



Guide de l'utilisateur

AWS Batch



AWS Batch: Guide de l'utilisateur

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

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

Table of Contents

Qu'est-ce que c'est AWS Batch ?	1
Vous en êtes un AWS Batch utilisateur pour la première fois ?	3
Services connexes	4
Accès AWS Batch	4
Composants de AWS Batch	5
Environnement de calcul	5
Files d'attente pour les jobs	6
Définitions des emplois	6
Tâches	6
Politique de planification	6
Ressources consommables	7
Environnement de service	7
Emploi dans le service	7
Con AWS Batch figuration	9
Création d'un compte IAM et d'un utilisateur administratif	9
Inscrivez-vous pour un Compte AWS	9
Création d'un utilisateur doté d'un accès administratif	10
Création de rôles IAM	12
Création d'une paire de clés	12
Création d'un VPC	14
Création d'un groupe de sécurité	15
Installez le AWS CLI	17
Tutoriels de mise en route	18
Commencer à EC2 utiliser Amazon à l'aide de l'assistant	18
Présentation	18
Prérequis	19
Étape 1 : Création d'un environnement informatique	20
Étape 2 : créer une file d'attente de tâches	20
Étape 3 : Création d'une définition de tâche	21
Étape 4 : Création d'un emploi	21
Étape 5 : Vérification et création	22
Étape 6 : Afficher le résultat de la tâche	22
Étape 7 : Nettoyez les ressources de votre didacticiel	22
Ressources supplémentaires	23

Débuter avec l'orchestration de Fargate à l'aide de l'assistant	23
Présentation de	23
Conditions préalables	24
Étape 1 : Création d'un environnement informatique	25
Étape 2 : créer une file d'attente de tâches	25
Étape 3 : Création d'une définition de tâche	26
Étape 4 : Création d'un emploi	27
Étape 5 : vérification et création	27
Étape 6 : Afficher le résultat de la tâche	27
Étape 7 : Nettoyez les ressources de votre didacticiel	28
Ressources supplémentaires	28
Commencer à utiliser AWS Batch et Fargate à l'aide du AWS CLI	29
Conditions préalables	29
Création d'un rôle d'exécution IAM	30
Création d'un environnement informatique	31
Création d'une file d'attente de tâches	32
Créer une définition de tâche	33
Soumettre et suivre une offre d'emploi	34
Afficher le résultat de la tâche	35
nettoyer des ressources ;	36
Passage en production	37
Étapes suivantes	38
Démarrer avec Amazon EKS	38
Présentation	40
Prérequis	41
Étape 1 : Créez votre cluster Amazon EKS pour AWS Batch	42
Étape 2 : Préparez votre cluster Amazon EKS pour AWS Batch	42
Étape 3 : Création d'un environnement informatique Amazon EKS	46
Étape 4 : créer une file d'attente de tâches et associer l'environnement informatique	48
Étape 5 : Création d'une définition de tâche	49
Étape 6 : Soumettre une offre d'emploi	50
Étape 7 : Afficher le résultat de la tâche	51
Étape 8 : (Facultatif) Soumettre une tâche avec des remplacements	51
Étape 9 : Nettoyez les ressources de votre didacticiel	52
Ressources supplémentaires	53
Commencer à utiliser AWS Batch les clusters privés Amazon EKS	53

Présentation	40
Prérequis	56
Étape 1 : Créez votre cluster EKS pour AWS Batch	57
Étape 2 : Préparez votre cluster EKS pour AWS Batch	59
Étape 3 : Création d'un environnement informatique Amazon EKS	62
Étape 4 : créer une file d'attente de tâches et associer l'environnement informatique	64
Étape 5 : créer un Amazon ECR avec un cache d'extraction	65
Étape 6 : enregistrer une définition de tâche	66
Étape 7 : Soumettre une tâche à exécuter	67
Étape 8 : Afficher le résultat de la tâche	67
Étape 9 : (Facultatif) Soumettre une tâche avec des remplacements	67
Étape 10 : Nettoyez les ressources de votre didacticiel	68
Ressources supplémentaires	53
Résolution des problèmes	69
Commencer à utiliser AWS Batch l' SageMaker IA	70
Présentation de	70
Conditions préalables	72
Étape 1 : Création d'un rôle d'exécution SageMaker AI	72
Étape 2 : Création de votre environnement de service	74
Étape 3 : Créez votre file SageMaker d'attente de tâches	75
Étape 4 : Création et soumission d'un poste de formation	76
Étape 5 : Surveiller l'état du travail	78
Étape 6 : Afficher le résultat de la tâche	80
Étape 7 : Nettoyez les ressources de votre didacticiel	83
Ressources supplémentaires	84
Le tableau de bord AWS Batch du widget	85
Ajouter le widget de file d'attente de tâches unique	85
Comment ajouter le widget CloudWatch Container Insights	86
Ajouter le widget Job logs	86
Environnements informatiques pour AWS Batch	88
Environnements informatiques gérés	88
Prise en compte lors de la création de tâches parallèles à nœuds multiples	91
Environnements informatiques non gérés	92
Création d'un environnement informatique	93
Tutoriel : Création d'un environnement informatique géré à l'aide des ressources Fargate	94

Tutoriel : Création d'un environnement informatique géré à l'aide EC2 des ressources Amazon	96
Tutoriel : Création d'un environnement informatique non géré à l'aide des ressources Amazon EC2	106
Tutoriel : Création d'un environnement informatique géré à l'aide des ressources Amazon EKS	107
Tutoriel : Création d'un environnement informatique non géré à l'aide des ressources Amazon EKS	113
Ressource : modèle d'environnement de calcul	114
Table de calcul du type d'instance	116
Mettre à jour un environnement informatique	119
Stratégies de mise à jour des environnements informatiques	120
Choisir la bonne stratégie de mise à jour	121
Effectuer des mises à jour de dimensionnement	123
Effectuer des mises à jour d'infrastructure	125
Effectuer des Blue/green mises à jour	130
Ressource de calcul AMIs	136
Spécification d'AMI de ressources de calcul	138
Tutoriel : Création d'une AMI de ressources de calcul	139
Utiliser une AMI de charge de travail du GPU	143
Obsolète d'Amazon Linux	149
Amazon EKS : dépréciation de l'AMI Amazon Linux 2	150
Amazon ECS : dépréciation de l'AMI Amazon Linux 2	150
Utiliser les modèles de lancement Amazon EC2	151
Modèles de lancement par défaut et de remplacement	153
Données utilisateur Amazon EC2 dans les modèles de lancement	154
Référence : exemples de modèles de lancement	155
Configuration du service de métadonnées d'instance (IMDS)	158
Scénarios de configuration	158
EC2 configurations	160
Comment migrer d'ECS AL2 vers ECS AL2 023	161
Stratégies d'allocation des types d'instances	163
Gestion de mémoire	165
Mémoire système de réserve	166
Tutoriel : Afficher la mémoire des ressources de calcul	166
Considérations relatives à la mémoire et au vCPU pour AWS Batch Amazon EKS	167

Environnements informatiques Fargate	172
Quand utiliser Fargate	173
Définitions de poste sur Fargate	173
Files d'attente pour les emplois sur Fargate	176
Environnements informatiques sur Fargate	176
Environnements informatiques Amazon EKS	177
Amazon EKS	179
AMI par défaut Amazon EKS	180
Environnements AMI mixtes	181
Versions de Kubernetes prises en charge	182
Mettre à jour la Kubernetes version de l'environnement de calcul	183
Responsabilité partagée des Kubernetes nœuds	183
Exécuter un DaemonSet sur des nœuds AWS Batch gérés	184
Personnalisez les modèles de lancement Amazon EKS	185
Comment passer d'EKS AL2 à EKS AL2023	190
Environnements de service	192
Que sont les environnements de service	192
Comment les environnements de service fonctionnent avec les autres AWS Batch	
composants	193
Bonnes pratiques pour les environnements de service	194
État et cycle de vie de l'environnement de service	195
Définitions des états de l'environnement de service	195
Création d'un environnement de service	198
Conditions préalables	198
Mettre à jour un environnement de service	200
Supprimer un environnement de service	201
Conditions préalables à la suppression	202
Files d'attente pour les jobs	205
Création d'une file d'attente de tâches	205
Création d'une file d'attente de EC2 tâches Amazon	206
Création d'une file d'attente de tâches Fargate	207
Création d'une file d'attente de tâches Amazon EKS	208
Création d'une file SageMaker d'attente de tâches	209
Modèle de file d'attente de tâches	212
Afficher une file d'attente de tâches	213
Afficher les informations relatives à la file d'attente	213

Supprimer une file d'attente de tâches	216
Politiques de planification équitables	216
Utiliser des identifiants de partage	217
Utiliser les politiques de planification	218
Utilisez la planification équitable	218
Tutoriel : Création d'une politique de planification	220
Référence : modèle de politique de planification	222
Planification axée sur les ressources	222
Créez des ressources consommables	224
Spécifier les ressources pour une tâche	224
Vérifier l'utilisation des ressources	226
Mettre à jour les quantités de ressources en cours d'utilisation	228
Trouvez des emplois nécessitant une ressource consommable	229
Supprimer une ressource consommable	230
Définitions des emplois	232
Création d'une définition de tâche à nœud unique	232
Tutoriel : Création d'une définition de tâche à nœud unique sur les ressources Amazon	
EC2	233
Création d'une définition de tâche à nœud unique sur les ressources Fargate	239
Création d'une définition de tâche à nœud unique sur les ressources Amazon EKS	246
Créez une définition de tâche à nœud unique avec plusieurs conteneurs sur les ressources	
Amazon EC2	252
Création d'une définition de tâche parallèle à nœuds multiples	259
Tutoriel : Création d'une définition de tâche parallèle à nœuds multiples sur Amazon	
Ressources EC2	259
Référence : Modèle de définition de job utilisant ContainerProperties	267
Paramètres de définition du job pour ContainerProperties	275
Créez des définitions de tâches à l'aide de EcsProperties	320
ContainerProperties par rapport aux définitions de EcsProperties poste	320
Modifications générales apportées au AWS Batch APIs	321
Définitions de tâches multi-conteneurs pour Amazon ECS	322
Définitions de tâches multi-conteneurs pour Amazon EKS	323
Référence : AWS Batch scénarios professionnels utilisant EcsProperties	324
Utiliser le pilote de journal awslogs	331
options du pilote de journalisation awslogs dans le type de données AWS Batch	
JobDefinition	332

Spécifiez une configuration de journal dans la définition de votre tâche	334
Spécifier les données sensibles	336
Utilisation de Secrets Manager	336
Utiliser le magasin de paramètres de Systems Manager	345
Authentification par registre privé pour les tâches	349
Autorisations IAM requises pour l'authentification de registre privé	350
Tutoriel : Création d'un secret pour l'authentification du registre privé	351
Volumes Amazon EFS	352
Considérations relatives aux volumes Amazon EFS	353
Utiliser les points d'accès Amazon EFS	354
Spécifiez un système de fichiers Amazon EFS dans votre définition de tâche	355
Exemples de définitions de tâches	358
Variables d'environnement	358
Substitution de paramètres	359
Tester les fonctionnalités du GPU	361
Tâche parallèle sur plusieurs nœuds	361
Tâches	363
Tutoriel : soumettre une offre d'emploi	364
Emplois dans les services	367
Charges utiles des tâches de maintenance	368
Soumettre une offre d'emploi	369
État du poste de service	371
Stratégies de nouvelle tentative d'emploi dans le service	372
Surveillez les tâches de service dans la file d'attente	375
Mettre fin à des tâches de service	377
États de l'emploi	378
Variables d'environnement de travail	381
Nouvelles tentatives de travail automatisées	383
Dépendances professionnelles	384
Expiration des délais d'exécution des tâches	385
Offres d'emploi Amazon EKS	387
Tutoriel : mapper une tâche en cours d'exécution à un pod et à un nœud	388
Tutoriel : Associer un module en cours d'exécution à sa tâche	389
Tâches parallèles sur plusieurs nœuds	390
Variables d'environnement	391
Groupes de nœuds	392

Cycle de vie du travail	393
Considérations relatives à l'environnement informatique	394
Tâches parallèles à nœuds multiples sur Amazon EKS	395
Exécution de tâches MNP	396
Création d'une définition de tâche Amazon EKS MNP	397
Soumettre une offre d'emploi Amazon EKS MNP	400
Remplacer une définition de tâche Amazon EKS MNP	400
Tâches liées à	401
Exemple de flux de travail sur une matrice	405
Utilisation d'un index de tâches matriciel	408
Exécuter des tâches GPU	414
Création d'un Kubernetes cluster basé sur un GPU sur Amazon EKS	419
Création d'une définition de tâche GPU Amazon EKS	421
Exécuter une tâche GPU dans votre cluster Amazon EKS	422
Afficher les tâches dans une file d'attente	422
Rechercher des offres d'emploi dans une file d'attente	423
Rechercher des AWS Batch offres d'emploi (AWS Console)	423
Rechercher et filtrer les AWS Batch offres d'emploi (AWS CLI)	425
Modes de mise en réseau pour les AWS Batch tâches	427
Afficher les journaux des tâches dans CloudWatch Logs	428
Consulter les informations relatives au AWS Batch poste	429
Sécurité dans AWS Batch	431
Gestion de l'identité et des accès	432
Public ciblé	432
Authentification par des identités	432
Gestion de l'accès à l'aide de politiques	434
Comment AWS Batch fonctionne avec IAM	436
Exemples de stratégies basées sur l'identité	441
AWS politiques gérées	444
Politiques, rôles et autorisations IAM	449
Structure des politiques IAM	450
Ressource : Exemples de politiques	454
Ressource : politique AWS Batch gérée	464
AWS Batch Rôle d'exécution IAM	465
Autorisations au niveau des ressources prises en charge	466
Tutoriel : Création du rôle d'exécution IAM	466

Tutoriel : Vérifiez le rôle d'exécution IAM	467
Utilisation des rôles liés à un service	468
Rôle d'instance Amazon ECS	483
Rôle de la flotte Amazon EC2 Spot	486
EventBridge Rôle IAM	489
Créer un Virtual Private Cloud	491
Création d'un VPC	491
Étapes suivantes	492
Points de terminaison d'un VPC	493
Considérations	493
Création d'un point de terminaison d'interface	494
Création d'une politique de point de terminaison	496
Validation de conformité	497
Sécurité de l'infrastructure	497
Prévention du problème de l'adjoint confus entre services	498
Exemple : rôle permettant d'accéder à un seul environnement informatique	499
Exemple : rôle pour accéder à plusieurs environnements informatiques	499
CloudTrail	500
AWS Batch informations dans CloudTrail	501
Référence : Comprendre les entrées du fichier AWS Batch journal	502
Résoudre les problèmes liés à l'IAM AWS Batch	504
Je ne suis pas autorisé à effectuer une action dans AWS Batch	504
Je ne suis pas autorisé à effectuer iam : PassRole	504
Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes AWS Batch ressources	505
AWS Step Functions	507
Tutoriel : Afficher les détails de la machine à états	507
Tutoriel : Modifier une machine à états	508
Tutoriel : Exécuter une machine à états	508
Amazon EventBridge	509
AWS Batch événements	510
Événements liés à la modification de l'état du poste	510
Événements bloqués dans la file d'offres d'emploi	512
Événements de modification de l'état des tâches de service	513
Événements bloqués dans la file d'attente des tâches de service	515
Tutoriel : Utiliser les notifications AWS utilisateur avec AWS Batch	516

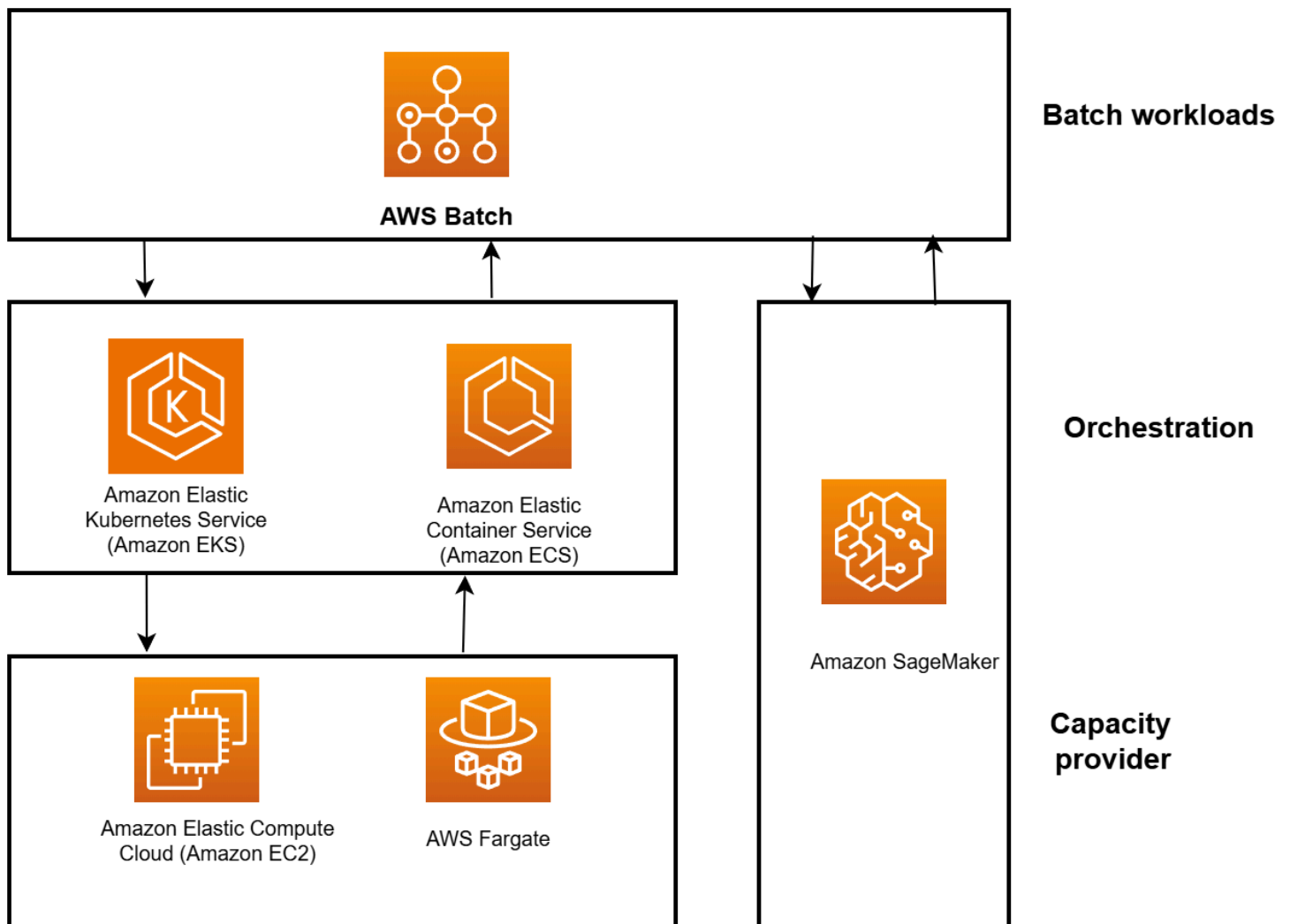
AWS Batch les emplois en tant que EventBridge cibles	517
Tutoriel : Création d'une tâche planifiée	518
Tutoriel : Création d'une règle avec un modèle d'événement	521
Tutoriel : Transformateur d'entrée Pass	524
Tutoriel : écoutez les événements liés à l' AWS Batch emploi	527
Prérequis	527
Tutoriel : Création de la fonction Lambda	528
Tutoriel : Enregistrer la règle de l'événement	528
Tutoriel : Testez votre configuration	531
Tutoriel : envoi d'alertes Amazon Simple Notification Service en cas d'échec d'une tâche	531
Prérequis	531
Tutoriel : créer une rubrique Amazon SNS et s'y abonner	532
Tutoriel : Enregistrer une règle d'événement	532
Tutoriel : Testez votre règle	534
Règle alternative : file d'attente des tâches Batch bloquée	535
Elastic Fabric Adapter	536
Moniteur AWS Batch	539
CloudWatch Journaux	540
Tutoriel : Ajouter une politique CloudWatch Logs IAM	541
Installation et configuration de l' CloudWatch agent	542
Tutoriel : Afficher CloudWatch les journaux	543
CloudWatch Informations sur les conteneurs	544
Activer Container Insights	545
Utiliser CloudWatch les journaux pour surveiller AWS Batch les tâches Amazon EKS	546
Prérequis	547
Installez le module complémentaire	547
Baliser vos ressources	549
Principes de base des étiquettes	549
Baliser vos ressources	550
Restrictions liées aux étiquettes	551
Tutoriel : Gérer les tags à l'aide de la console	552
Ajouter des balises sur une ressource individuelle lors de sa création	552
Ajouter et supprimer des balises pour une ressource individuelle	552
Gérez les balises à l'aide de la CLI ou de l'API	553
Bonnes pratiques	555
Quand utiliser AWS Batch	555

Liste de contrôle à exécuter à grande échelle	556
Optimisez les conteneurs et AMIs	557
Choisissez la bonne ressource d'environnement informatique	558
Amazon EC2 On-Demand ou Amazon EC2 Spot	559
Utilisez les meilleures pratiques d'Amazon EC2 Spot pour AWS Batch	560
Erreurs courantes et résolution des problèmes	562
Résolution des problèmes	566
AWS Batch	567
Configuration optimale du type d'instance pour recevoir des mises à jour automatiques de la famille d'instances	568
INVALIDenvironnement informatique	568
Offres d'emploi bloquées dans un RUNNABLE statut	571
Instances ponctuelles non étiquetées lors de la création	577
Les instances ponctuelles ne sont pas réduites	578
Impossible de récupérer les secrets de Secrets Manager	579
Impossible de contourner les exigences en ressources liées à la définition des tâches	580
Message d'erreur lorsque vous mettez à jour le <code>desiredvCpus</code> paramètre	581
AWS Batch sur Amazon EKS	582
INVALIDenvironnement informatique	582
AWS Batch sur Amazon EKS, le RUNNABLE statut de la tâche est bloqué	585
AWS Batch sur Amazon EKS, le STARTING statut de la tâche est bloqué	586
Vérifiez que le <code>aws-auth ConfigMap</code> est correctement configuré	587
Les autorisations ou les liaisons RBAC ne sont pas correctement configurées	588
Ressource : Quotas de service	591
Historique du document	593
.....	dci

Qu'est-ce que c'est AWS Batch ?

AWS Batch vous permet d'exécuter des charges de travail de calcul par lots sur le AWS Cloud. Le Batch computing est un moyen courant pour les développeurs, les scientifiques et les ingénieurs d'accéder à de grandes quantités de ressources informatiques. AWS Batch élimine le fardeau indifférencié lié à la configuration et à la gestion de l'infrastructure requise, comme c'est le cas avec les logiciels de calcul par lots traditionnels. Ce service peut allouer efficacement des ressources en réponse à des tâches soumises dans le but d'éliminer les contraintes de capacité, de réduire les coûts de calcul et de fournir des résultats rapidement.

En tant que service entièrement géré, il vous AWS Batch aide à exécuter des charges de travail de calcul par lots à n'importe quelle échelle. AWS Batch provisionne automatiquement les ressources informatiques et optimise la répartition de la charge de travail en fonction de la quantité et de l'échelle des charges de travail. Ainsi AWS Batch, il n'est pas nécessaire d'installer ou de gérer un logiciel de calcul par lots. Vous pouvez donc consacrer votre temps à l'analyse des résultats et à la résolution des problèmes.

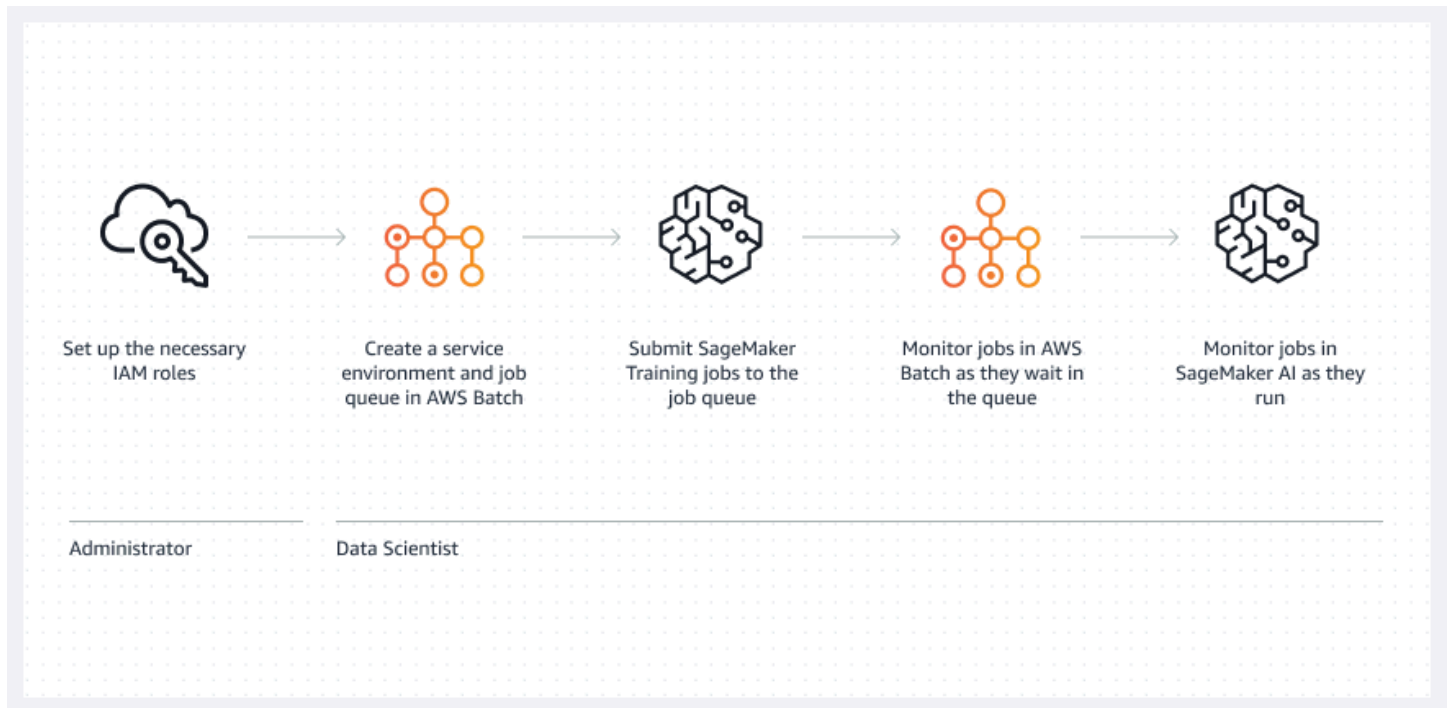


AWS Batch fournit toutes les fonctionnalités nécessaires pour exécuter des charges de travail à grande échelle et gourmandes en ressources informatiques en plus des services AWS gérés d'orchestration de conteneurs, Amazon ECS et Amazon EKS. AWS Batch est capable d'augmenter la capacité de calcul sur les EC2 instances Amazon et les ressources Fargate.

AWS Batch fournit un service entièrement géré pour les charges de travail par lots et fournit les capacités opérationnelles nécessaires pour optimiser ces types de charges de travail en termes de débit, de rapidité, d'efficacité des ressources et de coût.

AWS Batch permet également la mise en file d'attente des tâches de SageMaker formation, permettant aux scientifiques des données et aux ingénieurs du ML de soumettre des tâches de formation prioritaires à des files d'attente configurables. Cette fonctionnalité garantit que les charges de travail ML s'exécutent automatiquement dès que les ressources sont disponibles, éliminant ainsi le besoin de coordination manuelle et améliorant l'utilisation des ressources.

Pour les charges de travail d'apprentissage automatique, AWS Batch fournit des fonctionnalités de mise en file d'attente pour SageMaker les tâches de formation. Vous pouvez configurer des files d'attente avec des politiques spécifiques afin d'optimiser les coûts, les performances et l'allocation des ressources pour vos charges de travail de formation ML.



Cela fournit un modèle de responsabilité partagée dans lequel les administrateurs configurent l'infrastructure et les autorisations, tandis que les data scientists peuvent se concentrer sur la soumission et le suivi de leurs charges de travail de formation au machine learning. Les tâches sont automatiquement mises en file d'attente et exécutées en fonction des priorités configurées et de la disponibilité des ressources.

Vous en êtes un AWS Batch utilisateur pour la première fois ?

Si vous utilisez pour la première fois AWS Batch, nous vous recommandons de commencer par lire les sections suivantes :

- [Composants de AWS Batch](#)
- [Création d'un compte IAM et d'un utilisateur administratif](#)
- [Con AWS Batch figuration](#)
- [Commencer à utiliser les AWS Batch didacticiels](#)
- [Commencer à utiliser AWS Batch l' SageMaker IA](#)

Services connexes

AWS Batch est un service de calcul par lots entièrement géré qui planifie, planifie et exécute vos charges de travail conteneurisées d'apprentissage automatique par lots, de simulation et d'analyse sur l'ensemble des offres de AWS calcul, telles qu'Amazon ECS, Amazon EKS et les instances Spot ou AWS Fargate On-Demand. Pour plus d'informations sur chaque service de calcul géré, voir :

- [Guide de EC2 l'utilisateur Amazon](#)
- [Guide du développeur AWS Fargate](#)
- [Guide de l'utilisateur d'Amazon EKS](#)
- [Guide du développeur Amazon SageMaker AI](#)

Accès AWS Batch

Vous pouvez y accéder AWS Batch en utilisant les moyens suivants :

AWS Batch console

Interface Web dans laquelle vous créez et gérez des ressources.

AWS Command Line Interface

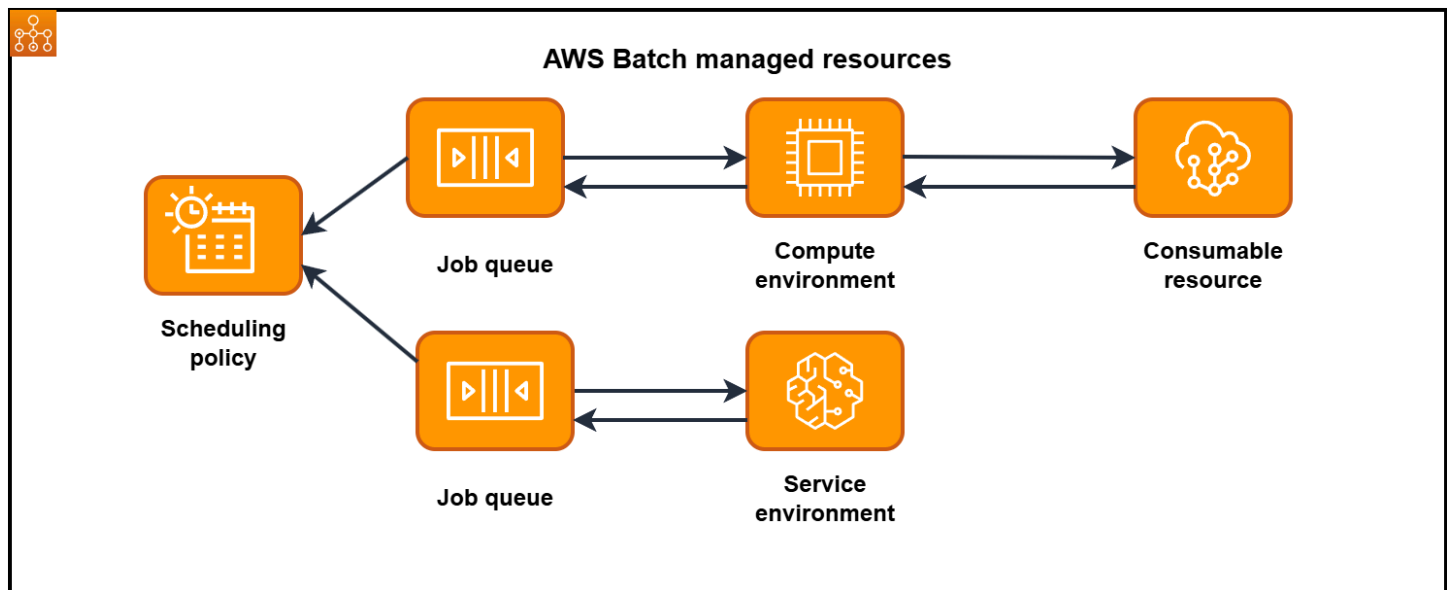
Interagissez à Services AWS l'aide de commandes dans l'interpréteur de commandes de votre ligne de commande. AWS Command Line Interface Il est pris en charge sur Windows, macOS et Linux. Pour plus d'informations à ce sujet AWS CLI, consultez le [Guide de AWS Command Line Interface l'utilisateur](#). Vous trouverez les AWS Batch commandes dans la [référence des AWS CLI commandes](#).

AWS SDKs

Si vous préférez créer des applications utilisant des langages spécifiques APIs au lieu de soumettre une demande via HTTP ou HTTPS, utilisez les bibliothèques, les exemples de code, les didacticiels et les autres ressources fournis par. AWS Ces bibliothèques fournissent des fonctions de base qui automatisent les tâches, telles que la signature cryptographique de vos demandes, les nouvelles tentatives et la gestion des réponses aux erreurs. Ces fonctions vous permettent de démarrer plus efficacement. Pour plus d'informations, consultez la section [Outils sur lesquels vous pouvez vous appuyer AWS](#).

Composants de AWS Batch

AWS Batch simplifie l'exécution de tâches par lots dans plusieurs zones de disponibilité au sein d'une même région. Vous pouvez créer des environnements de calcul AWS Batch au sein d'un nouveau VPC ou d'un VPC existant. Une fois qu'un environnement de calcul est opérationnel et associé à une file d'attente de tâches, vous pouvez définir des définitions de tâche qui spécifient les images de conteneur Docker pour exécuter vos tâches. Les images de conteneur sont stockées dans et extraites de registres de conteneur, qui peuvent exister dans votre infrastructure AWS ou en dehors de celle-ci.



Environnement de calcul

Un environnement de calcul est un ensemble de ressources informatiques gérées ou non gérées qui sont utilisées pour exécuter des tâches. Avec les environnements informatiques gérés, vous pouvez spécifier le type de calcul souhaité (Fargate EC2 ou) à plusieurs niveaux de détail. Vous pouvez configurer des environnements de calcul qui utilisent un type d'EC2 instance particulier, un modèle particulier tel que `c5.2xlarge` ou `m5.10xlarge`. Vous pouvez également choisir de spécifier uniquement que vous souhaitez utiliser les types d'instances les plus récents. Vous pouvez également spécifier le nombre minimum, souhaité et maximum de v CPUs pour l'environnement, ainsi que le montant que vous êtes prêt à payer pour une instance ponctuelle sous forme de pourcentage du prix de l'instance à la demande et un ensemble cible de sous-réseaux VPC. AWS Batch lance, gère et arrête efficacement les types de calcul selon les besoins. Vous pouvez également gérer vos propres environnements de calcul. En tant que tel, vous êtes responsable de la configuration et du dimensionnement des instances dans un cluster Amazon ECS AWS Batch créé

pour vous. Pour de plus amples informations, veuillez consulter [Environnements informatiques pour AWS Batch](#).

Files d'attente pour les jobs

Lorsque vous soumettez une AWS Batch tâche, vous la soumettez à une file d'attente de tâches particulière, où la tâche réside jusqu'à ce qu'elle soit planifiée dans un environnement informatique. Vous associez un ou plusieurs environnements informatiques à une file d'attente de tâches. Vous pouvez également attribuer des valeurs de priorité à ces environnements informatiques et même aux files d'attente de travail elles-mêmes. Par exemple, vous pouvez avoir une file d'attente de priorité élevée à laquelle vous soumettez des tâches urgentes, et une file d'attente de faible priorité pour les tâches qui peuvent être exécutées à tout moment lorsque les ressources de calcul sont moins chères. Pour de plus amples informations, veuillez consulter [Files d'attente pour les jobs](#).

Définitions des emplois

Une définition de tâche indique comment les tâches doivent être exécutées. Vous pouvez considérer une définition de poste comme un modèle pour les ressources de votre travail. Vous pouvez attribuer un rôle IAM à votre tâche afin de donner accès à d'autres AWS ressources. Vous spécifiez également les exigences en termes de mémoire et de processeur. La définition de tâche peut également contrôler des propriétés de conteneurs, des variables d'environnement et des points de montage pour un stockage permanent. De nombreuses spécifications d'une définition de tâche peuvent être remplacées en spécifiant de nouvelles valeurs lorsque vous soumettez des tâches individuelles. Pour de plus amples informations, consultez [Définitions des emplois](#).

Tâches

Une unité de travail (comme un script shell, un exécutable Linux ou une image de conteneur Docker) que vous soumettez à AWS Batch. Elle porte un nom et s'exécute en tant qu'application conteneurisée sur les EC2 ressources AWS Fargate Amazon de votre environnement informatique, en utilisant les paramètres que vous spécifiez dans une définition de tâche. Les tâches peuvent faire référence à d'autres tâches par leur nom ou leur identifiant, et peuvent dépendre de la réussite d'autres tâches ou de la disponibilité des [ressources](#) que vous spécifiez. Pour de plus amples informations, veuillez consulter [Tâches](#).

Politique de planification

Vous pouvez utiliser des politiques de planification pour configurer la manière dont les ressources de calcul d'une file d'attente de tâches sont réparties entre les utilisateurs ou les charges de travail. À

l'aide de politiques de planification de partage équitable, vous pouvez attribuer différents identifiants de partage aux charges de travail ou aux utilisateurs. Le planificateur de AWS Batch tâches utilise par défaut une stratégie « premier entré, premier sorti » (FIFO). Pour de plus amples informations, veuillez consulter [Politiques de planification équitables](#).

Ressources consommables

Une ressource consommable est une ressource nécessaire à l'exécution de vos tâches, telle qu'un jeton de licence tiers, la bande passante d'accès à la base de données, la nécessité de limiter les appels vers une API tierce, etc. Vous spécifiez les ressources consommables nécessaires à l'exécution d'une tâche, et Batch prend en compte ces dépendances de ressources lorsqu'il planifie une tâche. Vous pouvez réduire la sous-utilisation des ressources de calcul en n'allouant que les tâches disposant de toutes les ressources nécessaires. Pour de plus amples informations, veuillez consulter [Planification axée sur les ressources](#).

Environnement de service

Un environnement de service définit le mode AWS Batch d'intégration SageMaker pour l'exécution des tâches. Les environnements de service permettent AWS Batch de soumettre et de gérer des tâches SageMaker tout en fournissant les fonctionnalités de mise en file d'attente, de planification et de gestion des priorités de. AWS Batch Les environnements de service définissent les limites de capacité pour des types de services spécifiques tels que les tâches de SageMaker formation. Les limites de capacité contrôlent le maximum de ressources pouvant être utilisées par les tâches de service dans l'environnement. Pour de plus amples informations, veuillez consulter [Environnements de service pour AWS Batch](#).

Emploi dans le service

Une tâche de service est une unité de travail que vous soumettez AWS Batch pour être exécutée dans un environnement de service. Les tâches de service tirent AWS Batch parti des capacités de mise en file d'attente et de planification tout en déléguant l'exécution réelle au service externe. Par exemple, les tâches de SageMaker formation soumises en tant que tâches de service sont mises en file d'attente et hiérarchisées par AWS Batch, mais l'exécution des tâches de SageMaker formation se fait au sein SageMaker de l'infrastructure d'IA. Cette intégration permet aux scientifiques des données et aux ingénieurs du ML de bénéficier de AWS Batch la gestion automatisée de la charge de travail et de la mise en file d'attente prioritaire pour leurs charges de travail de formation à l' SageMaker IA. Les tâches de service peuvent faire référence à d'autres tâches par leur nom ou leur

identifiant et prendre en charge les dépendances entre les tâches. Pour de plus amples informations, veuillez consulter [Emplois de service à AWS Batch](#).

Con AWS Batch figuration

Si vous êtes déjà inscrit à Amazon Web Services (AWS) et que vous utilisez Amazon Elastic Compute Cloud (Amazon EC2) ou Amazon Elastic Container Service (Amazon ECS), vous pourrez bientôt utiliser AWS Batch. Le processus de configuration de ces services est similaire. Cela est dû au fait qu'il AWS Batch utilise des instances de conteneur Amazon ECS dans ses environnements informatiques. Pour utiliser le AWS CLI with AWS Batch, vous devez utiliser une version du AWS CLI compatible avec les dernières AWS Batch fonctionnalités. Si aucune AWS Batch fonctionnalité n'est prise en charge dans le AWS CLI, passez à la dernière version. Pour en savoir plus , consultez <http://aws.amazon.com/cli/>.

Note

Comme il AWS Batch utilise des composants d'Amazon EC2, vous utilisez la EC2 console Amazon pour la plupart de ces étapes.

Effectuez les tâches suivantes pour vous préparer AWS Batch.

Rubriques

- [Création d'un compte IAM et d'un utilisateur administratif](#)
- [Créez des rôles IAM pour vos environnements informatiques et vos instances de conteneur](#)
- [Créez une paire de clés pour vos instances](#)
- [Création d'un VPC](#)
- [Création d'un groupe de sécurité](#)
- [Installez le AWS CLI](#)

Création d'un compte IAM et d'un utilisateur administratif

Pour commencer, vous devez créer un AWS compte et un utilisateur unique auxquels des droits d'administration sont généralement accordés. Pour ce faire, suivez les didacticiels suivants :

Inscrivez-vous pour un Compte AWS

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

Pour vous inscrire à un Compte AWS

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

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

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

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

Création d'un utilisateur doté d'un accès administratif

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

Sécurisez votre Utilisateur racine d'un compte AWS

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

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

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

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

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

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

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

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

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

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

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

Attribution d'un accès à d'autres utilisateurs

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

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

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

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

Créez des rôles IAM pour vos environnements informatiques et vos instances de conteneur

Vos environnements AWS Batch informatiques et vos instances de conteneur ont besoin d'un compte AWS d'informations d'identification pour passer des appels à d'autres AWS APIs en votre nom. Créez un AWS Identity and Access Management rôle qui fournit ces informations d'identification à vos environnements informatiques et à vos instances de conteneur, puis associez ce rôle à vos environnements informatiques.

Note

Pour vérifier que votre compte AWS dispose des autorisations requises, consultez la section [Configuration initiale du service IAM pour votre compte](#).

L'environnement de calcul AWS Batch et les rôles d'instance de conteneur sont automatiquement créés pour vous lors de la première exécution de la console. Donc, si vous avez l'intention d'utiliser la console AWS Batch, vous pouvez passer à la section suivante. Si vous envisagez d'utiliser la CLI AWS, effectuez les procédures dans [Utilisation des rôles liés à un service pour AWS Batch](#) et [Tutoriel : Création du rôle d'exécution IAM](#) avant de créer votre premier environnement de calcul. [Rôle d'instance Amazon ECS](#)

Créez une paire de clés pour vos instances

AWS utilise le chiffrement à clé publique pour sécuriser les informations de connexion de votre instance. Une instance Linux, telle qu'une instance de conteneur d'environnement de calcul AWS Batch, n'a aucun mot de passe à utiliser pour l'accès SSH. Une paire de clés vous permet de vous connecter en toute sécurité à votre instance. Vous indiquez le nom de la paire de clés lors de la création de votre environnement de calcul, puis vous fournissez la clé privée lorsque vous vous connectez avec SSH.

Si vous n'avez pas encore créé de paire de clés, vous pouvez en créer une à l'aide de la console Amazon EC2. Notez que, si vous prévoyez de lancer plusieurs instances dans plusieurs régions AWS, créez une paire de clés dans chaque région. Pour plus d'informations sur les régions, consultez [Régions et zones de disponibilité](#) dans le guide de l'utilisateur Amazon EC2.

Création d'une paire de clés

1. Ouvrez la console Amazon EC2 à l'adresse <https://console.aws.amazon.com/ec2/>.

2. Dans la barre de navigation, sélectionnez un Région AWS pour la paire de clés. Vous pouvez sélectionner n'importe quelle région disponible, quel que soit votre lieu de résidence : toutefois, les paires de clés sont spécifiques à une région. Par exemple, si vous envisagez de lancer une instance dans la région USA Ouest (Oregon), créez une paire de clés pour l'instance dans la même région.
3. Dans le panneau de navigation, choisissez Paires de clés, puis Créer une paire de clés.
4. Dans la boîte de dialogue Créer une paire de clés, pour Nom de la paire de clés, saisissez le nom de la nouvelle paire de clés, puis choisissez Créer. Choisissez un nom dont vous vous souviendrez, tel que votre nom d'utilisateur, suivi de `-key-pair`, plus le nom de la région. Par exemple, `moi-key-pair-uswest2`.
5. Le fichier de clé privée est automatiquement téléchargé dans votre navigateur. Le nom de base du fichier est le nom que vous avez spécifié comme nom de votre paire de clés et l'extension du nom de fichier est `.pem`. Enregistrez le fichier de clé privée en lieu sûr.

 Important

C'est votre seule occasion d'enregistrer le fichier de clé privée. Vous devez fournir le nom de votre paire de clés lorsque vous lancez une instance et la clé privée correspondante chaque fois que vous vous connectez à l'instance.

6. Si vous utilisez un client SSH sur un ordinateur Mac ou Linux pour vous connecter à votre instance Linux, utilisez la commande suivante pour définir les autorisations de votre fichier de clé privée. Ainsi, tu es le seul à pouvoir le lire.

```
$ chmod 400 your_user_name-key-pair-region_name.pem
```

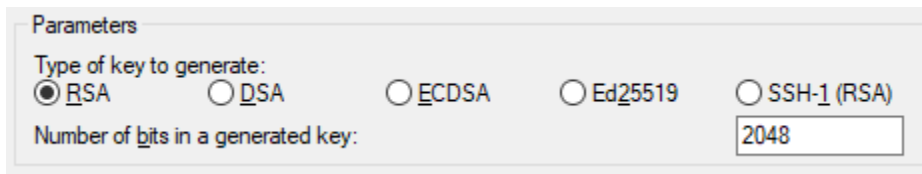
Pour plus d'informations, consultez [Amazon EC2 Key Pairs](#) dans le guide de EC2 l'utilisateur Amazon.

Pour vous connecter à votre instance à l'aide de votre paire de clés

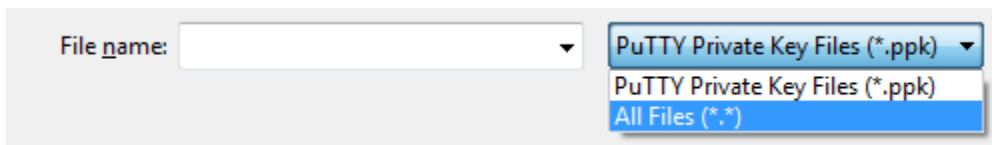
Pour vous connecter à votre instance Linux sur un ordinateur exécutant Mac ou Linux, spécifiez le fichier `.pem` dans le client SSH avec l'option `-i` et le chemin d'accès à votre clé privée. Pour vous connecter à votre instance Linux à partir d'un ordinateur exécutant Windows, utilisez l'un des deux MindTerm ou PuTTY. Si vous envisagez d'utiliser PuTTY, installez-le et suivez la procédure ci-dessous pour convertir le `.pem` fichier en fichier `.ppk`

(Facultatif) Pour préparer la connexion à une instance Linux depuis Windows à l'aide de PuTTY

1. Téléchargez PuTTY depuis <http://www.chiark.greenend.org.uk/~sgtatham/putty/>, puis installez-le. Assurez-vous d'installer toute la suite.
2. Démarrez PuTTYgen (par exemple, dans le menu Démarrer, choisissez Tous les programmes, PuTTY et PuTTYgen).
3. Sous Type of key to generate (Type de clé à générer), sélectionnez RSA. Si vous utilisez une version antérieure de PuTTYgen, choisissez SSH-2 RSA.



4. Choisissez Load (Charger). Par défaut, PuTTYgen affiche uniquement les fichiers portant l'extension .ppk. Pour trouver votre fichier .pem, choisissez l'option permettant d'afficher tous les types de fichiers.



5. Sélectionnez la clé privée créée dans la procédure précédente, puis sélectionnez Open (Ouvrir). Sélectionnez OK pour ignorer la boîte de dialogue de confirmation.
6. Choisissez Save private key (Enregistrer la clé privée). PuTTYgen affiche un avertissement concernant l'enregistrement de la clé sans phrase secrète. Choisissez Yes (Oui).
7. Spécifiez le même nom pour la clé que celui utilisé pour la paire de clés. PuTTY ajoute automatiquement l'extension de fichier .ppk.

Création d'un VPC

Avec Amazon Virtual Private Cloud (Amazon VPC), vous pouvez lancer AWS des ressources dans un réseau virtuel que vous avez défini. Nous vous recommandons vivement de lancer vos instances de conteneur dans un VPC.

Si vous avez un VPC par défaut, vous pouvez également ignorer cette section et passer à la tâche suivante. [Création d'un groupe de sécurité](#) Pour déterminer si vous disposez d'un VPC par défaut, consultez la section [Plateformes prises en charge dans la EC2 console Amazon dans](#) le guide de l'utilisateur Amazon EC2

Pour plus d'informations sur la création d'un Amazon VPC, consultez la section [Créer un VPC uniquement dans le guide de l'utilisateur Amazon VPC](#). Reportez-vous au tableau suivant pour déterminer les options à sélectionner.

Option	Valeur	
Ressources à créer	VPC uniquement	
Nom	Vous pouvez, si vous le souhaitez, nommer votre VPC.	
IPv4 Bloc d'adresse CIDR	IPv4 Entrée manuelle CIDR	
	La taille du bloc d'adresse CIDR doit être comprise entre /16 et /28.	
IPv6 Bloc d'adresse CIDR	Aucun IPv6 bloc CIDR	
Location	Par défaut	

Pour plus d'informations sur Amazon VPC, consultez [Qu'est-ce qu'Amazon VPC ?](#) dans le Guide de l'utilisateur Amazon VPC.

Création d'un groupe de sécurité

Les groupes de sécurité font office de pare-feu pour les instances de conteneurs d'environnement de calcul associées, en contrôlant le trafic entrant et le trafic sortant au niveau de l'instance de conteneur. Un groupe de sécurité ne peut être utilisé que dans le VPC dans lequel il est créé.

Vous pouvez ajouter des règles à un groupe de sécurité qui vous permettent de vous connecter à votre instance de conteneur à partir de votre adresse IP avec SSH. Vous pouvez aussi ajouter des règles qui permettent les accès HTTP et HTTPS entrants et sortants depuis n'importe quel emplacement. Ajoutez des règles pour ouvrir les ports qui sont requis par vos tâches.

Notez que si vous prévoyez de lancer des instances de conteneur dans plusieurs régions, vous devez créer un groupe de sécurité dans chaque région. Pour plus d'informations, consultez la section [Régions et zones de disponibilité](#) dans le guide de EC2 l'utilisateur Amazon.

 Note

Vous avez besoin de l'adresse IP publique de votre ordinateur local, que vous pouvez obtenir à l'aide d'un service. Par exemple, nous fournissons le service suivant : <http://checkip.amazonaws.com/> ou <https://checkip.amazonaws.com/>. Pour trouver un autre service qui fournit votre adresse IP, utilisez l'expression de recherche « what is my IP address » (quelle est mon adresse IP). Si vous vous connectez via un fournisseur de services Internet (ISP) ou derrière un pare-feu sans adresse IP statique, découvrez la plage d'adresses IP utilisées par les ordinateurs clients.

Pour créer un groupe de sécurité à l'aide de la console

1. Ouvrez la console Amazon VPC à l'adresse <https://console.aws.amazon.com/vpc/>.
2. Dans le panneau de navigation, choisissez Groupes de sécurité.
3. Sélectionnez Create security group (Créer un groupe de sécurité).
4. Saisissez un nom et une description pour le groupe de sécurité. Vous ne pouvez pas modifier le nom et la description d'un groupe de sécurité créé.
5. Dans VPC, choisissez le VPC.
6. (Facultatif) Par défaut, les nouveaux groupes de sécurité commencent uniquement par une règle sortante qui autorise tout le trafic à quitter la ressource. Vous devez ajouter des règles pour activer un trafic entrant ou limiter le trafic sortant.

AWS Batch les instances de conteneur ne nécessitent aucun port entrant pour être ouvert. Cependant, vous souhaitez peut-être ajouter une règle SSH. Ainsi, vous pouvez vous connecter à l'instance de conteneur et examiner les conteneurs dans les tâches à l'aide des commandes Docker. Si vous souhaitez que votre instance de conteneur héberge une tâche qui exécute un serveur Web, vous pouvez également ajouter des règles pour le protocole HTTP. Appliquez la procédure suivante pour ajouter ces règles de groupe de sécurité facultatives.

Dans l'onglet Entrant, créez les règles suivantes et choisissez Créer :

- Choisissez Add Rule (Ajouter une règle). Pour Type, choisissez HTTP. Pour Source, choisissez N'importe où (0.0.0.0/0).
- Choisissez Add Rule (Ajouter une règle). Pour Type, choisissez SSH. Pour Source, choisissez IP personnalisée et spécifiez l'adresse IP publique de votre ordinateur ou de votre réseau en notation CIDR (Classless Inter-Domain Routing). Si votre entreprise alloue des adresses à

partir d'une plage, spécifiez la plage complète, telle que `203.0.113.0/24`. Pour spécifier une adresse IP individuelle en notation CIDR, choisissez `My IP`. Cela ajoute le préfixe de routage `/32` à l'adresse IP publique.

 Note

Pour des raisons de sécurité, nous vous déconseillons d'autoriser l'accès SSH à votre instance à partir de toutes les adresses IP (`0.0.0.0/0`), mais uniquement à des fins de test et pour une courte période.

7. Vous pouvez ajouter des étiquettes maintenant ou ultérieurement. Pour ajouter une étiquette, choisissez `Ajouter une nouvelle étiquette`, puis entrez la clé et la valeur de l'étiquette.
8. Sélectionnez `Créer un groupe de sécurité`.

Pour créer un groupe de sécurité à l'aide de la ligne de commande, voir [> create-security-group](#) (AWS CLI)

Pour plus d'informations sur les groupes de sécurité, consultez la [section Utilisation des groupes de sécurité](#).

Installez le AWS CLI

Pour utiliser le AWS CLI with AWS Batch, installez la dernière AWS CLI version. Pour plus d'informations sur l'installation AWS CLI ou la mise à niveau vers la dernière version, consultez la section [Installation de l'interface de ligne de AWS commande](#) dans le guide de AWS Command Line Interface l'utilisateur.

Commencer à utiliser les AWS Batch didacticiels

Vous pouvez utiliser l'assistant de AWS Batch première exécution pour démarrer rapidement. AWS Batch Après avoir rempli les conditions préalables, vous pouvez utiliser l'assistant de première exécution pour créer un environnement informatique, une définition de tâche et une file d'attente de tâches.

Vous pouvez également soumettre un exemple de tâche « Hello World » à l'aide de l'assistant de AWS Batch première exécution pour tester votre configuration. Si vous avez déjà une image Docker dans laquelle vous souhaitez vous lancer AWS Batch, vous pouvez utiliser cette image pour créer une définition de tâche.

Ensuite, vous pouvez utiliser l'assistant de AWS Batch première exécution pour créer un environnement informatique, une file d'attente de tâches et soumettre un exemple de tâche Hello World.

Commencer à utiliser l' EC2 orchestration d'Amazon à l'aide de l'assistant

Amazon Elastic Compute Cloud (Amazon EC2) fournit une capacité de calcul évolutive dans le AWS Cloud. Avec Amazon, vous EC2 n'avez plus besoin d'investir dans du matériel dès le départ, ce qui vous permet de développer et de déployer des applications plus rapidement.

Vous pouvez utiliser Amazon EC2 pour lancer autant ou aussi peu de serveurs virtuels que nécessaire, configurer la sécurité et le réseau, et gérer le stockage. Amazon vous EC2 permet de passer à la hausse ou à la baisse pour faire face à l'évolution des exigences ou aux pics de popularité, réduisant ainsi le besoin de prévoir le trafic.

Présentation

Ce didacticiel explique comment configurer Amazon EC2 et l'exécuter à l' AWS Batch aide de l'assistantHello World.

Public visé

Ce didacticiel est destiné aux administrateurs système et aux développeurs chargés de la configuration, des tests et du déploiement AWS Batch.

Fonctionnalités utilisées

Ce didacticiel explique comment utiliser l'assistant de AWS Batch console pour :

- Création et configuration d'un environnement EC2 informatique Amazon
- Créez une file d'attente de tâches.
- Créer une définition de tâche
- Création et soumission d'une tâche à exécuter
- Afficher le résultat de la tâche dans CloudWatch

Temps requis

La réalisation de ce didacticiel devrait prendre environ 10 à 15 minutes.

Restrictions régionales

Aucune restriction nationale ou régionale n'est associée à l'utilisation de cette solution.

Coûts d'utilisation des ressources

La création d'un AWS compte est gratuite. Cependant, en mettant en œuvre cette solution, vous pouvez être redevable de tout ou partie des frais répertoriés dans le tableau suivant.

Description	Coût (dollar américain)
EC2 Instance Amazon	Vous payez pour chaque EC2 instance Amazon créée. Pour plus d'informations sur les tarifs, consultez Amazon EC2 Pricing .

Prérequis

Avant de commencer :

- Créez-en un Compte AWS si vous n'en avez pas.
- Créez le [rôle d'ecsInstanceRoleinstance](#).

Étape 1 : Création d'un environnement informatique

Important

Pour démarrer le plus simplement et le plus rapidement possible, ce didacticiel inclut des étapes avec les paramètres par défaut. Avant de créer pour une utilisation en production, nous vous recommandons de vous familiariser avec tous les paramètres et de déployer avec ceux qui répondent à vos besoins.

Pour créer un environnement de calcul pour une EC2 orchestration Amazon, procédez comme suit :

1. Ouvrez l'[assistant de première exécution de la AWS Batch console](#).
2. Pour Configurer le type de tâche et d'orchestration, choisissez Amazon Elastic Compute Cloud (Amazon EC2).
3. Choisissez Suivant.
4. Dans la section Configuration de l'environnement de calcul pour Nom, spécifiez un nom unique pour votre environnement de calcul. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
5. Pour Rôle d'instance, choisissez un rôle d'instance existant auquel sont attachées les autorisations IAM requises. Ce rôle d'instance permet aux instances de conteneur Amazon ECS de votre environnement informatique d'appeler les opérations d' AWS API requises. Pour de plus amples informations, veuillez consulter [Rôle d'instance Amazon ECS](#).

Le nom par défaut du rôle d'instance est `estecsInstanceRole`.

6. Pour la configuration de l'instance, vous pouvez conserver les paramètres par défaut.
7. Pour la configuration réseau, utilisez votre VPC par défaut pour la Région AWS.
8. Choisissez Suivant.

Étape 2 : créer une file d'attente de tâches

Une file d'attente de tâches stocke les tâches que vous avez soumises jusqu'à ce que le AWS Batch planificateur exécute la tâche sur une ressource de votre environnement informatique. Pour de plus amples informations, consultez [Files d'attente pour les jobs](#).

Pour créer une file d'attente de tâches pour une EC2 orchestration Amazon, procédez comme suit :

1. Pour la configuration de la file d'attente de tâches pour le nom, spécifiez un nom unique pour votre file d'attente de tâches. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
2. Pour toutes les autres options de configuration, vous pouvez conserver la valeur par défaut.
3. Choisissez Suivant.

Étape 3 : Création d'une définition de tâche

AWS Batch les définitions de tâches spécifient la manière dont les tâches doivent être exécutées. Même si chaque tâche doit faire référence à une définition de tâche, de nombreux paramètres spécifiés dans la définition de tâche peuvent être remplacés lors de l'exécution.

Pour créer la définition de tâche, procédez comme suit :

1. Pour créer une définition de tâche
 - a. pour Nom, spécifiez un nom unique pour votre file d'attente de tâches. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
 - b. Pour Commande (facultatif), vous pouvez passer `hello world` à un message personnalisé ou le laisser tel quel.
2. Pour toutes les autres options de configuration, vous pouvez conserver la valeur par défaut.
3. Choisissez Suivant.

Étape 4 : Création d'un emploi

Pour créer une tâche, procédez comme suit :

1. Dans la section Configuration de la tâche pour Nom, spécifiez un nom unique pour la tâche. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
2. Pour toutes les autres options de configuration, vous pouvez conserver la valeur par défaut.
3. Choisissez Suivant.

Étape 5 : Vérification et création

Sur la page Réviser et créer, passez en revue les étapes de configuration. Si vous devez apporter des modifications, choisissez Modifier. Lorsque vous avez terminé, choisissez Créer des ressources.

1. Pour Réviser et créer, choisissez Créer des ressources.
2. Une fenêtre s'ouvre dès le AWS Batch départ pour allouer vos ressources. Une fois terminé, choisissez Accéder au tableau de bord. Sur le tableau de bord, vous devriez voir toutes les ressources qui vous sont allouées et indiquer que le poste est en `Runnable` bon état. Votre tâche est planifiée pour être exécutée et devrait être terminée dans 2 à 3 minutes.

Étape 6 : Afficher le résultat de la tâche

Pour afficher le résultat du Job, procédez comme suit :

1. Dans le volet de navigation, sélectionnez Jobs.
2. Dans le menu déroulant Job Queue, sélectionnez la file d'attente des Job que vous avez créée pour le didacticiel.
3. Le tableau des tâches répertorie toutes vos tâches et leur statut actuel. Une fois que le statut du Job est atteint, choisissez le nom du Job pour afficher les détails du Job.
4. Dans le volet Détails, choisissez le nom du flux du journal. La CloudWatch console du Job s'ouvre et il devrait y avoir un événement avec le message de `hello world` ou votre message personnalisé.

Étape 7 : Nettoyez les ressources de votre didacticiel

L'EC2 instance Amazon vous est facturée lorsqu'elle est activée. Vous pouvez supprimer l'instance pour ne plus encourir de frais.

Pour supprimer les ressources que vous avez créées, procédez comme suit :

1. Dans le volet de navigation, choisissez Job queue.
2. Dans le tableau des files d'attente de tâches, choisissez la file d'attente de tâches que vous avez créée pour le didacticiel.
3. Choisissez Désactiver. Une fois que l'état de la file d'attente des Job est désactivé, vous pouvez choisir Delete.

4. Une fois la file d'attente de Job supprimée, dans le volet de navigation, sélectionnez Compute environments.
5. Choisissez l'environnement informatique que vous avez créé pour ce didacticiel, puis sélectionnez Désactiver. La désactivation complète de l'environnement informatique peut prendre 1 à 2 minutes.
6. Une fois que l'état de l'environnement informatique est désactivé, choisissez Supprimer. La suppression de l'environnement informatique peut prendre 1 à 2 minutes.

Ressources supplémentaires

Une fois le didacticiel terminé, vous souhaitez peut-être explorer les sujets suivants :

- Explorez les AWS Batch principaux composants. Pour de plus amples informations, veuillez consulter [Composants de AWS Batch](#).
- En savoir plus sur les différents [environnements informatiques](#) disponibles dans AWS Batch.
- En savoir plus sur les [files d'attente de Job](#) et leurs différentes options de planification.
- En savoir plus sur [les définitions de Job](#) et les différentes options de configuration.
- Apprenez-en davantage sur les différents types d'[emplois](#).

Démarrage AWS Batch et orchestration de Fargate à l'aide de l'assistant

AWS Fargate lance et adapte le calcul pour qu'il corresponde étroitement aux besoins en ressources que vous spécifiez pour le conteneur. Avec Fargate, vous n'avez pas besoin de surapprovisionner ou de payer pour des serveurs supplémentaires. Pour plus d'informations, consultez [Fargate](#).

Présentation de

Ce didacticiel explique comment configurer AWS Fargate et l'exécuter à l' AWS Batch aide de l'assistant. Hello World

Public visé

Ce didacticiel est destiné aux administrateurs système et aux développeurs chargés de la configuration, des tests et du déploiement AWS Batch.

Fonctionnalités utilisées

Ce didacticiel explique comment utiliser l'assistant de AWS Batch console pour :

- Création et configuration d'un environnement AWS informatique Fargate
- Créez une file d'attente de tâches.
- Créer une définition de tâche
- Création et soumission d'une tâche à exécuter
- Afficher le résultat de la tâche dans CloudWatch

Temps requis

La réalisation de ce didacticiel devrait prendre environ 10 à 15 minutes.

Restrictions régionales

Aucune restriction nationale ou régionale n'est associée à l'utilisation de cette solution.

Coûts d'utilisation des ressources

La création d'un AWS compte est gratuite. Cependant, en mettant en œuvre cette solution, vous pouvez être redevable de tout ou partie des frais répertoriés dans le tableau suivant.

Description	Coût (dollar américain)
La tarification est basée sur le vCPU, la mémoire, les systèmes d'exploitation, l'architecture du processeur et les ressources de stockage demandés pour la tâche ou le pod.	Pour plus d'informations sur la tarification, consultez la section Tarification de Fargate .

Conditions préalables

Avant de commencer :

- Créez-en un Compte AWS si vous n'en avez pas.
- Créez le rôle d'exécution de la tâche. Si vous n'avez pas encore créé le [rôle d'exécution des tâches](#), vous pouvez le créer dans le cadre de ce didacticiel.

Étape 1 : Création d'un environnement informatique

Important

Pour démarrer le plus simplement et le plus rapidement possible, ce didacticiel inclut des étapes avec les paramètres par défaut. Avant de créer pour une utilisation en production, nous vous recommandons de vous familiariser avec tous les paramètres et de déployer avec ceux qui répondent à vos besoins.

Pour créer un environnement de calcul pour une orchestration Fargate, procédez comme suit :

1. Ouvrez l'[assistant de première exécution de la AWS Batch console](#).
2. Pour Configurer le type de tâche et d'orchestration, choisissez Fargate.
3. Choisissez Suivant.
4. Dans la section Configuration de l'environnement de calcul pour Nom, spécifiez un nom unique pour votre environnement de calcul. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
5. Pour toutes les autres options de configuration, vous pouvez conserver la valeur par défaut.
6. Choisissez Suivant.

Étape 2 : créer une file d'attente de tâches

Une file d'attente de tâches stocke les tâches que vous avez soumises jusqu'à ce que le AWS Batch planificateur exécute la tâche sur une ressource de votre environnement informatique. Pour créer une file d'attente de tâches :

Pour créer une file d'attente de tâches pour une orchestration Fargate, procédez comme suit :

1. Dans la section Configuration de la file d'attente des tâches pour Nom, spécifiez un nom unique pour votre environnement informatique. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
2. Dans le champ Priorité, entrez 900 pour la file d'attente des tâches.
3. Pour toutes les autres options de configuration, vous pouvez conserver la valeur par défaut.

4. Choisissez Suivant.

Étape 3 : Création d'une définition de tâche

Pour créer la définition de tâche, procédez comme suit :

1. Dans la section Configuration générale :

- Dans la section Configuration générale pour Nom, spécifiez un nom unique pour votre environnement informatique. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).

2. Dans la section de configuration de la plateforme Fargate :

- a. Activez l'option Attribuer une adresse IP publique pour attribuer une adresse IP publique. Vous avez besoin d'une adresse IP publique pour télécharger l'image du conteneur, sauf si vous avez configuré un référentiel d'images privé.
- b. Pour le rôle d'exécution, choisissez un rôle d'exécution de tâche qui permet aux agents Amazon Elastic Container Service (Amazon ECS) de passer des AWS appels en votre nom. Choisissez `ecsTaskExecutionRôle` ou `BatchEcsTaskExecutionRole`.

Pour créer le rôle d'exécution, choisissez Créer un rôle d'exécution. Dans le mode Créer un rôle IAM, choisissez Créer un rôle IAM.

- i. Le paramètre d'autorisation de la console IAM est déjà configuré pour créer le rôle d'exécution.
- ii. Pour le type d'entité fiable, vérifiez que AWS le service est sélectionné.
- iii. Pour le service ou le cas utilisateur, vérifiez qu'Elastic Container Service est sélectionné.
- iv. Choisissez Suivant.
- v. Pour les politiques d'autorisations, vérifiez qu'Amazon ECSTask ExecutionRolePolicy est sélectionné.
- vi. Choisissez Suivant.
- vii. Pour Nom, vérifiez et créez, vérifiez que le nom du rôle est bien `BatchEcsTaskExecutionRole`.
- viii. Choisissez Créer un rôle.

- ix. Dans la AWS Batch console, cliquez sur le bouton d'actualisation situé à côté du rôle d'exécution. Choisissez le rôle BatchEcsTaskExecutionRole d'exécution.
3. Dans la section Configuration du conteneur :
 - Pour Command, vous pouvez passer `hello world` à un message personnalisé ou le laisser tel quel.
4. Pour toutes les autres options de configuration, vous pouvez conserver la valeur par défaut.
5. Choisissez Suivant.

Étape 4 : Création d'un emploi

Pour créer une tâche Fargate, procédez comme suit :

1. Dans la section Configuration de la tâche pour Nom, spécifiez un nom unique pour la tâche. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
2. Pour toutes les autres options de configuration, vous pouvez conserver la valeur par défaut.
3. Choisissez Suivant.

Étape 5 : vérification et création

Sur la page Réviser et créer, passez en revue les étapes de configuration. Si vous devez apporter des modifications, choisissez Modifier. Lorsque vous avez terminé, choisissez Créer des ressources.

Étape 6 : Afficher le résultat de la tâche

Pour afficher le résultat du Job, procédez comme suit :

1. Dans le volet de navigation, sélectionnez Jobs.
2. Dans le menu déroulant Job Queue, sélectionnez la file d'attente des Job que vous avez créée pour le didacticiel.
3. Le tableau des tâches répertorie toutes vos tâches et leur statut actuel. Une fois que le statut du Job est atteint, choisissez le nom du Job pour afficher les détails du Job.
4. Dans le volet Détails, choisissez le nom du flux du journal. La CloudWatch console du Job s'ouvre et il devrait y avoir un événement avec le message de `hello world` ou votre message personnalisé.

Étape 7 : Nettoyez les ressources de votre didacticiel

L'EC2 instance Amazon vous est facturée lorsqu'elle est activée. Vous pouvez supprimer l'instance pour ne plus encourir de frais.

Pour supprimer les ressources que vous avez créées, procédez comme suit :

1. Dans le volet de navigation, choisissez Job queue.
2. Dans le tableau des files d'attente de tâches, choisissez la file d'attente de tâches que vous avez créée pour le didacticiel.
3. Choisissez Désactiver. Une fois que l'état de la file d'attente des Job est désactivé, vous pouvez choisir Delete.
4. Une fois la file d'attente de Job supprimée, dans le volet de navigation, sélectionnez Compute environments.
5. Choisissez l'environnement informatique que vous avez créé pour ce didacticiel, puis sélectionnez Désactiver. La désactivation complète de l'environnement informatique peut prendre 1 à 2 minutes.
6. Une fois que l'état de l'environnement informatique est désactivé, choisissez Supprimer. La suppression de l'environnement informatique peut prendre 1 à 2 minutes.

Ressources supplémentaires

Une fois le didacticiel terminé, vous souhaitez peut-être explorer les sujets suivants :

- En savoir plus sur les [meilleures pratiques](#).
- Explorez les AWS Batch principaux composants. Pour de plus amples informations, veuillez consulter [Composants de AWS Batch](#).
- En savoir plus sur les différents [environnements informatiques](#) disponibles dans AWS Batch.
- En savoir plus sur les [files d'attente de Job](#) et leurs différentes options de planification.
- En savoir plus sur [les définitions de Job](#) et les différentes options de configuration.
- En savoir plus sur les différents types d'[emplois](#).

Commencer à utiliser AWS Batch et Fargate à l'aide du AWS CLI

Ce didacticiel explique comment configurer AWS Batch l' AWS Fargate orchestration et exécuter une simple tâche « Hello World » à l'aide du AWS Command Line Interface (AWS CLI). Vous apprendrez à créer des environnements informatiques, des files d'attente de tâches, des définitions de tâches et à soumettre des tâches à AWS Batch.

Rubriques

- [Conditions préalables](#)
- [Création d'un rôle d'exécution IAM](#)
- [Création d'un environnement informatique](#)
- [Création d'une file d'attente de tâches](#)
- [Créer une définition de tâche](#)
- [Soumettre et suivre une offre d'emploi](#)
- [Afficher le résultat de la tâche](#)
- [nettoyer des ressources ;](#)
- [Passage en production](#)
- [Étapes suivantes](#)

Conditions préalables

Avant de commencer ce didacticiel, assurez-vous de disposer des éléments suivants.

1. Le AWS CLI. Si vous devez l'installer, suivez les instructions du [Guide d'installation de l'AWS CLI](#). Vous pouvez également [utiliser AWS CloudShell](#), ce qui inclut le AWS CLI.
2. Vous avez configuré AWS CLI avec les informations d'identification appropriées. Si vous n'avez pas encore configuré vos informations d'identification, exécutez la commande `aws configure`.
3. Connaissance de base des interfaces de ligne de commande et des concepts de conteneurisation.
4. [Comment AWS Batch fonctionne avec IAM](#) pour créer et gérer AWS Batch des ressources, des rôles IAM et des ressources VPC dans votre. Compte AWS
5. Un ID de sous-réseau et un ID de groupe de sécurité provenant d'un VPC de votre. Compte AWS Si vous n'avez pas de VPC, vous pouvez en [créer un](#). Pour plus d'informations sur l'utilisation de AWS CLI pour récupérer ces ressources IDs, consultez [describe-subnets](#) et [describe-security-groups](#) dans le AWS CLI manuel de référence des commandes.

Temps requis : environ 15 à 20 minutes pour terminer ce didacticiel.

Coût : ce didacticiel utilise les ressources de calcul de Fargate. Le coût estimé pour terminer ce didacticiel est inférieur à 0,01 USD, en supposant que vous suiviez les instructions de nettoyage pour supprimer les ressources immédiatement après l'avoir terminé. La tarification Fargate est basée sur les ressources vCPU et mémoire consommées, facturées à la seconde avec un minimum d'une minute. Pour obtenir des informations sur les prix actuels, consultez la section [AWS Fargate sur les prix](#).

Création d'un rôle d'exécution IAM

AWS Batch nécessite un rôle d'exécution qui permet aux agents Amazon Elastic Container Service (Amazon ECS) de passer des AWS appels d'API en votre nom. Ce rôle est nécessaire pour que les tâches Fargate puissent extraire des images de conteneurs et écrire des journaux sur Amazon CloudWatch.

Création d'un document de politique de confiance

Tout d'abord, créez une politique de confiance qui permet au service des tâches Amazon ECS d'assumer ce rôle.

```
cat > batch-execution-role-trust-policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

Création du rôle d'exécution

La commande suivante crée un rôle IAM nommé `BatchEcsTaskExecutionRoleTutorial` à l'aide de la politique de confiance que vous venez de créer.

```
aws iam create-role \
  --role-name BatchEcsTaskExecutionRoleTutorial \
  --assume-role-policy-document file://batch-execution-role-trust-policy.json
```

Joindre la politique requise

Joignez la politique AWS gérée qui fournit les autorisations nécessaires à l'exécution des tâches Amazon ECS.

```
aws iam attach-role-policy \
  --role-name BatchEcsTaskExecutionRoleTutorial \
  --policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
```

Le rôle est maintenant prêt à être utilisé AWS Batch pour l'exécution de tâches Fargate.

Création d'un environnement informatique

Un environnement de calcul définit les ressources de calcul sur lesquelles vos tâches par lots seront exécutées. Dans le cadre de ce didacticiel, vous allez créer un environnement informatique Fargate géré qui provisionne et adapte automatiquement les ressources en fonction des exigences du poste.

Création de l'environnement informatique

La commande suivante crée un environnement de calcul Fargate. Remplacez l'exemple de sous-réseau et de groupe de sécurité par IDs les vôtres, conformément au [Conditions préalables](#).

```
aws batch create-compute-environment \
  --compute-environment-name my-fargate-compute-env \
  --type MANAGED \
  --state ENABLED \
  --compute-resources type=FARGATE,maxvCpus=128,subnets=subnet-  
a123456b,securityGroupIds=sg-a12b3456
```

Ce qui suit montre à quoi ressemble le résultat lorsque la commande est exécutée avec succès.

```
{
  "computeEnvironmentName": "my-fargate-compute-env",
  "computeEnvironmentArn": "arn:aws:batch:us-west-2:123456789012:compute-environment/  
my-fargate-compute-env"
}
```

Attendez que l'environnement informatique soit prêt

Vérifiez l'état de votre environnement informatique pour vous assurer qu'il est prêt avant de continuer.

```
aws batch describe-compute-environments \
  --compute-environments my-fargate-compute-env \
  --query 'computeEnvironments[0].status'
```

```
"VALID"
```

Lorsque le statut s'affiche `VALID`, votre environnement informatique est prêt à accepter des tâches.

Création d'une file d'attente de tâches

Une file d'attente de tâches stocke les tâches soumises jusqu'à ce que le AWS Batch planificateur les exécute sur les ressources de votre environnement informatique. Les tâches sont traitées par ordre de priorité dans la file d'attente.

Création de la file d'attente des tâches

La commande suivante crée une file d'attente de tâches de priorité 900 qui utilise votre environnement informatique Fargate.

```
aws batch create-job-queue \
  --job-queue-name my-fargate-job-queue \
  --state ENABLED \
  --priority 900 \
  --compute-environment-order order=1,computeEnvironment=my-fargate-compute-env
```

Ce qui suit montre à quoi ressemble le résultat lorsque la commande est exécutée avec succès.

```
{
  "jobQueueName": "my-fargate-job-queue",
  "jobQueueArn": "arn:aws:batch:us-west-2:123456789012:job-queue/my-fargate-job-queue"
}
```

Vérifiez que la file d'attente des tâches est prête

Vérifiez que votre file d'attente est en bon ENABLED état et prête à accepter des offres d'emploi.

```
aws batch describe-job-queues \  
  --job-queues my-fargate-job-queue \  
  --query 'jobQueues[0].state' "ENABLED"
```

Créer une définition de tâche

Une définition de tâche indique la manière dont les tâches doivent être exécutées, y compris l'image Docker à utiliser, les besoins en ressources et d'autres paramètres. Pour Fargate, vous utiliserez les besoins en ressources plutôt que les paramètres traditionnels de vCPU et de mémoire.

Création de la définition de tâche

La commande suivante crée une définition de tâche qui exécute une simple commande « hello world » à l'aide de l'image du conteneur busybox. 123456789012 Remplacez-le par votre Compte AWS identifiant actuel et remplacez l'exemple Région AWS par le vôtre.

```
aws batch register-job-definition \  
  --job-definition-name my-fargate-job-def \  
  --type container \  
  --platform-capabilities FARGATE \  
  --container-properties '{  
    "image": "busybox",  
    "resourceRequirements": [  
      {"type": "VCPU", "value": "0.25"},  
      {"type": "MEMORY", "value": "512"}  
    ],  
    "command": ["echo", "hello world"],  
    "networkConfiguration": {  
      "assignPublicIp": "ENABLED"  
    },  
    "executionRoleArn": "arn:aws:iam::123456789012:role/  
BatchEcsTaskExecutionRoleTutorial"  
  },  
{  
  "jobDefinitionName": "my-fargate-job-def",  
  "jobDefinitionArn": "arn:aws:batch:us-west-2:123456789012:job-definition/my-  
fargate-job-def:1",  
  "revision": 1  
}'
```

La définition de la tâche spécifie 0,25 vCPU et 512 Mo de mémoire, qui sont les ressources minimales pour une tâche Fargate. Le `assignPublicIp` paramètre est activé afin que le conteneur puisse extraire l'image Busybox depuis Docker Hub.

Soumettre et suivre une offre d'emploi

Maintenant que vous disposez de tous les composants nécessaires, vous pouvez ajouter une tâche à votre file d'attente et suivre sa progression.

Soumettre une offre d'emploi

La commande suivante soumet une tâche à votre file d'attente à l'aide de la définition de tâche que vous avez créée.

```
aws batch submit-job \
  --job-name my-hello-world-job \
  --job-queue my-fargate-job-queue \
  --job-definition my-fargate-job-def
```

Ce qui suit montre à quoi ressemble le résultat lorsque la commande est exécutée avec succès.

```
{
  "jobArn": "arn:aws:batch:us-west-2:123456789012:job/my-hello-world-job",
  "jobName": "my-hello-world-job",
  "jobId": "1509xmpl-4224-4da6-9ba9-1d1acc96431a"
}
```

Notez les informations `jobId` renvoyées dans la réponse, car vous les utiliserez pour suivre la progression de la tâche.

Surveiller le statut du travail

Utilisez l'identifiant de la tâche pour vérifier le statut de votre tâche. La tâche progressera dans plusieurs états : `SUBMITTED`, `PENDING`, `RUNNABLE`, `STARTING`, `RUNNING`, et enfin `SUCCEEDED` ou `FAILED`.

```
aws batch describe-jobs --jobs 1509xmpl-4224-4da6-9ba9-1d1acc96431a
```

Ce qui suit montre à quoi ressemble le résultat lorsque la commande est exécutée avec succès.

```
{
  "jobs": [
    {
      "jobArn": "arn:aws:batch:us-west-2:123456789012:job/my-hello-world-job",
      "jobName": "my-hello-world-job",
      "jobId": "1509xmpl-4224-4da6-9ba9-1d1acc96431a",
      "jobQueue": "arn:aws:batch:us-west-2:123456789012:job-queue/my-fargate-job-queue",
      "status": "SUCCEEDED",
      "createdAt": 1705161908000,
      "jobDefinition": "arn:aws:batch:us-west-2:123456789012:job-definition/my-fargate-job-def:1"
    }
  ]
}
```

Lorsque le statut s'affiche **SUCCEEDED**, votre travail est terminé avec succès.

Afficher le résultat de la tâche

Une fois votre tâche terminée, vous pouvez consulter ses résultats dans Amazon CloudWatch Logs.

Obtenez le nom du flux de log

Tout d'abord, récupérez le nom du flux de journal à partir des détails de la tâche. Remplacez l'exemple d'identifiant de tâche par le vôtre.

```
aws batch describe-jobs --jobs 1509xmpl-4224-4da6-9ba9-1d1acc96431a \
  --query 'jobs[0].attempts[0].containers[0].logStreamName' \
  --output text
```

```
my-fargate-job-def/default/1509xmpl-4224-4da6-9ba9-1d1acc96431a
```

Afficher les journaux des tâches

Utilisez le nom du flux de journal pour récupérer le résultat de la tâche à partir de CloudWatch Logs.

```
aws logs get-log-events \
  --log-group-name /aws/batch/job \
  --log-stream-name my-fargate-job-def/default/1509xmpl-4224-4da6-9ba9-1d1acc96431a \
  --query 'events[*].message' \
```



```
--output text
```

Le résultat indique « Bonjour tout le monde », confirmant que votre tâche a été exécutée correctement.

nettoyer des ressources ;

Pour éviter les frais récurrents, nettoyez les ressources que vous avez créées dans ce didacticiel. Vous devez supprimer les ressources dans le bon ordre en raison des dépendances.

Désactiver et supprimer la file d'attente des tâches

Tout d'abord, désactivez la file d'attente des tâches, puis supprimez-la.

```
aws batch update-job-queue \  
  --job-queue my-fargate-job-queue \  
  --state DISABLED
```

```
aws batch delete-job-queue \  
  --job-queue my-fargate-job-queue
```

Désactiver et supprimer l'environnement informatique

Une fois la file d'attente des tâches supprimée, désactivez et supprimez l'environnement informatique.

```
aws batch update-compute-environment \  
  --compute-environment my-fargate-compute-env \  
  --state DISABLED
```

```
aws batch delete-compute-environment \  
  --compute-environment my-fargate-compute-env
```

Nettoyez le rôle IAM

Supprimez la pièce jointe à la politique et supprimez le rôle IAM.

```
aws iam detach-role-policy \  
  --role-name BatchEcsTaskExecutionRoleTutorial \  
  --policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
```

```
aws iam delete-role \  
  --role-name BatchEcsTaskExecutionRoleTutorial
```

Supprimer les fichiers temporaires

Supprimez le fichier de politique de confiance que vous avez créé.

```
rm batch-execution-role-trust-policy.json
```

Toutes les ressources ont été nettoyées avec succès.

Passage en production

Ce didacticiel est conçu pour vous aider à comprendre comment AWS Batch fonctionne Fargate. Pour les déploiements en production, tenez compte des exigences supplémentaires suivantes :

Considérations relatives à la sécurité :

- Créez des groupes de sécurité dédiés avec un accès minimal requis au lieu d'utiliser des groupes de sécurité par défaut
- Utiliser des sous-réseaux privés avec NAT Gateway au lieu de l'attribution d'une adresse IP publique pour les conteneurs
- Stockez les images de conteneurs dans Amazon ECR au lieu d'utiliser des référentiels publics
- Implémenter des points de terminaison VPC pour les communications de AWS service afin d'éviter le trafic Internet

Considérations relatives à l'architecture :

- Déployez sur plusieurs zones de disponibilité pour une haute disponibilité
- Mettre en œuvre des stratégies de relance des tâches et des files d'attente lettre morte pour la gestion des erreurs
- Utilisez plusieurs files d'attente de tâches avec des priorités différentes pour la gestion de la charge de travail
- Configurer des politiques de dimensionnement automatique en fonction de la profondeur de la file d'attente et de l'utilisation des ressources

- Mettre en œuvre la surveillance et les alertes en cas d'échec des tâches et d'utilisation des ressources

Considérations opérationnelles :

- Configuration de CloudWatch tableaux de bord et d'alarmes pour la surveillance
- Mettre en œuvre une journalisation et des pistes d'audit appropriées
- Utiliser CloudFormation ou l'infrastructure AWS CDK for sous forme de code
- Mettre en place des procédures de sauvegarde et de reprise après sinistre

[Pour obtenir des conseils complets sur les architectures prêtes pour la production, consultez le AWS Well-Architected Framework et les meilleures pratiques en matière de sécurité.AWS](#)

Étapes suivantes

Maintenant que vous avez terminé ce didacticiel, vous pouvez découvrir des AWS Batch fonctionnalités plus avancées :

- [Files d'attente pour les jobs](#)— En savoir plus sur la planification des files d'attente et la gestion des priorités
- [Définitions des emplois](#)— Explorez les configurations avancées de définition des tâches, notamment les variables d'environnement, les volumes et les stratégies de nouvelle tentative
- [Environnements informatiques pour AWS Batch](#)— Comprendre les différents types d'environnements informatiques et les options de mise à l'échelle
- [Tâches parallèles sur plusieurs nœuds](#)— Exécute des tâches qui s'étendent sur plusieurs nœuds de calcul
- [Tâches liées à](#)— Soumettez efficacement un grand nombre de tâches similaires
- [Les meilleures pratiques pour AWS Batch](#)— Apprenez les techniques d'optimisation des charges de travail de production

Commencer à utiliser AWS Batch sur Amazon EKS

AWS Batch sur Amazon EKS est un service géré permettant de planifier et de dimensionner les charges de travail par lots dans les clusters Amazon EKS existants. AWS Batch ne crée, n'administre ni n'effectue d'opérations de cycle de vie de vos clusters Amazon EKS en votre nom. AWS Batch

l'orchestration fait évoluer vers le haut et vers le bas les nœuds gérés par ces nœuds AWS Batch et exécute des pods sur ces nœuds.

AWS Batch ne touche pas les nœuds, les groupes de nœuds à dimensionnement automatique ou les cycles de vie des pods qui ne sont pas associés aux environnements AWS Batch informatiques de votre cluster Amazon EKS. AWS Batch Pour fonctionner efficacement, son [rôle lié à un service](#) nécessite des autorisations de contrôle d'accès Kubernetes basé sur les rôles (RBAC) dans votre cluster Amazon EKS existant. Pour plus d'informations, consultez la section [Utilisation de l'autorisation RBAC](#) dans la Kubernetesdocumentation.

AWS Batch nécessite un espace de Kubernetes noms dans lequel il peut étendre les pods en tant que AWS Batch tâches. Nous recommandons un espace de noms dédié pour isoler les AWS Batch pods de vos autres charges de travail de cluster.

Une fois AWS Batch que vous avez obtenu l'accès RBAC et qu'un espace de noms a été établi, vous pouvez associer ce cluster Amazon EKS à un environnement AWS Batch informatique à l'aide de l'opération [CreateComputeEnvironment](#)API. Une file d'attente de tâches peut être associée à ce nouvel environnement informatique Amazon EKS. AWS Batch les tâches sont soumises à la file d'attente des tâches sur la base d'une définition de tâche Amazon EKS à l'aide de l'opération [SubmitJob](#)API. AWS Batch lance ensuite les nœuds AWS Batch gérés et place les tâches de la file d'attente sous forme de Kubernetes pods dans le cluster EKS associé à un environnement AWS Batch informatique.

Les sections suivantes expliquent comment procéder à la configuration AWS Batch sur Amazon EKS.

Table des matières

- [Présentation](#)
- [Prérequis](#)
- [Étape 1 : Créez votre cluster Amazon EKS pour AWS Batch](#)
- [Étape 2 : Préparez votre cluster Amazon EKS pour AWS Batch](#)
- [Étape 3 : Création d'un environnement informatique Amazon EKS](#)
- [Étape 4 : créer une file d'attente de tâches et associer l'environnement informatique](#)
- [Étape 5 : Création d'une définition de tâche](#)
- [Étape 6 : Soumettre une offre d'emploi](#)
- [Étape 7 : Afficher le résultat de la tâche](#)
- [Étape 8 : \(Facultatif\) Soumettre une tâche avec des remplacements](#)

- [Étape 9 : Nettoyez les ressources de votre didacticiel](#)
- [Ressources supplémentaires](#)

Présentation

Ce didacticiel explique comment effectuer AWS Batch une configuration avec Amazon EKS à AWS CLI l'aide `kubectl eksctl` des

Public visé

Ce didacticiel est destiné aux administrateurs système et aux développeurs chargés de la configuration, des tests et du déploiement AWS Batch.

Fonctionnalités utilisées

Ce didacticiel vous montre comment utiliser le AWS CLI, pour :

- Création et configuration d'un environnement informatique Amazon EKS
- Créez une file d'attente de tâches.
- Créer une définition de tâche
- Création et soumission d'une tâche à exécuter
- Soumettre une offre d'emploi avec des dérogations

Temps requis

La réalisation de ce didacticiel devrait prendre environ 30 à 40 minutes.

Restrictions régionales

Aucune restriction nationale ou régionale n'est associée à l'utilisation de cette solution.

Coûts d'utilisation des ressources

La création d'un AWS compte est gratuite. Cependant, en mettant en œuvre cette solution, vous pouvez être redevable de tout ou partie des frais répertoriés dans le tableau suivant.

Description	Coût (dollar américain)
Vous êtes facturé en fonction de l'heure du cluster	Varie en fonction de l'instance, voir la tarification d'Amazon EKS

Prérequis

Avant de commencer ce didacticiel, vous devez installer et configurer les outils et ressources suivants dont vous avez besoin pour créer et gérer à la fois les ressources Amazon EKS AWS Batch et les ressources Amazon EKS.

- **AWS CLI** : outil de ligne de commande pour travailler avec AWS , y compris Amazon EKS. Ce guide nécessite que vous utilisiez la version 2.8.6 ou ultérieure ou la version 1.26.0 ou ultérieure. Pour plus d'informations, consultez la section [Installation, mise à jour et désinstallation du AWS CLI dans le](#) guide de l'AWS Command Line Interface utilisateur. Après l'avoir installé AWS CLI, nous vous recommandons de le configurer également. Pour plus d'informations, voir [Configuration rapide avec aws configure](#) dans le Guide de AWS Command Line Interface l'utilisateur.
- **kubect1** : outil de ligne de commande pour travailler avec des clusters Kubernetes. Ce guide est compatible avec la version 1.23 ou une version ultérieure. Pour plus d'informations, veuillez consulter [Installation ou mise à jour de kubect1](#) dans le Guide de l'utilisateur Amazon EKS.
- **eksctl**— Un outil de ligne de commande pour travailler avec des clusters Amazon EKS qui automatise de nombreuses tâches individuelles. Ce guide est compatible avec la version 0.115.0 ou une version ultérieure. Pour plus d'informations, veuillez consulter [Installation ou mise à jour de eksctl](#) dans le Guide de l'utilisateur Amazon EKS.
- **Autorisations IAM requises** : le principal de sécurité IAM que vous utilisez doit être autorisé à utiliser les rôles IAM Amazon EKS et les rôles liés aux services CloudFormation, ainsi qu'un VPC et les ressources associées. Pour plus d'informations, consultez [les sections Actions, ressources et clés de condition pour Amazon Elastic Kubernetes Service et Utilisation de rôles liés à un service dans le guide de l'utilisateur IAM](#). Vous devez effectuer toutes les étapes de ce guide avec le même utilisateur.
- **Autorisations** : les utilisateurs qui appellent l'opération d'[CreateComputeEnvironment](#) API pour créer un environnement informatique utilisant les ressources Amazon EKS ont besoin d'autorisations pour l'opération `eks:DescribeCluster` d'API.
- **Compte AWS numéro** — Vous devez connaître votre Compte AWS identifiant. Suivez les instructions de la [section Afficher votre Compte AWS identifiant](#).
- **(Facultatif) CloudWatch** — Pour examiner les détails de [\(Facultatif\) Soumettre une tâche avec des remplacements](#), la journalisation doit être configurée. Pour de plus amples informations, veuillez consulter [Utiliser CloudWatch les journaux pour surveiller AWS Batch les tâches Amazon EKS](#).

Étape 1 : Créez votre cluster Amazon EKS pour AWS Batch

Important

Pour démarrer le plus simplement et le plus rapidement possible, ce didacticiel inclut des étapes avec les paramètres par défaut. Avant de créer pour une utilisation en production, nous vous recommandons de vous familiariser avec tous les paramètres et de déployer avec ceux qui répondent à vos besoins.

Une fois que vous avez installé les prérequis, vous devez créer votre cluster à l'aide `eksctl` de. La création du cluster peut prendre entre 10 et 15 minutes.

