorization token.`

Solution : Vous devez configurer un AWS CLI profil pour fonctionner avec CodeCatalyst les commandes. Pour plus d'informations, consultez [Configuration pour utiliser AWS CLI le CodeCatalyst](#).

# CodeCatalyst rapport de santé

Le rapport de CodeCatalyst santé d'Amazon est un tableau de bord public qui fournit aux utilisateurs une liste agrégée de up-to-the-minute notifications concernant les performances des ressources et la disponibilité des services CodeCatalyst ayant un impact important. Vous pouvez voir quelles ressources rencontrent des problèmes et peuvent affecter les applications dans CodeCatalyst. Cela vous permet de suivre les pannes et autres interruptions de ressources à l'échelle du système. Lorsqu'un incident survient, un indicateur bleu apparaît sur l'icône du rapport de santé. En outre, envoie CodeCatalyst automatiquement une alerte et une notification par e-mail à tous les utilisateurs ayant le rôle d'administrateur de Space dans le projet, fournissant des détails et un historique de l'incident en temps quasi réel.

Le tableau de bord fournit une liste de tous les événements actifs et un enregistrement de 100 incidents antérieurs survenus au cours des 30 derniers jours. Vous pouvez organiser la liste des incidents en fonction de la date de mise à jour de l'incident. Vous pouvez également actualiser la liste des incidents pour obtenir des mises à jour actualisées.

Voici un flux de travail possible pour utiliser le rapport CodeCatalyst de santé :

Mateo Jackson est un développeur du Budding Space avec des autorisations d'administrateur de Space. Lors de la tentative de création d'une pull request, il ne cesse de recevoir un message d'erreur. Il consulte ses e-mails et découvre qu'il a reçu un e-mail d'incident système généré automatiquement, CodeCatalyst fournissant un historique détaillé du problème système affectant son espace. Il choisit Afficher la mise à jour et est redirigé vers le rapport CodeCatalyst de santé où il peut consulter tous les incidents signalés par le système. Il choisit l'incident dans la liste pour en savoir plus. Un écran partagé s'ouvre et fournit un horodatage de la dernière mise à jour, l'historique, les fonctionnalités concernées, l'heure de début et l'état actuel de l'incident. Il peut également constater que le problème persiste, mais l'équipe de service a commencé à y travailler. Chaque fois que l'historique ou le statut de l'incident sont mis à jour, il reçoit un e-mail. S'il n'a pas accès à son courrier électronique, il peut choisir l'icône en forme de cloche dans le panneau supérieur pour accéder au bilan de CodeCatalyst santé.

## CodeCatalyst concepts de rapport de santé

L'apprentissage des concepts suivants vous aidera à comprendre le rapport de CodeCatalyst santé et à comprendre comment il vous permet de suivre l'état de vos applications, services et ressources.



## Incident

L'incident est l'événement du système qui affecte les applications et les ressources qu'il contient CodeCatalyst. Vous pouvez choisir l'incident pour afficher un historique détaillé de l'événement, y compris l'heure à laquelle il a commencé et si l'équipe de service travaille à sa résolution.

## Statut

Le statut est le statut en temps réel de l'incident. Il s'affichera comme En cours ou Résolu.

## Fonctionnalités impactées

Les fonctionnalités touchées sont les ressources ou les applications touchées par l'incident. Un seul incident peut affecter plusieurs domaines du système, notamment les pull requests, les problèmes, les flux de travail, les tests, le déploiement et la source.

## Mis à jour le

Updated on fournit un horodatage de la dernière mise à jour relative à l'incident.

# AWS Support pour Amazon CodeCatalyst

Lorsque vous créez un espace, vous devez en connecter un Compte AWS et le désigner comme compte de facturation pour votre espace. Le compte que Compte AWS vous désignez comme compte de facturation est également l'endroit où vous accédez à votre AWS Support plan pour Amazon CodeCatalyst. Si vous avez besoin d'assistance, vous pouvez créer des dossiers d'assistance à partir de ce formulaire désigné Compte AWS.

CodeCatalyst les utilisateurs d'un espace utilisent la CodeCatalyst page AWS Support for Amazon CodeCatalyst pour gérer les demandes d'assistance. Vous pouvez passer à un AWS Support plan tel que Business Support ou Enterprise Support pour créer et gérer des demandes de support CodeCatalyst technique dans CodeCatalyst. Support disponible par téléphone, sur le Web ou par chat pour les cas d'assistance.

Seuls les cas spécifiques au CodeCatalyst service et aux ressources peuvent être pris en AWS Support charge par Amazon CodeCatalyst. CodeCatalyst les ressources incluent les ressources déployées au sein CodeCatalyst et par les utilisateurs dans CodeCatalyst, mais elles n'incluent pas les ressources déployées pour des services tiers AWS ou autres. Si vous avez besoin d'assistance pour AWS un autre service, vous devez l'ouvrir via le AWS Management Console.

Pour modifier votre plan de support, consultez la section [Modification des plans d'assistance](#).

## Note


Les plans de Support aux développeurs ne sont pas conçus pour les environnements de production. Si un compte de facturation d'espace dispose d'un plan de Support aux développeurs, ce plan ne s'applique pas à tous les administrateurs et membres de l'espace AWS Support concernés CodeCatalyst.

## Facturation AWS Support pour Amazon CodeCatalyst

Lorsque vous créez un espace dans CodeCatalyst, les utilisateurs de l'espace peuvent créer et gérer des demandes d'assistance AWS Support pour Amazon CodeCatalyst. Vous pouvez créer deux types de dossiers clients :


- Les demandes d'assistance relatives aux comptes et à la facturation sont accessibles à tous les CodeCatalyst utilisateurs de l'espace. Vous pouvez obtenir de l'aide pour les questions relatives à la facturation et au compte en fonction de vos autorisations dans CodeCatalyst.
- Les dossiers de support technique vous mettent en contact avec un ingénieur du support technique pour obtenir de l'aide en cas de problèmes techniques liés au service et d'extensions d'applications tierces. Si vous avez un plan de support Basic, vous ne pouvez pas créer de cas de support technique.

Le compte Compte AWS désigné comme compte de facturation pour l'espace doit disposer d'un plan Business Support ou Enterprise Support pour l'espace à utiliser AWS Support CodeCatalyst pour les cas techniques.

 Note

Si votre espace est utilisé AWS Support pour Amazon CodeCatalyst à partir d'un compte ne disposant pas d'un plan Business Support ou Enterprise Support, vous pouvez toujours l'utiliser AWS Support pour Amazon CodeCatalyst pour les demandes de compte et de facturation.

Pour le support technique, vous devez ouvrir tous les dossiers via la CodeCatalyst console. Vous ne pouvez pas créer de dossiers de support technique pour CodeCatalyst from [AWS Support](#) in the AWS Management Console.


 Note

Les demandes d'augmentation des limites de service ne sont pas disponibles auprès AWS Support d'Amazon CodeCatalyst. Ces demandes ne peuvent être soumises que par l'utilisateur root pour le compte de facturation d'espace dans le AWS Support Center Console.

AWS Support for Amazon CodeCatalyst dispose des mêmes contrats de support que AWS Support, avec les considérations suivantes :


- Les listes de gravité, les temps de réponse et les SLA AWS Support s'appliquent aux demandes d'assistance CodeCatalyst, comme indiqué dans la section [Choix d'une gravité](#). AWS Support
- Les administrateurs de l'espace et les membres de l'espace ne peuvent pas utiliser les AWS Support API, le AWS SDK ou AWS Support l'application de Slack pour créer des dossiers

pour. CodeCatalyst CodeCatalyst les dossiers de support ne peuvent être soumis qu'à partir de CodeCatalyst.