```
$ eksctl create cluster --name my-cluster-name --region region-code
```

Dans la commande précédente, remplacez :

- *my-cluster-name* Remplacez-le par le nom que vous souhaitez utiliser pour votre cluster.
- *region-code* Remplacez-le par le Région AWS pour créer le cluster dans, par exemple `us-west-2`.

Le nom et la région du cluster sont nécessaires pour la suite de ce didacticiel.

Étape 2 : Préparez votre cluster Amazon EKS pour AWS Batch

Toutes les étapes sont obligatoires.

1. Créez un espace de noms dédié aux tâches AWS Batch

`kubectl` À utiliser pour créer un nouvel espace de noms.

```
$ namespace=my-aws-batch-namespace
```

```
$ cat - <<EOF | kubectl create -f -
{
  "apiVersion": "v1",
  "kind": "Namespace",
  "metadata": {
```

```

    "name": "${namespace}",
    "labels": {
      "name": "${namespace}"
    }
  }
}
EOF

```

Sortie :

```
namespace/my-aws-batch-namespace created
```

2. Permettre l'accès via le contrôle d'accès basé sur les rôles (RBAC)

kubectl À utiliser pour créer un Kubernetes rôle pour le cluster afin de AWS Batch permettre de surveiller les nœuds et les pods, et de lier le rôle. Vous devez effectuer cette opération une fois pour chaque cluster EKS.

```

$ cat - <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: aws-batch-cluster-role
rules:
  - apiGroups: [""]
    resources: ["namespaces"]
    verbs: ["get"]
  - apiGroups: [""]
    resources: ["nodes"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["events"]
    verbs: ["list"]
  - apiGroups: [""]
    resources: ["configmaps"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["apps"]
    resources: ["daemonsets", "deployments", "statefulsets", "replicasets"]
    verbs: ["get", "list", "watch"]

```



```

- apiGroups: ["rbac.authorization.k8s.io"]
  resources: ["clusterroles", "clusterrolebindings"]
  verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: aws-batch-cluster-role-binding
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: aws-batch-cluster-role
  apiGroup: rbac.authorization.k8s.io
EOF

```

Sortie :

```

clusterrole.rbac.authorization.k8s.io/aws-batch-cluster-role created
clusterrolebinding.rbac.authorization.k8s.io/aws-batch-cluster-role-binding created

```

3. Créez un Kubernetes rôle délimité par un espace de noms pour gérer et prolonger le cycle AWS Batch de vie des pods et les lier. Vous devez effectuer cette opération une fois pour chaque espace de noms unique.

```
$ namespace=my-aws-batch-namespace
```

```

$ cat - <<EOF | kubectl apply -f - --namespace "${namespace}"
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: aws-batch-compute-environment-role
  namespace: ${namespace}
rules:
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["create", "get", "list", "watch", "delete", "patch"]
- apiGroups: [""]
  resources: ["serviceaccounts"]
  verbs: ["get", "list"]

```

```

- apiGroups: ["rbac.authorization.k8s.io"]
  resources: ["roles", "rolebindings"]
  verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: aws-batch-compute-environment-role-binding
  namespace: ${namespace}
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: aws-batch-compute-environment-role
  apiGroup: rbac.authorization.k8s.io
EOF

```

Sortie :

```

role.rbac.authorization.k8s.io/aws-batch-compute-environment-role created
rolebinding.rbac.authorization.k8s.io/aws-batch-compute-environment-role-binding
created

```

4. Mettez à Kubernetes `aws-auth` jour la carte de configuration pour associer les autorisations RBAC précédentes au rôle lié au service. AWS Batch

Dans la commande suivante, remplacez :

- Remplacez *<your-account-number>* par votre Compte AWS numéro.

```

$ eksctl create iamidentitymapping \
  --cluster my-cluster-name \
  --arn "arn:aws:iam::<your-account-number>:role/AWSServiceRoleForBatch" \
  --username aws-batch

```

Sortie :

```
2022-10-25 20:19:57 [#] adding identity "arn:aws:iam::<your-account-number>:role/AWSServiceRoleForBatch" to auth ConfigMap
```

Note

Le chemin `aws-service-role/batch.amazonaws.com/` a été supprimé de l'ARN du rôle lié au service. Cela est dû à un problème avec la carte `aws-auth` de configuration. Pour plus d'informations, consultez la section [Les rôles dotés de chemins ne fonctionnent pas lorsque le chemin est inclus dans leur ARN dans le aws-authconfigmap](#).

Étape 3 : Création d'un environnement informatique Amazon EKS

AWS Batch les environnements informatiques définissent les paramètres des ressources de calcul pour répondre à vos besoins en matière de charge de travail par lots. Dans un environnement informatique géré, vous AWS Batch aide à gérer la capacité et les types d'instances des ressources de calcul (Kubernetesnœuds) au sein de votre cluster Amazon EKS. Ceci est basé sur la spécification des ressources de calcul que vous définissez lors de la création de l'environnement de calcul. Vous pouvez utiliser des instances EC2 à la demande ou des instances EC2 ponctuelles.

Maintenant que le rôle `AWSServiceRoleForBatch` lié au service a accès à votre cluster Amazon EKS, vous pouvez créer des AWS Batch ressources. Créez d'abord un environnement informatique qui pointe vers votre cluster Amazon EKS.

- Pour subnets exécuter `eksctl get cluster my-cluster-name` pour obtenir les sous-réseaux utilisés par le cluster.
- Pour les `securityGroupIds` paramètres, vous pouvez utiliser le même groupe de sécurité que le cluster Amazon EKS. Cette commande récupère l'ID du groupe de sécurité pour le cluster.

```
$ aws eks describe-cluster \
  --name my-cluster-name \
  --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

- Le `instanceRole` est créé lorsque vous créez le cluster. Pour trouver la `instanceRole` liste de toutes les entités qui utilisent la `AmazonEKSWorkerNodePolicy` politique :

```
$ aws iam list-entities-for-policy --policy-arn arn:aws:iam::aws:policy/
AmazonEKSWorkerNodePolicy
```

Le nom du rôle de stratégie contient le nom du cluster que vous avez créé `eksctl-my-cluster-name-nodegroup-example`.

Pour trouver l'instanceRoleARN, exécutez la commande suivante :

```
$ aws iam list-instance-profiles-for-role --role-name eksctl-my-cluster-name-
nodegroup-example
```

Sortie :

```
INSTANCEPROFILES      arn:aws:iam::<your-account-number>:instance-profile/
eks-04cb2200-94b9-c297-8dbe-87f12example
```

Pour plus d'informations, consultez les sections [Création du rôle IAM du nœud Amazon EKS](#) et [Activation de l'accès principal IAM à votre cluster](#) dans le guide de l'utilisateur Amazon EKS. Si vous utilisez un réseau de modules, consultez [la section Configuration du plug-in Amazon VPC CNI pour utiliser les rôles IAM Kubernetes pour les comptes de service dans le guide de l'utilisateur Amazon EKS](#).

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:region-code:your-account-number:cluster/my-cluster-
name",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 128,
    "instanceTypes": [
```

```
    "m5"
  ],
  "subnets": [
    "<eks-cluster-subnets-with-access-to-internet-for-image-pull>"
  ],
  "securityGroupIds": [
    "<eks-cluster-sg>"
  ],
  "instanceRole": "<eks-instance-profile>"
}
}
EOF
```

```
$ aws batch create-compute-environment --cli-input-json file:///./batch-eks-compute-environment.json
```

Remarques

- La maintenance d'un environnement informatique Amazon EKS est une responsabilité partagée. Pour de plus amples informations, veuillez consulter [Responsabilité partagée des Kubernetes nœuds](#).

Étape 4 : créer une file d'attente de tâches et associer l'environnement informatique

Important

Il est important de vérifier que l'environnement informatique est sain avant de continuer. L'opération [DescribeComputeEnvironments](#) API peut être utilisée à cette fin.

```
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1
```

Vérifiez que le status paramètre ne l'est pas INVALID. Si c'est le cas, examinez le statusReason paramètre correspondant à la cause. Pour de plus amples informations, veuillez consulter [Résolution des problèmes AWS Batch](#).

Les tâches soumises à cette nouvelle file d'attente sont exécutées en tant que modules sur des nœuds AWS Batch gérés qui ont rejoint le cluster Amazon EKS associé à votre environnement informatique.

```
$ cat <<EOF > ./batch-eks-job-queue.json
{
  "jobQueueName": "My-Eks-JQ1",
  "priority": 10,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "My-Eks-CE1"
    }
  ]
}
EOF
```

```
$ aws batch create-job-queue --cli-input-json file:///./batch-eks-job-queue.json
```

Étape 5 : Création d'une définition de tâche

La définition de Job suivante indique au module de rester en veille pendant 60 secondes.

```
$ cat <<EOF > ./batch-eks-job-definition.json
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "sleep",
            "60"
          ],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi"
            }
          }
        }
      ]
    }
  }
}
```

```

    }
  }
},
],
"metadata": {
  "labels": {
    "environment": "test"
  }
}
}
}
}
EOF

```

```
$ aws batch register-job-definition --cli-input-json file:///./batch-eks-job-definition.json
```

Remarques

- Il existe des considérations relatives aux memory paramètres cpu et. Pour de plus amples informations, veuillez consulter [Considérations relatives à la mémoire et au vCPU pour AWS Batch Amazon EKS](#).

Étape 6 : Soumettre une offre d'emploi

Exécutez la AWS CLI commande suivante pour soumettre un nouveau Job.

```
$ aws batch submit-job --job-queue My-Eks-JQ1 \
  --job-definition MyJob0nEks_Sleep --job-name My-Eks-Job1
```

Pour vérifier le statut d'un Job :

```
$ aws batch describe-jobs --job <jobId-from-submit-response>
```

Remarques

- Pour plus d'informations sur l'exécution de tâches sur les ressources Amazon EKS, consultez [Offres d'emploi Amazon EKS](#).

Étape 7 : Afficher le résultat de la tâche

Pour afficher le résultat du Job, procédez comme suit :

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans le volet de navigation, sélectionnez Jobs.
3. Dans le menu déroulant Job queue, sélectionnez la file d'attente des Job que vous avez créée pour le didacticiel.
4. Le tableau des tâches répertorie toutes vos tâches et leur statut actuel. Une fois que le statut du Job est atteint, choisissez le nom du Job *My-Eks-JQ1*, pour afficher les détails du Job.
5. Dans le volet Détails, les moments démarrés et arrêtés doivent être espacés d'une minute.

Étape 8 : (Facultatif) Soumettre une tâche avec des remplacements

Cette tâche remplace la commande transmise au conteneur. AWS Batch nettoie agressivement les capsules une fois les travaux terminés afin de réduire la charge à. Kubernetes Pour examiner les détails d'une tâche, la journalisation doit être configurée. Pour de plus amples informations, veuillez consulter [Utiliser CloudWatch les journaux pour surveiller AWS Batch les tâches Amazon EKS](#).

```
$ cat <<EOF > ./submit-job-override.json
{
  "jobName": "EksWithOverrides",
  "jobQueue": "My-Eks-JQ1",
  "jobDefinition": "MyJobOnEks_Sleep",
  "eksPropertiesOverride": {
    "podProperties": {
      "containers": [
        {
          "command": [
            "/bin/sh"
          ],
          "args": [
            "-c",
            "echo hello world"
          ]
        }
      ]
    }
  }
}
```



```
}  
EOF
```

```
$ aws batch submit-job --cli-input-json file:///./submit-job-override.json
```

Remarques

- Pour une meilleure visibilité des détails des opérations, activez la journalisation du plan de contrôle Amazon EKS. Pour plus d'informations, consultez la section [Connexion au plan de contrôle Amazon EKS](#) dans le guide de l'utilisateur Amazon EKS.
- Daemonsetset la kubelets surcharge affecte les ressources de vCPU et de mémoire disponibles, en particulier le dimensionnement et le placement des tâches. Pour de plus amples informations, veuillez consulter [Considérations relatives à la mémoire et au vCPU pour AWS Batch Amazon EKS](#).

Pour afficher le résultat du Job, procédez comme suit :

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans le volet de navigation, sélectionnez Jobs.
3. Dans le menu déroulant Job queue, sélectionnez la file d'attente des Job que vous avez créée pour le didacticiel.
4. Le tableau des tâches répertorie toutes vos tâches et leur statut actuel. Une fois que le statut du Job est atteint, choisissez le nom du Job pour afficher les détails du Job.
5. Dans le volet Détails, choisissez le nom du flux du journal. La CloudWatch console du Job s'ouvre et il devrait y avoir un événement avec le message de hello world ou votre message personnalisé.

Étape 9 : Nettoyez les ressources de votre didacticiel

L'EC2 instance Amazon vous est facturée lorsqu'elle est activée. Vous pouvez supprimer l'instance pour ne plus encourir de frais.

Pour supprimer les ressources que vous avez créées, procédez comme suit :

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans le volet de navigation, choisissez Job queue.

3. Dans le tableau des files d'attente de tâches, choisissez la file d'attente de tâches que vous avez créée pour le didacticiel.
4. Choisissez Désactiver. Une fois que l'état de la file d'attente des Job est désactivé, vous pouvez choisir Delete.
5. Une fois la file d'attente de Job supprimée, dans le volet de navigation, sélectionnez Compute environments.
6. Choisissez l'environnement informatique que vous avez créé pour ce didacticiel, puis sélectionnez Désactiver. La désactivation complète de l'environnement informatique peut prendre 1 à 2 minutes.
7. Une fois que l'état de l'environnement informatique est désactivé, choisissez Supprimer. La suppression de l'environnement informatique peut prendre 1 à 2 minutes.

Ressources supplémentaires

Une fois le didacticiel terminé, vous souhaitez peut-être explorer les sujets suivants :

- En savoir plus sur les [meilleures pratiques](#).
- Explorez les AWS Batch principaux composants. Pour de plus amples informations, veuillez consulter [Composants de AWS Batch](#).
- En savoir plus sur les différents [environnements informatiques](#) disponibles dans AWS Batch.
- En savoir plus sur les [files d'attente de Job](#) et leurs différentes options de planification.
- En savoir plus sur [les définitions de Job](#) et les différentes options de configuration.
- En savoir plus sur les différents types d'[emplois](#).

Commencer à utiliser AWS Batch les clusters privés Amazon EKS

AWS Batch est un service géré qui orchestre les charges de travail par lots dans vos clusters Amazon Elastic Kubernetes Service (Amazon EKS). Cela inclut la mise en file d'attente, le suivi des dépendances, la gestion des nouvelles tentatives et des priorités des tâches, la gestion des pods et le dimensionnement des nœuds. Cette fonctionnalité connecte votre cluster privé Amazon EKS existant à votre cluster Amazon EKS AWS Batch pour exécuter vos tâches à grande échelle. Vous pouvez utiliser [eksctl](#) (une interface de ligne de commande pour Amazon EKS), la AWS console ou le [AWS Command Line Interface](#) pour créer un cluster Amazon EKS privé avec toutes les autres ressources nécessaires.

Par défaut, les [clusters privés Amazon EKS](#) n'ont pas accès à inbound/outbound Internet et vous ne pouvez accéder au serveur d'API que depuis votre VPC ou un réseau connecté. Les points de terminaison Amazon VPC sont utilisés pour permettre un accès privé à d'autres services. AWS `eksctl` prend en charge la création de clusters entièrement privés à l'aide d'un Amazon VPC et de sous-réseaux préexistants. `eksctl` crée également des points de terminaison Amazon VPC dans le VPC Amazon fourni et modifie les tables de routage pour les sous-réseaux fournis.

Chaque sous-réseau doit être associé à une table de routage explicite, car cela `eksctl` ne modifie pas la table de routage principale. Votre [cluster](#) doit extraire des images d'un registre de conteneurs qui se trouve dans votre Amazon VPC. Vous pouvez également créer un Amazon Elastic Container Registry dans votre Amazon VPC et y copier des images de conteneurs pour que vos nœuds puissent les extraire. Pour plus d'informations, voir [Copier une image de conteneur d'un référentiel vers un autre référentiel](#). Pour commencer à utiliser les référentiels privés Amazon ECR, consultez les référentiels privés [Amazon ECR](#).

Vous pouvez éventuellement créer une [règle de cache](#) d'extraction avec Amazon ECR. Une fois qu'une règle de cache d'extraction est créée pour un registre public externe, vous pouvez extraire une image de ce registre public externe à l'aide de l'URI (Uniform Resource Identifier) de votre registre privé Amazon ECR. Amazon ECR crée ensuite un référentiel et met l'image en cache. Lorsqu'une image mise en cache est extraite à l'aide de l'URI du registre privé Amazon ECR, Amazon ECR vérifie le registre distant pour voir s'il existe une nouvelle version de l'image et met à jour votre registre privé jusqu'à une fois toutes les 24 heures.

Table des matières

- [Présentation](#)
- [Prérequis](#)
- [Étape 1 : Créez votre cluster EKS pour AWS Batch](#)
- [Étape 2 : Préparez votre cluster EKS pour AWS Batch](#)
- [Étape 3 : Création d'un environnement informatique Amazon EKS](#)
- [Étape 4 : créer une file d'attente de tâches et associer l'environnement informatique](#)
- [Étape 5 : créer un Amazon ECR avec un cache d'extraction](#)
- [Étape 6 : enregistrer une définition de tâche](#)
- [Étape 7 : Soumettre une tâche à exécuter](#)
- [Étape 8 : Afficher le résultat de la tâche](#)
- [Étape 9 : \(Facultatif\) Soumettre une tâche avec des remplacements](#)

- [Étape 10 : Nettoyez les ressources de votre didacticiel](#)
- [Ressources supplémentaires](#)
- [Résolution des problèmes](#)

Présentation

Ce didacticiel explique comment effectuer une configuration AWS Batch avec un Amazon EKS privé à l'aide du AWS CloudShellkubect1 eteksct1.

Public visé

Ce didacticiel est destiné aux administrateurs système et aux développeurs chargés de la configuration, des tests et du déploiement AWS Batch.

Fonctionnalités utilisées

Ce didacticiel vous montre comment utiliser le AWS CLI, pour :

- Utiliser Amazon Elastic Container Registry (Amazon ECR) pour stocker des images de conteneurs
- Création et configuration d'un environnement informatique Amazon EKS
- Créez une file d'attente de tâches.
- Créer une définition de tâche
- Création et soumission d'une tâche à exécuter
- Soumettre une offre d'emploi avec des dérogations

Temps requis

La réalisation de ce didacticiel devrait prendre entre 40 et 50 minutes.

Restrictions régionales

Aucune restriction nationale ou régionale n'est associée à l'utilisation de cette solution.

Coûts d'utilisation des ressources

La création d'un AWS compte est gratuite. Cependant, en mettant en œuvre cette solution, vous pouvez être redevable de tout ou partie des frais répertoriés dans le tableau suivant.

Description	Coût (dollar américain)
Vous êtes facturé en fonction de l'heure du cluster	Varie en fonction de l'instance, voir la tarification d'Amazon EKS
EC2 Instance Amazon	Vous payez pour chaque EC2 instance Amazon créée. Pour plus d'informations sur les tarifs, consultez Amazon EC2 Pricing .

Prérequis

Ce didacticiel utilise AWS CloudShell un shell pré-authentifié basé sur un navigateur que vous lancez directement depuis le AWS Management Console. Cela permet d'accéder au cluster une fois qu'il n'a plus d'accès public à Internet. Le AWS CLI `kubectl`, et `eksctl` peut-être déjà installé dans le cadre de AWS CloudShell. Pour plus d'informations AWS CloudShell, consultez le [guide de AWS CloudShell l'utilisateur](#). Vous pouvez également AWS CloudShell vous connecter au VPC de votre cluster ou à un [réseau connecté](#).

Pour exécuter des commandes `kubectl`, vous devez disposer d'un accès privé à votre cluster Amazon EKS. Cela signifie que tout le trafic vers le serveur API de votre cluster doit provenir du VPC de votre cluster ou d'un réseau connecté.

- **AWS CLI**— Un outil de ligne de commande permettant de travailler avec AWS des services, notamment Amazon EKS. Ce guide nécessite que vous utilisiez la version 2.8.6 ou ultérieure ou la version 1.26.0 ou ultérieure. Pour plus d'informations, consultez la section [Installation, mise à jour et désinstallation du AWS CLI dans le](#) guide de l'AWS Command Line Interface utilisateur. Après l'avoir installé AWS CLI, nous vous recommandons de le configurer également. Pour plus d'informations, consultez la section [Configuration rapide avec aws configure](#) dans le Guide de AWS Command Line Interface l'utilisateur.
- **kubectl** : outil de ligne de commande pour travailler avec des clusters Kubernetes. Ce guide est compatible avec la version 1.23 ou une version ultérieure. Pour plus d'informations, veuillez consulter [Installation ou mise à jour de kubectl](#) dans le Guide de l'utilisateur Amazon EKS.
- **eksctl**— Un outil de ligne de commande permettant de travailler avec des clusters Amazon EKS qui automatise de nombreuses tâches individuelles. Ce guide est compatible avec la version 0.115.0 ou une version ultérieure. Pour plus d'informations, veuillez consulter [Installation ou mise à jour de eksctl](#) dans le Guide de l'utilisateur Amazon EKS.

- Autorisations : les utilisateurs qui appellent l'opération d'[CreateComputeEnvironment](#) API pour créer un environnement informatique utilisant les ressources Amazon EKS ont besoin d'autorisations pour l'opération `eks:ListClusters` d'API `eks:DescribeCluster` et. Vous pouvez associer la politique [AWSBatchFullAccess](#) gérée à votre compte utilisateur en suivant les instructions relatives à l'[ajout et à la suppression d'autorisations d'identité IAM](#) dans le guide de l'utilisateur IAM.
- InstanceRole— Vous devez créer un InstanceRole pour vos nœuds Amazon EKS doté des `AmazonEC2ContainerRegistryPullOnly` politiques `AmazonEKSWorkerNodePolicy` et. Pour savoir comment créer le rôle IAM du nœud Amazon EKS, InstanceRole consultez [Création du rôle IAM du nœud Amazon EKS](#). Vous aurez besoin de l'ARN du InstanceRole.
- Compte AWS ID — Vous devez connaître votre Compte AWS identifiant. Suivez les instructions de la [section Afficher votre Compte AWS identifiant](#).
- (Facultatif) CloudWatch — Pour examiner les détails de [\(Facultatif\) Soumettre une tâche avec des remplacements](#), la journalisation doit être configurée. Pour de plus amples informations, veuillez consulter [Utiliser CloudWatch les journaux pour surveiller AWS Batch les tâches Amazon EKS](#).

Étape 1 : Créez votre cluster EKS pour AWS Batch

Important

Pour démarrer le plus simplement et le plus rapidement possible, ce didacticiel inclut des étapes avec les paramètres par défaut. Avant de créer pour une utilisation en production, nous vous recommandons de vous familiariser avec tous les paramètres et de déployer avec ceux qui répondent à vos besoins.

Nous vous recommandons `eksctl` d'utiliser le fichier de configuration suivant pour créer votre cluster. Pour configurer manuellement votre cluster, suivez les instructions de la section [Déployer des clusters privés avec un accès Internet limité](#) dans le guide de l'utilisateur Amazon EKS.

1. Ouvrez la [AWS CloudShell console](#) et définissez la région `us-east-1`. Pour le reste du didacticiel, assurez-vous que vous utilisez `us-east-1`.
2. Créez un cluster EKS privé dans la `us-east-1` région à l'aide de l'exemple de fichier de `eksctl` configuration. Enregistrez le fichier `yaml` dans votre AWS CloudShell environnement et nommez-le `clusterConfig.yaml`. Vous pouvez modifier `my-test-cluster` le nom que vous souhaitez utiliser pour votre cluster.

```
kind: ClusterConfig
apiVersion: eksctl.io/v1alpha5
metadata:
  name: my-test-cluster
  region: us-east-1
availabilityZones:
  - us-east-1a
  - us-east-1b
  - us-east-1c
managedNodeGroups:
  - name: ng-1
    privateNetworking: true
privateCluster:
  enabled: true
  skipEndpointCreation: false
```

3. Créez vos ressources à l'aide de la commande `eksctl create cluster -f clusterConfig.yaml`. La création d'un cluster peut prendre entre 10 et 15 minutes.
4. Une fois le cluster créé, vous devez ajouter votre adresse AWS CloudShell IP à la liste des autorisations. Pour trouver votre adresse AWS CloudShell IP, exécutez la commande suivante :

```
curl http://checkip.amazonaws.com
```

Une fois que vous avez l'adresse IP publique, vous devez créer une règle de liste d'autorisation :

```
aws eks update-cluster-config \
  --name my-test-cluster \
  --region us-east-1 \
  --resources-vpc-config
endpointPublicAccess=true,endpointPrivateAccess=true,publicAccessCidrs=["<Public
IP>/32"]
```

Appliquez ensuite la mise à jour au fichier de configuration kubectl :

```
aws eks update-kubeconfig --name my-test-cluster --region us-east-1
```

5. Pour vérifier que vous avez accès aux nœuds, exécutez la commande suivante :

```
kubectl get nodes
```

Le résultat de la commande est :

NAME	STATUS	ROLES	AGE	VERSION
ip-192-168-107-235.ec2.internal	Ready	none	1h	v1.32.3-eks-473151a
ip-192-168-165-40.ec2.internal	Ready	none	1h	v1.32.3-eks-473151a
ip-192-168-98-54.ec2.internal	Ready	none	1h	v1.32.1-eks-5d632ec

Étape 2 : Préparez votre cluster EKS pour AWS Batch

Toutes les étapes sont obligatoires et doivent être effectuées AWS CloudShell.

1. Création d'un espace de noms dédié aux tâches AWS Batch

kubectl À utiliser pour créer un nouvel espace de noms.

```
$ namespace=my-aws-batch-namespace
```

```
$ cat - <<EOF | kubectl create -f -
{
  "apiVersion": "v1",
  "kind": "Namespace",
  "metadata": {
    "name": "${namespace}",
    "labels": {
      "name": "${namespace}"
    }
  }
}
EOF
```

Sortie :

```
namespace/my-aws-batch-namespace created
```

2. Permettre l'accès via le contrôle d'accès basé sur les rôles (RBAC)

kubectl À utiliser pour créer un Kubernetes rôle pour le cluster afin de AWS Batch permettre de surveiller les nœuds et les pods, et de lier le rôle. Vous devez effectuer cette opération une fois pour chaque cluster Amazon EKS.


```

$ cat - <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: aws-batch-cluster-role
rules:
  - apiGroups: [""]
    resources: ["namespaces"]
    verbs: ["get"]
  - apiGroups: [""]
    resources: ["nodes"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["events"]
    verbs: ["list"]
  - apiGroups: [""]
    resources: ["configmaps"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["apps"]
    resources: ["daemonsets", "deployments", "statefulsets", "replicasets"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["clusterroles", "clusterrolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: aws-batch-cluster-role-binding
subjects:
  - kind: User
    name: aws-batch
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: aws-batch-cluster-role
  apiGroup: rbac.authorization.k8s.io
EOF

```

Sortie :

```
clusterrole.rbac.authorization.k8s.io/aws-batch-cluster-role created
clusterrolebinding.rbac.authorization.k8s.io/aws-batch-cluster-role-binding created
```

Créez un Kubernetes rôle délimité par un espace de noms pour gérer et prolonger le cycle AWS Batch de vie des pods et les lier. Vous devez effectuer cette opération une fois pour chaque espace de noms unique.

```
$ namespace=my-aws-batch-namespace
```

```
$ cat - <<EOF | kubectl apply -f - --namespace "${namespace}"
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: aws-batch-compute-environment-role
  namespace: ${namespace}
rules:
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["create", "get", "list", "watch", "delete", "patch"]
  - apiGroups: [""]
    resources: ["serviceaccounts"]
    verbs: ["get", "list"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["roles", "rolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: aws-batch-compute-environment-role-binding
  namespace: ${namespace}
subjects:
  - kind: User
    name: aws-batch
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: aws-batch-compute-environment-role
  apiGroup: rbac.authorization.k8s.io
```

EOF

Sortie :

```
role.rbac.authorization.k8s.io/aws-batch-compute-environment-role created
rolebinding.rbac.authorization.k8s.io/aws-batch-compute-environment-role-binding
created
```

Mettez à Kubernetes `aws-auth` jour la carte de configuration pour associer les autorisations RBAC précédentes au rôle lié au service. AWS Batch

```
$ eksctl create iamidentitymapping \
  --cluster my-test-cluster \
  --arn "arn:aws:iam::<your-account-ID>:role/AWSServiceRoleForBatch" \
  --username aws-batch
```

Sortie :

```
2022-10-25 20:19:57 [#] adding identity "arn:aws:iam::<your-account-ID>:role/
AWSServiceRoleForBatch" to auth ConfigMap
```

 Note

Le chemin `aws-service-role/batch.amazonaws.com/` a été supprimé de l'ARN du rôle lié au service. Cela est dû à un problème avec la carte `aws-auth` de configuration. Pour plus d'informations, consultez la section [Les rôles dotés de chemins ne fonctionnent pas lorsque le chemin est inclus dans leur ARN dans le `aws-authconfigmap`](#).

Étape 3 : Création d'un environnement informatique Amazon EKS

AWS Batch les environnements informatiques définissent les paramètres des ressources de calcul pour répondre à vos besoins en matière de charge de travail par lots. Dans un environnement informatique géré, vous AWS Batch aide à gérer la capacité et les types d'instances des ressources de calcul (Kubernetesnœuds) au sein de votre cluster Amazon EKS. Ceci est basé sur la spécification des ressources de calcul que vous définissez lors de la création de l'environnement de calcul. Vous pouvez utiliser des instances EC2 à la demande ou des instances EC2 ponctuelles.

Maintenant que le rôle `AWSServiceRoleForBatch` lié au service a accès à votre cluster Amazon EKS, vous pouvez créer des AWS Batch ressources. Créez d'abord un environnement informatique qui pointe vers votre cluster Amazon EKS.

- Pour subnets exécuter `eksctl get cluster my-test-cluster` pour obtenir les sous-réseaux utilisés par le cluster.
- Pour les `securityGroupIds` paramètres, vous pouvez utiliser le même groupe de sécurité que le cluster Amazon EKS. Cette commande récupère l'ID du groupe de sécurité pour le cluster.

```
$ aws eks describe-cluster \
  --name my-test-cluster \
  --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

- Utilisez l'ARN `instanceRole` que vous avez créé dans les prérequis.

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:us-east-1:<your-account-ID>:cluster/my-test-cluster",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 128,
    "instanceTypes": [
      "m5"
    ],
    "subnets": [
      "<eks-cluster-subnets-with-access-to-the-image-for-image-pull>"
    ],
    "securityGroupIds": [
      "<eks-cluster-sg>"
    ],
    "instanceRole": "<eks-instance-profile>"
  }
}
```

EOF

```
$ aws batch create-compute-environment --cli-input-json file:///./batch-eks-compute-environment.json
```

Remarques

- La maintenance d'un environnement informatique Amazon EKS est une responsabilité partagée. Pour plus d'informations, consultez [la section Sécurité dans Amazon EKS](#).

Étape 4 : créer une file d'attente de tâches et associer l'environnement informatique

Important

Il est important de vérifier que l'environnement informatique est sain avant de continuer. L'opération [DescribeComputeEnvironments](#) API peut être utilisée à cette fin.

```
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1
```

Vérifiez que le status paramètre ne l'est pas `INVALID`. Si c'est le cas, examinez le statusReason paramètre correspondant à la cause. Pour de plus amples informations, veuillez consulter [Résolution des problèmes AWS Batch](#).

Les tâches soumises à cette nouvelle file d'attente de tâches sont exécutées en tant que modules sur des nœuds AWS Batch gérés qui ont rejoint le cluster Amazon EKS associé à votre environnement informatique.

```
$ cat <<EOF > ./batch-eks-job-queue.json
{
  "jobQueueName": "My-Eks-JQ1",
  "priority": 10,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "My-Eks-CE1"
    }
  ]
}
```

```
}  
]  
}  
EOF
```

```
$ aws batch create-job-queue --cli-input-json file:///./batch-eks-job-queue.json
```

Étape 5 : créer un Amazon ECR avec un cache d'extraction

Comme le cluster ne dispose pas d'un accès public à Internet, vous devez créer un Amazon ECR pour les images de conteneur. Les instructions suivantes créent un Amazon ECR avec une règle de cache pull-through pour stocker l'image.

1. La commande suivante crée la règle de cache pull-through. Vous pouvez le remplacer *tutorial-prefix* par un autre préfixe.

```
aws ecr create-pull-through-cache-rule \  
  --ecr-repository-prefix "my-prefix" \  
  --upstream-registry-url "public.ecr.aws" \  
  --region us-east-1
```

2. Authentifiez-vous auprès de l'ECR public.

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --  
password-stdin <your-account-ID>.dkr.ecr.us-east-1.amazonaws.com
```

Vous pouvez maintenant extraire une image.

```
docker pull <your-account-ID>.dkr.ecr.us-east-1.amazonaws.com/my-prefix/  
amazonlinux/amazonlinux:2
```

3. Vous pouvez vérifier le référentiel et l'image en exécutant les commandes suivantes :

```
aws ecr describe-repositories
```

```
aws ecr describe-images --repository-name my-prefix/amazonlinux/amazonlinux
```

4. La chaîne d'image à utiliser pour extraire le conteneur est au format suivant :

```
<your-account-ID>.dkr.ecr.us-east-1.amazonaws.com/my-prefix/amazonlinux/  
amazonlinux:2
```

Étape 6 : enregistrer une définition de tâche

La définition de Job suivante indique au module de rester en veille pendant 60 secondes.

Dans le champ image de la définition de tâche, au lieu de fournir un lien vers une image dans un référentiel ECR public, fournissez le lien vers l'image stockée dans notre référentiel ECR privé.

Consultez l'exemple de définition de tâche suivant :

```
$ cat <<EOF > ./batch-eks-job-definition.json  
{  
  "jobDefinitionName": "MyJobOnEks_Sleep",  
  "type": "container",  
  "eksProperties": {  
    "podProperties": {  
      "hostNetwork": true,  
      "containers": [  
        {  
          "image": "<your-account-ID>.dkr.ecr.us-east-1.amazonaws.com/my-prefix/  
amazonlinux/amazonlinux:2",  
          "command": [  
            "sleep",  
            "60"  
          ],  
          "resources": {  
            "limits": {  
              "cpu": "1",  
              "memory": "1024Mi"  
            }  
          }  
        }  
      ],  
      "metadata": {  
        "labels": {  
          "environment": "test"  
        }  
      }  
    }  
  }  
}
```

```
}  
EOF
```

```
$ aws batch register-job-definition --cli-input-json file:///./batch-eks-job-  
definition.json
```

Remarques

- Il existe des considérations relatives aux memory paramètres cpu et. Pour de plus amples informations, veuillez consulter [Considérations relatives à la mémoire et au vCPU pour AWS Batch Amazon EKS](#).

Étape 7 : Soumettre une tâche à exécuter

Exécutez la AWS CLI commande suivante AWS CloudShell pour soumettre un nouveau Job et renvoie le JoBid unique.

```
$ aws batch submit-job --job-queue My-Eks-JQ1 \  
--job-definition MyJobOnEks_Sleep - -job-name My-Eks-Job1
```

Remarques

- Pour plus d'informations sur l'exécution de tâches sur les ressources Amazon EKS, consultez [Offres d'emploi Amazon EKS](#).

Étape 8 : Afficher le résultat de la tâche

Pour vérifier le statut d'un Job :

```
$ aws batch describe-jobs --job <JobID-from-submit-response>
```

Ils startedAt stoppedAt doivent être espacés d'une minute.

Étape 9 : (Facultatif) Soumettre une tâche avec des remplacements

Cette tâche remplace la commande transmise au conteneur.


```
$ cat <<EOF > ./submit-job-override.json
{
  "jobName": "EksWithOverrides",
  "jobQueue": "My-Eks-JQ1",
  "jobDefinition": "MyJobOnEks_Sleep",
  "eksPropertiesOverride": {
    "podProperties": {
      "containers": [
        {
          "command": [
            "/bin/sh"
          ],
          "args": [
            "-c",
            "echo hello world"
          ]
        }
      ]
    }
  }
}
EOF
```

```
$ aws batch submit-job - -cli-input-json file:///./submit-job-override.json
```

Remarques

- Pour une meilleure visibilité des détails des opérations, activez la journalisation du plan de contrôle Amazon EKS. Pour plus d'informations, consultez la section [Connexion au plan de contrôle Amazon EKS](#) dans le guide de l'utilisateur Amazon EKS.
- Daemonsetset la kubelets surcharge affecte les ressources de vCPU et de mémoire disponibles, en particulier le dimensionnement et le placement des tâches. Pour de plus amples informations, veuillez consulter [Considérations relatives à la mémoire et au vCPU pour AWS Batch Amazon EKS](#).

Étape 10 : Nettoyez les ressources de votre didacticiel

L'EC2 instance Amazon vous est facturée lorsqu'elle est activée. Vous pouvez supprimer l'instance pour ne plus encourir de frais.

Pour supprimer les ressources que vous avez créées, procédez comme suit :

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans le volet de navigation, choisissez Job queue.
3. Dans le tableau des files d'attente de tâches, choisissez la file d'attente de tâches que vous avez créée pour le didacticiel.
4. Dans Actions, choisissez Désactiver. Une fois que l'état de la file d'attente des Job est désactivé, vous pouvez choisir Delete.
5. Une fois la file d'attente de Job supprimée, dans le volet de navigation, sélectionnez Compute environments.
6. Choisissez l'environnement informatique que vous avez créé pour ce didacticiel, puis choisissez Désactiver des actions. La désactivation complète de l'environnement informatique peut prendre 1 à 2 minutes.
7. Une fois que l'état de l'environnement informatique est désactivé, choisissez Supprimer. La suppression de l'environnement informatique peut prendre 1 à 2 minutes.

Ressources supplémentaires

Une fois le didacticiel terminé, vous souhaitez peut-être explorer les sujets suivants :

- En savoir plus sur les [meilleures pratiques](#).
- Explorez les AWS Batch principaux composants. Pour de plus amples informations, veuillez consulter [Composants de AWS Batch](#).
- En savoir plus sur les différents [environnements informatiques](#) disponibles dans AWS Batch.
- En savoir plus sur les [files d'attente de Job](#) et leurs différentes options de planification.
- En savoir plus sur [les définitions de Job](#) et les différentes options de configuration.
- En savoir plus sur les différents types d'[emplois](#).

Résolution des problèmes

Si les nœuds lancés par AWS Batch n'ont pas accès au référentiel Amazon ECR (ou à tout autre référentiel) qui stocke votre image, vos tâches peuvent rester à l'état STARTING. Cela est dû au fait que le module ne pourra pas télécharger l'image et exécuter votre AWS Batch tâche. Si vous cliquez sur le nom du pod lancé par, AWS Batch vous devriez pouvoir voir le message d'erreur et confirmer le problème. Le message d'erreur doit ressembler à ce qui suit :

```
Failed to pull image "public.ecr.aws/amazonlinux/amazonlinux:2": rpc error: code =
Unknown desc = failed to pull and unpack image
"public.ecr.aws/amazonlinux/amazonlinux:2": failed to resolve reference
"public.ecr.aws/amazonlinux/amazonlinux:2": failed to do request: Head
"https://public.ecr.aws/v2/amazonlinux/amazonlinux/manifests/2": dial tcp: i/o timeout
```

Pour d'autres scénarios de dépannage courants, consultez la section [Résolution des problèmes AWS Batch](#). Pour un dépannage basé sur l'état du pod, consultez [Comment résoudre le problème de l'état du pod dans Amazon EKS ?](#).

Commencer à utiliser AWS Batch l' SageMaker IA

AWS Batch les tâches de service vous permettent de soumettre des tâches de SageMaker formation par le biais de files AWS Batch d'attente de tâches avec des fonctionnalités de planification, de priorisation et de mise en file d'attente. Ce didacticiel explique comment configurer et exécuter une tâche de SageMaker formation simple à l'aide de tâches AWS Batch de service.

Table des matières

- [Présentation de](#)
- [Conditions préalables](#)
- [Étape 1 : Création d'un rôle d'exécution SageMaker AI](#)
- [Étape 2 : Création de votre environnement de service](#)
- [Étape 3 : Créez votre file SageMaker d'attente de tâches](#)
- [Étape 4 : Création et soumission d'un poste de formation](#)
- [Étape 5 : Surveiller l'état du travail](#)
- [Étape 6 : Afficher le résultat de la tâche](#)
- [Étape 7 : Nettoyez les ressources de votre didacticiel](#)
- [Ressources supplémentaires](#)

Présentation de

Ce didacticiel explique comment configurer des tâches AWS Batch de service pour des tâches de SageMaker formation à l'aide du AWS CLI.

Public visé

Ce didacticiel est destiné aux scientifiques des données et aux développeurs chargés de configurer et d'exécuter des tâches de formation en apprentissage automatique à grande échelle.

Fonctionnalités utilisées

Ce didacticiel vous montre comment utiliser le AWS CLI pour :

- Créer un environnement de service pour les emplois SageMaker de formation
- Création d'une file d'attente SageMaker de tâches de formation
- Soumettre des tâches de service à l'aide de l'`SubmitServiceJobAPI`
- Surveillez l'état des tâches et visualisez les résultats
- CloudWatch Journaux d'accès aux offres de formation

Temps requis

Ce didacticiel dure environ 15 minutes.

Restrictions régionales

Ce didacticiel peut être suivi dans n'importe quelle AWS région où les deux AWS Batch et SageMaker l'IA sont disponibles.

Coûts d'utilisation des ressources

La création d'un AWS compte est gratuite. Cependant, la mise en œuvre de cette solution peut entraîner des coûts pour les ressources suivantes :

Description	Coût (dollar américain)
SageMaker Instances de formation à l'IA	Vous payez pour chaque instance d' SageMaker AI Training utilisée. Pour plus d'informations sur la tarification, consultez la section Tarification de l'SageMaker IA .
Stockage Amazon S3	Coût minimal pour le stockage des résultats des tâches de formation. Pour plus d'informations, consultez Tarification Amazon S3 .

Conditions préalables

Avant de commencer ce didacticiel, vous devez installer et configurer les outils et ressources suivants dont vous avez besoin pour créer et gérer à la fois AWS Batch des ressources d' SageMaker IA.

- AWS CLI— Un outil de ligne de commande pour travailler avec AWS les services, y compris AWS Batch l' SageMaker IA. Ce guide nécessite que vous utilisiez la version 2.8.6 ou ultérieure. Pour plus d'informations, consultez la section [Installation, mise à jour et désinstallation du AWS CLI dans le](#) guide de l'AWS Command Line Interface utilisateur. Après l'avoir installé AWS CLI, nous vous recommandons de le configurer également. Pour plus d'informations, voir [Configuration rapide avec aws configure](#) dans le Guide de AWS Command Line Interface l'utilisateur.

Étape 1 : Création d'un rôle d'exécution SageMaker AI

SageMaker L'IA utilise des rôles d'exécution pour effectuer des opérations en votre nom à l'aide d'autres AWS services. Vous devez créer un rôle d'exécution et accorder à l' SageMaker IA les autorisations nécessaires pour utiliser les services et les ressources nécessaires aux tâches de formation. Utilisez la politique AmazonSageMakerFullAccess gérée car elle inclut des autorisations pour Amazon S3.

Note

Suivez les instructions ci-dessous pour créer le rôle d'exécution SageMaker AI pour ce didacticiel.

Avant de créer un rôle d'exécution pour votre environnement de production, nous vous recommandons de consulter le [guide du développeur SageMaker AI intitulé Comment utiliser les rôles d'exécution de l'SageMaker IA](#).

1. Création du rôle IAM

Créez un fichier JSON nommé `sagemaker-trust-policy.json` avec la politique de confiance suivante :

JSON

```
{
```

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

Créez le rôle IAM à l'aide de la politique de confiance :

```
aws iam create-role \
  --role-name SageMakerExecutionRole \
  --assume-role-policy-document file://sagemaker-trust-policy.json \
  --description "Execution role for SageMaker training jobs"
```

2. Joindre des politiques gérées

Associez les politiques gérées requises au rôle :

```
aws iam attach-role-policy \
  --role-name SageMakerExecutionRole \
  --policy-arn arn:aws:iam::aws:policy/AmazonSageMakerFullAccess
```

```
aws iam attach-role-policy \
  --role-name SageMakerExecutionRole \
  --policy-arn arn:aws:iam::aws:policy/AmazonS3FullAccess
```

3. Notez le rôle ARN

Obtenez l'ARN du rôle, dont vous aurez besoin ultérieurement :

```
aws iam get-role --role-name SageMakerExecutionRole --query 'Role.Arn' --output
text
```

Enregistrez cet ARN tel que vous l'utiliserez lors de la création de la charge utile de votre tâche de formation.

Étape 2 : Création de votre environnement de service

Un environnement de service définit les contraintes de capacité pour les postes de SageMaker formation. L'environnement de service encapsule le nombre maximum d'instances de formation pouvant être exécutées simultanément.

Important

Lorsque vous créez votre premier environnement de service pour SageMaker Training, un rôle lié au service est `AWSBatchWithSageMaker` dans votre compte. Ce rôle permet de mettre en file d'attente et de gérer les tâches de SageMaker formation en votre nom. Pour plus d'informations sur ce rôle lié à un service et ses autorisations, consultez [la section appelée "SageMaker rôle d'intégration"](#)

Créez un environnement de service capable de gérer jusqu'à 5 instances :

```
aws batch create-service-environment \
  --service-environment-name TutorialServiceEnvironment \
  --service-environment-type SAGEMAKER_TRAINING \
  --capacity-limits capacityUnit=NUM_INSTANCES,maxCapacity=5
```

Sortie :

```
{
  "serviceEnvironmentName": "TutorialServiceEnvironment",
  "serviceEnvironmentArn": "arn:aws:batch:your-region:your-account-id:service-
environment/TutorialServiceEnvironment"
}
```

Vérifiez que votre environnement de service a bien été créé :

```
aws batch describe-service-environments --service-
environments TutorialServiceEnvironment
```

Sortie :

```
{
```

```

    "serviceEnvironments": [
      {
        "serviceEnvironmentName": "TutorialServiceEnvironment",
        "serviceEnvironmentArn": "arn:aws:batch:your-region:your-account-
id:service-environment/TutorialServiceEnvironment",
        "serviceEnvironmentType": "SAGEMAKER_TRAINING",
        "state": "ENABLED",
        "status": "VALID",
        "capacityLimits": [
          {
            "maxCapacity": 5,
            "capacityUnit": "NUM_INSTANCES"
          }
        ],
        "tags": {}
      }
    ]
  }
}

```

Pour plus d'informations sur les environnements de service, consultez [Environnements de service](#).

Étape 3 : Créez votre file SageMaker d'attente de tâches

Une file d'attente de SageMaker tâches gère la planification et l'exécution des tâches de service. Les tâches soumises à cette file d'attente seront envoyées à votre environnement de service en fonction de la capacité disponible.

Créez une file d'attente SageMaker de tâches de formation :

```

aws batch create-job-queue \
  --job-queue-name my-sm-training-fifo-jq \
  --job-queue-type SAGEMAKER_TRAINING \
  --priority 1 \
  --service-environment-order order=1,serviceEnvironment=TutorialServiceEnvironment

```

Sortie :

```

{
  "jobQueueName": "my-sm-training-fifo-jq",
  "jobQueueArn": "arn:aws:batch:your-region:your-account-id:job-queue/my-sm-training-
fifo-jq"
}

```


Vérifiez que votre file d'attente de tâches a bien été créée :

```
aws batch describe-job-queues --job-queues my-sm-training-fifo-jq
```

Sortie :

```
{
  "jobQueues": [
    {
      "jobQueueName": "my-sm-training-fifo-jq",
      "jobQueueArn": "arn:aws:batch:your-region:your-account-id:job-queue/my-sm-training-fifo-jq",
      "state": "ENABLED",
      "status": "VALID",
      "statusReason": "JobQueue Healthy",
      "priority": 1,
      "computeEnvironmentOrder": [],
      "serviceEnvironmentOrder": [
        {
          "order": 1,
          "serviceEnvironment": "arn:aws:batch:your-region:your-account-id:service-environment/TutorialServiceEnvironment"
        }
      ],
      "jobQueueType": "SAGEMAKER_TRAINING",
      "tags": {}
    }
  ]
}
```

Pour plus d'informations sur les files SageMaker d'attente de tâches, consultez [the section called “Création d'une file SageMaker d'attente de tâches”](#).

Étape 4 : Création et soumission d'un poste de formation

Vous allez maintenant créer une tâche de formation simple et la soumettre à votre liste d'offres d'emploi. Cet exemple utilise une tâche de formation de base « Bonjour tout le monde » qui illustre les fonctionnalités d'une tâche de service.

Créez un fichier nommé *my_training_job.json* avec le contenu suivant. Remplacez *your-account-id* par votre identifiant de AWS compte :

Note

S3OutputPath est nécessaire pour créer la tâche de SageMaker formation, mais les résultats de ce didacticiel ne sont pas stockés dans le compartiment Amazon S3 et vous pouvez utiliser le chemin dans le JSON suivant. Dans votre environnement de production, vous aurez besoin d'un compartiment Amazon S3 valide pour y stocker la sortie si vous le souhaitez.

```
{
  "TrainingJobName": "my-simple-training-job",
  "RoleArn": "arn:aws:iam::your-account-id:role/SageMakerExecutionRole",
  "AlgorithmSpecification": {
    "TrainingInputMode": "File",
    "TrainingImage": "763104351884.dkr.ecr.us-west-2.amazonaws.com/pytorch-
training:2.0.0-cpu-py310",
    "ContainerEntrypoint": [
      "echo",
      "hello world"
    ]
  },
  "ResourceConfig": {
    "InstanceType": "ml.c5.xlarge",
    "InstanceCount": 1,
    "VolumeSizeInGB": 1
  },
  "OutputDataConfig": {
    "S3OutputPath": "s3://your-s3-bucket/output"
  },
  "StoppingCondition": {
    "MaxRuntimeInSeconds": 30
  }
}
```

Soumettez le poste de formation à l'aide de l'[SubmitServiceJobAPI](#) :

```
aws batch submit-service-job \
  --job-queue my-sm-training-fifo-jq \
  --job-name my-batch-sm-job \
  --service-job-type SAGEMAKER_TRAINING \
  --retry-strategy attempts=1 \
```

```
--timeout-config attemptDurationSeconds=60 \
--service-request-payload file://my_training_job.json
```

Sortie :

```
{
  "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",
  "jobName": "my-batch-sm-job",
  "jobId": "your-job-id"
}
```

Pour plus d'informations sur les charges utiles des tâches de service, consultez [the section called “Charges utiles des tâches de maintenance”](#). Pour plus d'informations sur la soumission de tâches de service, consultez [the section called “Soumettre une offre d'emploi”](#).

Étape 5 : Surveiller l'état du travail

Vous pouvez suivre vos tâches de formation à l'aide des AWS Batch APIs méthodes suivantes : [DescribeServiceJobListServiceJobs](#), et [GetJobQueueSnapshot](#). Cette section présente différentes méthodes pour vérifier l'état des tâches et les informations relatives à la file d'attente.

Consultez les tâches en cours dans votre file d'attente :

```
aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq --job-status RUNNING
```

Sortie :

```
{
  "jobSummaryList": [
    {
      "latestAttempt": {
        "serviceResourceId": {
          "name": "TrainingJobArn",
          "value": "arn:aws:sagemaker:your-region:your-account-id:training-job/AWSBatch<my-simple-training-job><your-attempt-id>"
        }
      },
      "createdAt": 1753718760,
      "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",

```

```

        "jobId": "your-job-id",
        "jobName": "my-batch-sm-job",
        "serviceJobType": "SAGEMAKER_TRAINING",
        "status": "RUNNING",
        "startedAt": 1753718820
      }
    ]
  }

```

Afficher les offres d'emploi disponibles dans l'`RUNNABLE` État :

```

aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq --job-status RUNNABLE

```

Obtenez un aperçu des tâches à venir dans votre file d'attente :

```

aws batch get-job-queue-snapshot --job-queue my-sm-training-fifo-jq

```

Sortie :

```

{
  "frontOfQueue": {
    "jobs": [
      {
        "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",
        "earliestTimeAtPosition": 1753718880
      },
      {
        "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id-2",
        "earliestTimeAtPosition": 1753718940
      }
    ],
    "lastUpdatedAt": 1753718970
  }
}

```

Rechercher des offres d'emploi par nom :

```

aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq \

```

```
--filters name=JOB_NAME,values="my-batch-sm-job"
```

Sortie :

```
{
  "jobSummaryList": [
    {
      "latestAttempt": {
        "serviceResourceId": {
          "name": "TrainingJobArn",
          "value": "arn:aws:sagemaker:your-region:your-account-id:training-job/AWSBatch<my-simple-training-job><your-attempt-id>"
        }
      },
      "createdAt": 1753718760,
      "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",
      "jobId": "your-job-id",
      "jobName": "my-batch-sm-job",
      "serviceJobType": "SAGEMAKER_TRAINING",
      "status": "RUNNING"
    }
  ]
}
```

Pour plus d'informations sur le mappage de l'état des tâches, consultez [the section called “État du poste de service”](#).

Étape 6 : Afficher le résultat de la tâche

Une fois votre tâche terminée, vous pouvez consulter ses résultats et ses journaux grâce à l'API SageMaker IA. AWS Batch

Obtenez des informations détaillées sur votre travail auprès de AWS Batch :

```
aws batch describe-service-job \
  --job-id your-job-id
```

Sortie :

```
{
  "attempts": [
```

```

    {
      "serviceResourceId": {
        "name": "TrainingJobArn",
        "value": "arn:aws:sagemaker:your-region:your-account-id:training-job/
AWSBatch<my-simple-training-job><your-attempt-id>"
      },
      "startedAt": 1753718820,
      "stoppedAt": 1753718880,
      "statusReason": "Received status from SageMaker: Training job completed"
    }
  ],
  "createdAt": 1753718760,
  "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",
  "jobId": "your-job-id",
  "jobName": "my-batch-sm-job",
  "jobQueue": "arn:aws:batch:your-region:your-account-id:job-queue/my-sm-training-fifo-jq",
  "latestAttempt": {
    "serviceResourceId": {
      "name": "TrainingJobArn",
      "value": "arn:aws:sagemaker:your-region:your-account-id:training-job/
AWSBatch<my-simple-training-job><your-attempt-id>"
    }
  },
  "retryStrategy": {
    "attempts": 1,
    "evaluateOnExit": []
  },
  "serviceRequestPayload": "your-training-job-request-json",
  "serviceJobType": "SAGEMAKER_TRAINING",
  "startedAt": 1753718820,
  "status": "SUCCEEDED",
  "statusReason": "Received status from SageMaker: Training job completed",
  "stoppedAt": 1753718880,
  "tags": {},
  "timeoutConfig": {
    "attemptDurationSeconds": 60
  }
}

```

Cette commande renvoie des informations complètes sur la tâche, y compris l'ARN de la tâche de SageMaker formation, que vous pouvez utiliser pour accéder à la tâche directement via l' `SageMaker IA` :

```
aws sagemaker describe-training-job \  
  --training-job-name AWSBatch<my-simple-training-job><your-attempt-id>
```

Pour consulter les CloudWatch journaux relatifs à votre tâche de formation, obtenez d'abord le nom du flux de journaux :

```
aws logs describe-log-streams \  
  --log-group-name /aws/sagemaker/TrainingJobs \  
  --log-stream-name-prefix AWSBatchmy-simple-training-job
```

Sortie :

```
{  
  "logStreams": [  
    {  
      "logStreamName": "your-log-stream-name",  
      "creationTime": 1753718830,  
      "firstEventTimestamp": 1753718840,  
      "lastEventTimestamp": 1753718850,  
      "lastIngestionTime": 1753718860,  
      "uploadSequenceToken": upload-sequence-token,  
      "arn": "arn:aws:logs:your-region:your-account-id:log-group:/aws/sagemaker/  
TrainingJobs:log-stream:AWSBatch<my-simple-training-job><your-attempt-id>/algo-1-algo-  
id",  
      "storedBytes": 0  
    }  
  ]  
}
```

Récupérez ensuite les journaux en utilisant le nom du flux de journaux indiqué dans la réponse précédente :

```
aws logs get-log-events \  
  --log-group-name /aws/sagemaker/TrainingJobs \  
  --log-stream-name your-log-stream-name
```

Sortie :

```
{  
  "events": [  

```

```
{
  "timestamp": 1753718845,
  "message": "hello world",
  "ingestionTime": 1753718865
},
"nextForwardToken": "next-forward-token",
"nextBackwardToken": "next-backward-token"
}
```

La sortie du journal affiche le message « Bonjour tout le monde » envoyé par votre tâche de formation, confirmant que la tâche a été exécutée correctement.

Étape 7 : Nettoyez les ressources de votre didacticiel

Lorsque vous avez terminé le didacticiel, nettoyez les ressources que vous avez créées pour éviter des frais récurrents.

Tout d'abord, désactivez et supprimez la file d'attente des tâches :

```
aws batch update-job-queue \
  --job-queue my-sm-training-fifo-jq \
  --state DISABLED
```

Attendez que la file d'attente des tâches soit désactivée, puis supprimez-la :

```
aws batch delete-job-queue \
  --job-queue my-sm-training-fifo-jq
```

Ensuite, désactivez et supprimez l'environnement de service :

```
aws batch update-service-environment \
  --service-environment TutorialServiceEnvironment \
  --state DISABLED
```

Attendez que l'environnement de service soit désactivé, puis supprimez-le :

```
aws batch delete-service-environment \
  --service-environment TutorialServiceEnvironment
```


Ressources supplémentaires

Une fois le didacticiel terminé, vous souhaitez peut-être explorer les sujets suivants :

- Nous vous recommandons d'utiliser le PySDK pour créer des tâches de service et les soumettre à votre file d'attente de tâches, car PySDK possède des classes d'assistance et des utilitaires. Pour un exemple d'utilisation de PySDK, consultez les [exemples d'SageMaker IA](#) sur GitHub
- En savoir plus sur [the section called “Emplois dans les services”](#).
- Découvrez [Charges utiles des tâches de maintenance dans AWS Batch](#) des configurations de tâches de formation plus complexes.
- Apprenez-en davantage sur [Soumettre une offre d'emploi dans le secteur des services AWS Batch](#) l'SubmitServiceJobAPI.
- Passez [Mappage AWS Batch de l'état de la tâche du service avec SageMaker le statut de](#) en revue pour comprendre les transitions entre les emplois et les états.
- Consultez la [documentation du SDK SageMaker AI Python](#) pour découvrir d'autres méthodes riches en fonctionnalités pour créer et soumettre des tâches de SageMaker formation à l'aide de Python.
- Découvrez des [SageMakerexemples de blocs-notes](#) pour des flux de travail de machine learning plus complexes.

Le tableau de bord AWS Batch du widget

Grâce au AWS Batch tableau de bord, vous pouvez surveiller les tâches récentes, les files d'attente de tâches et les environnements informatiques. Par défaut, les widgets de tableau de bord suivants sont affichés :

- Vue d'ensemble des tâches : pour plus d'informations sur les AWS Batch offres d'emploi, consultez [Tâches](#).
- Vue d'ensemble des files d'attente de tâches : pour plus d'informations sur les files d'attente de AWS Batch tâches, consultez [Files d'attente pour les jobs](#).
- Vue d'ensemble de l'environnement informatique — Pour plus d'informations sur les environnements AWS Batch informatiques, voir [Environnements informatiques pour AWS Batch](#).

Vous pouvez personnaliser les widgets affichés sur la page du tableau de bord. Les sections suivantes décrivent les widgets supplémentaires que vous pouvez ajouter.

Rubriques

- [Comment ajouter le widget de file d'attente de tâches unique au AWS Batch tableau de bord](#)
- [Comment ajouter le widget CloudWatch Container Insights au AWS Batch tableau de bord](#)
- [Comment ajouter le widget Job logs au AWS Batch tableau de bord](#)

Comment ajouter le widget de file d'attente de tâches unique au AWS Batch tableau de bord

Le widget File d'attente de tâches unique affiche des informations détaillées sur une file d'attente de tâches unique.

Pour ajouter ce widget, procédez comme suit.

1. Ouvrez la [AWS Batch console](#).
2. Dans la barre de navigation, sélectionnez celui Région AWS que vous souhaitez.
3. Dans le panneau de navigation, sélectionnez Dashboard (Tableau de bord).
4. Sélectionnez Ajouter widgets.
5. Pour File d'attente de tâches unique, choisissez Ajouter un widget.

6. Pour la file d'attente des tâches, sélectionnez la file d'attente des tâches que vous souhaitez.
7. Dans le champ État de la tâche, choisissez les statuts de la tâche que vous souhaitez afficher.
8. (Facultatif) Désactivez Afficher les environnements informatiques connectés si vous ne souhaitez pas afficher les propriétés des environnements informatiques.
9. Pour les propriétés de l'environnement de calcul, sélectionnez les propriétés souhaitées.
10. Choisissez Ajouter.

Comment ajouter le widget CloudWatch Container Insights au AWS Batch tableau de bord

Ce widget affiche des métriques agrégées pour les environnements de AWS Batch calcul et les tâches. Pour plus d'informations sur Container Insights, consultez [the section called “CloudWatch Informations sur les conteneurs”](#).

Pour ajouter ce widget, procédez comme suit.

1. Ouvrez la [AWS Batch console](#).
2. Dans la barre de navigation, sélectionnez celle Région AWS que vous souhaitez.
3. Dans le panneau de navigation, sélectionnez Dashboard (Tableau de bord).
4. Sélectionnez Ajouter widgets.
5. Pour obtenir des informations sur les conteneurs, choisissez Ajouter un widget.
6. Pour Environnement de calcul, choisissez l'environnement de calcul de votre choix.
7. Choisissez Ajouter.

Comment ajouter le widget Job logs au AWS Batch tableau de bord

Ce widget affiche différents journaux de vos tâches en un seul endroit pratique. Pour plus d'informations sur les journaux des tâches, consultez [the section called “Afficher les journaux des tâches dans CloudWatch Logs”](#).

Pour ajouter ce widget, procédez comme suit.

1. Ouvrez la [AWS Batch console](#).
2. Dans la barre de navigation, sélectionnez celle Région AWS que vous souhaitez.

3. Dans le panneau de navigation, sélectionnez Dashboard (Tableau de bord).
4. Sélectionnez Ajouter widgets.
5. Pour Job logs, choisissez Ajouter un widget.
6. Dans le champ Job id, entrez le Job ID du job que vous souhaitez.
7. Choisissez Ajouter.

Environnements informatiques pour AWS Batch

Les files d'attente de tâches sont mappées à un ou plusieurs environnements informatiques. Les environnements de calcul contiennent les instances de conteneur Amazon ECS qui sont utilisées pour exécuter des tâches par lots conteneurisées. Un environnement informatique spécifique peut également être mappé à une ou plusieurs files d'attente de tâches. Au sein d'une file d'attente de tâches, les environnements informatiques associés ont chacun un ordre utilisé par le planificateur pour déterminer où les tâches prêtes à être exécutées seront exécutées. Si le premier environnement de calcul a un statut de `VALID` et dispose de ressources disponibles, la tâche est planifiée sur une instance de conteneur au sein de cet environnement informatique. Si le premier environnement de calcul a un statut de `INVALID` ou ne peut pas fournir de ressource de calcul appropriée, le planificateur tente d'exécuter la tâche sur l'environnement de calcul suivant.

Rubriques

- [Environnements informatiques gérés](#)
- [Environnements informatiques non gérés](#)
- [Création d'un environnement informatique](#)
- [Mettre à jour un environnement informatique dans AWS Batch](#)
- [Ressource de calcul AMIs](#)
- [Utilisez les modèles de lancement Amazon EC2 avec AWS Batch](#)
- [Configuration du service de métadonnées d'instance \(IMDS\)](#)
- [EC2 configurations](#)
- [Stratégies d'allocation des types d'instance pour AWS Batch](#)
- [Gestion de la mémoire des ressources informatiques](#)
- [Environnements informatiques Fargate](#)
- [Environnements informatiques Amazon EKS](#)

Environnements informatiques gérés

Vous pouvez utiliser un environnement informatique géré pour AWS Batch gérer la capacité et les types d'instances des ressources de calcul au sein de l'environnement. Ceci est basé sur les spécifications des ressources de calcul que vous définissez lors de la création de l'environnement de calcul. Vous pouvez choisir d'utiliser les instances Amazon EC2 On-Demand et les instances

Amazon EC2 Spot. Vous pouvez également utiliser les capacités Fargate et Fargate Spot dans votre environnement informatique géré. Lorsque vous utilisez des instances Spot, vous pouvez éventuellement définir un prix maximum. Ainsi, les instances Spot ne sont lancées que lorsque le prix des instances Spot est inférieur à un pourcentage spécifié du prix à la demande.

 Important

Les instances Fargate Spot ne sont pas prises en charge sur Windows containers on AWS Fargate. Une file d'attente de tâches sera bloquée si une FargateWindows tâche est soumise à une file d'attente de tâches qui utilise uniquement les environnements informatiques Fargate Spot.

 Important

AWS Batch crée et gère plusieurs AWS ressources en votre nom et au sein de votre compte, notamment les modèles de lancement Amazon EC2, les groupes Amazon EC2 Auto Scaling, les flottes Amazon EC2 Spot et les clusters Amazon ECS. Ces ressources gérées sont configurées spécifiquement pour garantir un AWS Batch fonctionnement optimal. La modification manuelle de ces ressources gérées par lots, sauf indication explicite dans la AWS Batch documentation, peut entraîner un comportement inattendu entraînant un environnement de INVALID calcul, un comportement de dimensionnement des instances sous-optimal, un retard dans le traitement de la charge de travail ou des coûts imprévus. Ces modifications manuelles ne peuvent pas être prises en charge de manière déterministe par le AWS Batch service. Utilisez toujours le Batch compatible APIs ou la console Batch pour gérer vos environnements informatiques.

Les environnements informatiques gérés lancent des instances Amazon EC2 dans le VPC et les sous-réseaux que vous spécifiez, puis les enregistrent auprès d'un cluster Amazon ECS. Les instances Amazon EC2 ont besoin d'un accès réseau externe pour communiquer avec le point de terminaison du service Amazon ECS. Certains sous-réseaux ne fournissent pas d'adresses IP publiques aux instances Amazon EC2. Si vos instances Amazon EC2 n'ont pas d'adresse IP publique, elles doivent utiliser la traduction d'adresses réseau (NAT) pour obtenir cet accès. Pour plus d'informations, veuillez consulter [NAT Gateways \(Passerelles NAT\)](#) dans le Guide de l'utilisateur Amazon VPC. Pour plus d'informations sur la création d'un VPC, consultez. [Créer un Virtual Private Cloud](#)

Par défaut, les environnements informatiques AWS Batch gérés utilisent une version récente et approuvée de l'AMI optimisée Amazon ECS pour les ressources de calcul. Cependant, vous souhaitez peut-être créer votre propre AMI à utiliser dans vos environnements informatiques gérés pour diverses raisons. Pour de plus amples informations, veuillez consulter [Ressource de calcul AMIs](#).

Note

AWS Batch ne le met pas automatiquement à niveau AMIs dans un environnement informatique après sa création. Par exemple, il ne met pas à jour le AMIs dans votre environnement informatique lorsqu'une nouvelle version de l'AMI optimisée Amazon ECS est publiée. Vous êtes responsable de la gestion du système d'exploitation client. Cela inclut les mises à jour et les correctifs de sécurité. Vous êtes également responsable de tout logiciel d'application ou utilitaire supplémentaire que vous installez sur les ressources informatiques. Il existe deux manières d'utiliser une nouvelle AMI pour vos AWS Batch tâches. La méthode initiale consiste à procéder comme suit :

1. Créez un nouvel environnement de calcul avec la nouvelle AMI.
2. Ajoutez l'environnement de calcul à une file d'attente de tâches existante.
3. Supprimez l'environnement de calcul antérieur de votre file d'attente de tâches.
4. Supprimez l'environnement de calcul antérieur.

En avril 2022, AWS Batch ajout d'un support amélioré pour la mise à jour des environnements informatiques. Pour de plus amples informations, veuillez consulter [Mettre à jour un environnement informatique dans AWS Batch](#). Pour utiliser la mise à jour améliorée des environnements informatiques à des fins de mise à jour AMIs, suivez les règles suivantes :

- Ne définissez pas le paramètre du rôle de service ([serviceRole](#)) ou définissez-le sur le rôle AWSServiceRoleForBatchlié au service.
- Définissez le paramètre de stratégie d'allocation ([allocationStrategy](#)) sur `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED` ou `SPOT_PRICE_CAPACITY_OPTIMIZED`.
- Définissez le paramètre de mise à jour vers la dernière version de l'image ([updateToLatestImageVersion](#)) sur `true`.

- Ne spécifiez pas d'ID d'AMI dans [imageId](#), [imageIdOverride](#)(in [ec2Configuration](#)) ou dans le modèle de lancement ([launchTemplate](#)). Dans ce cas, AWS Batch sélectionne l'AMI optimisée pour Amazon ECS la plus récente prise en charge AWS Batch au moment du lancement de la mise à jour de l'infrastructure. Vous pouvez également spécifier l'ID de l'AMI dans les `imageIdOverride` paramètres `imageId` ou dans le modèle de lancement identifié par les `LaunchTemplate` propriétés. La modification de l'une de ces propriétés déclenche une mise à jour de l'infrastructure. Si l'ID d'AMI est spécifié dans le modèle de lancement, il ne peut pas être remplacé en spécifiant un ID d'AMI dans les `imageIdOverride` paramètres `imageId` ou. Il ne peut être remplacé qu'en spécifiant un modèle de lancement différent. Ou, si la version du modèle de lancement est définie sur `$Default` ou `$Latest`, en définissant une nouvelle version par défaut pour le modèle de lancement (si c'est le cas `$Default`) ou en ajoutant une nouvelle version au modèle de lancement (si c'est le cas `$Latest`).

Si ces règles sont respectées, toute mise à jour qui lance une mise à jour de l'infrastructure entraîne une nouvelle sélection de l'ID AMI. Si le [version](#) paramètre du modèle de lancement ([launchTemplate](#)) est défini sur `$Latest` ou `$Default`, la version la plus récente ou par défaut du modèle de lancement est évaluée au moment de la mise à jour de l'infrastructure, même si elle n'[launchTemplate](#)a pas été mise à jour.

Prise en compte lors de la création de tâches parallèles à nœuds multiples

AWS Batch recommande de créer des environnements informatiques dédiés pour exécuter des tâches multi-nœuds parallèles (MNP) et des tâches non MNP. Cela est dû à la manière dont la capacité de calcul est créée dans votre environnement informatique géré. Lorsque vous créez un nouvel environnement informatique géré, si vous spécifiez une `minvCpu` valeur supérieure à zéro, vous AWS Batch créez un pool d'instances à utiliser uniquement avec des tâches non MNP. Si une tâche parallèle à nœuds multiples est soumise, AWS Batch crée une nouvelle capacité d'instance pour exécuter les tâches parallèles à nœuds multiples. Dans les cas où des tâches parallèles à nœud unique et à nœuds multiples sont exécutées dans le même environnement informatique où une `maxvCpus` valeur `minvCpus` ou une valeur est définie, si les ressources de calcul requises ne sont pas disponibles, il AWS Batch faudra attendre la fin des tâches en cours avant de créer les ressources de calcul nécessaires pour exécuter les nouvelles tâches.

Environnements informatiques non gérés

Dans un environnement informatique non géré, vous gérez vos propres ressources informatiques. AWS Batch prend en charge les environnements informatiques non gérés pour Amazon ECS et Amazon EKS, ce qui vous permet de garder le contrôle de votre infrastructure tout en tirant parti des fonctionnalités de planification des tâches de Batch.

Note

AWS Les ressources Fargate ne sont pas prises en charge dans les environnements informatiques non gérés.

Environnements informatiques Amazon ECS non gérés

Pour les environnements de calcul Amazon ECS non gérés, vous devez vérifier que l'AMI que vous utilisez pour vos ressources de calcul répond aux spécifications de l'AMI d'instance de conteneur Amazon ECS. Pour plus d'informations, consultez [Spécification d'AMI de ressources de calcul](#) et [Tutoriel : Création d'une AMI de ressources de calcul](#).

Après avoir créé votre environnement de calcul non géré, utilisez l'opération d'[DescribeComputeEnvironments](#) API pour afficher les détails de l'environnement de calcul. Trouvez le cluster Amazon ECS associé à l'environnement, puis lancez manuellement vos instances de conteneur dans ce cluster Amazon ECS.

La AWS CLI commande suivante fournit également l'ARN du cluster Amazon ECS.

```
$ aws batch describe-compute-environments \
  --compute-environments unmanagedCE \
  --query "computeEnvironments[].ecsClusterArn"
```

Pour plus d'informations, reportez-vous à la section consacrée au [Lancement d'une instance de conteneur Amazon ECS](#) dans le Guide du développeur Amazon Elastic Container Service. Lorsque vous lancez vos ressources de calcul, spécifiez l'ARN du cluster Amazon ECS que les ressources enregistrent avec les données utilisateur Amazon EC2 suivantes. **ecsClusterArn** Remplacez-le par l'ARN du cluster que vous avez obtenu avec la commande précédente.

```
#!/bin/bash
```

```
echo "ECS_CLUSTER=ecsClusterArn" >> /etc/ecs/ecs.config
```

Environnements informatiques Amazon EKS non gérés

Dans un environnement informatique Amazon EKS non géré, vous gérez vos propres nœuds Kubernetes tout en AWS Batch gérant la planification et le placement des tâches. Il vous permet de contrôler directement votre infrastructure Kubernetes en termes de sécurité, de conformité ou d'exigences opérationnelles. Vous êtes responsable du provisionnement et de la configuration de vos nœuds Amazon EKS, tout en les AWS Batch intégrant à votre cluster Amazon EKS existant pour planifier et exécuter des tâches.

Pour plus d'informations, consultez [Tutoriel : Création d'un environnement de calcul non géré à l'aide des ressources Amazon EKS](#).

Création d'un environnement informatique

Avant de pouvoir exécuter des tâches AWS Batch, vous devez créer un environnement informatique. Vous pouvez créer un environnement informatique géré AWS Batch qui gère les instances Amazon EC2 ou les ressources Fargate au sein de l'environnement en AWS fonction de vos spécifications. Vous pouvez également créer un environnement de calcul non géré dans lequel vous gérez la configuration de l'instance Amazon EC2 au sein de l'environnement.

Important

Les instances Fargate Spot ne sont pas prises en charge dans les scénarios suivants :

- Windows containers on AWS Fargate

Dans ces scénarios, une file d'attente de tâches sera bloquée si une tâche est soumise à une file d'attente de tâches qui utilise uniquement les environnements informatiques Fargate Spot.

Rubriques

- [Tutoriel : Création d'un environnement informatique géré à l'aide des ressources Fargate](#)
- [Tutoriel : Création d'un environnement informatique géré à l'aide EC2 des ressources Amazon](#)
- [Tutoriel : Création d'un environnement informatique non géré à l'aide des ressources Amazon EC2](#)
- [Tutoriel : Création d'un environnement informatique géré à l'aide des ressources Amazon EKS](#)

- [Tutoriel : Création d'un environnement informatique non géré à l'aide des ressources Amazon EKS](#)
- [Ressource : modèle d'environnement de calcul](#)
- [Table de calcul du type d'instance](#)

Tutoriel : Création d'un environnement informatique géré à l'aide des ressources Fargate

Procédez comme suit pour créer un environnement informatique géré à l'aide de AWS Fargate ressources.

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation, sélectionnez le Région AWS à utiliser.
3. Dans le panneau de navigation, choisissez Environnements de calcul.
4. Choisissez Créer.
5. Configurez l'environnement informatique.

Note

Les environnements de calcul pour les Windows containers on AWS Fargate tâches doivent disposer d'au moins un vCPU.

- a. Pour la configuration de l'environnement de calcul, choisissez Fargate.
 - b. Dans Nom, spécifiez un nom unique pour votre environnement informatique. Le nom peut contenir jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
 - c. Pour le rôle de service, choisissez un rôle lié au service qui permet au AWS Batch service d'appeler les opérations d' AWS API requises en votre nom. Par exemple, sélectionnez AWSServiceRoleForBatch. Pour de plus amples informations, veuillez consulter [Utilisation des rôles liés à un service pour AWS Batch](#).
 - d. (Facultatif) Développez les balises. Pour ajouter une balise, choisissez Ajouter une balise. Entrez ensuite un nom de clé et une valeur facultative. Choisissez Ajouter une balise.
 - e. Choisissez Page suivante.
6. Dans la section Configuration de l'instance :

- a. (Facultatif) Pour utiliser la capacité Fargate Spot, activez Fargate Spot. Pour plus d'informations sur Fargate Spot, consultez la section [Utilisation d' EC2 Amazon Spot](#) et de Fargate_spot.
 - b. Pour Maximum v CPUs, choisissez le nombre maximum de v CPUs que votre environnement informatique peut atteindre, quelle que soit la demande de la file d'attente des tâches.
 - c. Choisissez Page suivante.
7. Configurez la mise en réseau.

 Important

Les ressources de calcul ont besoin de communiquer avec le point de terminaison de service Amazon ECS service. Cela peut être via un point de terminaison d'un VPC d'interface ou via vos ressources de calcul ayant des adresses IP publiques.

Pour plus d'informations sur les points de terminaison d'un VPC d'interface, veuillez consulter [Points de terminaison d'un VPC d'interface Amazon ECS AWS PrivateLink](#) dans le Guide du développeur Amazon Elastic Container Service.

Si vous n'avez pas de point de terminaison d'un VPC d'interface configuré et que vos ressources de calcul n'ont pas d'adresses IP publiques, elles doivent utiliser la traduction d'adresse réseau (NAT) pour fournir cet accès. Pour de plus amples informations, veuillez consulter [Passerelles NAT](#) dans le Guide de l'utilisateur Amazon VPC. Pour de plus amples informations, veuillez consulter [the section called "Création d'un VPC"](#).

- a. Pour l'ID Virtual Private Cloud (VPC), choisissez un VPC sur lequel vous souhaitez lancer vos instances.
- b. Pour les sous-réseaux, choisissez les sous-réseaux à utiliser. Par défaut, tous les sous-réseaux du VPC sélectionné sont disponibles.

 Note

AWS Batch sur Fargate ne prend actuellement pas en charge les Zones Locales. Pour plus d'informations, consultez les [clusters Amazon ECS dans Local Zones, Wavelength Zones et AWS Outposts](#) dans le manuel Amazon Elastic Container Service Developer Guide.

- c. Dans Groupes de sécurité, choisissez un groupe de sécurité à attacher à vos instances. Le groupe de sécurité par défaut pour votre VPC est choisi.
 - d. Choisissez Page suivante.
8. Pour la révision, passez en revue les étapes de configuration. Si vous devez apporter des modifications, choisissez Modifier. Lorsque vous avez terminé, choisissez Create computing environment.

Tutoriel : Création d'un environnement informatique géré à l'aide EC2 des ressources Amazon

Procédez comme suit pour créer un environnement informatique géré à l'aide des ressources Amazon Elastic Compute Cloud (Amazon EC2).

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation, sélectionnez le Région AWS à utiliser.
3. Dans le panneau de navigation, choisissez Environments (Environnements).
4. Choisissez Créer un environnement, puis Compute environment.
5. Configurez l'environnement.
 - a. Pour la configuration de l'environnement de calcul, choisissez Amazon Elastic Compute Cloud (Amazon EC2).
 - b. Pour le type d'orchestration, choisissez Managed.
 - c. Dans Nom, spécifiez un nom unique pour votre environnement informatique. Le nom peut contenir jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
 - d. Pour le rôle de service, choisissez un rôle lié au service qui permet au AWS Batch service d'appeler les opérations d' AWS API requises en votre nom. Par exemple, sélectionnez AWSServiceRoleForBatch. Pour de plus amples informations, veuillez consulter [Utilisation des rôles liés à un service pour AWS Batch](#).
 - e. Pour Rôle d'instance, choisissez de créer un nouveau profil d'instance ou d'utiliser un profil d'instance existant disposant des autorisations IAM requises attachées. Ce profil d'instance permet aux instances de conteneur Amazon ECS créées pour votre environnement informatique d'appeler les opérations d' AWS API requises en votre nom. Pour de plus

amples informations, veuillez consulter [Rôle d'instance Amazon ECS](#). Si vous choisissez de créer un nouveau profil d'instance, le rôle requis (ecsInstanceRole) est créé pour vous.

- f. (Facultatif) Développez les balises.
 - i. (Facultatif) Pour les EC2 balises, choisissez Ajouter une balise pour ajouter une balise aux ressources lancées dans l'environnement informatique. Entrez ensuite un nom de clé et une valeur facultative. Choisissez Ajouter une balise.
 - ii. (Facultatif) Pour Tags, choisissez Ajouter un tag. Entrez ensuite un nom de clé et une valeur facultative. Choisissez Ajouter une balise.

Pour de plus amples informations, veuillez consulter [Marquez vos AWS Batch ressources](#).

- g. Choisissez Suivant.
6. Dans la section Configuration de l'instance :
- a. (Facultatif) Pour Activer l'utilisation d'instances Spot, activez Spot. Pour plus d'informations, consultez [Instances Spot](#).
 - b. (Spot uniquement) Pour le pourcentage maximum du prix à la demande, choisissez le pourcentage maximum que peut être le prix d'une instance Spot par rapport au prix à la demande pour ce type d'instance avant le lancement des instances. Par exemple, si votre prix maximum est de 20 %, le prix spot doit être inférieur à 20 % du prix à la demande actuel pour cette EC2 instance. Vous payez toujours le prix le plus bas (marché) et jamais plus que votre pourcentage maximal. Si vous laissez ce champ vide, la valeur par défaut est 100 % du prix à la demande.
 - c. (Spot uniquement) Pour le rôle de flotte Spot, choisissez un rôle Amazon EC2 Spot Fleet IAM existant à appliquer à votre environnement informatique Spot. Si vous ne possédez pas encore de rôle Amazon EC2 Spot Fleet IAM existant, vous devez d'abord en créer un. Pour de plus amples informations, veuillez consulter [Rôle de la flotte Amazon EC2 Spot](#).

 Important

Pour étiqueter vos instances Spot lors de leur création, votre rôle IAM Amazon EC2 Spot Fleet doit utiliser la nouvelle politique EC2 SpotFleetTaggingRole gérée par Amazon. La politique EC2 SpotFleetRole gérée par Amazon ne dispose pas des autorisations requises pour étiqueter les instances Spot. Pour plus d'informations,

consultez [Instances ponctuelles non étiquetées lors de la création](#) et [the section called “Baliser vos ressources”](#).

- d. Pour Minimum v CPUs, choisissez le nombre minimum de v maintenu par votre environnement informatique, quelle CPUs que soit la demande de la file d'attente des tâches.
- e. Pour Desired v CPUs, choisissez le nombre de v avec CPUs lequel votre environnement informatique démarre. À mesure que la demande de votre file d'attente augmente, vous AWS Batch pouvez augmenter le nombre de v souhaité CPUs dans votre environnement informatique et ajouter des EC2 instances, jusqu'au maximum CPUs v. À mesure que la demande diminue, vous AWS Batch pouvez diminuer le nombre de v souhaité CPUs dans votre environnement informatique et supprimer des instances, jusqu'au minimum CPUs v.
- f. Pour Maximum v CPUs, choisissez le nombre maximum de v CPUs que votre environnement informatique peut atteindre, quelle que soit la demande de la file d'attente des tâches.
- g. Pour les types d'instances autorisés, choisissez les types d' EC2 instances Amazon qui peuvent être lancés. Vous pouvez spécifier des familles d'instances pour lancer n'importe quel type d'instance au sein de ces familles (par exemple c5c5n,, oup3). Vous pouvez également spécifier des tailles spécifiques au sein d'une famille (par exemple c5.8xlarge). Les types d'instances Metal ne font pas partie des familles d'instances. Par exemple, c5 n'inclut pas c5.metal.

AWS Batch peut sélectionner le type d'instance qui vous convient si vous choisissez l'une des options suivantes :

- `optimal` pour sélectionner les types d'instances (parmi les familles d'instances c4, m4, r4, c5, m5,, et) qui répondent à la demande de vos files d'attente de travail.
- `default_x86_64` pour choisir des types d'instances basés sur x86 (parmi les familles d'instances m6i, c6i, r6i,, et) qui correspondent aux besoins en ressources de la file d'attente des tâches.
- `default_arm64` pour choisir des types d'instances basés sur x86 (parmi les familles d'instances m6g, c6g, r6g,, et) qui correspondent aux besoins en ressources de la file d'attente des tâches.

Note

À partir du 01/11/2025, le comportement `optimal` de sera modifié pour correspondre. `default_x86_64` Pendant le changement, vos familles d'instances peuvent être mises à jour vers une génération plus récente. Vous n'avez aucune action à effectuer pour que la mise à niveau ait lieu. Pour plus d'informations sur les modifications, consultez

Note

À partir du 01/11/2025, le comportement `optimal` de sera modifié pour correspondre. `default_x86_64` Pendant le

changement, vos familles d'instances peuvent être mises à

jour vers une génération plus récente. Vous n'avez aucune action à effectuer pour que la mise à niveau ait lieu.

AWS Batch supportait une seule option dans `InstanceTypes` `optimal` pour répondre à la demande de vos files d'attente de travail. Nous avons introduit deux nouvelles options de type d'instance : `default_x86_64` et `default_arm64`. Nous l'utiliserons `default_x86_64` si vous ne sélectionnez aucun type d'instance. Ces nouvelles options sélectionneront automatiquement des types d'instances rentables issus de différentes familles et générations en fonction des exigences de votre file d'attente, ce qui vous permettra d'exécuter rapidement vos charges de travail.

Dès qu'une capacité suffisante de nouveaux types d'instances devient disponible dans un Région AWS, le pool par défaut correspondant sera automatiquement mis à jour avec le nouveau type d'instance. L'option existante continuera d'être prise en charge et n'est pas obsolète, car elle sera prise en charge par les pools par défaut sous-jacents afin de fournir des instances mises à jour à l'avenir. Si vous utilisez `optimal`, aucune action de votre part n'est requise.

Sachez toutefois que seuls **ENABLED** les environnements de **VALID** calcul (CEs) seront automatiquement mis à jour avec les nouveaux types d'instances. Si vous en avez **INVALID** CEs, ils recevront **DISABLED** des mises à jour une fois qu'ils seront réactivés et qu'ils auront atteint un **VALID** état.

 Note

- La disponibilité de la famille d'instances varie en fonction de Région AWS. Par exemple, certaines Région AWS s peuvent ne pas avoir de familles d'instances de quatrième génération, mais avoir des familles d'instances de cinquième et sixième générations.
- Lorsque vous utilisez `default_x86_64` des ensembles d'`default_arm64` instances, sélectionnez AWS Batch des familles d'instances en fonction d'un équilibre entre rentabilité et performances. Bien que les instances de nouvelle génération offrent souvent un meilleur rapport prix/performances, vous AWS Batch pouvez choisir une famille d'instances de génération antérieure si elle offre la combinaison optimale de disponibilité, de coût et de performances pour votre charge de travail. Par exemple, dans un pays Région AWS où les instances `c6i` et `c7i` sont disponibles, vous AWS Batch pouvez sélectionner des instances `c6i` si elles offrent un meilleur rapport coût-efficacité pour les exigences spécifiques de votre travail. Pour plus d'informations sur les types d' AWS Batch instances et leur Région AWS disponibilité, consultez la section [Table de calcul des types d'instances](#).
- AWS Batch met régulièrement à jour vos instances dans les bundles par défaut vers des options plus récentes et plus économiques. Les mises à jour se font automatiquement sans aucune action de votre part. Vos charges de travail continuent de s'exécuter sans interruption pendant les mises à jour.

 Note

Quand vous créez un environnement de calcul, les types d'instances que vous sélectionnez pour cet environnement doivent partager la même architecture. Par exemple, vous ne pouvez pas mélanger des instances x86 et ARM dans le même environnement de calcul.

 Note

AWS Batch évoluera GPUs en fonction de la quantité requise dans vos files d'attente de travail. Pour utiliser la planification GPU, l'environnement de calcul doit inclure les types d'instances des g6 familles p3 p4 p5p6,g3,g3s,g4,g5,, ou.

- h. Pour Allocation strategy (Stratégie d'allocation), choisissez la stratégie d'allocation à utiliser lorsque vous sélectionnez les types d'instance dans la liste des types d'instance autorisés. BEST_FIT_PROGRESSIVE est généralement le meilleur choix pour les environnements de calcul à la EC2 demande, SPOT_CAPACITY_OPTIMIZED et SPOT_PRICE_CAPACITY_OPTIMIZED pour les environnements de calcul ponctuels. EC2 Pour de plus amples informations, veuillez consulter [the section called “Stratégies d'allocation des types d'instances”](#).
- i. Développez Additional configuration (Configuration supplémentaire).
 - i. (Facultatif) Pour Groupe de placement, entrez un nom de groupe de placement pour regrouper les ressources dans l'environnement informatique.
 - ii. (Facultatif) Pour la paire de EC2 clés, choisissez une paire de clés publique et privée comme informations d'identification de sécurité lorsque vous vous connectez à l'instance. Pour plus d'informations sur les paires de EC2 clés Amazon, consultez la section [Paires de EC2 clés Amazon et instances Linux](#).
 - iii. (Facultatif) Pour EC2 la configuration, choisissez les valeurs de remplacement du type d'image et de l'ID d'image AWS Batch pour fournir des informations permettant de sélectionner Amazon Machine Images (AMIs) pour les instances dans l'environnement informatique. Si le remplacement de l'ID d'image n'est pas spécifié pour chaque type d'image, AWS Batch sélectionne une [AMI récemment optimisée pour Amazon ECS](#). Si

aucun type d'image n'est spécifié, la valeur par défaut est une instance Amazon Linux 2 pour une instance autre que le GPU ou AWS Graviton.

 Important

Pour utiliser une AMI personnalisée, choisissez le type d'image, puis entrez l'ID d'AMI personnalisé dans la zone de remplacement de l'ID d'image.

[Amazon Linux 2](#)

Valeur par défaut pour toutes les familles d'instances AWS basées sur Graviton (par exemple, C6g M6gR6g, etT4g) et peut être utilisée pour tous les types d'instances autres que le GPU.