 Note

CodeCatalyst n'est pas totalement intégré à AWS Trusted Advisor AWS Incident Detection and Response. Validez CodeCatalyst le mode d'intégration pour vous assurer que vos pratiques commerciales sont conformes à l'intégration actuelle.

Vous devez être un utilisateur de l'espace dans lequel vous souhaitez demander de l'aide.

 Note

Si vous avez plusieurs constructeurs dans votre espace, nous vous recommandons de souscrire un plan Business Support ou Enterprise Support. Ces plans fournissent un soutien technique pour l'espace à un maximum de 5 000 constructeurs.

Le compte Compte AWS désigné comme compte de facturation pour l'espace utilise le `AWSRoleForCodeCatalystSupport` rôle et la politique [AmazonCodeCatalystSupportAccess](#) gérée. Cela permet CodeCatalyst aux utilisateurs d'un espace d'accéder à la CodeCatalyst page AWS Support pour Amazon. Pour plus d'informations sur ce rôle et cette politique, consultez [AmazonCodeCatalystSupportAccess](#). Pour d'autres considérations relatives à la facturation, consultez [la section Gestion de la facturation](#) dans le manuel Amazon CodeCatalyst Administrator Guide.

Voici un flux possible pour un constructeur qui crée un dossier de support dans CodeCatalyst :

Mateo Jackson est développeur sur un projet en CodeCatalyst. Après avoir souscrit au service Compte AWS de gestion de la facturation AWS Support pour Amazon CodeCatalyst et passé à un plan de support aux entreprises, tous les créateurs de l'espace peuvent créer des dossiers de support technique. Mateo soumet un dossier de support technique concernant l'échec d'un flux de travail dans le cadre de son projet. Mateo utilise la CodeCatalyst page AWS Support pour Amazon pour remplir le formulaire et créer un dossier, en fournissant l'identifiant du flux de travail et d'autres informations dans la demande. Le dossier est créé avec un numéro de dossier et inclut l'identifiant de compte du compte Compte AWS désigné comme compte de facturation et associé au plan de support pour l'espace.

Bien que tous les constructeurs puissent créer des demandes d'assistance dans AWS Support for CodeCatalyst, vous n'êtes pas facturé pour chaque demande créée. Vous pouvez ouvrir un nombre pratiquement illimité de dossiers et de contacts en fonction du plan AWS Support Premium que vous achetez sur votre compte de facturation d'espace.

#### Note

Le compte de facturation de l'espace est Compte AWS celui qui vous est facturé pour CodeCatalyst les utilisateurs et les ressources. Si vous avez effectué un déploiement vers d'autres servicesComptes AWS, contactez AWS Support via le AWS Management Console pour obtenir de l'aide concernant les ressources déployées vers d'autres services. Vous pouvez identifier celui vers lequel Compte AWS vous avez effectué le déploiement à partir du flux de travail.

## Configuration de votre espace AWS Support pour Amazon CodeCatalyst

AWS Supportfor Amazon CodeCatalyst gère les demandes de support dans le cadre de l'intégration de l'AWS SupportAPI avec CodeCatalyst.

Le `AWSRoleForCodeCatalystSupport` rôle est un rôle de service utilisé pour les demandes d'assistance dans votre espace. Le rôle doit être ajouté au compte de facturation désigné pour l'espace. Pour plus d'informations ou pour créer le rôle, consultez [Création du `AWSRoleForCodeCatalystSupport`rôle pour votre compte et votre espace](#).

#### Note

Pour un espace créé avant le 20 avril 2023, vous devez créer le rôle afin que le support fonctionne CodeCatalyst pour votre espace. Si vous créez un espace après le 20 avril 2023, vous pouvez créer le rôle lors de la création de l'espace, sur la page Informations de facturation dans CodeCatalyst, ou en cliquant sur le lien de la bannière d'assistance dans CodeCatalyst.

## Pour configurer le support pour votre espace

1. Lorsque vous créez un CodeCatalyst espace, il vous est demandé de connecter un compte de facturation. Le compte de facturation désigné pour l'espace sera facturé par AWS. Pour plus d'informations sur la création d'un espace, consultez [Inscrivez-vous pour créer votre premier espace et votre rôle de développeur](#).
2. Lorsque vous créez un CodeCatalyst espace, l'option est disponible pour créer le rôle de `AWSRoleForCodeCatalystSupport` service qui permet aux CodeCatalyst utilisateurs d'accéder au support. Le rôle utilise la politique gérée `AmazonCodeCatalystSupportAccess`. Le rôle doit être ajouté au compte Compte AWS désigné comme compte de facturation pour l'espace. Pour plus d'informations sur la création de ce rôle, consultez la page [Création du AWSRoleForCodeCatalystSupport rôle pour votre compte et votre espace](#).
3. Pour le compte de facturation désigné pour l'espace, il est recommandé à l'administrateur de l'espace de souscrire un plan Business Support ou Enterprise Support pour le Compte AWS. Tous les membres de l'espace seront en mesure de gérer les demandes d'assistance auprès AWS Support d'Amazon CodeCatalyst, et les canaux d'assistance seront alignés sur le AWS Support plan que vous avez acheté lorsque les intégrations seront terminées.
4. Pour créer et gérer des demandes de support dans CodeCatalyst, voir [Création d'un dossier de CodeCatalyst support dans CodeCatalyst](#).

## Accès au support pour CodeCatalyst AWS Management Console

Si le compte de facturation activé pour un espace est déconnecté, AWS Support les demandes associées au compte de facturation d'espace précédent et au plan de support associé ne seront plus visibles sur Amazon CodeCatalyst. AWS Support L'utilisateur root de ce compte de facturation peut consulter et résoudre les anciens dossiers à partir du AWS Management Console et peut configurer des autorisations IAM AWS Support permettant aux autres utilisateurs de consulter et de résoudre les anciens dossiers. Vous pourrez toujours bénéficier des avantages de votre plan d'assistance par rapport à tous les autres Services AWS et terminer les AWS Management Console demandes d'CodeCatalyst assistance qui n'ont pas été résolues auparavant.

Pour plus d'informations, voir [Mettre à jour, résoudre et rouvrir votre dossier](#) dans le Guide de l'AWS Support utilisateur.

Support pour des informations pratiques générales CodeCatalyst peuvent également être ouvertes dans le AWS Management Console, mais aucune assistance technique ne peut être reçue via ce

canal pour. CodeCatalyst Pour plus d'informations, consultez les sections [Création de dossiers de support et gestion de dossiers](#) dans le Guide de AWS Support l'utilisateur.

Voici un flux possible pour un utilisateur résolvant une demande d'assistance CodeCatalyst dans le cas de AWS Management Console :

Bien que tous les constructeurs puissent créer des dossiers d'assistance avec AWS Support for Amazon CodeCatalyst, les demandes d'assistance sont facturées à partir du compte désigné comme compte de facturation pour l'espace. Mateo Jackson est un développeur de projet CodeCatalyst qui a ouvert un dossier de support technique pour un flux de travail défaillant dans son projet. Cependant, le compte de facturation associé AWS Support à l'espace réservé à Amazon CodeCatalyst et à l'achat d'un plan Business Support a été déconnecté de l'espace. La seule façon pour Mateo de consulter les dernières communications et de résoudre les dossiers ouverts CodeCatalyst est de gérer l'identifiant du dossier depuis le AWS Support Centre dans le AWS Management Console. Pour ce faire, Mateo reçoit des autorisations IAM de la part de l'utilisateur root du compte de facturation d'espace précédent joint à son dossier d'assistance et résout le problème via la AWS Support console.