[Amazon Linux 2 \(GPU\)](#)

Par défaut pour toutes les familles d'instances GPU (par exemple P4 etG4) et peut être utilisé pour tous les types d'instances non AWS basés sur Graviton.

[Amazon Linux 2023](#)

AWS Batch prend en charge Amazon Linux 2023.

 Note

Amazon Linux 2023 ne prend pas en charge A1 les instances.

[Amazon Linux 2023 \(GPU\)](#)

Par défaut pour toutes les familles d'instances GPU (par exemple P4 etG4) et peut être utilisé pour tous les types d'instances non AWS basés sur Graviton.

 Note

L'AMI que vous choisissez pour un environnement de calcul doit correspondre à l'architecture des types d'instances que vous prévoyez d'utiliser pour cet environnement de calcul. Par exemple, si votre environnement de calcul utilise des types d'instances A1, l'AMI de ressource de calcul que vous choisissez

doit prendre en charge les instances ARM. Amazon ECS vend les versions x86 et ARM de l'AMI Amazon Linux 2 optimisée pour Amazon ECS. Pour plus d'informations, consultez l'[AMI Amazon Linux 2 optimisée pour Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.

- j. (Facultatif) Développez les modèles de lancement
- i. Pour le modèle de lancement par défaut, sélectionnez un modèle de EC2 lancement Amazon existant pour configurer vos ressources de calcul. La version par défaut du modèle est automatiquement renseignée. Pour de plus amples informations, veuillez consulter [Utilisez les modèles de lancement Amazon EC2 avec AWS Batch](#).

 Note

Dans un modèle de lancement, vous pouvez spécifier une AMI personnalisée que vous avez créée.

- ii. (Facultatif) Pour Version par défaut `$Default`, entrez `$Latest`, ou un numéro de version spécifique à utiliser.

 Note

Remarque : Si vous utilisez l'une des variables de substitution (`$Default` ou `$Latest`), elle appliquera le numéro de version par défaut actuel ou le dernier au moment de l'enregistrement de cette configuration. Si la version par défaut ou la dernière version change à l'avenir, vous devez mettre à jour les informations, elles ne seront pas automatiquement mises à jour.

 Important

Si le paramètre de version du modèle de lancement est `$Default` ou `$Latest`, la version par défaut ou la plus récente du modèle de lancement spécifié est évaluée lors d'une mise à jour de l'infrastructure. Si un autre ID d'AMI est sélectionné par défaut ou si la dernière version du modèle de lancement est sélectionnée, cet ID d'AMI est utilisé dans la mise à jour. Pour de plus amples

informations, veuillez consulter [the section called "Sélection des AMI lors des mises à jour de l'infrastructure"](#).

- iii. (Facultatif) Pour le modèle de lancement de remplacement, choisissez Ajouter un modèle de lancement de remplacement
 - A. (Facultatif) Pour le modèle de lancement, sélectionnez un modèle de EC2 lancement Amazon existant à utiliser pour des types et des familles d'instances spécifiques.
 - B. (Facultatif) Pour Version par défaut, entrez un numéro de version spécifique à utiliser `$Default`, ou `$Latest`.

 Note

Si vous utilisez la `$Latest` variable `$Default` or, elle AWS Batch appliquera les informations actuelles au moment de la création de l'environnement de calcul. Si la version par défaut ou la dernière version change à l'avenir, vous devez mettre à jour les informations via [UpdateComputeEnvironment](#) ou via le AWS Management Console - AWS Batch.

- C. (Facultatif) Pour les types d'instances Target, sélectionnez le type ou la famille d'instances auxquels vous souhaitez appliquer le modèle de lancement de remplacement.

 Note

Si vous spécifiez un modèle de lancement de remplacement, les types d'instances cibles sont requis. Pour plus d'informations, consultez [LaunchTemplateSpecificationOverride.targetInstanceTypes](#).

 Note

Si le type ou la famille d'instances que vous souhaitez sélectionner n'apparaît pas dans cette liste, passez en revue les sélections que vous y avez effectuées `Allowed instance types`.

k. Choisissez Suivant.

7. Dans la section Configuration du réseau :

 Important

Les ressources de calcul ont besoin de communiquer avec le point de terminaison de service Amazon ECS service. Cela peut être via un point de terminaison d'un VPC d'interface ou via vos ressources de calcul ayant des adresses IP publiques.

Pour plus d'informations sur les points de terminaison d'un VPC d'interface, veuillez consulter [Points de terminaison d'un VPC d'interface Amazon ECS AWS PrivateLink](#) dans le Guide du développeur Amazon Elastic Container Service.

Si vous n'avez pas de point de terminaison d'un VPC d'interface configuré et que vos ressources de calcul n'ont pas d'adresses IP publiques, elles doivent utiliser la traduction d'adresse réseau (NAT) pour fournir cet accès. Pour de plus amples informations, veuillez consulter [Passerelles NAT](#) dans le Guide de l'utilisateur Amazon VPC. Pour de plus amples informations, veuillez consulter [the section called "Création d'un VPC"](#).

- a. Pour l'ID Virtual Private Cloud (VPC), choisissez un VPC sur lequel lancer vos instances.
- b. Pour les sous-réseaux, choisissez les sous-réseaux à utiliser. Par défaut, tous les sous-réseaux du VPC sélectionné sont disponibles.

 Note

AWS Batch sur Amazon EC2 prend en charge les Zones Locales. Pour plus d'informations, consultez [Local Zones](#) dans le guide de EC2 l'utilisateur Amazon et [les clusters Amazon ECS dans Local Zones, Wavelength Zones et AWS Outposts](#) dans le manuel du développeur Amazon Elastic Container Service.

- c. (Facultatif) Pour les groupes de sécurité, choisissez un groupe de sécurité à associer à vos instances. Le groupe de sécurité par défaut pour votre VPC est choisi.

 Note

Remarque : Si vous utilisez l'une des variables de substitution (\$Default ou \$Latest), elle appliquera le numéro de version par défaut actuel ou le dernier au moment de l'enregistrement de cette configuration. Si la version par défaut ou la dernière

version change à l'avenir, vous devez mettre à jour les informations, elles ne seront pas automatiquement mises à jour.

8. Choisissez Page suivante.
9. Pour la révision, passez en revue les étapes de configuration. Si vous devez apporter des modifications, choisissez Modifier. Lorsque vous avez terminé, choisissez Create computing environment.

Tutoriel : Création d'un environnement informatique non géré à l'aide des ressources Amazon EC2

Procédez comme suit pour créer un environnement de calcul non géré à l'aide des ressources Amazon Elastic Compute Cloud (Amazon EC2).

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation, sélectionnez le Région AWS à utiliser.
3. Sur la page Environnements de calcul, choisissez Create.
4. Configurez l'environnement.
 - a. Pour la configuration de l'environnement de calcul, choisissez Amazon Elastic Compute Cloud (Amazon EC2).
 - b. Pour le type d'orchestration, choisissez Non géré.
5. Dans Nom, spécifiez un nom unique pour votre environnement informatique. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
6. Pour Rôle de service, choisissez un rôle qui permet au AWS Batch service d'appeler les opérations d' AWS API requises en votre nom.

Note

Vous ne pouvez pas l'utiliser AWSServiceRoleForBatch pour des environnements informatiques non gérés.

7. Pour Maximum v CPUs, choisissez le nombre maximum de v CPUs que votre environnement informatique peut atteindre, quelle que soit la demande de la file d'attente des tâches.

8. (Facultatif) Développez les balises. Pour ajouter une balise, choisissez Ajouter une balise. Entrez ensuite un nom de clé et une valeur facultative. Choisissez Ajouter une balise. Pour de plus amples informations, veuillez consulter [Marquez vos AWS Batch ressources](#).
9. Choisissez Page suivante.
10. Pour la révision, passez en revue les étapes de configuration. Si vous devez apporter des modifications, choisissez Modifier. Lorsque vous avez terminé, choisissez Créer un environnement de calcul.

Tutoriel : Création d'un environnement informatique géré à l'aide des ressources Amazon EKS

Procédez comme suit pour créer un environnement informatique géré à l'aide des ressources Amazon Elastic Kubernetes Service (Amazon EKS).

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation, sélectionnez le Région AWS à utiliser.
3. Dans le panneau de navigation, choisissez Environnements de calcul.
4. Choisissez Créer.
5. Pour la configuration de l'environnement informatique, choisissez Amazon Elastic Kubernetes Service (Amazon EKS).
6. Dans Nom, spécifiez un nom unique pour votre environnement informatique. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
7. Pour Rôle d'instance, choisissez un profil d'instance existant auquel sont attachées les autorisations IAM requises.

Note

Pour créer un environnement de calcul dans la AWS Batch console, choisissez un profil d'instance doté des `eks:DescribeCluster` autorisations `eks:ListClusters` et.

8. Pour le cluster EKS, choisissez un cluster Amazon EKS existant.
9. Pour Namespace, entrez un espace de Kubernetes noms pour regrouper vos AWS Batch processus dans le cluster.

10. (Facultatif) Développez les balises. Choisissez Ajouter une balise, puis entrez une paire clé-valeur.
11. Choisissez Page suivante.
12. (Facultatif) Pour utiliser des instances EC2 Spot, activez Activer l'utilisation d'instances Spot pour utiliser des instances EC2 Spot Amazon.
13. (Spot uniquement) Pour le pourcentage maximum du prix à la demande, choisissez le pourcentage maximum que peut être le prix d'une instance Spot par rapport au prix à la demande pour ce type d'instance avant le lancement des instances. Par exemple, si votre prix maximum est de 20 %, le prix spot doit être inférieur à 20 % du prix à la demande actuel pour cette EC2 instance. Vous payez toujours le prix le plus bas (marché) et jamais plus que votre pourcentage maximal. Si vous laissez ce champ vide, la valeur par défaut est 100 % du prix à la demande.
14. (Spot uniquement) Pour le rôle de flotte Spot, choisissez le rôle IAM de flotte Amazon EC2 Spot pour l'environnement SPOT informatique.

 Important

Ce rôle est obligatoire si la stratégie d'allocation est définie BEST_FIT ou non spécifiée.

15. (Facultatif) Pour Minimum v CPUs, choisissez le nombre minimum de v maintenu par votre environnement informatique, quelle CPUs que soit la demande de la file d'attente des tâches.
16. (Facultatif) Pour Maximum v CPUs, choisissez le nombre maximum de v CPUs que votre environnement informatique peut atteindre, quelle que soit la demande de la file d'attente des tâches.
17. Pour les types d'instances autorisés, choisissez les types d' EC2 instances Amazon qui peuvent être lancés. Vous pouvez spécifier des familles d'instances pour lancer n'importe quel type d'instance au sein de ces familles (par exemple c5c5n,, oup3). Vous pouvez également spécifier des tailles spécifiques au sein d'une famille (par exemple c5.8xlarge). Les types d'instances Metal ne font pas partie des familles d'instances. Par exemple, c5 n'inclut pas c5.meta1.

AWS Batch peut sélectionner le type d'instance qui vous convient si vous choisissez l'une des options suivantes :

- optimal pour sélectionner les types d'instances (parmi les familles d'instances c4, m4, m5, et) qui répondent à la demande de vos files d'attente de travail.

- `default_x86_64` pour choisir des types d'instances basés sur x86 (parmi les familles d'instances `m6i`, `c6ir6i`, et) qui correspondent aux besoins en ressources de la file d'attente des tâches.
- `default_arm64` pour choisir des types d'instances basés sur ARM (parmi les familles d'instances `m6g`, `c6gr6g`, et) qui correspondent aux besoins en ressources de la file d'attente des tâches.

 Note

À partir du 01/11/2025, le comportement optimal de sera modifié pour correspondre. `default_x86_64` Pendant le changement, vos familles d'instances peuvent être mises à jour vers une génération plus récente. Vous n'avez aucune action à effectuer pour que la mise à niveau ait lieu. Pour plus d'informations sur les modifications, consultez [Configuration optimale du type d'instance pour recevoir des mises à jour automatiques de la famille d'instances](#).

 Note

- La disponibilité de la famille d'instances varie en fonction de Région AWS. Par exemple, certaines Région AWS peuvent ne pas avoir de familles d'instances de quatrième génération mais avoir des familles d'instances de cinquième et sixième générations.
- Lorsque vous utilisez `default_x86_64` des ensembles d'instances `default_arm64`, sélectionnez AWS Batch des familles d'instances en fonction d'un équilibre entre rentabilité et performances. Bien que les instances de nouvelle génération offrent souvent un meilleur rapport prix/performances, vous AWS Batch pouvez choisir une famille d'instances de génération antérieure si elle offre la combinaison optimale de disponibilité, de coût et de performances pour votre charge de travail. Par exemple, dans un pays Région AWS où les instances `c6i` et `c7i` sont disponibles, vous AWS Batch pouvez sélectionner des instances `c6i` si elles offrent un meilleur rapport coût-efficacité pour les exigences spécifiques de votre travail. Pour plus d'informations sur les types d'instances AWS Batch et leur Région AWS disponibilité, consultez la section [Table de calcul des types d'instances](#).

- AWS Batch met régulièrement à jour vos instances dans les bundles par défaut vers des options plus récentes et plus économiques. Les mises à jour se font automatiquement sans aucune action de votre part. Vos charges de travail continuent de s'exécuter pendant les mises à jour sans interruption

 Note

Quand vous créez un environnement de calcul, les types d'instances que vous sélectionnez pour cet environnement doivent partager la même architecture. Par exemple, vous ne pouvez pas mélanger des instances x86 et ARM dans le même environnement de calcul.

 Note

AWS Batch évoluera GPUs en fonction de la quantité requise dans vos files d'attente de travail. Pour utiliser la planification GPU, l'environnement de calcul doit inclure les types d'instances des g6 familles p3 p4 p5p6,g3,g3s,g4,g5,, ou.

18. (Facultatif) Développez la configuration supplémentaire.

- (Facultatif) Pour Groupe de placement, entrez un nom de groupe de placement pour regrouper les ressources dans l'environnement informatique.
- Pour la stratégie d'allocation, choisissez BEST_FIT_PROGRESSIVE.
- (Facultatif) Pour la configuration Amazon Machine Images (AMIs), choisissez Ajouter la configuration Amazon Machine Images (amis).

Vous pouvez utiliser une AMI Amazon Linux optimisée pour Amazon EKS ou une AMI personnalisée.

- Pour utiliser une [AMI Amazon Linux optimisée pour Amazon EKS](#) :

A. Pour Type d'image, sélectionnez l'une des options suivantes :

- [Amazon Linux 2](#) : par défaut pour toutes les familles d'instances AWS basées sur Graviton (par exemple,, C6g M6gR6g, etT4g) et peut être utilisé pour tous les types d'instances autres que le GPU.

- [Amazon Linux 2 \(accéléré\)](#) : par défaut pour toutes les familles d'instances GPU (par exemple, P4 et G4) et peut être utilisé pour tous les types d'instances non AWS basés sur Graviton.
 - [Amazon Linux 2023](#) : AWS Batch compatible avec Amazon Linux 2023 (AL2023).
 - [Amazon Linux 2023 \(accéléré\)](#) : familles d'instances GPU pouvant être utilisées pour tous les types d'instances non AWS basés sur Graviton.
- B. Pour Kubernetes la version, entrez un [numéro de Kubernetes version](#).
- ii. Pour utiliser une AMI personnalisée :
- A. Pour Type d'image, choisissez le type d'AMI sur lequel est basée l'AMI personnalisée :
- [Amazon Linux 2](#) : par défaut pour toutes les familles d'instances AWS basées sur Graviton (par exemple, C6g, M6gR6g, et T4g) et peut être utilisé pour tous les types d'instances autres que le GPU.
 - [Amazon Linux 2 \(accéléré\)](#) : par défaut pour toutes les familles d'instances GPU (par exemple, P4 et G4) et peut être utilisé pour tous les types d'instances non AWS basés sur Graviton.
 - [Amazon Linux 2023](#) : AWS Batch prend en charge AL2 023.
 - [Amazon Linux 2023 \(accéléré\)](#) : familles d'instances GPU pouvant être utilisées pour tous les types d'instances non AWS basés sur Graviton.
- B. Pour remplacer l'ID d'image, entrez l'ID d'AMI personnalisé.
- C. Pour Kubernetes la version, entrez un [numéro de Kubernetes version](#).
- d. (Facultatif) Pour Modèle de lancement, choisissez un [modèle de lancement](#) existant.
- e. (Facultatif) Pour la version du modèle de lancement **\$Default**, entrez **\$Latest**, ou un numéro de version.
- f. (Facultatif) Pour le modèle de lancement Override, pour ajouter un override, choisissez Add override Launch template :
- i. (Facultatif) Dans Modèle de lancement, choisissez le modèle de lancement auquel ajouter la dérogation.
 - ii. (Facultatif) Pour la version du modèle de lancement, choisissez le numéro de version du modèle de lancement **\$Default**, ou **\$Latest**.

- iii. (Facultatif) Pour les types d'instances cibles, choisissez le type ou la famille d'instance auxquels cette dérogation doit être appliquée. Cela ne peut cibler que les types d'instances et les familles inclus dans les types d'instances autorisés.
- iv. (Facultatif) Pour UserData Type, choisissez l'initialisation du nœud EKS. N'utilisez ce champ que si une AMI est spécifiée dans le modèle de lancement ou en tant que remplacement du modèle de lancement. Choisissez EKS_NODEADM pour la personnalisation AMIs basée sur **EKS_AL2023 EKS_AL2023_NVIDIA** ou EKS_BOOSTRAP_SH pour et. EKS_AL2 EKS_AL_NVIDIA La valeur par défaut est EKS_BOOSTRAP_SH.

Vous utiliseriez UserData Type lorsque vous disposez d'un [environnement mixte](#) dans lequel vous utilisez à la fois AL2 un environnement personnalisé AL2 basé sur 023 AMIs dans le même environnement informatique.

- 19. Choisissez Page suivante.
- 20. Pour l'ID Virtual Private Cloud (VPC), choisissez un VPC sur lequel lancer les instances.
- 21. Pour les sous-réseaux, choisissez les sous-réseaux à utiliser. Par défaut, tous les sous-réseaux du VPC sélectionné sont disponibles.

 Note

AWS Batch sur Amazon EKS prend en charge les Zones Locales. Pour plus d'informations, consultez [Amazon EKS et Zones AWS Locales](#) dans le guide de l'utilisateur Amazon EKS.

- 22. (Facultatif) Pour les groupes de sécurité, choisissez un groupe de sécurité à associer à vos instances. Par défaut, le groupe de sécurité par défaut de votre VPC est sélectionné.
- 23. Choisissez Page suivante.
- 24. Pour la révision, passez en revue les étapes de configuration. Si vous devez apporter des modifications, choisissez Modifier. Lorsque vous avez terminé, choisissez Create computing environment.

Tutoriel : Création d'un environnement informatique non géré à l'aide des ressources Amazon EKS

Procédez comme suit pour créer un environnement de calcul non géré à l'aide des ressources Amazon Elastic Kubernetes Service (Amazon EKS).

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation en haut de la page, sélectionnez le Région AWS à utiliser.
3. Dans le panneau de navigation, choisissez Environnements de calcul.
4. Choisissez Créer.
5. Configurez l'environnement.
 - a. Pour la configuration de l'environnement informatique, choisissez Amazon Elastic Kubernetes Service (Amazon EKS).
 - b. Pour le type d'orchestration, choisissez Non géré.
6. Dans Nom, spécifiez un nom unique pour votre environnement informatique. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
7. Pour le cluster EKS, choisissez un cluster Amazon EKS existant. Pour créer un nouveau cluster EKS, suivez les étapes de la [page Créer un cluster Amazon EKS](#).
8. Pour Namespace, entrez un espace de Kubernetes noms pour regrouper vos AWS Batch processus dans le cluster.
9. (Facultatif) Pour Maximum v CPUs, spécifiez le nombre maximum de v CPUs disponibles pour la planification des tâches à partir de votre capacité allouée.
10. (Facultatif) Développez les balises. Choisissez Ajouter une balise, puis entrez une paire clé-valeur.
11. Choisissez Page suivante.
12. Pour la révision, passez en revue les étapes de configuration. Si vous devez apporter des modifications, choisissez Modifier. Lorsque vous avez terminé, choisissez Créer un environnement de calcul.

Affectation de nœuds de cluster Amazon EKS à un environnement informatique non géré

Après avoir créé l'environnement de calcul non géré, vous devez étiqueter vos nœuds Amazon EKS avec l'UUID de l'environnement de calcul.

Tout d'abord, obtenez l'UUID de l'environnement de calcul à partir du résultat de l'`DescribeComputeEnvironmentsAPI` :

```
$ aws batch describe-compute-environments \
  --compute-environments unmanagedEksCE \
  --query "computeEnvironments[].{name: computeEnvironmentName, uuid: uuid}"
```

Obtenez les informations sur le nœud :

```
kubectl get nodes -o name
```

Étiquetez les nœuds avec l' AWS Batch UUID de l'environnement de calcul :

```
kubectl label <node-name> batch.amazonaws.com/compute-environment-uuid=uuid
```

Ressource : modèle d'environnement de calcul

L'exemple suivant montre un modèle d'environnement de calcul vide. Vous pouvez utiliser ce modèle pour créer votre environnement informatique qui peut ensuite être enregistré dans un fichier et utilisé avec l' AWS CLI `--cli-input-json` option. Pour plus d'informations sur ces paramètres, consultez [CreateComputeEnvironment](#) la référence de l'AWS Batch API.

Note

Vous pouvez générer un modèle d'environnement de calcul à l'aide de la AWS CLI commande suivante.

```
$ aws batch create-compute-environment --generate-cli-skeleton
```

```
{
  "computeEnvironmentName": "",
```

```
"type": "UNMANAGED",
"state": "DISABLED",
"unmanagedvCpus": 0,
"computeResources": {
  "type": "EC2",
  "allocationStrategy": "BEST_FIT_PROGRESSIVE",
  "minvCpus": 0,
  "maxvCpus": 0,
  "desiredvCpus": 0,
  "instanceTypes": [
    ""
  ],
  "imageId": "",
  "subnets": [
    ""
  ],
  "securityGroupIds": [
    ""
  ],
  "ec2KeyPair": "",
  "instanceRole": "",
  "tags": {
    "KeyName": ""
  },
  "placementGroup": "",
  "bidPercentage": 0,
  "spotIamFleetRole": "",
  "launchTemplate": {
    "launchTemplateId": "",
    "launchTemplateName": "",
    "version": ""
  },
  "ec2Configuration": [
    {
      "imageType": "",
      "imageIdOverride": "",
      "imageKubernetesVersion": ""
    }
  ]
},
"serviceRole": "",
"tags": {
  "KeyName": ""
},
```



```

    "eksConfiguration": {
      "eksClusterArn": "",
      "kubernetesNamespace": ""
    }
  }
}

```

Table de calcul du type d'instance

Le tableau suivant répertorie le Région AWS mot clé de famille d'instances et les familles d'instances disponibles. AWS Batch essaiera d'allouer une instance de la dernière famille, mais comme la disponibilité de la famille d'instances varie en fonction des besoins, il est possible Région AWS que vous obteniez une génération de famille d'instances antérieure.

default_x86_64

Région	Familles d'instances
Tout Région AWS est ce soutien AWS Batch	6i, 6i, 6i c7i

default_arm64

Région	Familles d'instances
Tout Région AWS est ce soutien AWS Batch	6 mg, 6 g, 6 g c7g

Optimal

Région	Familles d'instances
- ap-northeast-1 - ap-northeast-2 - ap-south-1 - ap-southeast-1 - ap-southeast-2	m4, c4, r4

Région	Familles d'instances
- ca-central-1 - cn-north-1 - cn-northwest-1 - eu-central-1 - eu-west-1 - eu-west-2 - sa-east-1 - us-east-1 - us-east-2 - us-gov-west-1 - us-west-1 - us-west-2	

Région	Familles d'instances
• af-south-1 • ap-east-1 • ap-northeast-3 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ca-west-1 • eu-central-2 • eu-north-1 • eu-south-1 • eu-south-2 • eu-west-3 • il-central-1 • me-central-1 • me-south-1 • us-gov-east-1 • us-isob-east-1 • us-iso-east-1 • us-isof-south-1 • us-isof-east-1 • eu-isoe-west-1 • us-northeast-1	m5, c5, r5
• ap-southeast-5 • ap-southeast-7 • ap-est-2 • mx-central-1	m6, c6, r6

Mettre à jour un environnement informatique dans AWS Batch

AWS Batch propose plusieurs stratégies de mise à jour des environnements informatiques, chacune étant conçue pour des scénarios et des exigences de mise à jour spécifiques. Ces approches utilisent la même API de mise à jour sous-jacente mais représentent des méthodes prescriptives différentes pour gérer efficacement les mises à jour. Vous pouvez gérer ces mises à jour à l'aide de la AWS Batch console ou du AWS CLI. La compréhension de ces stratégies vous aide à choisir la méthode la mieux adaptée à vos besoins tout en minimisant les perturbations de vos charges de travail.

Cette rubrique fournit une vue d'ensemble des stratégies de mise à jour disponibles et des conseils sur le moment d'utiliser chaque approche. Pour les procédures détaillées, consultez les sections individuelles de chaque stratégie de mise à jour.

Important

AWS Batch crée et gère plusieurs AWS ressources en votre nom et au sein de votre compte, notamment les modèles Amazon EC2 Launch, les groupes Amazon EC2 Auto Scaling, les flottes Amazon EC2 Spot et les clusters Amazon ECS. Ces ressources gérées sont configurées spécifiquement pour garantir un AWS Batch fonctionnement optimal. La modification manuelle de ces ressources AWS Batch gérées, sauf indication contraire dans la AWS Batch documentation, peut entraîner des comportements inattendus, notamment des environnements INVALID informatiques, un comportement de dimensionnement des instances sous-optimal, des retards dans le traitement des charges de travail ou des coûts imprévus. Ces modifications manuelles ne peuvent pas être prises en charge de manière déterministe par le AWS Batch service. Utilisez toujours le support AWS Batch APIs ou la AWS Batch console pour gérer vos environnements informatiques.

Rubriques

- [Stratégies de mise à jour des environnements informatiques](#)
- [Choisir la bonne stratégie de mise à jour](#)
- [Effectuer des mises à jour de dimensionnement](#)
- [Effectuer des mises à jour d'infrastructure](#)
- [Effectuer des blue/green mises à jour pour les environnements informatiques](#)

Stratégies de mise à jour des environnements informatiques

Lorsque vous utilisez le dimensionnement ou les mises à jour de l'infrastructure, votre environnement informatique est mis à jour sur place. Pour la stratégie de blue/green mise à jour, vous créez un nouvel environnement informatique (vert), puis vous migrez votre charge de travail de l'ancien environnement informatique (bleu) vers le nouvel environnement informatique (vert).

AWS Batch propose trois stratégies différentes pour les mises à jour de l'environnement de calcul :

Mises à jour relatives

Les mises à jour de dimensionnement ajustent la capacité de votre environnement informatique en ajoutant ou en supprimant des instances sans remplacer les instances existantes. Il s'agit du scénario de mise à jour le plus rapide et ne nécessite aucun temps d'arrêt. Utilisez les mises à jour de dimensionnement lorsque vous devez modifier les paramètres de capacité (vCPUs). Ces mises à jour sont généralement effectuées en quelques minutes.

Les mises à jour Fargate sont effectuées selon les mêmes procédures que les mises à jour de dimensionnement. Pour de plus amples informations, veuillez consulter [Effectuer des mises à jour de dimensionnement](#).

Mises à jour de

Les mises à jour de l'infrastructure remplacent les instances de votre environnement informatique par de nouvelles instances dont les paramètres ont été mis à jour. Ces mises à jour nécessitent des configurations de rôles de service et de stratégie d'allocation spécifiques, mais elles réduisent au minimum les temps d'arrêt, les tâches en cours pouvant être interrompues. Utilisez les mises à jour de l'infrastructure lorsque vous devez modifier les types d'instances, la configuration de l'AMI, les paramètres réseau, le rôle de service, l'état de l'environnement ou d'autres composants de l'infrastructure. Ces mises à jour sont généralement effectuées en 10 à 30 minutes en fonction de l'achèvement de la tâche.

Pour de plus amples informations, veuillez consulter [Effectuer des mises à jour d'infrastructure](#).

Mises à jour bleu/vert

Blue/green updates create a new compute environment alongside your existing environment, allowing gradual workload transition with zero downtime. This approach provides the safest update path but requires running two environments temporarily. Use blue/green mises à jour lorsque vous n'avez besoin d'aucune interruption de service, que vous souhaitez tester les

modifications avant le déploiement complet, que vous avez besoin d'une fonctionnalité de restauration rapide ou que vous utilisez des configurations non prises en charge pour les mises à jour de l'infrastructure. Le temps nécessaire pour terminer est variable et vous pouvez le contrôler.

Pour de plus amples informations, veuillez consulter [Effectuer des blue/green mises à jour pour les environnements informatiques](#).

Choisir la bonne stratégie de mise à jour

Utilisez ce guide de décision pour sélectionner la stratégie de mise à jour la mieux adaptée à vos besoins :

Choisissez de dimensionner les mises à jour lorsque

Choisissez la stratégie de mise à jour du dimensionnement lorsque vous devez uniquement ajuster la capacité de calcul (vCPUs). Les mises à jour évolutives sont idéales lorsque vous avez besoin de mises à jour rapides, sans interruption ni modification de la configuration de l'infrastructure.

Pour connaître les procédures détaillées, consultez [Effectuer des mises à jour de dimensionnement](#).

Choisissez les mises à jour de l'infrastructure quand

Choisissez la stratégie de mise à jour de l'infrastructure lorsque vous devez modifier les types d'instances, les paramètres de l'AMI, le rôle du service, l'état de l'environnement ou la configuration réseau. Votre environnement doit utiliser le rôle `AWSServiceRoleForBatch` au service et une stratégie d'allocation de `BEST_FIT_PROGRESSIVESPOT_CAPACITY_OPTIMIZED`, ou `SPOT_PRICE_CAPACITY_OPTIMIZED`. Les mises à jour de l'infrastructure fonctionnent bien lorsqu'une interruption de travail est acceptable pendant la mise à jour et que vous souhaitez des mises à jour automatiques de la dernière AMI optimisée pour Amazon ECS.

Pour connaître les procédures détaillées, consultez [Effectuer des mises à jour d'infrastructure](#).

Choisissez les blue/green mises à jour lorsque

Choisissez la stratégie de blue/green mise à jour lorsque aucune interruption de service n'est requise pour vos charges de travail ou lorsque vous devez tester les modifications avant de transférer les charges de travail de production. Cette approche est essentielle lorsque la capacité de restauration rapide est importante, que votre environnement utilise une stratégie `BEST_FIT` d'allocation ou que

votre environnement n'utilise pas le rôle lié au `AWSServiceRoleForBatch` service. Blue/green les mises à jour sont également le meilleur choix lorsque vous utilisez des fonctionnalités personnalisées AMIs qui nécessitent des mises à jour manuelles ou des modifications de configuration majeures.

Pour connaître les procédures détaillées, consultez [Effectuer des blue/green mises à jour pour les environnements informatiques](#).

Considérations relatives à la mise à jour de

AWS Batch peut être mis à jour vers la dernière AMI optimisée pour Amazon ECS lors des mises à jour de [l'infrastructure](#) lorsque toutes les conditions suivantes sont remplies :

Note

Une fois la mise à jour de l'infrastructure terminée, elle `updateToLatestImageVersion` est définie sur `false`. Pour lancer une autre mise à jour `updateToLatestImageVersion`, il doit être réglé sur `true`.

- L'environnement de calcul utilise le rôle `AWSServiceRoleForBatch` lié au service
- La stratégie d'allocation est définie pour `BEST_FIT_PROGRESSIVE_SPOT_CAPACITY_OPTIMIZED`, ou `SPOT_PRICE_CAPACITY_OPTIMIZED`
- Aucun ID d'AMI n'est explicitement spécifié dans `imageId` `imageIdOverride` le modèle de lancement
- Le `updateToLatestImageVersion` est réglé sur `true`

Mises à jour des AMI à l'aide blue/green du déploiement

Vous devez utiliser blue/green le déploiement pour effectuer la mise à jour AMIs dans les scénarios suivants :

- Lors de l'utilisation d'une version spécifique de l'AMI optimisée pour Amazon ECS
- Lorsque l'ID de l'AMI est spécifié dans l'un des formats suivants :
 - Lancer le modèle (vous devez le mettre à jour ou le supprimer)
 - Paramètre `imageId`
 - Le `imageIdOverride` paramètre dans la EC2 configuration

- Lors de l'utilisation de la stratégie BEST_FIT d'allocation (ne prend pas en charge les mises à jour de l'infrastructure)
- Lorsque vous n'utilisez pas le rôle lié à AWSServiceRoleForBatch [un service](#)

Effectuer des mises à jour de dimensionnement

Les mises à jour de dimensionnement ajustent la capacité de votre environnement informatique en ajoutant ou en supprimant des instances. Il s'agit de la stratégie de mise à jour la plus rapide et elle ne nécessite pas le remplacement des instances existantes. Les mises à jour évolutives fonctionnent avec tous les types de rôles de service et toutes les stratégies d'allocation, ce qui en fait l'option de mise à jour la plus flexible.

Modifications qui déclenchent une mise à jour du dimensionnement

Lorsque vous modifiez uniquement les paramètres suivants, AWS Batch effectue une mise à jour du dimensionnement. Si vous modifiez l'un de ces paramètres ainsi que d'autres paramètres de l'environnement informatique, effectuez plutôt AWS Batch une [mise à jour de l'infrastructure](#).

Les paramètres suivants déclenchent des mises à jour de dimensionnement lorsqu'ils sont modifiés exclusivement :

- `desiredvCpus`— Définit le nombre cible de v CPUs pour l'environnement.
- `maxvCpus`— Définit le nombre maximum de v CPUs pouvant être lancés.
- `minvCpus`— Spécifie le nombre minimum de v CPUs à maintenir.

Pour les environnements informatiques Fargate, vous pouvez également modifier ces paramètres pour dimensionner les mises à jour :

- `securityGroupIds`— Groupe de sécurité IDs pour l'environnement informatique.
- `subnets`— Sous-réseaux pour l'environnement informatique.

Note

Nous vous recommandons de ne pas l'utiliser `desiredvCpus` pour lancer une mise à jour du dimensionnement car AWS Batch cela s'ajustera dynamiquement `desiredvCpus`. Vous devriez plutôt effectuer une mise à jour `minvCpus`.

Lors de la mise à jour `desiredvCpus`, la valeur doit être comprise entre `minvCpus` et `maxvCpus`. La nouvelle valeur doit être supérieure ou égale à la valeur actuelle `desiredvCpus`. Pour de plus amples informations, veuillez consulter [the section called "Message d'erreur lorsque vous mettez à jour le `desiredvCpus` paramètre"](#).

Important

Si vous modifiez l'un de ces paramètres de dimensionnement conjointement avec d'autres paramètres d'environnement de calcul (tels que les types d'instances, les AMI IDs ou les modèles de lancement), AWS Batch effectue une mise à jour de l'infrastructure plutôt qu'une mise à jour de dimensionnement. Les mises à jour de l'infrastructure prennent plus de temps et peuvent remplacer les instances existantes.

Performing scaling updates using the AWS Management Console

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans le volet de navigation, choisissez Environments, puis l'onglet Compute environments.
3. Sélectionnez l'environnement informatique à mettre à jour.
4. Choisissez Actions, puis Modifier.
5. Modifiez un ou plusieurs [paramètres qui prennent en charge les mises à jour de dimensionnement](#). Exemples :
 - Pour Minimum v CPUs, entrez le nombre minimum de CPUs v.
 - Pour Desired v CPUs, entrez le nombre souhaité de CPUs v.
 - Pour Maximum v CPUs, entrez le nombre maximum de CPUs v.
6. Sélectionnez Enregistrer les modifications.
7. Surveillez l'état de l'environnement informatique. La mise à jour devrait se terminer rapidement car elle ne concerne que des opérations de dimensionnement.

Performing scaling updates using the AWS CLI

Utilisez la `update-compute-environment` commande pour effectuer des mises à jour de dimensionnement. Les deux exemples suivants illustrent les opérations de dimensionnement

courantes. Vous pouvez modifier un ou plusieurs des [paramètres suivants qui prennent en charge les mises à jour de dimensionnement](#) :

- Cet exemple met à jour les valeurs souhaitées, minimales et maximales CPUs :

```
aws batch update-compute-environment \  
  --compute-environment your-compute-environment-name \  
  --compute-resources minvCpus=2,maxvCpus=8
```

Surveillance des mises à jour du dimensionnement

Surveillez vos mises à jour de dimensionnement à l'aide de la AWS Batch console pour consulter l'état de l'environnement de calcul et vérifier le nombre d'instances et les métriques de vCPU. Vous pouvez également utiliser la `describe-compute-environments` commande AWS CLI with the pour vérifier l'état et surveiller le nombre d'instances et les valeurs des vCPU.

Effectuer des mises à jour d'infrastructure

Les mises à jour de l'infrastructure remplacent les instances de votre environnement informatique par de nouvelles instances dont les paramètres ont été mis à jour. Cette stratégie de mise à jour prend plus de temps que le dimensionnement des mises à jour et nécessite des paramètres de rôle de service et de stratégie d'allocation spécifiques. Les mises à jour de l'infrastructure permettent de modifier les configurations fondamentales de l'environnement informatique tout en maintenant la disponibilité des services.

Important

Les mises à jour de l'infrastructure nécessitent le rôle `AWSServiceRoleForBatch` lié au service et une stratégie d'allocation de `BEST_FIT_PROGRESSIVESPOT_CAPACITY_OPTIMIZED`, ou `SPOT_PRICE_CAPACITY_OPTIMIZED`. Si votre environnement ne répond pas à ces exigences, utilisez plutôt des blue/green mises à jour.

Changements déclenchant des mises à jour de l'infrastructure

Lorsque vous modifiez l'un des paramètres suivants, AWS Batch effectue une mise à jour de l'infrastructure. Les mises à jour de l'infrastructure se produisent également lorsque vous modifiez ces paramètres en même temps que les paramètres de mise à jour du dimensionnement.

Les paramètres suivants déclenchent les mises à jour de l'infrastructure :

Configuration de calcul

- `allocationStrategy`— Détermine le mode de AWS Batch sélection des types d'instances.
- `instanceTypes`— Spécifie les types d' EC2 instances à utiliser.
- `bidPercentage`— Pourcentage maximum du prix à la demande pour les instances Spot.
- `type`— Type d'environnement de calcul (EC2 ou SPOT).

AMI et configuration de lancement

- `imageId`— AMI spécifique à utiliser pour les instances.
- `ec2Configuration`— EC2 configuration, y compris `imageIdOverride`.
- `launchTemplate`— paramètres EC2 du modèle de lancement.
- `ec2KeyPair`— Paire de clés SSH pour l'accès à l'instance.
- `updateToLatestImageVersion`— Paramètre de mise à jour automatique de l'AMI.

Mise en réseau et sécurité

- `subnets`— Sous-réseaux VPC où les instances sont lancées (pour les environnements de EC2 calcul).
- `securityGroupIds`— Groupes de sécurité pour les instances (pour les environnements EC2 informatiques).
- `placementGroup`— configuration du groupe de EC2 placement.

Autres paramètres

- `instanceRole`— Rôle IAM pour les EC2 instances.
- `tags`— Tags appliqués aux EC2 instances.

Important

Si vous modifiez des paramètres de mise à jour de l'infrastructure en même temps que des paramètres de mise à jour de dimensionnement (tels que `desiredvCpus` et `maxvCpus`,

ou `minvCpus`), AWS Batch effectue une mise à jour de l'infrastructure. Les mises à jour de l'infrastructure prennent plus de temps que les mises à niveau.

Sélection des AMI lors des mises à jour de l'infrastructure

Lors d'une mise à jour de l'infrastructure, l'ID AMI de l'environnement de calcul peut changer, selon qu'il AMIs est spécifié ou non dans l'un de ces trois paramètres. AMIs sont spécifiés dans le `imageId` (`incomputeResources`), `imageIdOverride` (`in ec2Configuration`) ou le modèle de lancement spécifié dans `launchTemplate`. Supposons qu'aucune AMI IDs ne soit spécifiée dans aucun de ces paramètres et que le `updateToLatestImageVersion` paramètre soit `true`. Ensuite, la dernière AMI optimisée Amazon ECS prise en charge par AWS Batch est utilisée pour toute mise à jour de l'infrastructure.

Si un ID d'AMI est spécifié dans au moins l'un de ces paramètres, la mise à jour dépend du paramètre qui a fourni l'ID d'AMI utilisé avant la mise à jour. Lorsque vous créez un environnement informatique, la priorité pour sélectionner un ID d'AMI est d'abord le modèle de lancement, puis le `imageId` paramètre, et enfin le `imageIdOverride` paramètre. Toutefois, si l'ID d'AMI utilisé provient du modèle de lancement, la mise à jour des `imageIdOverride` paramètres `imageId` ou des paramètres ne met pas à jour l'ID d'AMI. La seule façon de mettre à jour un ID d'AMI sélectionné dans le modèle de lancement est de mettre à jour le modèle de lancement. Si le paramètre de version du modèle de lancement est `$Default` ou `$Latest`, la version par défaut ou la dernière version du modèle de lancement spécifié est évaluée. Si un autre ID d'AMI est sélectionné par défaut ou si la dernière version du modèle de lancement est sélectionnée, cet ID d'AMI est utilisé dans la mise à jour.

Si le modèle de lancement n'a pas été utilisé pour sélectionner l'ID d'AMI, l'ID d'AMI spécifié dans les `imageIdOverride` paramètres `imageId` ou est utilisé. Si les deux sont spécifiés, l'ID d'AMI spécifié dans le `imageIdOverride` paramètre est utilisé.

Supposons que l'environnement de calcul utilise un ID d'AMI spécifié par les `launchTemplate` paramètres `imageId` `imageIdOverride`, ou, et que vous souhaitiez utiliser la dernière AMI optimisée pour Amazon ECS prise en charge par AWS Batch. Ensuite, la mise à jour doit supprimer les paramètres qui ont fourni l'AMI IDs. En `imageId` effet, cela nécessite de spécifier une chaîne vide pour ce paramètre. En `imageIdOverride` effet, cela nécessite de spécifier une chaîne vide pour le `ec2Configuration` paramètre.

Si l'ID d'AMI provient du modèle de lancement, vous pouvez passer à la dernière AMI optimisée pour Amazon ECS prise AWS Batch en charge de l'une des manières suivantes :

- Supprimez le modèle de lancement en spécifiant une chaîne vide pour le `launchTemplateName` paramètre `launchTemplateId` or. Cela supprime l'intégralité du modèle de lancement, plutôt que le seul ID de l'AMI.
- Si la version mise à jour du modèle de lancement ne spécifie pas d'ID d'AMI, le `updateToLatestImageVersion` paramètre doit être défini sur `true`.

Gestion des tâches pendant les mises à jour

Configurez le mode de gestion des tâches en cours lors d'une mise à jour de l'infrastructure à l'aide de la politique de mise à jour. Lorsque vous définissez `terminateJobsOnUpdate=true`, les tâches en cours d'exécution sont immédiatement interrompues, le `jobExecutionTimeoutMinutes` paramètre est ignoré et la mise à jour se poursuit dès que les instances peuvent être remplacées. Lorsque vous le définissez `terminateJobsOnUpdate=false`, les tâches en cours d'exécution se poursuivent pendant la période spécifiée avec un délai d'expiration par défaut de 30 minutes, et les tâches sont interrompues si elles dépassent le délai d'expiration.

Note

Pour réessayer des tâches interrompues lors d'une mise à jour, configurez une stratégie de nouvelle tentative de tâche. Pour de plus amples informations, veuillez consulter [the section called "Nouvelles tentatives de travail automatisées"](#).

Performing infrastructure updates using the AWS Management Console

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans le volet de navigation, choisissez Environments puis l'onglet Compute environments.
3. Sélectionnez l'environnement informatique à mettre à jour.
4. Choisissez Actions, puis Modifier.
5. Dans la section Comportement des mises à jour, configurez le mode de gestion des tâches en cours d'exécution :
 - Choisissez Mettre à jour l'AMI vers la dernière version pour mettre à jour l'AMI vers la dernière version.
 - Choisissez Terminer les tâches immédiatement après la mise à jour pour terminer les tâches lorsque le processus de mise à jour est exécuté.

- Pour le délai d'exécution du Job, entrez le nombre de minutes à attendre avant de démarrer le processus de mise à jour.
6. Modifiez un ou plusieurs [paramètres qui nécessitent une mise à jour de l'infrastructure](#).
Exemples :
- Rôle de l'instance
 - Utiliser des instances EC2 Spot
 - Types d'instances autorisés
 - Groupe de placement
 - EC2 paire de clés
 - EC2 configuration
 - Modèles de lancement
 - Sous-réseaux
 - Groupes de sécurité
7. Sélectionnez Enregistrer les modifications.
8. Surveillez l'état de l'environnement informatique. L'environnement s'affichera UPDATING pendant le processus de mise à jour.

Performing infrastructure updates using the AWS CLI

Utilisez la `update-compute-environment` commande pour modifier un ou plusieurs [paramètres nécessitant une mise à jour de l'infrastructure](#). Les trois exemples suivants sont des opérations d'infrastructure courantes.

- Cet exemple met à jour les types d'instances et configure la politique de mise à jour :

```
aws batch update-compute-environment \  
  --compute-environment your-compute-environment-name \  
  --compute-resources instanceTypes=default_x86_64 \  
  --update-policy terminateJobsOnUpdate=false,jobExecutionTimeoutMinutes=30
```

- Cet exemple met à jour les sous-réseaux et les groupes de sécurité VPC :

```
aws batch update-compute-environment \  
  --compute-environment your-compute-environment-name \  
  --compute-resources subnets=subnet-abcd1234,subnet-efgh5678 \  
  securityGroupIds=sg-abcd1234 \  
  --update-policy terminateJobsOnUpdate=false,jobExecutionTimeoutMinutes=30
```

```
--update-policy terminateJobsOnUpdate=true
```

- Cet exemple active les mises à jour automatiques de la dernière AMI optimisée pour Amazon ECS :

```
aws batch update-compute-environment \  
  --compute-environment your-compute-environment-name \  
  --compute-resources updateToLatestImageVersion=true \  
  --update-policy terminateJobsOnUpdate=false,jobExecutionTimeoutMinutes=60
```

Mises à jour de l'infrastructure

Surveillez les mises à jour de votre infrastructure à l'aide de la AWS Batch console pour suivre l'évolution de l'état de l'environnement informatique UPDATING, suivre la progression du remplacement des instances et vérifier l'absence d'échec des mises à jour. La mise à jour est réussie une fois que l'état de l'environnement de calcul est atteint VAILD. Vous pouvez également l'utiliser CloudWatch pour suivre les événements de cessation d'instance et surveiller l'état des tâches pendant la mise à jour. À l'aide de AWS CLI, utilisez la `describe-compute-environments` commande pour vérifier l'état et surveiller les événements du cycle de vie de l'instance.

Effectuer des blue/green mises à jour pour les environnements informatiques

Une blue/green mise à jour est une stratégie de mise à jour qui réduit les temps d'arrêt et les risques en créant un nouvel environnement informatique (vert) en plus de votre environnement informatique existant (bleu). Cette approche vous permet de transférer progressivement les charges de travail vers le nouvel environnement tout en maintenant le fonctionnement de l'environnement existant. Blue/green les mises à jour fournissent le chemin de mise à jour le plus sûr et fonctionnent avec tous les types de rôles de service ou de stratégie d'allocation.

Présentation

Les mises à jour bleu/vert offrent plusieurs avantages qui les rendent idéales pour les environnements de production. Ils n'offrent aucun temps d'arrêt en assurant le fonctionnement continu de vos charges de travail pendant le processus de mise à jour. Cette approche permet des fonctionnalités de restauration faciles, vous permettant de revenir rapidement à l'environnement d'origine en cas de problème. Vous pouvez mettre en œuvre une stratégie de transition progressive, en vérifiant les performances du nouvel environnement avant de transférer complètement vos

charges de travail de production. Cette méthode permet également une excellente atténuation des risques puisque l'environnement d'origine reste inchangé et opérationnel jusqu'à ce que vous choisissiez de le supprimer.

Quand des blue/green mises à jour sont nécessaires

Vous devez utiliser les blue/green mises à jour dans les situations suivantes :

- Lorsque votre environnement informatique utilise une stratégie BEST_FIT d'allocation (ne prend pas en charge les mises à jour de l'infrastructure)
- Lorsque votre environnement informatique n'utilise pas le rôle lié à `AWSServiceRoleForBatch` service
- Quand vous devez passer d'un type de rôle de service à un autre

Quand des blue/green mises à jour sont recommandées

Blue/green updates are particularly recommended for production environments where zero downtime is critical for your workloads. This approach works well when you need to test new configurations before transitioning production workloads, ensuring that changes meet your performance and reliability requirements. Choose blue/green mises à jour lorsque la fonctionnalité de restauration rapide est importante pour vos opérations, en particulier si vous effectuez une mise à jour personnalisée AMIs avec des modifications importantes. Cette méthode est également idéale lorsque vous souhaitez valider les caractéristiques de performance et le comportement avant de procéder entièrement aux modifications, afin de garantir la confiance dans votre processus de mise à jour.

Prérequis

Avant d'effectuer une blue/green mise à jour, assurez-vous d'avoir :

- [Autorisations IAM](#) appropriées pour créer et gérer des environnements informatiques
- Accès pour afficher et modifier les paramètres de la file d'attente des tâches
- Stratégies de nouvelle tentative de tâche configurées pour vos définitions de tâches afin de gérer les échecs potentiels pendant la transition. Pour de plus amples informations, veuillez consulter [Nouvelles tentatives de travail automatisées](#).
- L'ID AMI du nouvel environnement informatique. Cela peut être soit :
 - Une version récente et approuvée de l'AMI optimisée pour Amazon ECS (utilisée par défaut)

- Une AMI personnalisée qui répond aux spécifications de l'AMI d'instance de conteneur Amazon ECS. Lorsque vous utilisez une AMI personnalisée, vous pouvez la spécifier de l'une des manières suivantes :
 - Utilisation du champ Image ID override dans la configuration EC2
 - Le spécifier dans un modèle de lancement

Pour plus d'informations sur la création de AMIs personnalisations, consultez [Tutoriel : Création d'une AMI de ressources de calcul](#).

Avant de créer le nouvel environnement, vous devez enregistrer la configuration de votre environnement informatique existant. Pour ce faire, vous pouvez utiliser le AWS Management Console ou le AWS CLI.

Note

Les procédures suivantes expliquent comment effectuer une blue/green mise à jour qui ne modifie que l'AMI. Vous pouvez mettre à jour d'autres paramètres pour le nouvel environnement.

Important

Lorsque vous supprimez l'ancien environnement informatique (bleu), toutes les tâches en cours d'exécution sur ces instances échoueront car les instances seront arrêtées. Configurez des stratégies de nouvelle tentative dans vos définitions de tâches afin de gérer automatiquement ces échecs. Pour de plus amples informations, veuillez consulter [Nouvelles tentatives de travail automatisées](#).

Une fois que vous aurez confiance dans le nouvel environnement :

1. Modifiez la file d'attente des tâches pour supprimer l'ancien environnement informatique.
2. Attendez que toutes les tâches en cours dans l'ancien environnement soient terminées.
3. Supprimez l'ancien environnement de calcul.

Performing blue/green updates using the AWS Management Console

1. Clonez votre environnement informatique actuel

- a. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
 - b. Sélectionnez votre environnement informatique existant.
 - c. Choisissez Actions, puis Cloner.
 - d. Dans Nom, entrez un nom unique pour votre nouvel environnement informatique.
 - e. Choisissez Suivant.
 - f. Dans la section Configuration de l'instance, mettez à jour les paramètres de l'AMI :
 - i. Développez Additional configuration (Configuration supplémentaire).
 - ii. Pour la EC2 configuration, spécifiez le nouveau type d'AMI dans Type d'image et l'ID d'AMI dans le champ Image ID override.
 - g. Choisissez Suivant.
 - h. Pour la configuration du réseau, choisissez Next.
 - i. Passez en revue les autres paramètres qui sont automatiquement copiés à partir de votre environnement existant.
 - j. Choisissez Créer un environnement de calcul.
 - k. Attendez que le nouvel état de l'environnement informatique soit atteint `VALID`.
2. Modifier l'ordre de la file d'attente des tâches
 - a. Dans le volet de navigation, choisissez Job queues.
 - b. Sélectionnez la file d'attente de tâches associée à votre environnement informatique existant.
 - c. Choisissez Modifier.
 - d. Sous Environnement de calcul connecté, ajoutez le nouvel environnement de calcul :
 - Ajoutez le nouvel environnement informatique avec un numéro de commande supérieur à celui de l'environnement existant pour transférer la charge de travail.
 - Une fois que vous avez vérifié que le nouvel environnement fonctionne correctement, vous pouvez en faire l'environnement principal en lui attribuant un numéro de commande inférieur.
 - e. Choisissez Mettre à jour la file d'attente des tâches.
 3. Nettoyage
 - a. Surveillez l'exécution des tâches dans le nouvel environnement pour vous assurer que tout fonctionne comme prévu.

- b. Une fois que vous aurez confiance dans le nouvel environnement :
 1. Modifiez la file d'attente des tâches pour supprimer l'ancien environnement informatique.
 2. Attendez que toutes les tâches en cours dans l'ancien environnement soient terminées.
 3. Supprimez l'ancien environnement de calcul.

Performing blue/green updates using the AWS CLI

1. Pour obtenir la configuration à l'aide de AWS CLI, utilisez la commande suivante :

```
aws batch describe-compute-environments \  
  --compute-environments your-compute-environment-name
```

Enregistrez la sortie pour référence lors de la création du nouvel environnement.

2. Créez un nouvel environnement informatique en utilisant la configuration de votre environnement existant, mais avec la nouvelle AMI. Voici un exemple de structure de commande :

Remplacez les valeurs d'exemple par votre configuration réelle de l'étape précédente :

```
cat <<EOF > ./blue-green-compute-environment.json  
{  
  "computeEnvironmentName": "your-new-compute-environment-name",  
  "type": "MANAGED",  
  "state": "ENABLED",  
  "computeResources": {  
    "instanceRole": "arn:aws:iam::012345678901:instance-profile/  
ecsInstanceRole",  
    "type": "EC2",  
    "minvCpus": 2,  
    "desiredvCpus": 2,  
    "maxvCpus": 256,  
    "instanceTypes": [  
      "optimal"  
    ],  
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",  
    "ec2Configuration": [  
      {  

```

```
    "imageType": "ECS_AL2023",
    "imageIdOverride": "ami-0abcdef1234567890"
  },
],
"subnets": [
  "subnet-0abcdef1234567890"
],
"securityGroupIds": [
  "sg-0abcdef1234567890"
]
}
}
EOF
```

```
$ aws batch create-compute-environment --cli-input-json file:///./blue-green-
compute-environment.json
```

3. Attendez que le nouvel environnement soit disponible :

```
aws batch describe-compute-environments \
  --compute-environments your-new-compute-environment-name \
  --query 'computeEnvironments[].status'
```

4. Ajoutez le nouvel environnement informatique à votre file d'attente de tâches :

```
aws batch update-job-queue \
  --job-queue your-job-queue \
  --compute-environment-order order=1,computeEnvironment=your-existing-
environment \
  order=2,computeEnvironment=your-new-compute-environment-name
```

5. Une fois vérifié, effectuez une nouvelle mise à jour pour rendre le nouvel environnement principal :

```
aws batch update-job-queue \
  --job-queue your-job-queue \
  --compute-environment-order order=1,computeEnvironment=your-new-compute-
environment-name
```

Une fois toutes les tâches terminées dans l'ancien environnement, désactivez-le puis supprimez-le :

```
aws batch update-compute-environment \  
  --compute-environment your-existing-environment \  
  --state DISABLED
```

```
aws batch delete-compute-environment \  
  --compute-environment your-existing-environment
```

Ressource de calcul AMIs

Par défaut, les environnements informatiques AWS Batch gérés utilisent une version récente et approuvée de l'AMI optimisée Amazon ECS pour les ressources de calcul. Toutefois, vous souhaitez peut-être créer votre propre AMI à utiliser pour vos environnements informatiques gérés et non gérés. Si vous avez besoin de l'un des éléments suivants, nous vous recommandons de créer votre propre AMI :

- Augmenter la taille de stockage de la racine ou des volumes de données de votre AMI
- Ajout de volumes de stockage d'instance pour les types d' EC2 instances Amazon pris en charge
- Personnalisation de l'agent de conteneur Amazon ECS
- Personnalisation de Docker
- Configuration d'une AMI de charge de travail GPU pour permettre aux conteneurs d'accéder au matériel GPU sur les types d' EC2 instances Amazon pris en charge

Note

Une fois qu'un environnement de calcul est créé, AWS Batch ne le met pas à niveau AMIs dans l'environnement de calcul. AWS Batch ne met pas non plus à jour le AMIs dans votre environnement informatique lorsqu'une version plus récente de l'AMI optimisée Amazon ECS est disponible. Vous êtes responsable de la gestion du système d'exploitation client. Cela inclut les mises à jour et les correctifs de sécurité. Vous êtes également responsable de tout logiciel d'application ou utilitaire supplémentaire que vous installez sur les ressources informatiques. Pour utiliser une nouvelle AMI pour vos AWS Batch tâches, procédez comme suit :

1. Créez un nouvel environnement de calcul avec la nouvelle AMI.

2. Ajoutez l'environnement de calcul à une file d'attente de tâches existante.
3. Supprimez l'environnement de calcul antérieur de votre file d'attente de tâches.
4. Supprimez l'environnement de calcul antérieur.

En avril 2022, AWS Batch ajout d'un support amélioré pour la mise à jour des environnements informatiques. Pour de plus amples informations, veuillez consulter [Mettre à jour un environnement informatique dans AWS Batch](#). Pour utiliser la mise à jour améliorée des environnements informatiques à des fins de mise à jour AMIs, suivez les règles suivantes :

- Ne définissez pas le paramètre du rôle de service ([serviceRole](#)) ou définissez-le sur le rôle `AWSServiceRoleForBatch` lié au service.
- Définissez le paramètre de stratégie d'allocation ([allocationStrategy](#)) sur `BEST_FIT_PROGRESSIVESPOT_CAPACITY_OPTIMIZED`, ou `SPOT_PRICE_CAPACITY_OPTIMIZED`.
- Définissez le paramètre de mise à jour vers la dernière version de l'image ([updateToLatestImageVersion](#)) sur `true`.
- Ne spécifiez pas d'ID d'AMI dans [imageId](#), [imageIdOverride](#) (in [ec2Configuration](#)) ou dans le modèle de lancement ([launchTemplate](#)). Lorsque vous ne spécifiez pas d'ID d'AMI, sélectionnez AWS Batch l'AMI optimisée Amazon ECS la plus récente qui est prise en charge au moment du lancement de la mise à jour de l'infrastructure. Vous pouvez également spécifier l'ID de l'AMI dans les `imageIdOverride` paramètres `imageId` ou. Vous pouvez également spécifier le modèle de lancement identifié par les `LaunchTemplate` propriétés. La modification de l'une de ces propriétés déclenche une mise à jour de l'infrastructure. Si l'ID d'AMI est spécifié dans le modèle de lancement, il n'est pas possible de le remplacer en spécifiant un ID d'AMI dans les `imageIdOverride` paramètres `imageId` ou. L'ID AMI ne peut être remplacé qu'en spécifiant un modèle de lancement différent. Si la version du modèle de lancement est définie sur `$Default` ou `$Latest`, l'ID AMI peut être remplacé en définissant une nouvelle version par défaut pour le modèle de lancement (if `$Default`) ou en ajoutant une nouvelle version au modèle de lancement (if `$Latest`).

Si ces règles sont respectées, toute mise à jour qui lance une mise à jour de l'infrastructure entraîne une nouvelle sélection de l'ID AMI. Si le [version](#) paramètre du modèle de lancement ([launchTemplate](#)) est défini sur `$Latest` ou `$Default`, la version la plus

récente ou par défaut du modèle de lancement est évaluée au moment de la mise à jour de l'infrastructure, même si elle [launchTemplate](#)n'a pas été mise à jour.

Rubriques

- [Spécification d'AMI de ressources de calcul](#)
- [Tutoriel : Création d'une AMI de ressources de calcul](#)
- [Utiliser une AMI de charge de travail du GPU](#)
- [Obsolète d'Amazon Linux](#)
- [Amazon EKS : dépréciation de l'AMI Amazon Linux 2](#)
- [Amazon ECS : dépréciation de l'AMI Amazon Linux 2](#)

Spécification d'AMI de ressources de calcul

La spécification de base de l'AMI pour les ressources de AWS Batch calcul comprend les éléments suivants :

Obligatoire

- Une distribution Linux moderne qui exécute au moins la version 3.10 du noyau Linux sur une AMI de type virtualisation HVM. Les conteneurs Windows ne sont pas pris en charge.

Important

Les tâches parallèles à nœuds multiples ne peuvent être exécutées que sur des ressources de calcul lancées sur une instance Amazon Linux avec le `ecs-init` package installé. Nous vous recommandons d'utiliser l'AMI optimisée par défaut pour Amazon ECS lorsque vous créez votre environnement informatique. Vous pouvez le faire en ne spécifiant pas d'AMI personnalisée. Pour de plus amples informations, veuillez consulter [Tâches parallèles sur plusieurs nœuds](#).

- L'agent de conteneur Amazon ECS. Nous vous recommandons de choisir la dernière version . Pour plus d'informations, consultez la section [Installation de l'agent de conteneur Amazon ECS](#) dans le manuel du développeur Amazon Elastic Container Service.
- Le pilote de `awslogs` journal doit être spécifié en tant que pilote de journal disponible avec la variable d'`ECS_AVAILABLE_LOGGING_DRIVERS`environnement lorsque l'agent de conteneur

Amazon ECS est démarré. Pour plus d'informations, consultez [Configuration de l'agent du conteneur Amazon ECS](#) dans le Manuel du développeur Amazon Elastic Container Service.

- Un démon Docker qui exécute au moins la version 1.9 et toutes les dépendances d'exécution de Docker. Pour plus d'informations, consultez la section [Check runtime dependencies](#) dans la documentation Docker.

Note

Nous recommandons la version Docker fournie avec et testée avec la version de l'agent Amazon ECS correspondante que vous utilisez. Amazon ECS fournit un journal des modifications pour la variante Linux de l'AMI optimisée pour Amazon ECS sur GitHub. Pour plus d'informations, consultez [Journal des modifications](#).

Recommandée

- Un processus d'initialisation et de démarrage pour exécuter et surveiller l'agent Amazon ECS. L'AMI optimisée Amazon ECS utilise le processus de `ecs-init` démarrage, et d'autres systèmes d'exploitation peuvent l'utiliser `systemd`. Pour plus d'informations et des exemples, consultez la section [Exemples de scripts de configuration des données utilisateur d'une instance de conteneur](#) dans le manuel Amazon Elastic Container Service Developer Guide. Pour plus d'information `ecs-init`, consultez le [ecs-init projet](#) sur GitHub. Au minimum, les environnements informatiques gérés nécessitent que l'agent Amazon ECS démarre au démarrage. Si l'agent Amazon ECS ne s'exécute pas sur votre ressource de calcul, il ne peut pas accepter de tâches provenant de AWS Batch.

L'AMI optimisée pour Amazon ECS est préconfigurée conformément à ces exigences et recommandations. Nous vous recommandons d'utiliser l'AMI optimisée Amazon ECS ou une AMI Amazon Linux avec le `ecs-init` package installé pour vos ressources de calcul. Choisissez une autre AMI si votre application nécessite un système d'exploitation spécifique ou une version de Docker qui n'y est pas encore disponible. AMIs Pour plus d'informations, consultez l'[AMI optimisée pour Amazon ECS](#) dans le guide du développeur Amazon Elastic Container Service.

Tutoriel : Création d'une AMI de ressources de calcul

Vous pouvez créer votre propre AMI de ressources de calcul personnalisée à utiliser pour vos environnements informatiques gérés et non gérés. Pour plus d'informations, consultez [Spécification](#)

[d'AMI de ressources de calcul](#). Ensuite, après avoir créé une AMI personnalisée, vous pouvez créer un environnement informatique qui utilise cette AMI et à laquelle vous pouvez associer une file d'attente de tâches. Enfin, commencez à soumettre des tâches à cette file d'attente.

Pour créer une AMI de ressources de calcul personnalisée

1. Choisissez une AMI de base à partir de laquelle commencer. L'AMI de base doit utiliser la virtualisation HVM. L'AMI de base ne peut pas être une AMI Windows.

 Note

L'AMI que vous choisissez pour un environnement de calcul doit correspondre à l'architecture des types d'instances que vous prévoyez d'utiliser pour cet environnement de calcul. Par exemple, si votre environnement de calcul utilise des types d'instances A1, l'AMI de ressource de calcul que vous choisissez doit prendre en charge les instances ARM. Amazon ECS vend les versions x86 et ARM de l'AMI Amazon Linux 2 optimisée pour Amazon ECS. Pour plus d'informations, consultez l'[AMI Amazon Linux 2 optimisée pour Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.

L'AMI Amazon Linux 2 optimisée pour Amazon ECS est l'AMI par défaut pour les ressources de calcul dans les environnements informatiques gérés. L'AMI Amazon Linux 2 optimisée pour Amazon ECS est préconfigurée et testée AWS Batch par des AWS ingénieurs. Il s'agit d'une AMI minimale avec laquelle vous pouvez démarrer et qui permet à vos ressources de calcul de fonctionner AWS rapidement. Pour plus d'informations, consultez l'[AMI optimisée Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.

Vous pouvez également choisir une autre variante d'Amazon Linux 2 et installer le `ecs-init` package à l'aide des commandes suivantes. Pour plus d'informations, consultez la section [Installation de l'agent de conteneur Amazon ECS sur une EC2 instance Amazon Linux 2](#) dans le manuel Amazon Elastic Container Service Developer Guide :

```
$ sudo amazon-linux-extras disable docker
$ sudo amazon-linux-extras install ecs-init
```

Par exemple, si vous souhaitez exécuter des charges de travail GPU sur vos ressources AWS Batch informatiques, vous pouvez commencer par l'[AMI Amazon Linux Deep Learning](#).

Configurez ensuite l'AMI pour exécuter AWS Batch des tâches. Pour de plus amples informations, veuillez consulter [Utiliser une AMI de charge de travail du GPU](#).

 Important

Vous pouvez choisir une AMI de base qui ne prend pas en charge le `ecs-init` package. Toutefois, si vous le faites, vous devez configurer un moyen de démarrer l'agent Amazon ECS au démarrage et de le maintenir en cours d'exécution. Vous pouvez également consulter plusieurs exemples de scripts de configuration des données utilisateur utilisés `systemd` pour démarrer et surveiller l'agent de conteneur Amazon ECS. Pour plus d'informations, consultez la section [Exemples de scripts de configuration des données utilisateur d'une instance de conteneur](#) dans le manuel Amazon Elastic Container Service Developer Guide.

2. Lancez une instance à partir de l'AMI de base que vous avez sélectionnée avec les options de stockage appropriées pour votre AMI. Vous pouvez configurer la taille et le nombre de volumes Amazon EBS attachés, ou de volumes de stockage d'instance si le type d'instance que vous avez sélectionné les prend en charge. Pour plus d'informations, consultez [Launching an instance](#) et [Amazon EC2 Instance Store](#) dans le guide de EC2 l'utilisateur Amazon.
3. Connectez-vous à votre instance SSH et effectuez toutes les tâches de configuration nécessaires. Cela peut inclure l'une ou l'ensemble des étapes suivantes :
 - Installation de l'agent de conteneur Amazon ECS. Pour plus d'informations, consultez la section [Installation de l'agent de conteneur Amazon ECS](#) dans le manuel du développeur Amazon Elastic Container Service.
 - Configurer un script pour formater les volumes de stockage d'instances.
 - Ajout d'un volume de stockage d'instance ou de systèmes de fichiers Amazon EFS au `/etc/fstab` fichier afin qu'ils soient montés au démarrage.
 - Configuration des options Docker, telles que l'activation du débogage ou le réglage de la taille de l'image de base.
 - Installer des packages ou copier des fichiers.

Pour plus d'informations, consultez la section [Connexion à votre instance Linux à l'aide de SSH](#) dans le guide de l'EC2 utilisateur Amazon.

4. Si vous avez démarré l'agent de conteneur Amazon ECS sur votre instance, vous devez l'arrêter et supprimer tous les fichiers de points de contrôle de données persistants avant de créer votre

AMI. Sinon, si vous ne le faites pas, l'agent ne démarrera pas sur les instances lancées depuis votre AMI.

a. Arrêtez l'agent de conteneur Amazon ECS.

- AMI Amazon Linux 2 optimisée pour Amazon ECS :

```
sudo systemctl stop ecs
```

- AMI Amazon Linux optimisée pour Amazon ECS :

```
sudo stop ecs
```

b. Supprimez les fichiers de point de contrôle des données persistants. Par défaut, ces fichiers se trouvent dans le `/var/lib/ecs/data/` répertoire. Utilisez la commande suivante pour supprimer ces fichiers, le cas échéant.

```
sudo rm -rf /var/lib/ecs/data/*
```

5. Créez une nouvelle AMI à partir de votre instance en cours d'exécution. Pour plus d'informations, consultez la section [Création d'une AMI Linux basée sur Amazon EBS](#) dans le guide de EC2 l'utilisateur Amazon.

Pour utiliser votre nouvelle AMI avec AWS Batch

1. Une fois la nouvelle AMI créée, créez un environnement de calcul avec la nouvelle AMI. Pour ce faire, choisissez le type d'image et entrez l'ID d'AMI personnalisé dans la case de remplacement de l'ID d'image lorsque vous créez l'environnement AWS Batch informatique. Pour de plus amples informations, veuillez consulter [the section called "Tutoriel : Création d'un environnement informatique géré à l'aide EC2 des ressources Amazon"](#).

Note

L'AMI que vous choisissez pour un environnement de calcul doit correspondre à l'architecture des types d'instances que vous prévoyez d'utiliser pour cet environnement de calcul. Par exemple, si votre environnement de calcul utilise des types d'instances A1, l'AMI de ressource de calcul que vous choisissez doit prendre en charge les instances ARM. Amazon ECS vend les versions x86 et ARM de l'AMI Amazon Linux 2 optimisée pour Amazon ECS. Pour plus d'informations, consultez l'[AMI Amazon Linux](#)

[2 optimisée pour Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.

2. Créez une file d'attente de tâches et associez votre nouvel environnement de calcul. Pour de plus amples informations, veuillez consulter [Création d'une file d'attente de tâches](#).

Note

Tous les environnements de calcul associés à une file d'attente de tâches doivent partager la même architecture. AWS Batch ne prend pas en charge le mélange des types d'architectures d'environnements de calcul dans une file d'attente de tâches unique.

3. (Facultatif) Soumettez un exemple de tâche à votre nouvelle file d'attente de tâches. Pour plus d'informations, consultez [Exemples de définitions de tâches](#), [Création d'une définition de tâche à nœud unique](#) et [Tutoriel : soumettre une offre d'emploi](#).

Utiliser une AMI de charge de travail du GPU

Pour exécuter des charges de travail GPU sur vos ressources AWS Batch informatiques, vous devez utiliser une AMI compatible avec le GPU. Pour plus d'informations, consultez les [sections Travailler avec GPUs Amazon ECS et Amazon ECS optimisé](#) dans le manuel du développeur d'AMIs Amazon Elastic Container Service.

Dans les environnements informatiques gérés, si l'environnement de calcul spécifie des p3 types d'instances ou des familles d'instances, il AWS Batch utilise une AMI optimisée pour le GPU Amazon ECS. p4 p5 p6 g3 g3s g4 g5

Dans les environnements informatiques non gérés, une AMI optimisée pour le GPU Amazon ECS est recommandée. Vous pouvez utiliser les [GetParametersByPath](#) opérations AWS Command Line Interface ou AWS Systems Manager Parameter Store [GetParameterGetParameters](#), et pour récupérer les métadonnées de l'Amazon ECS recommandé optimisé pour le GPU AMIs.

Note

La famille d'p5instances n'est prise en charge que sur les versions égales ou ultérieures à celles 20230912 de l'AMI optimisée pour le GPU Amazon ECS et elle est incompatible p2 avec g2 les types d'instances. Si vous devez utiliser des p5 instances, assurez-

vous que votre environnement de calcul ne contient pas p2 d'g2 instances et qu'il utilise la dernière AMI Batch par défaut. La création d'un nouvel environnement de calcul utilisera la dernière AMI, mais si vous mettez à jour votre environnement de calcul pour l'inclure p5, vous pouvez vous assurer que vous utilisez la dernière AMI en la définissant [updateToLatestImageVersion](#) true dans les ComputeResource propriétés. Pour plus d'informations sur la compatibilité des AMI avec les instances GPU, consultez la section [Travailler avec GPUs sur Amazon ECS](#) dans le manuel du développeur Amazon Elastic Container Service.

Les exemples suivants montrent comment utiliser la [GetParameter](#) commande.

AWS CLI

```
$ aws ssm get-parameter --name /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended \
--region us-east-2 --output json
```

La sortie inclut les informations de l'AMI dans le Value paramètre.

```
{
  "Parameter": {
    "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended",
    "LastModifiedDate": 1555434128.664,
    "Value": "{\"schema_version\":1,\"image_name\": \"amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs\", \"image_id\": \"ami-083c800fe4211192f\", \"os\": \"Amazon Linux 2\", \"ecs_runtime_version\": \"Docker version 18.06.1-ce\", \"ecs_agent_version\": \"1.27.0\"}",
    "Version": 9,
    "Type": "String",
    "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended"
  }
}
```

Python

```
from __future__ import print_function

import json
```

```
import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameter(Name='/aws/service/ecs/optimized-ami/amazon-linux-2/
gpu/recommended')
jsonVal = json.loads(response['Parameter']['Value'])
print("image_id  = " + jsonVal['image_id'])
print("image_name = " + jsonVal['image_name'])
```

La sortie inclut uniquement l'ID d'AMI et le nom d'AMI :

```
image_id  = ami-083c800fe4211192f
image_name = amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs
```

Les exemples suivants illustrent l'utilisation de [GetParameters](#).

AWS CLI

```
$ aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended/image_name \
                               /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended/image_id \
                               --region us-east-2 --output json
```

La sortie inclut toutes les métadonnées pour chacun des paramètres :

```
{
  "InvalidParameters": [],
  "Parameters": [
    {
      "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id",
      "LastModifiedDate": 1555434128.749,
      "Value": "ami-083c800fe4211192f",
      "Version": 9,
      "Type": "String",
      "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_id"
    },
    {
```

```

        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name",
        "LastModifiedDate": 1555434128.712,
        "Value": "amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs",
        "Version": 9,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_name"
    }
]
}

```

Python

```

from __future__ import print_function

import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameters(
    Names=['/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name',
          '/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id'])
for parameter in response['Parameters']:
    print(parameter['Name'] + " = " + parameter['Value'])

```

La sortie inclut l'ID et le nom de l'AMI, en utilisant le chemin complet des noms.

```

/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_id =
ami-083c800fe4211192f
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_name = amzn2-
ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs

```

Les exemples suivants montrent comment utiliser la [GetParametersByPath](#) commande.

AWS CLI

```

$ aws ssm get-parameters-by-path --path /aws/service/ecs/optimized-ami/amazon-
linux-2/gpu/recommended \

```

```
--region us-east-2 --output json
```

La sortie inclut les métadonnées complètes pour tous les paramètres sous le chemin spécifié.

```
{
  "Parameters": [
    {
      "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
ecs_agent_version",
      "LastModifiedDate": 1555434128.801,
      "Value": "1.27.0",
      "Version": 8,
      "Type": "String",
      "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/ecs_agent_version"
    },
    {
      "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
ecs_runtime_version",
      "LastModifiedDate": 1548368308.213,
      "Value": "Docker version 18.06.1-ce",
      "Version": 1,
      "Type": "String",
      "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/ecs_runtime_version"
    },
    {
      "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id",
      "LastModifiedDate": 1555434128.749,
      "Value": "ami-083c800fe4211192f",
      "Version": 9,
      "Type": "String",
      "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_id"
    },
    {
      "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name",
      "LastModifiedDate": 1555434128.712,
      "Value": "amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs",
      "Version": 9,
      "Type": "String",

```



```

        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_name"
    },
    {
        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
os",
        "LastModifiedDate": 1548368308.143,
        "Value": "Amazon Linux 2",
        "Version": 1,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/os"
    },
    {
        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
schema_version",
        "LastModifiedDate": 1548368307.914,
        "Value": "1",
        "Version": 1,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/schema_version"
    }
]
}

```

Python

```

from __future__ import print_function

import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameters_by_path(Path='/aws/service/ecs/optimized-ami/amazon-
linux-2/gpu/recommended')
for parameter in response['Parameters']:
    print(parameter['Name'] + " = " + parameter['Value'])

```

La sortie inclut les valeurs de tous les noms de paramètres du chemin spécifié, en utilisant le chemin complet pour les noms.

```
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/ecs_agent_version =  
1.27.0  
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/ecs_runtime_version =  
Docker version 18.06.1-ce  
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_id =  
ami-083c800fe4211192f  
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_name = amzn2-  
ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs  
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/os = Amazon Linux 2  
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/schema_version = 1
```

Pour plus d'informations, consultez la section [Extraction des métadonnées AMI optimisées pour Amazon ECS](#) dans le manuel du développeur Amazon Elastic Container Service.

Obsolète d'Amazon Linux

L'AMI Amazon Linux (également appelée Amazon Linux 1) a atteint sa fin de vie le 31 décembre 2023. AWS Batch a mis fin à la prise en charge de l'AMI Amazon Linux car elle ne recevra aucune mise à jour de sécurité ni aucune correction de bogue à compter du 1er janvier 2024. Pour plus d'informations sur Amazon Linux end-of-life, consultez la [FAQ AL](#).

Nous vous recommandons de mettre à jour les environnements informatiques existants basés sur Amazon Linux vers Amazon Linux 2023 afin d'éviter les interruptions imprévues de la charge de travail et de continuer à recevoir des mises à jour de sécurité et autres.

Vos environnements informatiques utilisant l'AMI Amazon Linux peuvent continuer à fonctionner au-delà de la end-of-life date du 31 décembre 2023. Toutefois, ces environnements informatiques ne recevront plus de nouvelles mises à jour logicielles, de correctifs de sécurité ou de corrections de bogues de la part de AWS. Il est de votre responsabilité de maintenir ces environnements de calcul sur l'AMI Amazon Linux par la suite end-of-life. Nous recommandons de migrer les environnements AWS Batch informatiques vers Amazon Linux 2023 ou Amazon Linux 2 afin de maintenir des performances et une sécurité optimales.

Pour obtenir de l'aide lors AWS Batch de la migration de l'AMI Amazon Linux vers Amazon Linux 2023 ou Amazon Linux 2, consultez la section [Mise à jour des environnements de calcul - AWS Batch](#).

Amazon EKS : dépréciation de l'AMI Amazon Linux 2

AWS mettra fin à la prise en charge d'Amazon Linux 2 AMIs optimisé pour Amazon EKS à compter du 26/11/25. Nous vous recommandons de migrer les environnements informatiques AWS Batch Amazon EKS vers Amazon Linux 2023 avant le 26/11/25 afin de maintenir des performances et une sécurité optimales.

Bien que vous puissiez continuer à utiliser Amazon Linux 2 optimisé pour Amazon EKS fourni par lots AMIs sur vos environnements de calcul Amazon EKS au-delà de la end-of-support date du 26/11/25, ces environnements de calcul ne recevront plus de nouvelles mises à jour logicielles, de correctifs de sécurité ou de corrections de bogues de la part de ces environnements de calcul. AWS Il est de votre responsabilité de maintenir ces environnements de calcul sur l'AMI Amazon Linux 2 optimisée pour Amazon EKS par la suite end-of-life.

Pour plus d'informations sur l'Amazon EKS AL2 end-of-life, consultez la section relative à la [dépréciation de l'AMI Amazon EKS FAQs](#) dans le guide de l'utilisateur Amazon EKS.

Pour obtenir de l'aide AWS Batch sur la migration des environnements informatiques Amazon EKS d'Amazon Linux 2 vers Amazon Linux 2023, consultez [Comment passer d'EKS AL2 à EKS AL2023](#).

Amazon ECS : dépréciation de l'AMI Amazon Linux 2

AWS mettra fin au support d'Amazon Linux 2. À compter de janvier 2026, AWS Batch l'AMI par défaut pour les nouveaux environnements de calcul Amazon ECS passera d'Amazon Linux 2 à Amazon Linux 2023. Nous vous recommandons de migrer les environnements informatiques AWS Batch Amazon ECS vers Amazon Linux 2023 afin de maintenir des performances et une sécurité optimales.

Pour plus d'informations sur Amazon Linux 2 end-of-life, consultez [Amazon Linux 2 FAQs](#).

Pour plus d'informations sur les différences entre Amazon Linux 2 et Amazon Linux 2023, consultez [Comparer Amazon Linux 2023 et Amazon Linux 2](#) dans le guide de l'utilisateur Amazon Linux 2023.

Pour plus d'informations sur les modifications apportées à Amazon Linux 2023 pour les AMI optimisées pour Amazon ECS, consultez la section [Migration d'une AMI optimisée pour Amazon Linux 2 vers Amazon Linux 2023 optimisée pour Amazon ECS dans le guide de l'utilisateur](#) Amazon ECS.

Pour obtenir de l'aide AWS Batch sur la migration des environnements informatiques Amazon ECS d'Amazon Linux 2 vers Amazon Linux 2023, consultez [Comment migrer d'ECS AL2 vers ECS AL2 023](#).

Utilisez les modèles de lancement Amazon EC2 avec AWS Batch

AWS Batch prend en charge l'utilisation des modèles de lancement Amazon EC2 avec vos environnements informatiques EC2. Avec les modèles de lancement, vous pouvez modifier la configuration par défaut de vos ressources AWS Batch informatiques sans avoir à créer des modèles personnalisés AMIs.

Note

Les modèles de lancement ne sont pas pris en charge sur les AWS ressources Fargate.

Vous devez créer un modèle de lancement avant de pouvoir l'associer à un environnement de calcul. Vous pouvez créer un modèle de lancement dans la console Amazon EC2. Vous pouvez également utiliser le AWS CLI ou un AWS SDK. Par exemple, le fichier JSON suivant représente un modèle de lancement qui redimensionne le volume de données Docker pour l'AMI de ressources de AWS Batch calcul par défaut et le définit également pour qu'il soit chiffré.

```
{
  "LaunchTemplateName": "increase-container-volume-encrypt",
  "LaunchTemplateData": {
    "BlockDeviceMappings": [
      {
        "DeviceName": "/dev/xvda",
        "Ebs": {
          "Encrypted": true,
          "VolumeSize": 100,
          "VolumeType": "gp2"
        }
      }
    ]
  }
}
```

Vous pouvez créer le modèle de lancement précédent en enregistrant le JSON dans un fichier appelé `lt-data.json` et en exécutant la AWS CLI commande suivante.

```
aws ec2 --region <region> create-launch-template --cli-input-json file://lt-data.json
```

Pour plus d'informations sur les modèles de lancement, consultez [Lancer une instance à partir d'un modèle de lancement](#) dans le guide de l'utilisateur Amazon EC2.

Si vous utilisez un modèle de lancement pour créer votre environnement de calcul, vous pouvez déplacer les paramètres suivants de l'environnement de calcul existant dans votre modèle de lancement :

 Note

Supposons que l'un de ces paramètres (à l'exception des balises Amazon EC2) soit spécifié à la fois dans le modèle de lancement et dans la configuration de l'environnement de calcul. Les paramètres de l'environnement de calcul sont alors prioritaires. Les balises Amazon EC2 sont fusionnées entre le modèle de lancement et la configuration de l'environnement de calcul. En cas de collision sur la clé de la balise, la valeur de la configuration de l'environnement de calcul est prioritaire.

- Paire de clés Amazon EC2
- ID d'AMI Amazon EC2
- Groupe de sécurité IDs
- Balises Amazon EC2

Les paramètres du modèle de lancement suivants sont ignorés par AWS Batch :

- Type d'instance (spécifiez vos types d'instance souhaités lorsque vous créez votre environnement de calcul)
- Rôle d'instance (spécifiez votre rôle d'instance souhaité lorsque vous créez votre environnement de calcul)
- Sous-réseaux d'interface réseau (spécifiez vos sous-réseaux souhaités lorsque vous créez votre environnement de calcul)
- Options du marché des instances (AWS Batch doit contrôler la configuration de l'instance Spot)
- Désactiver la terminaison de l'API (AWS Batch doit contrôler le cycle de vie de l'instance)

AWS Batch met uniquement à jour le modèle de lancement avec une nouvelle version du modèle de lancement lors des mises à jour de l'infrastructure. Pour de plus amples informations, veuillez consulter [Mettre à jour un environnement informatique dans AWS Batch](#).

Modèles de lancement par défaut et de remplacement

Vous pouvez définir un modèle de lancement par défaut pour l'environnement de calcul et un modèle de lancement de remplacement pour des types et familles d'instances spécifiques. Cela peut vous être utile pour que le modèle par défaut soit utilisé pour la majorité des types d'instances dans les environnements informatiques.

Les variables `$Default` de substitution `$Latest` peuvent être utilisées au lieu de nommer une version spécifique. Si vous ne fournissez pas de modèle de lancement de remplacement, le modèle de lancement par défaut est automatiquement appliqué.

Si vous utilisez la `$Latest` variable `$Default` or, elle AWS Batch appliquera les informations actuelles au moment de la création de l'environnement de calcul. Si la version par défaut ou la dernière version change à l'avenir, vous devez mettre à jour les informations via [UpdateComputeEnvironment](#) ou via le AWS Management Console - AWS Batch.

Pour plus de flexibilité, vous pouvez définir des modèles de lancement de remplacement qui seront appliqués à des types ou familles d'instances de calcul spécifiques.

Note

Vous pouvez spécifier jusqu'à dix (10) modèles de lancement de remplacement par environnement informatique.

Utilisez le `targetInstanceTypes` paramètre pour sélectionner le type ou la famille d'instance qui doit utiliser ce modèle de lancement de remplacement. Le type ou la famille d'instance doit d'abord être identifié par le [instanceTypes](#) paramètre.

Si vous définissez des remplacements de modèles de lancement et que vous décidez de les supprimer ultérieurement, vous pouvez transmettre un tableau vide pour annuler le [overrides](#) paramètre dans l'opération d'[UpdateComputeEnvironment](#) API. Vous pouvez également choisir de ne pas inclure le `overrides` paramètre lors de la soumission de l'opération `UpdateComputeEnvironment` d'API. Pour plus d'informations, voir [LaunchTemplateSpecification.overrides](#)

Pour plus d'informations, consultez

[LaunchTemplateSpecificationOverride.targetInstanceTypes](#) le guide de référence des AWS Batch API.

Données utilisateur Amazon EC2 dans les modèles de lancement

Vous pouvez fournir des données utilisateur Amazon EC2 dans votre modèle de lancement qui est exécuté par [cloud-init](#) lors du lancement de vos instances. Vos données utilisateur peuvent exécuter des scénarios de configuration courants, notamment les suivants :

- [Inclusion d'utilisateurs ou de groupes](#)
- [Installations de packages](#)
- [Création de partitions et systèmes de fichiers](#)

Les données utilisateur Amazon EC2 figurant dans les modèles de lancement doivent être au format d'archive en [plusieurs parties MIME](#). Cela est dû au fait que vos données utilisateur sont fusionnées avec AWS Batch d'autres données utilisateur nécessaires à la configuration de vos ressources informatiques. Vous pouvez combiner plusieurs blocs de données utilisateur dans un seul fichier MIME multi-part. Par exemple, vous souhaitez peut-être combiner un bootbook dans le cloud qui configure le démon Docker avec un script shell de données utilisateur qui écrit les informations de configuration pour l'agent de conteneur Amazon ECS.

Si vous en utilisez AWS CloudFormation, le [AWS::CloudFormation::Init](#) type peut être utilisé avec le script d'assistance [cfn-init](#) pour exécuter des scénarios de configuration courants.

Un fichier MIME multi-part est constitué des composants suivants :

- Le type de contenu et la déclaration de limite : `Content-Type: multipart/mixed; boundary=="==BOUNDARY=="`
- La déclaration de version MIME : `MIME-Version: 1.0`
- Un ou plusieurs blocs de données utilisateur contenant les composants suivants :
 - La limite d'ouverture qui indique le début d'un bloc de données utilisateur : `--==BOUNDARY==`. Vous devez laisser la ligne avant cette limite vide.
 - La déclaration de type de contenu du bloc : `Content-Type: text/cloud-config; charset="us-ascii"`. Pour plus d'informations sur les types de contenu, consultez la [documentation sur Cloud-Init](#). Vous devez laisser la ligne après la déclaration de type de contenu vide.

- Le contenu des données utilisateur, tel qu'une liste de commandes ou de `cloud-init` directives du shell.
- La limite de fermeture qui indique la fin du fichier MIME en plusieurs parties :. --
==BOUNDARY==-- Vous devez laisser la ligne avant la limite de fermeture vide.

Note

Si vous ajoutez des données utilisateur à un modèle de lancement dans la console Amazon EC2, vous pouvez les coller sous forme de texte brut. Vous pouvez également le télécharger à partir d'un fichier. Si vous utilisez le AWS CLI ou un AWS SDK, vous devez d'abord base64 encoder les données utilisateur et envoyer cette chaîne comme valeur du UserData paramètre lorsque vous appelez [CreateLaunchTemplate](#), comme indiqué dans ce fichier JSON.

```
{
  "LaunchTemplateName": "base64-user-data",
  "LaunchTemplateData": {
    "UserData":
      "ewogICAgIkxhdW5jaFRlbXBsYXRlTmFtZSI6ICJpbmNyZWZzZS1jb250YWluZXItbm9sdW..."
  }
}
```

Rubriques

- [Référence : exemples de modèles de lancement Amazon EC2](#)

Référence : exemples de modèles de lancement Amazon EC2

Vous trouverez ci-dessous des exemples de fichiers MIME en plusieurs parties que vous pouvez utiliser pour créer vos propres modèles.

Exemples

- [Exemple : monter un système de fichiers Amazon EFS existant](#)
- [Exemple : remplacer la configuration par défaut de l'agent de conteneur Amazon ECS](#)
- [Exemple : monter un système de fichiers Amazon FSx for Lustre existant](#)

Exemple : monter un système de fichiers Amazon EFS existant

Exemple

Cet exemple de fichier MIME en plusieurs parties configure la ressource de calcul sur laquelle installer le `amazon-efs-utils` package et monter un système de fichiers Amazon EFS existant. `/mnt/efs`

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-efs-utils

runcmd:
- file_system_id_01=fs-abcdef123
- efs_directory=/mnt/efs

- mkdir -p ${efs_directory}
- echo "${file_system_id_01}:/ ${efs_directory} efs tls,_netdev" >> /etc/fstab
- mount -a -t efs defaults

--==MYBOUNDARY==--
```

Exemple : remplacer la configuration par défaut de l'agent de conteneur Amazon ECS

Exemple

Cet exemple de fichier MIME en plusieurs parties remplace les paramètres de nettoyage de l'image Docker par défaut par une ressource de calcul.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
echo ECS_IMAGE_CLEANUP_INTERVAL=60m >> /etc/ecs/ecs.config
```

```
echo ECS_IMAGE_MINIMUM_CLEANUP_AGE=60m >> /etc/ecs/ecs.config

---MYBOUNDARY---
```

Exemple : monter un système de fichiers Amazon FSx for Lustre existant

Exemple

Cet exemple de fichier MIME en plusieurs parties configure la ressource de calcul pour installer le `lustre2.10` package à partir de la bibliothèque Extras et y monter un système de fichiers existant FSx pour Lustre avec le nom `/scratch` de montage de. `fsx` Cet exemple concerne Amazon Linux 2. Pour les instructions d'installation pour les autres distributions Linux, consultez la section [Installation du client Lustre](#) dans le guide de l'utilisateur d'Amazon FSx pour Lustre. Pour plus d'informations, consultez la section [Montage automatique de votre système de FSx fichiers Amazon](#) dans le guide de l'utilisateur Amazon FSx for Lustre.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary==="MYBOUNDARY==="

---MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

runcmd:
- file_system_id_01=fs-0abcdef1234567890
- region=us-east-2
- fsx_directory=/scratch
- amazon-linux-extras install -y lustre2.10
- mkdir -p ${fsx_directory}
- mount -t lustre ${file_system_id_01}.fsx.${region}.amazonaws.com@tcp:fsx
  ${fsx_directory}

---MYBOUNDARY---
```

Dans les membres [volumes](#) et [MountPoints](#) des propriétés du conteneur, les points de montage doivent être mappés dans le conteneur.

```
{
  "volumes": [
    {
      "host": {
        "sourcePath": "/scratch"
```

```
    },
    "name": "Scratch"
  }
],
"mountPoints": [
  {
    "containerPath": "/scratch",
    "sourceVolume": "Scratch"
  }
],
}
```

Configuration du service de métadonnées d'instance (IMDS)

Le service de métadonnées d'instance (IMDS) fournit des métadonnées relatives à vos EC2 instances aux applications qui s'exécutent sur ces instances. Utilisez-le IMDSv2 pour toutes les nouvelles charges de travail et migrez les charges de travail existantes de IMDSv1 vers IMDSv2 pour améliorer la sécurité. Pour plus d'informations sur l'IMDS et la configuration de l'IMDS, consultez les sections [Utiliser les métadonnées d'instance pour gérer votre EC2 instance](#) et [Configurer les options de métadonnées d'instance pour les nouvelles instances](#) dans le guide de EC2 l'utilisateur Amazon.

Scénarios de configuration

Choisissez la méthode de configuration appropriée en fonction de la configuration de votre environnement informatique :

AMI par défaut sans modèle de lancement

Lorsque vous utilisez l' AWS Batch AMI par défaut et que vous ne spécifiez pas de modèle de lancement, choisissez l'une des options suivantes :

1. Utiliser l'AMI par défaut d'Amazon Linux 2023. Amazon Linux 2023 l'exige IMDSv2 par défaut. Lorsque vous créez votre environnement informatique, sélectionnez Amazon Linux 2023 comme type d'image.
2. Définissez la IMDSv2 configuration au niveau du compte : configurez votre AWS compte de manière à ce qu'il soit requis IMDSv2 pour toutes les nouvelles instances. Ce paramètre affecte toutes les nouvelles instances que vous lancez dans le compte. Pour obtenir des instructions, consultez [la section Définir IMDSv2 par défaut pour le compte](#) dans le guide de EC2 l'utilisateur Amazon.

 Note

La configuration IMDS au niveau du compte peut être remplacée par le modèle de lancement ou la configuration de l'AMI. Les paramètres du modèle de lancement ont priorité sur les paramètres au niveau du compte.

AMI personnalisée sans modèle de lancement

Lorsque vous utilisez une AMI personnalisée sans modèle de lancement, choisissez l'une des options suivantes :

1. Utiliser Amazon Linux 2023 comme base : créez votre AMI personnalisée en utilisant Amazon Linux 2023 comme image de base. Pour plus d'informations sur la création d' AMIs une version personnalisée pour Batch, consultez [Tutoriel : Création d'une AMI de ressources de calcul](#).
2. Configuration IMDSv2 dans votre AMI personnalisée : lorsque vous créez votre AMI personnalisée, configurez-la selon les besoins IMDSv2. Pour obtenir des instructions, consultez la section [Configurer les options de métadonnées d'instance pour une AMI personnalisée](#) dans le guide de EC2 l'utilisateur Amazon.
3. Définissez la IMDSv2 configuration au niveau du compte : configurez votre AWS compte de manière à ce qu'il soit requis IMDSv2 pour toutes les nouvelles instances. Ce paramètre affecte toutes les nouvelles instances que vous lancez dans le compte. Pour obtenir des instructions, consultez [la section Définir IMDSv2 par défaut pour le compte](#) dans le guide de EC2 l'utilisateur Amazon.

 Note

La configuration IMDS au niveau du compte peut être remplacée par le modèle de lancement ou la configuration de l'AMI. Les paramètres du modèle de lancement ont priorité sur les paramètres au niveau du compte.

Utilisation de modèles de lancement

Lorsque vous utilisez des modèles de lancement dans votre environnement informatique, ajoutez les options de métadonnées requises à votre modèle de lancement IMDSv2. Pour plus d'informations

sur l'utilisation de modèles de lancement avec Batch, consultez [Utilisez les modèles de lancement Amazon EC2 avec AWS Batch](#).

```
{
  "LaunchTemplateName": "batch-imsdv2-template",
  "VersionDescription": "IMDSv2 only template for Batch",
  "LaunchTemplateData": {
    "MetadataOptions": {
      "HttpTokens": "required"
    }
  }
}
```

Créez le modèle de lancement à l'aide de la AWS CLI :

```
aws ec2 create-launch-template --cli-input-json file://imds-template.json
```

EC2 configurations

AWS Batch utilise Amazon ECS optimisé AMIs pour les environnements informatiques EC2 Spot EC2 et Amazon. La valeur par défaut est [Amazon Linux 2](#) (ECS_AL2). À compter de janvier 2026, la valeur par défaut sera [AL2023](#) (ECS_AL2023).

AWS mettra fin au support d'Amazon Linux 2. Nous vous recommandons de migrer les environnements informatiques AWS Batch Amazon ECS vers Amazon Linux 2023 afin de maintenir des performances et une sécurité optimales. Pour de plus amples informations, veuillez consulter [Amazon ECS : dépréciation de l'AMI Amazon Linux 2](#).

Nous vous recommandons de mettre à jour les environnements informatiques existants basés sur Amazon Linux vers Amazon Linux 2023 afin d'éviter les interruptions imprévues de la charge de travail et de continuer à recevoir des mises à jour de sécurité et autres.

Pour obtenir de l'aide lors AWS Batch de la migration de l'AMI Amazon Linux vers Amazon Linux 2023, consultez [Comment migrer d'ECS AL2 vers ECS AL2 023](#)

Rubriques

- [Comment migrer d'ECS AL2 vers ECS AL2 023](#)

Comment migrer d'ECS AL2 vers ECS AL2 023

AL2023 est un système d'exploitation basé sur Linux conçu pour fournir un environnement sécurisé, stable et performant pour vos applications cloud. Pour plus d'informations sur les différences entre AL2 et AL2 023, consultez [Comparer Amazon Linux 2023 et Amazon Linux 2](#) dans le guide de l'utilisateur Amazon Linux 2023.

À compter de janvier 2026, AWS Batch l'AMI par défaut pour les nouveaux environnements de calcul Amazon ECS passera d'Amazon Linux 2 à Amazon Linux 2023, car le [support d'Amazon Linux 2 ne AWS](#) sera plus pris en charge. L'AMI par défaut est utilisée lorsque vous ne spécifiez aucune valeur pour le champ [ImageType.EC2Configuration](#) lors de la création d'un nouvel environnement informatique. Nous vous recommandons de migrer les environnements informatiques AWS Batch Amazon ECS vers Amazon Linux 2023 afin de maintenir des performances et une sécurité optimales.

En fonction de la configuration de votre environnement informatique, vous pouvez utiliser l'un des chemins de mise à niveau suivants de la version AL2 à AL2 023.

Mise à niveau à l'aide d'Ec2Configuration. ImageType

- Si vous n'utilisez pas de modèle de lancement ou si vous ne remplacez pas le modèle de lancement, modifiez [Ec2Configuration. ImageType](#) pour ECS_AL2023 (ou ECS_AL2023_NVIDIA lors de l'utilisation d'instances GPU), puis exécuter [UpdateComputeEnvironment](#).
- Si vous spécifiez une configuration [Ec2. ImageIdOverride](#) puis [Ec2Configuration. ImageType](#) doit correspondre au type d'AMI spécifié dans [Ec2Configuration. ImageIdOverride](#).

Si vous ne correspondez ImageIdOverride pas ImageType, l'environnement de calcul risque de ne pas fonctionner correctement.

Mise à niveau en utilisant des modèles de lancement

- Si vous utilisez un modèle de lancement qui spécifie une AMI basée sur ECS_AL2023, assurez-vous que votre modèle de lancement est compatible avec Amazon Linux 2023. Pour plus d'informations sur les modifications apportées à Amazon Linux 2023 pour les AMI optimisées pour Amazon ECS, consultez la section [Migration d'une AMI optimisée pour Amazon Linux 2 vers Amazon Linux 2023 optimisée pour Amazon ECS dans le guide de l'utilisateur Amazon ECS](#).
- Pour AL2 023 AMIs, vérifiez que les données utilisateur personnalisées ou les scripts d'initialisation sont compatibles avec l'environnement AL2 023 et le système de gestion des packages.

Mise à niveau en utilisant CloudFormation

- Si vous avez l'habitude de gérer vos environnements informatiques, mettez à jour votre modèle pour modifier la `ImageType` propriété dans le formulaire `Ec2Configuration` `ECS_AL2` vers `ECS_AL2023` (ou `ECS_AL2023_NVIDIA` lorsque vous utilisez des instances GPU) :

```
ComputeEnvironment:
  Type: AWS::Batch::ComputeEnvironment
  Properties:
    ComputeResources:
      Ec2Configuration:
        - ImageType: ECS_AL2023
```

Mettez ensuite à jour votre CloudFormation pile pour appliquer les modifications.

- Si votre CloudFormation modèle spécifie une AMI personnalisée en utilisant `ImageIdOverride`, assurez-vous que l'ID d'AMI correspond à une AL2 AMI basée sur 023 et correspond au `ImageType` paramètre.

Considérations concernant la migration

Lors de la migration d'Amazon Linux 2 vers Amazon Linux 2023, tenez compte des points suivants :

- Gestion des packages — Amazon Linux 2023 utilise `dnf` plutôt que `yum` pour la gestion des packages.
- Services système — Certains services système et leurs configurations peuvent différer entre AL2 et 023.
- Runtime du conteneur — La version 023 AL2 et la AL2 version 023 prennent en charge Docker, mais la version AL2 023 peut avoir des configurations par défaut différentes.
- Sécurité — La version AL2 023 inclut des fonctionnalités de sécurité améliorées et peut nécessiter des mises à jour des configurations liées à la sécurité.
- Service de métadonnées d'instance version 2 (IMDSv2) : IMDSv2 service orienté session qui nécessite une authentification basée sur des jetons pour accéder aux métadonnées de l'EC2 instance, offrant ainsi une sécurité renforcée. Pour plus d'informations sur l'IMDS, consultez la section [Comment fonctionne le service de métadonnées d'instance version 2](#) dans le guide de l'utilisateur Amazon.

Pour obtenir une liste complète des modifications et des considérations relatives à la migration, consultez la section [Migration d'une AMI Amazon Linux 2 vers Amazon Linux 2023 optimisée pour Amazon ECS](#) dans le guide de l'utilisateur Amazon ECS.

Stratégies d'allocation des types d'instance pour AWS Batch

Lorsqu'un environnement informatique géré est créé, AWS Batch sélectionne les types d'instances [instanceTypes](#) spécifiés qui répondent le mieux aux besoins des tâches. La stratégie d'allocation définit le comportement à adopter en cas de besoin de capacités supplémentaires. Ce paramètre ne s'applique pas aux tâches exécutées sur les ressources Fargate. Ne spécifiez pas ce paramètre.

BEST_FIT (default)

AWS Batch sélectionne le type d'instance qui répond le mieux aux besoins des tâches, en privilégiant le type d'instance le moins coûteux. Si des instances supplémentaires du type d'instance sélectionné ne sont pas disponibles, AWS Batch attend qu'elles soient disponibles. Si le nombre d'instances disponibles est insuffisant ou si l'utilisateur atteint les [quotas de EC2 service Amazon](#), les tâches supplémentaires ne sont pas exécutées tant que les tâches en cours ne sont pas terminées. Cette stratégie d'allocation permet de réduire les coûts mais peut limiter la mise à l'échelle. Si vous utilisez Spot Fleets avec BEST_FIT, le rôle Spot Fleet IAM doit être spécifié. BEST_FIT n'est pas pris en charge lors de la mise à jour des environnements informatiques. Pour de plus amples informations, veuillez consulter [Mettre à jour un environnement informatique dans AWS Batch](#).

Note

AWS Batch gère AWS les ressources de votre compte. Les environnements de calcul dotés de la stratégie d'allocation BEST_FIT utilisaient initialement les configurations de lancement par défaut. Cependant, l'utilisation de configurations de lancement avec de nouveaux AWS comptes sera limitée au fil du temps. Par conséquent, à compter de fin avril 2024, les environnements de calcul BEST_FIT nouvellement créés lanceront par défaut des modèles. Si votre rôle de service n'est pas autorisé à gérer les modèles de lancement, vous pouvez continuer à utiliser les configurations de lancement. Les environnements informatiques existants continueront d'utiliser des configurations de lancement.

BEST_FIT_PROGRESSIVE

AWS Batch sélectionne des types d'instances supplémentaires suffisamment grands pour répondre aux exigences des tâches de la file d'attente. Les types d'instances dont le coût est inférieur pour chaque unité de vCPU sont préférés. Si des instances supplémentaires des types d'instance précédemment sélectionnés ne sont pas disponibles, AWS Batch sélectionne de nouveaux types d'instance.

Note

Pour les [tâches parallèles à nœuds multiples](#), AWS Batch choisit le type d'instance optimal disponible. Si le type d'instance devient indisponible en raison d'une capacité insuffisante, les autres types d'instance de la famille ne sont pas lancés.

SPOT_CAPACITY_OPTIMIZED

AWS Batch sélectionne un ou plusieurs types d'instances suffisamment grands pour répondre aux exigences des tâches de la file d'attente. Les types d'instance les moins susceptibles d'être interrompus sont préférés. Cette stratégie d'allocation n'est disponible que pour les ressources de calcul d'instance Spot.

SPOT_PRICE_CAPACITY_OPTIMIZED

La stratégie d'allocation optimisée en termes de prix et de capacité prend en compte à la fois le prix et la capacité afin de sélectionner les groupes d'instances Spot les moins susceptibles d'être interrompus et dont le prix est le plus bas possible. Cette stratégie d'allocation n'est disponible que pour les ressources de calcul d'instance Spot.

Note

Nous vous recommandons d'utiliser SPOT_PRICE_CAPACITY_OPTIMIZED plutôt que SPOT_CAPACITY_OPTIMIZED dans la plupart des cas.

Les BEST_FIT stratégies BEST_FIT_PROGRESSIVE et utilisent des instances à la demande ou ponctuelles, et les SPOT_PRICE_CAPACITY_OPTIMIZED stratégies SPOT_CAPACITY_OPTIMIZED et utilisent des instances ponctuelles. Toutefois, il AWS Batch se peut que vous deviez le dépasser maxvCpus pour répondre à vos besoins en matière de capacité. Dans ce cas, AWS Batch ne maxvCpus dépassez jamais plus d'une instance.

Gestion de la mémoire des ressources informatiques

Lorsque l'agent de conteneur Amazon ECS enregistre une ressource de calcul dans un environnement informatique, il doit déterminer la quantité de mémoire que la ressource de calcul peut réserver pour vos tâches. En raison de la surcharge de mémoire de la plate-forme et de la mémoire occupée par le noyau du système, ce nombre est différent de la quantité de mémoire installée pour les EC2 instances Amazon. Par exemple, une instance `m4.large` a 8 Gio de mémoire installée. Cependant, cela ne se traduit pas toujours par exactement 8192 MiB de mémoire disponible pour les tâches lorsque la ressource de calcul est enregistrée.

Supposons que vous spécifiez 8 192 Mo pour la tâche et qu'aucune de vos ressources de calcul ne dispose de 8 192 Mo ou plus de mémoire disponible pour répondre à cette exigence. La tâche ne peut alors pas être placée dans votre environnement informatique. Si vous utilisez un environnement informatique géré, vous AWS Batch devez lancer un type d'instance plus important pour répondre à la demande.

L'AMI de ressources de AWS Batch calcul par défaut réserve également 32 Mo de mémoire à l'agent de conteneur Amazon ECS et à d'autres processus système critiques. Cette mémoire n'est pas disponible pour l'allocation des tâches. Pour de plus amples informations, veuillez consulter [Mémoire système de réserve](#).

L'agent de conteneur Amazon ECS utilise la fonction `ReadMemInfo()` Docker pour connaître la mémoire totale disponible pour le système d'exploitation. Linux fournit des utilitaires de ligne de commande pour déterminer la mémoire totale.

Exemple- Déterminer la mémoire totale Linux

La `free` commande renvoie la mémoire totale reconnue par le système d'exploitation.

```
$ free -b
```

Voici un exemple de sortie pour une `m4.large` instance qui exécute l'AMI Amazon Linux optimisée pour Amazon ECS.

	total	used	free	shared	buffers	cached
Mem:	8373026816	348180480	8024846336	90112	25534464	205418496
-/+ buffers/cache:		117227520	8255799296			

Cette instance dispose de 8373026816 octets de mémoire totale. Cela signifie que 7985 MiB sont disponibles pour les tâches.

Rubriques

- [Mémoire système de réserve](#)
- [Tutoriel : Afficher la mémoire des ressources de calcul](#)
- [Considérations relatives à la mémoire et au vCPU pour AWS Batch Amazon EKS](#)

Mémoire système de réserve

Si vos tâches occupent toute la mémoire d'une ressource informatique, il est possible que celles-ci soient confrontées à des processus critiques du système en termes de mémoire et qu'elles soient à l'origine d'une défaillance du système. L'agent de conteneur Amazon ECS fournit une variable de configuration appelée `ECS_RESERVED_MEMORY`. Vous pouvez utiliser cette variable de configuration pour supprimer un certain nombre de MiB de mémoire du pool alloué à vos tâches. Cela réserve de manière effective cette quantité de mémoire pour les processus système critiques.

L'AMI de ressources de AWS Batch calcul par défaut réserve 32 MiB de mémoire à l'agent de conteneur Amazon ECS et aux autres processus critiques du système. Nous recommandons de réserver une mémoire tampon de 5 % pour l'agent de conteneur Amazon ECS et les autres processus critiques du système.

Tutoriel : Afficher la mémoire des ressources de calcul

Vous pouvez voir la quantité de mémoire enregistrée par une ressource de calcul dans la console Amazon ECS ou lors de l'opération [DescribeContainerInstances](#) d'API. Si vous essayez de maximiser l'utilisation de vos ressources en fournissant à vos tâches autant de mémoire que possible pour un type d'instance particulier, vous pouvez observer la mémoire disponible pour cette ressource de calcul, puis attribuer cette quantité de mémoire à vos tâches.

Pour afficher la mémoire des ressources de calcul

1. Ouvrez la console à la <https://console.aws.amazon.com/ecs/version 2>.
2. Choisissez Clusters, puis choisissez le cluster qui héberge vos ressources de calcul à afficher.

Le nom du cluster pour votre environnement de calcul commence par le nom de votre environnement de calcul.

3. Choisissez Infrastructure.
4. Sous Instances de conteneur, choisissez l'instance de conteneur.

5. La section Ressources et réseau indique la mémoire enregistrée et disponible pour la ressource de calcul.

La valeur de la capacité de mémoire totale correspond à ce que la ressource de calcul a enregistrée auprès d'Amazon ECS lors de son premier lancement, et la valeur de mémoire disponible correspond à ce qui n'a pas encore été alloué aux tâches.

Considérations relatives à la mémoire et au vCPU pour AWS Batch Amazon EKS

AWS Batch Sur Amazon EKS, vous pouvez spécifier les ressources mises à la disposition d'un conteneur. Par exemple, vous pouvez spécifier `requests` des `limits` valeurs pour le vCPU et les ressources de mémoire.

Les contraintes suivantes s'appliquent à la spécification des ressources vCPU :

- Au moins un `vCPU requests` ou une `limits` valeur doivent être spécifiés.
- Une unité vCPU est équivalente à un cœur physique ou virtuel.
- La valeur du vCPU doit être saisie en nombres entiers ou par incréments de 0,25.
- La plus petite valeur de vCPU valide est 0,25.
- Si les deux sont spécifiés, la `requests` valeur doit être inférieure ou égale à la `limits` valeur. De cette façon, vous pouvez configurer des configurations de vCPU souples et matérielles.
- Les valeurs du vCPU ne peuvent pas être spécifiées sous forme MilliCPU. Par exemple, 100m n'est pas une valeur valide.
- AWS Batch utilise la `requests` valeur pour dimensionner les décisions. Si aucune `requests` valeur n'est spécifiée, elle est copiée dans la `requests` valeur. `limits`

Les contraintes suivantes s'appliquent à la spécification des ressources de mémoire :

- Au moins une `mémoire requests` ou une `limits` valeur doit être spécifiée.
- Les valeurs de mémoire doivent être comprises mebibytes entre (MiBs).
- Si les deux sont spécifiés, la `requests` valeur doit être égale à la `limits` valeur.
- AWS Batch utilise la `requests` valeur pour dimensionner les décisions. Si aucune `requests` valeur n'est spécifiée, elle est copiée dans la `requests` valeur. `limits`

Les contraintes suivantes s'appliquent à la spécification des ressources GPU :

- Si les deux sont spécifiés, la `requests` valeur doit être égale à la `limits` valeur.
- AWS Batch utilise la `requests` valeur pour dimensionner les décisions. Si aucune `requests` valeur n'est spécifiée, elle est copiée dans la `requests` valeur. `limits`

Exemple : définitions de tâches

La section suivante de AWS Batch la définition de tâche Amazon EKS configure les partages de vCPU souples. Cela permet AWS Batch à Amazon EKS d'utiliser toute la capacité du vCPU pour le type d'instance. Toutefois, si d'autres tâches sont en cours d'exécution, une valeur maximale de 2 v est attribuée à la tâche CPUs. La mémoire est limitée à 2 Go.

```
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": ["sleep", "60"],
          "resources": {
            "requests": {
              "cpu": "2",
              "memory": "2048Mi"
            }
          }
        }
      ]
    }
  }
}
```

La définition de tâche suivante AWS Batch sur Amazon EKS a une `request` valeur de 1 et alloue un maximum de 4 v CPUs à la tâche.

```
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
```

```

    "podProperties": {
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": ["sleep", "60"],
          "resources": {
            "requests": {
              "cpu": "1"
            },
            "limits": {
              "cpu": "4",
              "memory": "2048Mi"
            }
          }
        }
      ]
    }
  }
}

```

La définition AWS Batch de tâche Amazon EKS ci-dessous définit une valeur de vCPU 1 et une `limits` valeur de mémoire `limits` de 1 Go.

```

{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": ["sleep", "60"],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi"
            }
          }
        }
      ]
    }
  }
}

```

Lorsque vous AWS Batch traduisez une tâche AWS Batch sur Amazon EKS en un module Amazon EKS, AWS Batch copie la `limits` valeur dans la `requests` valeur. C'est le cas si `requests` aucune valeur n'est spécifiée. Lorsque vous soumettez l'exemple de définition de tâche précédent, le module spec est le suivant.

```
apiVersion: v1
kind: Pod
...
spec:
  ...
  containers:
    - command:
      - sleep
      - 60
      image: public.ecr.aws/amazonlinux/amazonlinux:2
      resources:
        limits:
          cpu: 1
          memory: 1024Mi
        requests:
          cpu: 1
          memory: 1024Mi
      ...
```

Réservations de processeur et de mémoire pour les nœuds

AWS Batch repose sur la logique par défaut du `bootstrap.sh` fichier pour les réservations de vCPU et de mémoire. Pour plus d'informations sur le `bootstrap.sh` fichier, consultez le fichier [bootstrap.sh](#). Lorsque vous dimensionnez vos ressources de vCPU et de mémoire, considérez les exemples suivants.

Note

Si aucune instance n'est en cours d'exécution, les réservations de vCPU et de mémoire peuvent initialement affecter la logique de AWS Batch dimensionnement et la prise de décision. Une fois les instances en cours d'exécution, AWS Batch ajuste les allocations initiales.

Exemple : réservation du processeur d'un nœud

La valeur de réservation du processeur est calculée en millicores en utilisant le nombre total de v CPUs disponibles pour l'instance.

Numéro de vCPU	Pourcentage réservé
1	6 %
2	1 %
3-4	0.5%
4 et plus	0,25 %

Sur la base des valeurs précédentes, ce qui suit est vrai :

- La valeur de réservation du processeur pour une `c5.large` instance de 2 V CPUs est de 70 m. Ceci est calculé de la manière suivante : $(1 \times 60) + (1 \times 10) = 70$ m.
- La valeur de réservation du processeur pour une `c5.24xlarge` instance de 96 v CPUs est de 310 m. Ceci est calculé de la manière suivante : $(1 \times 60) + (1 \times 10) + (2 \times 5) + (92 \times 2,5) = 310$ m.

Dans cet exemple, 1930 unités de vCPU millicore (calculées entre 2000 et 70) sont disponibles pour exécuter des tâches sur une instance. `c5.large` Supposons que votre tâche nécessite 2 (2 x 1 000 m) unités de vCPU, mais qu'elle ne convient pas à une seule instance. `c5.large` Cependant, une tâche qui nécessite des unités de 1.75 vCPU convient.

Exemple : réservation de mémoire d'un nœud

La valeur de réservation de mémoire est calculée en mégaoctets à l'aide des méthodes suivantes :

- Capacité de l'instance en mégaoctets. Par exemple, une instance de 8 Go vaut 7 748MiB.
- La `kubeReserved` valeur. La `kubeReserved` valeur est la quantité de mémoire à réserver aux démons du système. La `kubeReserved` valeur est calculée de la manière suivante : $((11 \times \text{nombre maximum de pods pris en charge par le type d'instance}) + 255)$. Pour plus d'informations sur le nombre maximal de pods pris en charge par un type d'instance, consultez [eni-max-pods.txt](#)
- La `HardEvictionLimit` valeur. Lorsque la mémoire disponible tombe en dessous de `HardEvictionLimit` cette valeur, l'instance tente d'expulser les pods.

La formule pour calculer la mémoire allouable est la suivante : $(instance\_capacity\_in\_MiB) - (11 * (maximum\_number\_of\_pods)) - 255 - (HardEvictionLimit\ value.)$.

Une `c5.large` instance prend en charge jusqu'à 29 pods. Pour une `c5.large` instance de 8 Go d'une `HardEvictionLimit` valeur de 100 Mo, la mémoire allouable est de 7074. MiB Ceci est calculé de la manière suivante : $(7748 - (11 * 29) - 255 - 100) = 7074$ MiB. Dans cet exemple, une MiB tâche de 8 192 ne convient pas à cette instance même s'il s'agit d'une instance 8 gibibyte (GiB).

DaemonSets

Lorsque vous l'utilisez `DaemonSets`, tenez compte des points suivants :

- Si aucune AWS Batch instance Amazon EKS n'est en cours d'exécution, cela `DaemonSets` peut initialement affecter la logique de AWS Batch dimensionnement et la prise de décision. AWS Batch alloue initialement 0,5 unité de vCPU et 500 MiB comme prévu. `DaemonSets` Une fois les instances en cours d'exécution, AWS Batch ajuste les allocations initiales.
- Si a `DaemonSet` définit des limites de vCPU ou de mémoire, les tâches AWS Batch Amazon EKS disposent de moins de ressources. Nous vous recommandons de limiter autant `DaemonSets` que possible le nombre de AWS Batch tâches assignées aux tâches.

Environnements informatiques Fargate

Fargate est une technologie que vous pouvez utiliser pour [exécuter](#) des conteneurs sans avoir AWS Batch à gérer des serveurs ou des clusters d'instances Amazon EC2. Avec Fargate, vous n'avez plus besoin d'allouer, de configurer ou de mettre à l'échelle des clusters de machines virtuelles pour exécuter des conteneurs. Vous n'avez plus à choisir de types de serveurs, décider quand mettre à l'échelle vos clusters ni optimiser les packs de clusters.

Lorsque vous exécutez vos tâches avec les ressources Fargate, vous empaquetez votre application dans des conteneurs, vous spécifiez les exigences en matière de processeur et de mémoire, vous définissez les politiques réseau et IAM, et vous lancez l'application. Chaque tâche Fargate possède sa propre limite d'isolation et ne partage pas le noyau sous-jacent, les ressources du processeur, les ressources de mémoire ou l'interface elastic network avec une autre tâche.

Rubriques

- [Quand utiliser Fargate](#)
- [Définitions de poste sur Fargate](#)
- [Files d'attente pour les emplois sur Fargate](#)

- [Environnements informatiques sur Fargate](#)

Quand utiliser Fargate

Nous recommandons d'utiliser Fargate dans la plupart des scénarios. Fargate lance et adapte le calcul pour qu'il corresponde étroitement aux besoins en ressources que vous spécifiez pour le conteneur. Avec Fargate, vous n'avez pas besoin de surapprovisionner ou de payer pour des serveurs supplémentaires. Vous n'avez pas non plus à vous soucier des spécificités des paramètres liés à l'infrastructure, tels que le type d'instance. Lorsque l'environnement informatique doit être étendu, les tâches exécutées sur les ressources Fargate peuvent démarrer plus rapidement. Généralement, le lancement d'une nouvelle instance Amazon EC2 prend quelques minutes. Toutefois, les tâches exécutées sur Fargate peuvent être provisionnées en 30 secondes environ. Le temps exact requis dépend de plusieurs facteurs, notamment de la taille de l'image du conteneur et du nombre de tâches.

Toutefois, nous vous recommandons d'utiliser Amazon EC2 si vos tâches nécessitent l'un des éléments suivants :

- Plus de 16 V CPUs
- Plus de 120 Gíbiots (GiB) de mémoire
- UNE CARTE GRAPHIQUE
- Une Amazon Machine Image (AMI) personnalisée
- N'importe lequel des paramètres [LinuxParameters](#)

Si vous avez un grand nombre de jobs, nous vous recommandons d'utiliser l'infrastructure Amazon EC2. Par exemple, si le nombre de tâches exécutées simultanément dépasse les limites de régulation de Fargate. En effet, avec EC2, les tâches peuvent être envoyées plus rapidement aux ressources EC2 qu'aux ressources Fargate. En outre, un plus grand nombre de tâches peuvent être exécutées simultanément lorsque vous utilisez EC2. Pour plus d'informations, consultez la section Quotas [de service Fargate dans le manuel Amazon Elastic Container Service Developer Guide](#).

Définitions de poste sur Fargate

AWS Batch les tâches activées AWS Fargate ne prennent pas en charge tous les paramètres de définition de tâches disponibles. Certains paramètres ne sont pas du tout pris en charge, tandis que d'autres se comportent différemment pour les tâches Fargate.

La liste suivante décrit les paramètres de définition de tâche qui ne sont pas valides ou qui sont restreints dans les tâches Fargate.

platformCapabilities

Doit être spécifié comme FARGATE.

```
"platformCapabilities": [ "FARGATE" ]
```

type

Doit être spécifié comme container.

```
"type": "container"
```

Paramètres dans containerProperties

executionRoleArn

Doit être spécifié pour les tâches exécutées sur les ressources Fargate. Pour plus d'informations, consultez [Rôles IAM pour les tâches](#) dans le Guide du développeur Amazon Elastic Container Service.

```
"executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole"
```

fargatePlatformConfiguration

(Facultatif, uniquement pour les définitions de tâches Fargate). Spécifie la version de la plateforme Fargate, LATEST ou pour une version récente de la plateforme. Les valeurs possibles pour platformVersion sont 1.3.0, 1.4.0, et LATEST (par défaut).

```
"fargatePlatformConfiguration": { "platformVersion": "1.4.0" }
```

instanceType, ulimits

Non applicable aux tâches exécutées sur les ressources Fargate.

memory, vcpus

Ces paramètres doivent être spécifiés dans resourceRequirements

privileged

Soit ne spécifiez pas ce paramètre, soit spécifiez-le false.

```
"privileged": false
```

resourceRequirements

Les exigences en matière de mémoire et de vCPU doivent être spécifiées à l'aide de valeurs [prises en charge](#). Les ressources GPU ne sont pas prises en charge pour les tâches exécutées sur des ressources Fargate.

Si vous utilisez la surveillance du temps GuardDuty d'exécution, l'agent de GuardDuty sécurité est légèrement surchargé en mémoire. La limite de mémoire doit donc inclure la taille de l'agent GuardDuty de sécurité. Pour plus d'informations sur les limites de mémoire des agents de GuardDuty sécurité, consultez la section [Limites de processeur et de mémoire](#) dans le Guide de GuardDuty l'utilisateur. Pour plus d'informations sur les meilleures pratiques, consultez la section [Comment corriger les erreurs liées au manque de mémoire sur mes tâches Fargate après avoir activé la surveillance](#) du temps d'exécution dans le manuel Amazon ECS Developer Guide.

```
"resourceRequirements": [  
  {"type": "MEMORY", "value": "512"},  
  {"type": "VCPU", "value": "0.25"}  
]
```

Paramètres dans linuxParameters

devices, maxSwap, sharedMemorySize, swappiness, tmpfs

Non applicable aux tâches exécutées sur les ressources Fargate.

Paramètres dans logConfiguration

logDriver

Seuls awslogs et splunk sont pris en charge. Pour de plus amples informations, veuillez consulter [Utiliser le pilote de journal awslogs](#).

Membres en networkConfiguration

assignPublicIp

Si le sous-réseau privé n'est pas connecté à une passerelle NAT pour envoyer du trafic vers Internet, « » [assignPublicIp](#) doit être « ENABLED ». Pour de plus amples informations, veuillez consulter [AWS Batch Rôle d'exécution IAM](#).

Files d'attente pour les emplois sur Fargate

AWS Batch les files d'attente de travail activées AWS Fargate sont essentiellement inchangées. La seule restriction est que les environnements informatiques répertoriés dans `computeEnvironmentOrder` doivent tous être des environnements FARGATE informatiques Fargate (ou). FARGATE_SPOT Les environnements informatiques EC2 et Fargate ne peuvent pas être combinés.

Environnements informatiques sur Fargate

AWS Batch les environnements de calcul AWS Fargate activés ne prennent pas en charge tous les paramètres d'environnement de calcul disponibles. Certains paramètres ne sont pas du tout pris en charge. D'autres ont des exigences spécifiques pour Fargate.

La liste suivante décrit les paramètres d'environnement de calcul non valides ou restreints dans les tâches Fargate.

type

Ce paramètre doit être `MANAGED`.

```
"type": "MANAGED"
```

Paramètres de l'`computeResources` objet

`allocationStrategy`, `bidPercentage`, `desiredvCpus`, `imageId`, `instanceTypes`, `ec2Configuration`, `ec2KeyPair`, `instanceRole`, `launchTemplate`, `minvCpus`, `placementGroup`, `spotIamFleetRole`

Ils ne s'appliquent pas aux environnements informatiques Fargate et ne peuvent pas être fournis.

subnets

Si aucune passerelle NAT n'est attachée aux sous-réseaux répertoriés dans ce paramètre, le `assignPublicIp` paramètre de la définition de tâche doit être défini sur `ENABLED`

tags

Cela ne s'applique pas aux environnements informatiques Fargate et ne peut pas être fourni. Pour spécifier des balises pour les environnements de calcul Fargate, utilisez `tags` le paramètre qui ne figure pas dans l'objet. `computeResources`

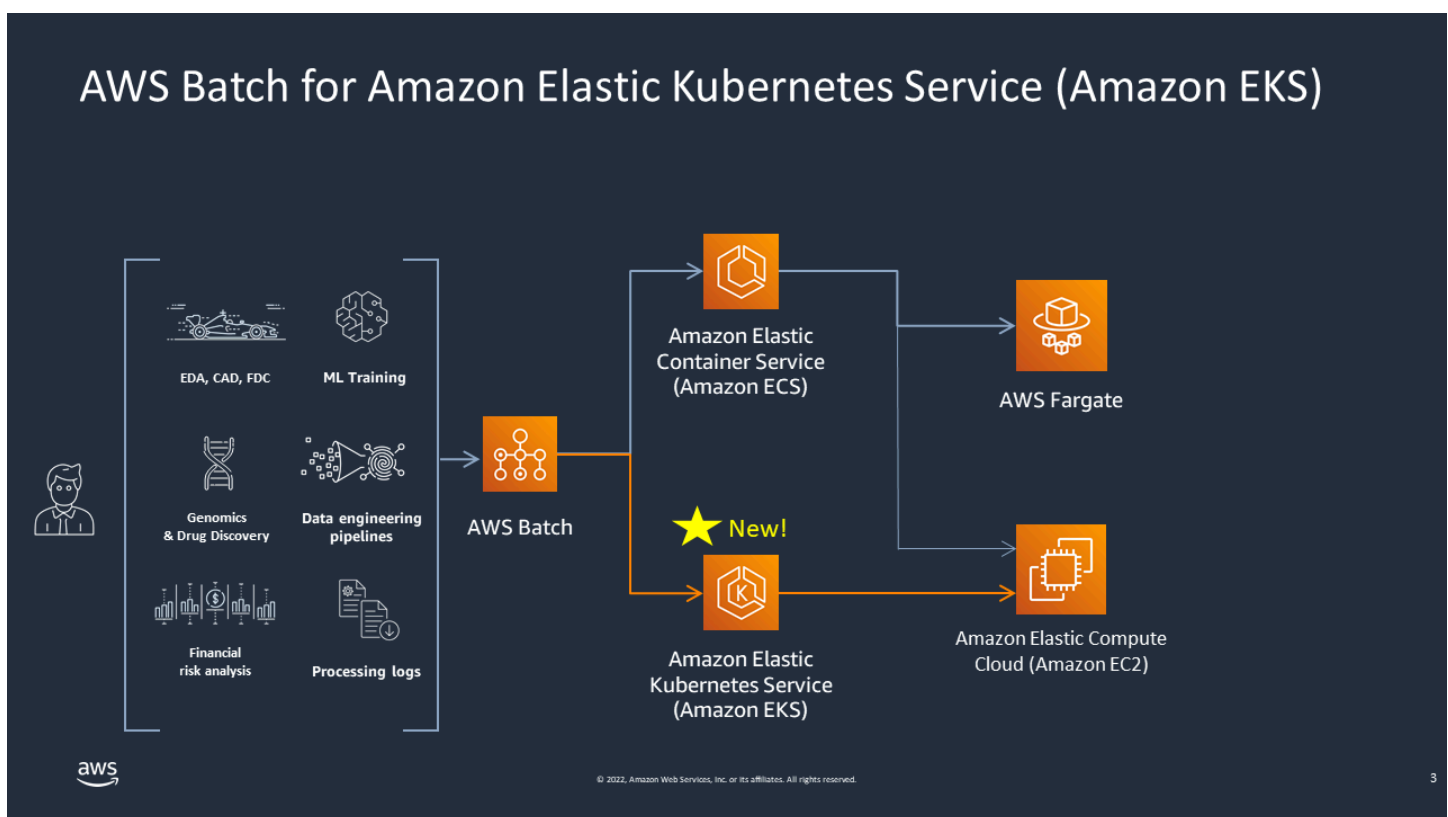
type

Ce doit être FARGATE ou FARGATE_SPOT.

```
"type": "FARGATE_SPOT"
```

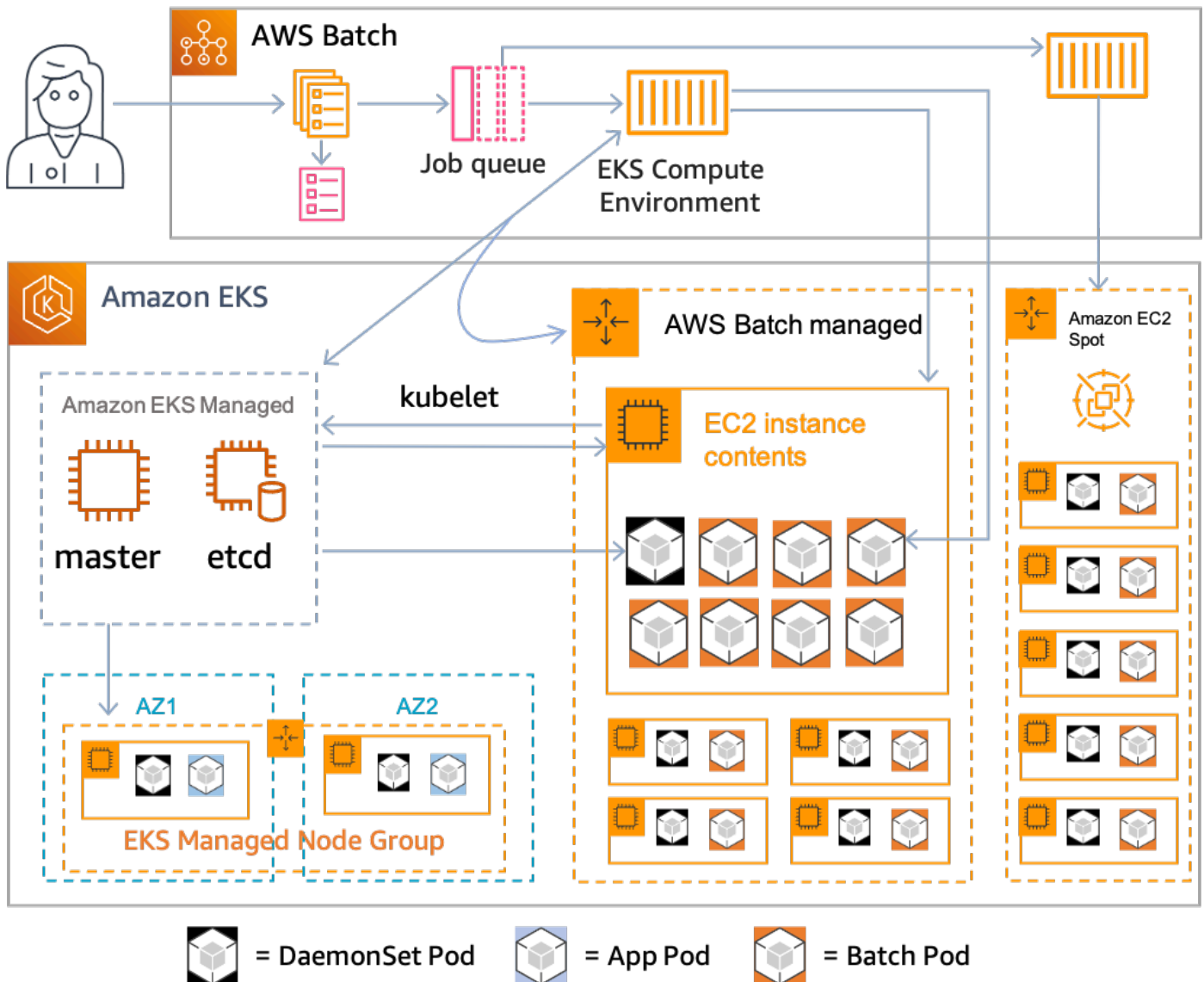
Environnements informatiques Amazon EKS

[Commencer à utiliser AWS Batch sur Amazon EKS](#) fournit un petit guide pour créer des environnements informatiques EKS. Cette section fournit plus de détails sur les environnements informatiques Amazon EKS.



AWS Batch simplifie vos charges de travail par lots sur les clusters Amazon EKS en fournissant des fonctionnalités de gestion par lots. Cela inclut la mise en file d'attente, le suivi des dépendances, la gestion des nouvelles tentatives et des priorités des tâches, la gestion des pods et le dimensionnement des nœuds. AWS Batch peut gérer plusieurs zones de disponibilité et plusieurs types et tailles d'instances Amazon EC2. AWS Batch intègre plusieurs des meilleures pratiques Amazon EC2 Spot pour exécuter vos charges de travail de manière tolérante aux pannes, ce qui

permet de réduire les interruptions. Vous pouvez l'utiliser AWS Batch pour exécuter une poignée de tâches du jour au lendemain ou des millions de tâches critiques en toute confiance.



AWS Batch est un service géré qui orchestre les charges de travail par lots dans vos Kubernetes clusters gérés par Amazon Elastic Kubernetes Service (Amazon EKS). AWS Batch effectue cette orchestration en dehors de vos clusters à l'aide d'un modèle de « superposition ». Comme il s'agit d'un service géré, il n'y a aucun Kubernetes composant (par exemple, des opérateurs ou des ressources personnalisées) à installer ou à gérer dans votre cluster. AWS Batch nécessite uniquement que votre cluster soit configuré avec des contrôles d'accès basés sur les rôles (RBAC) qui permettent à AWS Batch de communiquer avec le serveur d'API de Kubernetes. AWS Batch utilise des appels d'API de Kubernetes pour créer, surveiller et supprimer des pods et des nœuds.

AWS Batch dispose d'une logique de dimensionnement intégrée pour dimensionner Kubernetes les nœuds en fonction de la charge de la file d'attente des tâches, avec des optimisations en termes d'allocation de capacité de travail. Lorsque la file d'attente des tâches est vide AWS Batch , réduisez le nombre de nœuds à la capacité minimale que vous avez définie, qui est nulle par défaut. AWS Batch gère le cycle de vie complet de ces nœuds et les décore d'étiquettes et de teintures. De cette façon, les autres Kubernetes charges de travail ne sont pas placées sur les nœuds gérés par AWS Batch. Les exceptions à cette règle sont DaemonSets celles qui peuvent cibler AWS Batch les nœuds pour fournir la surveillance et les autres fonctionnalités nécessaires à la bonne exécution des tâches. De plus, AWS Batch n'exécute pas de tâches, en particulier de pods, sur les nœuds de votre cluster qu'il ne gère pas. Ainsi, vous pouvez utiliser une logique et des services de dimensionnement distincts pour les autres applications du cluster.

Pour soumettre des offres d'emploi AWS Batch, vous interagissez directement avec l' AWS Batch API. AWS Batch traduit les tâches en podspecs puis crée les demandes pour placer des pods sur des nœuds gérés par AWS Batch votre cluster Amazon EKS. Vous pouvez utiliser des outils tels que l'affichage `kubectl` des pods et des nœuds en cours d'exécution. Lorsqu'un pod a terminé son exécution, il AWS Batch supprime le pod qu'il a créé afin de réduire la charge sur le Kubernetes système.

Vous pouvez commencer en connectant un cluster Amazon EKS valide à AWS Batch. Attachez-y ensuite une file d'attente de AWS Batch tâches et enregistrez une définition de tâche Amazon EKS à l'aide d'attributs podspec équivalents. Enfin, soumettez les tâches à l'aide de l'opération d'[SubmitJob](#)API faisant référence à la définition de tâche. Pour de plus amples informations, veuillez consulter [Commencer à utiliser AWS Batch sur Amazon EKS](#).

Amazon EKS

Rubriques

- [AMI par défaut Amazon EKS](#)
- [Environnements AMI mixtes](#)
- [Versions de Kubernetes prises en charge](#)
- [Mettre à jour la Kubernetes version de l'environnement de calcul](#)
- [Responsabilité partagée des Kubernetes nœuds](#)
- [Exécuter un DaemonSet sur des nœuds AWS Batch gérés](#)
- [Personnalisez les modèles de lancement Amazon EKS](#)
- [Comment passer d'EKS AL2 à EKS AL2023](#)

AMI par défaut Amazon EKS

Lorsque vous créez un environnement informatique Amazon EKS, il n'est pas nécessaire de spécifier une Amazon Machine Image (AMI). AWS Batch sélectionne une AMI optimisée pour Amazon EKS en fonction de la Kubernetes version et du type d'instance spécifiés dans votre [CreateComputeEnvironment](#) demande. En général, nous vous recommandons d'utiliser la sélection d'AMI par défaut. Pour plus d'informations sur Amazon EKS optimized AMIs, consultez [Amazon Linux optimisé pour Amazon EKS AMIs](#) dans le guide de l'utilisateur Amazon EKS.

Important

Amazon Linux 2023 AMIs est activé par défaut AWS Batch pour Amazon EKS. AWS mettra fin au support pour Amazon EKS, AL2 optimisé et AL2 accéléré AMIs, à compter du 26 novembre 25. Vous pouvez continuer à utiliser Amazon Linux 2 optimisé pour Amazon EKS AWS Batch fourni AMIs sur vos environnements informatiques Amazon EKS au-delà de la end-of-support date du 26/11/25, mais ces environnements de calcul ne recevront plus de nouvelles mises à jour logicielles, de correctifs de sécurité ou de corrections de bogues de la part de ces environnements informatiques. AWS Pour plus d'informations sur la mise à niveau de AL2 vers AL2023, consultez [Comment passer d'EKS AL2 à EKS AL2023](#) le guide de AWS Batch l'utilisateur.

Exécutez la commande suivante pour voir quel type d'AMI a AWS Batch été sélectionné pour votre environnement de calcul Amazon EKS. L'exemple suivant est un type d'instance non GPU.

```
# compute CE example: indicates Batch has chosen the AL2 x86 or ARM EKS 1.32 AMI,
# depending on instance types
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1 \
    | jq '.computeEnvironments[].computeResources.ec2Configuration'
[
  {
    "imageType": "EKS_AL2",
    "imageKubernetesVersion": "1.32"
  }
]
```

L'exemple suivant est un type d'instance de GPU.

```
# GPU CE example: indicates Batch has choosen the AL2 x86 EKS Accelerated 1.32 AMI
```

```
$ aws batch describe-compute-environments --compute-environments My-Eks-GPU-CE \
  | jq '.computeEnvironments[].computeResources.ec2Configuration'

[
  {
    "imageType": "EKS_AL2_NVIDIA",
    "imageKubernetesVersion": "1.32"
  }
]
```

Environnements AMI mixtes

Vous pouvez utiliser des remplacements de modèles de lancement pour créer des environnements informatiques avec Amazon Linux 2 (AL2) et Amazon Linux 2023 (AL2023) AMIs. Cela est utile pour utiliser différentes AMIs architectures ou pendant les périodes de migration lors de la transition de AL2 vers AL2023.

Note

AWS mettra fin au support pour Amazon EKS, AL2 optimisé et AL2 accéléré AMIs, à compter du 26 novembre 25. Bien que vous puissiez continuer à utiliser Amazon Linux 2 optimisé pour Amazon EKS AWS Batch fourni AMIs sur vos environnements informatiques Amazon EKS au-delà de la end-of-support date du 26/11/25, ces environnements de calcul ne recevront plus de nouvelles mises à jour logicielles, de correctifs de sécurité ou de corrections de bogues de la part de ces environnements informatiques. AWS Les environnements AMI mixtes peuvent être utiles pendant la période de transition, car ils vous permettent de migrer progressivement les charges de travail AL2023 tout en maintenant la compatibilité avec les charges de travail AL2 existantes.

Exemple de configuration utilisant les deux types d'AMI :

```
{
  "computeResources": {
    "launchTemplate": {
      "launchTemplateId": "TemplateId",
      "version": "1",
      "userDataType": "EKS_BOOTSTRAP_SH",
      "overrides": [
        {
          "instanceType": "c5.large",
```

```
        "imageId": "ami-al2-custom",
        "userDataType": "EKS_BOOTSTRAP_SH"
    },
    {
        "instanceType": "c6a.large",
        "imageId": "ami-al2023-custom",
        "userDataType": "EKS_NODEADM"
    }
]
},
"instanceTypes": ["c5.large", "c6a.large"]
}
```

Versions de Kubernetes prises en charge

AWS Batch sur Amazon, EKS prend actuellement en charge les Kubernetes versions suivantes :

- 1.34
- 1.33
- 1.32
- 1.31
- 1.30
- 1.29

Un message d'erreur semblable au suivant peut s'afficher lorsque vous utilisez l'opération `CreateComputeEnvironmentAPI` ou l'opération `UpdateComputeEnvironmentAPI` pour créer ou mettre à jour un environnement de calcul. Ce problème se produit si vous spécifiez une Kubernetes version non prise en charge dans `EC2Configuration`.

At least one imageKubernetesVersion in EC2Configuration is not supported.

Pour résoudre ce problème, supprimez l'environnement informatique, puis recréez-le avec une Kubernetes version prise en charge.

Vous pouvez effectuer une mise à niveau de version mineure sur votre cluster Amazon EKS. Par exemple, vous pouvez mettre à niveau le cluster de 1.xx à 1.yy même si la version mineure n'est pas prise en charge.

Toutefois, l'état de l'environnement de calcul peut changer `INVALID` après une mise à jour majeure de la version. Par exemple, si vous effectuez une mise à niveau d'une version majeure de `1.xx` vers `2.yy`. Si la version majeure n'est pas prise en charge par AWS Batch, un message d'erreur semblable au suivant s'affiche.

```
reason=CLIENT_ERROR - ... EKS Cluster version [2.yy] is unsupported
```

Mettre à jour la Kubernetes version de l'environnement de calcul

Avec AWS Batch, vous pouvez mettre à jour la Kubernetes version d'un environnement informatique pour prendre en charge les mises à niveau du cluster Amazon EKS. La Kubernetes version d'un environnement de calcul est la version AMI Amazon EKS pour les Kubernetes nœuds qui sont AWS Batch lancés pour exécuter des tâches. Vous pouvez effectuer une mise à niveau de Kubernetes version sur leurs nœuds Amazon EKS avant ou après avoir mis à jour la version du plan de contrôle du cluster Amazon EKS. Nous vous recommandons de mettre à jour les nœuds après la mise à niveau du plan de contrôle. Pour plus d'informations, consultez la section [Mise à jour d'une Kubernetes version de cluster Amazon EKS](#) dans le guide de l'utilisateur Amazon EKS.

Pour mettre à niveau la Kubernetes version d'un environnement informatique, utilisez l'opération [UpdateComputeEnvironment](#) API.

```
$ aws batch update-compute-environment \
  --compute-environment <compute-environment-name> \
  --compute-resources \
    'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.32}]'
```

Responsabilité partagée des Kubernetes nœuds

La maintenance des environnements informatiques est une responsabilité partagée.

- Ne modifiez ou ne supprimez pas les AWS Batch nœuds, les étiquettes, les tâches, les espaces de noms, les modèles de lancement ou les groupes de mise à l'échelle automatique. N'ajoutez pas de salissures aux nœuds AWS Batch gérés. Si vous apportez l'une de ces modifications, votre environnement informatique ne peut pas être pris en charge et des défaillances, notamment des instances inactives, se produisent.
- Ne ciblez pas vos pods sur des nœuds AWS Batch gérés. Si vous ciblez vos pods vers les nœuds gérés, une mise à l'échelle interrompue et des files d'attente de tâches bloquées se produisent.

Exécutez des charges de travail qui ne sont pas utilisées AWS Batch sur des nœuds autogérés ou des groupes de nœuds gérés. Pour plus d'informations, consultez [Groupes de nœuds gérés](#) dans le Guide de l'utilisateur Amazon EKS.

- Vous pouvez cibler un DaemonSet pour l'exécuter sur des nœuds AWS Batch gérés. Pour de plus amples informations, veuillez consulter [Exécuter un DaemonSet sur des nœuds AWS Batch gérés](#).

AWS Batch ne met pas automatiquement à jour l'environnement informatique AMIs. Il est de votre responsabilité de les mettre à jour. Exécutez la commande suivante pour mettre à jour la dernière version de votre AMIs AMI.

```
$ aws batch update-compute-environment \  
  --compute-environment <compute-environment-name> \  
  --compute-resources 'updateToLatestImageVersion=true'
```

AWS Batch ne met pas automatiquement à jour la Kubernetes version. Exécutez la commande suivante pour mettre à jour la Kubernetes version de votre environnement informatique vers **1.32**.

```
$ aws batch update-compute-environment \  
  --compute-environment <compute-environment-name> \  
  --compute-resources \  
    'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.32}]'
```

Lors de la mise à jour vers une AMI ou une Kubernetes version plus récente, vous pouvez spécifier si vous souhaitez mettre fin aux tâches lors de leur mise à jour (`terminateJobsOnUpdate`) et combien de temps attendre avant qu'une instance ne soit remplacée si les tâches en cours ne se terminent pas (`jobExecutionTimeoutMinutes`.) Pour plus d'informations, consultez la section [Mettre à jour un environnement informatique dans AWS Batch](#) et la politique de mise à jour de l'infrastructure ([UpdatePolicy](#)) définie dans le fonctionnement de l'[UpdateComputeEnvironment](#) API.

Exécuter un DaemonSet sur des nœuds AWS Batch gérés

AWS Batch altère les Kubernetes nœuds AWS Batch gérés. Vous pouvez cibler un DaemonSet pour l'exécuter sur des nœuds AWS Batch gérés avec les éléments suivants `tolerations`.

```
tolerations:  
  - key: "batch.amazonaws.com/batch-node"  
    operator: "Exists"
```

Une autre façon de le faire est d'utiliser ce qui suit `tolerations`.

```
tolerations:
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"
  effect: "NoSchedule"
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"
  effect: "NoExecute"
```

Personnalisez les modèles de lancement Amazon EKS

AWS Batch sur Amazon, EKS prend en charge les modèles de lancement. Il existe des contraintes quant à ce que peut faire votre modèle de lancement.

Important

- Pour EKS AL2 AMIs, AWS Batch fonctionne `/etc/eks/bootstrap.sh`. Ne l'exécutez pas `/etc/eks/bootstrap.sh` dans votre modèle ou vos cloud-init user-data scripts de lancement. Vous pouvez ajouter des paramètres supplémentaires en plus du `--kubenet-extra-args` paramètre dans le [fichier bootstrap.sh](#). Pour ce faire, définissez la `AWS_BATCH_KUBENET_EXTRA_ARGS` variable dans le `/etc/aws-batch/batch.config` fichier. Consultez l'exemple suivant pour plus de détails.
- Pour EKS AL2023, AWS Batch utilise le [NodeConfigSpec](#) depuis EKS pour que les instances rejoignent le cluster EKS. AWS Batch se [ClusterDetails](#) renseigne [NodeConfigSpec](#) pour le cluster EKS et vous n'avez pas besoin de les spécifier.

Note

Nous vous recommandons de ne définir aucun des [NodeConfigSpec](#) paramètres suivants dans le modèle de lancement, car AWS Batch cela remplacerait vos valeurs. Pour de plus amples informations, veuillez consulter [Responsabilité partagée des Kubernetes nœuds](#).

- Taints
- Cluster Name
- apiServerEndpoint

- `certificatAuthority`
- CIDR
- Ne créez pas d'étiquettes avec le préfixe `batch.amazonaws.com/`

Note

Si le modèle de lancement est modifié après [CreateComputeEnvironment](#) son appel, [UpdateComputeEnvironment](#) doit être appelé pour évaluer la version du modèle de lancement à remplacer.

Rubriques

- [Ajouter kubelet des arguments supplémentaires](#)
- [Configuration de l'environnement d'exécution du conteneur](#)
- [Monter un volume Amazon EFS](#)
- [IPv6 soutien](#)

Ajouter **kubelet** des arguments supplémentaires

AWS Batch prend en charge l'ajout d'arguments supplémentaires à la `kubelet` commande. Pour la liste des paramètres pris en charge, consultez [kubelet](#) la Kubernetes documentation. Dans l'exemple suivant pour EKS AL2 AMIs, `--node-labels mylabel=helloworld` est ajouté à la ligne de `kubelet` commande.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=="MYBOUNDARY=="

--MYBOUNDARY--
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
mkdir -p /etc/aws-batch

echo AWS_BATCH_KUBELET_EXTRA_ARGS="\--node-labels mylabel=helloworld\" >> /etc/
aws-batch/batch.config
```

```
--==MYBOUNDARY==--
```

Pour EKS AL2023 AMIs, le format de fichier est YAML. Pour la liste des paramètres pris en charge, consultez [NodeConfigSpec](#) la Kubernetesdocumentation. Dans l'exemple suivant pour EKS AL2023 AMIs, `--node-labels mylabel=helloworld` est ajouté à la ligne de kubelet commande.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=="MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: application/node.eks.aws

apiVersion: node.eks.aws/v1alpha1
kind: NodeConfig
spec:
  kubelet:
    flags:
      - --node-labels=mylabel=helloworld

--==MYBOUNDARY==--
```

Configuration de l'environnement d'exécution du conteneur

Vous pouvez utiliser la variable d' AWS Batch `CONTAINER_RUNTIME` environnement pour configurer le runtime du conteneur sur un nœud géré. L'exemple suivant définit le runtime du conteneur sur `containerd when bootstrap.sh runs`. Pour plus d'informations, consultez [containerd](#) la Kubernetesdocumentation.

Si vous utilisez une version optimisée EKS_AL2023 ou une EKS_AL2023_NVIDIA AMI, il n'est pas nécessaire de spécifier le runtime du conteneur car seul `containerd` est pris en charge.

Note

La variable d'`CONTAINER_RUNTIME` environnement est équivalente à l'`--container-runtime` option de `bootstrap.sh`. Pour plus d'informations, consultez [Options](#) la Kubernetesdocumentation.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=="MYBOUNDARY=="
```



```

--==MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
mkdir -p /etc/aws-batch

echo CONTAINER_RUNTIME=containerd >> /etc/aws-batch/batch.config

--==MYBOUNDARY==--

```

Monter un volume Amazon EFS

Vous pouvez utiliser des modèles de lancement pour monter des volumes sur le nœud. Dans l'exemple suivant, les `runcmd` paramètres `cloud-config` packages et sont utilisés. Pour plus d'informations, consultez les [exemples de configuration du cloud](#) dans la `cloud-init` documentation.

```

MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=="MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-efs-utils

runcmd:
- file_system_id_01=fs-abcdef123
- efs_directory=/mnt/efs

- mkdir -p ${efs_directory}
- echo "${file_system_id_01}:/ ${efs_directory} efs _netdev,noresvport,tls,iam 0 0"
  >> /etc/fstab
- mount -t efs -o tls ${file_system_id_01}:/ ${efs_directory}

--==MYBOUNDARY==--

```

Pour utiliser ce volume dans le travail, il doit être ajouté dans le paramètre [EksProperties](#) à [RegisterJobDefinition](#). L'exemple suivant représente une grande partie de la définition de la tâche.

```

{
  "jobDefinitionName": "MyJobOnEks_EFS",

```

```

    "type": "container",
    "eksProperties": {
      "podProperties": {
        "containers": [
          {
            "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
            "command": ["ls", "-la", "/efs"],
            "resources": {
              "limits": {
                "cpu": "1",
                "memory": "1024Mi"
              }
            },
            "volumeMounts": [
              {
                "name": "efs-volume",
                "mountPath": "/efs"
              }
            ]
          }
        ],
        "volumes": [
          {
            "name": "efs-volume",
            "hostPath": {
              "path": "/mnt/efs"
            }
          }
        ]
      }
    }
  }
}

```

Dans le nœud, le volume Amazon EFS est monté dans le `/mnt/efs` répertoire. Dans le conteneur destiné à la tâche Amazon EKS, le volume est monté dans le `/efs` répertoire.

IPv6 soutien

AWS Batch prend en charge les clusters Amazon EKS dotés d' IPv6 adresses. Aucune personnalisation n'est requise pour le AWS Batch support. Toutefois, avant de commencer, nous vous recommandons de consulter les considérations et conditions décrites dans la section [Attribuer des IPv6 adresses aux pods et aux services](#) dans le guide de l'utilisateur Amazon EKS.

Comment passer d'EKS AL2 à EKS AL2023

Les Amazon EKS optimisés AMIs sont disponibles en deux familles basées sur Amazon Linux 2 (AL2) et Amazon Linux 2023 (AL2023). AL2023 est un système d'exploitation basé sur Linux conçu pour fournir un environnement sécurisé, stable et performant pour vos applications cloud. Pour plus d'informations sur les différences entre [Amazon Linux 2 AL2 et Amazon Linux 2023, AL2023](#) consultez le [guide de l'utilisateur Amazon EKS](#) et consultez la section [Mise à niveau d'Amazon Linux 2 vers Amazon Linux 2023](#).

Important

AWS mettra fin au support pour Amazon EKS, AL2 optimisé et AL2 accéléré AMIs, à compter du 26 novembre 2025. Nous vous recommandons de migrer les environnements informatiques AWS Batch Amazon EKS vers Amazon Linux 2023 avant le 26/11/25 afin de maintenir des performances et une sécurité optimales. Bien que vous puissiez continuer à utiliser Amazon Linux 2 optimisé pour Amazon EKS AWS Batch fourni AMIs sur vos environnements informatiques Amazon EKS au-delà de la end-of-support date du 26/11/25, ces environnements de calcul ne recevront plus de nouvelles mises à jour logicielles, de correctifs de sécurité ou de corrections de bogues de la part de ces environnements informatiques. AWS Il est de votre [responsabilité de maintenir](#) ces environnements de calcul sur l'AMI Amazon Linux 2 optimisée pour Amazon EKS par la suite end-of-life.

En fonction de la configuration de votre environnement informatique, vous pouvez utiliser l'un des chemins de mise à niveau suivants de AL2 à AL2023.

Mise à niveau à l'aide d'Ec2Configuration. ImageType

- Si vous n'utilisez pas de modèle de lancement ou si vous ne remplacez pas le modèle de lancement, modifiez [Ec2Configuration. ImageType](#) vers EKS_AL2023 ou EKS_AL2023_NVIDIA puis exécutez [UpdateComputeEnvironment](#).
- Si vous spécifiez une configuration [Ec2. ImageIdOverride](#) puis [Ec2Configuration. ImageType](#) doit correspondre au type d'AMI spécifié dans [Ec2Configuration. ImageIdOverride](#).

Si vous ne correspondez ImageIdOverride pas ImageType, le nœud ne rejoindra pas le cluster.

Mise à niveau en utilisant des modèles de lancement

- Si des arguments `kubelet` supplémentaires sont définis dans un modèle de lancement ou un remplacement de modèle de lancement, ils doivent être mis à jour selon le nouveau [format kubelet d'arguments supplémentaires](#).

Si vous ne correspondez pas au format des arguments `kubelet` supplémentaires, les arguments supplémentaires ne sont pas appliqués.

- Car AL2023 AMIs containerd est le seul environnement d'exécution de conteneur pris en charge. Il n'est pas nécessaire de spécifier l'environnement d'exécution du conteneur EKS_AL2023 dans le modèle de lancement.

Vous ne pouvez pas spécifier un environnement d'exécution de conteneur personnalisé avec EKS_AL2023.

- Si vous utilisez un modèle de lancement ou un remplacement de modèle de lancement qui spécifie une AMI basée sur EKS_AL2023, vous devez définir [UserDataType](#) sur `EKS_NODEADM`

Si l'AMI `userDataType` et ne correspondent pas, le nœud ne rejoindra pas le cluster EKS.

Environnements de service pour AWS Batch

Les environnements de service permettent AWS Batch de s'intégrer à SageMaker l'IA. Un environnement de service contient les paramètres de configuration spécifiques à l' IA nécessaires AWS Batch pour soumettre et gérer les tâches de SageMaker formation tout en fournissant des fonctionnalités AWS Batch de mise en file d'attente, de planification et de gestion des priorités.

Grâce aux environnements de service, les data scientists et les ingénieurs du machine learning peuvent soumettre des tâches de SageMaker formation prioritaires aux files d'attente pour les tâches de maintenance. Cette intégration élimine le besoin de coordination manuelle des charges de travail de machine learning, évite les dépenses excessives accidentelles et améliore l'utilisation des ressources dans les flux de travail d'apprentissage automatique de votre entreprise.

Rubriques

- [Dans quoi se trouvent les environnements de service AWS Batch](#)
- [État et cycle de vie de l'environnement de service dans AWS Batch](#)
- [Créez un environnement de service dans AWS Batch](#)
- [Mettre à jour un environnement de service dans AWS Batch](#)
- [Supprimer un environnement de service dans AWS Batch](#)

Dans quoi se trouvent les environnements de service AWS Batch

Un environnement de service est une AWS Batch ressource qui contient les paramètres de configuration nécessaires à l'intégration AWS Batch à l' SageMaker IA. Les environnements de service permettent AWS Batch de soumettre et de gérer les tâches de SageMaker formation tout en fournissant des AWS Batch fonctionnalités de gestion des files d'attente, de planification et des priorités.

Les environnements de service répondent aux défis courants auxquels sont confrontées les équipes de data science lorsqu'elles gèrent des charges de travail liées au machine learning. Organisations limitent souvent le nombre d'instances disponibles pour les modèles de formation afin d'éviter les dépenses excessives accidentelles, de respecter les contraintes budgétaires, de réduire les coûts grâce aux instances réservées ou d'utiliser des types d'instances spécifiques pour les charges de travail. Cependant, les data scientists peuvent souhaiter exécuter plus de charges de travail

simultanément que ce qui est possible avec les instances qui leur sont allouées, ce qui nécessite une coordination manuelle pour décider quelles charges de travail s'exécutent et quand.

Ce défi de coordination concerne les organisations de toutes tailles, qu'il s'agisse d'équipes ne comptant que quelques data scientists ou d'opérations à grande échelle. À mesure que les entreprises se développent, la complexité augmente, ce qui nécessite plus de temps pour gérer la coordination de la charge de travail et nécessite souvent l'intervention de l'administrateur de l'infrastructure. Ces efforts manuels font perdre du temps et réduisent l'efficacité des instances, ce qui entraîne des coûts réels pour les clients.

Grâce aux environnements de service, les data scientists et les ingénieurs ML peuvent soumettre des tâches de SageMaker formation prioritaires à des files d'attente configurables, garantissant ainsi l'exécution automatique des charges de travail sans intervention dès que les ressources sont disponibles. Cette intégration tire parti des fonctionnalités étendues AWS Batch de mise en file d'attente et de planification, permettant aux clients de personnaliser leurs politiques de mise en file d'attente et de planification en fonction des objectifs de leur organisation.

Comment les environnements de service fonctionnent avec les autres AWS Batch composants

Les environnements de service s'intègrent à d'autres AWS Batch composants pour permettre la mise en file d'attente des tâches de SageMaker formation :

- **Files d'attente de tâches** : les environnements de service sont associés aux files d'attente pour permettre à la file d'attente de traiter les tâches de service pour les SageMaker tâches de formation.
- **Tâches de service** : lorsque vous soumettez une tâche de service à une file d'attente associée à un environnement de service AWS Batch , utilisez la configuration de l'environnement pour soumettre la tâche de SageMaker formation correspondante
- **Politiques de planification** - Les environnements de service fonctionnent avec AWS Batch des politiques de planification pour hiérarchiser et gérer l'ordre d'exécution des tâches de SageMaker formation

Cette intégration vous permet de tirer parti des fonctionnalités AWS Batch éprouvées de mise en file d'attente et de planification tout en conservant les fonctionnalités complètes et la flexibilité des tâches de SageMaker formation.

Bonnes pratiques pour les environnements de service

Les environnements de service fournissent des fonctionnalités permettant de gérer les tâches de SageMaker formation à grande échelle. Le respect de ces meilleures pratiques vous permet d'optimiser les coûts, les performances et l'efficacité opérationnelle tout en évitant les problèmes de configuration courants qui peuvent avoir un impact sur vos flux de travail d'apprentissage automatique.

Lorsque vous planifiez la capacité de l'environnement de service, tenez compte des quotas et limites spécifiques qui s'appliquent à la mise en file d'attente des tâches de SageMaker formation. Chaque environnement de service possède une limite de capacité maximale exprimée en nombre d'instances, qui contrôle directement le nombre de tâches de SageMaker formation pouvant être exécutées simultanément. La compréhension de ces limites permet d'éviter la saturation des ressources et de garantir des délais d'exécution des tâches prévisibles.

La performance optimale de l'environnement de service dépend de la compréhension des caractéristiques uniques de la planification des tâches de SageMaker formation. Contrairement aux emplois conteneurisés traditionnels, les emplois de service passent par un SCHEDULED État tandis que l' SageMaker IA acquiert et fournit les instances de formation nécessaires. Cela signifie que les heures de début des tâches peuvent varier considérablement en fonction de la disponibilité des instances et de la capacité régionale.

Important

Les environnements de service ont des quotas spécifiques qui peuvent avoir un impact sur votre capacité à faire évoluer les charges de travail de SageMaker formation. Vous pouvez créer jusqu'à 50 environnements de service par compte, chaque file d'attente de tâches ne prenant en charge qu'un seul environnement de service associé. En outre, la charge utile des demandes de service pour les tâches individuelles est limitée à 10 KiB, et l'`SubmitServiceJobAPI` est limitée à 5 transactions par seconde et par compte. La compréhension de ces limites lors de la planification des capacités permet d'éviter les contraintes d'évolutivité inattendues.

Une surveillance efficace des environnements de service nécessite une attention particulière à la fois aux indicateurs AWS Batch des services fournis par l' SageMaker IA. [Les transitions d'état de poste](#) fournissent des informations précieuses sur les performances du système, en particulier le temps passé dans SCHEDULED l'état, qui indique les modèles de disponibilité des capacités. Les

environnements de service conservent leurs propres états de cycle de vie, similaires à ceux des environnements informatiques `CREATING`, en passant par `VALID`, `INVALID`, et `DELETING` des états dont l'état de fonctionnement doit être surveillé. Organisations dotées de pratiques de surveillance éprouvées suivent généralement la profondeur des files d'attente, le taux d'achèvement des tâches et les modèles d'utilisation des instances afin d'optimiser la configuration de leur environnement de service au fil du temps.

État et cycle de vie de l'environnement de service dans AWS Batch

Les environnements de service conservent des états du cycle de vie qui indiquent leur état opérationnel actuel et leur état de préparation à traiter les tâches de SageMaker formation. La compréhension de ces états vous permet de surveiller l'état de l'environnement de service, de résoudre les problèmes de configuration et de garantir un traitement fiable des tâches. Le système de gestion des états suit les modèles établis des environnements informatiques tout en répondant aux exigences uniques de l'intégration des postes de SageMaker formation.

Les états de l'environnement de service sont gérés automatiquement en AWS Batch fonction de la validation de la configuration, de la disponibilité des ressources et des contrôles de santé opérationnels. Contrairement aux environnements informatiques qui gèrent l'infrastructure physique, les environnements de service se concentrent sur la validation de la configuration et la préparation à l'intégration avec les services d' SageMaker IA. Les transitions d'état permettent de savoir si votre environnement de service est capable de soumettre et de gérer avec succès les tâches de SageMaker formation.

Définitions des états de l'environnement de service

Les environnements de service peuvent se trouver dans l'un des quatre états possibles qui indiquent leur état opérationnel actuel et leur état de préparation à traiter les tâches de SageMaker formation. Chaque état représente une phase spécifique du cycle de vie de l'environnement de service, de la création initiale à la suppression finale en passant par la préparation opérationnelle. Le tableau suivant décrit chaque état et sa signification :

State	Description
CREATING	État initial lorsque vous créez un environnement de service. Au cours de cet état, AWS Batch valide les paramètres de configura

State	Description
	tion et établit l'intégration avec les services d'SageMaker IA. L'environnement de service ne peut pas traiter les tâches et les files d'attente de tâches associées n'accepteront pas les soumissions de tâches de service. Le processus de création se termine généralement en quelques secondes pour les environnements de service correctement configurés.
VALID	État opérationnel indiquant que l'environnement de service a passé avec succès tous les contrôles de validation de configuration et qu'il est prêt à traiter les tâches de SageMaker formation. Cet état indique que la configuration de l'environnement de service est correcte, que toutes les autorisations requises sont en place et que AWS Batch vous pouvez soumettre des tâches à SageMaker AI en votre nom. Les environnements de service passent la majeure partie de leur cycle de vie opérationnel dans cet état.
INVALID	État indiquant que l'environnement de service a rencontré un problème de configuration ou d'autorisations qui l'empêche de traiter les tâches de SageMaker formation. Les files d'attente associées à des environnements de service non valides ne peuvent pas traiter les nouvelles soumissions de tâches de service tant que les problèmes sous-jacents ne sont pas résolus.

State	Description
DELETING	État qui se produit lorsque vous demandez la suppression d'un environnement de service. Dans cet état, AWS Batch garantit qu'aucune tâche de SageMaker formation active n'est associée à l'environnement et effectue les opérations de nettoyage nécessaires. Les environnements de service dans cet état ne peuvent pas traiter les nouvelles soumissions de tâches, et le processus de suppression s'achève une fois que toutes les ressources associées ont été correctement nettoyées.

Transitions d'état de l'environnement de

Les transitions d'état de l'environnement de service se produisent automatiquement en fonction des modifications de configuration, des résultats de validation et de la surveillance de l'état de fonctionnement. Le AWS Batch service surveille en permanence l'état de l'environnement de service et met à jour les états en conséquence. La compréhension de ces transitions vous permet d'anticiper le moment où les modifications de configuration prendront effet et comment résoudre les problèmes à l'origine d'états non valides.

Une fois la création et la validation réussies, les environnements de service passent de CREATING à VALID. Cette transition confirme que tous les paramètres de configuration sont corrects, que les autorisations IAM requises sont correctement configurées et que l'environnement de service peut s'intégrer avec succès aux services d' SageMaker IA. Une fois dans l'VALID état, les files d'attente de tâches associées peuvent commencer à traiter les soumissions de tâches de service.

Les environnements de service passent d'une situation VALID à une INVALID situation où la validation de la configuration échoue ou lorsque les dépendances ne sont plus disponibles. Cela peut être dû à des modifications du rôle IAM, à des modifications des limites de capacité qui violent les quotas ou à des modifications des ressources externes qui affectent le fonctionnement de l'environnement de service. Le champ Motif du statut fournit des informations spécifiques sur la cause de l'état non valide.

Les environnements de service peuvent revenir à l'INVALIDÉtat d'origine INVALID une fois les problèmes sous-jacents résolus. Cela peut impliquer la mise à jour des autorisations IAM, la correction des configurations de capacité ou le rétablissement de l'accès aux ressources requises. AWS La transition se produit généralement automatiquement une fois que l'on AWS Batch détecte que les problèmes de configuration ont été résolus.

Créez un environnement de service dans AWS Batch

Avant de pouvoir exécuter des tâches de SageMaker formation dans AWS Batch, vous devez créer un environnement de service. Vous pouvez créer un environnement de service contenant les paramètres de configuration requis AWS Batch pour intégrer les services d' SageMaker IA et soumettre des tâches de SageMaker formation en votre nom.

Conditions préalables

Avant de créer un environnement de service, assurez-vous de disposer des éléments suivants :

- Autorisations IAM : autorisations permettant de créer et de gérer des environnements de service. Pour de plus amples informations, veuillez consulter [AWS Batch Politiques, rôles et autorisations IAM](#).

Create a service environment (AWS Console)

Utilisez la AWS Batch console pour créer un environnement de service via l'interface Web.

Pour créer un environnement de service

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans le panneau de navigation, choisissez Environments (Environnements).
3. Choisissez Créer un environnement, puis sélectionnez Environnement de service.
4. Pour la configuration de l'environnement de service, choisissez SageMaker AI.
5. Dans Nom, entrez un nom unique pour votre environnement de service. Les caractères valides sont a-z, A-Z, 0-9, les tirets (-) et les traits de soulignement (_).
6. Pour Nombre maximal d'instances, entrez le nombre maximum d'instances de formation simultanées.
7. (Facultatif) Ajoutez des balises en choisissant Ajouter une balise et en saisissant des paires clé-valeur.

8. Choisissez Suivant.
9. Passez en revue les détails du nouvel environnement de service et choisissez Créer un environnement de service.

Create a service environment (AWS CLI)

Utilisez la `create-service-environment` commande pour créer un environnement de service avec la AWS CLI.

Pour créer un environnement de service

1. Créez un environnement de service avec les paramètres de base requis :

```
aws batch create-service-environment \  
  --service-environment-name my-sagemaker-service-env \  
  --service-environment-type SAGEMAKER_TRAINING \  
  --capacity-limits capacityUnit=NUM_INSTANCES,maxCapacity=10
```

2. (Facultatif) Créez un environnement de service avec des balises :

```
aws batch create-service-environment \  
  --service-environment-name my-sagemaker-service-env \  
  --service-environment-type SAGEMAKER_TRAINING \  
  --capacity-limits capacityUnit=NUM_INSTANCES,maxCapacity=10 \  
  --tags team=data-science,project=ml-training
```

3. Vérifiez que l'environnement de service a été créé avec succès :

```
aws batch describe-service-environments \  
  --service-environment my-sagemaker-service-env
```

L'environnement de service apparaît dans la liste des environnements avec un CREATING état. Lorsque la création est terminée avec succès, l'état change VALID et l'environnement de service est prêt à recevoir une file d'attente de tâches de service afin que l'environnement de service puisse commencer à traiter les tâches.

Mettre à jour un environnement de service dans AWS Batch

Vous pouvez mettre à jour un environnement de service pour modifier ses limites de capacité, son état de fonctionnement ou mettre à jour les balises de ressources. Les mises à jour de l'environnement de service vous permettent d'ajuster la capacité en fonction de l'évolution de vos exigences en matière de charge de SageMaker formation ou de modifier les paramètres opérationnels sans recréer l'environnement. Avant de mettre à jour un environnement de service, déterminez quels paramètres peuvent être modifiés et déterminez l'impact des modifications sur l'exécution des tâches.

Vous pouvez modifier les limites de capacité, l'état ou les balises d'un environnement de service.

Update a service environment (AWS Console)

Utilisez la AWS Batch console pour mettre à jour un environnement de service via l'interface Web.

Pour mettre à jour un environnement de service

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans le panneau de navigation, choisissez Environments (Environnements).
3. Choisissez l'onglet Environnement de service.
4. Choisissez l'environnement de service à mettre à jour.
5. Choisissez Actions, puis choisissez l'une des options suivantes :
 - État - Choisissez Activer ou Désactiver pour modifier l'état.
 - Limite de capacité : modifiez le nombre maximum d'instances
6. Choisissez Enregistrer les modifications pour appliquer les modifications.

L'environnement de service est mis à jour immédiatement. Vérifiez les détails de l'environnement pour vous assurer que les modifications ont été correctement appliquées. Si vous avez désactivé l'environnement de service, les files d'attente de tâches associées cesseront de traiter les nouvelles soumissions de service jusqu'à ce que vous le réactiviez.

Update a service environment (AWS CLI)

Utilisez la `update-service-environment` commande pour modifier un environnement de service à l'aide de la AWS CLI.

Pour mettre à jour les limites de capacité de l'environnement de service

1. Mettez à jour la limite de capacité pour un environnement de service :

```
aws batch update-service-environment \  
  --service-environment my-sagemaker-service-env \  
  --capacity-limits capacityUnit=NUM_INSTANCES,maxCapacity=20
```

2. Vérifiez que la mise à jour a été correctement appliquée :

```
aws batch describe-service-environments \  
  --service-environments my-sagemaker-service-env
```

Pour mettre à jour l'état de l'environnement de service

1. Désactivez un environnement de service pour arrêter le traitement des nouvelles tâches :

```
aws batch update-service-environment \  
  --service-environment my-sagemaker-service-env \  
  --state DISABLED
```

2. Réactivez un environnement de service pour reprendre le traitement :

```
aws batch update-service-environment \  
  --service-environment my-sagemaker-service-env \  
  --state ENABLED
```

Les mises à jour de l'environnement de service prennent effet immédiatement. Surveillez l'état de l'environnement de service pour vous assurer que les mises à jour sont effectuées correctement avant de soumettre de nouvelles tâches.

Supprimer un environnement de service dans AWS Batch

Vous pouvez supprimer un environnement de service lorsqu'il n'est plus nécessaire pour vos tâches de SageMaker formation. La suppression d'un environnement de service entraîne la suppression de la configuration et empêche de nouvelles soumissions de tâches. Avant de supprimer un environnement de service, assurez-vous qu'aucune tâche de SageMaker formation active n'en dépend et qu'aucune file d'attente de tâches n'est associée à l'environnement de service.

Important

La suppression de l'environnement de service est irréversible. Une fois supprimé, vous ne pouvez pas restaurer l'environnement de service ou sa configuration. Si vous avez besoin de fonctionnalités similaires à l'avenir, vous devez créer un nouvel environnement de service avec les paramètres requis. Envisagez de désactiver l'environnement de service plutôt que de le supprimer si vous devez le réactiver ultérieurement.

Note

La suppression de tous les environnements de service de votre compte ne supprime pas automatiquement le rôle lié au service créé pour l'intégration de AWS Batch l' SageMaker IA. Le rôle lié au service reste disponible pour la création d'un futur environnement de service. Si vous souhaitez supprimer le rôle lié à un service, vous devez le supprimer séparément à l'aide d'IAM après vous être assuré qu'aucun environnement de service n'existe dans votre compte.

Conditions préalables à la suppression

Avant de pouvoir supprimer un environnement de service, vous devez dissocier toute file d'attente de tâches de service, puis désactiver l'environnement de service.

Avant de supprimer un environnement de service :

- Vérifiez les tâches actives : assurez-vous qu'aucune tâche de SageMaker formation n'est actuellement en cours d'exécution dans l'environnement de service.
- Passez en revue les files d'attente de tâches : identifiez les files d'attente de tâches associées à l'environnement de service et associez la file d'attente de tâches à un autre environnement de service ou désactivez et supprimez la file d'attente de tâches.

Gestion des files d'attente de tâches : les files d'attente associées à un environnement de service supprimé peuvent toujours exister mais ne peuvent pas traiter les tâches de service. Vous devez supprimer les files d'attente de tâches inutilisées ou les associer à un autre environnement de service avant de supprimer l'environnement de service d'origine.

Delete a service environment (AWS Console)

Utilisez la AWS Batch console pour supprimer un environnement de service via l'interface Web.

Pour supprimer un environnement de service

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans le panneau de navigation, choisissez Environments (Environnements).
3. Choisissez l'onglet Environnement de service, puis choisissez un environnement de service.
4. Si l'environnement de service est activé, choisissez Actions, puis Désactiver.
5. Une fois l'environnement de service désactivé, choisissez Actions, puis Supprimer.
6. Dans la boîte de dialogue de confirmation, choisissez Confirmer.

L'environnement de service affiche un DELETING état lors de la suppression. Une fois la suppression terminée, l'environnement de service disparaît de la liste des environnements.

Delete a service environment (AWS CLI)

Utilisez la `delete-service-environment` commande pour supprimer un environnement de service à l'aide de la AWS CLI.

Pour supprimer un environnement de service

1. Vérifiez les files d'attente de tâches associées à l'environnement de service :

```
aws batch describe-job-queues
```

Si des files d'attente de tâches sont associées à l'environnement de service, vous pouvez soit [dissocier la file d'attente](#) de l'environnement de service et l'associer à un autre environnement de service, soit supprimer la file d'attente de tâches.

2. Désactivez l'environnement de service :

```
aws batch update-service-environment \
  --service-environment my-sagemaker-service-env \
  --state DISABLED
```

3. Supprimez l'environnement de service :

```
aws batch delete-service-environment \
```



```
--service-environment my-sagemaker-service-env
```

4. Surveillez le processus de suppression :

```
aws batch describe-service-environments \  
  --service-environment my-sagemaker-service-env
```

L'environnement de service passe à DELETING l'état pendant le processus de suppression. Une fois la suppression terminée, l'environnement de service n'est plus répertorié dans les opérations de description. Les files d'attente de tâches associées restent mais ne peuvent pas traiter les tâches de service tant qu'elles ne sont pas associées à un environnement de service différent.

Files d'attente pour les jobs

Les tâches sont soumises à une file d'attente où elles résident jusqu'à ce qu'elles puissent être planifiées pour être exécutées dans un environnement informatique. Un AWS compte peut comporter plusieurs files d'attente de travail. Par exemple, vous pouvez créer une file d'attente qui utilise des instances Amazon EC2 On-Demand pour les tâches hautement prioritaires et une autre file d'attente qui utilise des instances ponctuelles Amazon EC2 pour les tâches peu prioritaires. Les files d'attente de tâches ont une priorité qui est utilisée par le planificateur pour déterminer quels travaux dans quelle file d'attente doivent être évalués pour être exécutés en premier.

Rubriques

- [Création d'une file d'attente de tâches](#)
- [Afficher une file d'attente de tâches dans AWS Batch](#)
- [Supprimer une file d'attente de tâches dans AWS Batch](#)
- [Politiques de planification équitables](#)
- [Planification axée sur les ressources](#)

Création d'une file d'attente de tâches

Avant de pouvoir soumettre des tâches AWS Batch, vous devez créer une file d'attente de tâches. Lorsque vous créez une file d'attente de tâches, vous y associez un ou plusieurs environnements informatiques et vous attribuez un ordre de préférence.

Vous définissez également la priorité de la file d'attente des tâches qui détermine l'ordre dans lequel le planificateur AWS Batch place les tâches. Cela signifie que, si un environnement informatique est associé à plusieurs files d'attente de tâches, la priorité est donnée à la file d'attente de tâches la plus prioritaire.

Rubriques

- [Création d'une file d'attente de EC2 tâches Amazon](#)
- [Création d'une file d'attente de tâches Fargate](#)
- [Création d'une file d'attente de tâches Amazon EKS](#)
- [Créez une file d'attente SageMaker de tâches de formation dans AWS Batch](#)
- [Modèle de file d'attente de tâches](#)

Création d'une file d'attente de EC2 tâches Amazon

Procédez comme suit pour créer une file d'attente de tâches pour Amazon Elastic Compute Cloud (Amazon EC2).

Pour créer une file d'attente de EC2 tâches Amazon

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation, sélectionnez le Région AWS à utiliser.
3. Dans le volet de navigation, choisissez Job queues.
4. Choisissez Créer.
5. Pour le type d'orchestration, choisissez Amazon Elastic Compute Cloud (Amazon EC2).
6. Dans Nom, entrez un nom unique pour votre file d'attente de tâches. Le nom peut comporter jusqu'à 128 caractères et peut contenir des lettres majuscules et minuscules, des chiffres et des traits de soulignement (_).
7. Pour Priorité, entrez un nombre entier pour la priorité de la file d'attente des tâches. Les files d'attente de tâches ayant une priorité plus élevée sont exécutées avant les files d'attente de tâches de moindre priorité associées au même environnement informatique. La priorité est déterminée par ordre décroissant. Par exemple, une file d'attente de tâches avec une valeur de priorité 10 se voit accorder la préférence en termes de planification par rapport à une file d'attente de tâches avec une valeur de priorité 1.
8. (Facultatif) Pour la politique de planification Amazon Resource Name (ARN), choisissez une politique de planification existante.
9. Pour les environnements informatiques connectés, sélectionnez un ou plusieurs environnements informatiques dans la liste à associer à la file d'attente des tâches. Sélectionnez les environnements informatiques dans l'ordre dans lequel vous souhaitez que la file d'attente tente de placer la file d'attente des tâches. Le planificateur de tâches utilise l'ordre dans lequel vous sélectionnez les environnements de calcul pour déterminer quel environnement de calcul démarre une tâche donnée. Avant de pouvoir les associer à une file d'attente de tâches, les environnements informatiques doivent être dans VALID cet état. Vous pouvez associer jusqu'à trois environnements de calcul à une file d'attente de tâches. Si vous ne disposez pas d'un environnement informatique existant, choisissez Créer un environnement de calcul

 Note

Tous les environnements informatiques associés à une file d'attente de tâches doivent partager le même modèle de provisionnement. AWS Batch ne prend pas en charge le mélange de modèles de provisionnement dans une seule file d'attente de tâches.

10. Pour l'ordre de l'environnement de calcul, choisissez les flèches haut et bas pour configurer l'ordre souhaité.
11. Choisissez Créer une file d'attente de tâches pour terminer et créer votre file d'attente de tâches.

Création d'une file d'attente de tâches Fargate

Procédez comme suit pour créer une file d'attente de tâches pour AWS Fargate.