#### Important

Si vous modifiez le compte de facturation désigné pour votre espace, votre AWS Support plan sera toujours accessible jusqu'à la fin du mois AWS Management Console uniquement. Vous devrez effectuer un rachat AWS Support sur le compte de facturation mis à jour pour continuer à accéder à vos demandes d'assistance créées précédemment dans CodeCatalyst. Nous vous recommandons d'attendre d'avoir résolu tous vos dossiers d'assistance pour changer de compte de facturation d'espace afin d'éviter tout impact sur l'accès à vos demandes d'assistance par le biais AWS Support d'Amazon CodeCatalyst.

## Création d'un dossier de CodeCatalyst support dans CodeCatalyst

Vous pouvez créer un dossier d'assistance sur la CodeCatalyst page AWS Support dédiée à Amazon.

Un AWS Builder ID ne peut être pris en charge que pour l'alias avec lequel il est authentifié et uniquement pour les ressources en fonction de ses autorisations. Les options de compte et de facturation sont disponibles pour tous les administrateurs et membres de l'espace. Toutefois, les

utilisateurs ne peuvent obtenir de l'assistance que pour les ressources auxquelles ils ont accès CodeCatalyst et non pour ce qui est de la gestion de la facturation du compte.

Vous pouvez créer un dossier de compte et de facturation ou un dossier de support technique pour vos CodeCatalyst ressources en utilisant la AWS Support CodeCatalyst page d'accueil de votre espace.

#### Note

Seuls les cas spécifiques au CodeCatalyst service et aux ressources peuvent être pris en AWS Support charge par Amazon CodeCatalyst. CodeCatalyst les ressources incluent les ressources déployées au sein CodeCatalyst et par les utilisateurs dans CodeCatalyst, mais elles n'incluent pas les ressources déployées pour des services tiers AWS ou autres. Si vous avez besoin d'assistance pour AWS un autre service, vous devez l'ouvrir via le AWS Management Console.

Pour créer un dossier de support dans CodeCatalyst

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.

#### Tip

Si vous appartenez à plusieurs espaces, choisissez-en un dans la barre de navigation supérieure.

3. En haut de la page, sélectionnez ? icône, puis choisissez Support.
4. Choisissez Create case (Créer une demande).
5. Choisissez l'une des options suivantes :
  - Account and billing (Compte et facturation)
  - Technical (Technique)

#### Note


AWS Support Pour Amazon CodeCatalyst, si un plan Business Support ou Enterprise Support est ajouté au compte de facturation de l'espace, le support CodeCatalyst



technique sera disponible pour tous les administrateurs de l'espace et les membres de l'espace. Pour plus d'informations sur le dépannage, consultez [Je ne peux pas créer de dossiers de support technique pour mon espace](#).

AWS Supportles plans ne s'étendent pas sur plusieurs espaces. Si vous êtes membre de plusieurs espaces, l'administrateur de votre espace devra acheter un plan AWS Support Premium pour chaque espace afin de bénéficier d'une assistance technique pour tous les espaces.


6. Choisissez le Service, la Category (Catégorie), et la Severity (Sévérité). Pour plus d'informations sur le choix d'une sévérité, voir [Choisir une sévérité](#).
  - General guidance
  - System impaired
  - Production system impaired
  - Production system down
  - Business-critical system down
7. Choisissez Next step: Additional information (Étape suivante : informations supplémentaires).
8. Sur la page Additional information (Informations supplémentaires), pour Subject (Sujet), saisissez un titre concernant votre problème.
9. Pour la Description, suivez les invites pour décrire votre cas, comme par exemple les suivantes :
  - Informations de dépannage spécifiques CodeCatalyst, telles que l'identifiant du flux de travail, les journaux ou les captures d'écran
  - Messages d'erreur que vous avez reçus
  - Étapes de dépannage que vous avez suivies

 Note

Ne partagez aucune information sensible dans les correspondances, telles que les informations d'identification, les cartes de crédit, les URL signées ou les informations personnelles identifiables.

10. (Facultatif) Choisissez Attach files (Joindre des fichiers) pour ajouter des fichiers pertinents à votre cas, tels que des journaux d'erreurs ou des captures d'écran. Vous pouvez attacher jusqu'à 3 fichiers. La taille de chaque fichier peut aller jusqu'à 5 Mo.

11. Dans Nom de l'espace, le nom de votre espace s'affiche.
12. Dans Nom du constructeur, le nom complet associé à votre identifiant de AWS constructeur est automatiquement renseigné.
13. (Facultatif) Choisissez le projet dans Nom du projet (le cas échéant).

 Note

Seuls les projets pour lesquels vous avez des autorisations seront affichés. Si vous avez besoin d'accéder à un autre projet, demandez à l'administrateur de votre projet de vous fournir l'accès avant de créer un dossier d'assistance.

14. Choisissez l'étape suivante : Contactez-nous.
15. Dans Langue de contact préférée, choisissez la langue par défaut. Seul l'anglais est disponible pour le moment.
16. Choisissez l'option Web, téléphone ou chat pour la méthode de contact.
17. Vérifiez les détails de votre dossier, puis choisissez Soumettre. Votre numéro d'ID de dossier et votre résumé apparaissent.

Le dossier de support est créé au niveau de l'espace et est consultable par tous les membres ayant accès à l'espace et au projet (s'ils sont sélectionnés) définis dans votre dossier de support. Pour le moment, il n'est pas possible d'omettre un dossier d'assistance pour les utilisateurs individuels.

## Résolution d'un dossier d'assistance dans CodeCatalyst

Vous pouvez résoudre les demandes d'assistance en cours AWS Support depuis la CodeCatalyst page dédiée à Amazon.

Vous devez avoir un rôle d'administrateur de l'espace ou de membre de l'espace dans lequel vous souhaitez résoudre un dossier d'assistance. Si vous n'avez pas le rôle d'administrateur de l'espace, ou si un projet a été sélectionné lors de la création du dossier, vous devez également être membre du projet pour consulter et résoudre le dossier.

Pour résoudre un dossier de support ouvert dans CodeCatalyst

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.

**i** Tip

Si vous appartenez à plusieurs espaces, choisissez-en un dans la barre de navigation supérieure.

3. En haut de la page, sélectionnez ? icône, puis choisissez AWS SupportAmazon CodeCatalyst.
4. Choisissez le lien correspondant au dossier d'assistance que vous souhaitez gérer. Choisissez Resolve case (Résoudre le cas).

## Réouverture d'un dossier de support dans CodeCatalyst

Vous pouvez utiliser la fonctionnalité de réouverture d'un dossier d'assistance résolu AWS Support depuis la CodeCatalyst page dédiée à Amazon.

**i** Note

Vous pouvez rouvrir votre cas de support jusqu'à 14 jours à compter de la résolution de votre problème. Toutefois, vous ne pouvez pas rouvrir un cas inactif depuis plus de 14 jours. Si vous ne parvenez pas à rouvrir votre dossier, ouvrez-en un nouveau et incluez le numéro de dossier précédent comme référence.

Si vous rouvrez un cas existant qui contient des informations différentes de celles de votre problème actuel, l'agent de support peut vous demander de créer un nouveau cas.

Pour rouvrir un dossier de support dans CodeCatalyst

1. Ouvrez la CodeCatalyst console à l'[adresse https://codecatalyst.aws/](https://codecatalyst.aws/).
2. Accédez à votre CodeCatalyst espace.

**i** Tip

Si vous appartenez à plusieurs espaces, choisissez-en un dans la barre de navigation supérieure.

3. En haut de la page, sélectionnez ? icône, puis choisissez AWS Support pour CodeCatalyst.

4. Choisissez le lien correspondant au dossier d'assistance que vous souhaitez gérer. Sélectionnez Reopen (Rouvrir). Cliquez sur OK sur l'écran de confirmation, puis sur Soumettre.
5. Remplissez la description avec les dernières informations concernant le même problème. Ne partagez aucune information sensible dans les correspondances, telles que les informations d'identification, les cartes de crédit, les URL signées ou les informations personnelles identifiables.