Pour créer une file d'attente de tâches Fargate

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation, sélectionnez le Région AWS à utiliser.
3. Dans le volet de navigation, choisissez Job queues.
4. Choisissez Créer.
5. Pour le type d'orchestration, choisissez Fargate.
6. Dans Nom, entrez un nom unique pour votre file d'attente de tâches. Le nom peut comporter jusqu'à 128 caractères et peut contenir des lettres majuscules et minuscules, des chiffres et des traits de soulignement (_).
7. Pour Priorité, entrez un nombre entier pour la priorité de la file d'attente des tâches. Les files d'attente de tâches ayant une priorité plus élevée sont exécutées avant les files d'attente de tâches de moindre priorité associées au même environnement informatique. La priorité est déterminée par ordre décroissant. Par exemple, une file d'attente de tâches avec une valeur de priorité 10 se voit accorder la préférence en termes de planification par rapport à une file d'attente de tâches avec une valeur de priorité 1.
8. (Facultatif) Pour la politique de planification Amazon Resource Name (ARN), choisissez une politique de planification existante.
9. Pour les environnements informatiques connectés, sélectionnez un ou plusieurs environnements informatiques dans la liste à associer à la file d'attente des tâches. Sélectionnez les

environnements informatiques dans l'ordre dans lequel vous souhaitez que la file d'attente tente de placer la file d'attente des tâches. Le planificateur de tâches utilise l'ordre dans lequel vous sélectionnez les environnements de calcul pour déterminer quel environnement de calcul démarre une tâche donnée. Avant de pouvoir les associer à une file d'attente de tâches, les environnements informatiques doivent être dans VALID cet état. Vous pouvez associer jusqu'à trois environnements de calcul à une file d'attente de tâches.

Note

Tous les environnements informatiques associés à une file d'attente de tâches doivent partager le même modèle de provisionnement. AWS Batch ne prend pas en charge le mélange de modèles de provisionnement dans une seule file d'attente de tâches.

10. Pour l'ordre de l'environnement de calcul, choisissez les flèches vers le haut et vers le bas pour configurer l'ordre souhaité.
11. Choisissez Créer une file d'attente de tâches pour terminer et créer votre file d'attente de tâches.

Création d'une file d'attente de tâches Amazon EKS

Procédez comme suit pour créer une file d'attente de tâches pour Amazon Elastic Kubernetes Service (Amazon EKS).

Pour créer une file d'attente de tâches Amazon EKS

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation, sélectionnez le Région AWS à utiliser.
3. Dans le volet de navigation, choisissez Job queues.
4. Choisissez Créer.
5. Pour le type d'orchestration, choisissez Amazon Elastic Kubernetes Service (Amazon EKS).
6. Dans Nom, entrez un nom unique pour votre file d'attente de tâches. Le nom peut comporter jusqu'à 128 caractères et peut contenir des lettres majuscules et minuscules, des chiffres et des traits de soulignement (_).
7. Dans Priorité, saisissez une valeur d'entier pour la priorité de la file d'attente de tâches. Les files d'attente de tâches ayant une priorité plus élevée sont exécutées avant les files d'attente de tâches de moindre priorité associées au même environnement informatique. La priorité est déterminée par ordre décroissant. Par exemple, une file d'attente de tâches avec une valeur

de priorité 10 se voit accorder la préférence en termes de planification par rapport à une file d'attente de tâches avec une valeur de priorité 1.

8. (Facultatif) Pour la politique de planification Amazon Resource Name (ARN), choisissez une politique de planification existante.
9. Pour les environnements informatiques connectés, sélectionnez un ou plusieurs environnements informatiques dans la liste à associer à la file d'attente des tâches. Sélectionnez les environnements informatiques dans l'ordre dans lequel vous souhaitez que la file d'attente tente de placer la file d'attente des tâches. Le planificateur de tâches utilise l'ordre dans lequel vous sélectionnez les environnements de calcul pour déterminer quel environnement de calcul démarre une tâche donnée. Avant de pouvoir les associer à une file d'attente de tâches, les environnements informatiques doivent être dans VALID cet état. Vous pouvez associer jusqu'à trois environnements de calcul à une file d'attente de tâches.

 Note

Tous les environnements informatiques associés à une file d'attente de tâches doivent partager le même modèle de provisionnement. AWS Batch ne prend pas en charge le mélange de modèles de provisionnement dans une seule file d'attente de tâches.

 Note

Tous les environnements informatiques associés à une file d'attente de tâches doivent partager la même architecture. AWS Batch ne prend pas en charge le mélange de types d'architecture d'environnement de calcul dans une seule file d'attente de tâches.

10. Pour l'ordre de l'environnement de calcul, choisissez les flèches vers le haut et vers le bas pour configurer l'ordre souhaité.
11. Choisissez Créer une file d'attente de tâches pour terminer et créer votre file d'attente de tâches.

Créez une file d'attente SageMaker de tâches de formation dans AWS Batch

SageMaker Les files d'attente pour les tâches de formation s'intègrent directement au service d'SageMaker intelligence artificielle pour permettre une planification des tâches sans serveur sans que vous ayez à gérer l'infrastructure informatique sous-jacente.

Prérequis

Avant de créer une file d'attente de tâches de SageMaker formation, assurez-vous d'avoir :

- Environnement de service : environnement de service qui définit les limites de capacité. Pour de plus amples informations, veuillez consulter [Créez un environnement de service dans AWS Batch](#).
- Autorisations IAM : autorisations permettant de créer et de gérer des files AWS Batch d'attente de travail et des environnements de service. Pour de plus amples informations, veuillez consulter [AWS Batch Politiques, rôles et autorisations IAM](#).

Create a SageMaker Training job queue (AWS Batch console)

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans le volet de navigation, choisissez Job queues puis Create.
3. Pour le type d'orchestration, choisissez SageMaker Entraînement.
4. Pour la configuration de la file d'attente des Job :
 - a. Dans Nom, entrez le nom de la file d'attente de Job.
 - b. pour Priorité, entrez une valeur comprise entre 0 et 1000. Une file d'attente de Job avec une priorité plus élevée est privilégiée pour les environnements de service.
 - c. (Facultatif) Pour la politique de planification Amazon Resource Name (ARN), choisissez une politique de planification existante.
 - d. Pour les environnements de service connectés, sélectionnez un environnement de service dans la liste à associer à la file d'attente des tâches.
5. (Facultatif) Pour les limites d'état des Job :
 - a. Pour Mauvaise configuration, choisissez SERVICE_ENVIRONMENT_MAX_RESOURCE et entrez la durée maximale d'exécution (secondes).
 - b. Pour Capacité, choisissez INSUFFICIENT_INSTANCE_CAPACITY et entrez la durée maximale d'exécution (secondes).
6. Choisissez Créer une file d'attente de tâches

Create a SageMaker Training job queue (AWS CLI)

Utilisez la `create-job-queue` commande pour créer une file d'attente SageMaker de tâches de formation.

L'exemple suivant crée une file d'attente de tâches de SageMaker formation de base qui utilise un environnement de service :

```
aws batch create-job-queue \  
  --job-queue-name my-sm-training-fifo-jq \  
  --job-queue-type SAGEMAKER_TRAINING \  
  --priority 1 \  
  --service-environment-order order=1,serviceEnvironment=ExampleServiceEnvironment
```

ExampleServiceEnvironment Remplacez-le par le nom de votre environnement de service.

La commande renvoie un résultat semblable à ce qui suit :

```
{  
  "jobQueueName": "my-sm-training-fifo-jq",  
  "jobQueueArn": "arn:aws:batch:region:account:job-queue/my-sm-training-fifo-jq"  
}
```

Après avoir créé votre file d'attente de tâches, vérifiez qu'elle a été créée avec succès et qu'elle est dans un état valide.

Utilisez la `describe-job-queues` commande pour afficher les détails de votre file d'attente de tâches :

```
aws batch describe-job-queues --job-queues my-sm-training-fifo-jq
```

La commande renvoie un résultat semblable à ce qui suit :

```
{  
  "jobQueues": [  
    {  
      "jobQueueName": "my-sm-training-fifo-jq",  
      "jobQueueArn": "arn:aws:batch:region:account:job-queue/my-sm-training-fifo-jq",  
      "state": "ENABLED",  
      "status": "VALID",  
      "statusReason": "JobQueue Healthy",  
      "priority": 1,  
      "computeEnvironmentOrder": [],  
      "serviceEnvironmentOrder": [  
        {  
          "order": 1,  

```



```

        "serviceEnvironment": "arn:aws:batch:region:account:service-
environment/ExampleServiceEnvironment"
      },
    ],
    "jobQueueType": "SAGEMAKER_TRAINING",
    "tags": {},
    "jobStateTimeLimitActions": []
  }
]
}

```

Assurez-vous que :

- Le state est ENABLED
- Le status est VALID
- Le statusReason est JobQueue Healthy
- Le jobQueueType est SAGEMAKER_TRAINING
- Les serviceEnvironmentOrder références de votre environnement de service

Modèle de file d'attente de tâches

Ce qui suit est un modèle de file d'attente de tâches vide. Vous pouvez utiliser ce modèle pour créer votre file d'attente de tâches. Vous pouvez ensuite enregistrer cette file d'attente de tâches dans un fichier et l'utiliser avec l' AWS CLI `--cli-input-json` option. Pour plus d'informations sur ces paramètres, consultez [CreateJobQueue](#) la référence de l'AWS Batch API.

Note

Vous pouvez générer un modèle de file d'attente de tâches à l'aide de la AWS CLI commande suivante.

```
$ aws batch create-job-queue --generate-cli-skeleton
```

```

{
  "computeEnvironmentOrder": [
    {
      "computeEnvironment": "",

```

```
        "order": 0
      }
    ],
    "jobQueueName": "",
    "jobStateTimeLimitActions": [
      {
        "state": "RUNNABLE",
        "action": "CANCEL",
        "maxTimeSeconds": 0,
        "reason": ""

      }
    ],
    "priority": 0,
    "schedulingPolicyArn": "",
    "state": "ENABLED",
    "tags": {
      "KeyName": ""
    }
  }
}
```

Afficher une file d'attente de tâches dans AWS Batch

Une fois que vous avez créé une file d'attente de tâches et que vous avez soumis les tâches, il est important de pouvoir suivre sa progression. Vous pouvez utiliser la page Détails des tâches pour consulter, gérer et surveiller votre file d'attente de tâches.

Afficher les informations relatives à la file d'attente

Dans la AWS Batch console, sélectionnez Queues de tâches dans le volet de navigation et choisissez la file d'attente de tâches de votre choix pour en afficher les détails. Sur cette page, vous pouvez consulter et gérer votre file d'attente de tâches et consulter des informations supplémentaires sur les opérations de la file d'attente, telles que l'instantané de la file d'attente de tâches, les limites d'état des tâches, l'ordre de l'environnement, les balises et le code JSON de la file d'attente de tâches.

Détails de la file d'attente de tâches

Cette section fournit une vue d'ensemble et des options de maintenance pour la file d'attente des tâches. Il est important de noter que vous trouverez le nom de ressource Amazon (ARN) dans cette section.

Pour trouver ces informations via le AWS Command Line Interface, utilisez l'[DescribeJobQueues](#) opération associée au nom de la file d'attente des tâches ou à l'ARN correspondant.

Actions actives

Pour les files d'attente de travail utilisant la planification du partage équitable, AWS Batch fournit une visibilité sur la manière dont les différents identifiants de partage consomment de la capacité. Ces informations vous aident à comprendre la distribution des ressources et à identifier les parts susceptibles de nécessiter un ajustement.

Note

L'onglet Active shares n'est présent que lorsque l'algorithme de planification de la file d'attente de tâches est Fair-share.

La section des 20 principaux partages actifs indique les identifiants de partage dont les tâches sont planifiées, de début et en cours d'exécution. Cette vue inclut :

- Nom de l'identifiant du partage : identifiant unique du partage.

Les identifiants de partage sont des libellés qui regroupent les tâches pour une planification équitable. Lorsque vous soumettez des tâches avec le même identifiant de partage, AWS Batch traite-les comme faisant partie de la même charge de travail à des fins d'allocation de ressources. Les identifiants de partage permettent de garantir une répartition équitable de la capacité de calcul entre les différentes équipes, les différents projets ou les différents types de charge de travail. Pour de plus amples informations, veuillez consulter [Utiliser des identifiants de partage pour identifier les charges de travail](#).

- Utilisation de la capacité : quantité de ressources que la tâche est configurée pour utiliser. Ceci est mesuré dans instances ou cpu en vCPU fonction de l'environnement.
- Action Afficher les offres d'emploi : lien permettant de voir toutes les offres d'emploi associées à ce partage.

Vous pouvez consulter ces informations sous forme de liste ou de graphique :

- Affichage en liste - Affichage tabulaire avec chiffres de capacité exacts
- Affichage graphique : graphique à barres visuel illustrant l'utilisation relative

Aperçu de la file d'attente de tâches

Cette section fournit une liste statique des 100 premières RUNNABLE tâches en file d'attente. Vous pouvez utiliser le champ de recherche pour affiner la liste en recherchant des informations dans n'importe quelle colonne de la section des résultats. Les tâches figurant dans la zone des résultats instantanés sont triées en fonction de la stratégie d'exécution de la file d'attente des tâches. Pour les files d'attente de tâches first-in-first-out (FIFO), l'ordre des tâches est basé sur l'heure de soumission. Pour les files d'attente de [tâches planifiées équitablement, l'ordre des tâches est basé sur la priorité des tâches et l'utilisation du partage](#). Le champ Capacité requise indique la quantité de ressources que la tâche est configurée pour utiliser.

Les résultats étant un instantané de la file d'attente des tâches, la liste des résultats n'est pas automatiquement mise à jour. Pour mettre à jour la liste, choisissez l'actualisation en haut de la section. Cliquez sur le lien hypertexte du nom de la tâche pour accéder aux détails de la tâche et consulter le statut de la tâche ainsi que d'autres informations connexes.

Pour trouver ces informations via le AWS CLI, utilisez l'[GetJobQueueSnapshot](#) opération associée au nom de la file d'attente des tâches ou à l'ARN correspondant.

```
aws batch get-job-queue-snapshot --job-queue my-sm-training-fifo-jq
```

Limites relatives à l'état des tâches

Utilisez cet onglet pour consulter les informations de configuration relatives à la durée pendant laquelle une tâche peut rester dans un RUNNABLE état avant son annulation.

Pour trouver ces informations via le AWS CLI, utilisez l'[DescribeJobQueues](#) opération associée au nom de la file d'attente des tâches ou à l'ARN correspondant.

Ordre environnemental

Si votre file d'attente de tâches s'exécute dans plusieurs environnements, cet onglet fournit leur ordre et une vue d'ensemble.

Pour trouver ces informations via le AWS CLI, utilisez l'[DescribeJobQueues](#) opération associée au nom de la file d'attente des tâches ou à l'ARN correspondant.

Étiquettes

Utilisez cet onglet pour consulter et gérer les balises associées à cette file d'attente de tâches.

JSON

Utilisez cet onglet pour copier le code JSON associé à cette file d'attente de tâches. Vous pouvez ensuite réutiliser le JSON pour les AWS CloudFormation modèles et les AWS CLI scripts.

Supprimer une file d'attente de tâches dans AWS Batch

Lorsque vous n'avez plus besoin de votre file d'attente de tâches, vous pouvez la désactiver et la supprimer.

Delete a job queue (AWS Batch console)

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans le volet de navigation, choisissez Job queues, puis choisissez une file d'attente de jobs.
3. Choisissez Actions, puis Désactiver.
4. Une fois que l'état de la file d'attente de tâches est désactivé, choisissez Actions, puis Supprimer.
5. Dans la fenêtre modale, choisissez Supprimer la file d'attente des tâches.

Delete a job queue (AWS CLI)

1. Désactivez la file d'attente des tâches pour empêcher la soumission de nouvelles tâches :

```
aws batch update-job-queue \  
  --job-queue my-sm-training-fifo-jq \  
  --state DISABLED
```

2. Attendez que les tâches en cours soient terminées, puis supprimez la file d'attente des tâches :

```
aws batch delete-job-queue \  
  --job-queue my-sm-training-fifo-jq
```

Politiques de planification équitables

Le AWS Batch planificateur évalue quand, où et comment exécuter les tâches soumises à une file d'attente de tâches. Si vous ne spécifiez pas de politique de planification lorsque vous créez une file

d'attente de tâches, le planificateur de AWS Batch tâches utilise par défaut une stratégie « premier entré, premier sorti » (FIFO). Une stratégie FIFO peut entraîner le « blocage » d'emplois importants par rapport à des emplois soumis plus tôt. En spécifiant une politique de planification différente, vous pouvez allouer des ressources de calcul en fonction de vos besoins spécifiques.

 Note

Si vous souhaitez planifier l'ordre spécifique dans lequel les tâches sont exécutées, utilisez le [dependsOn](#) paramètre in [SubmitJob](#) pour spécifier les dépendances de chaque tâche.

Si vous créez une politique de planification et que vous l'associez à une file d'attente de tâches, la planification équitable est activée. Si la file d'attente des tâches dispose d'une politique de planification, celle-ci détermine l'ordre dans lequel les tâches sont exécutées. Pour de plus amples informations, veuillez consulter [Utilisez des politiques de planification de partage équitable pour attribuer des identifiants de partage](#).

Rubriques

- [Utiliser des identifiants de partage pour identifier les charges de travail](#)
- [Utilisez des politiques de planification de partage équitable pour attribuer des identifiants de partage](#)
- [Utilisez la planification équitable pour faciliter la planification des tâches](#)
- [Tutoriel : Création d'une politique de planification](#)
- [Référence : modèle de politique de planification](#)

Utiliser des identifiants de partage pour identifier les charges de travail

Vous pouvez utiliser des identifiants de partage pour étiqueter les tâches et différencier les utilisateurs des charges de travail. Le AWS Batch planificateur suit l'utilisation de chaque identifiant de partage à l'aide de la $(T * weightFactor)$ formule, où *Test* l'utilisation du vCPU au fil du temps. Le planificateur sélectionne les tâches les moins utilisées à partir de l'identifiant de partage. Vous pouvez utiliser un identifiant de partage sans le remplacer.

 Note

Les identifiants de partage sont uniques au sein d'une file d'attente de tâches et ne sont pas agrégés entre les files d'attente de tâches.

Vous pouvez définir une priorité de planification équitable pour configurer l'ordre dans lequel les tâches sont exécutées sur un identifiant de partage. Les tâches dont la priorité de planification est plus élevée sont planifiées en premier. Si vous ne spécifiez pas de politique de planification équitable, toutes les tâches soumises à la file d'attente des tâches sont planifiées dans l'ordre FIFO. Lorsque vous soumettez une tâche, vous ne pouvez pas spécifier d'identifiant de partage ni de priorité de planification équitable.

 Note

Les ressources de calcul associées sont réparties de manière égale entre tous les identifiants de partage, sauf dérogation explicite.

Utilisez des politiques de planification de partage équitable pour attribuer des identifiants de partage

Vous pouvez utiliser des politiques de planification pour configurer la manière dont les ressources de calcul d'une file d'attente de tâches sont réparties entre les utilisateurs ou les charges de travail. À l'aide de politiques de planification de partage équitable, vous pouvez attribuer différents identifiants de partage aux charges de travail ou aux utilisateurs. AWS Batch attribue à chaque identifiant de partage un pourcentage du total des ressources disponibles pendant une période donnée.

Le pourcentage de juste part est calculé à l'aide des valeurs `shareDecaySeconds` et `shareDistribution`. Vous pouvez ajouter du temps à l'analyse du partage équitable en attribuant un temps de déclin des parts à la politique. L'ajout de temps donne plus de poids au temps et moins au poids défini. Vous pouvez réserver des ressources de calcul pour les identifiants de partage qui ne sont pas actifs en spécifiant une réservation de calcul. Pour de plus amples informations, veuillez consulter [SchedulingPolicyDetail](#).

Utilisez la planification équitable pour faciliter la planification des tâches

La planification équitable fournit un ensemble de contrôles pour faciliter la planification des tâches.

Note

Pour plus d'informations sur les paramètres des politiques de planification, consultez [SchedulingPolicyDetail](#).

- Secondes de déclin des parts : période (en secondes) utilisée par le AWS Batch planificateur pour calculer un pourcentage de partage équitable pour chaque identifiant de partage. Une valeur de zéro indique que seule l'utilisation actuelle est mesurée. Un temps de décomposition plus long donne plus de poids au temps.

Note

La période de décroissance est calculée comme suit : $shareDecaySeconds + OrderMinutes$ où $OrderMinutes$ est le temps dans l'ordre en minutes.

- Calculer la réservation : empêche les tâches associées à un identifiant de partage unique d'utiliser toutes les ressources associées à la file d'attente des tâches. Le ratio réservé `ActiveFairShares` correspond $(computeReservation/100)^{ActiveFairShares}$ au nombre d'identifiants de partage actifs.

Note

Si un identifiant de partage comporte des tâches dans un RUNNING état,,, ou SUBMITTED,,, PENDING,, RUNNABLE,, STARTING,,,,,,,,,,,,,,,,,,,,, Une fois la période de désintégration expirée, un identifiant de partage est considéré comme inactif.

- Facteur de pondération : facteur de pondération pour l'identifiant de l'action. La valeur par défaut est 1. Une valeur inférieure permet d'exécuter les tâches issues de l'identifiant de partage ou donne une durée d'exécution supplémentaire à l'identifiant de partage. Par exemple, les tâches qui utilisent un identifiant de partage avec un facteur de pondération de 0,125 (1/8) se voient attribuer huit fois les ressources de calcul des tâches qui utilisent un identifiant de partage avec un facteur de pondération de 1.

 Note

Vous devez uniquement définir cet attribut lorsque vous devez mettre à jour le facteur de pondération par défaut de 1.

Lorsque la file d'attente des tâches est active et que des tâches sont en cours de traitement, vous pouvez consulter la liste des 100 premières RUNNABLE tâches via l'instantané de la file d'attente des tâches. Pour de plus amples informations, veuillez consulter [Afficher une file d'attente de tâches dans AWS Batch](#).

Tutoriel : Création d'une politique de planification

Avant de créer une file d'attente de tâches avec une politique de planification, vous devez créer une politique de planification. Lorsque vous créez une politique de planification équitable, vous associez un ou plusieurs identifiants de partage ou partagez des préfixes d'identifiant à des pondérations pour la file d'attente et vous attribuez éventuellement une période de décroissance et calculez la réservation à la politique.

Pour créer une politique de planification

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation, sélectionnez la région à utiliser.
3. Dans le volet de navigation, choisissez Politiques de planification, puis Créer.
4. Dans Nom, entrez un nom unique pour votre politique de planification. Il peut comporter jusqu'à 128 lettres (majuscules et minuscules), des chiffres, des tirets ou des traits de soulignement.
5. (Facultatif) Pour les secondes de décroissance du partage, entrez une valeur entière pour le temps de déclin des actions prévu par la politique de planification du partage équitable. Un temps de dégradation des parts plus long nécessitera une utilisation beaucoup plus importante des ressources informatiques sur une plus longue période lors de la planification des tâches. Cela peut permettre aux tâches utilisant un identifiant de partage d'utiliser temporairement plus de ressources de calcul que le poids de cet identifiant de partage ne le permettrait si cet identifiant de partage n'avait pas utilisé de ressources de calcul récemment.
6. (Facultatif) Pour Calculer la réservation, entrez une valeur entière pour la réservation de calcul de la politique de planification de partage équitable. La réservation de calcul contiendra une

partie de v CPUs en réserve à utiliser pour les identifiants de partage qui ne sont pas actifs actuellement.

Le ratio réservé `ActiveFairShares` correspond

$(\text{computeReservation}/100)^{\text{ActiveFairShares}}$ au nombre d'identifiants de partage actifs.

Par exemple, une `computeReservation` valeur de 50 indique qu'il AWS Batch faut réserver 50 % du maximum de processeurs virtuels disponibles s'il n'y a qu'un seul identifiant de partage, 25 % s'il existe deux identifiants de partage et 12,5 % s'il y a trois identifiants de partage.

Une `computeReservation` valeur de 25 indique que 25 % du processeur virtuel maximum disponible AWS Batch devrait être réservé s'il n'y a qu'un seul identifiant de partage, 6,25 % s'il existe deux identifiants de partage et 1,56 % s'il existe trois identifiants de partage.

7. Dans la section Attributs de partage, vous pouvez spécifier l'identifiant de partage et le poids de chaque identifiant de partage à associer à la politique de planification du partage équitable.
 - a. Choisissez Ajouter un identifiant de partage.
 - b. Pour Identifiant de partage, spécifiez l'identifiant de partage. Si la chaîne se termine par « * », cela devient un préfixe d'identifiant de partage utilisé pour faire correspondre les identifiants de partage pour les tâches. Tous les identifiants de partage et les préfixes d'identifiant de partage d'une politique de planification doivent être uniques et ne peuvent pas se chevaucher. Par exemple, vous ne pouvez pas avoir le préfixe d'identifiant de partage « UserA* » et l'identifiant de partage « UserA1 » dans la même politique de planification de partage équitable.
 - c. Dans le champ Facteur de pondération, spécifiez le poids relatif de l'identifiant d'action. La valeur par défaut est 1.0. Une valeur inférieure a une priorité plus élevée pour les ressources de calcul. Si un préfixe d'identifiant de partage est utilisé, les tâches dont les identifiants de partage commencent par le préfixe partageront le facteur de pondération. Cela augmente efficacement le facteur de pondération pour ces tâches, en abaissant leur priorité individuelle tout en maintenant le même facteur de pondération pour le préfixe d'identification des actions.
8. (Facultatif) Dans la section Balises, vous pouvez spécifier la clé et la valeur de chaque balise à associer à la politique de planification. Pour de plus amples informations, veuillez consulter [Marquez vos AWS Batch ressources](#).
9. Choisissez Soumettre pour terminer et créer votre politique de planification.

Référence : modèle de politique de planification

Un modèle de politique de planification vide est illustré ci-dessous. Vous pouvez utiliser ce modèle pour créer votre politique de planification qui peut ensuite être enregistrée dans un fichier et utilisée avec l' AWS CLI `--cli-input-json` option. Pour plus d'informations sur ces paramètres, consultez [CreateSchedulingPolicy](#) la référence de l'AWS Batch API.

Note

Vous pouvez générer un modèle de file d'attente de tâches à l'aide de la AWS CLI commande suivante.

```
$ aws batch create-scheduling-policy --generate-cli-skeleton
```

```
{
  "name": "",
  "fairsharePolicy": {
    "shareDecaySeconds": 0,
    "computeReservation": 0,
    "shareDistribution": [
      {
        "shareIdentifier": "",
        "weightFactor": 0.0
      }
    ]
  },
  "tags": {
    "KeyName": ""
  }
}
```

Planification axée sur les ressources

AWS Batch planifie les tâches en fonction du vCPU, du GPU et de la disponibilité de la mémoire dans l'environnement informatique (CE) associé à la file d'attente de tâches (JQ). Mais parfois, la simple disponibilité de ces ressources CE ne garantit pas le succès de la tâche, car elle peut dépendre d'autres ressources requises. Ces tâches sont donc annulées ou résiliées. Cela se traduit par une

utilisation inefficace des ressources informatiques. Pour résoudre ce problème, la planification axée sur les ressources peut vérifier la disponibilité des ressources dépendantes autres que le CE avant de planifier l'exécution de la tâche sur un CE.

AWS Batch La planification axée sur les ressources vous permet de planifier les tâches en fonction des ressources consommables nécessaires à leur exécution : jetons de licence tiers, bande passante d'accès à la base de données, nécessité de limiter les appels vers une API tierce, etc. Vous spécifiez les ressources consommables nécessaires à l'exécution d'une tâche, et Batch prend en compte ces dépendances de ressources lorsqu'il planifie une tâche. Vous pouvez éviter les interventions manuelles afin d'éliminer les échecs de travail et les longues attentes causés par une pénurie de ressources consommables. Vous pouvez réduire la sous-utilisation des ressources de calcul en n'allouant que les tâches disposant de toutes les ressources nécessaires.

La planification basée sur les ressources est disponible pour les politiques de planification FIFO et Fair-share et peut être utilisée avec toutes les plateformes de calcul prises en charge par Batch, notamment EKS, ECS et Fargate. Il peut être utilisé avec des tâches Array, des tâches MNP (Multi-Node parallel) et avec des tâches Batch classiques.

Pour configurer une planification adaptée aux ressources, vous devez d'abord spécifier toutes les ressources consommables nécessaires à l'exécution de vos tâches, ainsi que le nombre total disponible pour chaque ressource. Ensuite, pour chaque tâche nécessitant une ressource consommable, vous spécifiez le nom et les quantités requises de chaque ressource requise. Batch enregistre le nombre de ressources consommables disponibles pour les tâches de vos files d'attente et garantit qu'une tâche est planifiée pour s'exécuter uniquement lorsque toutes les ressources consommables nécessaires sont disponibles pour que la tâche s'exécute correctement.

Rubriques

- [Créez des ressources consommables](#)
- [Spécifiez les ressources nécessaires pour exécuter une tâche](#)
- [Vérifiez le nombre de ressources utilisées et disponibles](#)
- [Mettre à jour la quantité d'une ressource lorsqu'elle est utilisée par des tâches](#)
- [Trouvez les tâches qui nécessitent une ressource consommable spécifique](#)
- [Supprimer une ressource consommable](#)

Créez des ressources consommables

Vous devez d'abord créer les ressources consommables qui représentent les ressources non CE consommées lors de l'exécution d'une tâche et qui ne sont disponibles qu'en quantité limitée. Chaque ressource consommable possède :

- nom de ressource (`consumableResourceName`) qui doit être unique au niveau du compte.
- (facultatif) type de ressource (`resourceType`) qui indique si la ressource est disponible pour être réutilisée une fois la tâche terminée. Cela peut être l'un des suivants :
 - REPLENISHABLE (default)
 - NON_REPLENISHABLE
- quantité totale (`totalQuantity`) qui indique le montant total de la ressource consommable disponible.

Le nombre maximum de ressources consommables par compte est de 50 000.

Console :

1. Dans le panneau de navigation de gauche de la [AWS Batch console](#), sélectionnez Ressources consommables.
2. Choisissez Créer une ressource consommable.
3. Entrez un nom de ressource unique, la quantité totale de ressources et indiquez si le type de ressource est réapprovisionnementnable ou non réapprovisionnementnable.
4. Choisissez Créer une ressource consommable.

API :

Utilisez l'[CreateConsumableResourceAPI](#) pour définir les ressources souhaitées.

Spécifiez les ressources nécessaires pour exécuter une tâche

Lorsque vous enregistrez une tâche, vous pouvez spécifier le nom d'une ou de plusieurs ressources que vous avez créées (`consumableResource`) et la quantité de cette ressource requise par chaque instance de la tâche (`quantity`).

Batch garde une trace des unités disponibles de chaque ressource à tout moment. Pour chaque tâche de la file d'attente, le planificateur Batch garantit que votre tâche ne s'exécute que lorsque les dépendances de ressources spécifiées sont disponibles.

Si aucune ressource consommable pour la tâche n'est disponible lorsque la tâche atteint la tête de la file d'attente, la tâche attendra jusqu'à ce que toutes les ressources requises soient disponibles ou que la limite de temps de l'état de la tâche soit atteinte (voir [Afficher une file d'attente de tâches dans AWS Batch](#)). **RUNNABLE** Une fois que Batch a confirmé que toutes les ressources sont disponibles, le job passe à l'**STARTING** état puis à **RUNNING**. Les ressources sont verrouillées une fois que la tâche est **STARTING** déplacée vers ou sont déverrouillées lorsque la tâche est déplacée vers **SUCCEEDED** ou **FAILED**.

Vous pouvez également mettre à jour la quantité de ressource nécessaire pour une tâche spécifique lorsque vous soumettez la tâche.

Console :

Pour spécifier les ressources et les quantités nécessaires lorsque vous définissez une tâche, procédez comme suit :

1. Définissez une tâche à l'aide de l'assistant de définition de tâche depuis la [AWS Batch console](#) (Définitions de tâches -> Créer).
2. Dans l'étape 4 de l'assistant : Configuration des conteneurs, sous Ressource consommable, sélectionnez le nom d'une ressource requise dans la liste. Dans le champ Valeur demandée, entrez la quantité de cette ressource requise par une instance de cette tâche, puis choisissez Ajouter une ressource consommable.
3. Répétez l'étape précédente pour toutes les ressources consommables requises par la tâche. Vous pouvez spécifier jusqu'à 5 ressources pour chaque tâche que vous définissez.
4. Vous verrez la liste des ressources consommables que vous avez créées une fois que vous aurez terminé l'assistant de définition de tâche, mais avant de choisir Créer une définition de tâche.

Pour mettre à jour les quantités de ressources nécessaires lorsque vous soumettez une offre d'emploi :

1. Dans le volet de navigation gauche de la [AWS Batch console](#), choisissez Tâches, puis Soumettre une nouvelle tâche.

2. À l'étape 2 de l'assistant : Configuration des remplacements, sous Dérogations de ressources consommables, entrez une nouvelle valeur demandée pour toute ressource consommable dont vous souhaitez annuler la quantité nécessaire pour la tâche.
3. Une fois que vous avez effectué toutes les modifications que vous souhaitez effectuer pour cette tâche, cliquez sur Suivant pour passer à Réviser et soumettre.

API :

Lorsque vous enregistrez une tâche auprès de l'[RegisterJobDefinitionAPI](#), utilisez la `consumableResourceList` `consumableResourceProperties` partie de la demande pour spécifier les ressources consommables requises pour exécuter une instance de la tâche, ainsi que la quantité de chacune d'elles.

Lorsque vous soumettez une tâche avec l'[SubmitJobAPI](#), vous pouvez remplacer la liste des ressources consommables et la quantité de chacune d'elles en utilisant la `consumableResourcePropertiesOverride` partie de la demande. Notez que cela remplace uniquement la quantité de ressource requise par chaque instance de la tâche, et non la quantité totale disponible.

Vérifiez le nombre de ressources utilisées et disponibles

Batch vous permet de demander le nombre de ressources disponibles (`availableQuantity`), le nombre de ressources utilisées (`inUseQuantity`) et le total des ressources (`totalQuantity`) à un moment donné.

Une fois qu'une tâche passe à `STARTING` cet état, les ressources consommées sont soustraites de la quantité disponible de cette ressource. Si c'est le cas `REPLENISHABLE`, le nombre de ressources consommées sera ajouté à la quantité disponible dès que la tâche sera passée à l'état `SUCCEEDED` ou `FAILED`, et la quantité totale restera la même. Si c'est le cas `NON_REPLENISHABLE`, le nombre de ressources consommées est soustrait à la fois du total et des quantités disponibles et ne sera pas additionné, que la tâche passe à l'`FAILED` état `SUCCEEDED` ou.

Note

Ces informations peuvent être décalées de 30 secondes au maximum.

Console :

1. Dans le panneau de navigation de gauche de la [AWS Batch console](#), sélectionnez Ressources consommables.
2. Sélectionnez l'onglet Réapprovisionnement ou Non réapprovisionnement pour afficher les ressources de ce type que vous avez créées.
3. Pour chaque ressource réapprovisionnement, la console affiche le nom, la quantité totale de la ressource, le nombre actuellement utilisé et le nombre encore disponible, ainsi qu'un calcul de l'utilisation (le nombre de ressources utilisées divisé par la quantité totale de cette ressource).

Pour chaque ressource non réapprovisionnement, la console affiche le nom, le numéro actuellement utilisé et le nombre de ressources encore disponibles.

Vous pouvez également consulter les informations actuelles sur les ressources consommables à partir d'une page détaillée de la tâche dans la console.

1. Dans le panneau de navigation de gauche de la [AWS Batch console](#), choisissez Tâches, puis sélectionnez le nom d'une tâche pour ouvrir la page de détails de cette tâche.
2. Les informations relatives aux ressources réapprovisionnement et aux ressources non réapprovisionnement peuvent être consultées si le travail l'exige. Pour les deux types, la console affiche le nom de la ressource, la quantité demandée pour le travail, le nombre encore disponible, le nombre actuellement utilisé, la quantité totale de la ressource, ainsi qu'un calcul de l'utilisation actuelle (le nombre de ressources utilisées par le travail divisé par la quantité totale de cette ressource).

API :

Utilisez l'[DescribeConsumableResourceAPI](#) qui renvoie les informations suivantes :

```
{
  "availableQuantity": number,
  "consumableResourceArn": "string",
  "consumableResourceName": "string",
  "createdAt": number,
  "inUseQuantity": number,
  "resourceType": "string",
  "tags": {
    "string" : "string"
  },
  "totalQuantity": number
```



```
}
```

L'[ListConsumableResourcesAPI](#) indique également le nombre de ressources utilisées (`inUseQuantity`) et le nombre total de ressources actuellement disponibles (`totalQuantity`) dans le cadre de la liste de toutes les ressources consommables que vous avez créées dans votre compte. Cette API vous permet également de filtrer la requête de liste des ressources consommables en fonction du nom de la ressource consommable.

Mettre à jour la quantité d'une ressource lorsqu'elle est utilisée par des tâches

Vous pouvez rétablir la quantité totale d'une ressource à une nouvelle valeur, l'ajouter à la quantité totale ou la soustraire.

Si la nouvelle quantité totale que vous spécifiez est supérieure à la quantité précédente, Batch planifie davantage de tâches en conséquence. Si la nouvelle quantité totale est inférieure à la précédente et qu'aucune unité de cette ressource n'est utilisée, Batch réduit simplement la quantité totale (ou disponible). Si des unités sont utilisées, Batch réduit immédiatement la quantité disponible et, une fois les tâches terminées, Batch réduit la quantité totale (disponible) afin qu'elle atteigne finalement le nouveau numéro.

Console :

1. Dans le panneau de navigation de gauche de la [AWS Batch console](#), sélectionnez Ressources consommables.
2. Sélectionnez l'onglet Réapprovisionnement ou Non réapprovisionnement pour afficher les ressources de ce type que vous avez créées.
3. Pour les ressources réapprovisionnables :
 1. Choisissez la ressource que vous souhaitez mettre à jour, puis sélectionnez Actions et choisissez Définir les ressources, Ajouter des ressources ou Supprimer des ressources.
 2. Une fenêtre contextuelle apparaît dans laquelle vous pouvez définir la valeur totale, ajouter des ressources ou supprimer des ressources en fonction de l'action que vous avez choisie à l'étape précédente. Entrez la quantité que vous souhaitez définir comme nouvelle valeur totale, que vous souhaitez ajouter à la quantité totale ou que vous souhaitez soustraire de la quantité totale, puis sélectionnez OK.

Pour les ressources non réapprovisionnables :

1. Choisissez la ressource que vous souhaitez mettre à jour, puis sélectionnez Actions et choisissez Définir les ressources, Ajouter des ressources ou Supprimer des ressources.
2. Une fenêtre contextuelle apparaît dans laquelle vous pouvez définir la valeur disponible, ajouter des ressources ou supprimer des ressources en fonction de l'action que vous avez choisie à l'étape précédente. Entrez la quantité que vous souhaitez définir comme nouvelle valeur disponible, que vous souhaitez ajouter à la quantité disponible ou que vous souhaitez soustraire de la quantité disponible, puis sélectionnez OK.

API :

Utilisez l'[UpdateConsumableResourceAPI](#) pour définir une nouvelle quantité totale pour la ressource, ou pour augmenter ou réduire la quantité totale.

Trouvez les tâches qui nécessitent une ressource consommable spécifique

Batch vous permet de récupérer une liste de tâches qui nécessitent une ressource consommable spécifique.

Console :

1. Dans le panneau de navigation de gauche de la [AWS Batch console](#), sélectionnez Ressources consommables.
2. Dans la liste, sélectionnez le nom de la ressource consommable. La page de détails de cette ressource s'ouvre.
3. Sous Rechercher des emplois, entrez les filtres que vous souhaitez appliquer à la liste des emplois. Vous pouvez filtrer en fonction du nom de la tâche (« égal à », « commence par »), de la plage de dates (date de création de la tâche) et de critères supplémentaires (« file d'attente des tâches », « définition de la tâche », « identifiant de tâche partagé »). Pour chaque type de filtre que vous souhaitez appliquer, sélectionnez l'une des options disponibles dans la liste déroulante et entrez les informations supplémentaires demandées.

Choisissez Rechercher.

4. Une liste (filtrée) des tâches nécessitant la ressource consommable s'affiche, y compris le nom de la tâche, le statut, le nombre d'unités demandées de la ressource consommable, les autres ressources consommables nécessaires, etc. À l'aide de cette liste, vous pouvez sélectionner une ou plusieurs tâches à annuler ou à terminer. Vous pouvez également sélectionner le nom d'une tâche pour ouvrir la page détaillée de cette tâche.

5. Sous Rechercher des tâches, vous pouvez désormais actualiser les résultats ou effacer la recherche et recommencer à zéro.

API :

Vous pouvez obtenir une liste des tâches qui utilisent une ressource consommable spécifique grâce à l'[ListJobsByConsumableResourceAPI](#). Cette API vous permet également de filtrer la requête de liste des tâches en utilisant le statut ou le nom de la tâche.

Supprimer une ressource consommable

Vous pouvez supprimer une ressource consommable à tout moment, même lorsque les tâches nécessitant cette ressource sont toujours en cours d'exécution. Une fois qu'une ressource consommable est supprimée, il peut y avoir un intervalle entre le moment où la commande de suppression est reçue et le moment où le planificateur de tâches honore la suppression. Il est donc possible que les tâches consommant la ressource soient planifiées juste après l'appel de suppression. Si la ressource consommable supprimée possède le type de ressource (`resourceType`) `REPLENISHABLE`, cela sera ignoré une fois les tâches terminées. Si vous supprimez une ressource consommable et que vous la recréez sous le même nom, elle est considérée comme la même ressource et elle peut être utilisée par `RUNNABLE` des tâches.

Console :

1. Dans le panneau de navigation de gauche de la [AWS Batch console](#), sélectionnez Ressources consommables.
2. Sélectionnez l'onglet Réapprovisionnement ou Non réapprovisionnement pour afficher les ressources de ce type que vous avez créées.
3. Sélectionnez chaque ressource que vous souhaitez supprimer, puis choisissez Supprimer. Une fenêtre contextuelle Supprimer la ressource consommable apparaît. Pour confirmer la suppression, choisissez Supprimer.

Vous pouvez également supprimer une ressource consommable de sa page détaillée dans la console.

1. Dans le panneau de navigation de gauche de la [AWS Batch console](#), sélectionnez Ressources consommables.

2. Sélectionnez l'onglet Réapprovisionnement ou Non réapprovisionnement pour afficher les ressources de ce type que vous avez créées.
3. Choisissez le nom de la ressource que vous souhaitez supprimer. La page de détails de la ressource consommable s'affiche. Sélectionnez Delete (Supprimer). Une fenêtre contextuelle Supprimer la ressource consommable apparaît. Pour confirmer la suppression, choisissez Supprimer.

API :

Utilisez l'[DeleteConsumableResourceAPI](#) pour supprimer une ressource consommable.

Définitions des emplois

AWS Batch les définitions de tâches spécifient la manière dont les tâches doivent être exécutées. Bien que chaque tâche doive faire référence à une définition de tâche, de nombreux paramètres spécifiés dans la définition de tâche peuvent être remplacés lors de l'exécution.

Parmi les attributs spécifiés dans une définition de tâche figurent les suivants :

- Quelle image Docker utiliser avec le conteneur dans votre travail.
- Combien de v CPUs et quelle quantité de mémoire utiliser avec le conteneur.
- La commande que le conteneur doit exécuter lors de son démarrage.
- Quelles variables d'environnement (le cas échéant) doivent être transmises au conteneur lors de son démarrage.
- Tous les volumes de données qui doivent être utilisés avec le conteneur.
- Quel rôle IAM (le cas échéant) votre travail doit utiliser pour les AWS autorisations.

Table des matières

- [Création d'une définition de tâche à nœud unique](#)
- [Création d'une définition de tâche parallèle à nœuds multiples](#)
- [Modèle de définition de travail qui utilise ContainerProperties](#)
- [Créez des définitions de tâches à l'aide de EcsProperties](#)
- [Utiliser le pilote de journal awslogs](#)
- [Spécifier les données sensibles](#)
- [Authentification par registre privé pour les tâches](#)
- [Volumes Amazon EFS](#)
- [Exemples de définitions de tâches](#)

Création d'une définition de tâche à nœud unique

Avant de pouvoir exécuter des tâches AWS Batch, vous devez créer une définition de tâche. Ce processus varie légèrement entre les tâches parallèles à nœud unique et à nœuds multiples. Cette rubrique explique spécifiquement comment créer une définition de tâche pour une tâche qui n'est pas

une AWS Batch tâche parallèle à plusieurs nœuds (également connue sous le nom de planification en groupe).

Vous pouvez créer une définition de tâche parallèle à plusieurs nœuds sur les ressources Amazon Elastic Container Service. Pour de plus amples informations, veuillez consulter [the section called “Création d'une définition de tâche parallèle à nœuds multiples”](#).

Rubriques

- [Création d'une définition de tâche à nœud unique sur Amazon Resources EC2](#)
- [Création d'une définition de tâche à nœud unique sur les ressources Fargate](#)
- [Création d'une définition de tâche à nœud unique sur les ressources Amazon EKS](#)
- [Créez une définition de tâche à nœud unique avec plusieurs conteneurs sur les ressources Amazon EC2](#)

Création d'une définition de tâche à nœud unique sur Amazon Resources EC2

Procédez comme suit pour créer une définition de tâche à nœud unique sur les ressources Amazon Elastic Compute Cloud (Amazon EC2).

Pour créer une nouvelle définition de tâche sur Amazon EC2 Resources, procédez comme suit :

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation, choisissez le Région AWS à utiliser.
3. Dans le volet de navigation, choisissez Définitions de tâches.
4. Choisissez Créer.
5. Pour le type d'orchestration, choisissez Amazon Elastic Compute Cloud (Amazon EC2).
6. Pour la configuration de la EC2 plate-forme, désactivez l'option Activer le traitement parallèle à nœuds multiples.
7. Dans Nom, entrez un nom unique pour votre définition de tâche. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
8. (Facultatif) Dans le champ Délai d'exécution, entrez la valeur du délai d'expiration (en secondes). Le délai d'exécution est le délai avant la fin d'une tâche inachevée. Si une tentative dépasse le délai imparti, elle est arrêtée et passe à un FAILED statut. Pour de plus amples informations,

veuillez consulter [Expiration des délais d'exécution des tâches](#). La valeur minimale est 60 secondes.

9. (Facultatif) Activez la priorité de planification. Entrez une valeur de priorité de planification comprise entre 0 et 100. Les valeurs les plus élevées reçoivent une priorité plus élevée.
10. (Facultatif) Pour les tentatives de tâche, entrez le nombre de AWS Batch tentatives de transfert de la tâche vers le RUNNABLE statut. Entrez un nombre compris entre 1 et 10.
11. (Facultatif) Pour les conditions de la stratégie Réessayer, choisissez Ajouter une évaluation à la sortie. Entrez au moins une valeur de paramètre, puis choisissez une action. Pour chaque ensemble de conditions, l'action doit être définie sur Réessayer ou sur Quitter. Ces actions signifient ce qui suit :
 - Réessayer — AWS Batch Réessaie jusqu'à ce que le nombre de tentatives de travail que vous avez spécifié soit atteint.
 - Quitter : AWS Batch arrête de réessayer la tâche.

 Important

Si vous choisissez Ajouter une évaluation à la sortie, vous devez configurer au moins un paramètre et soit choisir une action, soit choisir Supprimer l'évaluation à la sortie.

12. (Facultatif) Développez les balises, puis choisissez Ajouter une balise pour ajouter des balises à la ressource. Entrez une clé et une valeur facultative, puis choisissez Add tag (Ajouter une balise).
13. (Facultatif) Activez les balises de propagation pour propager les balises de la tâche et de la définition de la tâche vers la tâche Amazon ECS.
14. Choisissez Page suivante.
15. Dans la section Configuration du conteneur :
 - a. Dans Image, choisissez l'Dockerimage à utiliser pour votre travail. Par défaut, les images dans le registre Docker Hub sont disponibles. Vous pouvez également spécifier d'autres référentiels avec *repository-url/image:tag*. Le nom peut contenir jusqu'à 225 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-), des traits de soulignement (_), des signes deux-points (:), des barres obliques (/) et des signes numériques (#). Ce paramètre se mappe à Image dans la section [Create a container](#) (Création d'un conteneur) de [Docker Remote API](#) (L'API Docker à distance) et le paramètre IMAGE de [docker run](#).

 Note

L'architecture des images Docker doit correspondre à l'architecture du processeur des ressources de calcul sur lesquelles elles sont planifiées. Par exemple, des images Docker basées sur ARM ne peuvent s'exécuter que sur des ressources de calcul basées sur ARM.

- Les images contenues dans les référentiels publics Amazon ECR utilisent les conventions complètes `registry/repository[:tag]` ou de `registry/repository[@digest]` dénomination (par exemple, `public.ecr.aws/registry_alias/my-web-app:latest`).
 - Les images contenues dans les référentiels Amazon ECR utilisent la convention de `registry/repository[:tag]` dénomination complète (par exemple, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`).
 - Les images dans les référentiels officiels sur Docker Hub utilisent un nom unique (par exemple, `ubuntu` ou `mongo`).
 - Les images dans les autres référentiels sur Docker Hub sont qualifiées par un nom d'organisation (par exemple, `amazon/amazon-ecs-agent`).
 - Les images dans les autres référentiels en ligne sont qualifiées par un nom de domaine (par exemple, `quay.io/assemblyline/ubuntu`).
- b. Pour `Command`, saisissez les commandes dans le champ comme leur équivalent sous forme de tableau de chaînes JSON.

Ce paramètre correspond à `Cmd` dans la section relative à la [création d'un conteneur](#) de [Docker Remote API](#) et au paramètre `COMMAND` de [docker run](#). Pour plus d'informations sur le Docker CMD paramètre, consultez <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note

Vous pouvez utiliser des valeurs par défaut pour la substitution de paramètres et des espaces réservés dans votre commande. Pour de plus amples informations, veuillez consulter [Paramètres](#).

- c. (Facultatif) Pour le rôle d'exécution, spécifiez un rôle IAM qui autorise les agents de conteneur Amazon ECS à effectuer des appels d' AWS API en votre nom. Cette

fonctionnalité utilise les rôles IAM d'Amazon ECS pour les tâches. Pour plus d'informations, consultez les [rôles IAM d'exécution de tâches Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.

- d. Pour la configuration du rôle Job, choisissez un rôle IAM autorisé à accéder au AWS APIs. Cette fonctionnalité utilise les rôles IAM d'Amazon ECS pour les tâches. Pour plus d'informations, consultez [Rôles IAM pour les tâches](#) dans le Guide du développeur Amazon Elastic Container Service.

 Note

Seuls les rôles dotés d'une relation de confiance avec Amazon Elastic Container Service Task Role sont présentés ici. Pour plus d'informations sur la création d'un rôle IAM pour vos AWS Batch tâches, consultez la section [Création d'un rôle IAM et d'une politique pour vos tâches](#) dans le manuel Amazon Elastic Container Service Developer Guide.

16. Pour Paramètres, choisissez Ajouter des paramètres pour ajouter des espaces réservés de substitution de paramètres sous forme de paires clé et valeur facultatives.
17. Dans la section Configuration de l'environnement :
 - a. Pour v CPUs, entrez le nombre de v CPUs à réserver pour le conteneur. Ce paramètre correspond à CpuShares dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option `--cpu-shares` correspond à [docker run](#). Chaque processeur virtuel est équivalente à 1 024 parts de processeur. Vous devez spécifier au moins un vCPU.
 - b. Pour Mémoire, entrez la limite de mémoire disponible pour le conteneur. Si votre conteneur tente de dépasser la quantité de mémoire spécifiée ici, il est arrêté. Ce paramètre correspond à Memory dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option `--memory` correspond à [docker run](#). Vous devez spécifier au moins 4 Mio de mémoire pour une tâche.

 Note

Pour optimiser l'utilisation de vos ressources, donnez la priorité à la mémoire pour les tâches d'un type d'instance spécifique. Pour de plus amples informations, veuillez consulter [Gestion de la mémoire des ressources informatiques](#).

- c. Dans le champ Nombre de GPUs, choisissez le nombre de GPUs à réserver pour le conteneur.
 - d. (Facultatif) Pour les variables d'environnement, choisissez Ajouter une variable d'environnement pour ajouter des variables d'environnement sous forme de paires nom-valeur. Ces variables sont transmises au conteneur.
 - e. (Facultatif) Pour les secrets, choisissez Ajouter un secret pour ajouter des secrets sous forme de paires nom-valeur. Ces secrets sont exposés dans le conteneur. [Pour plus d'informations, consultez : LogConfiguration SecretOptions.](#)
18. Choisissez Page suivante.
19. Dans la section de configuration Linux :
- a. Dans Interface, saisissez le nom d'utilisateur à utiliser à l'intérieur du conteneur. Ce paramètre correspond à `User` dans la section [Create a container](#) (Créer un conteneur) de [l'API Docker à distance](#) et l'option `--user` correspond à [docker run](#).
 - b. (Facultatif) Pour accorder au conteneur de tâches des autorisations élevées sur l'instance hôte (comme pour l'`root`utilisateur), faites glisser le curseur Privileged vers la droite. Ce paramètre correspond à `Privileged` dans la section [Create a container](#) (Créer un conteneur) de [l'API Docker à distance](#) et l'option `--privileged` correspond à [docker run](#).
 - c. (Facultatif) Activez Activer le processus d'initialisation pour exécuter un `init` processus dans le conteneur. Ce processus transmet les signaux et récolte les processus.
20. (Facultatif) Dans la section Configuration du système de fichiers :
- a. Activez l'option Enable read only filesystem (Activer le système de fichiers en lecture seule) pour supprimer l'accès en écriture au volume.
 - b. Pour Taille de la mémoire partagée, entrez la taille (en MiB) du `/dev/shm` volume.
 - c. Pour Taille maximale de swap, entrez la quantité totale de mémoire d'échange (en MiB) que le conteneur peut utiliser.
 - d. Pour Swappiness, entrez une valeur comprise entre 0 et 100 pour indiquer le comportement de swappiness du conteneur. Si vous ne spécifiez aucune valeur et que le swap est activé, la valeur par défaut est 60. [Pour plus d'informations, consultez : LinuxParameters swappiness.](#)
 - e. (Facultatif) Développez la configuration supplémentaire.
 - f. (Facultatif) Pour les fichiers Tmpfs, choisissez Ajouter des fichiers tmpfs pour ajouter un montage. `tmpfs`

- g. (Facultatif) Pour Appareils, choisissez Ajouter un appareil pour ajouter un appareil :
 - i. Sous Chemin du conteneur, spécifiez le chemin d'accès de l'instance du conteneur pour exposer l'appareil mappé à l'instance de l'hôte. Si vous laissez ce champ vide, le chemin de l'hôte est utilisé dans le conteneur.
 - ii. Sous Host path (Chemin de l'hôte), spécifiez le chemin d'accès d'un appareil dans l'instance de l'hôte.
 - iii. Pour Autorisations, choisissez une ou plusieurs autorisations à appliquer à l'appareil. Les autorisations disponibles sont READ, WRITE et MKNOD.
- h. (Facultatif) Pour la configuration des volumes, choisissez Ajouter un volume pour créer une liste de volumes à transmettre au conteneur. Entrez le nom et le chemin source du volume, puis choisissez Ajouter un volume. Vous pouvez également choisir d'activer l'option Activer EFS.
- i. (Facultatif) Pour les points de montage, choisissez Ajouter une configuration de points de montage pour ajouter des points de montage pour les volumes de données. Vous devez spécifier le volume source et le chemin du conteneur. Ces points de montage sont transmis Docker daemon à une instance de conteneur. Vous pouvez également choisir de rendre le volume en lecture seule.
- j. (Facultatif) Pour la configuration Ulimits, choisissez Add ulimit pour ajouter une ulimits valeur au conteneur. Entrez les valeurs Nom, Limite souple et Limite stricte, puis choisissez Ajouter une limite.

21. Dans la section Propriétés de la tâche :

- a. Pour Rôle d'exécution - conditionnel, choisissez un rôle pour autoriser les agents Amazon ECS à effectuer des appels d' AWS API en votre nom. Pour plus d'informations sur la création d'un rôle d'exécution, consultez [Tutoriel : Création du rôle d'exécution IAM](#).
- b. Choisissez la commande Enable ECS execute pour permettre l'accès direct au shell du conteneur Amazon ECS et contourner le système d'exploitation hôte. Vous devez choisir un rôle de tâche.

 Important

La commande ECS execute nécessite que le système de fichiers soit accessible en écriture.

- c. Pour le rôle Task, choisissez un rôle Amazon ECS Identity and Access Management (IAM) pour autoriser le conteneur à effectuer des appels d' AWS API en votre nom. Pour plus d'informations, consultez le [rôle IAM de la tâche Amazon ECS](#) dans le manuel du développeur Amazon Elastic Container Service.

22. (Facultatif) Dans la section Configuration de la journalisation :

- a. Pour Log driver, choisissez le log driver à utiliser. [Pour plus d'informations sur les pilotes de journal disponibles, voir : LogDriverLogConfiguration.](#)

 Note

Par défaut, le pilote de awslogs journal est utilisé.

- b. Pour Options, choisissez Ajouter une option pour ajouter une option. Entrez une paire nom-valeur, puis choisissez Ajouter une option.
- c. Pour Secrets, choisissez Ajouter un secret. Entrez une paire nom-valeur, puis choisissez Ajouter un secret pour ajouter un secret.

 Tip

[Pour plus d'informations, consultez : LogConfiguration SecretOptions.](#)

23. Choisissez Page suivante.

24. Pour la révision de la définition du Job, passez en revue les étapes de configuration. Si vous devez apporter des modifications, choisissez Modifier. Lorsque vous avez terminé, choisissez Créer une définition de tâche.

Création d'une définition de tâche à nœud unique sur les ressources Fargate

Procédez comme suit pour créer une définition de tâche à nœud unique sur les AWS Fargate ressources.

Pour créer une nouvelle définition de tâche sur les ressources Fargate :

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation supérieure, choisissez le Région AWS à utiliser.

3. Dans le volet de navigation, choisissez Définitions de tâches.
4. Choisissez Créer.
5. Pour Type d'orchestration, choisissez Fargate. Pour de plus amples informations, veuillez consulter [Environnements informatiques Fargate](#).
6. Dans Nom, entrez un nom unique pour votre définition de tâche. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
7. (Facultatif) Pour le délai d'exécution, entrez la valeur du délai d'expiration (en secondes). Le délai d'exécution est le délai avant la fin d'une tâche inachevée. Si une tentative dépasse le délai imparti, elle est arrêtée et passe à un FAILED statut. Pour de plus amples informations, veuillez consulter [Expiration des délais d'exécution des tâches](#). La valeur minimale est 60 secondes.
8. (Facultatif) Activez la priorité de planification. Entrez une valeur de priorité de planification comprise entre 0 et 100. Les valeurs les plus élevées ont une priorité plus élevée que les valeurs faibles.
9. (Facultatif) Développez les balises, puis choisissez Ajouter une balise pour ajouter des balises à la ressource. Activez Propager les balises pour propager les balises à partir de la tâche et de la définition de la tâche.
10. Dans la section de configuration de la plateforme Fargate :
 - a. Pour Runtime Platform, choisissez l'architecture de l'environnement de calcul.
 - b. Pour Famille de systèmes d'exploitation, choisissez le système d'exploitation pour l'environnement informatique.
 - c. Pour Architecture du processeur, choisissez l'architecture vCPU.
 - d. Pour la version de la plateforme Fargate, LATEST entrez ou une version d'environnement d'exécution spécifique.
 - e. (Facultatif) Activez l'option Attribuer une adresse IP publique pour attribuer une adresse IP publique à une interface réseau de tâches Fargate. Pour qu'une tâche exécutée dans un sous-réseau privé envoie du trafic sortant vers Internet, le sous-réseau privé nécessite qu'une passerelle NAT soit attachée pour acheminer les demandes vers Internet. Vous pouvez le faire afin de pouvoir extraire des images de conteneurs. Pour plus d'informations, consultez [Mise en réseau des tâches Amazon ECS](#) dans le Guide du développeur Amazon Elastic Container Service.

- f. (Facultatif) Pour Stockage éphémère, entrez la quantité de stockage éphémère à allouer à la tâche. La quantité de stockage éphémère doit être comprise entre 21 GiB et 200 GiB. Par défaut, 20 GiB de stockage éphémère sont alloués si vous ne saisissez aucune valeur.

 Note

Le stockage éphémère nécessite la version 1.4 ou ultérieure de la plateforme Fargate.

- g. Pour le rôle d'exécution, spécifiez un rôle IAM qui autorise le conteneur Amazon ECS et les agents Fargate à AWS effectuer des appels d'API en votre nom. Cette fonctionnalité utilise les rôles IAM d'Amazon ECS pour la fonctionnalité des tâches. Pour plus d'informations, notamment sur les prérequis de configuration, consultez les [rôles IAM d'exécution de tâches Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.
- h. Pour les tentatives de tâche, entrez le nombre de AWS Batch tentatives de transfert de la tâche vers un RUNNABLE statut. Entrez un nombre compris entre 1 et 10.
- i. (Facultatif) Pour les conditions de la stratégie Réessayer, choisissez Ajouter une évaluation à la sortie. Entrez au moins une valeur de paramètre, puis choisissez une action. Pour chaque ensemble de conditions, l'action doit être définie sur Réessayer ou sur Quitter. Ces actions signifient ce qui suit :
- Réessayer — AWS Batch Réessaie jusqu'à ce que le nombre de tentatives de travail que vous avez spécifié soit atteint.
 - Quitter : AWS Batch arrête de réessayer la tâche.

 Important

Si vous choisissez Ajouter une évaluation à la sortie, vous devez configurer au moins un paramètre et choisir une action ou choisir Supprimer l'évaluation à la sortie.

11. Choisissez Page suivante.

12. Dans la section Configuration du conteneur :

- a. Pour Image , choisissez l'image Docker à utiliser pour votre tâche. Par défaut, les images dans le registre Docker Hub sont disponibles. Vous pouvez également spécifier

d'autres référentiels avec `repository-url/image:tag`. Le nom peut contenir jusqu'à 225 caractères. Il peut contenir des majuscules et des minuscules, des chiffres, des tirets (-), des traits de soulignement (_), deux points (:), des points (.), des barres obliques (/) et des signes dièse (#). Ce paramètre se mappe à Image dans la section [Create a container](#) (Création d'un conteneur) de [Docker Remote API](#) (L'API Docker à distance) et le paramètre IMAGE de [docker run](#).

 Note

L'architecture des images Docker doit correspondre à l'architecture du processeur des ressources de calcul sur lesquelles elles sont planifiées. Par exemple, des images Docker basées sur ARM ne peuvent s'exécuter que sur des ressources de calcul basées sur ARM.

- Les images contenues dans les référentiels publics Amazon ECR utilisent les conventions complètes `registry/repository[:tag]` ou de `registry/repository[@digest]` dénomination (par exemple, `public.ecr.aws/registry_alias/my-web-app:latest`).
 - Les images contenues dans les référentiels Amazon ECR utilisent la convention de `registry/repository[:tag]` dénomination complète (par exemple, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`).
 - Les images dans les référentiels officiels sur Docker Hub utilisent un nom unique (par exemple, `ubuntu` ou `mongo`).
 - Les images dans les autres référentiels sur Docker Hub sont qualifiées par un nom d'organisation (par exemple, `amazon/amazon-ecs-agent`).
 - Les images dans les autres référentiels en ligne sont qualifiées par un nom de domaine (par exemple, `quay.io/assemblyline/ubuntu`).
- b. Pour Command, entrez les commandes dans le champ en tant qu'équivalent dans un tableau de chaînes JSON.

Ce paramètre correspond à Cmd dans la section relative à la [création d'un conteneur](#) de [Docker Remote API](#) et au paramètre COMMAND de [docker run](#). Pour plus d'informations sur le Docker CMD paramètre, consultez <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note

Vous pouvez utiliser des valeurs par défaut pour la substitution de paramètres et des espaces réservés dans votre commande. Pour de plus amples informations, veuillez consulter [Paramètres](#).

- c. (Facultatif) Ajoutez des paramètres à la définition de tâche sous forme de mappages nom-valeur pour remplacer les valeurs par défaut de définition de tâche. Pour ajouter un paramètre, procédez comme suit :
- Pour Paramètres, choisissez Ajouter des paramètres, entrez une paire nom-valeur, puis choisissez Ajouter un paramètre.

 Important

Si vous choisissez Ajouter un paramètre, vous devez configurer au moins un paramètre ou choisir Supprimer le paramètre

- d. Dans la section Configuration de l'environnement :
- i. Pour la configuration du rôle Job, choisissez un rôle IAM autorisé à accéder au AWS APIs. Cette fonctionnalité utilise les rôles IAM d'Amazon ECS pour la fonctionnalité des tâches. Pour plus d'informations, consultez [Rôles IAM pour les tâches](#) dans le Guide du développeur Amazon Elastic Container Service.

 Note

Seuls les rôles dotés d'une relation de confiance avec Amazon Elastic Container Service Task Role sont présentés ici. Pour plus d'informations sur la création d'un rôle IAM pour vos AWS Batch tâches, consultez la section [Création d'un rôle IAM et d'une politique pour vos tâches](#) dans le manuel Amazon Elastic Container Service Developer Guide.

- ii. Pour v CPUs, entrez le nombre de v CPUs à réserver pour le conteneur. Ce paramètre correspond à CpuShares dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option `--cpu-shares` correspond à [docker run](#). Chaque

processeur virtuel est équivalente à 1 024 parts de processeur. Vous devez spécifier au moins un vCPU.

- iii. Pour Mémoire, entrez la limite de mémoire disponible pour le conteneur. Si votre conteneur tente de dépasser la mémoire spécifiée ici, il est arrêté. Ce paramètre correspond à Memory dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option `--memory` correspond à [docker run](#). Vous devez spécifier au moins 4 Mio de mémoire pour une tâche.

Si vous utilisez la surveillance du temps GuardDuty d'exécution, l'agent de GuardDuty sécurité est légèrement surchargé en mémoire. La limite de mémoire doit donc inclure la taille de l'agent GuardDuty de sécurité. Pour plus d'informations sur les limites de mémoire des agents de GuardDuty sécurité, consultez la section [Limites de processeur et de mémoire](#) dans le Guide de GuardDuty l'utilisateur. Pour plus d'informations sur les meilleures pratiques, consultez la section [Comment corriger les erreurs liées au manque de mémoire sur mes tâches Fargate après avoir activé la surveillance](#) du temps d'exécution dans le manuel Amazon ECS Developer Guide.

 Note

Pour optimiser l'utilisation de vos ressources, donnez la priorité à la mémoire pour les tâches d'un type d'instance spécifique. Pour de plus amples informations, veuillez consulter [Gestion de la mémoire des ressources informatiques](#).

- e. (Facultatif) Pour les variables d'environnement, choisissez Ajouter une variable d'environnement pour ajouter des variables d'environnement sous forme de paires nom-valeur. Ces variables sont transmises au conteneur.
 - f. (Facultatif) Pour les secrets, choisissez Ajouter un secret pour ajouter des secrets sous forme de paires nom-valeur. Ces secrets sont exposés dans le contenant. [Pour plus d'informations, consultez : LogConfiguration SecretOptions](#).
 - g. Choisissez Page suivante.
13. (Facultatif) Dans la section de configuration Linux :
- a. Pour Utilisateur, entrez un nom d'utilisateur à utiliser dans le conteneur.
 - b. Activez l'option Activer le processus d'initialisation pour exécuter un processus d'initialisation dans le conteneur. Ce processus transmet les signaux et récolte les processus.

- c. Activez l'option Enable read only filesystem (Activer le système de fichiers en lecture seule) pour supprimer l'accès en écriture au volume.
 - d. (Facultatif) Développez la configuration supplémentaire.
 - e. Pour la configuration des points de montage, choisissez Ajouter une configuration de points de montage pour ajouter des points de montage pour les volumes de données. Vous devez spécifier le volume source et le chemin du conteneur. Ces points de montage sont transmis Docker daemon à une instance de conteneur.
 - f. Pour la configuration des volumes, choisissez Ajouter un volume pour créer une liste de volumes à transmettre au conteneur. Entrez un nom et un chemin source pour le volume, puis choisissez Ajouter un volume.
 - g. Dans la section Propriétés de la tâche :
 - i. Pour Rôle d'exécution - conditionnel, choisissez un rôle pour autoriser les agents Amazon ECS à effectuer des appels d' AWS API en votre nom. Pour plus d'informations sur la création d'un rôle d'exécution, consultez [Tutoriel : Création du rôle d'exécution IAM](#).
 - ii. Choisissez la commande Enable ECS execute pour permettre l'accès direct au shell du conteneur Amazon ECS et contourner le système d'exploitation hôte. Vous devez choisir un rôle de tâche.
-  **Important**

La commande ECS execute nécessite que le système de fichiers soit accessible en écriture.
- iii. Pour le rôle Task, choisissez un rôle Amazon ECS Identity and Access Management (IAM) pour autoriser le conteneur à effectuer des appels d' AWS API en votre nom. Pour plus d'informations, consultez le [rôle IAM de la tâche Amazon ECS](#) dans le manuel du développeur Amazon Elastic Container Service.
 - h. Dans la section Configuration de la journalisation :
 - i. (Facultatif) Pour Log driver, choisissez le log driver à utiliser. [Pour plus d'informations sur les pilotes de journal disponibles, voir : LogDriverLogConfiguration](#).

 Note

Par défaut, le pilote de `awslogs` journal est utilisé.

- ii. (Facultatif) Pour Options, choisissez Ajouter une option pour ajouter une option. Entrez une paire nom-valeur, puis choisissez Ajouter une option.
- iii. (Facultatif) Pour Secrets, choisissez Ajouter un secret pour ajouter un secret. Entrez ensuite une paire nom-valeur, puis choisissez Ajouter un secret.

 Tip

[Pour plus d'informations, consultez : LogConfiguration SecretOptions.](#)

14. Choisissez Page suivante.
15. Pour la révision de la définition du Job, passez en revue les étapes de configuration. Si vous devez apporter des modifications, choisissez Modifier. Lorsque vous avez terminé, choisissez Créer une définition de tâche.

Création d'une définition de tâche à nœud unique sur les ressources Amazon EKS

Procédez comme suit pour créer une définition de tâche à nœud unique sur Amazon Elastic Kubernetes Service (Amazon EKS).

Pour créer une nouvelle définition de tâche sur les ressources Amazon EKS :

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation supérieure, choisissez le Région AWS à utiliser.
3. Dans le volet de navigation, choisissez Définitions de tâches.
4. Choisissez Créer.
5. Pour le type d'orchestration, choisissez Elastic Kubernetes Service (EKS).
6. Dans Nom, entrez un nom unique pour votre définition de tâche. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).

7. (Facultatif) Dans le champ Délai d'exécution, entrez la valeur du délai d'expiration (en secondes). Le délai d'exécution est le délai avant la fin d'une tâche inachevée. Si une tentative dépasse le délai imparti, elle est arrêtée et passe à un FAILED statut. Pour de plus amples informations, veuillez consulter [Expiration des délais d'exécution des tâches](#). La valeur minimale est 60 secondes.
8. (Facultatif) Activez la priorité de planification. Entrez une valeur de priorité de planification comprise entre 0 et 100. Les valeurs les plus élevées ont une priorité plus élevée que les valeurs faibles.
9. (Facultatif) Développez les balises, puis choisissez Ajouter une balise pour ajouter des balises à la ressource.
10. Choisissez Page suivante.
11. Dans la section des podpropriétés d'EKS :
 - a. Pour le nom du compte de service, entrez un compte qui fournit une identité pour les processus exécutés dans un pod.
 - b. Activez le réseau hôte pour utiliser le modèle de Kubernetes pod réseau et ouvrez un port d'écoute pour les connexions entrantes. Désactivez ce paramètre pour les communications sortantes uniquement.
 - c. Pour la politique DNS, choisissez l'une des options suivantes :
 - Aucune valeur (null) : pod ignore les paramètres DNS de l'environnement Kubernetes.
 - Par défaut : pod hérite de la configuration de résolution des noms du nœud sur lequel il s'exécute.

 Note

Si aucune politique DNS n'est spécifiée, Default (valeur par défaut) n'est pas la politique DNS par défaut. Au lieu de cela, ClusterFirst est utilisé.

- ClusterFirst— Toute requête DNS qui ne correspond pas au suffixe de domaine du cluster configuré est transmise au serveur de noms en amont hérité du nœud.
 - ClusterFirstWithHostNet— À utiliser si le réseau hôte est activé.
- d. (Facultatif) Pour Volumes, sélectionnez Ajouter un volume, puis :
 - i. Ajoutez un nom à votre volume.
 - ii. (Facultatif) Ajoutez le chemin de l'hôte pour le répertoire sur l'hôte.

- iii. (Facultatif) Ajoutez une limite de taille et une limite de taille pour configurer un [Kubernetes EmptyDir](#).
 - iv. (Facultatif) Indiquez un nom secret pour le module et indiquez si le secret est facultatif.
 - v. (Facultatif) Définissez un nom de réclamation pour associer une [réclamation de volume persistant](#) Kubernetes au pod, et indiquez si elle est en lecture seule.
- e. (Facultatif) Pour les étiquettes du module, choisissez Ajouter des étiquettes du module, puis entrez une paire nom-valeur.

 Important

Le préfixe d'une étiquette de module ne peut pas contenir `kubernetes.io/k8s.io/`, `oubatch.amazonaws.com/`.

- f. (Facultatif) Pour les annotations Pod, choisissez Ajouter des annotations, puis entrez une paire nom-valeur.

 Important

Le préfixe d'une annotation de module ne peut pas contenir `kubernetes.io/k8s.io/`, `oubatch.amazonaws.com/`.

- g. Choisissez Page suivante.
- h. Dans la section Configuration du conteneur :
- i. Dans Nom, entrez un nom unique pour le conteneur. Le nom doit commencer par une lettre ou un chiffre et peut comporter jusqu'à 63 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres et des traits d'union (-).
 - ii. Dans Image, choisissez l'Dockerimage à utiliser pour votre travail. Par défaut, les images dans le registre Docker Hub sont disponibles. Vous pouvez également spécifier d'autres référentiels avec *repository-url/image:tag*. Le nom peut comporter jusqu'à 255 caractères. Il peut contenir des majuscules et des minuscules, des chiffres, des tirets (-), des traits de soulignement (_), deux points (:), des points (.), des barres obliques (/) et des signes dièse (#). Ce paramètre correspond Image à la section [Créer un conteneur](#) de l'[API Docker Remote](#) et au IMAGE paramètre de [docker run](#)

 Note

L'architecture des images Docker doit correspondre à l'architecture du processeur des ressources de calcul sur lesquelles elles sont planifiées. Par exemple, des images Docker basées sur ARM ne peuvent s'exécuter que sur des ressources de calcul basées sur ARM.

- Les images contenues dans les référentiels publics Amazon ECR utilisent les conventions complètes `registry/repository[:tag]` ou de `registry/repository[@digest]` dénomination (par exemple, `public.ecr.aws/registry_alias/my-web-app:latest`).
 - Les images contenues dans les référentiels Amazon ECR utilisent la convention de `registry/repository[:tag]` dénomination complète (par exemple, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`).
 - Les images dans les référentiels officiels sur Docker Hub utilisent un nom unique (par exemple, `ubuntu` ou `mongo`).
 - Les images dans les autres référentiels sur Docker Hub sont qualifiées par un nom d'organisation (par exemple, `amazon/amazon-ecs-agent`).
 - Les images dans les autres référentiels en ligne sont qualifiées par un nom de domaine (par exemple, `quay.io/assemblyline/ubuntu`).
- iii. (Facultatif) Pour la politique d'extraction d'images, choisissez quand les images sont extraites.
 - iv. (Facultatif) Dans Commande, entrez une JSON commande à transmettre au conteneur.
 - v. (Facultatif) Dans Arguments, entrez les arguments à transmettre au conteneur. Si aucun argument n'est fourni, la commande `container image` est utilisée.
- i. (Facultatif) Vous pouvez ajouter des paramètres à la définition de tâche sous forme de mappages nom-valeur pour remplacer les valeurs par défaut de définition de tâche. Pour ajouter un paramètre, procédez comme suit :
 - Pour Paramètres, entrez une paire nom-valeur, puis choisissez Ajouter un paramètre.

 Important

Si vous choisissez Ajouter un paramètre, vous devez configurer au moins un paramètre ou choisir Supprimer le paramètre

- j. Dans la section Configuration de l'environnement :
 - i. Pour v CPUs, entrez le nombre de v CPUs à réserver pour le conteneur. Ce paramètre correspond à CpuShares dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option `--cpu-shares` correspond à [docker run](#). Chaque processeur virtuel est équivalente à 1 024 parts de processeur. Vous devez spécifier au moins un vCPU.
 - ii. Pour Mémoire, entrez la limite de mémoire disponible pour le conteneur. Si votre conteneur tente de dépasser la mémoire spécifiée ici, il est arrêté. Ce paramètre correspond à Memory dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option `--memory` correspond à [docker run](#). Vous devez spécifier au moins 4 Mio de mémoire pour une tâche.

 Note

Pour optimiser l'utilisation de vos ressources, donnez la priorité à la mémoire pour les tâches d'un type d'instance spécifique. Pour de plus amples informations, veuillez consulter [Gestion de la mémoire des ressources informatiques](#).

- k. (Facultatif) Pour les variables d'environnement, choisissez Ajouter une variable d'environnement pour ajouter des variables d'environnement sous forme de paires nom-valeur. Ces variables sont transmises au conteneur.
- l. (Facultatif) Pour le montage du volume :
 - i. Choisissez Ajouter un montage de volume.
 - ii. Entrez un nom, puis un chemin de montage dans le conteneur dans lequel le volume est monté. Entrez a SubPath pour spécifier un sous-chemin à l'intérieur du volume référencé au lieu de sa racine.
 - iii. Choisissez Lecture seule pour supprimer les autorisations d'écriture sur le volume.
 - iv. Choisissez Ajouter un montage de volume.

- m. (Facultatif) Pour Exécuter en tant qu'utilisateur, entrez un ID utilisateur pour exécuter le processus de conteneur.

 Note

L'ID utilisateur doit figurer dans l'image pour que le conteneur puisse fonctionner.

- n. (Facultatif) Pour Exécuter en tant que groupe, entrez un ID de groupe pour exécuter le runtime du processus de conteneur.

 Note

L'ID de groupe doit exister dans l'image pour que le conteneur puisse fonctionner.

- o. (Facultatif) Pour accorder au conteneur de votre tâche des autorisations élevées sur l'instance hôte (comme pour l'rootutilisateur), faites glisser le curseur Privileged vers la droite. Ce paramètre correspond à Privileged dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option `--privileged` correspond à [docker run](#).
- p. (Facultatif) Activez le système de fichiers racine en lecture seule pour supprimer l'accès en écriture au système de fichiers racine.
- q. (Facultatif) Activez Exécuter en tant qu'utilisateur non-root pour exécuter les conteneurs en pod tant qu'utilisateur non-root.

 Note

Si l'option Exécuter en tant que non-root est activée, l'image est kubelet validée au moment de l'exécution pour vérifier qu'elle ne s'exécute pas avec l'UID 0.

- r. Choisissez Page suivante.
12. Pour la révision de la définition du Job, passez en revue les étapes de configuration. Si vous devez apporter des modifications, choisissez Modifier. Lorsque vous avez terminé, choisissez Créer une définition de tâche.

Créez une définition de tâche à nœud unique avec plusieurs conteneurs sur les ressources Amazon EC2

Procédez comme suit pour créer une définition de tâche à nœud unique avec plusieurs conteneurs sur les ressources Amazon Elastic Compute Cloud (Amazon EC2).

Pour créer une nouvelle définition de tâche sur Amazon EC2 Resources, procédez comme suit :

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation, choisissez le Région AWS à utiliser.
3. Dans le volet de navigation, choisissez Définitions de tâches.
4. Choisissez Créer.
5. Pour le type d'orchestration, choisissez Amazon Elastic Compute Cloud (Amazon EC2).
6. Pour la structure de définition du Job, désactivez le traitement de la structure Use legacy ContainerProperties.
7. Pour la configuration de la EC2 plate-forme, désactivez l'option Activer le traitement parallèle à nœuds multiples.
8. Choisissez Suivant.
9. Dans la section Configuration générale, entrez ce qui suit :
 - a. Dans Nom, entrez un nom unique pour votre définition de tâche. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
 - b. Pour Execution timeout (facultatif), entrez la valeur du délai (en secondes). Le délai d'exécution est le délai avant la fin d'une tâche inachevée. Si une tentative dépasse le délai imparti, elle est arrêtée et passe à un FAILED statut. Pour de plus amples informations, veuillez consulter [Expiration des délais d'exécution des tâches](#). La valeur minimale est 60 secondes.
 - c. Activez la priorité de planification (facultatif). Entrez une valeur de priorité de planification comprise entre 0 et 100. Les valeurs les plus élevées reçoivent une priorité plus élevée.
 - d. Développer les balises : facultatif, puis choisissez Ajouter une balise pour ajouter des balises à la ressource. Entrez une clé et une valeur facultative, puis choisissez Add tag (Ajouter une balise).
 - e. Activez les balises de propagation pour propager les balises de la tâche et de la définition de la tâche vers la tâche Amazon ECS.

10. Dans la section Stratégie de réessayer - facultatif, entrez ce qui suit :

- a. Pour les tentatives de tâche, entrez le nombre de AWS Batch tentatives de transfert de la tâche vers le RUNNABLE statut. Entrez un nombre compris entre 1 et 10.
- b. Pour les conditions de la stratégie Réessayer, choisissez Ajouter une évaluation à la sortie. Entrez au moins une valeur de paramètre, puis choisissez une action. Pour chaque ensemble de conditions, l'action doit être définie sur Réessayer ou sur Quitter. Ces actions signifient ce qui suit :
 - Réessayer — AWS Batch Réessaie jusqu'à ce que le nombre de tentatives de travail que vous avez spécifié soit atteint.
 - Quitter : AWS Batch arrête de réessayer la tâche.

 Important

Si vous choisissez Ajouter une évaluation à la sortie, vous devez configurer au moins un paramètre et soit choisir une action, soit choisir Supprimer l'évaluation à la sortie.

11. Dans la section Propriétés de la tâche, entrez les informations suivantes :

- a. Pour Rôle d'exécution - conditionnel, choisissez un rôle pour autoriser les agents Amazon ECS à effectuer des appels d' AWS API en votre nom. Pour plus d'informations sur la création d'un rôle d'exécution, consultez [Tutoriel : Création du rôle d'exécution IAM](#).
- b. Choisissez la commande Enable ECS execute pour permettre l'accès direct au shell du conteneur Amazon ECS et contourner le système d'exploitation hôte. Vous devez choisir un rôle de tâche.

 Important

La commande ECS execute nécessite que le système de fichiers soit accessible en écriture.

- c. Pour le rôle Task, choisissez un rôle Amazon ECS Identity and Access Management (IAM) pour autoriser le conteneur à effectuer des appels d' AWS API en votre nom. Pour plus d'informations, consultez le [rôle IAM de la tâche Amazon ECS](#) dans le manuel du développeur Amazon Elastic Container Service.

- d. Pour le mode `IPCHost`, sélectionnez `task`, ou `none`. Si cette option `host` est spécifiée, tous les conteneurs inclus dans les tâches qui ont spécifié le mode `IPC` hôte sur la même instance de conteneur partagent les mêmes ressources `IPC` avec l'instance Amazon EC2 hôte. Si une tâche est spécifiée, tous les conteneurs qui se trouvent dans la tâche spécifiée partagent les mêmes ressources `IPC`. Si aucune n'est spécifiée, les ressources `IPC` présentes dans les conteneurs d'une tâche sont privées et ne sont pas partagées avec les autres conteneurs d'une tâche ou de l'instance de conteneur. Si aucune valeur n'est spécifiée, le partage de l'espace de noms de ressource `IPC` dépend du paramètre du démon Docker sur l'instance de conteneur.
 - e. Pour le mode `PID`, choisissez `host` ou `task`. Par exemple, la surveillance des sidecars peut avoir besoin de `pidMode` pour accéder à des informations sur d'autres conteneurs exécutés dans le cadre de la même tâche. Si cette `host` option est spécifiée, tous les conteneurs des tâches qui ont spécifié le mode `PID` hôte sur la même instance de conteneur partagent le même espace de noms de processus avec l'instance Amazon EC2 hôte. Si `task` est spécifié, tous les conteneurs de la tâche spécifiée partagent le même espace de noms de processus. Si aucune valeur n'est spécifiée, la valeur par défaut est un espace de noms privé pour chaque conteneur.
12. Dans la section **Ressources consommables**, entrez ce qui suit :
- a. Entrez un nom unique et la valeur demandée.
 - b. Vous pouvez ajouter d'autres ressources consommables en choisissant **Ajouter une ressource consommable**.
13. Dans la section **Stockage**, entrez les informations suivantes :
- a. Entrez un nom et un chemin source pour le volume, puis choisissez **Ajouter un volume**. Vous pouvez également choisir d'activer **Activer EFS**.
 - b. Vous pouvez ajouter d'autres volumes en choisissant **Ajouter un volume**.
14. Pour **Paramètres**, choisissez **Ajouter des paramètres** pour ajouter des espaces réservés de substitution de paramètres sous forme de paires clé et valeur facultatives.
15. Choisissez **Page suivante**.
16. Dans la section **Configuration du conteneur** :
- a. Pour **Name (Nom)**, saisissez un nom pour le conteneur.
 - b. Pour le conteneur essentiel, activez si le contenant est essentiel.

- c. Dans Image, choisissez l'`Dockerimage` à utiliser pour votre travail. Par défaut, les images dans le registre Docker Hub sont disponibles. Vous pouvez également spécifier d'autres référentiels avec `repository-url/image:tag`. Le nom peut contenir jusqu'à 225 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-), des traits de soulignement (_), des signes deux-points (:), des barres obliques (/) et des signes numériques (#). Ce paramètre se mappe à Image dans la section [Create a container](#) (Création d'un conteneur) de [Docker Remote API](#) (L'API Docker à distance) et le paramètre IMAGE de [docker run](#).

 Note

L'architecture des images Docker doit correspondre à l'architecture du processeur des ressources de calcul sur lesquelles elles sont planifiées. Par exemple, des images Docker basées sur ARM ne peuvent s'exécuter que sur des ressources de calcul basées sur ARM.

- Les images contenues dans les référentiels publics Amazon ECR utilisent les conventions complètes `registry/repository[:tag]` ou de `registry/repository[@digest]` dénomination (par exemple, `public.ecr.aws/registry_alias/my-web-app:latest`).
 - Les images contenues dans les référentiels Amazon ECR utilisent la convention de `registry/repository[:tag]` dénomination complète (par exemple, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`).
 - Les images dans les référentiels officiels sur Docker Hub utilisent un nom unique (par exemple, `ubuntu` ou `mongo`).
 - Les images dans les autres référentiels sur Docker Hub sont qualifiées par un nom d'organisation (par exemple, `amazon/amazon-ecs-agent`).
 - Les images dans les autres référentiels en ligne sont qualifiées par un nom de domaine (par exemple, `quay.io/assemblyline/ubuntu`).
- d. Pour les besoins en ressources, configurez chacune des options suivantes :
- i. Pour v CPUs, choisissez le numéro de CPUs pour le conteneur.
 - ii. Pour Mémoire, choisissez la quantité de mémoire pour le conteneur.
 - iii. Pour le GPU, en option, choisissez le nombre de GPUs pour le conteneur.

- e. Dans Interface, saisissez le nom d'utilisateur à utiliser à l'intérieur du conteneur.
- f. Activez l'option Enable read only filesystem (Activer le système de fichiers en lecture seule) pour supprimer l'accès en écriture au volume.
- g. Activez Privileged pour accorder au conteneur de tâches des autorisations élevées sur l'instance hôte, comme pour l'utilisateur root.
- h. Pour Command, saisissez les commandes dans le champ comme leur équivalent sous forme de tableau de chaînes JSON.

Ce paramètre correspond à Cmd dans la section relative à la [création d'un conteneur](#) de [Docker Remote API](#) et au paramètre COMMAND de [docker run](#). Pour plus d'informations sur le Docker CMD paramètre, consultez <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note

Vous pouvez utiliser des valeurs par défaut pour la substitution de paramètres et des espaces réservés dans votre commande. Pour de plus amples informations, veuillez consulter [Paramètres](#).

- i. Pour les informations d'identification du référentiel (facultatif), entrez l'ARN du secret contenant vos informations d'identification.
- j. Pour les variables d'environnement (facultatif), choisissez Ajouter des variables d'environnement pour ajouter des variables d'environnement à transmettre au conteneur.
- k. Dans la section Paramètres Linux - facultatif :
 - i. Activez l'option Activer le processus d'initialisation pour exécuter un processus d'initialisation dans le conteneur.
 - ii. Pour Taille de la mémoire partagée, entrez la taille (en MiB) du volume /dev/shm
 - iii. Pour Taille maximale de swap, entrez la quantité totale de mémoire d'échange (en MiB) que le conteneur peut utiliser.
 - iv. Pour Swappiness, entrez une valeur comprise entre 0 et 100 pour indiquer le comportement de swappiness du conteneur. Si vous ne spécifiez aucune valeur et que le swap est activé, la valeur par défaut est 60.
 - v. Pour Appareils, choisissez Ajouter un appareil pour ajouter un appareil :

- A. Sous Chemin du conteneur, spécifiez le chemin d'accès de l'instance du conteneur pour exposer l'appareil mappé à l'instance de l'hôte. Si vous laissez ce champ vide, le chemin de l'hôte est utilisé dans le conteneur.
 - B. Sous Host path (Chemin de l'hôte), spécifiez le chemin d'accès d'un appareil dans l'instance de l'hôte.
 - C. Pour Autorisations, choisissez une ou plusieurs autorisations à appliquer à l'appareil. Les autorisations disponibles sont READ, WRITE et MKNOD.
- vi. Pour Tmpfs, choisissez Ajouter des tmpfs pour ajouter un montage. tmpfs

I.

 Note

L'enregistrement de Firelens doit être effectué dans un conteneur dédié. Pour configurer la journalisation de Firelens :

- Dans chaque conteneur, à l'exception de votre conteneur Firelens dédié, configurez le pilote Logging sur `awsfirelens`
- Dans votre conteneur Firelens, définissez la configuration Firelens (facultative) et la configuration de journalisation (facultative) pour la destination de journalisation.

Dans la section Configuration de Firelens - optionnelle :

 Important

AWS Batch applique le mode `host réseau` aux tâches Amazon ECS autres que MNP et Fargate. [L'utilisateur root est requis](#) pour Amazon ECS Firelens. Lors de l'exécution de tâches utilisant le mode `host réseau`, Amazon ECS déconseille d'exécuter des conteneurs en utilisant l'utilisateur root (UID 0) pour [une meilleure sécurité](#). Par conséquent, toutes les tâches ECS autres que MNP et Fargate avec journalisation Firelens ne respecteront pas les meilleures pratiques en matière de sécurité.

- i. Pour Type, choisissez `fluentd` soit `fluentbit`.
- ii. Dans le champ Options, entrez la name/value paire de l'option. Vous pouvez ajouter d'autres options en utilisant l'option Ajouté.

- m. Dans la section Configuration de la journalisation - facultatif :
- i. Pour Log driver, choisissez le log driver à utiliser. [Pour plus d'informations sur les pilotes de journal disponibles, voir : LogDriverLogConfiguration.](#)

 Note

Par défaut, le pilote de awslogs journal est utilisé.

- ii. Pour Options, choisissez Ajouter une option pour ajouter une option. Entrez une paire nom-valeur, puis choisissez Ajouter une option.
- iii. Pour Secrets, choisissez Ajouter un secret. Entrez une paire nom-valeur, puis choisissez Ajouter un secret pour ajouter un secret.

 Tip

[Pour plus d'informations, consultez : LogConfiguration SecretOptions.](#)

- n. Pour Points de montage (facultatif), choisissez Ajouter des points de montage pour ajouter des points de montage pour les volumes de données. Vous devez spécifier le volume source et le chemin du conteneur.
- o. Pour les secrets (facultatif), choisissez Ajouter un secret pour ajouter un secret. Entrez ensuite une paire nom-valeur, puis choisissez Ajouter un secret.

 Tip

[Pour plus d'informations, consultez : LogConfiguration SecretOptions.](#)

- p. Pour Ulimits (facultatif), choisissez Add ulimit pour ajouter une ulimits valeur au conteneur. Entrez les valeurs Nom, Limite souple et Limite stricte, puis choisissez Ajouter une limite.
- q. Pour Dépendances (facultatif), choisissez Ajouter des dépendances de conteneur. Choisissez le nom du conteneur et son état pour déterminer quand ce conteneur démarre.
17. Si vous n'avez configuré qu'un seul conteneur, vous devez choisir Ajouter un conteneur et terminer la configuration du nouveau conteneur. Sinon, choisissez Next pour passer en revue.

Création d'une définition de tâche parallèle à nœuds multiples

Avant de pouvoir exécuter des tâches AWS Batch, vous devez créer une définition de tâche. Ce processus varie légèrement entre les tâches parallèles à nœud unique et à nœuds multiples. Cette rubrique explique spécifiquement comment créer une définition de tâche pour une tâche parallèle à AWS Batch plusieurs nœuds (également appelée planification en groupe). Pour de plus amples informations, veuillez consulter [Tâches parallèles sur plusieurs nœuds](#).

Note

AWS Fargate ne prend pas en charge les tâches parallèles à nœuds multiples.

Table des matières

- [Tutoriel : Création d'une définition de tâche parallèle à nœuds multiples sur Amazon Resources EC2](#)

Tutoriel : Création d'une définition de tâche parallèle à nœuds multiples sur Amazon Resources EC2

Pour créer une définition de tâche parallèle à plusieurs nœuds sur les ressources Amazon Elastic Compute Cloud (Amazon EC2).

Note

Pour créer une définition de tâche à nœud unique, consultez [Création d'une définition de tâche à nœud unique sur Amazon Resources EC2](#).

Pour créer une définition de tâche parallèle à nœuds multiples sur Amazon EC2 Resources :

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation, sélectionnez le Région AWS à utiliser.
3. Dans le volet de navigation, choisissez Job definitions.
4. Choisissez Créer.
5. Pour le type d'orchestration, choisissez Amazon Elastic Compute Cloud (Amazon EC2).

6. Pour Enable multi-node parallel, activez le mode multi-node parallel.
7. Dans Nom, entrez un nom unique pour votre définition de tâche. Le nom peut comporter jusqu'à 128 caractères et peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
8. (Facultatif) Pour le délai d'exécution, spécifiez le nombre maximal de secondes pendant lesquelles vous souhaitez que les tentatives de travail s'exécutent. Si une tentative dépasse le délai imparti, elle est arrêtée et passe à un FAILED statut. Pour de plus amples informations, veuillez consulter [Expiration des délais d'exécution des tâches](#).
9. (Facultatif) Activez la priorité de planification. Entrez une valeur de priorité de planification comprise entre 0 et 100. Les valeurs les plus élevées ont une priorité plus élevée que les valeurs faibles.
10. (Facultatif) Pour les tentatives de tâche, entrez le nombre de AWS Batch tentatives de transfert de la tâche vers le RUNNABLE statut. Entrez un nombre compris entre 1 et 10.
11. (Facultatif) Pour les conditions de la stratégie Réessayer, choisissez Ajouter une évaluation à la sortie. Entrez au moins une valeur de paramètre, puis choisissez une action. Pour chaque ensemble de conditions, l'action doit être définie sur Réessayer ou sur Quitter. Ces actions signifient ce qui suit :
 - Réessayer — AWS Batch Réessaie jusqu'à ce que le nombre de tentatives de travail que vous avez spécifié soit atteint.
 - Quitter : AWS Batch arrête de réessayer la tâche.

 Important

Si vous choisissez Ajouter une évaluation à la sortie, vous devez configurer au moins un paramètre et soit choisir une action, soit choisir Supprimer l'évaluation à la sortie.

12. (Facultatif) Développez les balises, puis choisissez Ajouter une balise pour ajouter des balises à la ressource. Entrez une clé et une valeur facultative, puis choisissez Ajouter une étiquette. Vous pouvez également activer les balises Propagate pour propager les balises de la tâche et de la définition de la tâche vers la tâche Amazon ECS.
13. Choisissez Page suivante.
14. Pour Nombre de nœuds, saisissez le nombre total de nœuds à utiliser pour la tâche.
15. Pour Nœud principal, saisissez l'index de nœud à utiliser pour le nœud principal. L'index de nœud principal par défaut est 0.

16. Pour Type d'instance, choisissez un type d'instance.

 Note

Le type d'instance que vous choisissez s'applique à tous les nœuds.

17. Pour Paramètres, choisissez Ajouter des paramètres pour ajouter des espaces réservés de substitution de paramètres sous forme de paires clé et valeur facultatives.

18. Dans la section Plages de nœuds :

- a. Sélectionnez Ajouter une plage de nœuds. Cela crée une section de plage de nœuds.
- b. Pour Nœuds cible, spécifiez la plage de votre groupe de nœuds, en utilisant la notation *range_start:range_end*.

Vous pouvez créer jusqu'à cinq plages de nœuds pour les nœuds que vous avez spécifiés pour votre tâche. Les plages de nœuds utilisent la valeur d'index d'un nœud et l'index de nœud commence à 0. Assurez-vous que la valeur de l'indice de fin de plage de votre groupe de nœuds final est inférieure d'un au nombre de nœuds que vous avez spécifié. Supposons, par exemple, que vous ayez spécifié 10 nœuds et que vous souhaitiez utiliser un seul groupe de nœuds. Ensuite, votre fourchette finale est 9.

- c. Dans Image, choisissez l'Dockerimage à utiliser pour votre travail. Par défaut, les images dans le registre Docker Hub sont disponibles. Vous pouvez également spécifier d'autres référentiels avec *repository-url/image:tag*. Le nom peut comporter jusqu'à 225 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-), des traits de soulignement (_), des signes deux-points (:), des barres obliques (/) et des signes numériques (#). Ce paramètre se mappe à Image dans la section [Create a container](#) (Création d'un conteneur) de [Docker Remote API](#) (L'API Docker à distance) et le paramètre IMAGE de [docker run](#).

 Note

L'architecture des images Docker doit correspondre à l'architecture du processeur des ressources de calcul sur lesquelles elles sont planifiées. Par exemple, des images Docker basées sur ARM ne peuvent s'exécuter que sur des ressources de calcul basées sur ARM.

- Les images contenues dans les référentiels publics Amazon ECR utilisent les conventions complètes `registry/repository[:tag]` ou de `registry/repository[@digest]` dénomination (par exemple, `public.ecr.aws/registry_alias/my-web-app:latest`).
 - Les images contenues dans les référentiels Amazon ECR utilisent la convention de `registry/repository[:tag]` dénomination complète. Par exemple, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`
 - Les images dans les référentiels officiels sur Docker Hub utilisent un nom unique (par exemple, `ubuntu` ou `mongo`).
 - Les images dans les autres référentiels sur Docker Hub sont qualifiées par un nom d'organisation (par exemple, `amazon/amazon-ecs-agent`).
 - Les images dans les autres référentiels en ligne sont qualifiées par un nom de domaine (par exemple, `quay.io/assemblyline/ubuntu`).
- d. Pour Command, saisissez les commandes dans le champ comme leur équivalent sous forme de tableau de chaînes JSON.

Ce paramètre correspond à `Cmd` dans la section relative à la [création d'un conteneur](#) de [Docker Remote API](#) et au paramètre `COMMAND` de [docker run](#). Pour plus d'informations sur le Docker CMD paramètre, consultez <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note

Vous pouvez utiliser des valeurs par défaut pour la substitution de paramètres et des espaces réservés dans votre commande. Pour de plus amples informations, veuillez consulter [Paramètres](#).

- e. Pour `v CPUs`, spécifiez le nombre de `v CPUs` à réserver pour le conteneur. Ce paramètre correspond à `CpuShares` dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option `--cpu-shares` correspond à [docker run](#). Chaque processeur virtuel est équivalente à 1 024 parts de processeur. Vous devez spécifier au moins un vCPU.
- f. Dans Mémoire, précisez la limite stricte (en Mio) de la mémoire à présenter au conteneur de la tâche. Si votre conteneur tente de dépasser la mémoire spécifiée ici, il est arrêté. Ce paramètre correspond à `Memory` dans la section [Create a container](#) (Créer un conteneur) de

l'[API Docker à distance](#) et l'option `--memory` correspond à [docker run](#). Vous devez spécifier au moins 4 Mio de mémoire pour une tâche.

 Note

Pour optimiser l'utilisation de vos ressources, vous pouvez fournir à vos tâches autant de mémoire que possible pour un type d'instance spécifique. Pour de plus amples informations, veuillez consulter [Gestion de la mémoire des ressources informatiques](#).

- g. (Facultatif) Dans Nombre de GPUs, spécifiez le nombre d'utilisations de GPUs votre travail. La tâche s'exécute sur un conteneur dont le nombre spécifié GPUs est épinglé à ce conteneur.
- h. (Facultatif) Pour le rôle Job, vous pouvez spécifier un rôle IAM qui fournit au conteneur de votre travail l'autorisation d'utiliser le AWS APIs. Cette fonctionnalité utilise les rôles IAM d'Amazon ECS pour la fonctionnalité des tâches. Pour plus d'informations, notamment sur les prérequis de configuration, consultez la section [Rôles IAM pour les tâches](#) du manuel Amazon Elastic Container Service Developer Guide.

 Note

Pour les tâches exécutées sur les ressources Fargate, un rôle de tâche est requis.

 Note

Seuls les rôles dotés d'une relation de confiance avec Amazon Elastic Container Service Task Role sont présentés ici. Pour plus d'informations sur la création d'un rôle IAM pour vos AWS Batch tâches, consultez la section [Création d'un rôle IAM et d'une politique pour vos tâches](#) dans le manuel Amazon Elastic Container Service Developer Guide.

- i. (Facultatif) Pour le rôle d'exécution, spécifiez un rôle IAM qui autorise les agents de conteneur Amazon ECS à effectuer des appels d' AWS API en votre nom. Cette fonctionnalité utilise les rôles IAM d'Amazon ECS pour la fonctionnalité des tâches. Pour plus d'informations, consultez les [rôles IAM d'exécution de tâches Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.

19. (Facultatif) Développez la configuration supplémentaire :

- a. Pour les variables d'environnement, choisissez Ajouter une variable d'environnement pour ajouter des variables d'environnement sous forme de paires nom-valeur. Ces variables sont transmises au conteneur.
- b. Pour la configuration du rôle Job, vous pouvez spécifier un rôle IAM qui fournit au conteneur de votre job l'autorisation d'utiliser le AWS APIs. Cette fonctionnalité utilise les rôles IAM d'Amazon ECS pour la fonctionnalité des tâches. Pour plus d'informations, notamment sur les prérequis de configuration, consultez la section [Rôles IAM pour les tâches](#) du manuel Amazon Elastic Container Service Developer Guide.

Note

Pour les tâches exécutées sur les ressources Fargate, un rôle de tâche est requis.

Note

Seuls les rôles dotés d'une relation de confiance avec Amazon Elastic Container Service Task Role sont présentés ici. Pour plus d'informations sur la création d'un rôle IAM pour vos AWS Batch tâches, consultez la section [Création d'un rôle IAM et d'une politique pour vos tâches](#) dans le manuel Amazon Elastic Container Service Developer Guide.

- c. Pour le rôle d'exécution, spécifiez un rôle IAM qui autorise les agents de conteneur Amazon ECS à effectuer des appels AWS d'API en votre nom. Cette fonctionnalité utilise les rôles IAM d'Amazon ECS pour la fonctionnalité des tâches. Pour plus d'informations, consultez les [rôles IAM d'exécution de tâches Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.

20. Dans la section Configuration de sécurité :

- a. (Facultatif) Pour accorder au conteneur de votre tâche des privilèges élevés sur l'instance hôte (similaires à ceux de l'utilisateur root), activez Privileged. Ce paramètre correspond à Privileged dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option `--privileged` correspond à [docker run](#).

- b. (Facultatif) Dans Utilisateur, entrez le nom d'utilisateur à utiliser dans le conteneur. Ce paramètre correspond à `User` dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option `--user` correspond à [docker run](#).
- c. (Facultatif) Pour les secrets, choisissez Ajouter un secret pour ajouter des secrets sous forme de paires nom-valeur. Ces secrets sont exposés dans le contenant. [Pour plus d'informations, consultez : LogConfiguration SecretOptions](#).

21. Dans la section de configuration Linux :

- a. Activez l'option Enable read only filesystem (Activer le système de fichiers en lecture seule) pour supprimer l'accès en écriture au volume.
- b. (Facultatif) Activez l'option Activer init le processus pour exécuter un `init` processus dans le conteneur. Ce processus transmet les signaux et récolte les processus.
- c. Pour Taille de la mémoire partagée, entrez la taille (en MiB) du `/dev/shm` volume.
- d. Pour Taille maximale de swap, entrez la quantité totale de mémoire d'échange (en MiB) que le conteneur peut utiliser.
- e. Pour Swappiness, entrez une valeur comprise entre 0 et 100 pour indiquer le comportement de swappiness du conteneur. Si vous ne spécifiez aucune valeur et que l'échange est activé, la valeur par défaut est 60. [Pour plus d'informations, consultez : LinuxParameters swappiness](#).
- f. (Facultatif) Pour Appareils, choisissez Ajouter un appareil pour ajouter un appareil :
 - i. Sous Chemin du conteneur, spécifiez le chemin d'accès de l'instance du conteneur pour exposer l'appareil mappé à l'instance de l'hôte. Si vous laissez ce champ vide, le chemin de l'hôte est utilisé dans le conteneur.
 - ii. Sous Host path (Chemin de l'hôte), spécifiez le chemin d'accès d'un appareil dans l'instance de l'hôte.
 - iii. Pour Autorisations, choisissez une ou plusieurs autorisations à appliquer à l'appareil. Les autorisations disponibles sont READ, WRITE et MKNOD.

22. (Facultatif) Pour les points de montage, choisissez Ajouter une configuration de points de montage pour ajouter des points de montage pour les volumes de données. Vous devez spécifier le volume source et le chemin du conteneur. Ces points de montage sont transmis au Docker démon sur une instance de conteneur. Vous pouvez également choisir de rendre le volume en lecture seule.

23. (Facultatif) Pour la configuration Ulimits, choisissez Add ulimit pour ajouter une ulimits valeur au conteneur. Entrez les valeurs Nom, Limite souple et Limite stricte, puis choisissez Ajouter une limite.
24. (Facultatif) Pour la configuration des volumes, choisissez Ajouter un volume pour créer une liste de volumes à transmettre au conteneur. Entrez le nom et le chemin source du volume, puis choisissez Ajouter un volume. Vous pouvez également choisir d'activer Activer EFS.
25. (Facultatif) Pour les fichiers Tmpfs, choisissez Ajouter des fichiers tmpfs pour ajouter un montage. tmpfs
26. Dans la section Propriétés de la tâche :
 - a. Pour Rôle d'exécution - conditionnel, choisissez un rôle pour autoriser les agents Amazon ECS à effectuer des appels d' AWS API en votre nom. Pour plus d'informations sur la création d'un rôle d'exécution, consultez [Tutoriel : Création du rôle d'exécution IAM](#).

b.



Important

Pour utiliser la commande ECS execute, votre environnement de calcul doit respecter les [considérations relatives à l'environnement de calcul pour les tâches parallèles à nœuds multiples](#).

Choisissez la commande Enable ECS execute pour permettre l'accès direct au shell du conteneur Amazon ECS et contourner le système d'exploitation hôte. Vous devez choisir un rôle de tâche.



Important

La commande ECS execute nécessite que le système de fichiers soit accessible en écriture.

- c. Pour le rôle Task, choisissez un rôle Amazon ECS Identity and Access Management (IAM) pour autoriser le conteneur à effectuer des appels d' AWS API en votre nom. Pour plus d'informations, consultez le [rôle IAM de la tâche Amazon ECS](#) dans le manuel du développeur Amazon Elastic Container Service.
27. (Facultatif) Dans la section Configuration de la journalisation :
 - a. Pour Log driver, choisissez le log driver à utiliser. [Pour plus d'informations sur les pilotes de journal disponibles, voir : LogDriverLogConfiguration](#).

Note

Par défaut, le pilote de `awslogs` journal est utilisé.

- b. Pour Options, choisissez Ajouter une option pour ajouter une option. Entrez une paire nom-valeur, puis choisissez Ajouter une option.
- c. Pour Secrets, choisissez Ajouter un secret. Entrez une paire nom-valeur, puis choisissez Ajouter un secret pour ajouter un secret.

Tip

[Pour plus d'informations, consultez : LogConfiguration SecretOptions.](#)

- 28. Choisissez Page suivante.
- 29. Pour la révision de la définition du Job, passez en revue les étapes de configuration. Si vous devez apporter des modifications, choisissez Modifier. Lorsque vous avez terminé, choisissez Créer une définition de tâche.

Modèle de définition de travail qui utilise ContainerProperties

Ce qui suit est un modèle de définition de tâche vide qui inclut un seul conteneur. Vous pouvez utiliser ce modèle pour créer votre définition de tâche, qui peut ensuite être enregistrée dans un fichier et utilisée avec l' AWS CLI `--cli-input-json` option. Pour obtenir plus d'informations sur ces paramètres, consultez [JobDefinition](#).

Note

Vous pouvez générer un modèle de définition de tâche à conteneur unique à l'aide de la AWS CLI commande suivante :

```
$ aws batch register-job-definition --generate-cli-skeleton
```

```
{  
  "jobDefinitionName": "",  
  "type": "container",
```



```
"parameters": {
  "KeyName": ""
},
"schedulingPriority": 0,
"containerProperties": {
  "image": "",
  "vcpus": 0,
  "memory": 0,
  "command": [
    ""
  ],
  "jobRoleArn": "",
  "executionRoleArn": "",
  "volumes": [
    {
      "host": {
        "sourcePath": ""
      },
      "name": "",
      "efsVolumeConfiguration": {
        "fileSystemId": "",
        "rootDirectory": "",
        "transitEncryption": "ENABLED",
        "transitEncryptionPort": 0,
        "authorizationConfig": {
          "accessPointId": "",
          "iam": "DISABLED"
        }
      }
    }
  ],
  "environment": [
    {
      "name": "",
      "value": ""
    }
  ],
  "mountPoints": [
    {
      "containerPath": "",
      "readOnly": true,
      "sourceVolume": ""
    }
  ],

```

```
"readonlyRootFilesystem": true,
"privileged": true,
"ulimits": [
  {
    "hardLimit": 0,
    "name": "",
    "softLimit": 0
  }
],
"user": "",
"instanceType": "",
"resourceRequirements": [
  {
    "value": "",
    "type": "MEMORY"
  }
],
"linuxParameters": {
  "devices": [
    {
      "hostPath": "",
      "containerPath": "",
      "permissions": [
        "WRITE"
      ]
    }
  ],
  "initProcessEnabled": true,
  "sharedMemorySize": 0,
  "tmpfs": [
    {
      "containerPath": "",
      "size": 0,
      "mountOptions": [
        ""
      ]
    }
  ],
  "maxSwap": 0,
  "swappiness": 0
},
"logConfiguration": {
  "logDriver": "syslog",
  "options": {
```

```

        "KeyName": ""
    },
    "secretOptions": [
        {
            "name": "",
            "valueFrom": ""
        }
    ]
},
"secrets": [
    {
        "name": "",
        "valueFrom": ""
    }
],
"networkConfiguration": {
    "assignPublicIp": "DISABLED"
},
"fargatePlatformConfiguration": {
    "platformVersion": ""
}
},
"nodeProperties": {
    "numNodes": 0,
    "mainNode": 0,
    "nodeRangeProperties": [
        {
            "targetNodes": "",
            "container": {
                "image": "",
                "vcpus": 0,
                "memory": 0,
                "command": [
                    ""
                ],
                "jobRoleArn": "",
                "executionRoleArn": "",
                "volumes": [
                    {
                        "host": {
                            "sourcePath": ""
                        },
                        "name": "",
                        "efsVolumeConfiguration": {

```

```

        "fileSystemId": "",
        "rootDirectory": "",
        "transitEncryption": "DISABLED",
        "transitEncryptionPort": 0,
        "authorizationConfig": {
            "accessPointId": "",
            "iam": "ENABLED"
        }
    }
},
"environment": [
    {
        "name": "",
        "value": ""
    }
],
"mountPoints": [
    {
        "containerPath": "",
        "readOnly": true,
        "sourceVolume": ""
    }
],
"readonlyRootFilesystem": true,
"privileged": true,
"ulimits": [
    {
        "hardLimit": 0,
        "name": "",
        "softLimit": 0
    }
],
"user": "",
"instanceType": "",
"resourceRequirements": [
    {
        "value": "",
        "type": "MEMORY"
    }
],
"linuxParameters": {
    "devices": [
        {

```

```

        "hostPath": "",
        "containerPath": "",
        "permissions": [
            "WRITE"
        ]
    },
],
"initProcessEnabled": true,
"sharedMemorySize": 0,
"tmpfs": [
    {
        "containerPath": "",
        "size": 0,
        "mountOptions": [
            ""
        ]
    }
],
"maxSwap": 0,
"swappiness": 0
},
"logConfiguration": {
    "logDriver": "awslogs",
    "options": {
        "KeyName": ""
    },
    "secretOptions": [
        {
            "name": "",
            "valueFrom": ""
        }
    ]
},
"secrets": [
    {
        "name": "",
        "valueFrom": ""
    }
],
"networkConfiguration": {
    "assignPublicIp": "DISABLED"
},
"fargatePlatformConfiguration": {
    "platformVersion": ""
}

```

```

        }
    }
}
],
},
"retryStrategy": {
    "attempts": 0,
    "evaluateOnExit": [
        {
            "onStatusReason": "",
            "onReason": "",
            "onExitCode": "",
            "action": "RETRY"
        }
    ]
},
"propagateTags": true,
"timeout": {
    "attemptDurationSeconds": 0
},
"tags": {
    "KeyName": ""
},
"platformCapabilities": [
    "EC2"
],
"eksProperties": {
    "podProperties": {
        "serviceAccountName": "",
        "hostNetwork": true,
        "dnsPolicy": "",
        "containers": [
            {
                "name": "",
                "image": "",
                "imagePullPolicy": "",
                "command": [
                    ""
                ],
                "args": [
                    ""
                ],
                "env": [
                    {

```

```

        "name": "",
        "value": ""
    }
],
"resources": {
    "limits": {
        "KeyName": ""
    },
    "requests": {
        "KeyName": ""
    }
},
"volumeMounts": [
    {
        "name": "",
        "mountPath": "",
        "readOnly": true
    }
],
"securityContext": {
    "runAsUser": 0,
    "runAsGroup": 0,
    "privileged": true,
    "readOnlyRootFilesystem": true,
    "runAsNonRoot": true
}
},
"volumes": [
    {
        "name": "",
        "hostPath": {
            "path": ""
        },
        "emptyDir": {
            "medium": "",
            "sizeLimit": ""
        },
        "secret": {
            "secretName": "",
            "optional": true
        }
    }
]

```

```
}  
  }  
}
```

Paramètres de définition du job pour ContainerProperties

Les définitions de Job utilisées [ContainerProperties](#) sont divisées en plusieurs parties :

- Le nom de la définition de tâche
- Type de définition de poste
- L'espace réservé par défaut pour la substitution de paramètres
- Les propriétés du conteneur pour la tâche
- Les propriétés Amazon EKS pour la définition des tâches qui sont nécessaires pour les tâches exécutées sur les ressources Amazon EKS
- Les propriétés du nœud nécessaires pour une tâche parallèle à nœuds multiples
- Les fonctionnalités de la plateforme nécessaires pour exécuter les tâches sur les ressources Fargate
- Les détails de propagation des balises par défaut de la définition de tâche
- La stratégie de nouvelle tentative par défaut pour la définition de la tâche
- Priorité de planification par défaut pour la définition de tâche
- Les balises par défaut pour la définition de la tâche
- Le délai d'expiration par défaut pour la définition de la tâche

Table des matières

- [Nom de la définition de tâche](#)
- [Type](#)
- [Paramètres](#)
- [Propriétés du contenant](#)
- [Propriétés d'Amazon EKS](#)
- [Capacités de la plateforme](#)
- [Propager les balises](#)
- [Propriétés de nœud](#)
- [Stratégie de nouvelle tentative](#)

- [Priorité de planification](#)
- [Balises](#)
- [Timeout](#)

Nom de la définition de tâche

`jobDefinitionName`

Lorsque vous enregistrez une définition de tâche, vous indiquez un nom. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_). La première définition de tâche enregistrée sous ce nom reçoit une révision de 1. Toutes les définitions de tâches enregistrées avec ce nom par la suite se voient attribuer un numéro de révision incrémentiel.

Type : String

Obligatoire : oui

Type

`type`

Lorsque vous enregistrez une définition de tâche, vous indiquez le type de tâche. Si la tâche s'exécute sur les ressources Fargate, elle n'est pas prise en charge `multinode`. Pour plus d'informations sur les tâches parallèles à plusieurs nœuds, consultez [Création d'une définition de tâche parallèle à nœuds multiples](#).

Type : String

Valeurs valides : `container` | `multinode`

Obligatoire : oui

Paramètres

`parameters`

Lorsque vous soumettez une tâche, vous pouvez spécifier des paramètres qui remplacent les espaces réservés ou remplacent les paramètres de définition de tâche par défaut. Les paramètres

de demandes de soumission de tâche prévalent sur les valeurs par défaut dans une définition de tâche. Cela signifie que vous pouvez utiliser la même définition de tâche pour plusieurs tâches utilisant le même format. Vous pouvez également modifier les valeurs de la commande par programmation au moment de l'envoi.

Type : mappage chaîne/chaîne

Obligatoire : non

Lorsque vous enregistrez une définition de tâche, vous pouvez utiliser les espaces réservés de substitution de paramètres dans le champ `command` des propriétés de conteneur de la tâche. La syntaxe est la suivante.

```
"command": [  
  "ffmpeg",  
  "-i",  
  "Ref::inputfile",  
  "-c",  
  "Ref::codec",  
  "-o",  
  "Ref::outputfile"  
]
```

Dans l'exemple ci-dessus, aucun espace réservé de substitution de paramètre `Ref::inputfile`, `Ref::codec` et `Ref::outputfile` n'est présent dans la commande. Vous pouvez utiliser l'`parameters` objet dans la définition de tâche pour définir des valeurs par défaut pour ces espaces réservés. Par exemple, pour définir une valeur par défaut pour l'espace réservé `Ref::codec`, vous précisez les éléments suivants dans la définition de tâche :

```
"parameters" : {"codec" : "mp4"}
```

Lorsque cette définition de tâche est soumise pour être exécutée, l'`Ref::codec` argument de la commande pour le conteneur est remplacé par la valeur par défaut, `mp4`.

Propriétés du contenant

Lorsque vous enregistrez une définition de tâche, spécifiez une liste de propriétés de conteneur qui sont transmises au démon Docker sur une instance de conteneur lorsque la tâche est placée. Les propriétés de conteneurs suivantes sont autorisées dans une définition de tâche. Pour les tâches à

nœud unique, ces propriétés de conteneur sont définies au niveau de la définition de tâche. Pour les tâches parallèles à plusieurs nœuds, les propriétés de conteneur sont définies au niveau [Propriétés de nœud](#), pour chaque groupe de nœuds.

command

La commande transmise au conteneur. Ce paramètre correspond à Cmd dans la section relative à la [création d'un conteneur](#) de [Docker Remote API](#) et au paramètre COMMAND de [docker run](#). Pour plus d'informations sur le CMD paramètre Docker, consultez <https://docs.docker.com/engine/reference/builder/#cmd>.

```
"command": ["string", ...]
```

Type : tableau de chaînes

Obligatoire : non

environment

Variables d'environnement à transmettre à un conteneur. Ce paramètre correspond à Env dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option `--env` correspond à [docker run](#).

Important

Nous vous déconseillons d'utiliser des variables d'environnement en texte clair pour les informations sensibles, comme les données d'informations d'identification.

Note

Les variables d'environnement ne doivent pas commencer par `AWS_BATCH`. Cette convention de dénomination est réservée aux variables définies par le AWS Batch service.

Type : tableau de paires clé-valeur

Obligatoire : non

name

Nom de la variable d'environnement.

Type : String

Obligatoire : oui lorsque environment est utilisé.

value

Valeur de la variable d'environnement.

Type : String

Obligatoire : oui lorsque environment est utilisé.

```
"environment" : [  
  { "name" : "envName1", "value" : "envValue1" },  
  { "name" : "envName2", "value" : "envValue2" }  
]
```

executionRoleArn

Lorsque vous enregistrez une définition de tâche, vous pouvez spécifier un rôle IAM. Le rôle fournit à l'agent de conteneur Amazon ECS l'autorisation d'appeler en votre nom les actions d'API spécifiées dans ses politiques associées. Les tâches exécutées sur les ressources Fargate doivent fournir un rôle d'exécution. Pour de plus amples informations, veuillez consulter [AWS Batch Rôle d'exécution IAM](#).

Type : chaîne

Obligatoire : non

fargatePlatformConfiguration

Configuration de la plate-forme pour les tâches exécutées sur les ressources Fargate. Les tâches exécutées sur EC2 des ressources ne doivent pas spécifier ce paramètre.

Type : objet [FargatePlatformConfiguration](#)

Obligatoire : non

platformVersion

La version AWS de la plateforme Fargate utilisée pour les tâches, LATEST ou pour utiliser une version récente et approuvée de la AWS plateforme Fargate.

Type : String

Par défaut: LATEST

Obligatoire : non

image

Image utilisée pour démarrer une tâche. Cette chaîne est transmise directement au démon Docker. Les images dans le registre Docker Hub sont disponibles par défaut. Vous pouvez également spécifier d'autres référentiels avec *repository-url/image:tag*. Il peut comporter jusqu'à 255 lettres (majuscules et minuscules), des chiffres, des tirets, des traits de soulignement, deux points, des points, des barres obliques et des signes dièse. Ce paramètre se mappe à Image dans la section [Create a container](#) (Création d'un conteneur) de [Docker Remote API](#) (L'API Docker à distance) et le paramètre IMAGE de [docker run](#).

Note

L'architecture des images Docker doit correspondre à l'architecture du processeur des ressources de calcul sur lesquelles elles sont planifiées. Par exemple, des images Docker basées sur ARM ne peuvent s'exécuter que sur des ressources de calcul basées sur ARM.

- Les images contenues dans les référentiels publics Amazon ECR utilisent les conventions complètes `registry/repository[:tag]` ou de `registry/repository[@digest]` dénomination (par exemple, `public.ecr.aws/registry_alias/my-web-app:latest`).
- Les images contenues dans les référentiels Amazon ECR utilisent la convention de `registry/repository:[tag]` dénomination complète. Par exemple, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`.
- Les images dans les référentiels officiels sur Docker Hub utilisent un nom unique (par exemple, `ubuntu` ou `mongo`).
- Les images dans les autres référentiels sur Docker Hub sont qualifiées par un nom d'organisation (par exemple, `amazon/amazon-ecs-agent`).
- Les images dans les autres référentiels en ligne sont qualifiées par un nom de domaine (par exemple, `quay.io/assemblyline/ubuntu`).

Type : String

Obligatoire : oui

instanceType

Type d'instance à utiliser pour une tâche parallèle à plusieurs nœuds. Tous les groupes de nœuds dans une tâche parallèle à plusieurs nœuds doivent utiliser le même type d'instance. Ce paramètre n'est pas valide pour les tâches de conteneur à nœud unique ou pour les tâches exécutées sur des ressources Fargate.

Type : chaîne

Obligatoire : non

jobRoleArn

Lorsque vous enregistrez une définition de tâche, vous pouvez spécifier un rôle IAM. Le rôle autorise le conteneur de tâches à appeler les actions d'API qui sont spécifiées dans ses stratégies associées à votre place. Pour plus d'informations, consultez [Rôles IAM pour les tâches](#) dans le Guide du développeur Amazon Elastic Container Service.

Type : chaîne

Obligatoire : non

linuxParameters

Modifications spécifiques à Linux appliquées au conteneur, telles que les détails des mappages des appareils.

```
"linuxParameters": {
  "devices": [
    {
      "hostPath": "string",
      "containerPath": "string",
      "permissions": [
        "READ", "WRITE", "MKNOD"
      ]
    }
  ],
  "initProcessEnabled": true/false,
  "sharedMemorySize": 0,
  "tmpfs": [
    {
      "containerPath": "string",
      "size": integer,
```

```
        "mountOptions": [
            "string"
        ]
    },
    "maxSwap": integer,
    "swappiness": integer
}
```

Type : objet [LinuxParameters](#)

Obligatoire : non

devices

Liste des appareils mappés au conteneur. Ce paramètre correspond à Devices dans la section [Create a container](#) (Créer un conteneur) de [Docker Remote API](#) et l'option `--device` à la commande [docker run](#).

 Note

Ce paramètre ne s'applique pas aux tâches qui s'exécutent sur des ressources Fargate.

Type : tableau d'objets [Périphérique](#)

Obligatoire : non

hostPath

Chemin où se trouve le périphérique disponible dans l'instance de conteneur hôte.

Type : String

Obligatoire : oui

containerPath

Le chemin où l'appareil est exposé dans le conteneur se trouve. Si cela n'est pas spécifié, le périphérique est exposé sur le même chemin que le chemin de l'hôte.

Type : chaîne

Obligatoire : non

permissions

Autorisations pour l'appareil dans le conteneur. Si cela n'est pas spécifié, les autorisations sont définies sur `READWRITE`, et `MKNOD`.

Type : tableau de chaînes

Obligatoire : non

Valeurs valides : `READ` | `WRITE` | `MKNOD`

initProcessEnabled

Si ce paramètre est défini sur « `true` », exécutez un processus `init` dans le conteneur afin de transmettre des signaux et de récolter les processus. Ce paramètre est mappé à l'option `--init` de [docker run](#). Ce paramètre nécessite la version 1.25 de l'API Docker à distance ou une version supérieure sur votre instance de conteneur. Pour vérifier la version de l'API Docker à distance de votre instance de conteneur, connectez-vous à votre instance de conteneur et exécutez la commande suivante : `sudo docker version | grep "Server API version"`

Type : booléen

Obligatoire : non

maxSwap

Quantité totale de mémoire d'échange (en MiB) qu'une tâche peut utiliser. Ce paramètre est converti en l'option `--memory-swap` de la commande [docker run](#), où la valeur est la somme de la mémoire du conteneur plus la valeur `maxSwap`. Pour plus d'informations, veuillez consulter les [détails de --memory-swap](#) de la documentation Docker.

Si la valeur `0` est spécifiée pour `maxSwap`, le conteneur n'utilise pas l'échange. Les valeurs acceptées sont `0` ou n'importe quel nombre entier positif. Si le `maxSwap` paramètre est omis, le conteneur utilise la configuration d'échange pour l'instance de conteneur sur laquelle il s'exécute. Une valeur `maxSwap` doit être définie pour que le paramètre `swappiness` soit utilisé.

Note

Ce paramètre ne s'applique pas aux tâches qui s'exécutent sur des ressources Fargate.

Type : entier

Obligatoire : non

`sharedMemorySize`

Valeur correspondant à la taille (en Mio) du volume `/dev/shm`. Ce paramètre est mappé à l'option `--shm-size` de [docker run](#).

 Note

Ce paramètre ne s'applique pas aux tâches qui s'exécutent sur des ressources Fargate.

Type : entier

Obligatoire : non

`swappiness`

Vous pouvez utiliser ce paramètre pour régler le comportement d'échange de mémoire d'un conteneur. `swappiness` La valeur de 0 empêche l'échange, sauf en cas d'absolue nécessité. Avec la valeur `swappiness` pour 100, l'échange de pages a lieu de manière agressive. Les valeurs acceptées sont les nombres entiers compris entre 0 et 100. Si le paramètre `swappiness` n'est pas spécifié, la valeur par défaut 60 est utilisée. Si aucune valeur n'est spécifiée pour `maxSwap`, le paramètre est ignoré. Si `maxSwap` est défini sur 0, le conteneur n'a pas recours à l'échange. Ce paramètre est mappé à l'option `--memory-swappiness` de [docker run](#).

Tenez compte des points suivants lorsque vous utilisez une configuration d'échange par conteneur.

- L'espace d'échange doit être activé et alloué sur l'instance de conteneur pour que les conteneurs puissent l'utiliser.

 Note

Le swap n'est pas activé par défaut pour les appareils optimisés pour Amazon ECS. Vous devez activer l'échange sur l'instance pour utiliser cette fonction. Pour plus d'informations, consultez [Instance Store Swap Volumes](#) dans le guide de EC2

l'utilisateur Amazon ou [Comment allouer de la mémoire pour qu'elle fonctionne comme espace de swap dans une EC2 instance Amazon à l'aide d'un fichier d'échange ?](#) .

- Les paramètres de l'espace d'échange ne sont pris en charge que pour les définitions de tâches utilisant EC2 des ressources.
- Si les paramètres `maxSwap` et `swappiness` sont omis d'une définition de tâche, chaque conteneur a une valeur `swappiness` par défaut de 60. L'utilisation totale du swap est limitée à deux fois la réserve de mémoire du conteneur.

 Note

Ce paramètre ne s'applique pas aux tâches qui s'exécutent sur des ressources Fargate.

Type : entier

Obligatoire : non

`tmpfs`

Chemin du conteneur, options de montage et taille du montage `tmpfs`.

Type : tableau d'objets [Tmpfs](#)

 Note

Ce paramètre ne s'applique pas aux tâches qui s'exécutent sur des ressources Fargate.

Obligatoire : non

`containerPath`

Chemin d'accès absolu du fichier dans le conteneur où le volume `tmpfs` est monté.

Type : String

Obligatoire : oui

mountOptions

Liste des options de montage du volume tmpfs.

Valeurs valides : "defaults" | ro "" | rw "" | suid "" | nosuid "" | dev "nodev" | "exec" | "noexec" | "sync" | "async" | "dirsync" | remount "" | mand "|" nomand "latime" » | noatime "" | diratime "» | nodiratime "|" bind" | "rbind" | unbindable |runbindable" | "private" | rprivate "|" | shared |rshared" |slave" » | rslave "relatime" | norelatime "» | « strictatime » | « nostrictatime » | » mode« "|" uid "|" gid "|" nr_inodes "|" nr_blocks "|" mpol »

Type : tableau de chaînes

Obligatoire : non

size

Taille (en Mio) du volume tmpfs.

Type : entier

Obligatoire : oui

logConfiguration

Spécification de configuration du journal pour la tâche.

Ce paramètre correspond à LogConfig dans la section [Create a container](#) (Créer un conteneur) de [Docker Remote API](#) et l'option `--log-driver` à la commande [docker run](#). Par défaut, les conteneurs utilisent le même pilote de journalisation que celui utilisé par le démon Docker. Toutefois, le conteneur peut utiliser un pilote de journalisation différent de celui du daemon Docker en spécifiant un pilote de journal avec ce paramètre dans la définition du conteneur. Pour utiliser un pilote de journalisation différent pour un conteneur, le système de journalisation doit être configuré sur l'instance du conteneur ou sur un autre serveur de journaux afin de fournir des options de journalisation à distance. Pour plus d'informations sur les options relatives aux différents pilotes du journal pris en charge, consultez la section [Configure logging drivers](#) (Configurer les pilotes de journalisation) dans la documentation Docker.

Note

AWS Batch prend actuellement en charge un sous-ensemble des pilotes de journalisation disponibles pour le démon Docker (indiqué dans le type de [LogConfiguration](#) données).

Ce paramètre nécessite la version 1.18 de l'API Docker à distance ou une version supérieure sur votre instance de conteneur. Pour vérifier la version de l'API Docker à distance de votre instance de conteneur, connectez-vous à votre instance de conteneur et exécutez la commande suivante :

```
sudo docker version | grep "Server API version"
```

```
"logConfiguration": {
  "devices": [
    {
      "logDriver": "string",
      "options": {
        "optionName1" : "optionValue1",
        "optionName2" : "optionValue2"
      }
      "secretOptions": [
        {
          "name" : "secretOptionName1",
          "valueFrom" : "secretOptionArn1"
        },
        {
          "name" : "secretOptionName2",
          "valueFrom" : "secretOptionArn2"
        }
      ]
    }
  ]
}
```

Type : objet [LogConfiguration](#)

Obligatoire : non

logDriver

Le pilote de journal à utiliser pour la tâche. AWS Batch Active le pilote de awslogs journal par défaut. Les valeurs valides répertoriées pour ce paramètre sont les pilotes de journal avec lesquels l'agent de conteneur Amazon ECS peut communiquer par défaut.

Ce paramètre correspond à LogConfig dans la section [Create a container](#) (Créer un conteneur) de [Docker Remote API](#) et l'option `--log-driver` à la commande [docker run](#). Par défaut, les tâches utilisent le même pilote de journalisation que celui utilisé par le démon Docker. Cependant, la tâche peut utiliser un pilote de journalisation différent de celui du démon Docker en spécifiant un pilote de journal avec ce paramètre dans la définition de

la tâche. Si vous souhaitez spécifier un autre pilote de journalisation pour une tâche, le système de journalisation doit être configuré sur l'instance de conteneur dans l'environnement informatique. Vous pouvez également le configurer sur un autre serveur de journaux pour fournir des options de journalisation à distance. Pour plus d'informations sur les options relatives aux différents pilotes du journal pris en charge, consultez la section [Configure logging drivers](#) (Configurer les pilotes de journalisation) dans la documentation Docker.

 Note

AWS Batch prend actuellement en charge un sous-ensemble des pilotes de journalisation disponibles pour le démon Docker. Les pilotes de journal supplémentaires qui ne sont pas pris en charge actuellement seront peut-être disponibles dans les prochaines versions de l'agent de conteneur Amazon ECS.

Les pilotes de journal pris en charge sont `awslogs`, `fluentd`, `gelf`, `json-file`, `journald`, `logentries`, `syslog` et `splunk`.

 Note

Les tâches exécutées sur les ressources Fargate sont limitées aux `awslogs` pilotes et `log. splunk`

Ce paramètre nécessite la version 1.18 de l'API Docker à distance ou une version supérieure sur votre instance de conteneur. Pour vérifier la version de l'API Docker à distance de votre instance de conteneur, connectez-vous à votre instance de conteneur et exécutez la commande suivante : `sudo docker version | grep "Server API version"`

 Note

L'agent de conteneur Amazon ECS qui s'exécute sur une instance de conteneur doit enregistrer les pilotes de journalisation disponibles sur cette instance avec la variable d'environnement `ECS_AVAILABLE_LOGGING_DRIVERS`. Dans le cas contraire, les conteneurs placés sur cette instance ne peuvent pas utiliser ces options de configuration du journal. Pour plus d'informations, consultez [Configuration de l'agent](#)

[du conteneur Amazon ECS](#) dans le Manuel du développeur Amazon Elastic Container Service.

awslogs

Spécifie le pilote de CloudWatch journalisation Amazon Logs. Pour plus d'informations, consultez le [pilote Utiliser le pilote de journal awslogs de journalisation Amazon CloudWatch Logs](#) dans la documentation Docker.

fluentd

Spécifie le pilote de journalisation Fluentd. Pour plus d'informations, notamment sur l'utilisation et les options, consultez le [pilote de journalisation Fluentd](#) dans la documentation Docker.

gelf

Spécifie le pilote de journalisation GELF (Graylog Extended Format). Pour plus d'informations, notamment sur l'utilisation et les options, consultez le [pilote de journalisation Graylog Extended Format](#) dans la documentation Docker.

journald

Spécifie le pilote de journalisation journald. Pour plus d'informations, notamment sur son utilisation et ses options, consultez le [pilote de journalisation Journald](#) dans la documentation Docker.

json-file

Spécifie le pilote de journalisation du fichier JSON. Pour plus d'informations, notamment sur l'utilisation et les options, consultez le [pilote de journalisation de fichiers JSON](#) dans la documentation Docker.

splunk

Spécifie le pilote de journalisation Splunk. Pour plus d'informations, notamment sur l'utilisation et les options, consultez le [pilote de journalisation Splunk](#) dans la documentation Docker.

syslog

Spécifie le pilote de journalisation syslog. Pour plus d'informations, notamment sur l'utilisation et les options, consultez le [pilote de journalisation Syslog](#) dans la documentation Docker.

Type : String

Obligatoire : oui

Valeurs valides : awslogs | fluentd | gelf | journald | json-file | splunk | syslog

 Note

Si vous avez un pilote personnalisé qui n'est pas répertorié précédemment et que vous souhaitez utiliser avec l'agent de conteneur Amazon ECS, vous pouvez bifurquer le projet d'agent de conteneur Amazon ECS [disponible sur GitHub et le](#) personnaliser pour qu'il fonctionne avec ce pilote. Nous vous conseillons d'envoyer des demandes d'extraction pour les modifications que vous souhaitez inclure. Cependant, Amazon Web Services ne prend actuellement pas en charge les demandes qui exécutent des copies modifiées de ce logiciel.

options

Options de configuration du journal à envoyer à un pilote de journal pour la tâche.

Ce paramètre nécessite la version 1.19 de l'API Docker à distance ou une version supérieure sur votre instance de conteneur.

Type : mappage chaîne/chaîne

Obligatoire : non

secretOptions

Objet qui représente le secret à transmettre à la configuration de journal. Pour de plus amples informations, veuillez consulter [Spécifier les données sensibles](#).

Type : tableau d'objets

Obligatoire : non

name

Nom de l'option de pilote de journal à définir dans le job.

Type : String

Obligatoire : oui

valueFrom

Le nom de ressource Amazon (ARN) du secret à exposer à la configuration du journal du conteneur. Les valeurs prises en charge sont soit l'ARN complet du secret Secrets Manager, soit l'ARN complet du paramètre dans le magasin de paramètres SSM.

 Note

Si le paramètre SSM Parameter Store existe au même Région AWS endroit que la tâche que vous lancez, vous pouvez utiliser l'ARN complet ou le nom du paramètre. Si le paramètre existe dans une autre région, l'ARN complet doit être spécifié.

Type : String

Obligatoire : oui

memory

Ce paramètre est obsolète, [resourceRequirements](#) utilisez-le à la place.

Le nombre de MiB de mémoire réservé à la tâche.

À titre d'exemple d'utilisation [resourceRequirements](#), si votre définition de tâche contient une syntaxe similaire à la suivante.

```
"containerProperties": {  
  "memory": 512  
}
```

La syntaxe équivalente utilisée [resourceRequirements](#) est la suivante.

```
"containerProperties": {  
  "resourceRequirements": [  
    {  
      "type": "MEMORY",  
      "value": "512"  
    }  
  ]  
}
```



```
}
```

Type : entier

Obligatoire : oui

mountPoints

Points de montage des volumes de données dans votre conteneur. Ce paramètre correspond à Volumes dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option `--volume` correspond à [docker run](#).

```
"mountPoints": [  
  {  
    "sourceVolume": "string",  
    "containerPath": "string",  
    "readOnly": true/false  
  }  
]
```

Type : tableau d'objets

Obligatoire : non

sourceVolume

Nom du volume à monter.

Type : String

Obligatoire : oui lorsque mountPoints est utilisé.

containerPath

Le chemin sur le conteneur où monter le volume hôte.

Type : String

Obligatoire : oui lorsque mountPoints est utilisé.

readOnly

Si cette valeur est `true`, le conteneur ne peut accéder au volume qu'en lecture. Si cette valeur est `false`, le conteneur peut écrire sur le volume.

Type : booléen

Obligatoire : non

Par défaut : false

networkConfiguration

Configuration réseau pour les tâches exécutées sur les ressources Fargate. Les tâches exécutées sur EC2 des ressources ne doivent pas spécifier ce paramètre.

```
"networkConfiguration": {  
  "assignPublicIp": "string"  
}
```

Type : tableau d'objets

Obligatoire : non

assignPublicIp

Indique si la tâche présente une adresse IP publique. Cela est nécessaire si la tâche nécessite un accès réseau sortant.

Type : String

Valeurs valides : ENABLED | DISABLED

Obligatoire : non

Par défaut : DISABLED

privileged

Lorsque ce paramètre est défini sur true (vrai), des autorisations élevées inhérentes à l'instance de conteneur hôte sont accordées au conteneur (semblable à l'utilisateur root). Ce paramètre correspond à Privileged dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option --privileged correspond à [docker run](#). Ce paramètre ne s'applique pas aux tâches qui s'exécutent sur des ressources Fargate. Ne le fournissez pas et ne le spécifiez pas comme faux.

```
"privileged": true/false
```

Type : booléen

Obligatoire : non

readonlyRootFilesystem

Quand ce paramètre a la valeur `true`, le conteneur ne dispose que d'un accès en lecture seule au système de fichiers racine. Ce paramètre correspond à `ReadonlyRootfs` dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option `--read-only` correspond à [docker run](#).

```
"readonlyRootFilesystem": true/false
```

Type : booléen

Obligatoire : non

resourceRequirements

Type et quantité d'une ressource à attribuer à un conteneur. Les ressources prises en charge comprennent GPU, MEMORY et VCPU.

```
"resourceRequirements" : [  
  {  
    "type": "GPU",  
    "value": "number"  
  }  
]
```

Type : tableau d'objets

Obligatoire : non

type

Type d'une ressource à attribuer à un conteneur. Les ressources prises en charge comprennent GPU, MEMORY et VCPU.

Type : String

Obligatoire : oui lorsque `resourceRequirements` est utilisé.

value

La quantité de la ressource spécifiée à réserver pour le conteneur. Les valeurs varient en fonction du type spécifié.

type="GPU"

Le nombre de billets physiques GPUs à réserver pour le conteneur. Le nombre de conteneurs GPUs réservés à tous les conteneurs d'une tâche ne peut pas dépasser le nombre de conteneurs disponibles GPUs sur la ressource de calcul sur laquelle la tâche est lancée.

type="MEMORY"

La limite stricte (en Mio) de la mémoire à présenter le conteneur. Si votre conteneur tente de dépasser la mémoire spécifiée ici, il sera désactivé. Ce paramètre correspond à Memory dans la section [Create a container](#) (Créer un conteneur) de [Docker Remote API](#) et l'option `--memory` à la commande [docker run](#). Vous devez spécifier au moins 4 Mio de mémoire pour une tâche. Cela est obligatoire, mais vous pouvez le spécifier en plusieurs endroits pour les tâches parallèles à plusieurs nœuds (MNP). Elle doit être spécifiée pour chaque nœud au moins une fois. Ce paramètre correspond à Memory dans la section [Create a container](#) (Créer un conteneur) de [Docker Remote API](#) tandis que l'option `--memory` correspond à la commande [docker run](#).

 Note

Si vous essayez de maximiser l'utilisation de vos ressources en fournissant à vos tâches autant de mémoire que possible pour un type d'instance particulier, consultez [Gestion de la mémoire des ressources informatiques](#).

Pour les tâches exécutées sur des ressources Fargate, `value` elles doivent correspondre à l'une des valeurs prises en charge. De plus, les VCPU valeurs doivent être l'une des valeurs prises en charge pour cette valeur de mémoire.

VCPU	MEMORY
0,25 vCPU	512, 1024 et 2048 MiB
0,5 vCPU	1024 à 4096 MiB par incréments de 1024 MiB
1 vCPU	2048-8192 MiB par incréments de 1024 MiB
2 vCPU	4096-16384 MiB par incréments de 1024 MiB

VCPU	MEMORY
4 vCPU	8192-30720 MiB par incréments de 1024 MiB
8 vCPU	16384-61440 MiB par incréments de 4096 MiB
16 vCPU	32768-122880 MiB par incréments de 8192 MiB

type="VCPU"

Le nombre de v CPUs réservés à la tâche. Ce paramètre correspond à CpuShares dans la section [Create a container](#) (Créer un conteneur) de [Docker Remote API](#) tandis que l'option `--cpu-shares` correspond à la commande [docker run](#). Chaque processeur virtuel est équivalente à 1 024 parts de processeur. Pour les tâches exécutées sur EC2 des ressources, vous devez spécifier au moins un vCPU. Cela est obligatoire mais peut être spécifié à plusieurs endroits. Elle doit être spécifiée pour chaque nœud au moins une fois.

Pour les tâches exécutées sur des ressources `FargateValue`, elles doivent correspondre à l'une des valeurs prises en charge et les valeurs doivent être MEMORY l'une des valeurs prises en charge pour cette valeur de processeur virtuel. Les valeurs prises en charge sont 0,25, 0,5, 1, 2, 4, 8 et 16.

La valeur par défaut du quota de ressources du vCPU Fargate On-Demand est de 6 v. CPUs Pour plus d'informations sur les quotas de Fargate, consultez la section Quotas de [AWS Fargate](#) dans le. Référence générale d'Amazon Web Services

Type : String

Obligatoire : oui lorsque `resourceRequirements` est utilisé.

secrets

Les secrets du travail exposés sous forme de variables d'environnement. Pour de plus amples informations, veuillez consulter [Spécifier les données sensibles](#).

```
"secrets": [
  {
    "name": "secretName1",
    "valueFrom": "secretArn1"
  },
  {
```

```
    "name": "secretName2",  
    "valueFrom": "secretArn2"  
  }  
  ...  
]
```

Type : tableau d'objets

Obligatoire : non

name

Nom de la variable d'environnement contenant le secret.

Type : String

Obligatoire : oui lorsque `secrets` est utilisé.

valueFrom

Secret à exposer au conteneur. Les valeurs prises en charge sont soit le nom Amazon Resource (ARN) complet du secret Secrets Manager, soit l'ARN complet du paramètre dans le magasin de paramètres SSM.

 Note

Si le paramètre SSM Parameter Store existe au même Région AWS endroit que la tâche que vous lancez, vous pouvez utiliser l'ARN complet ou le nom du paramètre. Si le paramètre existe dans une autre région, l'ARN complet doit être spécifié.

Type : String

Obligatoire : oui lorsque `secrets` est utilisé.

ulimits

Une liste de valeurs `ulimits` à définir dans le conteneur. Ce paramètre correspond à `Ulimits` dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option `--ulimit` correspond à [docker run](#).

```
"ulimits": [  
  {
```

```
    "name": string,  
    "softLimit": integer,  
    "hardLimit": integer  
  }  
  ...  
]
```

Type : tableau d'objets

Obligatoire : non

name

Le type d'ulimit.

Type : String

Obligatoire : oui lorsque ulimits est utilisé.

hardLimit

La limite stricte du type ulimit.

Type : entier

Obligatoire : oui lorsque ulimits est utilisé.

softLimit

La limite flexible du type ulimit.

Type : entier

Obligatoire : oui lorsque ulimits est utilisé.

user

Le nom utilisateur à utiliser à l'intérieur du conteneur. Ce paramètre correspond à User dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option `--user` correspond à [docker run](#).

```
"user": "string"
```

Type : chaîne

Obligatoire : non

vcpus

Ce paramètre est obsolète, [resourceRequirements](#) utilisez-le à la place.

Le nombre de v CPUs réservé au conteneur.

À titre d'exemple d'utilisation `resourceRequirements`, si votre définition de tâche contient des lignes similaires à celles-ci :

```
"containerProperties": {  
  "vcpus": 2  
}
```

Les lignes équivalentes utilisées [resourceRequirements](#) sont les suivantes.

```
"containerProperties": {  
  "resourceRequirements": [  
    {  
      "type": "VCPU",  
      "value": "2"  
    }  
  ]  
}
```

Type : entier

Obligatoire : oui

volumes

Lorsque vous enregistrez une définition de tâche, vous pouvez éventuellement préciser une liste de volumes qui sont transmis au daemon Docker sur une instance de conteneur. Les paramètres suivants sont autorisés dans les propriétés du conteneur :

```
"volumes": [  
  {  
    "name": "string",  
    "host": {  
      "sourcePath": "string"  
    },  
    "efsVolumeConfiguration": {
```



```
[
  {
    "authorizationConfig": {
      "accessPointId": "string",
      "iam": "string"
    },
    "fileSystemId": "string",
    "rootDirectory": "string",
    "transitEncryption": "string",
    "transitEncryptionPort": number
  }
]
```

name

Nom du volume. Il peut comporter jusqu'à 255 lettres (majuscules et minuscules), des chiffres, des tirets ou des traits de soulignement. Ce nom est référencé dans le `sourceVolume` paramètre de définition de conteneur `mountPoints`.

Type : chaîne

Obligatoire : non

host

Le contenu du paramètre `host` détermine si le volume de données persiste sur l'instance de conteneur hôte et où il est stocké. Si le paramètre `host` est vide, le daemon Docker attribue un chemin hôte pour votre volume de données. Cependant, il n'est pas garanti que les données persistent une fois que le conteneur qui leur est associé cesse de fonctionner.

Note

Ce paramètre ne s'applique pas aux tâches qui s'exécutent sur des ressources Fargate.

Type : objet

Obligatoire : non

sourcePath

Le chemin d'accès sur l'instance de conteneur hôte qui est présenté au conteneur. Si ce paramètre est vide, le démon Docker attribue un chemin hôte pour vous.

Si le paramètre `host` contient un emplacement de fichier `sourcePath`, le volume de données persiste à l'emplacement spécifié sur l'instance de conteneur hôte jusqu'à ce que vous le supprimiez manuellement. Si la valeur `sourcePath` n'existe pas sur l'instance de conteneur hôte, le démon Docker la crée. Si l'emplacement n'existe pas, le contenu du chemin source est exporté.

Type : chaîne

Obligatoire : non

`efsVolumeConfiguration`

Ce paramètre est spécifié lorsque vous utilisez un système de fichiers Amazon Elastic File System pour le stockage des tâches. Pour de plus amples informations, veuillez consulter [Volumes Amazon EFS](#).

Type : objet

Obligatoire : non

`authorizationConfig`

Détails de configuration des autorisations pour le système de fichiers Amazon EFS.

Type : chaîne

Obligatoire : non

`accessPointId`

ID de point d'accès Amazon EFS à utiliser. Si un point d'accès est spécifié, la valeur du répertoire racine spécifiée dans le `EFSVolumeConfiguration` doit être omise ou définie sur `/`. Cela applique le chemin défini sur le point d'accès EFS. Si un point d'accès est utilisé, le chiffrement de transit doit être activé dans `EFSVolumeConfiguration`. Pour plus d'informations, consultez [Utilisation des points d'accès Amazon EFS](#) dans le Guide de l'utilisateur Amazon Elastic File System.

Type : chaîne

Obligatoire : non

`iam`

Détermine s'il faut utiliser le rôle IAM de AWS Batch tâche défini dans une définition de tâche lors du montage du système de fichiers Amazon EFS. Si cette

option est activée, le chiffrement en transit doit être activé dans la configuration `EFSVolumeConfiguration`. Si ce paramètre est omis, la valeur par défaut `DISABLED` est utilisée. Pour de plus amples informations, veuillez consulter [Utiliser les points d'accès Amazon EFS](#).

Type : String

Valeurs valides : `ENABLED` | `DISABLED`

Obligatoire : non

`fileSystemId`

ID du système de fichiers Amazon EFS à utiliser.

Type : chaîne

Obligatoire : non

`rootDirectory`

Répertoire du système de fichiers Amazon EFS à monter en tant que répertoire racine à l'intérieur de l'hôte. Si ce paramètre est omis, la racine du volume Amazon EFS est utilisée. Si vous le spécifiez/, cela a le même effet que l'omission de ce paramètre. La longueur maximale est de 4 096 caractères.

Important

Si un point d'accès EFS est spécifié dans `leauthorizationConfig`, le paramètre du répertoire racine doit être omis ou défini sur `/`. Cela applique le chemin défini sur le point d'accès Amazon EFS.

Type : chaîne

Obligatoire : non

`transitEncryption`

Détermine si vous souhaitez activer ou non le chiffrement des données Amazon EFS en transit entre l'hôte Amazon ECS et le serveur Amazon EFS. Le chiffrement en transit doit être activé si l'autorisation Amazon EFS IAM est utilisée. Si ce paramètre est omis, la valeur par défaut `DISABLED` est utilisée. Pour plus d'informations, consultez [Chiffrement des données en transit](#) dans le Guide de l'utilisateur Amazon Elastic File System.

Type : String

Valeurs valides : ENABLED | DISABLED

Obligatoire : non

`transitEncryptionPort`

Port à utiliser lors de l'envoi de données chiffrées entre l'hôte Amazon ECS et le serveur Amazon EFS. Si vous ne spécifiez pas de port de chiffrement en transit, il utilise la stratégie de sélection de port adoptée par l'assistant de montage Amazon EFS. La valeur doit être comprise entre 0 et 65 535. Pour plus d'informations, consultez [Assistant de montage EFS](#) dans le Guide de l'utilisateur Amazon Elastic File System User.

Type : entier

Obligatoire : non

Propriétés d'Amazon EKS

Objet ayant plusieurs propriétés spécifiques à des tâches basées sur Amazon EKS. Cela ne doit pas être spécifié pour les définitions de tâches basées sur Amazon ECS.

`podProperties`

Les propriétés des ressources du Kubernetes module d'une tâche.

Type : objet [EksPodProperties](#)

Obligatoire : non

`containers`

Propriétés du conteneur utilisé sur le pod Amazon EKS.

Type : objet [EksContainer](#)

Obligatoire : non

`args`

Tableau d'arguments pour le point d'entrée. Si cela n'est pas spécifié, le CMD de l'image de conteneur est utilisée. Cela correspond au `args` membre figurant dans la partie [Entrypoint](#) du [Pod](#) in. Kubernetes Les références aux variables d'environnement sont étendues à l'aide de l'environnement du conteneur.

Si la variable d'environnement référencée n'existe pas, la référence de la commande ne change pas. Par exemple, si la référence porte sur « `$(NAME1)` » et que la variable d'environnement `NAME1` n'existe pas, la chaîne de commande reste « `$(NAME1)` ». `$$` est remplacé par `$`, et la chaîne résultante n'est pas développée. Par exemple, `$(VAR_NAME)` est transmis en tant que `$(VAR_NAME)`, que la variable d'environnement `VAR_NAME` existe ou non. Pour plus d'informations, consultez [CMD](#) dans la référence Dockerfile et [Définir une commande et des arguments pour un pod](#) dans la documentation. Kubernetes

Type : tableau de chaînes

Obligatoire : non

command

Point d'entrée du conteneur. Il n'est pas exécuté dans un shell. Si cela n'est pas spécifié, le `ENTRYPOINT` de l'image de conteneur est utilisé. Les références aux variables d'environnement sont étendues à l'aide de l'environnement du conteneur.

Si la variable d'environnement référencée n'existe pas, la référence de la commande ne change pas. Par exemple, si la référence porte sur « `$(NAME1)` » et que la variable d'environnement `NAME1` n'existe pas, la chaîne de commande reste « `$(NAME1)` ». `$$` est remplacé par `$`, et la chaîne résultante n'est pas développée. Par exemple, `$(VAR_NAME)` est transmis en tant que `$(VAR_NAME)`, que la variable d'environnement `VAR_NAME` existe ou non. Ce point d'entrée ne peut pas être mis à jour. Pour plus d'informations, consultez [ENTRYPOINT](#) dans la référence Dockerfile et [Définir une commande et des arguments pour un conteneur et Entrypoint](#) dans la documentation. Kubernetes

Type : tableau de chaînes

Obligatoire : non

env

Variables d'environnement à transmettre à un conteneur.

 Note

Les variables d'environnement ne peuvent pas commencer par « `AWS_BATCH` ». Cette convention de dénomination est réservée aux variables AWS Batch définies.

Type : tableau d'objets [EksContainerEnvironmentVariable](#)

Obligatoire : non

name

Nom de la variable d'environnement.

Type : String

Obligatoire : oui

value

Valeur de la variable d'environnement.

Type : chaîne

Obligatoire : non

image

Image Docker utilisée pour démarrer le conteneur.

Type : String

Obligatoire : oui

imagePullPolicy

Stratégie d'extraction d'images pour le conteneur. Les valeurs prises en charge sont Always, IfNotPresent et Never. Ce paramètre a pour valeur par défaut IfNotPresent. Toutefois, si la balise :latest est spécifiée, elle prend par défaut la valeur Always. Pour plus d'informations, consultez la section [Mise à jour des images](#) dans la Kubernetesdocumentation.

Type : chaîne

Obligatoire : non

name

Nom du conteneur. Si le nom n'est pas spécifié, le nom par défaut « Default » est utilisé. Chaque conteneur d'un pod doit porter un nom unique.

Type : chaîne

Obligatoire : non

resources

Le type et la quantité de ressources à attribuer à un conteneur. Les ressources prises en charge comprennent `memory`, `cpu` et `nvidia.com/gpu`. Pour plus d'informations, consultez la section [Gestion des ressources pour les pods et les conteneurs](#) dans la Kubernetesdocumentation.

Type : objet [EksContainerResourceRequirements](#)

Obligatoire : non

limits

Type et quantité de ressources spécifiés à réserver pour le conteneur. Les valeurs varient en fonction du name spécifié. Les ressources peuvent être demandées à l'aide des objets `limits` ou `requests`.

memory

Limite stricte de mémoire (en Mio) du conteneur, en entiers, avec un suffixe « Mi ». Si votre conteneur tente de dépasser la mémoire spécifiée, il est résilié. Vous devez spécifier au moins 4 Mio de mémoire pour une tâche. La `memory` peut être spécifiée dans `limits`, `requests` ou les deux. Si `memory` est spécifié dans les deux, la valeur spécifiée dans `limits` doit être égale à la valeur spécifiée dans `requests`.



Note

Pour optimiser l'utilisation de vos ressources, fournissez à vos tâches autant de mémoire que possible pour le type d'instance spécifique que vous utilisez. Pour savoir comment procéder, veuillez consulter la section [Gestion de la mémoire des ressources informatiques](#).

cpu

Le numéro CPUs est réservé au conteneur. Les valeurs doivent correspondre à un multiple pair de 0.25. Le `cpu` peut être spécifié dans `limits`, `requests` ou les deux. Si le `cpu` est spécifié dans les deux, la valeur spécifiée dans `limits` doit être au moins égale à la valeur spécifiée dans `requests`.

nvidia.com/gpu

Le numéro GPUs est réservé au conteneur. Les valeurs doivent correspondre à un entier. `memory` peut être spécifié dans `limits`, `requests` ou les deux. Si `memory` est spécifié dans les deux, la valeur spécifiée dans `limits` doit être égale à la valeur spécifiée dans `requests`.

Type : mappage chaîne/chaîne

Contraintes de longueur de valeur : longueur minimale de 1. Longueur maximale de 256.

Obligatoire : non

`requests`

Type et quantité de ressources spécifiés à demander pour le conteneur. Les valeurs varient en fonction du name spécifié. Les ressources peuvent être demandées à l'aide des objets `limits` ou `requests`.

`memory`

Limite stricte de mémoire (en Mio) du conteneur, en entiers, avec un suffixe « Mi ». Si votre conteneur tente de dépasser la mémoire spécifiée, il est résilié. Vous devez spécifier au moins 4 Mio de mémoire pour une tâche. La `memory` peut être spécifiée dans `limits`, `requests` ou les deux. Si `memory` est spécifié dans les deux, la valeur spécifiée dans `limits` doit être égale à la valeur spécifiée dans `requests`.

 Note

Si vous essayez de maximiser l'utilisation de vos ressources en fournissant à vos tâches autant de mémoire que possible pour un type d'instance particulier, consultez [Gestion de la mémoire des ressources informatiques](#).

`cpu`

Le nombre de CPUs ceux-ci sont réservés au conteneur. Les valeurs doivent correspondre à un multiple pair de 0.25. Le `cpu` peut être spécifié dans `limits`, `requests` ou les deux. Si le `cpu` est spécifié dans les deux, la valeur spécifiée dans `limits` doit être au moins égale à la valeur spécifiée dans `requests`.

nvidia.com/gpu

Le nombre de GPUs ceux-ci sont réservés au conteneur. Les valeurs doivent correspondre à un entier. `nvidia.com/gpu` peut être spécifié dans `limits`, `requests` ou les deux. Si `nvidia.com/gpu` est spécifié dans les deux, la valeur spécifiée dans `limits` doit être égale à la valeur spécifiée dans `requests`.

Type : mappage chaîne/chaîne

Contraintes de longueur de valeur : longueur minimale de 1. Longueur maximale de 256.

Obligatoire : non

securityContext

Contexte de sécurité d'une tâche. Pour plus d'informations, consultez la section [Configurer un contexte de sécurité pour un pod ou un conteneur](#) dans la Kubernetesdocumentation.

Type : objet [EksContainerSecurityContext](#)

Obligatoire : non

privileged

Lorsque ce paramètre est défini sur `true`, des autorisations élevées inhérentes à l'instance de conteneur hôte sont accordées au conteneur. Le niveau des autorisations est similaire à celui des autorisations des `root` utilisateurs. La valeur par défaut est `false`. Ce paramètre correspond à la `privileged` politique décrite dans les [politiques de sécurité du module Privileged](#) de la Kubernetesdocumentation.

Type : booléen

Obligatoire : non

readOnlyRootFilesystem

Lorsque ce paramètre est défini sur `true`, le conteneur ne dispose que d'un accès en lecture seule au système de fichiers racine. La valeur par défaut est `false`. Ce paramètre correspond à la `ReadOnlyRootFilesystem` politique décrite dans les [politiques de sécurité du module Volumes et systèmes de fichiers](#) de la Kubernetesdocumentation.

Type : booléen

Obligatoire : non

`runAsGroup`

Lorsque ce paramètre est spécifié, le conteneur est exécuté avec l'ID de groupe spécifié (`gid`). Si ce paramètre n'est pas spécifié, le groupe spécifié dans les métadonnées d'image correspond au groupe par défaut. Ce paramètre correspond à `RunAsGroup` la `MustRunAs` politique décrite dans les [politiques de sécurité du module Utilisateurs et groupes](#) de la `Kubernetesdocumentation`.

Type : long

Obligatoire : non

`runAsNonRoot`

Lorsque ce paramètre est spécifié, le conteneur est exécuté en tant qu'utilisateur avec un `uid` autre que 0. Si ce paramètre n'est pas spécifié, cette règle est appliquée. Ce paramètre correspond à `RunAsUser` la `MustRunAsNonRoot` politique décrite dans les [politiques de sécurité du module Utilisateurs et groupes](#) de la `Kubernetesdocumentation`.

Type : long

Obligatoire : non

`runAsUser`

Lorsque ce paramètre est spécifié, le conteneur est exécuté avec l'ID utilisateur spécifié (`uid`). Si ce paramètre n'est pas spécifié, le groupe spécifié dans les métadonnées d'image correspond au groupe par défaut. Ce paramètre correspond à `RunAsUser` la `MustRunAs` politique décrite dans les [politiques de sécurité du module Utilisateurs et groupes](#) de la `Kubernetesdocumentation`.

Type : long

Obligatoire : non

`volumeMounts`

Montages de volumes pour un conteneur dans le cadre d'une tâche Amazon EKS. Pour plus d'informations sur les volumes et les montages de volumes `Kubernetes`, consultez la section [Volumes](#) de la `Kubernetesdocumentation`.

Type : tableau d'objets [EksContainerVolumeMount](#)

Obligatoire : non

`mountPath`

Le chemin d'accès du conteneur, sur lequel le volume est monté.

Type : chaîne

Obligatoire : non

`name`

Nom du montage de volume. Ce nom doit correspondre au nom d'un des volumes du pod.

Type : chaîne

Obligatoire : non

`readOnly`

Si cette valeur est `true`, le conteneur ne peut accéder au volume qu'en lecture. Dans le cas contraire, le conteneur peut écrire sur le volume. La valeur par défaut est `false`.

Type : booléen

Obligatoire : non

`dnsPolicy`

Stratégie DNS du pod. La valeur par défaut est `ClusterFirst`. Si le paramètre `hostNetwork` n'est pas spécifié, la valeur par défaut est `ClusterFirstWithHostNet`. `ClusterFirst` indique que toute requête DNS ne correspondant pas au suffixe de domaine du cluster configuré est transmise au serveur de noms en amont hérité du nœud. Si aucune valeur n'a été spécifiée `dnsPolicy` dans l'opération [RegisterJobDefinition](#) d'API, aucune valeur n'est renvoyée `dnsPolicy` par l'une [DescribeJobDefinitions](#) ou l'autre des opérations d'[DescribeJobs](#) API. Le paramètre de spécification du pod contient `ClusterFirst` ou `ClusterFirstWithHostNet`, selon la valeur du paramètre `hostNetwork`. Pour plus d'informations, consultez la [politique DNS de Pod](#) dans la Kubernetes documentation.

Valeurs valides : `Default` | `ClusterFirst` | `ClusterFirstWithHostNet`

Type : chaîne

Obligatoire : non

hostNetwork

Indique si le pod utilise l'adresse IP réseau des hôtes. La valeur par défaut est `true`. Le réglage de cette option `false` permet d'activer le modèle de mise en réseau du Kubernetes pod. La plupart des AWS Batch charges de travail concernent uniquement les sorties et ne nécessitent pas la surcharge liée à l'allocation d'adresses IP pour chaque module pour les connexions entrantes. Pour plus d'informations, consultez les sections [Espaces de noms des hôtes](#) et [Mise en réseau](#) des pods dans la Kubernetesdocumentation.

Type : booléen

Obligatoire : non

serviceAccountName

Nom du compte de service utilisé pour exécuter le pod. Pour plus d'informations, consultez les [Kubernetessections Comptes](#) de service et [Configurer un compte de Kubernetes service pour assumer un rôle IAM](#) dans le Guide de l'utilisateur Amazon EKS et [Configurer les comptes de service pour les pods](#) dans la Kubernetesdocumentation.

Type : chaîne

Obligatoire : non

volumes

Spécifie les volumes d'une définition de tâche qui utilise des ressources Amazon EKS.

Type : tableau d'objets [EksVolume](#)

Obligatoire : non

Dir vide

Spécifie la configuration d'un Kubernetes `emptyDir` volume. Un volume `emptyDir` est créé pour la première fois lorsqu'un pod est attribué à un nœud. Il existe tant que ce pod fonctionne sur ce nœud. Le volume `emptyDir` est initialement vide. Tous les conteneurs du pod peuvent lire et écrire les fichiers du volume `emptyDir`. Toutefois, le volume `emptyDir` peut être monté sur des chemins identiques ou différents dans chaque conteneur. Lorsqu'un pod est supprimé d'un nœud pour quelque raison que ce soit, les

données du `emptyDir` sont définitivement supprimées. Pour plus d'informations, consultez [EmptyDir](#) dans la Kubernetes documentation.

Type : objet [EksEmptyDir](#)

Obligatoire : non

medium

Support sur lequel stocker le volume. La valeur par défaut est une chaîne vide utilisant le stockage du nœud.

""

(Par défaut) Utilisez le stockage sur disque du nœud.

« Mémoire »

Utilisez le volume `tmpfs` soutenu par la mémoire vive du nœud. Le contenu du volume est perdu lorsque le nœud redémarre, et tout stockage présent sur le volume est pris en compte dans la limite de mémoire du conteneur.

Type : chaîne

Obligatoire : non

Limite de taille

Taille maximale du volume. Par défaut, aucune taille maximale n'est définie.

Type : String

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : non

Chemin de l'hôte

Spécifie la configuration d'un Kubernetes `hostPath` volume. Un volume `hostPath` monte un fichier ou un répertoire existant depuis le système de fichiers du nœud hôte dans votre pod. Pour plus d'informations, consultez [HostPath](#) dans la Kubernetes documentation.

Type : objet [EksHostPath](#)

Obligatoire : non

chemin

Chemin du fichier ou du répertoire sur l'hôte à monter dans les conteneurs du pod.

Type : chaîne

Obligatoire : non

name

Le nom du volume. Le nom doit être autorisé en tant que nom de sous-domaine DNS. Pour plus d'informations, consultez la section [Noms de sous-domaines DNS](#) dans la Kubernetesdocumentation.

Type : String

Obligatoire : oui

secret

Spécifie la configuration d'un Kubernetes secret volume. Pour plus d'informations, voir [secret](#) dans la Kubernetesdocumentation.

Type : objet [EksSecret](#)

Obligatoire : non

facultatif

Spécifie si le secret ou les clés du secret doivent être définis.

Type : booléen

Obligatoire : non

Nom secret

Le nom du secret. Le nom doit être autorisé en tant que nom de sous-domaine DNS. Pour plus d'informations, consultez la section [Noms de sous-domaines DNS](#) dans la Kubernetesdocumentation.

Type : String

Obligatoire : oui

Capacités de la plateforme

`platformCapabilities`

Les fonctionnalités de la plateforme requises par la définition du poste. Si aucune valeur n'est spécifiée, elle est par défaut EC2. Pour les tâches exécutées sur des ressources FARGATE Fargate, est spécifié.

Note

Si la tâche s'exécute sur les ressources Amazon EKS, vous ne devez pas le spécifier `platformCapabilities`.

Type : String

Valeurs valides : EC2 | FARGATE

Obligatoire : non

Propager les balises

`propagateTags`

Spécifie s'il faut propager les balises de tâche ou de la définition de tâche vers la tâche Amazon ECS correspondante. Si aucune valeur n'est spécifiée, les balises ne sont pas propagées. Les balises ne peuvent être propagées aux tâches que lorsque celles-ci sont créées. Pour les balises portant le même nom, les balises de tâche ont la priorité sur les balises de définition de tâche. Si le nombre total de balises combinées entre la tâche et la définition de la tâche est supérieur à 50, la tâche est déplacée vers l'FAILED état.

Note

Si la tâche s'exécute sur les ressources Amazon EKS, vous ne devez pas le spécifier `propagateTags`.

Type : booléen

Obligatoire : non

Propriétés de nœud

nodeProperties

Lorsque vous enregistrez une définition de tâche parallèle à nœuds multiples, vous devez spécifier une liste de propriétés de nœuds. Ces propriétés de nœud définissent le nombre de nœuds à utiliser dans votre tâche, l'index des nœuds principaux et les différentes plages de nœuds à utiliser. Si la tâche s'exécute sur les ressources Fargate, vous ne pouvez pas le spécifier. `nodeProperties` Utilisez à la place `containerProperties`. Les propriétés de nœud suivantes sont autorisées dans une définition de tâche. Pour de plus amples informations, veuillez consulter [Tâches parallèles sur plusieurs nœuds](#).

Note

Si la tâche s'exécute sur les ressources Amazon EKS, vous ne devez pas le spécifier `nodeProperties`.

Type : objet [NodeProperties](#)

Obligatoire : non

mainNode

Spécifie l'index de nœud pour le nœud principal d'une tâche parallèle à plusieurs nœuds. Cette valeur d'index de nœud doit être inférieure au nombre de nœuds.

Type : entier

Obligatoire : oui

numNodes

Le nombre de nœuds associé à une tâche parallèle à plusieurs nœuds.

Type : entier

Obligatoire : oui

nodeRangeProperties

Une liste des plages de nœuds et leurs propriétés associées avec une tâche parallèle à plusieurs nœuds.

 Note

Un groupe de nœuds constitue un groupe identique de nœuds de tâches qui partagent tous les mêmes propriétés de conteneur. Vous pouvez spécifier jusqu'à AWS Batch à cinq groupes de nœuds distincts pour chaque tâche.

Type : tableau d'objets [NodeRangeProperty](#)

Obligatoire : oui

`targetNodes`

La plage de nœuds en utilisant les valeurs d'index de nœud. Une plage de nœuds 0:3 indique les nœuds dont les valeurs d'index sont comprises entre 0 et 3. Si la valeur de la plage de départ est omise (:n), 0 est utilisé pour démarrer la plage. Si la valeur de fin de la plage n'est pas spécifiée (n:), l'index de nœud le plus élevé possible est utilisé pour terminer la plage. Vos plages de nœuds cumulatives doivent tenir compte de tous les nœuds (0:n). Vous pouvez imbriquer des plages de nœuds, par exemple 0:10 et 4:5. Dans ce cas, les propriétés 4:5 de la plage remplacent les 0:10 propriétés.

Type : chaîne

Obligatoire : non

`container`

Les détails du conteneur pour la plage de nœuds. Pour de plus amples informations, veuillez consulter [Propriétés du contenant](#).

Type : objet [ContainerProperties](#)

Obligatoire : non

Stratégie de nouvelle tentative

`retryStrategy`

Lorsque vous enregistrez une définition de tâche, vous pouvez, si vous le souhaitez, définir une stratégie de nouvelle tentative à adopter pour les tâches en échec qui sont soumises

avec cette définition de tâche. Toute stratégie de nouvelle tentative spécifiée au cours d'une [SubmitJob](#) opération remplace la stratégie de nouvelle tentative définie ici. Par défaut, chaque tâche est lancée une seule fois. Si vous spécifiez plusieurs tentatives, la tâche est réessayée en cas d'échec. Parmi les exemples de tentative d'échec, citons le renvoi d'un code de sortie différent de zéro par la tâche ou l'arrêt de l'instance de conteneur. Pour de plus amples informations, veuillez consulter [Nouvelles tentatives de travail automatisées](#).

Type : objet [RetryStrategy](#)

Obligatoire : non

`attempts`

Nombre de fois où une tâche est passée en statut `RUNNABLE`. Vous pouvez indiquer entre 1 et 10 tentatives. Si `attempts` est supérieur à un, la tâche est relancée ce nombre de fois en cas d'échec, jusqu'à ce que son statut devienne `RUNNABLE`.

```
"attempts": integer
```

Type : entier

Obligatoire : non

`evaluateOnExit`

Tableau contenant jusqu'à 5 objets qui spécifient les conditions dans lesquelles la tâche est réessayée ou échoue. Si ce paramètre est spécifié, le paramètre `attempts` doit également être spécifié. Si elle `evaluateOnExit` est spécifiée mais qu'aucune des entrées ne correspond, la tâche est réessayée.

```
"evaluateOnExit": [  
  {  
    "action": "string",  
    "onExitCode": "string",  
    "onReason": "string",  
    "onStatusReason": "string"  
  }  
]
```

Type : tableau d'objets [EvaluateOnExit](#)

Obligatoire : non

action

Spécifie l'action à exécuter si toutes les conditions spécifiées (onStatusReason, onReason et onExitCode) sont remplies. Les valeurs ne sont pas sensibles à la casse.

Type : String

Obligatoire : oui

Valeurs valides : RETRY | EXIT

onExitCode

Contient un motif global à faire correspondre à la représentation décimale du résultat ExitCode renvoyé pour une tâche. Le modèle peut contenir jusqu'à 512 caractères. Il ne peut contenir que des chiffres. Il ne peut pas contenir de lettres ou de caractères spéciaux. Il peut éventuellement se terminer par un astérisque (*) ; seul le début de la chaîne doit constituer une correspondance exacte.

Type : chaîne

Obligatoire : non

onReason

Contient un motif global correspondant à Reason celui renvoyé pour une tâche. Le modèle peut contenir jusqu'à 512 caractères. Il peut contenir des lettres, des chiffres, des points (.), des points (:) et des espaces blancs (espaces, onglets). Il peut éventuellement se terminer par un astérisque (*) ; seul le début de la chaîne doit constituer une correspondance exacte.

Type : chaîne

Obligatoire : non

onStatusReason

Contient un motif global correspondant à StatusReason celui renvoyé pour une tâche. Le modèle peut contenir jusqu'à 512 caractères. Il peut contenir des lettres, des chiffres, des points (.), des points (:) et des espaces blancs (espaces, onglets). Il peut

éventuellement se terminer par un astérisque (*) ; seul le début de la chaîne doit constituer une correspondance exacte.

Type : chaîne

Obligatoire : non

Priorité de planification

`schedulingPriority`

Priorité de planification pour les tâches soumises avec cette définition de tâche. Cela ne concerne que les emplois en attente d'emploi dans le cadre d'une politique de partage équitable. Les tâches avec une priorité de planification plus élevée sont planifiées avant celles dont la priorité de planification est inférieure.

La valeur minimale prise en charge est 0 et la valeur maximale prise en charge est 9999.

Type : entier

Obligatoire : non

Balises

`tags`

Balises de paire clé-valeur à associer à la définition de la tâche. Pour de plus amples informations, veuillez consulter [Marquez vos AWS Batch ressources](#).

Type : mappage chaîne/chaîne

Obligatoire : non

Timeout

`timeout`

Vous pouvez configurer un délai d'expiration pour vos tâches de telle sorte que si une tâche dure plus longtemps, AWS Batch elle soit interrompue. Pour de plus amples informations, veuillez consulter [Expiration des délais d'exécution des tâches](#). Si une tâche est interrompue en raison

d'un délai d'attente, aucune nouvelle tentative n'est faite. Toute configuration de temporisation spécifiée au cours d'une [SubmitJob](#) opération remplace la configuration de temporisation définie ici. Pour de plus amples informations, veuillez consulter [Expiration des délais d'exécution des tâches](#).

Type : objet [JobTimeout](#)

Obligatoire : non

`attemptDurationSeconds`

Durée en secondes (mesurée à partir de l'`startedAt` horodatage de la tentative de tâche) après la AWS Batch fin des tâches inachevées. La valeur minimale pour le délai d'expiration est 60 secondes.

Pour les tâches de tableau, le délai s'applique aux tâches enfants, et non à la tâche de tableau parent.

Pour les tâches parallèles à plusieurs nœuds (MNP), le délai s'applique à l'ensemble de la tâche, et non aux nœuds individuels.

Type : entier

Obligatoire : non

Créez des définitions de tâches à l'aide de `EcsProperties`

Grâce aux définitions de AWS Batch tâches utilisées [EcsProperties](#), vous pouvez modéliser du matériel, des capteurs, des environnements 3D et d'autres simulations dans des conteneurs séparés. Vous pouvez utiliser cette fonctionnalité pour organiser de manière logique les composants de votre charge de travail et les séparer de l'application principale. Cette fonctionnalité peut être utilisée avec AWS Batch Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS) et AWS Fargate

`ContainerProperties` par rapport aux définitions de **`EcsProperties`** poste

Vous pouvez choisir d'utiliser [ContainerProperties](#) ou de [EcsProperties](#) définir des tâches selon votre cas d'utilisation. À un niveau élevé, l'exécution de AWS Batch tâches avec `EcsProperties` est similaire à l'exécution de tâches avec un `ContainerProperties`.

L'ancienne structure de définition des tâches, en utilisant `ContainerProperties`, reste prise en charge. Si vous disposez actuellement de flux de travail utilisant cette structure, vous pouvez continuer à les exécuter.

La principale différence réside dans le fait qu'un nouvel objet a été ajouté à la définition de tâche pour tenir compte des définitions `EcsProperties` basées sur des définitions.

Par exemple, une définition de tâche utilisée `ContainerProperties` sur Amazon ECS et Fargate possède la structure suivante :

```
{
  "containerProperties": {
    ...
    "image": "my_ecr_image1",
    ...
  },
  ...
}
```

Une définition de tâche utilisée `EcsProperties` sur Amazon ECS et Fargate possède la structure suivante :

```
{
  "ecsProperties": {
    "taskProperties": [{
      "containers": [
        {
          ...
          "image": "my_ecr_image1",
          ...
        },
        {
          ...
          "image": "my_ecr_image2",
          ...
        },
      ],
    }
  ]
}
```

Modifications générales apportées au AWS Batch APIs

Vous trouverez ci-dessous un aperçu de certaines des principales différences entre les types de données `ContainerProperties` et les `EcsProperties` API :

- La plupart des paramètres utilisés y `ContainerProperties` apparaissent `TaskContainerProperties`. Certains exemples incluent `command`, `image`, `privileged`, `secrets`, et `users`. Ils se trouvent tous à l'intérieur [TaskContainerProperties](#).
- Certains `TaskContainerProperties` paramètres n'ont pas d'équivalent fonctionnel dans l'ancienne structure. Certains exemples incluent `dependsOn`, `essential`, `name`, `ipcMode`, et `tpidMode`. Pour plus d'informations, consultez [EcsTaskDetails](#) et [TaskContainerProperties](#).

De plus, certains `ContainerProperties` paramètres n'ont pas d'équivalent ou d'application dans la `EcsProperties` structure. In [taskProperties](#), `container` a été remplacé par `containers` sorte que le nouvel objet puisse accepter jusqu'à dix éléments. [Pour plus d'informations, voir : Propriétés du conteneur et : conteneursRegisterJobDefinition.EcsTaskProperties](#)

- `taskRoleArn` est fonctionnellement équivalent à `jobRoleArn`. Pour plus d'informations, voir [EcsTaskProperties: taskRoleArn](#) et [ContainerProperties: jobRoleArn](#).
- Vous pouvez inclure de un (1) à dix (10) conteneurs dans la `EcsProperties` structure. [Pour plus d'informations, voir : conteneursEcsTaskProperties](#).
- Les objets `taskProperties` et `InstanceTypes` sont des tableaux, mais ils n'acceptent actuellement qu'un seul élément. [Par exemple, :TaskProperties et:InstanceTypesEcsProperties.NodeRangeProperty](#)

Définitions de tâches multi-conteneurs pour Amazon ECS

Pour s'adapter à la structure multi-conteneurs d'Amazon ECS, certains types de données d'API sont différents. Par exemple,

- [ecsProperties](#) est le même niveau que `containerProperties` dans la définition du conteneur unique. Pour plus d'informations, consultez [EcsProperties](#) dans le Guide de référence des API AWS Batch .
- [taskProperties](#) contient les propriétés définies pour la tâche Amazon ECS. Pour plus d'informations, consultez [EcsProperties](#) dans le Guide de référence des API AWS Batch .
- [containers](#) inclut des informations similaires à celles `containerProperties` de la définition du conteneur unique. La principale différence est que `containers` vous pouvez définir jusqu'à dix conteneurs. Pour plus d'informations, consultez [ECSTaskProperties:Containers](#) dans le guide de référence des AWS Batch API.

- [essential](#) le paramètre indique comment le conteneur affecte la tâche. Tous les conteneurs essentiels doivent être terminés avec succès (sortie en tant que 0) pour que le travail progresse. Si un conteneur marqué comme essentiel échoue (sort en tant que non-0), la tâche échoue.

La valeur par défaut est `true` et au moins un conteneur doit être marqué comme `essential`. Pour plus d'informations, consultez [essential](#) dans le Guide de référence des API AWS Batch .

- Avec le [dependsOn](#) paramètre, vous pouvez définir une liste de dépendances de conteneurs. Pour plus d'informations, consultez [dependsOn](#) dans le Guide de référence des API AWS Batch .

Note

La complexité de la `dependsOn` liste et le temps d'exécution du conteneur associé peuvent affecter l'heure de début de votre tâche. Si les dépendances mettent du temps à s'exécuter, la tâche restera inchangée jusqu'à ce `STARTING` qu'elle soit terminée.

Pour plus d'informations sur la structure `ecsProperties` et, consultez la syntaxe des [RegisterJobDefinition](#) demandes pour [ECSPProperties](#).

Définitions de tâches multi-conteneurs pour Amazon EKS

Pour s'adapter à la structure multi-conteneurs d'Amazon EKS, certains types de données d'API sont différents. Par exemple,

- [name](#) est un identifiant unique pour le conteneur. Cet objet n'est pas obligatoire pour un seul conteneur, mais il est obligatoire lors de la définition de plusieurs conteneurs dans un pod. Lorsqu'il `name` n'est pas défini pour des conteneurs individuels, le nom par défaut `default`, est appliqué.
- [initContainers](#) sont définis dans le type de [eksPodProperties](#) données. Ils s'exécutent avant les conteneurs d'applications, s'exécutent toujours jusqu'à la fin et doivent s'achever correctement avant le démarrage du conteneur suivant.

Ces conteneurs sont enregistrés auprès de l'agent Amazon EKS Connector et conservent les informations d'enregistrement dans le magasin de données principal Amazon Elastic Kubernetes Service. L'`initContainers` objet peut accepter jusqu'à dix (10) éléments. Pour plus d'informations, consultez la section [Init Containers](#) dans la Kubernetes documentation.

Note

L'`initContainers` objet peut affecter l'heure de début de votre tâche. Si leur exécution `initContainers` prend du temps, la tâche restera dans son `STARTING` état jusqu'à ce qu'elle soit terminée.

- [shareProcessNamespace](#) indique si les conteneurs du pod peuvent partager le même espace de noms de processus. Les valeurs par défaut sont `false`. Définissez ce paramètre sur `true` pour permettre aux conteneurs de voir et de signaler les processus dans d'autres conteneurs situés dans le même module.
- Chaque contenant a de l'importance. Tous les conteneurs doivent être terminés avec succès (sortie sous la forme 0) pour que la tâche réussisse. Si un conteneur échoue (sort sous une autre forme que 0), la tâche échoue.

Pour plus d'informations sur la structure `eksProperties` et, consultez la syntaxe des [RegisterJobDefinition](#) demandes pour [EksProperties](#).