# Quotas pour CodeCatalyst

Le tableau suivant décrit les quotas et les limites pour Amazon CodeCatalyst. Vous trouverez des informations supplémentaires sur des aspects spécifiques CodeCatalyst dans les rubriques suivantes :

- [Quotas pour les référentiels sources dans CodeCatalyst](#)
- [Quotas d'identité, d'autorisation et d'accès dans CodeCatalyst](#)
- [Quotas pour les flux de travail dans CodeCatalyst](#)
- [Quotas pour les environnements de développement dans CodeCatalyst](#)
- [Quotas pour les projets dans CodeCatalyst](#)
- [Quotas pour les plans dans CodeCatalyst](#)
- [Quotas de places dans CodeCatalyst](#)
- [Quotas pour les numéros parus dans CodeCatalyst](#)

|   |   |
|---|---|
| Nombre maximum d'espaces dans un compte   | 5   |
| Nombre maximum d'espaces qu'un utilisateur peut créer au cours d'un mois calendaire | 5   |
| Nombre minimum de Comptes AWS pour un espace  | 1   |
| Nombre maximum de connexions au compte pour un espace                               | 5 000   |
| Nombre maximal de connexions VPC pour un espace                                     | 100   |
| Nombre maximum de projets dans un espace  | 100   |
| Nombre maximum de projets auxquels un utilisateur peut appartenir                   | 1 000   |
| Descriptions des espaces  | Les descriptions des espaces sont facultatives. S'ils sont spécifiés, ils doivent comporter |

entre 0 et 200 caractères. Ils peuvent contenir n'importe quelle combinaison de lettres, de chiffres, d'espaces, de points, de traits de soulignement, de virgules, de tirets et les caractères spéciaux suivants :

? & \$ % + = / \ ; : \n \t \r

## Noms des espaces

Les noms des espaces doivent être uniques d'un côté à l'autre CodeCatalyst. Vous ne pouvez pas réutiliser les noms des espaces supprimés.

Les noms d'espaces doivent comporter entre 3 et 63 caractères. Ils doivent également commencer par un caractère alphanumérique.

Les noms d'espaces peuvent contenir n'importe quelle combinaison de lettres, de chiffres, de points, de traits de soulignement et de tirets. Ils ne peuvent contenir aucun des caractères suivants :

! ? @ # \$ % ^ & \* ( ) + = { } [ ]  
| / \ > < ~ ` ' " ; :

# Historique du document pour Amazon CodeCatalyst

Le tableau suivant décrit l'historique de la documentation et les mises à jour de la documentation globale de CodeCatalyst.

| Modification   | Description   | Date           |
|--|---|----------------|
| <a href="#">Nouveau contenu : utilisation de tâches pour décomposer les problèmes en objectifs plus petits</a> | Ajout de contenu pour faciliter le lancement de tâches en cas de problème. Des tâches peuvent être ajoutées aux problèmes pour mieux les décomposer, les organiser et en suivre le déroulement.   | 4 avril 2024   |
| <a href="#">Contenu mis à jour : Limites des GitHub actions dans CodeCatalyst</a>                              | Mise à jour de la <a href="#">Limites des GitHub actions dans CodeCatalyst</a> rubrique pour indiquer que GitHub les actions s'exécutent sur une image Docker d'environnement d'exécution plus ancienne.  | 2 avril 2024   |
| <a href="#">Contenu mis à jour : Travailler avec des artefacts</a>   | La <a href="#">Travailler avec des artefacts</a> rubrique a été mise à jour pour inclure deux nouveaux sous-sujets : et <a href="#">Puis-je partager des artefacts sans les spécifier en tant que sorties ou entrées ?</a> <a href="#">Puis-je partager des artefacts entre les flux de travail ?</a> | 2 avril 2024   |
| <a href="#">Nouveau contenu : référence d'action « AWS CDK déployer »</a>                                      | Ajout d'une nouvelle CloudAssemblyRootPath propriété au <a href="#">référence</a>   | 1er avril 2024 |

---

|   |   |              |
|---|---|--------------|
|   | <a href="#">d'action « AWS CDK déployer »</a> .   |              |
| <a href="#">Contenu mis à jour : Utilisation des images Docker de l'environnement d'exécution</a> | La <a href="#">Utilisation des images Docker de l'environnement d'exécution</a> rubrique a été mise à jour pour inclure des informations sur la nouvelle image de l'environnement d'exécution de mars 2024.   | 26 mars 2024 |
| <a href="#">Contenu mis à jour : utilisation des rôles</a>  | Consolidation des informations relatives aux autorisations relatives aux rôles dans un tableau unique. Le tableau figure dans une nouvelle <a href="#">Autorisations disponibles pour chaque rôle</a> rubrique.   | 18 mars 2024 |
| <a href="#">Nouveau contenu : Afficher tous les espaces et projets d'un utilisateur</a>           | Ajout d'informations concernant l'affichage d'une liste sur la page d'accueil de l'utilisateur qui montre chaque CodeCatalyst espace ou projet pour l'utilisateur connecté. CodeCatalyst veuillez consulter <a href="#">Afficher tous les espaces et projets d'un utilisateur</a> . | 18 mars 2024 |
| <a href="#">Nouveau contenu : Exemple : une gâchette avec une traction et des branches</a>        | Ajout d'un exemple de déclencheur de pull request. J'ai apporté de petites corrections tout au long du <a href="#">Utilisation de déclencheurs</a> sujet.   | 11 mars 2024 |



---

|  |  |                 |
|--|--|-----------------|
| <a href="#">Contenu mis à jour : utilisation des rôles</a>   | Mise à jour de la documentation relative aux rôles afin d'inclure les autorisations de création, de suppression et d'affichage d'environnements.                               | 4 mars 2024     |
| <a href="#">Contenu mis à jour : Tutoriel : Utilisation des fonctionnalités génératives de l'IA</a>            | Le didacticiel a été mis à jour pour refléter les modifications apportées lors de la création et de l'attribution de problèmes à Amazon Q.                                     | 4 mars 2024     |
| <a href="#">Nouveau contenu : bloquer les GitHub pull requests fusionne en cas d'échec des flux de travail</a> | Le nouveau contenu ajouté sur la façon de bloquer une GitHub pull request fusionne avec les règles de protection des succursales en cas d'échec des flux de travail.           | 4 mars 2024     |
| <a href="#">Nouveau contenu : Problèmes liés aux composants</a>  | Ajout de nouveau contenu sur la façon de travailler avec les composants des problèmes en tant que développeur de plans personnalisés.  | 27 février 2024 |
| <a href="#">Contenu mis à jour : Types d'action</a>  | La <a href="#">CodeCatalyst actions</a> rubrique a été mise à jour pour inclure une liste des actions de CodeCatalyst Labs.  | 21 février 2024 |
| <a href="#">Contenu mis à jour : utilisation des pull requests</a>   | La documentation a été mise à jour pour refléter les nouvelles fonctionnalités avec des règles d'approbation et des exigences de remplacement pour fusionner une pull request. | 15 février 2024 |

[Contenu mis à jour : fusion des pull requests](#)

Ajout de la documentation pour les pull requests afin d'inclure des informations sur le remplacement des exigences de fusion pour fusionner une pull request qui n'a pas encore reçu les approbations des réviseurs requis ou qui n'a pas encore satisfait aux règles d'approbation.

15 février 2024

[Nouveau contenu : gestion des règles d'approbation](#)

Ajout de la documentation pour les pull requests afin d'inclure des informations sur la création et la gestion des règles d'approbation.