Référence : AWS Batch scénarios professionnels utilisant `EcsProperties`

Pour illustrer comment les définitions de AWS Batch poste utilisées `EcsProperties` peuvent être structurées en fonction de vos besoins, cette rubrique présente les [RegisterJobDefinition](#) charges utiles suivantes. Vous pouvez copier ces exemples dans un fichier, les personnaliser selon vos besoins, puis utiliser le AWS Command Line Interface (AWS CLI) pour appeler `RegisterJobDefinition`.

AWS Batch offre d'emploi pour Amazon ECS sur Amazon EC2

Voici un exemple de AWS Batch tâche pour Amazon Elastic Container Service sur Amazon Elastic Compute Cloud :

```
{
  "jobDefinitionName": "multicontainer-ecs-ec2",
  "type": "container",
  "ecsProperties": {
    "taskProperties": [
      {
        "containers": [
```

```
    "name": "c1",
    "essential": false,
    "command": [
        "echo",
        "hello world"
    ],
    "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
    "resourceRequirements": [
        {
            "type": "VCPU",
            "value": "2"
        },
        {
            "type": "MEMORY",
            "value": "4096"
        }
    ]
},
{
    "name": "c2",
    "essential": false,
    "command": [
        "echo",
        "hello world"
    ],
    "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
    "resourceRequirements": [
        {
            "type": "VCPU",
            "value": "2"
        },
        {
            "type": "MEMORY",
            "value": "4096"
        }
    ]
},
{
    "name": "c3",
    "essential": true,
    "command": [
        "echo",
        "hello world"
    ],
}
```

```

        "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
        "firelensConfiguration": {
            "type": "fluentbit",
            "options": {
                "enable-ecs-log-metadata": "true"
            }
        },
        "resourceRequirements": [
            {
                "type": "VCPU",
                "value": "6"
            },
            {
                "type": "MEMORY",
                "value": "12288"
            }
        ]
    }
}

```

AWS Batch offre d'emploi pour Amazon ECS sur Fargate

Voici un exemple de AWS Batch tâche pour Amazon Elastic Container Service sur AWS Fargate :

```

{
    "jobDefinitionName": "multicontainer-ecs-fargate",
    "type": "container",
    "platformCapabilities": [
        "FARGATE"
    ],
    "ecsProperties": {
        "taskProperties": [
            {
                "containers": [
                    {
                        "name": "c1",
                        "command": [
                            "echo",
                            "hello world"
                        ]
                    }
                ]
            }
        ]
    }
}

```

```

    ],
    "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
    "resourceRequirements": [
      {
        "type": "VCPU",
        "value": "2"
      },
      {
        "type": "MEMORY",
        "value": "4096"
      }
    ]
  },
  {
    "name": "c2",
    "essential": true,
    "command": [
      "echo",
      "hello world"
    ],
    "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
    "resourceRequirements": [
      {
        "type": "VCPU",
        "value": "6"
      },
      {
        "type": "MEMORY",
        "value": "12288"
      }
    ]
  }
],
"executionRoleArn": "arn:aws:iam::1112223333:role/ecsTaskExecutionRole"
}
]
}
}

```

AWS Batch job pour Amazon EKS

Voici un exemple de AWS Batch tâche pour Amazon Elastic Kubernetes Service :

```
{
  "jobDefinitionName": "multicontainer-eks",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "shareProcessNamespace": true,
      "initContainers": [
        {
          "name": "init-container",
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "echo"
          ],
          "args": [
            "hello world"
          ],
          "resources": {
            "requests": {
              "cpu": "1",
              "memory": "512Mi"
            }
          }
        },
        {
          "name": "init-container-2",
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "echo",
            "my second init container"
          ],
          "resources": {
            "requests": {
              "cpu": "1",
              "memory": "512Mi"
            }
          }
        }
      ],
      "containers": [
        {
          "name": "c1",
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
```

```

        "echo world"
    ],
    "resources": {
        "requests": {
            "cpu": "1",
            "memory": "512Mi"
        }
    }
},
{
    "name": "sleep-container",
    "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
    "command": [
        "sleep",
        "20"
    ],
    "resources": {
        "requests": {
            "cpu": "1",
            "memory": "512Mi"
        }
    }
}
]
}
}
}

```

Tâche AWS Batch MNP avec plusieurs conteneurs par nœud

Voici un exemple de AWS Batch tâche MNP (multi-node parallel) avec plusieurs conteneurs par nœud :

```

{
  "jobDefinitionName": "multicontainer-mnp",
  "type": "multinode",
  "nodeProperties": {
    "numNodes": 6,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes": "0:5",
        "ecsProperties": {

```

```
"taskProperties": [
  {
    "containers": [
      {
        "name": "range05-c1",
        "command": [
          "echo",
          "hello world"
        ],
        "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
        "resourceRequirements": [
          {
            "type": "VCPU",
            "value": "2"
          },
          {
            "type": "MEMORY",
            "value": "4096"
          }
        ]
      },
      {
        "name": "range05-c2",
        "command": [
          "echo",
          "hello world"
        ],
        "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
        "resourceRequirements": [
          {
            "type": "VCPU",
            "value": "2"
          },
          {
            "type": "MEMORY",
            "value": "4096"
          }
        ]
      }
    ]
  }
]
```

```
]
}
}
```

Utiliser le pilote de journal awslogs

Par défaut, AWS Batch permet au pilote de `awslogs` journal d'envoyer des informations de journal à CloudWatch Logs. Vous pouvez utiliser cette fonctionnalité pour afficher les différents journaux de vos conteneurs en un seul endroit pratique et éviter que les journaux de vos conteneurs n'occupent de l'espace disque sur vos instances de conteneurs. Cette rubrique vous aide à configurer le pilote de `awslogs` journal dans vos définitions de tâches.

Note

Dans la AWS Batch console, vous pouvez configurer le pilote de `awslogs` journal dans la section Configuration de la journalisation lorsque vous créez une définition de tâche.

Note

Le type d'informations enregistrées par les conteneurs dans le cadre de votre tâche dépend principalement de leur `ENTRYPOINT` commande. Par défaut, les journaux capturés affichent la sortie de commande que vous voyez normalement dans un terminal interactif si vous avez exécuté le conteneur localement, à savoir les `STDERR` I/O flux `STDOUT` et. Le pilote de `awslogs` journal transmet simplement ces journaux de Docker à CloudWatch Logs. Pour plus d'informations sur la façon dont les journaux Docker sont traités, et notamment sur les autres façons de capturer différentes données de fichiers ou différents flux, consultez [View logs for a container or service](#) dans la documentation Docker.

Pour envoyer des journaux système depuis vos instances de conteneur vers CloudWatch Logs, consultez [Utilisation CloudWatch des journaux avec AWS Batch](#). Pour plus d'informations sur CloudWatch les journaux, consultez la section [Surveillance des fichiers journaux](#) et [CloudWatch des quotas de journaux](#) dans le guide de l'utilisateur Amazon CloudWatch Logs.

options du pilote de journalisation awslogs dans le type de données AWS Batch JobDefinition

Le pilote de `awslogs journal` prend en charge les options suivantes dans les définitions de AWS Batch tâches. Pour plus d'informations, consultez la section [Pilote de journalisation des CloudWatch journaux](#) dans la documentation Docker.

`awslogs-region`

Obligatoire : non

Spécifiez la région dans laquelle le pilote de `awslogs journal` doit envoyer vos journaux Docker. Par défaut, la région utilisée est la même que celle de la tâche. Vous pouvez choisir d'envoyer tous vos journaux provenant de tâches situées dans différentes régions vers une seule région dans CloudWatch Logs. Cela leur permet d'être tous visibles à partir d'un seul endroit. Vous pouvez également les séparer par région pour une approche plus précise. Toutefois, lorsque vous choisissez cette option, assurez-vous que les groupes de journaux spécifiés existent dans la région que vous avez spécifiée.

`awslogs-group`

Obligatoire : Facultatif

Avec `awslogs-group` cette option, vous pouvez spécifier le groupe de journaux auquel le pilote de `awslogs journal` envoie ses flux de journaux. Si cela n'est pas spécifié, `aws/batch/job` est utilisé.

`awslogs-stream-prefix`

Obligatoire : Facultatif

Avec `awslogs-stream-prefix` cette option, vous pouvez associer un flux de journal au préfixe spécifié et à l'ID de tâche Amazon ECS de la AWS Batch tâche à laquelle appartient le conteneur. Si vous spécifiez un préfixe avec cette option, le format du flux de journaux est le suivant :

```
prefix-name/default/ecs-task-id
```

`awslogs-datetime-format`

Obligatoire : non

Cette option définit un modèle de démarrage à plusieurs lignes au format `strftime` Python. Un message de journal se compose d'une ligne qui correspond au modèle et de toutes les lignes suivantes qui ne correspondent pas au modèle. Ainsi, la ligne mise en correspondance est le délimiteur entre les messages de journalisation.

Ce format peut, par exemple, servir à analyser une sortie comme une pile de vidage, laquelle pourrait, dans le cas contraire, être consignée en plusieurs entrées. Le modèle adéquat permet de la capturer dans une seule entrée.

Pour de plus amples informations, veuillez consulter [awslogs-datetime-format](#).

Cette option est toujours prioritaire si `awslogs-datetime-format` et `awslogs-multiline-pattern` sont configurés.

 Note

La journalisation multiligne effectue l'analyse et la mise en correspondance des expressions régulières de tous les messages de journalisation. Cela peut avoir un impact négatif sur les performances de journalisation.

`awslogs-multiline-pattern`

Obligatoire : non

Cette option définit un modèle de démarrage à plusieurs lignes à l'aide d'une expression régulière. Un message de journal se compose d'une ligne qui correspond au modèle et de toutes les lignes suivantes qui ne correspondent pas au modèle. Ainsi, la ligne correspondante est le délimiteur entre les messages du journal.

Pour plus d'informations, consultez [awslogs-multiline-pattern](#) la documentation Docker.

Cette option est ignorée si `awslogs-datetime-format` est également configuré.

 Note

La journalisation multiligne effectue l'analyse et la mise en correspondance des expressions régulières de tous les messages de journalisation. Cela peut avoir un impact négatif sur les performances de journalisation.

awslogs-create-group

Obligatoire : non

Spécifiez si vous voulez que le groupe de journaux soit automatiquement créé. Si cette option n'est pas spécifiée, l'emplacement par défaut est `false`.

Warning

Cette option n'est pas recommandée. Nous vous recommandons de créer le groupe de CloudWatch journaux à l'avance à l'aide de l'action Logs [CreateLogGroup](#) API car chaque tâche tente de créer le groupe de journaux, ce qui augmente le risque d'échec de la tâche.

Note

La politique IAM de votre rôle d'exécution doit inclure l'`logs:CreateLogGroup` autorisation avant toute tentative d'utilisation `awslogs-create-group`.

Spécifiez une configuration de journal dans la définition de votre tâche

Par défaut, AWS Batch active le pilote de `awslogs` journal. Cette section décrit comment personnaliser la configuration du `awslogs` journal pour une tâche. Pour de plus amples informations, veuillez consulter [Création d'une définition de tâche à nœud unique](#).

Les extraits de code JSON de configuration de journal suivants contiennent un `logConfiguration` objet spécifié pour chaque tâche. L'un concerne une WordPress tâche qui envoie des journaux à un groupe de journaux appelé `awslogs-wordpress`, et l'autre concerne un conteneur MySQL qui envoie des journaux à un groupe de journaux appelé `awslogs-mysql`. Les deux conteneurs utilisent le préfixe de flux de journal `awslogs-example`.

```
"logConfiguration": {
  "logDriver": "awslogs",
  "options": {
    "awslogs-group": "awslogs-wordpress",
    "awslogs-stream-prefix": "awslogs-example"
  }
}
```

```
}
```

```
"logConfiguration": {  
  "logDriver": "awslogs",  
  "options": {  
    "awslogs-group": "awslogs-mysql",  
    "awslogs-stream-prefix": "awslogs-example"  
  }  
}
```

Dans la AWS Batch console, la configuration du journal pour la définition de wordpress tâche est spécifiée comme indiqué dans l'image suivante.

Log configuration

Log driver

awslogs ▼

Options

Name	Value	
awslogs-group ▼	awslogs-wordpress	Remove option
awslogs-stream-prefix ▼	awslogs-example	Remove option

Add option

Secrets

Add secret

Après avoir enregistré une définition de tâche avec le pilote de awslogs journal dans une configuration de journal de définition de tâche, vous pouvez soumettre une tâche avec cette définition de tâche pour commencer à envoyer des CloudWatch journaux à Logs. Pour de plus amples informations, veuillez consulter [Tutoriel : soumettre une offre d'emploi](#).

Spécifier les données sensibles

Vous pouvez ainsi injecter des données sensibles dans vos tâches en les stockant dans des AWS Secrets Manager secrets ou dans des paramètres AWS Systems Manager Parameter Store, puis en les référençant dans votre définition de tâche. AWS Batch

Les secrets d'une offre d'emploi peuvent être exposés de différentes manières :

- Pour injecter des données sensibles dans vos conteneurs sous forme de variables d'environnement, utilisez le paramètre de définition de `secrets` tâche.
- Pour référencer des informations sensibles dans la configuration du journal d'une tâche, utilisez le paramètre de définition de `secretOptions` tâche.

Rubriques

- [Spécifiez les données sensibles avec Secrets Manager](#)
- [Spécifiez les données sensibles avec Systems Manager Parameter Store](#)

Spécifiez les données sensibles avec Secrets Manager

Avec AWS Batch, vous pouvez injecter des données sensibles dans vos tâches en stockant vos données sensibles en AWS Secrets Manager secret, puis en les référençant dans votre définition de tâche. Les données sensibles stockées dans les secrets de Secrets Manager peuvent être exposées à une tâche sous forme de variables d'environnement ou dans le cadre de la configuration du journal.

Lorsque vous injectez un secret en tant que variable d'environnement, vous pouvez spécifier une clé JSON ou une version d'un secret à injecter. Ce processus vous permet de contrôler les données sensibles exposées à votre travail. Pour de plus amples informations sur la gestion des versions de secrets, veuillez consulter [Termes et concepts clés pour AWS Secrets Manager](#) dans le Guide de l'utilisateur AWS Secrets Manager .

Considérations à prendre en compte lors de la spécification de données sensibles à l'aide de Secrets Manager

Les points suivants doivent être pris en compte lors de l'utilisation de Secrets Manager pour spécifier des données sensibles pour des tâches.

- Pour injecter un secret à l'aide d'une clé JSON ou d'une version spécifique d'un secret, la version 1.37.0 ou ultérieure de l'agent de conteneur Amazon ECS doit être installée sur l'instance de conteneur de votre environnement informatique. Toutefois, nous vous recommandons d'utiliser la dernière version de l'agent de conteneur. Pour plus d'informations sur la vérification de la version de votre agent et la mise à jour vers la dernière version, consultez la section [Mise à jour de l'agent de conteneur Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.

Pour injecter le contenu complet d'un secret en tant que variable d'environnement ou pour injecter un secret dans une configuration de journal, votre instance de conteneur doit disposer de la version 1.23.0 ou ultérieure de l'agent de conteneur.

- Seuls les secrets qui stockent des données texte, qui sont des secrets créés avec le `SecretString` paramètre de l'[CreateSecretAPI](#), sont pris en charge. Les secrets qui stockent des données binaires, qui sont des secrets créés avec le `SecretBinary` paramètre de l'[CreateSecretAPI](#), ne sont pas pris en charge.
- Lorsque vous utilisez une définition de tâche qui fait référence aux secrets de Secrets Manager pour récupérer des données sensibles pour vos tâches, si vous utilisez également des points de terminaison VPC d'interface, vous devez créer les points de terminaison VPC d'interface pour Secrets Manager. Pour de plus amples informations, veuillez consulter [Utilisation de Secrets Manager avec des points de terminaison de VPC](#) dans le Guide de l'utilisateur AWS Secrets Manager .
- Des données sensibles sont injectées dans votre tâche lors de son lancement initial. Si le secret est ensuite mis à jour ou pivoté, la tâche ne reçoit pas automatiquement la valeur mise à jour. Vous devez lancer une nouvelle tâche pour forcer le service à lancer une nouvelle tâche avec la valeur secrète mise à jour.

Autorisations IAM requises pour les secrets AWS Batch

Pour utiliser cette fonctionnalité, vous devez disposer du rôle d'exécution et le référencer dans votre définition de tâche. Cela permet à l'agent de conteneur d'extraire les ressources Secrets Manager nécessaires. Pour de plus amples informations, veuillez consulter [AWS Batch Rôle d'exécution IAM](#).

Pour donner accès aux secrets de Secrets Manager que vous créez, ajoutez manuellement les autorisations suivantes en tant que politique intégrée au rôle d'exécution. Pour obtenir des informations, consultez la section [Ajout et suppression de politiques IAM](#) dans le Guide de l'utilisateur IAM.

- `secretsmanager:GetSecretValue` : obligatoire si vous faites référence à un secret Secrets Manager.
- `kms:Decrypt` : obligatoire uniquement si votre secret utilise une clé KMS personnalisée et non la clé par défaut. L'ARN de votre clé personnalisée doit être ajouté en tant que ressource.

L'exemple suivant de politique en ligne ajoute les autorisations requises.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:secretsmanager:us-east-2:777777777777:secret:<secret_name>",
        "arn:aws:kms:us-east-2:777777777777:key/<key_id>"
      ]
    }
  ]
}
```

Injection de données sensibles sous la forme d'une variable d'environnement

Vous pouvez spécifier les éléments suivants dans votre définition de tâche :

- L'`secret` objet contenant le nom de la variable d'environnement à définir dans le job
- Amazon Resource Name (ARN) du secret Secrets Manager
- Paramètres supplémentaires contenant les données sensibles à présenter à la tâche

L'exemple suivant montre la syntaxe complète qui doit être spécifiée pour le secret Secrets Manager.

```
arn:aws:secretsmanager:region:aws_account_id:secret:secret-name:json-key:version-stage:version-id
```

La section suivante décrit les paramètres supplémentaires. Ces paramètres sont facultatifs.

Toutefois, si vous ne les utilisez pas, vous devez inclure les deux points : pour utiliser les valeurs par défaut. Des exemples sont donnés ci-dessous pour plus de contexte.

json-key

Spécifie le nom de la clé dans une paire clé-valeur avec la valeur que vous souhaitez définir comme valeur de variable d'environnement. Seules les valeurs au format JSON sont prises en charge. Si vous ne spécifiez pas de clé JSON, le contenu complet du secret est utilisé.

version-stage

Spécifie l'étiquette intermédiaire de la version d'un secret que vous souhaitez utiliser. Si une étiquette intermédiaire de version est spécifiée, vous ne pouvez pas spécifier d'ID de version. Si aucune étape de version n'est spécifiée, le comportement par défaut consiste à récupérer le secret avec l'étiquette AWSCURRENT intermédiaire.

Les étiquettes intermédiaires sont utilisées pour suivre les différentes versions d'un secret lorsqu'elles sont mises à jour ou font l'objet d'une rotation. Chaque version d'un secret a une ou plusieurs étiquettes intermédiaires et un ID. Pour plus d'informations, consultez la section [Termes et concepts clés de AWS Secrets Manager](#) dans le guide de AWS Secrets Manager l'utilisateur.

version-id

Spécifie l'identifiant unique de la version du secret que vous souhaitez utiliser. Si un ID de version est spécifié, vous ne pouvez pas spécifier d'étiquette intermédiaire de version. Si aucun ID de version n'est spécifié, le comportement par défaut consiste à récupérer le secret avec l'étiquette AWSCURRENT intermédiaire.

IDs Les versions sont utilisées pour suivre les différentes versions d'un secret lors de leur mise à jour ou de leur rotation. Chaque version d'un secret a un ID. Pour plus d'informations, consultez la section [Termes et concepts clés de AWS Secrets Manager](#) dans le guide de AWS Secrets Manager l'utilisateur.

Exemples de définitions de conteneur

Les exemples suivants montrent comment vous pouvez référencer des secrets Secrets Manager dans vos définitions de conteneur.

Exemple référencement d'un secret complet

Voici un extrait d'une définition de tâche montrant le format lorsque vous référencez le texte complet d'un secret Secrets Manager.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-AbCdEf"
    }]
  }]
}
```

Exemple référencement d'une clé spécifique dans un secret

Voici un exemple de sortie d'une `get-secret-value` commande [≥](#) qui affiche le contenu d'un secret ainsi que le libellé de la version intermédiaire et l'ID de version qui lui sont associés.

```
{
  "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
  "Name": "appauthexample",
  "VersionId": "871d9eca-18aa-46a9-8785-981dd39ab30c",
  "SecretString": "{\"username1\":\"password1\",\"username2\":\"password2\", \"username3\":\"password3\"}",
  "VersionStages": [
    "AWSCURRENT"
  ],
  "CreateDate": 1581968848.921
}
```

Référence d'une clé spécifique de la sortie précédente dans une définition de conteneur en spécifiant le nom de la clé à la fin de l'ARN.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf:username1::"
    }]
  }]
}
```

```
    ]]
  }
```

Exemple référencement d'une version secrète spécifique

Voici un exemple de sortie d'une commande [>describe-secret](#) qui affiche le contenu non chiffré d'un secret ainsi que les métadonnées de toutes les versions du secret.

```
{
  "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
  "Name": "appauthexample",
  "Description": "Example of a secret containing application authorization data.",
  "RotationEnabled": false,
  "LastChangedDate": 1581968848.926,
  "LastAccessedDate": 1581897600.0,
  "Tags": [],
  "VersionIdsToStages": {
    "871d9eca-18aa-46a9-8785-981dd39ab30c": [
      "AWSCURRENT"
    ],
    "9d4cb84b-ad69-40c0-a0ab-cead36b967e8": [
      "AWSPREVIOUS"
    ]
  }
}
```

Référence d'une étiquette intermédiaire de version spécifique à partir de la sortie précédente dans une définition de conteneur en spécifiant le nom de la clé à la fin de l'ARN.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf::AWSPREVIOUS:"
    }]
  }]
}
```

Référence un ID de version spécifique de la sortie précédente dans une définition de conteneur en spécifiant le nom de clé à la fin de l'ARN.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf::9d4cb84b-ad69-40c0-a0ab-cead36b967e8"
    }]
  }]
}
```

Exemple référencement d'une clé spécifique et d'une étiquette intermédiaire de version d'un secret

Ce qui suit montre comment référencer à la fois une clé spécifique dans une étiquette secrète et une étiquette de mise en scène de version spécifique.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf:username1:AWSPREVIOUS:"
    }]
  }]
}
```

Pour spécifier une clé et un ID de version spécifiques, utilisez la syntaxe suivante.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf:username1::9d4cb84b-ad69-40c0-a0ab-cead36b967e8"
    }]
  }]
}
```

Injecter des données sensibles dans une configuration de journal

Lorsque vous spécifiez une option `logConfiguration` dans votre définition de tâche, vous pouvez indiquer `secretOptions` le nom de l'option de pilote de journal à définir dans le conteneur et l'ARN complet du secret Secrets Manager contenant les données sensibles à présenter au conteneur.

Ce qui suit est un extrait d'une définition de tâche indiquant le format lors de la référence à un secret de Secrets Manager.

```
{
  "containerProperties": [{
    "logConfiguration": [{
      "logDriver": "splunk",
      "options": {
        "splunk-url": "https://cloud.splunk.com:8080"
      },
      "secretOptions": [{
        "name": "splunk-token",
        "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-AbCdEf"
      }]
    }]
  }]
}
```

Créez un AWS Secrets Manager secret

Vous pouvez utiliser la console Secrets Manager afin de créer un secret pour vos données sensibles. Pour plus d'informations, veuillez consulter la rubrique [Didacticiel : création et récupération d'un secret](#) dans le Guide de l'utilisateur AWS Secrets Manager .

Pour créer un secret basique

Utilisez Secrets Manager afin de créer un secret pour vos données sensibles.

1. Ouvrez la console Secrets Manager à l'adresse <https://console.aws.amazon.com/secretsmanager/>.
2. Choisissez Store a new secret (Stocker un nouveau secret).
3. Pour Select secret type (Sélectionner un type de secret), choisissez Other type of secrets (Autre type de secrets).

4. Précisez les détails de votre secret personnalisé sous forme de paires de Key (Clé) et Value (Valeur). Par exemple, vous pouvez spécifier la clé `UserName`, puis fournir le nom d'utilisateur approprié comme valeur. Ajoutez une deuxième clé avec le nom `Password` et le texte du mot de passe comme valeur. Vous pouvez également ajouter des entrées pour un nom de base de données, une adresse de serveur ou un port TCP. Vous pouvez ajouter autant de paires que nécessaire pour stocker les informations dont vous avez besoin.

Vous pouvez également choisir l'onglet Plaintext (Texte brut) et saisir la valeur du secret comme vous le souhaitez.

5. Choisissez la clé de AWS KMS chiffrement que vous souhaitez utiliser pour chiffrer le texte protégé dans le secret. Si vous ne choisissez aucune clé, Secrets Manager vérifie s'il existe une clé par défaut pour le compte et l'utilise si c'est le cas. S'il n'existe aucune clé par défaut, Secrets Manager en crée une automatiquement. Vous pouvez également choisir `Add new key` (Ajouter une clé) pour créer une clé KMS personnalisée spécifiquement pour ce secret. Pour créer votre propre clé KMS, vous devez être autorisé à créer des clés KMS dans votre compte.
6. Choisissez Suivant.
7. Pour Secret name (Nom du secret), saisissez un chemin facultatif et un nom, comme **production/MyAwesomeAppSecret** ou **development/TestSecret**, puis choisissez Next (Suivant). Vous pouvez éventuellement ajouter une description pour vous aider à vous souvenir de la finalité de ce secret ultérieurement.

Le nom du secret doit contenir uniquement des lettres ASCII, des chiffres ou les caractères suivants : `/_+=.@-`

8. (Facultatif) À ce stade, vous pouvez configurer la rotation de votre secret. Pour cette procédure, gardez l'option `Disable automatic rotation` (Désactiver la rotation automatique) et choisissez Next (Suivant).

Pour plus d'informations sur la façon de configurer la rotation sur des secrets nouveaux ou existants, voir [Rotation de vos AWS Secrets Manager secrets](#).

9. Vérifiez vos paramètres, puis choisissez `Store secret` (Stocker le secret) pour enregistrer tous les éléments que vous avez saisis en tant que nouveau secret dans Secrets Manager.

Spécifiez les données sensibles avec Systems Manager Parameter Store

Vous pouvez ainsi injecter des données sensibles dans vos conteneurs en les stockant dans les paramètres du AWS Systems Manager Parameter Store, puis en les référençant dans la définition de votre conteneur. AWS Batch

Rubriques

- [Considérations à prendre en compte lors de la spécification de données sensibles à l'aide du magasin de paramètres de Systems Manager](#)
- [Autorisations IAM requises pour les secrets AWS Batch](#)
- [Injectez des données sensibles en tant que variable d'environnement](#)
- [Injecter des données sensibles dans une configuration de journal](#)
- [Création d'un AWS Systems Manager paramètre Parameter Store](#)

Considérations à prendre en compte lors de la spécification de données sensibles à l'aide du magasin de paramètres de Systems Manager

Les éléments suivants doivent être pris en compte lors de la spécification de données sensibles pour les conteneurs à l'aide des paramètres Systems Manager Parameter Store.

- Cette fonctionnalité nécessite que votre instance de conteneur dispose de la version 1.23.0 ou ultérieure de l'agent de conteneur. Toutefois, nous vous recommandons d'utiliser la dernière version de l'agent de conteneur. Pour plus d'informations sur la vérification de la version de votre agent et la mise à jour vers la dernière version, consultez la section [Mise à jour de l'agent de conteneur Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.
- Les données sensibles sont injectées dans le conteneur pour votre tâche lors du démarrage initial du conteneur. Si le paramètre secret ou Parameter Store est ensuite mis à jour ou fait l'objet d'une rotation, le conteneur ne reçoit pas la valeur mise à jour automatiquement. Vous devez lancer une nouvelle tâche pour forcer le lancement d'une nouvelle tâche avec des secrets mis à jour.

Autorisations IAM requises pour les secrets AWS Batch

Pour utiliser cette fonctionnalité, vous devez disposer du rôle d'exécution et le référencer dans votre définition de tâche. Cela permet à l'agent de conteneur Amazon ECS de récupérer les AWS Systems Manager ressources nécessaires. Pour de plus amples informations, veuillez consulter [AWS Batch Rôle d'exécution IAM](#).

Pour donner accès aux AWS Systems Manager paramètres du magasin de paramètres que vous créez, ajoutez manuellement les autorisations suivantes en tant que politique intégrée au rôle d'exécution. Pour obtenir des informations, consultez la section [Ajout et suppression de politiques IAM](#) dans le Guide de l'utilisateur IAM.

- `ssm:GetParameters` : obligatoire lorsque vous référencez un paramètre Systems Manager Parameter Store dans une définition de tâche.
- `secretsmanager:GetSecretValue` : obligatoire si vous référencez directement un secret Secrets Manager ou si votre paramètre Systems Manager Parameter Store référence un secret Secrets Manager dans une définition de tâche.
- `kms:Decrypt` : obligatoire uniquement si votre secret utilise une clé KMS personnalisée et non la clé par défaut. L'ARN de votre clé personnalisée doit être ajouté en tant que ressource.

L'exemple suivant de politique en ligne ajoute les autorisations requises :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetParameters",
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:ssm:us-east-2:999999999999:parameter/<parameter_name>",
        "arn:aws:secretsmanager:us-east-2:999999999999:secret:<secret_name>",
        "arn:aws:kms:us-east-2:999999999999:key/<key_id>"
      ]
    }
  ]
}
```

Injectez des données sensibles en tant que variable d'environnement

Dans votre définition de conteneur, spécifiez `secrets` avec le nom de la variable d'environnement à définir dans le conteneur et l'ARN complet du paramètre Systems Manager Parameter Store contenant les données sensibles à présenter au conteneur.

Ce qui suit est un extrait d'une définition de tâche indiquant le format lors du référencement d'un paramètre du magasin de paramètres de Systems Manager. Si le paramètre Systems Manager Parameter Store existe dans la même région que la tâche que vous lancez, vous pouvez utiliser l'ARN complet ou le nom du paramètre. Si le paramètre existe dans une autre région, l'ARN complet doit être spécifié.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
    }]
  }]
}
```

Injecter des données sensibles dans une configuration de journal

Dans votre définition de conteneur, lorsque vous spécifiez `logConfiguration`, vous pouvez spécifier `secretOptions` avec le nom de l'option du pilote de journal à définir dans le conteneur et l'ARN complet du paramètre Systems Manager Parameter Store contenant les données sensibles à présenter au conteneur.

Important

Si le paramètre Systems Manager Parameter Store existe dans la même région que la tâche que vous lancez, vous pouvez utiliser l'ARN complet ou le nom du paramètre. Si le paramètre existe dans une autre région, l'ARN complet doit être spécifié.

Ce qui suit est un extrait d'une définition de tâche indiquant le format lors du référencement d'un paramètre du magasin de paramètres de Systems Manager.

```
{
  "containerProperties": [{
```



```
"logConfiguration": [{
  "logDriver": "fluentd",
  "options": {
    "tag": "fluentd demo"
  },
  "secretOptions": [{
    "name": "fluentd-address",
    "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
  }]
}]
}]
}
```

Création d'un AWS Systems Manager paramètre Parameter Store

Vous pouvez utiliser la AWS Systems Manager console pour créer un paramètre Systems Manager Parameter Store pour vos données sensibles. Pour plus d'informations, veuillez consulter la rubrique [Procédure : Créer et utiliser un paramètre dans une commande \(console\)](#) dans le Guide de l'utilisateur AWS Systems Manager .

Pour créer un paramètre Parameter Store

1. Ouvrez la AWS Systems Manager console à l'adresse <https://console.aws.amazon.com/systems-manager/>.
2. Dans le panneau de navigation, choisissez Parameter Store, Create Parameter (Créer un paramètre).
3. Dans le champ Name (Nom), saisissez une hiérarchie et un nom de paramètre. Par exemple, saisissez test/database_password.
4. Pour Description, saisissez une description facultative.
5. Pour Type, choisissez String StringList, ou SecureString.

Note

- Si vous le souhaitez SecureString, le champ KMS Key ID apparaît. Si vous ne fournissez pas d'ID de clé KMS, un ARN de clé KMS, un nom d'alias ou un ARN d'alias, le système utilise alias/aws/ssm. Il s'agit de la clé KMS par défaut de Systems Manager. Pour éviter d'utiliser cette clé, choisissez une clé personnalisée. Pour plus d'informations, veuillez consulter la rubrique [Utilisation de paramètres de chaîne sécurisée](#) dans le Guide de l'utilisateur AWS Systems Manager .

- Lorsque vous créez un paramètre de chaîne sécurisée dans la console à l'aide du paramètre `key-id` avec un nom d'alias de clé KMS personnalisée ou un ARN d'alias, vous devez spécifier le préfixe `alias/` avant l'alias. Voici un exemple d'ARN :

```
arn:aws:kms:us-east-2:123456789012:alias/MyAliasName
```

Voici un exemple de nom d'alias :

```
alias/MyAliasName
```

6. Pour Value (Valeur), tapez une valeur. Par exemple, `MyFirstParameter`. Si vous le souhaitez `SecureString`, la valeur est masquée exactement telle que vous l'avez saisie.
7. Sélectionnez `Create parameter` (Créer un paramètre).

Authentification par registre privé pour les tâches

L'authentification du registre privé pour les tâches utilisées vous AWS Secrets Manager permet de stocker vos informations d'identification en toute sécurité, puis de les référencer dans votre définition de tâche. Cela permet de référencer des images de conteneurs qui existent dans des registres privés en dehors de ceux AWS qui nécessitent une authentification dans vos définitions de tâches. Cette fonctionnalité est prise en charge par les jobs hébergés sur les EC2 instances Amazon et Fargate.

Important

Si votre définition de tâche fait référence à une image stockée dans Amazon ECR, cette rubrique ne s'applique pas. Pour plus d'informations, consultez [Utilisation d'images Amazon ECR avec Amazon ECS](#) dans le Guide de l'utilisateur Amazon Elastic Container Registry.

Pour les tâches hébergées sur EC2 des instances Amazon, cette fonctionnalité nécessite une version 1.19.0 ou une version ultérieure de l'agent de conteneur. Toutefois, nous vous recommandons d'utiliser la dernière version de l'agent de conteneur. Pour plus d'informations sur la façon de vérifier la version de votre agent et de passer à la dernière version, consultez la section [Mise à jour de l'agent de conteneur Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.

Pour les jobs hébergés sur Fargate, cette fonctionnalité nécessite une version de plateforme 1.2.0 ou une version ultérieure. Pour plus d'informations, consultez les [AWS versions de la plateforme Fargate Linux](#) dans le guide du développeur Amazon Elastic Container Service.

Dans votre définition du conteneur, spécifiez l'objet `repositoryCredentials` avec les détails du secret que vous avez créé. Le secret auquel vous faites référence peut provenir d'un compte différent Région AWS ou différent de celui de la tâche qui l'utilise.

Note

Lorsque vous utilisez l' AWS Batch API AWS CLI, ou AWS SDK, si le secret existe dans la même tâche Région AWS que la tâche que vous lancez, vous pouvez utiliser l'ARN complet ou le nom du secret. Si le secret existe dans un autre compte, l'ARN complet du secret doit être spécifié. Lorsque vous utilisez le AWS Management Console, l'ARN complet du secret doit toujours être spécifié.

Voici un extrait de définition de tâche qui indique les paramètres requis :

```
"containerProperties": [  
  {  
    "image": "private-repo/private-image",  
    "repositoryCredentials": {  
      "credentialsParameter":  
        "arn:aws:secretsmanager:region:123456789012:secret:secret_name"  
    }  
  }  
]
```

Autorisations IAM requises pour l'authentification de registre privé

Le rôle d'exécution est requis pour utiliser cette fonctionnalité. Cela permet à l'agent de conteneur d'extraire l'image de conteneur. Pour de plus amples informations, veuillez consulter [AWS Batch Rôle d'exécution IAM](#).

Pour donner accès aux secrets que vous créez, ajoutez les autorisations suivantes en tant que politique intégrée au rôle d'exécution. Pour plus d'informations, consultez [Ajout et suppression de politiques IAM](#).

- `secretsmanager:GetSecretValue`

- `kms:Decrypt` : requis uniquement si votre clé utilise une clé KMS personnalisée et non la clé par défaut. L'Amazon Resource Name (ARN) de votre clé personnalisée doit être ajouté en tant que ressource.

Voici un exemple de stratégie en ligne qui ajoute les autorisations.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret_name",
        "arn:aws:kms:us-east-1:123456789012:key/key_id"
      ]
    }
  ]
}
```

Tutoriel : Création d'un secret pour l'authentification du registre privé

Procédez comme suit pour créer un secret pour vos informations d'identification de registre privées avec AWS Secrets Manager.

Créez un secret de base

1. Ouvrez la AWS Secrets Manager console à l'adresse <https://console.aws.amazon.com/secretsmanager/>.
2. Choisissez Store a new secret (Stocker un nouveau secret).
3. Pour Select secret type (Sélectionner un type de secret), choisissez Other type of secrets (Autre type de secrets).

4. Sélectionner Plaintext (Texte brut) et saisissez vos informations d'identification de registre privé au format suivant :

```
{
  "username" : "privateRegistryUsername",
  "password" : "privateRegistryPassword"
}
```

5. Choisissez Suivant.
6. Pour Secret name (Nom du secret), saisissez un chemin facultatif et un nom, comme **production/MyAwesomeAppSecret** ou **development/TestSecret**, puis choisissez Next (Suivant). Vous pouvez éventuellement ajouter une description pour vous aider à vous souvenir de la finalité de ce secret ultérieurement.

Le nom du secret doit contenir uniquement des lettres ASCII, des chiffres ou les caractères suivants : `/_+=.@-`.

7. (Facultatif) À ce stade, vous pouvez configurer la rotation de votre secret. Pour cette procédure, gardez l'option Disable automatic rotation (Désactiver la rotation automatique) et choisissez Next (Suivant).

Pour obtenir des instructions sur la façon de configurer la rotation sur des secrets nouveaux ou existants, voir [Rotation de vos AWS Secrets Manager secrets](#).

8. Vérifiez vos paramètres, puis choisissez Store secret (Stocker le secret) pour enregistrer tous les éléments que vous avez saisis en tant que nouveau secret dans Secrets Manager.

Enregistrez une définition de tâche et, sous Registre privé, activez l'authentification du registre privé. Ensuite, dans ARN ou nom Secrets Manager, saisissez l'Amazon Resource Name (ARN) du secret. Pour de plus amples informations, veuillez consulter [Autorisations IAM requises pour l'authentification de registre privé](#).

Volumes Amazon EFS

Amazon Elastic File System (Amazon EFS) fournit un stockage de fichiers simple et évolutif à utiliser dans le cadre de vos AWS Batch tâches. Avec Amazon EFS, la capacité de stockage est élastique. Il s'adapte automatiquement au fur et à mesure que vous ajoutez et supprimez des fichiers. Vos applications peuvent disposer de l'espace de stockage qui leur est nécessaire, au moment où elles en ont besoin.

Vous pouvez utiliser les systèmes de fichiers Amazon EFS AWS Batch pour exporter les données des systèmes de fichiers sur votre flotte d'instances de conteneurs. Ainsi, vos tâches ont accès au même espace de stockage permanent. En revanche, vous devez configurer votre AMI de l'instance de conteneur pour monter le système de fichiers Amazon EFS avant le démarrage du démon Docker. En outre, vos définitions de tâches doivent faire référence à des montages de volumes sur l'instance de conteneur pour utiliser le système de fichiers. Les sections suivantes vous aident à commencer à utiliser Amazon EFS avec AWS Batch.

Considérations relatives aux volumes Amazon EFS

Les informations suivantes doivent être prises en compte lors de l'utilisation de volumes Amazon EFS :

- Pour les tâches utilisant EC2 des ressources, la prise en charge du système de fichiers Amazon EFS a été ajoutée en version préliminaire publique avec la version AMI optimisée pour Amazon ECS 20191212 avec la version 1.35.0 de l'agent de conteneur. Cependant, le support du système de fichiers Amazon EFS est entré en disponibilité générale avec la version AMI optimisée pour Amazon ECS 20200319 avec la version 1.38.0 de l'agent de conteneur, qui contenait le point d'accès Amazon EFS et les fonctionnalités d'autorisation IAM. Nous vous recommandons d'utiliser la version optimisée de l'AMI Amazon ECS 20200319 ou une version ultérieure pour tirer parti de ces fonctionnalités. Pour plus d'informations, consultez les [versions d'AMI optimisées pour Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.

Note

Si vous créez votre propre AMI, vous devez utiliser l'agent de conteneur 1.38.0 ou version ultérieure, `ecs-init` version 1.38.0-1 ou ultérieure, et exécuter les commandes suivantes sur votre instance Amazon. EC2 Tout cela est destiné à activer le plug-in de volume Amazon ECS. Les commandes varient selon que vous utilisez Amazon Linux 2 ou Amazon Linux comme image de base.

Amazon Linux 2

```
$ yum install amazon-efs-utils
systemctl enable --now amazon-ecs-volume-plugin
```

Amazon Linux

```
$ yum install amazon-efs-utils
```

```
sudo shutdown -r now
```

- Pour les tâches utilisant les ressources Fargate, la prise en charge du système de fichiers Amazon EFS a été ajoutée lors de l'utilisation de la version 1.4.0 ou ultérieure de la plateforme. Pour plus d'informations, consultez les versions AWS de la [plateforme Fargate](#) dans le guide du développeur Amazon Elastic Container Service.
- Lorsque vous spécifiez des volumes Amazon EFS dans des tâches utilisant des ressources Fargate, Fargate crée un conteneur de supervision chargé de gérer le volume Amazon EFS. Le conteneur du superviseur utilise une petite partie de la mémoire de la tâche. Le conteneur de supervision est visible lors de l'interrogation du point de terminaison de métadonnées de tâche version 4. Pour de plus amples informations, veuillez consulter [Point de terminaison des métadonnées de tâches version 4](#) dans le Guide de l'utilisateur Amazon Elastic Container Service pour AWS Fargate.

Utiliser les points d'accès Amazon EFS

Les points d'accès Amazon EFS sont des points d'entrée spécifiques à une application dans un système de fichiers EFS qui vous aident à gérer l'accès des applications aux ensembles de données partagés. Pour de plus amples informations sur les points d'accès Amazon EFS et sur la façon de les contrôler, veuillez consulter [Utilisation des points d'accès Amazon EFS](#) dans le Guide de l'utilisateur Amazon Elastic File System.

Les points d'accès peuvent appliquer de manière forcée une identité d'utilisateur, y compris les groupes POSIX de l'utilisateur, pour toutes les demandes de système de fichiers effectuées via le point d'accès. Les points d'accès peuvent également appliquer de manière forcée un répertoire racine différent pour le système de fichiers afin que les clients puissent uniquement accéder aux données stockées dans le répertoire spécifié ou dans les sous-répertoires.

Note

Lors de la création d'un point d'accès EFS, vous spécifiez un chemin d'accès sur le système de fichiers qui servira de répertoire racine. Lorsque vous référencez le système de fichiers EFS avec un ID de point d'accès dans votre définition de tâche AWS Batch, le répertoire racine doit être omis ou défini sur /. Cela applique le chemin défini sur le point d'accès EFS.

Vous pouvez utiliser un rôle IAM de AWS Batch travail pour garantir que des applications spécifiques utilisent un point d'accès spécifique. En combinant des politiques IAM avec des points d'accès, vous pouvez facilement fournir un accès sécurisé à des ensembles de données spécifiques pour vos applications. Cette fonctionnalité utilise les rôles IAM d'Amazon ECS pour la fonctionnalité des tâches. Pour plus d'informations, consultez [Rôles IAM pour les tâches](#) dans le Guide du développeur Amazon Elastic Container Service.

Spécifiez un système de fichiers Amazon EFS dans votre définition de tâche

Pour utiliser les volumes du système de fichiers Amazon EFS pour vos conteneurs, vous devez spécifier les configurations de volume et de point de montage dans votre définition de tâche. L'extrait de code JSON de définition de tâche suivant montre la syntaxe des `mountPoints` objets volumes et d'un conteneur :

```
{
  "containerProperties": [
    {
      "image": "amazonlinux:2",
      "command": [
        "ls",
        "-la",
        "/mount/efs"
      ],
      "mountPoints": [
        {
          "sourceVolume": "myEfsVolume",
          "containerPath": "/mount/efs",
          "readOnly": true
        }
      ],
      "volumes": [
        {
          "name": "myEfsVolume",
          "efsVolumeConfiguration": {
            "fileSystemId": "fs-12345678",
            "rootDirectory": "/path/to/my/data",
            "transitEncryption": "ENABLED",
            "transitEncryptionPort": integer,
            "authorizationConfig": {
              "accessPointId": "fsap-1234567890abcdef1",
```



```
    "iam": "ENABLED"
  }
}
}
```

efsVolumeConfiguration

Type : objet

Obligatoire : non

Ce paramètre est spécifié lorsque vous utilisez des volumes Amazon EFS.

fileSystemId

Type : String

Obligatoire : oui

ID du système de fichiers Amazon EFS à utiliser.

rootDirectory

Type : chaîne

Obligatoire : non

Répertoire du système de fichiers Amazon EFS à monter en tant que répertoire racine à l'intérieur de l'hôte. Si ce paramètre est omis, la racine du volume Amazon EFS est utilisée. La spécification de / a le même effet que l'omission de ce paramètre. Il peut comporter jusqu'à 4 096 caractères.

Important

Si un point d'accès EFS est spécifié dans `leauthorizationConfig`, le paramètre du répertoire racine doit être omis ou défini sur `/`. Cela applique le chemin défini sur le point d'accès EFS.

transitEncryption

Type : String

Valeurs valides : ENABLED | DISABLED

Obligatoire : non

Détermine s'il faut activer le chiffrement pour les données Amazon EFS en transit entre l' AWS Batch hôte et le serveur Amazon EFS. Le chiffrement en transit doit être activé si l'autorisation Amazon EFS IAM est utilisée. Si ce paramètre est omis, la valeur par défaut DISABLED est utilisée. Pour plus d'informations, consultez [Chiffrement des données en transit](#) dans le Guide de l'utilisateur Amazon Elastic File System.

transitEncryptionPort

Type : entier

Obligatoire : non

Port à utiliser lors de l'envoi de données chiffrées entre l' AWS Batch hôte et le serveur Amazon EFS. Si vous ne spécifiez pas de port de chiffrement en transit, il utilise la stratégie de sélection de port adoptée par l'assistant de montage Amazon EFS. La valeur doit être comprise entre 0 et 65 535. Pour plus d'informations, consultez [Assistant de montage EFS](#) dans le Guide de l'utilisateur Amazon Elastic File System User.

authorizationConfig

Type : objet

Obligatoire : non

Détails de configuration des autorisations pour le système de fichiers Amazon EFS.

accessPointId

Type : chaîne

Obligatoire : non

ID du point d'accès à utiliser. Si un point d'accès est spécifié, la valeur du répertoire racine `efsVolumeConfiguration` doit être omise ou définie sur `/`. Cela applique le chemin défini sur le point d'accès EFS. Si un point d'accès est utilisé, le chiffrement de transit

doit être activé dans `EFSVolumeConfiguration`. Pour plus d'informations, consultez [Utilisation des points d'accès Amazon EFS](#) dans le Guide de l'utilisateur Amazon Elastic File System.

`iam`

Type : String

Valeurs valides : ENABLED | DISABLED

Obligatoire : non

Détermine s'il faut utiliser le rôle IAM de AWS Batch tâche défini dans une définition de tâche lors du montage du système de fichiers Amazon EFS. Si cette option est activée, le chiffrement en transit doit être activé dans la configuration `EFSVolumeConfiguration`. Si ce paramètre est omis, la valeur par défaut DISABLED est utilisée. Pour plus d'informations sur le rôle d'exécution IAM, consultez [AWS Batch Rôle d'exécution IAM](#).

Exemples de définitions de tâches

Les exemples de définition de tâches présentés dans les rubriques suivantes montrent comment utiliser des modèles courants tels que les variables d'environnement, la substitution de paramètres et les montages de volume.

Table des matières

- [Variables d'environnement](#)
- [Substitution de paramètres](#)
- [Tester les fonctionnalités du GPU](#)
- [Tâche parallèle sur plusieurs nœuds](#)

Variables d'environnement

L'exemple de définition de tâche suivant utilise des variables d'environnement pour spécifier un type de fichier et une URL Amazon S3. Cet exemple particulier est tiré du billet de blog [Creating a Simple « Fetch & Run » AWS Batch Job](#) compute. Le `fetch_and_run.sh` script décrit dans le billet de blog utilise ces variables d'environnement pour télécharger le `myjob.sh` script depuis S3 et déclarer son type de fichier.

Même si les variables de commande et d'environnement sont codées en dur dans la définition de tâche dans cet exemple, vous pouvez spécifier des remplacements de commande et de variable d'environnement pour rendre la définition de tâche plus polyvalente.

```
{
  "jobDefinitionName": "fetch_and_run",
  "type": "container",
  "containerProperties": {
    "image": "123456789012.dkr.ecr.us-east-1.amazonaws.com/fetch_and_run",
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "2000"
      },
      {
        "type": "VCPU",
        "value": "2"
      }
    ],
    "command": [
      "myjob.sh",
      "60"
    ],
    "jobRoleArn": "arn:aws:iam::123456789012:role/AWSBatchS3ReadOnly",
    "environment": [
      {
        "name": "BATCH_FILE_S3_URL",
        "value": "s3://amzn-s3-demo-source-bucket/myjob.sh"
      },
      {
        "name": "BATCH_FILE_TYPE",
        "value": "script"
      }
    ],
    "user": "nobody"
  }
}
```

Substitution de paramètres

L'exemple de définition de tâche suivant illustre l'utilisation de la substitution de paramètres et la définition de valeurs par défaut.

Les déclarations `Ref::` de la section `command` sont utilisées afin de définir des espaces réservés pour la substitution de paramètres. Lorsque vous soumettez une tâche avec cette définition de tâche, vous spécifiez les paramètres de remplacement pour renseigner ces valeurs, par exemple `inputfile` et `outputfile`. La `parameters` section qui suit définit une valeur par défaut pour `codec`, mais vous pouvez remplacer ce paramètre si nécessaire.

Pour de plus amples informations, veuillez consulter [Paramètres](#).

```
{
  "jobDefinitionName": "ffmpeg_parameters",
  "type": "container",
  "parameters": {"codec": "mp4"},
  "containerProperties": {
    "image": "my_repo/ffmpeg",
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "2000"
      },
      {
        "type": "VCPU",
        "value": "2"
      }
    ],
    "command": [
      "ffmpeg",
      "-i",
      "Ref::inputfile",
      "-c",
      "Ref::codec",
      "-o",
      "Ref::outputfile"
    ],
    "jobRoleArn": "arn:aws:iam::123456789012:role/ECSTask-S3FullAccess",
    "user": "nobody"
  }
}
```

Tester les fonctionnalités du GPU

L'exemple suivant de définition de tâche vérifie si l'AMI de la charge de travail GPU décrit dans [Utiliser une AMI de charge de travail du GPU](#) est configurée correctement. Cet exemple de définition de tâche exécute l'[exemple](#) de classificateur MNIST TensorFlow approfondi de GitHub

```
{
  "containerProperties": {
    "image": "tensorflow/tensorflow:1.8.0-devel-gpu",
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "32000"
      },
      {
        "type": "VCPU",
        "value": "8"
      }
    ],
    "command": [
      "sh",
      "-c",
      "cd /tensorflow/tensorflow/examples/tutorials/mnist; python mnist_deep.py"
    ]
  },
  "type": "container",
  "jobDefinitionName": "tensorflow_mnist_deep"
}
```

Vous pouvez créer un fichier avec le texte JSON précédent appelé, `tensorflow_mnist_deep.json` puis enregistrer une définition de AWS Batch tâche à l'aide de la commande suivante :

```
aws batch register-job-definition --cli-input-json file://tensorflow_mnist_deep.json
```

Tâche parallèle sur plusieurs nœuds

L'exemple de définition de tâche suivant illustre une tâche parallèle à plusieurs nœuds. Pour plus d'informations, consultez la section [Création d'un flux de travail de dynamique moléculaire étroitement couplé avec des tâches parallèles à nœuds multiples AWS Batch dans le blog AWS Compute](#).

```
{
  "jobDefinitionName": "gromacs-jobdef",
  "jobDefinitionArn": "arn:aws:batch:us-east-2:123456789012:job-definition/gromacs-
jobdef:1",
  "revision": 6,
  "status": "ACTIVE",
  "type": "multinode",
  "parameters": {},
  "nodeProperties": {
    "numNodes": 2,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes": "0:1",
        "container": {
          "image": "123456789012.dkr.ecr.us-east-2.amazonaws.com/gromacs_mpi:latest",
          "resourceRequirements": [
            {
              "type": "MEMORY",
              "value": "24000"
            },
            {
              "type": "VCPU",
              "value": "8"
            }
          ],
          "command": [],
          "jobRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
          "ulimits": [],
          "instanceType": "p3.2xlarge"
        }
      }
    ]
  }
}
```

Tâches

Les emplois sont l'unité de travail créée par AWS Batch. Les tâches peuvent être invoquées sous forme d'applications conteneurisées qui s'exécutent sur des instances de conteneur Amazon ECS dans un cluster ECS.

Les tâches de conteneurs peuvent référencer une image, une commande et des paramètres de conteneur. Pour de plus amples informations, veuillez consulter [JobDefinition](#).

Vous pouvez soumettre un grand nombre de tâches simples indépendantes.

Rubriques

- [Tutoriel : soumettre une offre d'emploi](#)
- [Emplois de service à AWS Batch](#)
- [États de l'emploi](#)
- [AWS Batch variables d'environnement de travail](#)
- [Nouvelles tentatives de travail automatisées](#)
- [Dépendances professionnelles](#)
- [Expiration des délais d'exécution des tâches](#)
- [Offres d'emploi Amazon EKS](#)
- [Tâches parallèles sur plusieurs nœuds](#)
- [Tâches parallèles à nœuds multiples sur Amazon EKS](#)
- [Tâches liées à](#)
- [Exécuter des tâches GPU](#)
- [Afficher les AWS Batch tâches dans une file d'attente](#)
- [Rechercher AWS Batch des offres d'emploi dans une file d'attente](#)
- [Modes de mise en réseau pour les AWS Batch tâches](#)
- [Afficher les journaux des AWS Batch tâches dans CloudWatch Logs](#)
- [Consulter les informations relatives au AWS Batch poste](#)

Tutoriel : soumettre une offre d'emploi

Après avoir enregistré une définition de tâche, vous pouvez la soumettre en tant que tâche à une file d'attente de AWS Batch tâches. Vous pouvez remplacer de nombreux paramètres spécifiés dans la définition de la tâche lors de l'exécution.

Pour soumettre une tâche

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation, sélectionnez le Région AWS à utiliser.
3. Dans le volet de navigation, sélectionnez Tâches.
4. Choisissez Soumettre un nouveau travail.
5. Dans Nom, entrez un nom unique pour votre définition de tâche. Le nom peut comporter jusqu'à 128 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
6. Pour Définition du travail, choisissez une définition de tâche existante pour votre tâche. Pour de plus amples informations, veuillez consulter [Création d'une définition de tâche à nœud unique](#).
7. Pour la file d'attente de tâches, choisissez une file d'attente de tâches existante. Pour de plus amples informations, veuillez consulter [Création d'une file d'attente de tâches](#).
8. Pour les dépendances des tâches, sélectionnez Ajouter des dépendances des tâches.
 - Dans le champ Job id, entrez l'ID du job pour toutes les dépendances. Choisissez ensuite Ajouter des dépendances de tâches. Une tâche peut comporter jusqu'à 20 dépendances. Pour de plus amples informations, veuillez consulter [Dépendances professionnelles](#).
9. (Tâches de tableau uniquement) Pour Taille du tableau, spécifiez une taille de tableau entre 2 et 10 000.
10. (Facultatif) Développez les balises, puis choisissez Ajouter une balise pour ajouter des balises à la ressource. Entrez une clé et une valeur facultative, puis choisissez Ajouter une étiquette.
11. Choisissez Page suivante.
12. Dans la section Job overrides :
 - a. (Facultatif) Pour Priorité de planification, entrez une valeur de priorité de planification comprise entre 0 et 100. Les valeurs les plus élevées reçoivent une priorité plus élevée.
 - b. (Facultatif) Pour les tentatives de tâche, entrez le nombre maximal de AWS Batch tentatives de transfert de la tâche vers un RUNNABLE statut. Vous pouvez saisir un nombre compris

entre 1 et 10. Pour de plus amples informations, veuillez consulter [Nouvelles tentatives de travail automatisées](#).

- c. (Facultatif) Pour le délai d'exécution, entrez la valeur du délai d'expiration (en secondes). Le délai d'exécution est le délai avant la fin d'une tâche inachevée. Si une tentative dépasse le délai imparti, elle est arrêtée et passe à un FAILED statut. Pour de plus amples informations, veuillez consulter [Expiration des délais d'exécution des tâches](#). La valeur minimale est 60 secondes.

 Important

Ne comptez pas sur les tâches exécutées sur les ressources de Fargate pour qu'elles s'exécutent pendant plus de 14 jours. Après 14 jours, il est possible que les ressources de Fargate ne soient plus disponibles et que le travail soit probablement résilié.

- d. (Facultatif) Activez les balises de propagation pour propager les balises de la tâche et de la définition de la tâche vers la tâche Amazon ECS.

13. Développez Additional configuration (Configuration supplémentaire).

14. (Facultatif) Pour les conditions de la stratégie Réessayer, choisissez Ajouter une évaluation à la sortie. Entrez au moins une valeur de paramètre, puis choisissez une action. Pour chaque ensemble de conditions, l'action doit être définie sur Réessayer ou sur Quitter. Ces actions signifient ce qui suit :

- Réessayer — AWS Batch Réessaie jusqu'à ce que le nombre de tentatives de travail que vous avez spécifié soit atteint.
- Quitter : AWS Batch arrête de réessayer la tâche.

 Important

Si vous choisissez Ajouter une évaluation à la sortie, configurez au moins un paramètre et choisissez soit une action, soit Supprimer l'évaluation à la sortie.

15. Pour Paramètres, choisissez Ajouter des paramètres pour ajouter des espaces réservés pour la substitution de paramètres. Entrez ensuite une clé et une valeur facultative.

16. Dans la section Dérogations de conteneurs :

- a. Pour Command, entrez les commandes dans le champ en tant qu'équivalent dans un tableau de chaînes JSON.

Ce paramètre correspond à Cmd dans la section [Create a container \(Création d'un conteneur\)](#) de [Docker Remote API \(API distante Docker\)](#) et au paramètre COMMAND de [docker run](#). Pour plus d'informations sur le CMD paramètre Docker, consultez <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note

Ce paramètre ne peut pas contenir de chaîne vide.

- b. Pour v CPUs, entrez le nombre de v CPUs à réserver pour le conteneur. Ce paramètre correspond à CpuShares dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option --cpu-shares correspond à [docker run](#). Chaque processeur virtuel est équivalente à 1 024 parts de processeur. Vous devez spécifier au moins un vCPU.
- c. Pour Mémoire, entrez la limite de mémoire disponible pour le conteneur. Si votre conteneur tente de dépasser la mémoire spécifiée ici, il est arrêté. Ce paramètre correspond à Memory dans la section [Create a container](#) (Créer un conteneur) de l'[API Docker à distance](#) et l'option --memory correspond à [docker run](#). Vous devez spécifier au moins 4 Mio de mémoire pour une tâche.

 Note

Pour optimiser l'utilisation de vos ressources, donnez la priorité à la mémoire pour les tâches d'un type d'instance spécifique. Pour de plus amples informations, veuillez consulter [Gestion de la mémoire des ressources informatiques](#).

- d. (Facultatif) Dans Nombre de GPUs, choisissez le nombre de GPUs à réserver pour le conteneur.
- e. (Facultatif) Pour les variables d'environnement, choisissez Ajouter une variable d'environnement pour ajouter des variables d'environnement sous forme de paires nom-valeur. Ces variables sont transmises au conteneur.
- f. Choisissez Page suivante.

- g. Pour la révision du Job, passez en revue les étapes de configuration. Si vous devez apporter des modifications, choisissez Modifier. Lorsque vous avez terminé, choisissez Créer une définition de tâche.

Emplois de service à AWS Batch

AWS Batch les tâches de service vous permettent de soumettre des demandes aux AWS services par le biais de files AWS Batch d'attente de travail. Actuellement, AWS Batch soutient les emplois SageMaker de formation en tant qu'emplois de service. Contrairement aux tâches conteneurisées AWS Batch qui gèrent l'exécution du conteneur sous-jacent, les tâches de service permettent de AWS Batch fournir des fonctionnalités de planification des tâches et de mise en file d'attente tandis que le AWS service cible (tel que l' SageMaker IA) gère l'exécution réelle des tâches.

AWS Batch for SageMaker Training jobs permet aux data scientists de soumettre des tâches de formation prioritaires à des files d'attente configurables, garantissant ainsi l'exécution des charges de travail sans intervention dès que les ressources sont disponibles. Cette fonctionnalité permet de relever des défis courants tels que la coordination des ressources, la prévention des dépenses excessives accidentelles, le respect des contraintes budgétaires, l'optimisation des coûts grâce aux instances réservées et l'élimination du besoin de coordination manuelle entre les membres de l'équipe.

Les tâches de service diffèrent des tâches conteneurisées de plusieurs manières principales :

- Soumission de tâches : les tâches de service doivent être soumises à l'aide de l'[SubmitServiceJob](#) API. Les tâches de service ne peuvent pas être soumises via la AWS Batch console.
- Exécution des tâches : AWS Batch planifie et met en file d'attente les tâches de service, mais le AWS service cible exécute la charge de travail réelle des tâches.
- Identifiants de ressources : les tâches de service ARNs contiennent « tâche de service » au lieu de « tâche » pour les distinguer des tâches conteneurisées.

Pour commencer à travailler sur les emplois AWS Batch de service pour SageMaker la formation, voir [the section called “Commencer à utiliser AWS Batch l' SageMaker IA”](#).

Rubriques

- [Charges utiles des tâches de maintenance dans AWS Batch](#)

- [Soumettre une offre d'emploi dans le secteur des services AWS Batch](#)
- [Mappage AWS Batch de l'état de la tâche du service avec SageMaker le statut de](#)
- [Stratégies de nouvelle tentative d'emploi dans AWS Batch](#)
- [Surveiller les tâches de service dans une AWS Batch file d'attente](#)
- [Mettre fin à des tâches de service](#)

Charges utiles des tâches de maintenance dans AWS Batch

Lorsque vous soumettez des tâches de service à l'aide de [SubmitServiceJob](#), vous fournissez deux paramètres clés qui définissent la tâche : `serviceJobType` et `serviceRequestPayload`.

- `serviceJobType` Spécifie le AWS service qui exécutera la tâche. Pour les tâches de SageMaker formation, cette valeur est `SAGEMAKER_TRAINING`.
- `serviceRequestPayload` Il s'agit d'une chaîne codée en JSON qui contient la demande complète qui serait normalement envoyée directement au service cible. Pour les tâches de SageMaker formation, cette charge utile contient les mêmes paramètres que ceux que vous utiliseriez avec l'[CreateTrainingJob](#) API SageMaker AI.

Pour une liste complète de tous les paramètres disponibles et leurs descriptions, consultez la référence de l'[CreateTrainingJob](#) API SageMaker AI. Tous les paramètres pris en charge par `CreateTrainingJob` peuvent être inclus dans la charge utile de votre tâche de service.

Pour des exemples de configurations de tâches de formation supplémentaires [APIs, consultez la CLI et SDKs](#) le [Guide du développeur d'SageMaker IA](#).

Nous vous recommandons d'utiliser le PySDK pour créer des tâches de service, car PySDK possède des classes d'assistance et des utilitaires. Pour un exemple d'utilisation de PySDK, consultez les [exemples d'SageMaker IA](#) sur GitHub

Exemple de charge utile d'une tâche de service

L'exemple suivant montre une charge utile d'une tâche de service simple pour une tâche de SageMaker formation qui exécute un script de formation « hello world » :

Cette charge utile serait transmise sous forme de chaîne JSON au `serviceRequestPayload` paramètre lors de l'appel `SubmitServiceJob`.

```
{
```

```

"TrainingJobName": "my-simple-training-job",
"RoleArn": "arn:aws:iam::123456789012:role/SageMakerExecutionRole",
"AlgorithmSpecification": {
  "TrainingInputMode": "File",
  "TrainingImage": "763104351884.dkr.ecr.us-west-2.amazonaws.com/pytorch-
training:2.0.0-cpu-py310",
  "ContainerEntrypoint": [
    "echo",
    "hello world"
  ]
},
"ResourceConfig": {
  "InstanceType": "ml.c5.xlarge",
  "InstanceCount": 1,
  "VolumeSizeInGB": 1
},
"OutputDataConfig": {
  "S3OutputPath": "s3://your-output-bucket/output"
},
"StoppingCondition": {
  "MaxRuntimeInSeconds": 30
}
}

```

Soumettre une offre d'emploi dans le secteur des services AWS Batch

Pour soumettre des tâches de service à AWS Batch, vous utilisez l'[SubmitServiceJob](#) API. Vous pouvez soumettre des tâches à l'aide du AWS CLI ou du SDK.

Si vous n'avez pas encore de rôle d'exécution, vous devez en créer un avant de pouvoir soumettre votre tâche de service. Pour créer le rôle d'exécution SageMaker AI, voir [Comment utiliser les rôles d'exécution SageMaker AI](#) dans le [guide SageMaker AI Developer](#).

Flux de travail de soumission des tâches de service

Lorsque vous soumettez une tâche de service, suivez AWS Batch ce flux de travail :

1. AWS Batch reçoit votre [SubmitServiceJob](#) demande et valide les paramètres AWS Batch spécifiques. Le `serviceRequestPayload` est transmis sans validation.
2. La tâche entre dans l'`SUBMITTED` état et est placée dans la file d'attente de tâches spécifiée
3. AWS Batch évalue s'il existe une capacité disponible dans l'environnement de service pour les `RUNNABLE` tâches situées en tête de file d'attente

4. Si la capacité est disponible, le travail est transféré à `AI SCHEDULED` et le travail a été transmis à l' `SageMaker IA`
5. Lorsque la capacité a été acquise et que l' `SageMaker IA` a téléchargé les données de la tâche de service, l'initialisation de la tâche de service commence et la tâche est remplacée par. `STARTING`
6. Lorsque `SageMaker IA` commence à exécuter la tâche, son statut passe à `RUNNING`.
7. Pendant que l' `SageMaker IA` exécute la tâche, `AWS Batch` surveille sa progression et fait correspondre les états du service aux états des `AWS Batch` tâches. Pour plus de détails sur la façon dont les états des tâches de service sont mappés, voir [???](#)
8. Lorsque la tâche de service est terminée, elle est `SUCCEEDED` déplacée vers et toute sortie est prête à être téléchargée.

Conditions préalables

Avant de soumettre une demande de service, assurez-vous d'avoir :

- Environnement de service : environnement de service qui définit les limites de capacité. Pour de plus amples informations, veuillez consulter [Créez un environnement de service dans AWS Batch](#).
- `SageMaker` file d'attente de `SageMaker` tâches : file d'attente de tâches permettant de planifier les tâches. Pour de plus amples informations, veuillez consulter [Créez une file d'attente SageMaker de tâches de formation dans AWS Batch](#).
- Autorisations IAM : autorisations permettant de créer et de gérer des files `AWS Batch` d'attente de travail et des environnements de service. Pour de plus amples informations, veuillez consulter [AWS Batch Politiques, rôles et autorisations IAM](#).

Soumettre une tâche de service à l'aide de la AWS CLI

Voici comment soumettre une tâche de service à l'aide de la AWS CLI :

```
aws batch submit-service-job \
  --job-name "my-sagemaker-training-job" \
  --job-queue "my-sagemaker-job-queue" \
  --service-job-type "SAGEMAKER_TRAINING" \
  --service-request-payload '{"TrainingJobName\\": "\\sagemaker-training-job-example\\", "\\AlgorithmSpecification\\": {"TrainingImage\\": "\\123456789012.dkr.ecr.us-east-1.amazonaws.com/pytorch-inference:1.8.0-cpu-py3\\", "\\TrainingInputMode\\": "\\File\\", "\\ContainerEntrypoint\\": [\\"sleep\\", \\"1\\"]}, "\\RoleArn\\": "\\arn:aws:iam::123456789012:role/SageMakerExecutionRole\\", "\\OutputDataConfig\\":
```

```
{\"S3OutputPath\": \"s3://example-bucket/model-output/\"}, \"ResourceConfig\":
{\"InstanceType\": \"ml.m5.large\", \"InstanceCount\": 1, \"VolumeSizeInGB\": 1}}'
--client-token \"unique-token-12345\"
```

Pour plus d'informations sur les `serviceRequestPayload` paramètres, consultez [the section called “Charges utiles des tâches de maintenance”](#).

Mappage AWS Batch de l'état de la tâche du service avec SageMaker le statut de

Lorsque vous soumettez des tâches à une file d'attente de SageMaker tâches en utilisant [SubmitServiceJob](#), AWS Batch gère le cycle de vie des tâches et mappe [les états des AWS Batch tâches aux états](#) des tâches de SageMaker formation équivalents. Les tâches de service, telles que les tâches de SageMaker formation, suivent un cycle de vie différent de celui des tâches traditionnelles liées aux conteneurs. Bien que les tâches de service partagent la plupart des États avec les tâches de conteneur, elles introduisent l'`SCHEDULED` état et présentent des comportements de nouvelle tentative différents, en particulier lorsqu'il s'agit de gérer les erreurs de capacité insuffisante du service cible.

Le tableau suivant indique l'état de la AWS Batch tâche et le SageMaker statut/ correspondant : `SecondaryStatus`

État du lot	SageMaker État principal de l'IA	SageMaker Statut secondaire de l'IA	Description
SUBMITTED	N/A	N/A	Job soumis à la file d'attente, en attente d'évaluation par le planificateur.
RUNNABLE	N/A	N/A	Job est en file d'attente et prêt à être planifié. Les emplois dans cet état sont démarrés dès que des ressources suffisantes sont disponibles dans l'environnement de service. Les emplois peuvent rester dans cet état indéfiniment lorsque des ressources suffisantes ne sont pas disponibles.

État du lot	SageMaker État principal de l'IA	SageMaker Statut secondaire de l'IA	Description
SCHEDULED	InProgress	Pending	Tâche de service soumise avec succès à l'SageMaker IA
STARTING	InProgress	Downloading	SageMaker Tâche de formation : téléchargement de données et d'images. La capacité des tâches de formation a été acquise et l'initialisation des tâches commence.
RUNNING	InProgress	Training	SageMaker Algorithme d'exécution des tâches de formation
RUNNING	InProgress	Uploading	SageMaker Tâche de formation : téléchargement d'artefacts de sortie une fois la formation terminée
SUCCEEDED	Completed	Completed	SageMaker Tâche de formation terminée avec succès. Le chargement des artefacts de sortie est terminé.
FAILED	Failed	Failed	SageMaker La tâche de formation a rencontré une erreur irrécupérable.
FAILED	Stopped	Stopped	SageMaker L'utilisation de la tâche de formation a été arrêtée manuellement <code>StopTrainingJob</code> .

Stratégies de nouvelle tentative d'emploi dans AWS Batch

Les stratégies de relance des tâches de service permettent AWS Batch de réessayer automatiquement les tâches de service ayant échoué dans des conditions spécifiques.

Les tâches de service peuvent nécessiter plusieurs tentatives pour plusieurs raisons :

- Problèmes de service temporaires : des erreurs de service internes, des ralentissements ou des interruptions temporaires peuvent entraîner l'échec des tâches lors de leur soumission ou de leur exécution.
- Échec de l'initialisation de l'entraînement : les problèmes rencontrés lors du démarrage de la tâche, tels que les problèmes d'extraction d'images ou les erreurs d'initialisation, peuvent être résolus lors d'une nouvelle tentative.

En configurant des stratégies de relance appropriées, vous pouvez améliorer les taux de réussite des tâches et réduire le besoin d'intervention manuelle, en particulier pour les charges de travail de formation de longue durée.

Note

Les tâches de service réessaient automatiquement certains types d'échecs, tels que les erreurs de capacité insuffisante, sans que les tentatives de nouvelle tentative configurées soient épuisées. Votre stratégie de nouvelle tentative gère principalement d'autres types d'échecs, tels que les erreurs d'algorithme ou les problèmes de service.

Configuration des stratégies de nouvelle tentative

Les stratégies de nouvelles tentatives des tâches de service sont configurées à l'aide de [ServiceJobRetryStrategy](#), qui prend en charge à la fois le nombre de tentatives simples et la logique de nouvelles tentatives conditionnelles.

Nouvelle tentative de configuration

La stratégie de nouvelle tentative la plus simple indique le nombre de nouvelles tentatives à effectuer en cas d'échec d'une tâche de service :

```
{
  "retryStrategy": {
    "attempts": 3
  }
}
```

Cette configuration permet de retenter la tâche de service jusqu'à 3 fois en cas d'échec.

 Important

La `attempts` valeur représente le nombre total de fois que la tâche peut être placée dans `RUNNABLE` cet état, y compris la tentative initiale. Une valeur de 3 signifie que la tâche sera tentée une seule fois au départ, puis réessayée jusqu'à 2 fois en cas d'échec.

Réessayez la configuration avec `evaluateOnExit`

Vous pouvez utiliser le `evaluateOnExit` paramètre pour spécifier les conditions dans lesquelles les tâches doivent être réessayées ou autorisées à échouer. Cela est utile lorsque différents types de défaillances nécessitent un traitement différent.

Le `evaluateOnExit` tableau peut contenir jusqu'à 5 stratégies de nouvelle tentative, chacune spécifiant une action (`RETRYonEXIT`) et des conditions basées sur des raisons de statut :

```
{
  "retryStrategy": {
    "attempts": 5,
    "evaluateOnExit": [
      {
        "action": "RETRY",
        "onStatusReason": "Received status from SageMaker: InternalServerError*"
      },
      {
        "action": "EXIT",
        "onStatusReason": "Received status from SageMaker: ValidationException*"
      },
      {
        "action": "EXIT",
        "onStatusReason": "*"
      }
    ]
  }
}
```

Cette configuration :

- Réessaie les tâches qui échouent en raison d'erreurs internes au serveur SageMaker AI
- Échoue immédiatement les tâches qui rencontrent des exceptions de validation (erreurs du client qui ne seront pas résolues par une nouvelle tentative)

- Inclut une règle fourre-tout permettant de remédier à tout autre type de défaillance

Correspondance entre le statut et le motif

Le `onStatusReason` paramètre prend en charge la correspondance de modèles comportant jusqu'à 512 caractères. Les modèles peuvent utiliser des caractères génériques (*) et correspondre aux raisons de statut renvoyées par l' SageMaker IA.

Pour les tâches de service, les messages d'état provenant de l' SageMaker IA sont précédés du préfixe « Statut reçu de SageMaker : » pour les distinguer des messages AWS Batch générés. Les modèles courants incluent :

- `Received status from SageMaker: InternalServerError*`- Corrigez les erreurs de service internes
- `Received status from SageMaker: ValidationException*`- Correspond aux erreurs de validation du client
- `Received status from SageMaker: ResourceLimitExceeded*`- Correspond aux erreurs de limite de ressources
- `*CapacityError*`- Correspondre aux défaillances liées à la capacité

Tip

Utilisez des modèles de correspondance spécifiques pour gérer les différents types d'erreur de manière appropriée. Par exemple, réessayez des erreurs internes au serveur, mais échouez immédiatement en cas d'erreur de validation indiquant des problèmes liés aux paramètres de la tâche.

Surveiller les tâches de service dans une AWS Batch file d'attente

Vous pouvez suivre le statut des tâches de votre file d'attente de SageMaker formation à l'aide `delist-service-jobs`, et `get-job-queue-snapshot`.

Consultez les tâches en cours dans votre file d'attente :

```
aws batch list-service-jobs \  
  --job-queue my-sm-training-fifo-jq \  
  --
```

```
--job-status RUNNING
```

Consultez les offres d'emploi en attente :

```
aws batch list-service-jobs \  
  --job-queue my-sm-training-fifo-jq \  
  --job-status RUNNABLE
```

Afficher les tâches qui ont été soumises SageMaker mais qui ne sont pas encore en cours d'exécution :

```
aws batch list-service-jobs \  
  --job-queue my-sm-training-fifo-jq \  
  --job-status SCHEDULED
```

Obtenez un aperçu des offres d'emploi figurant en tête de liste :

```
aws batch get-job-queue-snapshot --job-queue my-sm-training-fifo-jq
```

Cette commande indique l'ordre des tâches de service à venir dans votre file d'attente.

Obtenez des informations détaillées sur les tâches de service

Utilisez cette [DescribeServiceJob](#) opération pour obtenir des informations complètes sur une tâche de service spécifique, notamment son statut actuel, les identifiants des ressources de service et des informations détaillées sur les tentatives.

Afficher des informations détaillées sur une tâche spécifique :

```
aws batch describe-service-job \  
  --job-id a4d6c728-8ee8-4c65-8e2a-9a5e8f4b7c3d
```

Cette commande renvoie des informations complètes sur la tâche, notamment :

- Job ARN et statut actuel
- Identifiants de ressources de service (tels que l'ARN de SageMaker la tâche de formation)
- Planification de la priorité et configuration des nouvelles tentatives
- Charge utile de la demande de service contenant les paramètres de service d'origine
- Informations détaillées sur les tentatives avec heures de début et d'arrêt

- Messages d'état provenant du service cible

Surveiller les offres SageMaker de formation

Lorsque vous SageMaker surveillez les tâches de formation AWS Batch, vous pouvez accéder à la fois aux informations relatives aux AWS Batch tâches et aux détails des tâches de SageMaker formation sous-jacentes.

L'identifiant de ressource de service indiqué dans les détails de la tâche contient l'ARN de la tâche de SageMaker formation :

```
{
  "latestAttempt": {
    "serviceResourceId": {
      "name": "TrainingJobArn",
      "value": "arn:aws:sagemaker:us-east-1:123456789012:training-job/my-training-job"
    }
  }
}
```

Vous pouvez utiliser cet ARN pour obtenir des informations supplémentaires directement auprès de SageMaker :

```
aws sagemaker describe-training-job \
  --training-job-name my-training-job
```

Surveillez l'avancement des tâches en vérifiant à la fois AWS Batch le statut et le statut des tâches de SageMaker formation. Le statut du AWS Batch poste indique le cycle de vie global du travail, tandis que le statut du poste de SageMaker formation fournit des informations spécifiques au service sur le processus de formation.

Mettre fin à des tâches de service

Utilisez cette [TerminateServiceJob](#) opération pour arrêter une tâche de service en cours d'exécution.

Mettre fin à une tâche de service spécifique :

```
aws batch terminate-service-job \
  --job-id a4d6c728-8ee8-4c65-8e2a-9a5e8f4b7c3d \
```

```
--reason "Job terminated by user request"
```

Lorsque vous mettez fin à une tâche de service, AWS Batch arrête la tâche et avertit le service cible. Pour les emplois de SageMaker formation, cela mettra également fin à l'emploi de formation en SageMaker IA.

États de l'emploi

Lorsque vous soumettez une tâche à une file d'attente de AWS Batch tâches, la tâche entre dans l'`SUBMITTED` état. Il passe ensuite par les états suivants jusqu'à ce qu'il fonctionne (sortie avec le code 0) ou échoue (sortie avec un autre code que zéro). Les tâches AWS Batch peuvent présenter les états suivants :

SUBMITTED

Une tâche soumise à la file d'attente et qui n'a pas encore été évaluée par le planificateur. Le planificateur évalue la tâche pour déterminer si elle présente des dépendances en attente sur la réussite de l'exécution de toutes les autres tâches. Si des dépendances sont présentes, la tâche est déplacée vers `PENDING`. Si aucune dépendance n'est présente, la tâche est déplacée vers `RUNNABLE`.

PENDING

Une tâche qui se trouve dans la file d'attente et qui n'est pas encore en mesure de s'exécuter en raison d'une dépendance à l'égard d'une autre tâche ou d'une autre ressource. Une fois les dépendances satisfaites, la tâche est déplacée vers `RUNNABLE`.

Note

Les parents des tâches matricielles sont informés `PENDING` lorsqu'une tâche enfant est mise à jour `RUNNABLE` et conservent `PENDING` leur statut pendant l'exécution des tâches enfants. Pour consulter ces emplois, filtrez par `PENDING` statut jusqu'à ce que tous les emplois secondaires atteignent l'état terminal.

RUNNABLE

Une tâche qui se trouve dans la file d'attente, n'a pas de dépendances en attente, et est donc prête pour la planification sur un hôte. Les tâches dans cet état sont démarrées dès que des

ressources suffisantes sont disponibles dans l'un des environnements informatiques mappés à la file d'attente des tâches. Cependant, elles peuvent rester dans cet état indéfiniment lorsque les ressources suffisantes sont indisponibles.

 Note

Si vos tâches ne progressent pas vers `STARTING`, consultez la section [Offres d'emploi bloquées dans un `RUNNABLE` statut](#) de résolution des problèmes.

STARTING

Ces tâches ont été planifiées pour un hôte et les opérations de lancement de conteneurs correspondantes sont en cours. Une fois l'image de conteneur extraite et le conteneur lancé, la tâche prend l'état `RUNNING`.

La durée d'extraction de l'image, la durée d'exécution d'Amazon EKS InitContainer et la durée de résolution d'Amazon ECS ContainerDependency se produisent à l'état `STARTING`. Le temps nécessaire pour extraire une image pour votre tâche est équivalent à la durée pendant laquelle votre tâche sera à l'état de départ.

Par exemple, s'il faut trois minutes pour extraire l'image correspondant à votre tâche, celle-ci restera en état de démarrage pendant trois minutes. Si l'exécution d'InitContainers prend au total dix minutes, votre tâche Amazon EKS sera démarrée pendant dix minutes. Si votre tâche Amazon ECS ContainerDependencies est définie, celle-ci sera en cours de démarrage jusqu'à ce que toutes les dépendances des conteneurs (leur temps d'exécution) soient résolues. Le démarrage n'est pas inclus dans les délais ; la durée commence à courir. Pour plus d'informations, consultez la section [États des tâches](#).

RUNNING

La tâche s'exécute en tant que tâche de conteneur sur une instance de conteneur Amazon ECS au sein d'un environnement informatique. Lorsque le conteneur de tâche se ferme, le code de sortie du processus détermine si la tâche a réussi ou échoué. Un code de sortie 0 indique la réussite et tout code de sortie autre que zéro indique qu'il s'agit d'un échec. Si la tâche associée à une tentative échouée présente des tentatives restantes dans sa configuration de stratégie de nouvelle tentative facultative, la tâche est déplacée de nouveau vers `RUNNABLE`. Pour de plus amples informations, veuillez consulter [Nouvelles tentatives de travail automatisées](#).

 Note

Les journaux des RUNNING tâches sont disponibles dans CloudWatch Logs. Le groupe de journaux est `/aws/batch/job`, et le format du nom du flux de journaux est le suivant : `first200CharsOfJobDefinitionName/default/ecs_task_id`. Ce format est susceptible de changer dans le futur.

Une fois qu'une tâche a atteint le RUNNING statut, vous pouvez récupérer par programmation le nom de son flux de journal à l'aide de l'opération [DescribeJobs](#) API.

Pour plus d'informations, consultez la section [Afficher les données de journal envoyées aux CloudWatch journaux](#) dans le guide de l'utilisateur Amazon CloudWatch Logs.

Par défaut, ces journaux n'expirent jamais. Vous pouvez toutefois modifier la durée de conservation. Pour plus d'informations, consultez la section [Conservation des données du journal des modifications dans CloudWatch les journaux](#) du guide de l'utilisateur Amazon CloudWatch Logs.

SUCCEEDED

La tâche a réussi avec un code de sortie de 0. L'état des SUCCEEDED offres d'emploi est conservé AWS Batch pendant au moins 7 jours.

 Note

Les journaux des SUCCEEDED tâches sont disponibles dans CloudWatch Logs. Le groupe de journaux est `/aws/batch/job`, et le format du nom du flux de journaux est le suivant : `first200CharsOfJobDefinitionName/default/ecs_task_id`. Ce format est susceptible de changer dans le futur.

Une fois qu'une tâche a atteint le RUNNING statut, vous pouvez récupérer par programmation le nom de son flux de journal à l'aide de l'opération [DescribeJobs](#) API.

Pour plus d'informations, consultez la section [Afficher les données de journal envoyées aux CloudWatch journaux](#) dans le guide de l'utilisateur Amazon CloudWatch Logs.

Par défaut, ces journaux n'expirent jamais. Vous pouvez toutefois modifier la durée de conservation. Pour plus d'informations, consultez la section [Conservation des données du journal des modifications dans CloudWatch les journaux](#) du guide de l'utilisateur Amazon CloudWatch Logs.

FAILED

La tâche a échoué dans toutes les tentatives disponibles. L'état des FAILED offres d'emploi est conservé AWS Batch pendant au moins 7 jours.

Note

Les journaux des FAILED tâches sont disponibles dans CloudWatch Logs. Le groupe de journaux est `/aws/batch/job`, et le format du nom du flux de journaux est le suivant : `first200CharsOfJobDefinitionName/default/ecs_task_id`. Ce format est susceptible de changer dans le futur.

Une fois qu'une tâche a atteint le RUNNING statut, vous pouvez récupérer son flux de journal par programmation à l'aide de l'opération [DescribeJobsAPI](#). Pour plus d'informations, consultez la section [Afficher les données de journal envoyées aux CloudWatch journaux](#) dans le guide de l'utilisateur Amazon CloudWatch Logs. Par défaut, ces journaux n'expirent jamais. Vous pouvez toutefois modifier la durée de conservation. Pour plus d'informations, consultez la section [Conservation des données du journal des modifications dans CloudWatch les journaux](#) du guide de l'utilisateur Amazon CloudWatch Logs.

AWS Batch variables d'environnement de travail

AWS Batch définit des variables d'environnement spécifiques dans les tâches de conteneur. Ces variables d'environnement fournissent une introspection pour les conteneurs contenus dans les tâches. Vous pouvez utiliser les valeurs de ces variables dans la logique de vos applications. Toutes les variables AWS Batch définies commencent par le `AWS_BATCH_` préfixe. Il s'agit d'un préfixe de variable d'environnement protégé. Vous ne pouvez pas utiliser ce préfixe pour vos propres variables dans les définitions de tâches ou les remplacements.

Les variables d'environnement suivantes sont disponibles dans les conteneurs de tâche :

AWS_BATCH_CE_NAME

Cette variable est définie sur le nom de l'environnement informatique dans lequel votre tâche est placée.

AWS_BATCH_JOB_ARRAY_INDEX

Cette variable est uniquement définie dans les tâches de tableau enfants. L'index de tâche de tableau commence à 0. Chaque tâche enfant reçoit un numéro d'index unique. Par exemple, une tâche de tableau avec 10 enfants a les valeurs d'index de 0 à 9. Vous pouvez utiliser cette valeur d'index pour contrôler la façon dont les enfants de votre tâche de tableau sont différenciés. Pour de plus amples informations, veuillez consulter [Utilisez l'index des tâches matriciel pour contrôler la différenciation des tâches](#).

AWS_BATCH_JOB_ARRAY_SIZE

Cette variable est définie sur la taille de la tâche de tableau parent. La taille de la tâche de matrice parent est transmise à la tâche de matrice enfant dans cette variable.

AWS_BATCH_JOB_ATTEMPT

Cette variable est définie sur le numéro de tentative de tâche. La première tentative est numérotée 1. Pour de plus amples informations, veuillez consulter [Nouvelles tentatives de travail automatisées](#).

AWS_BATCH_JOB_ID

Cette variable est définie sur l'ID de la AWS Batch tâche.

AWS_BATCH_JOB_KUBERNETES_NODE_UID

Cette variable est définie comme l'KubernetesUID de l'objet nœud qui se trouve dans le cluster Kubernetes sur lequel le pod s'exécute. Cette variable n'est définie que pour les tâches exécutées sur les ressources Amazon EKS. Pour plus d'informations, consultez [UIDs](#) la Kubernetesdocumentation.

AWS_BATCH_JOB_MAIN_NODE_INDEX

Cette variable est uniquement définie dans les tâches parallèles à plusieurs nœuds. Cette variable est définie sur le numéro d'index du nœud principal de la tâche. Le code de votre application peut le comparer AWS_BATCH_JOB_MAIN_NODE_INDEX à AWS_BATCH_JOB_NODE_INDEX celui d'un nœud individuel pour déterminer s'il s'agit du nœud principal.

AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS

Cette variable n'est définie que dans les nœuds enfants de tâches parallèles à nœuds multiples. Cette variable n'est pas présente sur le nœud principal, mais elle est définie sur l'IPv4 adresse privée du nœud principal de la tâche. Votre code d'application du nœud enfant peut utiliser cette adresse pour communiquer avec le nœud principal.

AWS_BATCH_JOB_NODE_INDEX

Cette variable est uniquement définie dans les tâches parallèles à plusieurs nœuds. Cette variable est définie sur le numéro d'index du nœud du nœud. L'index du nœud commence à 0 et chaque nœud reçoit un numéro d'index unique. Par exemple, une tâche parallèle à plusieurs nœuds avec 10 enfants dispose de valeurs d'index de 0 à 9.

AWS_BATCH_JOB_NUM_NODES

Cette variable est uniquement définie dans les tâches parallèles à plusieurs nœuds. Cette variable est définie sur le nombre de nœuds que vous avez demandés pour votre tâche parallèle à nœuds multiples.

AWS_BATCH_JQ_NAME

Cette variable est définie sur le nom de la file d'attente pour laquelle votre tâche a été soumise.

Nouvelles tentatives de travail automatisées

Vous pouvez appliquer une stratégie de nouvelle tentative à vos tâches et définitions de tâches qui permet à vos tâches d'être automatiquement retentées si elles échouent. Les scénarios de défaillance possibles sont les suivants :

- Tout code de sortie autre que zéro à partir d'un conteneur de tâche
- Défaillance ou arrêt de l'instance Amazon EC2
- Erreur ou panne de AWS service interne

Lorsqu'une tâche est soumise à une file d'attente de tâches et placée dans l'`RUNNING` état considéré comme une tentative. Par défaut, chaque tâche a une tentative de passer à l'état `SUCCEEDED` ou `FAILED`. Cependant, les flux de travail de définition et de soumission des tâches peuvent être utilisés pour spécifier une stratégie de nouvelle tentative comprenant entre 1 et 10 tentatives. S'il [evaluateOnExit](#) est spécifié, il peut contenir jusqu'à 5 stratégies de nouvelle tentative. Si cette [evaluateOnExit](#) option est spécifiée, mais qu'aucune des stratégies de nouvelle tentative ne correspond, la tâche est réessayée. Pour les tâches qui ne correspondent pas à la sortie, ajoutez une entrée finale qui se ferme pour une raison ou une autre. Par exemple, cet `evaluateOnExit` objet possède deux entrées contenant des actions de `RETRY` et une entrée finale contenant une action de `EXIT`.

```
"evaluateOnExit": [
```

```
{
  "action": "RETRY",
  "onReason": "AGENT"
},
{
  "action": "RETRY",
  "onStatusReason": "Task failed to start"
},
{
  "action": "EXIT",
  "onReason": "*"
}
]
```

Lors de l'exécution, la variable d'environnement `AWS_BATCH_JOB_ATTEMPT` est définie sur le numéro de tentative de tâche correspondant du conteneur. La première tentative est numérotée 1 et les tentatives suivantes sont classées par ordre croissant (par exemple, 2, 3, 4).

Supposons, par exemple, qu'une tentative de travail échoue pour une raison quelconque et que le nombre de tentatives spécifié dans la configuration des nouvelles tentatives soit supérieur à ce `AWS_BATCH_JOB_ATTEMPT` nombre. Ensuite, la tâche est replacée dans son `RUNNABLE` état actuel. Pour de plus amples informations, veuillez consulter [États de l'emploi](#).