15 février 2024

[Contenu mis à jour : utilisation des rôles](#)

Mise à jour de la documentation relative aux rôles afin d'inclure les autorisations nécessaires pour utiliser les règles d'approbation et les pull requests.

14 février 2024

[Contenu mis à jour : Comment corriger les erreurs « La définition du flux de travail ne comporte aucune erreur » ?](#)

La [Comment corriger les erreurs « La définition du flux de travail ne comporte aucune erreur » ?](#) section a été mise à jour pour inclure davantage de conseils de résolution des problèmes.

9 février 2024

[Nouveau contenu : Affichage de l'état du flux de travail](#)

Ajout d'une section qui décrit les statuts des flux de travail.

9 février 2024

|  |  |                  |
|--|--|------------------|
| <a href="#">Mettre à jour le contenu : Quotas pour les flux de travail dans CodeCatalyst</a> | Mise à jour de la <a href="#">Quotas pour les flux de travail dans CodeCatalyst</a> rubrique avec le nombre maximum d'actions par flux de travail et le nombre maximum d'environnements associés à un quota Compte AWS par espace. | 7 février 2024   |
| <a href="#">Contenu mis à jour : Création d'un environnement</a>                             | La <a href="#">Création d'un environnement</a> section a été mise à jour pour indiquer que vous pouvez utiliser un maximum d'une connexion à un compte par environnement.  | 31 janvier 2024  |
| <a href="#">Nouveau contenu : référentiel de plans GitHub personnalisés</a>                  | Ajout d'un nouveau contenu pour GitHub le référentiel mis à la disposition du public.  | 10 janvier 2024  |
| <a href="#">Contenu mis à jour : Configuration de npm avec CodeCatalyst</a>                  | Instructions de configuration générales mises à jour pour l'utilisation de npm with CodeCatalyst et clarification de l' <code>always-auth=true</code> option.  | 5 janvier 2024   |
| <a href="#">Contenu mis à jour : utilisation des pull requests</a>                           | Mise à jour de la documentation pour refléter les nouvelles fonctionnalités avec les fonctionnalités d'IA générative incluses dans CodeCatalyst.   | 28 novembre 2023 |

---

|   |   |                  |
|---|---|------------------|
| <a href="#">Contenu mis à jour : création d'un problème</a>   | Mise à jour de la documentation pour refléter les nouvelles fonctionnalités avec les fonctionnalités d'IA générative incluses dans CodeCatalyst.  | 28 novembre 2023 |
| <a href="#">Nouveau contenu : Tutoriel : Utilisation des fonctionnalités génératives de l'IA</a>                            | Ajout d'un didacticiel pour l'utilisation des fonctionnalités d'IA générative d'Amazon CodeCatalyst.  | 28 novembre 2023 |
| <a href="#">Nouveau contenu : plans personnalisés et gestion du cycle de vie</a>  | Ajout de nouveau contenu pour l'utilisation de plans personnalisés et de fonctionnalités de gestion du cycle de vie sur Amazon CodeCatalyst.  | 27 novembre 2023 |
| <a href="#">Contenu mis à jour : Tutoriel : Création d'un projet avec le plan d'application Web moderne à trois niveaux</a> | Le didacticiel a été mis à jour avec des correctifs et des informations de dépannage.   | 22 novembre 2023 |
| <a href="#">Contenu mis à jour : Utilisation de déclencheurs</a>  | Correction de quelques exemples et descriptions liés aux déclencheurs de pull request. Ajout d'une <a href="#">Considérations relatives aux déclencheurs lors du branchement</a> section. | 22 novembre 2023 |

|  |   |                  |
|--|---|------------------|
| <a href="#">Nouveau contenu : Connectez-vous avec SSO</a>  | Ajout d'informations sur la connexion avec l'authentification unique (SSO) et de liens vers des informations sur la configuration et la gestion d'un CodeCatalyst espace prenant en charge la fédération des identités. Consultez <a href="#">Configuration CodeCatalyst</a> et <a href="#">Connectez-vous avec SSO</a> . | 17 novembre 2023 |
| <a href="#">Contenu mis à jour : utilisation des rôles</a>   | Mise à jour de la documentation relative aux rôles afin d'inclure les autorisations pour travailler avec les équipes, les connexions VPC, l'authentification unique et les ressources des machines.   | 16 novembre 2023 |
| <a href="#">Contenu mis à jour : utilisation des pull requests</a>                                     | Mise à jour de la documentation pour refléter les modifications apportées à la façon dont les modifications relatives à une pull request sont affichées .   | 16 novembre 2023 |
| <a href="#">Contenu mis à jour : quotas pour CodeCatalyst</a>  | Mise à jour de la <a href="#">Quotas pour CodeCatalyst</a> rubrique avec le nombre maximum de connexions VPC pour un quota d'espace.  | 16 novembre 2023 |
| <a href="#">Nouveau contenu : Utilisation d'environnements de développement avec une connexion VPC</a> | Ajout de documentation pour l'utilisation des environnements de développement avec une connexion VPC.   | 16 novembre 2023 |

|  |  |                  |
|--|--|------------------|
| <a href="#">Nouveau contenu : Gestion d'équipes pour un espace et pour CodeCatalyst des projets</a>                                      | Ajout d'informations sur l'utilisation d'équipes avec des espaces. Consultez <a href="#">Gérer des équipes</a> et <a href="#">Gestion des équipes pour les projets</a> .     | 16 novembre 2023 |
| <a href="#">Nouveau contenu : gestion des ressources des machines pour les plans et les flux de travail dans un espace</a>               | Ajout d'informations sur l'utilisation des ressources de la machine avec des espaces. veuillez consulter <a href="#">Gestion des ressources des machines</a> .               | 16 novembre 2023 |
| <a href="#">Nouveau contenu : gestion des ressources informatiques pour les plans et les flux de travail dans un CodeCatalyst projet</a> | Ajout d'informations sur l'utilisation des ressources de la machine CodeCatalyst dans les projets. veuillez consulter <a href="#">Gestion des ressources de la machine</a> . | 16 novembre 2023 |
| <a href="#">Nouveau contenu : association d'une connexion VPC à un environnement</a>   | Ajout d'une documentation pour associer une connexion VPC à un environnement, qui peut être utilisée dans un flux de travail.  | 16 novembre 2023 |
| <a href="#">Nouveau contenu</a>  | Publication initiale du <a href="#">guide Amazon CodeCatalyst Administrator</a> .  | 16 novembre 2023 |
| <a href="#">Nouveau contenu : référence d'action « AWS CDK déployer »</a>  | Ajout d'une nouvelle CdkCliVersion propriété au <a href="#">référence d'action « AWS CDK déployer »</a> et au <a href="#">référence d'action « AWS CDK bootstrap »</a> .     | 14 novembre 2023 |

|   |   |                  |
|---|---|------------------|
| <a href="#">Contenu mis à jour : utilisation des rôles</a>  | Mise à jour de la documentation relative aux rôles afin d'inclure les autorisations permettant d'utiliser les règles de branche.  | 13 novembre 2023 |
| <a href="#">Contenu mis à jour : résolution des problèmes liés aux référentiels sources, aux flux de travail et aux environnements de développement</a> | Les rubriques de résolution des problèmes ont été mises à jour pour inclure des informations sur l'utilisation des règles de branche.   | 13 novembre 2023 |
| <a href="#">Contenu mis à jour : Créez et testez une référence d'actions</a>  | Mise à jour de la documentation de la Environment propriété. Il s'agit désormais d'un champ facultatif pour les actions de construction et de test.   | 13 novembre 2023 |
| <a href="#">Nouveau contenu : Gérer les règles des succursales</a>  | Ajout de documentation pour les branches afin d'inclure des informations sur l'affichage des règles applicables aux branches dans un référentiel source, ainsi que sur la création et la gestion des règles de branche. | 13 novembre 2023 |
| <a href="#">Contenu mis à jour : utilisation des pull requests</a>  | La documentation a été mise à jour pour refléter les modifications apportées à la façon dont les informations relatives à une pull request sont affichées.  | 10 novembre 2023 |

---

|  |   |                   |
|--|---|-------------------|
| <a href="#">Contenu mis à jour : Utilisation de la mise en cache de fichiers</a>             | Mise à jour de la documentation pour inclure les limites de mise en cache des fichiers.   | 10 novembre 2023  |
| <a href="#">Contenu mis à jour : Tutoriel : Déploiement d'une application sur Amazon EKS</a> | Mise à jour de la documentation pour mentionner le plan de déploiement de l'application EKS.  | 9 novembre 2023   |
| <a href="#">Nouveau contenu : Packages en CodeCatalyst</a>                                   | Ajout de la documentation pour l'utilisation des packages dans CodeCatalyst.  | 1er novembre 2023 |
| <a href="#">Contenu nouveau et mis à jour : utilisation des rôles</a>                        | Mise à jour de la documentation pour quatre nouveaux rôles CodeCatalyst : utilisateur avancé, accès limité, réviseur et lecture seule.  | 1er novembre 2023 |
| <a href="#">Contenu mis à jour : Utilisation des paramètres de sortie des GitHub actions</a> | Mise à jour des exemples pour utiliser le fichier d'GITHUB_OUTPUT environnement au lieu de la set-output commande. L'utilisation de fichiers d'environnement GitHub est la méthode recommandée pour définir les paramètres de sortie. | 24 octobre 2023   |
| <a href="#">Nouveau contenu : Utilisation de déclencheurs</a>                                | Ajout de documentation pour les déclencheurs de planification.  | 16 octobre 2023   |



|   |  |                   |
|---|--|-------------------|
| <a href="#">Contenu mis à jour :</a><br><a href="#">Référence d'action « Déployer vers un cluster Kubernetes »</a>                                | Ajout d'informations sur l'utilisation du CodeCatalystWorkflowDevelopmentRole- <i>spaceName</i> rôle dans les <a href="#">Tutoriel : Déploiement d'une application sur Amazon EKS</a> rubriques <a href="#">Référence d'action « Déployer vers un cluster Kubernetes »</a> et.   | 22 septembre 2023 |
| <a href="#">Contenu mis à jour : nouveau nom du rôle et nouvelle politique pour le CodeCatalystWorkflowDevelopmentRole- <i>spaceName</i> rôle</a> | Mise à jour des étapes et des descriptions de rôle pour le changement de nom du rôle de développeur en CodeCatalystWorkflowDevelopmentRole- <i>spaceName</i> . Le rôle de développeur utilise désormais la politique AdministratorAccess AWS gérée. Consultez <a href="#">Comprendre le rôle CodeCatalystWorkflowDevelopmentRole- <i>spaceName</i> du service</a> et <a href="#">Inscrivez-vous pour créer votre premier espace et votre rôle de développeur</a> . | 20 septembre 2023 |
| <a href="#">Contenu mis à jour : Utilisation des variables</a>  | Deux nouveaux concepts ont été introduits : les variables définies par l'utilisateur et les variables prédéfinies. Ces concepts devraient faciliter la lecture et la compréhension de la <a href="#">Utilisation des variables</a> section.  | 19 septembre 2023 |

|   |   |                  |
|---|---|------------------|
| <a href="#">Contenu mis à jour : utilisation des commits</a>                                    | La documentation a été mise à jour pour tenir compte de la modification des informations affichées et fournir des détails sur l'affichage des validations avec plusieurs parents.           | 7 septembre 2023 |
| <a href="#">Nouveau contenu : Utilisation de déclencheurs</a>                                   | L'exemple suivant a été ajouté à la <a href="#">Utilisation de déclencheurs</a> rubrique : <a href="#">Exemple : un simple déclencheur « push to main »</a>                                 | 6 septembre 2023 |
| <a href="#">Contenu mis à jour : utilisation des pull requests</a>                              | Mise à jour de la documentation pour refléter le changement d'ordre d'affichage de la branche source et de la branche de destination lors de la création d'une pull request.                | 30 août 2023     |
| <a href="#">Nouveau contenu : Afficher et modifier la branche par défaut</a>                    | Ajout d'une documentation pour les branches afin d'inclure des informations sur l'affichage et la modification de la branche par défaut d'un référentiel source.                            | 30 août 2023     |
| <a href="#">Contenu mis à jour : Référence d'action « Déployer vers un cluster Kubernetes »</a> | Ajout d'une note concernant Helm et Kustomize à la description de la <a href="#">Manifests</a> propriété dans le <a href="#">Référence d'action « Déployer vers un cluster Kubernetes »</a> | 15 août 2023     |

|  |   |                 |
|--|---|-----------------|
| <a href="#">Nouveau contenu : Gestion des pièces jointes aux problèmes</a>                           | Ajout de documentation pour travailler avec et gérer les pièces jointes en cas de problème.   | 15 août 2023    |
| <a href="#">Contenu mis à jour : Utilisation de déclencheurs</a>                                     | Amélioration et extension de la documentation relative aux déclencheurs de flux de travail.   | 11 août 2023    |
| <a href="#">Nouveau contenu : résolution des problèmes liés aux autorisations de rôle</a>            | Ajout d'informations sur la mise à jour des autorisations de rôle pour exécuter un flux de travail nécessitant un accès à Amazon CodeGuru. veuillez consulter <a href="#">Le flux de travail moderne du plan d'application Web à trois niveaux OnPullRequest échoue en raison d'une erreur d'autorisation pour Amazon CodeGuru.</a> | 11 août 2023    |
| <a href="#">Nouveau contenu : Comment corriger les messages « Le flux de travail est inactif » ?</a> | La rubrique de résolution des problèmes suivante a été ajoutée : <a href="#">Comment corriger les messages « Le flux de travail est inactif » ?</a>   | 11 août 2023    |
| <a href="#">Nouveau contenu : Ajout de l'action « Déployer vers le cluster Kubernetes »</a>          | Ajout de documentation pour l'action de déploiement vers le cluster Kubernetes. Pour plus d'informations, consultez <a href="#">Ajout de l'action « Déployer vers le cluster Kubernetes »</a> et <a href="#">Tutoriel : Déploiement d'une application sur Amazon EKS.</a>   | 27 juillet 2023 |

[Mises à jour concernant la façon dont les événements de gestion sont enregistrés pour un CodeCatalyst espace](#)

Ajout d'informations sur la manière dont les événements de gestion sont enregistrés pour des actions spécifiques dans un CodeCatalyst espace avec AWS CloudTrail. Ajout d'informations sur la façon dont tous les événements d'un espace peuvent être visualisés à l'aide de la list-event-logs commande. veuillez consulter [Surveillance sur Amazon CodeCatalyst](#).

20 juillet 2023

[Contenu mis à jour : Utilisation de déclencheurs](#)

Mise à jour de la documentation pour indiquer que les déclencheurs de pull request sont désormais pris en charge par les référentiels GitHub sources. Auparavant, les déclencheurs de pull request n'étaient pris en charge qu'avec les référentiels CodeCatalyst sources.

14 juillet 2023

[Contenu mis à jour : Référence de définition du flux de travail](#)

Correction d'une erreur de formatage dans le bloc de Compute code.

27 juin 2023

[Contenu mis à jour : Quotas pour les flux de travail dans CodeCatalyst](#)

Mise à jour de la [Quotas pour les flux de travail dans CodeCatalyst](#) rubrique indiquant la durée maximale pendant laquelle une action peut exécuter un quota.

27 juin 2023

---

|  |  |              |
|--|--|--------------|
| <a href="#">Contenu mis à jour : protection des données</a>                      | Mise à jour de la documentation pour inclure des informations supplémentaires sur la réplication des données.  | 26 juin 2023 |
| <a href="#">Nouveau contenu : Utilisation des versions d'action</a>              | Ajout d'un <a href="#">Utilisation des versions d'action</a> sujet.  | 21 juin 2023 |
| <a href="#">Contenu mis à jour : Utilisation des environnements</a>              | Clarification de <a href="#">Quelles actions soutiennent les environnements ?</a> la section.  | 14 juin 2023 |
| <a href="#">Contenu mis à jour : documentation sur les problèmes réorganisée</a> | Réorganisation de la plupart des documents relatifs aux problèmes afin de mieux les aligner sur l'ensemble de la documentation et les flux d'utilisateurs. | 31 mai 2023  |
| <a href="#">Contenu mis à jour : Issues View Switcher</a>                        | Différents flux d'utilisateurs ont été mis à jour pour les aligner sur le sélecteur de vue des problèmes mis à jour.                                       | 31 mai 2023  |
| <a href="#">Contenu mis à jour : gestion des notifications</a>                   | Mise à jour de la documentation relative aux notifications afin d'inclure des informations sur la configuration des notifications personnelles de Slack.   | 30 mai 2023  |
| <a href="#">Contenu mis à jour : gestion des notifications</a>                   | Mise à jour de la documentation relative aux notifications afin d'inclure des informations sur la configuration des notifications personnelles de Slack.   | 30 mai 2023  |

|   |   |             |
|---|---|-------------|
| <a href="#"><u>Nouveau contenu : modèle de CodeCatalyst confiance</u></a>                     | Ajout d'une nouvelle rubrique contenant des informations sur le modèle de confiance , qui permet CodeCatalyst d'assumer le rôle de service dans le connecté Compte AWS. Ajout d'une nouvelle section sur les principes de service définis pour CodeCatalyst. veuillez consulter <a href="#"><u>CodeCatalyst modèle de confiance</u></a> . | 20 mai 2023 |
| <a href="#"><u>Contenu mis à jour : Travailler avec des sources</u></a>                       | Simplification des instructions dans <a href="#"><u>Référencement de fichiers dans un référentiel source</u></a> .  | 10 mai 2023 |
| <a href="#"><u>Contenu mis à jour : Quotas pour les flux de travail dans CodeCatalyst</u></a> | Mise à jour de la <a href="#"><u>Quotas pour les flux de travail dans CodeCatalyst</u></a> rubrique relative à la longueur maximale d'un quota de valeur de variable en sortie.   | 10 mai 2023 |
| <a href="#"><u>Nouveau contenu : Travailler avec des artefacts</u></a>                        | Ajout de deux exemples : <a href="#"><u>Exemple : Référencement d'un fichier dans un seul artefact</u></a> et <a href="#"><u>Exemple : Référencer un fichier dans un artefact en présence d'un WorkflowSource</u></a> .   | 10 mai 2023 |

|  |   |               |
|--|---|---------------|
| <a href="#">Contenu mis à jour : utilisation des pull requests</a>   | Mise à jour de la documentation relative aux pull requests afin d'inclure des informations sur la configuration des préférences de courrier électronique pour les événements de pull request.   | 21 avril 2023 |
| <a href="#">Contenu mis à jour : gestion des notifications</a>       | Mise à jour de la documentation relative aux notifications afin d'inclure des informations sur la configuration des préférences d'e-mail pour les événements de pull request.   | 21 avril 2023 |
| <a href="#">Mises à jour des politiques gérées</a>                   | Ajout du <a href="#">Stratégie AWS gérée : AmazonCodeCatalyst FullAccess</a> <a href="#">Stratégie AWS gérée : AmazonCodeCatalyst ReadOnlyAccess</a> , et des politiques <a href="#">Politique gérée par AWS : AmazonCodeCatalystSupportAccess</a> gérées. veuillez consulter <a href="#">CodeCatalyst mises à jour des politiques AWS gérées</a> . | 20 avril 2023 |
| <a href="#">Nouveau contenu : Supprimer une cible de déploiement</a> | Ajout d'un <a href="#">Supprimer une cible de déploiement</a> sujet.  | 20 avril 2023 |
| <a href="#">Nouveau contenu : Types d'action</a>                     | Ajout d'un <a href="#">CodeCatalyst Actions menées par les laboratoires</a> sujet.  | 20 avril 2023 |

|  |   |               |
|--|---|---------------|
| <a href="#">Mises à jour relatives à la gestion d'un utilisateur ayant le rôle d'administrateur de l'espace dans un espace</a> | Ajout d'informations sur la suppression ou la modification du rôle d'un utilisateur ayant le rôle d'administrateur de l'espace dans un espace. veuillez consulter <a href="#">Supprimer ou modifier le rôle d'un utilisateur ayant le rôle d'administrateur de l'espace</a> . | 19 avril 2023 |
| <a href="#">Mises à jour pour l'administration des environnements de développement</a>   | Ajout d'informations sur l'administration des environnements de développement en tant qu'administrateur de l'espace. veuillez consulter <a href="#">Administration des environnements de développement pour un espace</a> .   | 19 avril 2023 |
| <a href="#">Contenu mis à jour : problèmes de recherche et de visualisation</a>  | Le sujet et les <a href="#">Recherche et affichage des problèmes</a> sous-sujets ont été réorganisés.   | 19 avril 2023 |
| <a href="#">Contenu mis à jour : Travailler avec le calcul</a>   | Ajout du support pour l'architecture Arm64 sur Amazon Linux 2.  | 19 avril 2023 |
| <a href="#">Nouveau contenu : transfert des problèmes au sein des groupes</a>  | Ajout de <a href="#">documentation pour le transfert des problèmes au sein des groupes</a> du forum et de la vue Tous les problèmes.  | 19 avril 2023 |



|  |  |               |
|--|--|---------------|
| <a href="#">Contenu mis à jour : Quotas pour les flux de travail dans CodeCatalyst</a>   | Mise à jour de la <a href="#">Quotas pour les flux de travail dans CodeCatalyst</a> rubrique avec les quotas manquants et mise à jour de la taille totale maximale du quota de variables de sortie d'une seule action à 120 Ko (au lieu de 2 Ko).  | 18 avril 2023 |
| <a href="#">Nouveau contenu : Afficher le code source d'une action</a>   | Ajout d'un <a href="#">Afficher le code source d'une action</a> sujet.   | 18 avril 2023 |
| <a href="#">Nouveau contenu : Réessayer les scénarios de test</a>  | Ajout d'un <a href="#">Réessayer les scénarios de test</a> sujet.  | 11 avril 2023 |
| <a href="#">Nouveau contenu : Arrêt d'une exécution de flux de travail</a>   | Ajout d'un <a href="#">Arrêt d'une exécution de flux de travail</a> sujet.   | 10 avril 2023 |
| <a href="#">Nouveau contenu : ajout de sections pour le balisage des ressources pour les connexions de compte entre AWS et Amazon CodeCatalyst</a> | Ajout d'informations pour le balisage des ressources de connexion aux comptes et la gestion des politiques IAM pour les ressources de connexion. Consultez <a href="#">Utilisation de balises pour contrôler l'accès aux ressources de connexion au compte</a> et <a href="#">CodeCatalyst référence aux autorisations</a> . | 6 avril 2023  |
| <a href="#">Nouveau contenu : Types d'action</a>   | Ajout d'un <a href="#">Types d'action</a> sujet.   | 6 avril 2023  |

|  |   |              |
|--|---|--------------|
| <a href="#">Contenu mis à jour :<br/>Référence d'action « Deploy<br/>AWS CloudFormation Stack »</a>        | Description de la parameter<br>-overrides propriété<br>mise à jour. Il prend désormais<br>en charge les fichiers JSON.  | 5 avril 2023 |
| <a href="#">Nouveau contenu : Création<br/>d'un projet CodeCatalyst avec<br/>un GitHub référentiel lié</a> | Ajout d'une nouvelle section<br><a href="#">Création d'un projet sur<br/>Amazon CodeCatalyst</a> intitulée<br><a href="#">Création d'un projet avec un<br/>GitHub référentiel lié</a> avec<br>des instructions pour créer<br>un projet lié à votre GitHub<br>référentiel. | 5 avril 2023 |
| <a href="#">Contenu mis à jour : utilisation<br/>des notifications</a>                                     | Mise à jour de la documenta<br>tion relative aux notifications<br>afin d'inclure des informati<br>ons sur la configuration des e-<br>mails relatifs aux événements<br>du projet.  | 31 mars 2023 |
| <a href="#">Nouveau contenu</a>  | Publication initiale du <a href="#">guide<br/>Amazon CodeCatalyst Action<br/>Development Kit</a> .  | 31 mars 2023 |
| <a href="#">Contenu mis à jour : restructu<br/>ration de la section Spaces<br/>d'Amazon CodeCatalyst</a>   | Mise à jour de la section<br>Espaces en supprimant les<br>pages de destination et en<br>consolidant les rubriques.  | 29 mars 2023 |

|  |  |              |
|--|--|--------------|
| <a href="#">Contenu mis à jour : Tutoriel : Déploiement d'une application sur Amazon ECS</a>                 | <a href="#">Étape 1 : configurer un AWS utilisateur et AWS CloudShell</a><br>Modifié pour décrire comment créer un utilisateur dans AWS IAM Identity Center au lieu de AWS Identity and Access Management. Il n'est plus recommandé de créer des utilisateurs IAM. | 23 mars 2023 |
| <a href="#">Contenu mis à jour : utilisation des rôles</a>   | Mise à jour de la documentation relative aux rôles d'administrateur de l'espace, d'administrateur de projet et de contributeur afin d'inclure les autorisations permettant de lier les problèmes aux pull requests.  | 13 mars 2023 |
| <a href="#">Contenu mis à jour : utilisation des pull requests</a>   | Mise à jour de la documentation relative aux pull requests afin d'inclure des informations sur les problèmes de liaison aux pull requests.   | 13 mars 2023 |
| <a href="#">Contenu mis à jour : résolution des problèmes</a>  | Mise à jour de la documentation relative aux problèmes afin d'inclure des informations sur le lien entre les problèmes et les pull requests.   | 13 mars 2023 |
| <a href="#">Nouveau contenu : Afficher le statut et les détails de toutes les exécutions de votre projet</a> | Ajout d'une section qui décrit la nouvelle page d'exécution du flux de travail agrégé.   | 8 mars 2023  |

|   |   |                 |
|---|---|-----------------|
| <a href="#">Nouveau contenu : Comment corriger les erreurs « La définition du flux de travail ne comporte aucune erreur » ?</a> | Ajout d'une section expliquant comment résoudre les erreurs « La définition du flux de travail comporte des erreurs ».  | 7 mars 2023     |
| <a href="#">Contenu mis à jour : Création, modification et suppression d'un flux de travail</a>                                 | Les instructions ont été mises à jour pour refléter la nouvelle interface utilisateur.  | 3 mars 2023     |
| <a href="#">Nouveau contenu : En utilisant universal-test-runner</a>  | Ajout d'un <a href="#">En utilisant universal-test-runner</a> sujet.  | 3 mars 2023     |
| <a href="#">Contenu mis à jour : Créez, testez et déployez avec des flux de travail dans CodeCatalyst</a>                       | Diverses sections ont été mises à jour pour refléter les nouveaux filtres de noms de référentiel source, de branche et de flux de travail sur la page de résumé des flux de travail.                                  | 2 mars 2023     |
| <a href="#">Nouveau contenu : Affichage de la qualité du code et de l'état du déploiement par validation CodeCatalyst</a>       | Ajout d'une section sur l'affichage de la qualité du code et de l'état du déploiement par validation.   | 27 février 2023 |
| <a href="#">Nouveau contenu : variables «BranchName" et CommitId "»</a>   | Ajout d'une nouvelle variable <code>BranchName</code> prédéfinie.   | 16 février 2023 |
| <a href="#">Contenu mis à jour : Gestion des membres de l'espace sur Amazon CodeCatalyst</a>                                    | Informations mises à jour sur la modification des rôles des membres, l'invitation de membres et la suppression de membres dans deux nouveaux tableaux en fonction du rôle attribué à l'utilisateur dans CodeCatalyst. | 15 février 2023 |

|   |   |                  |
|---|---|------------------|
| <a href="#">Contenu mis à jour : ajout d'étapes pour la gestion des PAT dans la CodeCatalyst console Amazon</a> | Ajout d'étapes pour afficher, créer et supprimer des PAT dans la console.   | 15 février 2023  |
| <a href="#">Contenu mis à jour : Utilisation des images Docker de l'environnement d'exécution</a>               | Ajout d'outils supplémentaires au tableau des versions des outils d'imagerie par défaut.  | 10 janvier 2023  |
| <a href="#">Contenu mis à jour : Travailler avec des artefacts</a>  | Correction d'un chemin d'artefact.  | 3 janvier 2023   |
| <a href="#">Contenu mis à jour : Travailler avec des sources</a>  | Correction d'un chemin source.  | 3 janvier 2023   |
| <a href="#">Contenu mis à jour : Référence GitHub d'action « Actions »</a>                                      | Correction de l'extrait de code dans la Steps section.  | 3 janvier 2023   |
| <a href="#">Contenu mis à jour : mise à jour d'une pull request</a>   | Mise à jour de la documentation pour inclure des informations sur la mise à jour des réviseurs obligatoires ou facultatifs pour une pull request. | 23 décembre 2022 |
| <a href="#">Nouveau contenu : Utilisation de la mise en cache de fichiers</a>                                   | Ajout d'une page pour la mise en cache de fichiers dans un flux de travail.   | 20 décembre 2022 |
| <a href="#">Contenu mis à jour : utilisation des pull requests</a>  | Mise à jour de la documentation relative aux pull requests afin d'inclure des informations sur les notifications.                                 | 16 décembre 2022 |
| <a href="#">Nouveau contenu : référence d'action « AWS CDK déployer »</a>                                       | Ajout d'une nouvelle <code>CdkRootPath</code> propriété.  | 16 décembre 2022 |

---

|  |  |                   |
|--|--|-------------------|
| <a href="#">Nouveau contenu : Partage du calcul entre les actions</a>          | Ajout d'un <a href="#">Partage du calcul entre les actions</a> sujet.          | 14 décembre 2022  |
| <a href="#">Contenu mis à jour : Travailler avec des artefacts</a>             | Exemples corrigés montrant comment spécifier les artefacts d'entrée.           | 13 décembre 2022  |
| <a href="#">Nouveau contenu : Référence GitHub d'action « Actions »</a>        | Ajout d'une page de référence dédiée à l'action GitHub Actions.                | 13 décembre 2022  |
| <a href="#">Contenu mis à jour : quotas pour les projets dans CodeCatalyst</a> | Mise à jour de la documentation avec un maximum de 100 projets dans un espace. | 2 décembre 2022   |
| <a href="#">Nouveau contenu</a>  | Publication initiale du guide de CodeCatalyst l'utilisateur Amazon.            | 1er décembre 2022 |

# AWS Glossaire

Pour la AWS terminologie la plus récente, consultez le [AWS glossaire](#) dans la Glossaire AWS référence.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.