Note

Les tâches annulées ou résiliées ne sont pas réessayées. De même, les tâches qui échouent en raison d'une définition de tâche non valide ne sont pas réessayées.

Pour plus d'informations, reportez-vous aux [Stratégie de nouvelle tentative](#) sections [Création d'une définition de tâche à nœud unique](#), [Tutoriel : soumettre une offre d'emploi](#) et [Codes d'erreur des tâches arrêtées](#).

Dépendances professionnelles

Lorsque vous soumettez une AWS Batch tâche, vous pouvez spécifier la tâche IDs dont elle dépend. Dans ce cas, le AWS Batch planificateur veille à ce que votre tâche ne soit exécutée qu'une fois les dépendances spécifiées terminées avec succès. Une fois effectuée, la tâche dépendante passe de `PENDING` à `RUNNABLE` puis `STARTING` et `RUNNING`. Si l'une des dépendances de tâche échoue, la tâche dépendante passera automatiquement de `PENDING` à `FAILED`.

Par exemple, une Tâche A peut exprimer une dépendance à plus de 20 autres tâches devant être effectuées avant qu'elle ne puisse s'exécuter. Vous pouvez ensuite envoyer des tâches supplémentaires qui dépendent de la Tâche A et jusqu'à 19 autres tâches.

Pour les tâches de tableau, vous pouvez spécifier un type de dépendance SEQUENTIAL sans spécifier d'ID de tâche, afin que chaque tâche de tableau enfant se termine de façon séquentielle à partir de l'index 0. Vous pouvez également spécifier une dépendance de type N_TO_N avec un ID de tâche. Ainsi, chaque enfant de l'index de cette tâche doit attendre que l'enfant de l'index correspondant de chaque dépendance soit terminé avant de pouvoir commencer. Pour de plus amples informations, veuillez consulter [Tâches liées à](#).

Pour soumettre une AWS Batch tâche comportant des dépendances, consultez [Tutoriel : soumettre une offre d'emploi](#).

[Planification axée sur les ressources](#) vous permet de planifier des tâches en fonction des ressources consommables nécessaires à leur exécution. Vous spécifiez les ressources consommables nécessaires à l'exécution d'une tâche, et Batch prend en compte ces dépendances de ressources lorsqu'il planifie une tâche. Vous pouvez réduire la sous-utilisation des ressources informatiques en n'allouant que les tâches disposant de toutes les ressources nécessaires. La planification basée sur les ressources est disponible pour les politiques de planification FIFO et fair-share et peut être utilisée avec toutes les plateformes de calcul prises en charge par Batch, notamment EKS, ECS et Fargate. Il peut être utilisé avec des tâches Array, des tâches MNP (Multi-Node parallel) et avec des tâches Batch classiques.

Expiration des délais d'exécution des tâches

Vous pouvez configurer un délai d'expiration pour vos tâches de sorte que si une tâche s'exécute plus longtemps que ce délai, AWS Batch arrête la tâche. Par exemple, vous pouvez avoir une tâche dont vous savez qu'elle ne devrait prendre que 15 minutes. Parfois, votre application est bloquée dans une boucle et s'exécute indéfiniment. Ainsi, vous pouvez définir un délai d'expiration de 30 minutes pour mettre fin à la tâche bloquée.

Important

Par défaut, il n'y a pas de délai d'expiration de la tâche. Si vous ne définissez pas de délai d'expiration de la tâche, celle-ci s'exécute jusqu'à ce que le conteneur sorte.

Vous spécifiez un paramètre `attemptDurationSeconds` qui doit être au moins égal à 60 secondes, que ce soit dans votre définition de tâche ou lorsque vous soumettez la tâche. Lorsque ce nombre de secondes s'est écoulé après l'`startedAt` du démarrage de la tentative de tâche, AWS Batch met fin à la tâche. Sur la ressource de calcul, le conteneur de tâches reçoit un signal `SIGTERM` pour donner à votre application la possibilité de se fermer de façon appropriée. Si le conteneur est encore en cours d'exécution après 30 secondes, un signal `SIGKILL` est envoyé pour forcer la fermeture du conteneur.

Les arrêts dus à des délais d'expiration sont traités au mieux. Vous ne devez pas vous attendre à ce que votre délai d'attente expire exactement au moment où la tentative de travail expire (cela peut prendre quelques secondes de plus). Si votre application a besoin d'une exécution de délai d'expiration précise, vous devez implémenter cette logique au sein de l'application. Si un grand nombre de tâches arrivent à expiration simultanément, les arrêts dus à des délais d'expiration sont traités comme une file d'attente premier entré, premier sorti, où les tâches sont arrêtées par lots.

 Note

Il n'y a pas de délai d'expiration maximal pour une AWS Batch tâche.

Si une tâche est interrompue pour avoir dépassé le délai imparti, elle n'est pas réessayée. Si une tentative de tâche échoue toute seule, elle peut être relancée si les nouvelles tentatives sont activées, et le décompte du délai d'expiration repart à zéro pour la nouvelle tentative.

 Important

Les tâches exécutées sur les ressources Fargate ne peuvent pas s'attendre à être exécutées pendant plus de 14 jours. Si le délai d'expiration dépasse 14 jours, les ressources de Fargate risquent de ne plus être disponibles et le travail sera interrompu.

Pour les tâches de tableau, les tâches enfants ont la même configuration de délai d'expiration que la tâche parente.

Pour plus d'informations sur la soumission d'une AWS Batch tâche avec une configuration de délai d'expiration, consultez [Tutoriel : soumettre une offre d'emploi](#).

Offres d'emploi Amazon EKS

Un travail est la plus petite unité de travail dans laquelle il s'agit AWS Batch. Une AWS Batch tâche sur Amazon EKS est one-to-one mappée vers un Kubernetes pod. Une définition de AWS Batch tâche est un modèle de AWS Batch tâche. Lorsque vous soumettez une AWS Batch tâche, vous faites référence à une définition de tâche, vous ciblez une file d'attente de tâches et vous donnez un nom à une tâche. Dans la définition d'une AWS Batch tâche sur Amazon EKS, le paramètre [EKSProperties](#) définit l'ensemble de paramètres pris en charge par une tâche AWS Batch sur Amazon EKS. Dans une [SubmitJob](#) demande, le [eksPropertiesOverride](#) paramètre permet de remplacer certains paramètres courants. Ainsi, vous pouvez utiliser des modèles de définitions de tâches pour plusieurs tâches. Lorsqu'une tâche est envoyée à votre cluster Amazon EKS, AWS Batch elle est transformée en tâche podspec (Kind: Pod). podspec utilise des AWS Batch paramètres supplémentaires pour garantir que les tâches sont correctement dimensionnées et planifiées. AWS Batch combine des étiquettes et des nuances pour garantir que les tâches ne s'exécutent que sur les nœuds AWS Batch gérés et que les autres pods ne s'exécutent pas sur ces nœuds.

Important

- Si le `hostNetwork` paramètre n'est pas défini explicitement dans une définition de tâche Amazon EKS, le mode réseau du pod est AWS Batch défini par défaut sur le mode hôte. Plus précisément, les paramètres suivants sont appliqués : `hostNetwork=true` et `dnsPolicy=ClusterFirstWithHostNet`.
- AWS Batch nettoie les modules de travail peu après qu'un module ait terminé sa tâche. Pour consulter les journaux des applications pod, configurez un service de journalisation pour votre cluster. Pour de plus amples informations, veuillez consulter [Utiliser CloudWatch les journaux pour surveiller AWS Batch les tâches Amazon EKS](#).

Rubriques

- [Tutoriel : mapper une tâche en cours d'exécution à un pod et à un nœud](#)
- [Tutoriel : Associer un module en cours d'exécution à sa tâche](#)

Tutoriel : mapper une tâche en cours d'exécution à un pod et à un nœud

Les nodeName paramètres podProperties d'une tâche en cours d'exécution ont été podName définis pour la tentative de tâche en cours. Utilisez l'opération [DescribeJobs](#) API pour afficher ces paramètres.

Voici un exemple de sortie.

```
$ aws batch describe-jobs --job 2d044787-c663-4ce6-a6fe-f2baf7e51b04
{
  "jobs": [
    {
      "status": "RUNNING",
      "jobArn": "arn:aws:batch:us-east-1:123456789012:job/2d044787-c663-4ce6-a6fe-f2baf7e51b04",
      "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/MyJobOnEks_SleepWithRequestsOnly:1",
      "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/My-Eks-JQ1",
      "jobId": "2d044787-c663-4ce6-a6fe-f2baf7e51b04",
      "eksProperties": {
        "podProperties": {
          "nodeName": "ip-192-168-55-175.ec2.internal",
          "containers": [
            {
              "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
              "resources": {
                "requests": {
                  "cpu": "1",
                  "memory": "1024Mi"
                }
              }
            }
          ]
        },
        "podName": "aws-batch.b0aca953-ba8f-3791-83e2-ed13af39428c"
      }
    }
  ]
}
```

Pour une tâche pour laquelle les nouvelles tentatives sont activées, la podName fin nodeName de chaque tentative terminée figure dans le paramètre de eksAttempts liste de l'opération

d'[DescribeJobs](#) API. La podName fin nodeName de la tentative d'exécution en cours se trouve dans l'podPropertiesobjet.

Tutoriel : Associer un module en cours d'exécution à sa tâche

Un pod possède des étiquettes qui indiquent le jobId et uuid de l'environnement informatique auquel il appartient. AWS Batch injecte des variables d'environnement afin que l'exécution de la tâche puisse référencer les informations de la tâche. Pour de plus amples informations, veuillez consulter [AWS Batch variables d'environnement de travail](#). Vous pouvez consulter ces informations en exécutant la commande suivante. La sortie est la suivante.

```
$ kubectl describe pod aws-batch.14638eb9-d218-372d-ba5c-1c9ab9c7f2a1 -n my-aws-batch-namespace
Name:          aws-batch.14638eb9-d218-372d-ba5c-1c9ab9c7f2a1
Namespace:     my-aws-batch-namespace
Priority:       0
Node:          ip-192-168-45-88.ec2.internal/192.168.45.88
Start Time:    Wed, 26 Oct 2022 00:30:48 +0000
Labels:        batch.amazonaws.com/compute-environment-uuid=5c19160b-d450-31c9-8454-86cf5b30548f
               batch.amazonaws.com/job-id=f980f2cf-6309-4c77-a2b2-d83fbba0e9f0
               batch.amazonaws.com/node-uid=a4be5c1d-9881-4524-b967-587789094647
...
Status:        Running
IP:            192.168.45.88
IPs:
  IP: 192.168.45.88
Containers:
  default:
    Image:      public.ecr.aws/amazonlinux/amazonlinux:2
    ...
  Environment:
    AWS_BATCH_JOB_KUBERNETES_NODE_UID:  a4be5c1d-9881-4524-b967-587789094647
    AWS_BATCH_JOB_ID:                   f980f2cf-6309-4c77-a2b2-d83fbba0e9f0
    AWS_BATCH_JQ_NAME:                  My-Eks-JQ1
    AWS_BATCH_JOB_ATTEMPT:              1
    AWS_BATCH_CE_NAME:                  My-Eks-CE1
...

```

Fonctionnalités prises en charge par AWS Batch Amazon EKS Jobs

Voici les fonctionnalités AWS Batch spécifiques qui sont également communes aux Kubernetes tâches exécutées sur Amazon EKS :

- [Dépendances professionnelles](#)
- [Tâches liées à](#)
- [Expiration des délais d'exécution des tâches](#)
- [Nouvelles tentatives de travail automatisées](#)
- [Utilisez la planification équitable pour faciliter la planification des tâches](#)

Kubernetes **Secrets** et **ServiceAccounts**

AWS Batch prend en charge le référencement Kubernetes Secrets et ServiceAccounts. Vous pouvez configurer des pods pour utiliser les rôles IAM Amazon EKS pour les comptes de service. Pour plus d'informations, consultez [la section Configuration des pods pour utiliser un compte Kubernetes de service](#) dans le [guide de l'utilisateur Amazon EKS](#).

Documents connexes

- [Considérations relatives à la mémoire et au vCPU pour AWS Batch Amazon EKS](#)
- [Exécuter des tâches GPU](#)
- [Offres d'emploi bloquées dans un RUNNABLE statut](#)

Tâches parallèles sur plusieurs nœuds

Vous pouvez utiliser des tâches parallèles à nœuds multiples pour exécuter des tâches uniques couvrant plusieurs instances Amazon EC2. Avec les tâches parallèles à AWS Batch nœuds multiples (également appelées planification en groupe), vous pouvez exécuter des applications informatiques hautes performances à grande échelle et un entraînement de modèles de GPU distribués sans avoir à lancer, configurer et gérer directement les ressources Amazon EC2. Une tâche parallèle à AWS Batch nœuds multiples est compatible avec n'importe quel framework prenant en charge la communication inter-nœuds basée sur l'IP. Les exemples incluent Apache MXNet TensorFlow, Caffe2 ou MPI (Message Passing Interface).

Les tâches parallèles à nœuds multiples sont soumises en tant que tâche unique. Toutefois, votre définition de tâche (ou les remplacements de nœud de soumission de tâche) spécifie le nombre de

nœuds à créer pour la tâche et les groupes de nœuds à créer. Chaque tâche parallèle à plusieurs nœuds contient un nœud principal qui est lancé en premier. Une fois le nœud principal activé, les nœuds enfants sont lancés et démarrés. La tâche n'est terminée que si le nœud principal sort. Tous les nœuds enfants sont ensuite arrêtés. Pour de plus amples informations, veuillez consulter [Groupes de nœuds](#).

Les nœuds de tâches parallèles à nœuds multiples sont à locataire unique. Cela signifie qu'un seul conteneur de tâches est exécuté sur chaque instance Amazon EC2.

Le statut de la tâche finale (SUCCEEDED ou FAILED) est déterminé par le statut de la tâche finale du nœud principal. Pour connaître le statut d'une tâche parallèle à nœuds multiples, décrivez la tâche à l'aide de l'ID de tâche qui a été renvoyé lorsque vous l'avez soumise. Si vous avez besoin des informations relatives aux nœuds enfants, décrivez chaque nœud enfant individuellement. Vous pouvez adresser les nœuds à l'aide de la `#N` notation (en commençant par 0). Par exemple, pour accéder aux détails du deuxième nœud d'une tâche, décrivez `aws_batch_job_id` #1 à l'aide de l'opération AWS Batch [DescribeJobs](#) API. Les informations `started`, `stoppedAt`, `statusReason` et `exit` concernant une tâche en parallèle à plusieurs nœuds sont renseignées à partir du nœud principal.

Si vous spécifiez des nouvelles tentatives de travail, une défaillance du nœud principal entraîne une nouvelle tentative. Les défaillances du nœud enfant n'entraînent pas d'autres tentatives. Chaque nouvelle tentative d'une tâche parallèle à plusieurs nœuds met à jour la tentative correspondante de ses nœuds enfants associés.

Pour exécuter des tâches parallèles sur plusieurs nœuds AWS Batch, le code de votre application doit contenir les frameworks et les bibliothèques nécessaires à la communication distribuée.

Rubriques

- [Variables d'environnement](#)
- [Groupes de nœuds](#)
- [Cycle de vie des tâches pour les tâches MNP](#)
- [Considérations relatives à l'environnement informatique pour MNP avec AWS Batch](#)

Variables d'environnement

Au moment de l'exécution, chaque nœud est configuré avec les variables d'environnement standard que toutes les AWS Batch tâches reçoivent. En outre, les nœuds sont configurés avec les variables d'environnement suivantes, spécifiques aux tâches parallèles à nœuds multiples :

AWS_BATCH_JOB_MAIN_NODE_INDEX

Cette variable est définie sur le numéro d'index du nœud principal de la tâche. Le code de votre application peut le comparer `AWS_BATCH_JOB_MAIN_NODE_INDEX` à `AWS_BATCH_JOB_NODE_INDEX` celui d'un nœud individuel pour déterminer s'il s'agit du nœud principal.

AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS

Cette variable n'est définie que dans les nœuds enfants de tâches parallèles à nœuds multiples. Cette variable n'est pas présente sur le nœud principal. Cette variable est définie sur l'IPv4 adresse privée du nœud principal de la tâche. Votre code d'application du nœud enfant peut utiliser cette adresse pour communiquer avec le nœud principal.

AWS_BATCH_JOB_NODE_INDEX

Cette variable est définie sur le numéro d'index du nœud du nœud. L'index du nœud commence à 0 et chaque nœud reçoit un numéro d'index unique. Par exemple, une tâche parallèle à plusieurs nœuds avec 10 enfants dispose de valeurs d'index de 0 à 9.

AWS_BATCH_JOB_NUM_NODES

Cette variable est définie sur le nombre de nœuds que vous avez demandés pour votre tâche parallèle à plusieurs nœuds.

Groupes de nœuds

Un groupe de nœuds est un groupe identique de nœuds de travail qui partagent tous les mêmes propriétés de conteneur. Vous pouvez spécifier jusqu' AWS Batch à cinq groupes de nœuds distincts pour chaque tâche.

Chaque groupe peut avoir ses propres images de conteneur, commandes, variables d'environnement, etc. Par exemple, vous pouvez soumettre une tâche qui nécessite une seule `c5.xlarge` instance pour le nœud principal et cinq nœuds enfants d'`c5.xlargeinstance`. Chacun de ces groupes de nœuds distincts peut spécifier des images de conteneur ou des commandes différentes à exécuter pour chaque tâche.

Sinon, tous les nœuds de votre tâche peuvent utiliser un seul groupe de nœuds. De plus, le code de votre application peut différencier les rôles des nœuds tels que le nœud principal et le nœud enfant. Pour ce faire, il compare la variable d'`AWS_BATCH_JOB_MAIN_NODE_INDEX` environnement à sa propre valeur pour `AWS_BATCH_JOB_NODE_INDEX`. Vous pouvez avoir jusqu'à 1 000 nœuds

par tâche. Il s'agit de la limite par défaut pour les instances d'un cluster Amazon ECS. Vous pouvez [demander à augmenter cette limite](#).

 Note

Actuellement, tous les groupes de nœuds d'une tâche parallèle à nœuds multiples doivent utiliser le même type d'instance.

Cycle de vie des tâches pour les tâches MNP

Lorsque vous soumettez une tâche parallèle à nœuds multiples, la tâche passe au SUBMITTED statut. Ensuite, le travail attend la fin de toutes les dépendances du travail. La tâche passe également au RUNNABLE statut. Enfin, AWS Batch fournit la capacité d'instance requise pour exécuter votre tâche et lance ces instances.

Chaque tâche parallèle à plusieurs nœuds contient un nœud principal. Le nœud principal est une sous-tâche unique que AWS Batch surveille afin de déterminer le résultat de la tâche multi-nœuds soumise. Le nœud principal est lancé en premier et il passe au statut STARTING. La valeur du délai d'attente spécifiée dans le `attemptDurationSeconds` paramètre s'applique à l'ensemble de la tâche et non aux nœuds.

Lorsque le nœud principal atteint le RUNNING statut après l'exécution du conteneur du nœud, les nœuds enfants sont lancés et passent également au STARTING statut. Les nœuds enfants s'affichent dans un ordre aléatoire. Il n'existe aucune garantie sur la planification ou l'ordre de lancement des nœuds enfants. Pour garantir que tous les nœuds des tâches sont dans le même RUNNING état après l'exécution du conteneur du nœud, le code de votre application peut interroger l'AWS Batch API pour obtenir des informations sur le nœud principal et le nœud enfant. Le code de l'application peut également attendre que tous les nœuds soient en ligne avant de démarrer une tâche de traitement distribué. L'adresse IP privée du nœud principal est disponible en tant que variable d'environnement `AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS` dans chaque nœud enfant. Votre code d'application peut utiliser ces informations pour se coordonner et communiquer les données entre chaque tâche.

À mesure que les nœuds individuels se terminent, ils se déplacent vers SUCCEEDED ou FAILED selon leur code de sortie. Si le nœud principal se termine, la tâche est considérée comme terminée et tous les nœuds enfants sont arrêtés. Si un nœud enfant meurt, aucune action AWS Batch n'est entreprise sur les autres nœuds de la tâche. Si vous ne souhaitez pas que votre travail se poursuive

avec un nombre réduit de nœuds, vous devez en tenir compte dans le code de votre application. Cette opération met fin à la tâche ou l'annule.

Considérations relatives à l'environnement informatique pour MNP avec AWS Batch

Plusieurs points sont à prendre en compte lors de la configuration des environnements de calcul pour exécuter des tâches parallèles à plusieurs nœuds avec AWS Batch.

- Les tâches parallèles à nœuds multiples ne sont pas prises en charge dans les environnements UNMANAGED informatiques.
- Si vous souhaitez soumettre des tâches parallèles à nœuds multiples à un environnement informatique, créez un groupe de placement de clusters dans une seule zone de disponibilité et associez-le à vos ressources de calcul. Cela permet de maintenir vos tâches parallèles à plusieurs nœuds sur un groupe logique d'instances à proximité avec un potentiel de flux réseau élevé. Pour plus d'informations, veuillez consulter la rubrique [Groupes de placement](#) dans le Guide de l'utilisateur Amazon EC2.
- Les tâches parallèles à nœuds multiples ne sont pas prises en charge dans les environnements informatiques qui utilisent des instances Spot.
- AWS Batch les tâches parallèles à nœuds multiples utilisent le mode awsvpc réseau Amazon ECS, qui confère à vos conteneurs de tâches parallèles à nœuds multiples les mêmes propriétés réseau que les instances Amazon EC2. Chaque conteneur de tâche parallèle à plusieurs nœuds obtient sa propre interface réseau Elastic, une adresse IP privée principale et un nom d'hôte DNS interne. L'interface réseau est créée dans le même sous-réseau VPC que sa ressource de calcul hôte.
- Votre environnement informatique ne peut pas être associé à plus de cinq groupes de sécurité. Les interfaces réseau élastiques créées et associées à une tâche MNP utiliseront les groupes de sécurité spécifiés dans votre environnement informatique. Si vous ne spécifiez aucun groupe de sécurité, le groupe de sécurité par défaut du VPC est utilisé.
- Le mode awsvpc réseau ne fournit pas les interfaces réseau élastiques pour les tâches parallèles à nœuds multiples avec des adresses IP publiques. Pour accéder à Internet, vos ressources de calcul doivent être lancées dans un sous-réseau privé configuré pour utiliser une passerelle NAT. Pour plus d'informations, veuillez consulter [NAT Gateways \(Passerelles NAT\)](#) dans le Guide de l'utilisateur Amazon VPC. Les communications entre nœuds doivent utiliser l'adresse IP privée ou le nom d'hôte DNS du nœud. Les tâches parallèles à nœuds multiples qui s'exécutent sur des ressources informatiques situées dans des sous-réseaux publics ne disposent pas d'un accès

réseau sortant. Pour créer un VPC avec des sous-réseaux privés et une passerelle NAT, consultez [Créer un Virtual Private Cloud](#).

- Les interfaces réseau élastiques créées et associées à vos ressources informatiques ne peuvent pas être détachées manuellement ni modifiées par votre compte. Cela permet d'éviter la suppression accidentelle d'une interface Elastic network associée à une tâche en cours d'exécution. Pour libérer les interfaces réseau Elastic pour une tâche, arrêtez-la.
- Votre environnement informatique doit avoir une valeur maximale de v suffisante CPUs pour prendre en charge votre tâche parallèle sur plusieurs nœuds.
- Votre quota d'instances Amazon EC2 inclut le nombre d'instances nécessaires à l'exécution de votre tâche. Supposons, par exemple, que votre tâche nécessite 30 instances, mais que votre compte ne puisse exécuter que 20 instances dans une région. Ensuite, votre travail restera bloqué dans RUNNABLE son statut.
- Si vous spécifiez un type d'instance pour un groupe de nœuds dans une tâche parallèle à nœuds multiples, votre environnement de calcul doit lancer ce type d'instance.

Tâches parallèles à nœuds multiples sur Amazon EKS

Vous pouvez utiliser AWS Batch Amazon Elastic Kubernetes Service pour exécuter des tâches multi-nœuds parallèles (MNP) (également appelées planification groupée) sur vos clusters gérés. Kubernetes Cette option est couramment utilisée pour les tâches volumineuses, étroitement couplées et à hautes performances qui ne peuvent pas être exécutées sur une seule instance Amazon Elastic Compute Cloud. Pour de plus amples informations, veuillez consulter [Tâches parallèles sur plusieurs nœuds](#).

Vous pouvez utiliser cette fonctionnalité pour exécuter des applications informatiques hautes performances Kubernetes spécifiques gérées par Amazon EKS, pour la formation de grands modèles linguistiques et pour d'autres tâches d'intelligence artificielle (IA) /apprentissage automatique (ML).

Rubriques

- [Exécution de tâches MNP](#)
- [Création d'une définition de tâche Amazon EKS MNP](#)
- [Soumettre une offre d'emploi Amazon EKS MNP](#)
- [Remplacer une définition de tâche Amazon EKS MNP](#)

Exécution de tâches MNP

AWS Batch prend en charge les tâches MNP sur Amazon Elastic Container Service et Amazon EKS à l'aide d'Amazon EC2. Vous trouverez ci-dessous des informations plus détaillées sur les paramètres d'instance et de conteneur de la fonctionnalité.

Quotas d'instance pour MNP sur Amazon EKS

- Jusqu'à 1 000 instances peuvent être utilisées pour une seule tâche MNP.
- Jusqu'à 5 000 instances peuvent rejoindre un seul cluster Amazon EKS.
- Jusqu'à 5 environnements informatiques peuvent être mis en cluster et attachés à une file d'attente de tâches.

Par exemple, vous pouvez faire évoluer jusqu'à 5 environnements informatiques en cluster dans une file d'attente de tâches et 1 000 instances dans chaque environnement informatique.

Outre les paramètres de l'instance, il est important de noter que vous ne pouvez pas utiliser Fargate pour les tâches MNP via l'un ou l'autre des services.

Vous ne pouvez utiliser qu'un seul type d'instance dans chaque tâche MNP. Vous pouvez modifier le type d'instance en mettant à jour l'environnement de calcul ou lorsque vous définissez un nouvel environnement de calcul. Vous pouvez également spécifier le type d'instance et indiquer les exigences en matière de vCPU et de mémoire lors de la création de la définition de tâche.

Quotas de conteneurs pour MNP sur Amazon EKS

- Une tâche parallèle à nœuds multiples prend en charge un pod par nœud.
- Jusqu'à 10 conteneurs (ou 10 conteneurs init). Pour plus d'informations, consultez [Init Containers](#) dans la documentation de Kubernetes.) dans chaque pod.
- Jusqu'à 5 plages de nœuds par tâche MNP.
- Jusqu'à 10 images de conteneur distinctes dans chaque plage de nœuds.

Par exemple, vous pouvez exécuter jusqu'à 10 000 conteneurs dans une seule tâche MNP contenant 5 plages de nœuds et un total de 50 images uniques.

Exécution de tâches MNP dans un Amazon VPC privé et un cluster Amazon EKS

Les tâches MNP peuvent être exécutées sur n'importe quel cluster Amazon EKS, qu'il dispose d'un accès Internet public ou non. Lorsque vous utilisez un cluster Amazon EKS avec un accès réseau privé uniquement, assurez-vous qu'il AWS Batch peut accéder au plan de contrôle Amazon EKS et au serveur Kubernetes d'API géré. Vous pouvez accorder l'accès nécessaire via les points de terminaison Amazon Virtual Private Cloud. Pour plus d'informations, consultez [Configurer un service de point de terminaison](#).

Les pods du cluster Amazon EKS ne peuvent pas télécharger d'image depuis une source publique car le VPC privé n'a pas accès à Internet. Votre cluster Amazon EKS doit extraire des images d'un registre de conteneurs intégré à votre Amazon VPC. Vous pouvez créer un [Amazon Elastic Container Registry \(Amazon ECR\) dans votre Amazon](#) VPC et y copier des images de conteneur pour que vos nœuds puissent y accéder.

Vous pouvez également créer une règle de cache d'extraction avec Amazon ECR. Une fois qu'une règle de cache d'extraction est créée pour un registre public externe, vous pouvez simplement extraire une image de ce registre public externe à l'aide de l'URI de votre registre privé Amazon ECR. Amazon ECR crée ensuite un référentiel et met l'image en cache. Lorsqu'une image mise en cache est extraite à l'aide de l'URI du registre privé Amazon ECR, Amazon ECR vérifie le registre distant pour voir s'il existe une nouvelle version de l'image et met à jour votre registre privé jusqu'à une fois toutes les 24 heures. Pour plus d'informations, consultez [Création d'une règle de cache d'extraction dans Amazon ECR](#).

Notification d'erreur

Si vos offres d'emploi MNP sont bloquées, vous pouvez recevoir des notifications via Amazon AWS Management Console et Amazon EventBridge. Par exemple, si une tâche MNP est bloquée en tête de file d'attente, vous pouvez être averti du problème et recevoir des informations sur sa cause afin que vous puissiez prendre des mesures rapides pour débloquer votre file d'attente de tâches. Vous pouvez éventuellement mettre fin automatiquement à la tâche MNP si aucune action n'est entreprise dans un laps de temps défini dans le modèle de file d'attente des tâches. Pour de plus amples informations, consultez [Événements bloqués dans la file d'offres d'emploi](#).

Création d'une définition de tâche Amazon EKS MNP

Pour définir et exécuter des tâches MNP sur Amazon EKS, de nouveaux paramètres sont intégrés aux opérations [RegisterJobDefinition](#) et à l'[SubmitJobAPI](#).

- Utilisez la [eksPropertiesnodeProperties](#) section ci-dessous pour définir votre définition de tâche MNP.
- À utiliser [eksPropertiesOverride](#) sous la [nodePropertyOverrides](#) section pour remplacer les paramètres définis dans la définition de tâche lors de la soumission d'une tâche MNP.

Ces actions peuvent être définies par le biais d'opérations d'API et du AWS Management Console.

Référence : Enregistrez la charge utile de la demande de définition de tâche Amazon EKS MNP

L'exemple suivant montre comment enregistrer une définition de tâche Amazon EKS MNP avec deux nœuds.

```
{
  "jobDefinitionName": "MyEksMnpJobDefinition",
  "type": "multinode",
  "nodeProperties": {
    "numNodes": 2,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes" : "0:",
        "eksProperties": {
          "podProperties": {
            "containers": [
              {
                "name": "test-eks-container-1",
                "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                "command": [
                  "sleep",
                  "60"
                ],
                "resources": {
                  "limits": {
                    "cpu": "1",
                    "memory": "1024Mi"
                  }
                },
                "securityContext": {
                  "runAsUser": 1000,
                  "runAsGroup": 3000,
```

```
        "privileged":true,
        "readOnlyRootFilesystem":true,
        "runAsNonRoot":true
    }
}
],
"initContainers": [
    {
        "name":"init-ekscontainer",
        "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
        "command": [
            "echo",
            "helloWorld"
        ],
        "resources": {
            "limits": {
                "cpu": "1",
                "memory": "1024Mi"
            }
        }
    }
]
},
"metadata": {
    "labels": {
        "environment" : "test"
    }
}
}
}
}
```

Pour enregistrer la définition de tâche à l'aide de AWS CLI, copiez-la dans un fichier local nommé `MyEksMnpJobDefinition.json` et exécutez la commande suivante.

```
aws batch register-job-definition --cli-input-json file://MyEksMnpJobDefinition.json
```

Vous recevrez la réponse JSON suivante.

```
{
  "jobDefinitionName": "MyEksMnpJobDefinition",
```

```
"jobDefinitionArn": "arn:aws:batch:us-east-1:0123456789:job-definition/
MyEksMnpJobDefinition:1",
  "revision": 1
}
```

Soumettre une offre d'emploi Amazon EKS MNP

Pour soumettre une tâche à l'aide de la définition de tâche enregistrée, entrez la commande suivante. Remplacez la valeur de `<EKS_JOB_QUEUE_NAME>` par le nom ou l'ARN d'une file d'attente de tâches préexistante associée à un environnement informatique Amazon EKS.

```
aws batch submit-job --job-queue <EKS_JOB_QUEUE_NAME> \
  --job-definition MyEksMnpJobDefinition \
  --job-name myFirstEksMnpJob
```

Vous recevrez la réponse JSON suivante.

```
{
  "jobArn": "arn:aws:batch:region:account:job/9b979cce-9da0-446d-90e2-ffa16d52af68",
  "jobName": "myFirstEksMnpJob",
  "jobId": "<JOB_ID>"
}
```

Vous pouvez vérifier l'état de la tâche à l'aide du JoBid renvoyé à l'aide de la commande suivante.

```
aws batch describe-jobs --jobs <JOB_ID>
```

Remplacer une définition de tâche Amazon EKS MNP

Vous pouvez éventuellement remplacer les détails de la définition de la tâche (par exemple, modifier la taille de la tâche MNP ou les détails de la tâche enfant). Vous trouverez ci-dessous un exemple de charge utile de demande JSON pour soumettre une tâche MNP à cinq nœuds, ainsi que les modifications apportées à la commande du `test-eks-container-1` conteneur.

```
{
  "numNodes": 5,
  "nodePropertyOverrides": [
    {
```

```

    "targetNodes": "0:",
    "eksPropertiesOverride": {
      "podProperties": {
        "containers": [
          {
            "name": "test-eks-container-1",
            "command": [
              "sleep",
              "150"
            ]
          }
        ]
      }
    }
  }
}

```

Pour soumettre une tâche avec ces remplacements, enregistrez l'exemple dans un fichier local, `eks-mnp-job-nodeoverride.json`, et utilisez le AWS CLI pour soumettre le travail avec les remplacements.

Tâches liées à

Une tâche matricielle est une tâche qui partage des paramètres communs, tels que la définition de la tâcheCPU, v et la mémoire. Il s'exécute comme un ensemble de tâches de base connexes mais distinctes qui peuvent être distribuées sur plusieurs hôtes et peuvent s'exécuter simultanément. Les tâches matricielles constituent le moyen le plus efficace d'exécuter des tâches extrêmement parallèles, telles que des simulations de Monte Carlo, des balayages paramétriques ou des tâches de rendu de grande envergure.

AWS Batch les tâches matricielles sont soumises comme les tâches normales. Cependant, vous spécifiez une taille de tableau (entre 2 et 10 000) pour définir le nombre de tâches enfants devant s'exécuter dans le tableau. Si vous envoyez une tâche avec une taille de tableau 1 000, une seule tâche s'exécute et génère 1 000 tâches enfants. La tâche de tableau est une référence ou un pointeur pour gérer toutes les tâches enfants. Ainsi, vous pouvez soumettre des charges de travail importantes en une seule requête. Le délai spécifié dans le `attemptDurationSeconds` paramètre s'applique à chaque tâche enfant. La tâche du tableau parent n'a pas de délai d'expiration.

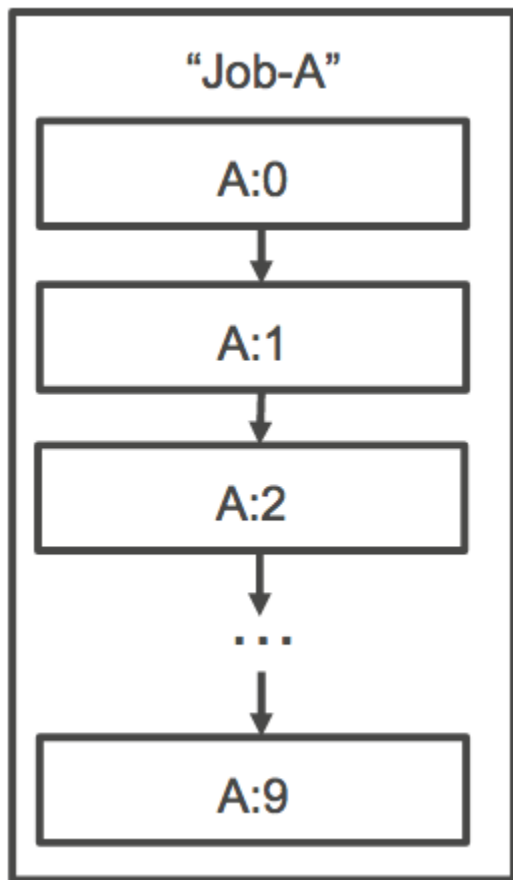
Lorsque vous soumettez une tâche de matrice, la tâche de matrice parent reçoit un ID de AWS Batch tâche normal. Chaque tâche secondaire possède le même identifiant de base. Toutefois, l'index du

tableau pour le travail enfant est ajouté à la fin de l'ID parent, par exemple *exemple_job_ID:0* pour le premier travail enfant du tableau.

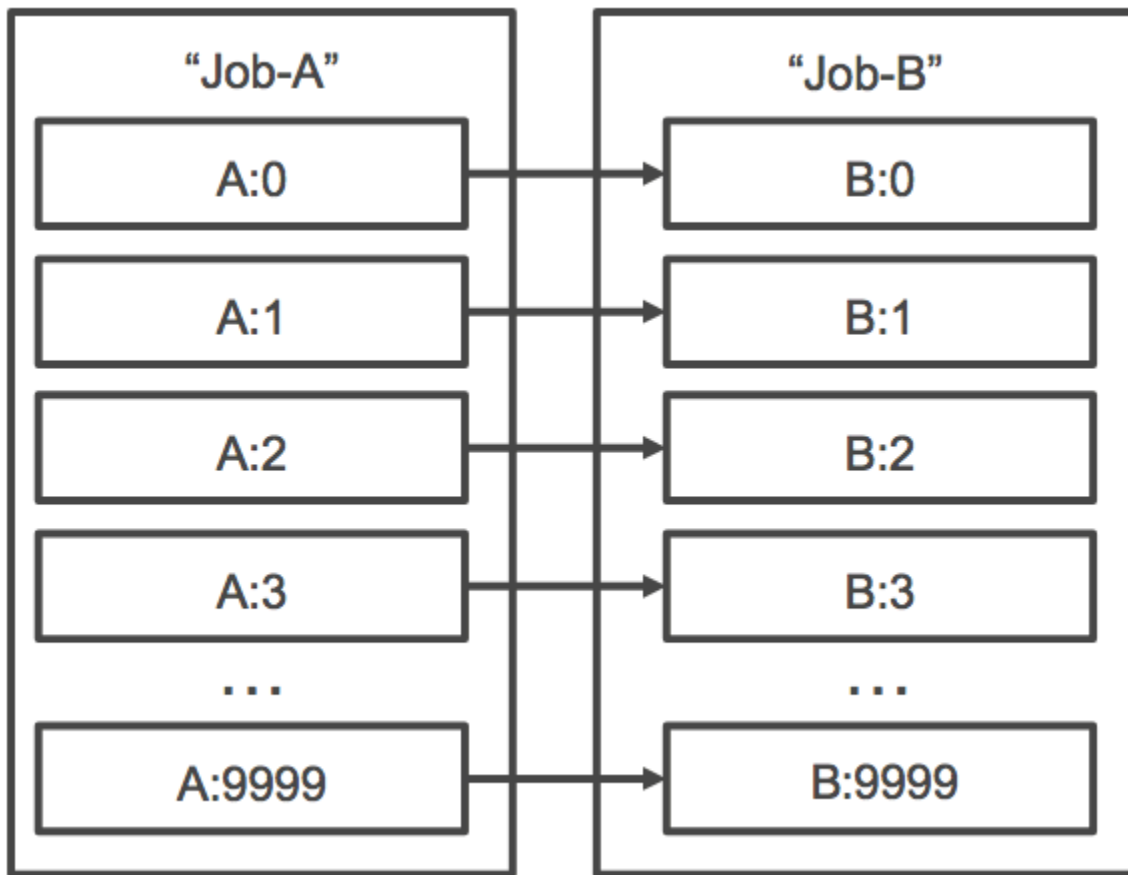
La tâche de matrice parent peut entrer un SUCCEEDED statut SUBMITTED PENDING FAILED,, ou. Une tâche parent de tableau est mise à jour PENDING lorsqu'une tâche enfant est mise à jour vers RUNNABLE. Pour plus d'informations sur les interdépendances entre les emplois, consultez [Dépendances professionnelles](#).

Lors de l'exécution, la variable d'environnement AWS_BATCH_JOB_ARRAY_INDEX est définie sur le numéro d'index de tableau de la tâche correspondant du conteneur. Le premier index des tâches du tableau est numéroté 0, et les tentatives suivantes sont classées par ordre croissant (par exemple, 1, 2 et 3). Vous pouvez utiliser cette valeur d'index pour contrôler la façon dont les enfants de votre tâche de tableau sont différenciés. Pour de plus amples informations, veuillez consulter [Utilisez l'index des tâches matriciel pour contrôler la différenciation des tâches](#).

Pour les dépendances de tâches de tableau, vous pouvez spécifier un type pour une dépendance, comme SEQUENTIAL ou N_TO_N. Vous pouvez spécifier un type de dépendance SEQUENTIAL (sans spécifier d'ID de tâche), afin que chaque tâche de tableau enfant se termine de façon séquentielle à partir de l'index 0. Par exemple, si vous envoyez une tâche de tableau dont la taille de tableau est 100, et que vous spécifiez une dépendance de type SEQUENTIAL, 100 tâches enfants sont générées de façon séquentielle, et la première tâche enfant doit être effectuée avant que la tâche enfant suivante ne commence. Le schéma ci-dessous montre une Tâche A, une tâche de tableau de taille 10. Chaque tâche de l'index enfant de la Tâche A dépend de la tâche enfant précédente. La tâche A :1 ne peut pas commencer tant que la tâche A :0 n'est pas terminée.



Vous pouvez également spécifier une dépendance de type N_TO_N avec un ID de tâche pour les tâches de tableau. Ainsi, chaque enfant de l'index de cette tâche doit attendre que l'enfant de l'index correspondant de chaque dépendance soit terminé avant de pouvoir commencer. La figure suivante montre le Job A et le Job B, deux tâches matricielles d'une taille de tableau de 10 000 chacune. Chaque tâche de l'index enfant de la Tâche B dépend de l'index correspondant de la Tâche A. Tâche B:1 ne peut pas démarrer tant que Tâche A:1 n'est pas terminée.



Si vous annulez ou mettez fin à une tâche de la baie parent, toutes les tâches enfants sont annulées ou terminées en même temps. Vous pouvez annuler ou mettre fin à des emplois individuels pour enfants (ce qui les fait passer à un FAILED statut) sans affecter les autres emplois pour enfants. Toutefois, si une tâche de matrice enfant échoue (seule ou en annulant ou en mettant fin manuellement à la tâche), la tâche parent échoue également. Dans ce scénario, le travail du parent passe à celui FAILED où tous les emplois des enfants sont terminés.

Pour plus d'informations sur la recherche et le filtrage des tâches de matrice, consultez [Rechercher des offres d'emploi dans une file d'attente](#).

Rubriques

- [Exemple de flux de travail sur une matrice](#)
- [Utilisez l'index des tâches matriciel pour contrôler la différenciation des tâches](#)

Exemple de flux de travail sur une matrice

Un flux de travail courant pour les AWS Batch clients consiste à exécuter une tâche de configuration préalable, à exécuter une série de commandes sur un grand nombre de tâches d'entrée, puis à terminer par une tâche qui agrège les résultats et écrit des données récapitulatives sur Amazon S3, DynamoDB, Amazon Redshift ou Aurora.

Par exemple :

- JobA: tâche standard, hors matrice, qui effectue une liste rapide et une validation des métadonnées des objets d'un compartiment Amazon S3, BucketA. La syntaxe [SubmitJob](#)JSON est la suivante.

```
{
  "jobName": "JobA",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobA-list-and-validate:1"
}
```

- JobB: une tâche matricielle de 10 000 copies qui en dépend JobA qui exécute des commandes gourmandes en ressources processeur sur chaque objet BucketA et télécharge les résultats vers. BucketB La syntaxe [SubmitJob](#)JSON est la suivante.

```
{
  "jobName": "JobB",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobB-CPU-Intensive-Processing:1",
  "containerOverrides": {
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "4096"
      },
      {
        "type": "VCPU",
        "value": "32"
      }
    ]
  }
  "arrayProperties": {
    "size": 10000
  },
}
```

```

    "dependsOn": [
      {
        "jobId": "JobA_job_ID"
      }
    ]
  }
}

```

- JobC: une autre tâche de matrice de 10 000 copies qui dépend JobB d'un modèle de N_TO_N dépendance, qui exécute des commandes gourmandes en mémoire sur chaque élément contenu BucketB, écrit des métadonnées dans DynamoDB et télécharge le résultat obtenu vers BucketC La syntaxe [SubmitJobJSON](#) est la suivante.

```

{
  "jobName": "JobC",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobC-Memory-Intensive-Processing:1",
  "containerOverrides": {
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "32768"
      },
      {
        "type": "VCPU",
        "value": "1"
      }
    ]
  }
  "arrayProperties": {
    "size": 10000
  },
  "dependsOn": [
    {
      "jobId": "JobB_job_ID",
      "type": "N_TO_N"
    }
  ]
}

```

- JobD: tâche matricielle qui exécute 10 étapes de validation, chacune devant interroger DynamoDB et susceptible d'interagir avec l'un des compartiments Amazon S3 ci-dessus. Chacune des étapes de la procédure JobD exécute la même commande. Cependant, le comportement est différent

en fonction de la valeur de la variable d'AWS_BATCH_JOB_ARRAY_INDEX environnement dans le conteneur de la tâche. Ces étapes de validation s'exécutent de manière séquentielle (par exemple, JobD:0 puis JobD:1). La syntaxe [SubmitJob](#)JSON est la suivante.

```
{
  "jobName": "JobD",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobD-Sequential-Validation:1",
  "containerOverrides": {
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "32768"
      },
      {
        "type": "VCPU",
        "value": "1"
      }
    ]
  },
  "arrayProperties": {
    "size": 10
  },
  "dependsOn": [
    {
      "jobId": "JobC_job_ID"
    },
    {
      "type": "SEQUENTIAL"
    }
  ]
}
```

- JobE: tâche finale, hors matrice, qui effectue quelques opérations de nettoyage simples et envoie une notification Amazon SNS avec un message indiquant que le pipeline est terminé et un lien vers l'URL de sortie. La syntaxe [SubmitJob](#)JSON est la suivante.

```
{
  "jobName": "JobE",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobE-Cleanup-and-Notification:1",
```

```
"parameters": {
  "SourceBucket": "s3://amzn-s3-demo-source-bucket",
  "Recipient": "pipeline-notifications@mycompany.com"
},
"dependsOn": [
  {
    "jobId": "JobD_job_ID"
  }
]
}
```

Utilisez l'index des tâches matriciel pour contrôler la différenciation des tâches

Ce didacticiel explique comment utiliser la variable d'AWS_BATCH_JOB_ARRAY_INDEX environnement pour différencier les emplois pour enfants. Chaque tâche enfant est affectée à cette variable. L'exemple utilise le numéro d'index de la tâche enfant pour lire une ligne spécifique dans un fichier. Ensuite, il remplace le paramètre associé à ce numéro de ligne par une commande dans le conteneur de la tâche. Le résultat est que vous pouvez avoir plusieurs AWS Batch tâches qui exécutent la même image Docker et les mêmes arguments de commande. Cependant, les résultats sont différents car l'index des tâches du tableau est utilisé comme modificateur.

Dans ce didacticiel, vous créez un fichier texte qui contient toutes les couleurs de l'arc-en-ciel, chacune sur sa propre ligne. Vous créez ensuite un script de point d'entrée pour un conteneur Docker qui convertit l'index en une valeur pouvant être utilisée comme numéro de ligne dans le fichier couleur. L'indice commence à zéro, mais les numéros de ligne commencent à un. Créez un Dockerfile qui copie les fichiers de couleur et d'index sur l'image du conteneur et définit ENTRYPOINT l'image dans le script du point d'entrée. Le Dockerfile et les ressources sont créés à partir d'une image Docker transmise à Amazon ECR. Vous enregistrez ensuite une définition de tâche qui utilise votre nouvelle image de conteneur, vous soumettez une tâche AWS Batch matricielle avec cette définition de tâche et vous visualisez les résultats.

Rubriques

- [Conditions préalables](#)
- [Création d'une image de conteneur](#)
- [Transférez votre image sur Amazon ECR](#)
- [Création et enregistrement d'une définition de tâche](#)

- [Soumettre une tâche AWS Batch matricielle](#)
- [Afficher les journaux des tâches de votre matrice](#)

Conditions préalables

Ce flux de travail didactique comporte les prérequis suivants :

- Un environnement AWS Batch informatique. Pour de plus amples informations, veuillez consulter [Création d'un environnement informatique](#).
- Une file AWS Batch d'attente de tâches et un environnement informatique associé. Pour de plus amples informations, veuillez consulter [Création d'une file d'attente de tâches](#).
- Ils AWS CLI sont installés sur votre système local. Pour plus d'informations, reportez-vous à la section [>Installation du AWS Command Line Interface dans le](#) guide de l'AWS Command Line Interface utilisateur.
- Docker installé sur votre système local. Pour en savoir plus, consultez la section [À propos de Docker CE](#) dans la documentation Docker.

Création d'une image de conteneur

Vous pouvez utiliser le `AWS_BATCH_JOB_ARRAY_INDEX` dans une définition de tâche dans le paramètre de commande. Toutefois, nous vous recommandons de créer une image de conteneur qui utilise plutôt la variable dans un script de point d'entrée. Cette section décrit comment créer une image de conteneur de ce type.

Pour créer votre image de conteneur Docker

1. Créez un répertoire à utiliser en tant qu'espace de travail de votre image Docker et accédez à celui-ci.
2. Créez un fichier nommé `colors.txt` dans le répertoire de votre espace de travail et collez-y le texte suivant.

```
red
orange
yellow
green
blue
indigo
```

```
violet
```

3. Créez un fichier nommé `print-color.sh` dans le répertoire de votre espace de travail et collez-y le texte suivant.

 Note

La variable `LINE` est définie sur le code `AWS_BATCH_JOB_ARRAY_INDEX + 1` car l'index de tableau commence à 0 mais les numéros de ligne commencent à 1. La `COLOR` variable est définie sur la couleur associée à son numéro de ligne. `colors.txt`

```
#!/bin/sh
LINE=$((AWS_BATCH_JOB_ARRAY_INDEX + 1))
COLOR=$(sed -n ${LINE}p /tmp/colors.txt)
echo My favorite color of the rainbow is $COLOR.
```

4. Créez un fichier nommé `Dockerfile` dans le répertoire de votre espace de travail et collez-y le contenu suivant. Ce `Dockerfile` copie les fichiers précédents vers votre conteneur et définit le script de point d'entrée à exécuter au lancement du conteneur.

```
FROM busybox
COPY print-color.sh /tmp/print-color.sh
COPY colors.txt /tmp/colors.txt
RUN chmod +x /tmp/print-color.sh
ENTRYPOINT /tmp/print-color.sh
```

5. Générez votre image Docker.

```
$ docker build -t print-color .
```

6. Testez votre conteneur avec le script suivant. Ce script définit la `AWS_BATCH_JOB_ARRAY_INDEX` variable sur 0 localement, puis l'incrémente pour simuler ce que fait une tâche matricielle avec sept enfants.

```
$ AWS_BATCH_JOB_ARRAY_INDEX=0
while [ $AWS_BATCH_JOB_ARRAY_INDEX -le 6 ]
do
    docker run -e AWS_BATCH_JOB_ARRAY_INDEX=$AWS_BATCH_JOB_ARRAY_INDEX print-color
    AWS_BATCH_JOB_ARRAY_INDEX=$((AWS_BATCH_JOB_ARRAY_INDEX + 1))
done
```

```
done
```

En voici la sortie.

```
My favorite color of the rainbow is red.  
My favorite color of the rainbow is orange.  
My favorite color of the rainbow is yellow.  
My favorite color of the rainbow is green.  
My favorite color of the rainbow is blue.  
My favorite color of the rainbow is indigo.  
My favorite color of the rainbow is violet.
```

Transférez votre image sur Amazon ECR

Maintenant que vous avez créé et testé votre conteneur Docker, transférez-le vers un référentiel d'images. Cet exemple utilise Amazon ECR, mais vous pouvez utiliser un autre registre, tel que DockerHub.

1. Créez un référentiel d'images Amazon ECR pour stocker l'image de votre conteneur. Cet exemple utilise uniquement le AWS CLI, mais vous pouvez également utiliser le AWS Management Console. Pour plus d'informations, consultez la section [Création d'un référentiel](#) dans le guide de l'utilisateur d'Amazon Elastic Container Registry.

```
$ aws ecr create-repository --repository-name print-color
```

2. Marquez votre print-color image avec l'URI de votre référentiel Amazon ECR renvoyé à l'étape précédente.

```
$ docker tag print-color aws_account_id.dkr.ecr.region.amazonaws.com/print-color
```

3. Connectez-vous à votre registre Amazon ECR. Pour plus d'informations, veuillez consulter [Authentification de registre](#) dans le Guide de l'utilisateur Amazon Elastic Container Registry.

```
$ aws ecr get-login-password \  
  --region region | docker login \  
  --username AWS \  
  --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

4. Transférez votre image sur Amazon ECR.


```
$ docker push aws_account_id.dkr.ecr.region.amazonaws.com/print-color
```

Création et enregistrement d'une définition de tâche

Maintenant que votre image Docker se trouve dans un registre d'images, vous pouvez la spécifier dans une définition de AWS Batch tâche. Vous pourrez ensuite l'utiliser ultérieurement pour exécuter une tâche matricielle. Cet exemple utilise uniquement le AWS CLI. Cependant, vous pouvez également utiliser le AWS Management Console. Pour de plus amples informations, veuillez consulter [Création d'une définition de tâche à nœud unique](#).

Pour créer une définition de tâche

1. Créez un fichier nommé `print-color-job-def.json` dans le répertoire de votre espace de travail et collez-y le texte suivant. Remplacez l'URI du référentiel d'images par l'URI de votre image.

```
{
  "jobDefinitionName": "print-color",
  "type": "container",
  "containerProperties": {
    "image": "aws_account_id.dkr.ecr.region.amazonaws.com/print-color",
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "250"
      },
      {
        "type": "VCPU",
        "value": "1"
      }
    ]
  }
}
```

2. Enregistrez la définition de tâche auprès de AWS Batch.

```
$ aws batch register-job-definition --cli-input-json file://print-color-job-def.json
```

Soumettre une tâche AWS Batch matricielle

Après avoir enregistré votre définition de tâche, vous pouvez soumettre une tâche AWS Batch matricielle qui utilise votre nouvelle image de conteneur.

Pour soumettre une tâche AWS Batch matricielle

1. Créez un fichier nommé `print-color-job.json` dans le répertoire de votre espace de travail et collez-y le texte suivant.

Note

Cet exemple utilise la file d'attente des tâches mentionnée dans la [the section called "Conditions préalables"](#) section.

```
{
  "jobName": "print-color",
  "jobQueue": "existing-job-queue",
  "arrayProperties": {
    "size": 7
  },
  "jobDefinition": "print-color"
}
```

2. Soumettez le travail à votre AWS Batch file d'attente. Notez l'ID de tâche renvoyé dans la sortie.

```
$ aws batch submit-job --cli-input-json file://print-color-job.json
```

3. Décrivez l'état de la tâche et attendez que la tâche passe à SUCCEEDED.

Afficher les journaux des tâches de votre matrice

Une fois que votre tâche a atteint le SUCCEEDED statut, vous pouvez consulter les CloudWatch journaux depuis le conteneur de la tâche.

Pour consulter les journaux de votre travail dans CloudWatch Logs

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans le panneau de navigation de gauche, choisissez Tâches.

3. Pour Files d'attente, sélectionnez une file d'attente.
4. Dans la section État, choisissez succès.
5. Pour afficher toutes les tâches enfants de votre tâche de tableau, sélectionnez l'ID de tâche qui a été renvoyé à la section précédente.
6. Pour afficher les journaux du conteneur de la tâche, sélectionnez l'une des tâches enfants et choisissez Afficher les journaux.

Filter events	
Time (UTC +00:00)	Message
2018-07-13	
No older events found at the moment. Retry.	
▶ 20:16:20	My favorite color of the rainbow is red.
No newer events found at the moment. Retry.	

7. Affichez les journaux des autres tâches enfants. Chaque tâche renvoie une couleur différente de l'arc-en-ciel.

Exécuter des tâches GPU

Les tâches GPU vous aident à exécuter des tâches qui utilisent celles d'une instance GPUs.

Les types d'instances GPU Amazon EC2 suivants sont pris en charge. [Pour plus d'informations, consultez les instances Amazon EC2 G3, les instances Amazon EC2 G4, les instances Amazon EC2 G5, les instances Amazon EC2 G6, les instances Amazon EC2 P2, les instances Amazon EC2 P3, les instances Amazon EC2 P4d, les instances Amazon EC2 P5, les instances Amazon EC2 P6, les instances Amazon EC2 Trn1, les instances Amazon EC2 Trn2, les instances Amazon EC2 Inf1, les instances Amazon EC2 Inf2, les instances Amazon EC2 D1 et les instances Amazon EC2 D12.](#)

Type d'instance	GPUs	Mémoire GPU	v CPUs	Mémoire	Bande passante du réseau
g3s.xlarge	1	8 GiO	4	30,5 GiB	10 Gbit/s
g3.4xlarge	1	8 GiO	16	122 GiB	Jusqu'à 10 Gbit/s
g3.8xlarge	2	16 GiO	32	244 GiB	10 Gbit/s

Type d'instance	GPUs	Mémoire GPU	v CPUs	Mémoire	Bande passante du réseau
g3.16xlarge	4	32 GiO	64	488 GiB	25 Gb/s
g4dn.xlarge	1	16 GiO	4	16 GiO	Jusqu'à 25 Gbit/s
g4dn.2xlarge	1	16 GiO	8	32 GiO	Jusqu'à 25 Gbit/s
g4dn.4xlarge	1	16 GiO	16	64 Go	Jusqu'à 25 Gbit/s
g4dn.8xlarge	1	16 GiO	32	128 GiO	50 Gb/s
g4dn.12xlarge	4	64 Go	48	192 Go	50 Gb/s
g4dn.16xlarge	1	16 GiO	64	256 GiO	50 Gb/s
g5.xlarge	1	24 GiO	4	16 GiO	Jusqu'à 10 Gbit/s
g5.2xlarge	1	24 GiO	8	32 GiO	Jusqu'à 10 Gbit/s
g5.4xlarge	1	24 GiO	16	64 Go	Jusqu'à 25 Gbit/s
g5.8xlarge	1	24 GiO	32	128 GiO	25 Gb/s
g5.16xlarge	1	24 GiO	64	256 GiO	25 Gb/s
g5.12xlarge	4	96 GiB	48	192 Go	40 Gbit/s
g5.24xlarge	4	96 GiB	96	384 Go	50 Gb/s
g5.48xlarge	8	192 Go	192	768 GiO	100 Gbit/s
g5g.xlarge	1	16 GiO	4	8 GiO	Jusqu'à 10 Gbit/s
g5g.2xlarge	1	16 GiO	8	16 GiO	Jusqu'à 10 Gbit/s
g5g.4xlarge	1	16 GiO	16	32 GiO	Jusqu'à 10 Gbit/s
g5g.8xlarge	1	16 GiO	32	64 Go	12 Gb/s
g5g.16xlarge	2	32 GiO	64	128 GiO	25 Gb/s

Type d'instance	GPUs	Mémoire GPU	v CPUs	Mémoire	Bande passante du réseau
g5g.metal	2	32 GiO	64	128 Gio	25 Gb/s
g6.xlarge	1	24 GiO	4	16 GiO	Jusqu'à 10 Gbit/s
g6.2xlarge	1	24 GiO	8	32 GiO	Jusqu'à 10 Gbit/s
g6.4xlarge	1	24 GiO	16	64 Go	Jusqu'à 25 Gbit/s
g6.8xlarge	1	24 GiO	32	128 Gio	25 Gb/s
g6.16xlarge	1	24 GiO	64	256 Gio	25 Gb/s
g6.12xlarge	4	96 GiB	48	192 Go	40 Gbit/s
g6.24xlarge	4	96 GiB	96	384 Go	50 Gb/s
g6.48xlarge	8	192 Go	192	768 Gio	100 Gbit/s
g6e.xlarge	1	48 GiO	4	32 GiO	Jusqu'à 20 Gbit/s
g6e.2xlarge	1	48 GiO	8	64 Go	Jusqu'à 20 Gbit/s
g6e.4xlarge	1	48 GiO	16	128 Gio	20 Gbit/s
g6e.8xlarge	1	48 GiO	32	256 Gio	25 Gb/s
g6e.16xlarge	1	48 GiO	64	512 Gio	35 Gbit/s
g6e.12xlarge	4	192 Go	48	384 Go	100 Gbit/s
g6e.24xlarge	4	192 Go	96	768 Gio	200 Gb/s
g6e.48xlarge	8	384 Go	192	1536 GiB	400 Gb/s
gr6.4xlarge	1	24 GiO	16	128 Gio	Jusqu'à 25 Gbit/s
gr6.8xlarge	1	24 GiO	32	256 Gio	25 Gb/s
p2.xlarge	1	12 GiO	4	61 GiB	Élevée

Type d'instance	GPUs	Mémoire GPU	v CPUs	Mémoire	Bande passante du réseau
p2.8xlarge	8	96 GiB	32	488 GiB	10 Gbit/s
p2.16xlarge	16	192 Go	64	732 GiB	20 Gbit/s
p3.2xlarge	1	16 GiO	8	61 GiB	Jusqu'à 10 Gbit/s
p3.8xlarge	4	64 Go	32	244 GiB	10 Gbit/s
p3.16xlarge	8	128 Gio	64	488 GiB	25 Gb/s
p3dn.24xlarge	8	256 Gio	96	768 Gio	100 Gbit/s
p4d.24xlarge	8	320 GiB	96	1152 GiB	400 Gb/s
p4de.24xlarge	8	640 GiB	96	1152 GiB	400 Gb/s
p5.48xlarge	8	640 GiB	192	2 Tio	3200 Gbit/s
p5e.48xlarge	8	1128 GiB	192	2 Tio	3200 Gbit/s
p5en.48xlarge	8	1128 GiB	192	2 Tio	3200 Gbit/s
p6-b200.48xlarge	8	1440 GiB	192	2 Tio	100 Gbit/s
trn1.2xlarge	1	32 GiO	8	32 GiO	Jusqu'à 12,5 Gbps
trn1.32xlarge	16	512 Gio	128	512 Gio	800 Gb/s
trn1n.32xlarge	16	512 Gio	128	512 Gio	1600 Gbit/s
trn2.48xlarge	16	1,5 Tio	192	2 Tio	3,2 Tbit/s
inf1.xlarge	1	8 GiO	4	8 GiO	Jusqu'à 25 Gbit/s
inf1.2xlarge	1	8 GiO	8	16 GiO	Jusqu'à 25 Gbit/s
inf1.6xlarge	4	32 GiO	24	48 GiO	25 Gb/s
inf1.24xlarge	16	128 Gio	96	192 Go	100 Gbit/s

Type d'instance	GPUs	Mémoire GPU	v CPUs	Mémoire	Bande passante du réseau
inf2.xlarge	1	32 GiO	4	16 GiO	Jusqu'à 15 Gbit/s
inf2.8xlarge	1	32 GiO	32	128 Gio	Jusqu'à 25 Gbit/s
inf2.24xlarge	6	192 Go	96	384 Go	50 Gb/s
inf2.48xlarge	12	384 Go	192	768 Gio	100 Gbit/s
dl1.24xlarge	8	256 Gio	96	768 Gio	400 Gb/s
dl2q.24xlarge	8	128 Gio	96	768 Gio	100 Gbit/s

Note

Pour les tâches GPU, AWS Batch seuls les types d'instance dotés de NVIDIA sont pris en charge GPUs. Par exemple, la [G4ad](#) famille n'est pas prise en charge pour la planification du GPU. Vous pouvez toujours utiliser cette [G4ad](#) option AWS Batch en définissant uniquement les exigences en termes de processeur virtuel et de mémoire dans la définition de la tâche, puis en accédant directement aux GPU hôtes en personnalisant les données [utilisateur d'un modèle de lancement Amazon EC2](#) avec une AMI optimisée pour le calcul Amazon ECS ou Amazon EKS, ou une AMI personnalisée pour utiliser AMD. GPUs

Les types d'instances qui utilisent une ARM64 architecture sont pris en charge pour les tâches GPU sur mesure AMIs fournies AWS Batch ou pour les données utilisateur Amazon EC2 permettant d'y accéder GPUs par un code et des configurations personnalisés. Par exemple, la famille d'[G5g](#) instances.

Le paramètre [ResourceRequirements](#) pour la définition de tâche indique le nombre de GPUs à épingle au conteneur. Ce nombre de GPUs n'est disponible pour aucune autre tâche exécutée sur cette instance pendant la durée de cette tâche. Dans un environnement informatique, tous les types d'instances qui exécutent des tâches GPU doivent appartenir aux familles d'g6 instances p3 p4 p5 p6 g3g3s,g4,g5,, ou. Si cela n'est pas fait, une tâche du GPU risque de rester bloquée dans l'`RUNNABLE` état.

Les tâches qui n'utilisent pas le GPU peuvent être exécutées sur des instances de GPU. Cependant, leur exécution sur les instances GPU peut être plus coûteuse que sur des instances similaires autres que le GPU. En fonction du vCPU, de la mémoire et du temps nécessaires, ces tâches non liées au GPU peuvent bloquer l'exécution des tâches GPU.

Rubriques

- [Création d'un Kubernetes cluster basé sur un GPU sur Amazon EKS](#)
- [Création d'une définition de tâche GPU Amazon EKS](#)
- [Exécuter une tâche GPU dans votre cluster Amazon EKS](#)

Création d'un Kubernetes cluster basé sur un GPU sur Amazon EKS

Avant de créer un Kubernetes cluster basé sur un GPU sur Amazon EKS, vous devez avoir effectué les étapes décrites dans [Commencer à utiliser AWS Batch sur Amazon EKS](#). En outre, tenez également compte des points suivants :

- AWS Batch prend en charge les types d'instances avec NVIDIA GPUs.
- Par défaut, AWS Batch sélectionne l'AMI accélérée Amazon EKS avec la Kubernetes version qui correspond à la version de votre plan de contrôle du cluster Amazon EKS.

```
$ cat <<EOF > ./batch-eks-gpu-ce.json
{
  "computeEnvironmentName": "My-Eks-GPU-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:<region>:<account>:cluster/<cluster-name>",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 1024,
    "instanceTypes": [
      "p3dn.24xlarge",
      "p4d.24xlarge"
    ],
    "subnets": [
```



```

    "<eks-cluster-subnets-with-access-to-internet-for-image-pull>"
  ],
  "securityGroupIds": [
    "<eks-cluster-sg>"
  ],
  "instanceRole": "<eks-instance-profile>"
}
}
EOF

$ aws batch create-compute-environment --cli-input-json file:///./batch-eks-gpu-ce.json

```

AWS Batch ne gère pas le plug-in du périphérique NVIDIA GPU à votre place. Vous devez installer ce plug-in dans votre cluster Amazon EKS et l'autoriser à cibler les AWS Batch nœuds. Pour plus d'informations, consultez la section [Activation du support GPU in Kubernetes](#) on GitHub.

Pour configurer le plug-in de NVIDIA périphérique (DaemonSet) afin de cibler les AWS Batch nœuds, exécutez les commandes suivantes.

```

# pull nvidia daemonset spec
$ curl -O https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/v0.12.2/nvidia-device-plugin.yml
# using your favorite editor, add Batch node toleration
# this will allow the DaemonSet to run on Batch nodes
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"

$ kubectl apply -f nvidia-device-plugin.yml

```

Nous vous déconseillons de mélanger des charges de travail basées sur le calcul (processeur et mémoire) avec des charges de travail basées sur le GPU dans les mêmes associations d'environnement informatique et de file d'attente de tâches. Cela est dû au fait que les tâches de calcul peuvent utiliser la capacité du GPU.

Pour attacher des files d'attente de tâches, exécutez les commandes suivantes.

```

$ cat <<EOF > ./batch-eks-gpu-jq.json
{
  "jobQueueName": "My-Eks-GPU-JQ1",
  "priority": 10,
  "computeEnvironmentOrder": [
    {

```

```

        "order": 1,
        "computeEnvironment": "My-Eks-GPU-CE1"
    }
]
}
EOF

```

```
$ aws batch create-job-queue --cli-input-json file:///./batch-eks-gpu-jq.json
```

Création d'une définition de tâche GPU Amazon EKS

Seule `nvidia.com/gpu` est prise en charge pour le moment et la valeur de ressource que vous définissez doit être un nombre entier. Vous ne pouvez pas utiliser de fractions de GPU. Pour plus d'informations, consultez la section [Planification GPUs](#) dans la Kubernetes documentation.

Pour enregistrer une définition de tâche GPU pour Amazon EKS, exécutez les commandes suivantes.

```

$ cat <<EOF > ./batch-eks-gpu-jd.json
{
  "jobDefinitionName": "MyGPUJobOnEks_Smi",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
        {
          "image": "nvcr.io/nvidia/cuda:10.2-runtime-centos7",
          "command": ["nvidia-smi"],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi",
              "nvidia.com/gpu": "1"
            }
          }
        }
      ]
    }
  }
}
EOF

```

```
$ aws batch register-job-definition --cli-input-json file://./batch-eks-gpu-jd.json
```

Exécuter une tâche GPU dans votre cluster Amazon EKS

La ressource GPU n'est pas compressible. AWS Batch crée une spécification de module pour les tâches GPU où la valeur de la requête est égale à la valeur des limites. Il s'agit d'une Kubernetes exigence.

Pour soumettre une tâche GPU, exécutez les commandes suivantes.

```
$ aws batch submit-job --job-queue My-Eks-GPU-JQ1 --job-definition MyGPUJobOnEks_Smi --
job-name My-Eks-GPU-Job

# locate information that can help debug or find logs (if using Amazon CloudWatch Logs
with Fluent Bit)
$ aws batch describe-jobs --job <job-id> | jq '.jobs[].eksProperties.podProperties |
{podName, nodeName}'
{
  "podName": "aws-batch.f3d697c4-3bb5-3955-aa6c-977fcf1cb0ca",
  "nodeName": "ip-192-168-59-101.ec2.internal"
}
```

Afficher les AWS Batch tâches dans une file d'attente

Vous pouvez consulter et filtrer vos offres d'emploi dans AWS Batch. Cette fonctionnalité permet d'afficher une file d'attente de tâches existante et de filtrer ses tâches selon l'une des trois options suivantes.

La recherche et le filtrage permettent de récupérer des tâches qui ne sont pas dans un état terminal (SUCCEEDED ou FAILED). Une fois que l'état d'une tâche est SUCCEEDED atteint, FAILED vous devriez être en mesure de la récupérer pendant sept jours au maximum. Vous pouvez toujours consulter les EventBridge journaux d'une offre d'emploi CloudWatch ou d'Amazon.

Utilisez cette procédure pour répertorier toutes les tâches d'une file d'attente de tâches dans la AWS Batch console. Vous pouvez éventuellement utiliser le champ Filtrer les résultats pour affiner les résultats en fonction des critères que vous spécifiez.

1. Accédez à la [console AWS Batch](#).
2. Dans le volet de navigation, sélectionnez Jobs.

3. Développez la liste déroulante des files d'attente de tâches et choisissez la file d'attente de tâches dans laquelle vous souhaitez effectuer la recherche.

 Note

Vous ne pouvez rechercher des offres d'emploi que dans une seule file d'attente à la fois.

4. Dans le champ Filtrer les résultats, entrez les mots clés à inclure dans les résultats. Vous pouvez utiliser ce champ pour filtrer par nom du Job, Status ou Job ID. Selon la propriété, il se peut que vous deviez définir des opérateurs supplémentaires, tels que égal (=) ou contient (:).

 Note

SageMaker Les files d'attente pour les tâches de formation prennent uniquement en charge le filtrage par nom de tâche et ID de tâche

5. Choisissez Rechercher.

Rechercher AWS Batch des offres d'emploi dans une file d'attente

Vous pouvez rechercher et filtrer vos offres d'emploi à AWS Batch l'aide de la fonction de recherche d'emploi. Cette fonctionnalité permet de rechercher dans une file d'attente de tâches existante et de filtrer ses tâches.

La recherche et le filtrage permettent de récupérer des tâches qui ne sont pas dans un état terminal (SUCCEEDED ou FAILED). Une fois que l'état d'une tâche est SUCCEEDED atteint, FAILED vous devriez être en mesure de la récupérer pendant sept jours au maximum. Vous pouvez toujours consulter les EventBridge journaux d'une offre d'emploi CloudWatch ou d'Amazon.

Pour effectuer une recherche à l'aide de plusieurs critères simultanément, utilisez la fonction de recherche avancée. Par exemple, vous pouvez inclure l'un ou l'ensemble des filtres suivants : statut, plage de dates et critères supplémentaires (tels que le nom de la tâche, la définition de la tâche ou l'identifiant de la tâche).

Rechercher des AWS Batch offres d'emploi (AWS Console)

Utilisez cette procédure pour rechercher les tâches dans une file d'attente de tâches dans la AWS Batch console.

1. Accédez à la [console AWS Batch](#).
2. Dans le volet de navigation, sélectionnez Jobs.
3. Activez la recherche avancée.
4. Développez la liste déroulante des files d'attente de tâches et choisissez la file d'attente de tâches dans laquelle vous souhaitez effectuer la recherche.

 Note

Vous ne pouvez rechercher des offres d'emploi que dans une seule file d'attente à la fois.

5. Pour les options de recherche :
 - a. Pour la liste déroulante des statuts, vous pouvez choisir un ou plusieurs statuts sur lesquels filtrer. Pour plus d'informations, consultez [États de l'emploi](#) et [État du poste de service](#).

 Note

Les parents des tâches matricielles sont informés PENDING lorsqu'une tâche enfant est mise à jour RUNNABLE et conservent PENDING leur statut pendant l'exécution des tâches enfants. Pour consulter ces emplois, filtrez par PENDING statut jusqu'à ce que tous les emplois secondaires atteignent l'état terminal.

- b. Choisissez Plage de dates pour filtrer les résultats en fonction d'une plage de dates et d'heures.
 - Choisissez le mode relatif pour rechercher des emplois dont la date de création se situe dans un intervalle de temps à compter de la date et de l'heure actuelles.
 - Choisissez le mode Absolu pour rechercher des emplois dont la date de création se situe dans une plage de dates et d'heures que vous spécifiez.
 - c. Dans le champ Critères supplémentaires, entrez les mots clés à inclure dans les résultats de recherche. Par exemple, vous pouvez utiliser ce champ pour effectuer une recherche par nom du Job, par définition du Job, par ID du Job ou par identifiant de partage. Selon la propriété, il se peut que vous deviez définir des opérateurs supplémentaires, tels que égal (=) ou contient (:).

 Note

SageMaker Les files d'attente pour les tâches de formation prennent uniquement en charge le filtrage par nom de tâche et ID de tâche

 Note

Lorsque vous filtrez par identifiant de partage, vous pouvez également spécifier le statut de la tâche. Il s'agit d'une exception à la restriction selon laquelle d'autres filtres excluent le filtrage du statut des tâches.

6. Choisissez Rechercher.

Rechercher et filtrer les AWS Batch offres d'emploi (AWS CLI)

Utilisez cette procédure pour répertorier toutes les tâches d'une file d'attente avec le AWS CLI. Vous pouvez éventuellement utiliser le paramètre `-filters` pour affiner les résultats en fonction des critères que vous spécifiez.

Search job queue (AWS CLI)

Vous pouvez utiliser la commande [list-jobs](#) pour rechercher et filtrer une file d'attente de tâches.

Par exemple, vous pouvez effectuer une recherche dans une file d'attente de tâches en fonction du nom de la tâche :

```
aws batch list-jobs \  
  --job-queue my-job-queue \  
  --filters name=JOB_NAME,values="my-job"
```

Filtrez les offres d'emploi par identifiant de partage :

```
aws batch list-jobs \  
  --job-queue my-job-queue \  
  --filters name=SHARE_IDENTIFIER,values="my-share"
```

Lorsque vous filtrez par identifiant de partage, vous pouvez inclure le statut de la tâche :

```
aws batch list-jobs \  
  --job-queue my-job-queue \  
  --job-status RUNNING \  
  --filters name=SHARE_IDENTIFIER,values="my-share"
```

Dans l'exemple précédent, apportez les modifications suivantes :

- *my-job-queue* Remplacez-le par le nom de votre file d'attente de tâches.
- Remplacez *my-job* par le nom de votre poste.
- *my-share* Remplacez-le par l'identifiant de partage selon lequel vous souhaitez filtrer.

Search service job queue (AWS CLI)

Vous pouvez utiliser cette [list-service-jobs](#) commande pour rechercher et filtrer une file d'attente de tâches de service.

Par exemple, vous pouvez effectuer une recherche dans une file d'attente de tâches de service en fonction du nom de la tâche :

```
aws batch list-service-jobs \  
  --job-queue my-sm-queue \  
  --filters name=JOB_NAME,values="my-sm-job"
```

Filtrez les tâches de service par identifiant de partage :

```
aws batch list-service-jobs \  
  --job-queue my-sm-queue \  
  --filters name=SHARE_IDENTIFIER,values="my-share"
```

Dans l'exemple précédent, apportez les modifications suivantes :

- *my-sm-queue* Remplacez-le par le nom de votre file d'attente de tâches de service.
- *my-sm-job* Remplacez-le par le nom de votre tâche de service.
- *my-share* Remplacez-le par l'identifiant de partage selon lequel vous souhaitez filtrer.

Modes de mise en réseau pour les AWS Batch tâches

Le tableau suivant décrit les modes de mise en réseau et leur utilisation typique pour les types de AWS Batch tâches. Consultez les liens dans la colonne « Type de travail » pour plus de détails concernant les considérations et les comportements.

Type de tâche	Mode (s) réseau pris en charge	Usage typique
Tâche simple ECS-EC2	host	Utilisé pour les charges de travail par lots les plus évolutives et les plus embarrassantes qui nécessitent uniquement une sortie vers un vPC défini dans l'environnement de calcul.
Tâche parallèle à plusieurs nœuds ECS-EC2	awsvpc	Utilisé pour les charges de travail distribuées à plusieurs hôtes (nœuds) étroitement couplées, modélisées comme une tâche unique avec des communications coordonnées entre les nœuds de tâches.
Tâche simple avec ECS-Fargate	awsvpc	Un véritable système sans serveur pour des charges de travail par lots extrêmement parallèles. Modèle de travail d'isolation des conteneurs généralement le plus bas et le plus élevé.
Tâche simple sur EKS-EC2	hôte et pod	Utilisé pour les charges de travail par lots hautement évolutives et extrêmement parallèles qui nécessitent uniquement une sortie vers

Type de tâche	Mode (s) réseau pris en charge	Usage typique
		un vPC défini dans l'environnement de calcul. La valeur par défaut est le réseau hôte.
Tâche parallèle à nœuds multiples EKS-EC2	hôte et pod	Utilisé pour les charges de travail distribuées à plusieurs hôtes (nœuds) étroitement couplées, modélisées comme une tâche unique avec des communications coordonnées entre les nœuds du pod. La valeur par défaut est le réseau hôte.

Afficher les journaux des AWS Batch tâches dans CloudWatch Logs

Vous pouvez [configurer vos AWS Batch tâches](#) pour envoyer des informations de journal à Amazon CloudWatch Logs. Ainsi, vous pouvez consulter les différents journaux de vos tâches en un seul endroit pratique. Pour de plus amples informations, veuillez consulter [Utilisation CloudWatch des journaux avec AWS Batch](#).

Vous pouvez également utiliser les journaux des tâches dans la AWS Batch console pour surveiller ou résoudre les problèmes liés à une AWS Batch tâche.

1. Ouvrez la [AWS Batch console](#).
2. Choisissez Tâches. Pour des informations plus détaillées sur le tri et le filtrage des tâches dans la file d'attente des tâches, consultez [Afficher les AWS Batch tâches dans une file d'attente](#) et [Rechercher des offres d'emploi dans une file d'attente](#)
3. Pour la file d'attente des tâches, choisissez la file d'attente des tâches que vous souhaitez.

 Tip

Si plusieurs tâches figurent dans la file d'attente, vous pouvez activer la recherche et le filtrage pour trouver une tâche plus rapidement. Pour de plus amples informations, veuillez consulter [Rechercher AWS Batch des offres d'emploi dans une file d'attente](#).

4. Dans le champ Statut, choisissez le statut du poste que vous souhaitez.
5. Choisissez le travail que vous souhaitez et la page de détails s'ouvre.
6. Sur la page Détails, faites défiler la page vers le bas jusqu'au nom du flux du journal et choisissez le lien. Le lien ouvre la page Amazon CloudWatch Logs correspondant à la tâche.
7. (Facultatif) Si c'est la première fois que vous consultez les journaux, une autorisation peut vous être demandée.

Dans Autorisation requise, entrez **OK**, puis choisissez Autoriser pour accepter les CloudWatch frais Amazon.

 Note

Pour révoquer votre autorisation d'encaisser des CloudWatch frais, procédez comme suit :

1. Dans le volet de navigation de gauche, choisissez Permissions.
2. Pour Job logs, choisissez Edit.
3. Désactivez la case à CloudWatch cocher Autoriser le lot à utiliser.
4. Sélectionnez Enregistrer les modifications.

Consulter les informations relatives au AWS Batch poste

Vous pouvez consulter les informations relatives à la AWS Batch tâche, telles que le statut, la définition de la tâche et les informations relatives au conteneur.

1. Ouvrez la [AWS Batch console](#).
2. Choisissez Tâches.
3. Pour la file d'attente des tâches, choisissez la file d'attente des tâches que vous souhaitez.

 Tip

Si plusieurs tâches figurent dans la file d'attente, vous pouvez activer la recherche et filtrer pour trouver une offre d'emploi plus rapidement. Pour de plus amples informations, veuillez consulter [Rechercher AWS Batch des offres d'emploi dans une file d'attente](#).

4. Choisissez le poste que vous souhaitez.

 Note

Vous pouvez également utiliser le AWS Command Line Interface (AWS CLI) pour afficher les détails d'une AWS Batch tâche. Pour plus d'informations, consultez [describe-jobs](#) dans le manuel de référence des [AWS CLI commandes](#).

Sécurité dans AWS Batch

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le [modèle de responsabilité partagée](#) décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- **Sécurité du cloud** : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS Cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de [AWS conformité Programmes](#) de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS Batch, voir [AWS Services concernés par programme de conformitéAWS](#) .
- **Sécurité dans le cloud** : votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre entreprise, ainsi que la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de son utilisation AWS Batch. Les rubriques suivantes expliquent comment procéder à la configuration AWS Batch 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 AWS Batch ressources.

Rubriques

- [Identity and Access Management pour AWS Batch](#)
- [AWS Batch Politiques, rôles et autorisations IAM](#)
- [AWS Batch Rôle d'exécution IAM](#)
- [Créer un Virtual Private Cloud](#)
- [Utiliser un point de terminaison d'interface pour accéder AWS Batch](#)
- [Validation de conformité pour AWS Batch](#)
- [Sécurité de l'infrastructure dans AWS Batch](#)
- [Prévention du problème de l'adjoint confus entre services](#)
- [Journalisation des appels d' AWS Batch API avec AWS CloudTrail](#)

- [Résoudre les problèmes liés à l'IAM AWS Batch](#)

Identity and Access Management pour AWS Batch

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser AWS Batch les ressources. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- [Public ciblé](#)
- [Authentification par des identités](#)
- [Gestion de l'accès à l'aide de politiques](#)
- [Comment AWS Batch fonctionne avec IAM](#)
- [Exemples de politiques basées sur l'identité pour AWS Batch](#)
- [AWS politiques gérées pour AWS Batch](#)

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction de votre rôle :

- Utilisateur du service : demandez des autorisations à votre administrateur si vous ne pouvez pas accéder aux fonctionnalités (voir [Résoudre les problèmes liés à l'IAM AWS Batch](#))
- Administrateur du service : déterminez l'accès des utilisateurs et soumettez les demandes d'autorisation (voir [Comment AWS Batch fonctionne avec IAM](#))
- Administrateur IAM : rédigez des politiques pour gérer l'accès (voir [Exemples de politiques basées sur l'identité pour AWS Batch](#))

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en tant qu'identité fédérée à l'aide d'informations d'identification provenant d'une source d'identité telle que AWS IAM Identity Center (IAM Identity Center), d'une authentification unique ou d'informations d'identification. Google/Facebook Pour plus d'informations sur la connexion, consultez [Connexion à votre Compte AWS](#) dans le Guide de l'utilisateur Connexion à AWS .

Pour l'accès par programmation, AWS fournit un SDK et une CLI pour signer les demandes de manière cryptographique. Pour plus d'informations, consultez [Signature AWS Version 4 pour les demandes d'API](#) dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une seule identité de connexion appelée utilisateur Compte AWS root qui dispose d'un accès complet à toutes Services AWS les ressources. Il est vivement déconseillé d'utiliser l'utilisateur racine pour vos tâches quotidiennes. Pour les tâches qui requièrent des informations d'identification de l'utilisateur racine, consultez [Tâches qui requièrent les informations d'identification de l'utilisateur racine](#) dans le Guide de l'utilisateur IAM.

Identité fédérée

Il est recommandé d'obliger les utilisateurs humains à utiliser la fédération avec un fournisseur d'identité pour accéder à Services AWS l'aide d'informations d'identification temporaires.

Une identité fédérée est un utilisateur provenant de l'annuaire de votre entreprise, de votre fournisseur d'identité Web ou Directory Service qui y accède à Services AWS l'aide d'informations d'identification provenant d'une source d'identité. Les identités fédérées assument des rôles qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Pour plus d'informations, consultez [Qu'est-ce que IAM Identity Center ?](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un [utilisateur IAM](#) est une identité qui dispose d'autorisations spécifiques pour une seule personne ou application. Nous vous recommandons d'utiliser ces informations d'identification temporaires au lieu des utilisateurs IAM avec des informations d'identification à long terme. Pour plus d'informations, voir [Exiger des utilisateurs humains qu'ils utilisent la fédération avec un fournisseur d'identité pour accéder à AWS l'aide d'informations d'identification temporaires](#) dans le guide de l'utilisateur IAM.

[Les groupes IAM](#) spécifient une collection d'utilisateurs IAM et permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Pour plus d'informations, consultez [Cas d'utilisation pour les utilisateurs IAM](#) dans le Guide de l'utilisateur IAM.

Rôles IAM

Un [rôle IAM](#) est une identité dotée d'autorisations spécifiques qui fournit des informations d'identification temporaires. Vous pouvez assumer un rôle en [passant d'un rôle d'utilisateur à un rôle IAM \(console\)](#) ou en appelant une opération d' AWS API AWS CLI ou d'API. Pour plus d'informations, consultez [Méthodes pour endosser un rôle](#) dans le Guide de l'utilisateur IAM.

Les rôles IAM sont utiles pour l'accès des utilisateurs fédérés, les autorisations temporaires des utilisateurs IAM, les accès intercompte, les accès entre services et les applications exécutées sur Amazon EC2. Pour plus d'informations, consultez [Accès intercompte aux ressources dans IAM](#) dans le Guide de l'utilisateur IAM.

Gestion de l'accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique définit les autorisations lorsqu'elles sont associées à une identité ou à une ressource. AWS évalue ces politiques lorsqu'un directeur fait une demande. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations les documents de politique JSON, consultez [Vue d'ensemble des politiques JSON](#) dans le Guide de l'utilisateur IAM.

À l'aide de politiques, les administrateurs précisent qui a accès à quoi en définissant quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Un administrateur IAM crée des politiques IAM et les ajoute aux rôles, que les utilisateurs peuvent ensuite assumer. Les politiques IAM définissent les autorisations quelle que soit la méthode que vous utilisez pour exécuter l'opération.

Politiques basées sur l'identité

Les stratégies basées sur l'identité sont des documents de stratégie d'autorisations JSON que vous attachez à une identité (utilisateur, groupe ou rôle). Ces politiques contrôlent les actions que peuvent exécuter ces identités, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez [Définition d'autorisations IAM personnalisées avec des politiques gérées par le client](#) dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être des politiques intégrées (intégrées directement dans une seule identité) ou des politiques gérées (politiques autonomes associées à plusieurs identités). Pour découvrir comment choisir entre des politiques gérées et en ligne, consultez [Choix entre les politiques gérées et les politiques en ligne](#) dans le Guide de l'utilisateur IAM.

Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Les exemples incluent les politiques de confiance de rôle IAM et les stratégies de compartiment Amazon S3. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Vous devez [spécifier un principal](#) dans une politique basée sur les ressources.

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Autres types de politique

AWS prend en charge des types de politiques supplémentaires qui peuvent définir les autorisations maximales accordées par les types de politiques les plus courants :

- Limites d'autorisations : une limite des autorisations définit le nombre maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM. Pour plus d'informations, consultez [Limites d'autorisations pour des entités IAM](#) dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCPs) — Spécifiez les autorisations maximales pour une organisation ou une unité organisationnelle dans AWS Organizations. Pour plus d'informations, consultez [Politiques de contrôle de service](#) dans le Guide de l'utilisateur AWS Organizations .
- Politiques de contrôle des ressources (RCPs) : définissez le maximum d'autorisations disponibles pour les ressources de vos comptes. Pour plus d'informations, voir [Politiques de contrôle des ressources \(RCPs\)](#) dans le guide de AWS Organizations l'utilisateur.
- Politiques de session : politiques avancées que vous passez en tant que paramètre lorsque vous créez par programmation une session temporaire pour un rôle ou un utilisateur fédéré. Pour plus d'informations, consultez [Politiques de session](#) dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section [Logique d'évaluation des politiques](#) dans le guide de l'utilisateur IAM.

Comment AWS Batch fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à AWS Batch, découvrez les fonctionnalités IAM disponibles. AWS Batch

Fonctionnalités IAM que vous pouvez utiliser avec AWS Batch

Fonctionnalité IAM	AWS Batch soutien
Politiques basées sur l'identité	Oui
Politiques basées sur les ressources	Non
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique	Oui
ACLs	Non
ABAC (étiquettes dans les politiques)	Oui
Informations d'identification temporaires	Oui
Autorisations de principal	Oui
Rôles de service	Oui
Rôles liés à un service	Oui

Pour obtenir une vue d'ensemble de la façon dont AWS Batch les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez la section [AWS Services compatibles avec IAM](#) dans le Guide de l'utilisateur IAM.

Politiques basées sur l'identité pour AWS Batch

Prend en charge les politiques basées sur l'identité : oui

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. 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 [Définition d'autorisations IAM personnalisées avec des politiques gérées par le client](#) 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. 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 AWS Batch

Pour consulter des exemples de politiques AWS Batch basées sur l'identité, consultez. [Exemples de politiques basées sur l'identité pour AWS Batch](#)

Actions politiques pour AWS Batch

Prend en charge les actions de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des AWS Batch actions, reportez-vous à la section [Actions définies par AWS Batch](#) dans la référence d'autorisation de service.

Les actions de politique en AWS Batch cours utilisent le préfixe suivant avant l'action :

```
batch
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [  
    "batch:action1",  
    "batch:action2"  
]
```

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `Describe`, incluez l'action suivante :

```
"Action": "batch:Describe*"
```

Pour consulter des exemples de politiques AWS Batch basées sur l'identité, consultez [Exemples de politiques basées sur l'identité pour AWS Batch](#)

Ressources politiques pour AWS Batch

Prend en charge les ressources de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Il est recommandé de définir une ressource à l'aide de son [Amazon Resource Name \(ARN\)](#). Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

Pour consulter la liste des types de AWS Batch ressources et leurs caractéristiques ARNs, voir [Ressources définies par AWS Batch](#) dans la référence d'autorisation de service. Pour savoir grâce à quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez [Actions définies par AWS Batch](#).

Clés de condition de politique pour AWS Batch

Prend en charge les clés de condition de politique spécifiques au service : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` indique à quel moment les instructions s'exécutent en fonction de critères définis. 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. Pour voir toutes les clés de condition AWS globales, voir les clés de [contexte de condition AWS globales](#) dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de AWS Batch condition, voir [Clés de condition pour AWS Batch](#) la référence d'autorisation de service. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, voir [Actions définies par AWS Batch](#).

Contrôle d'accès basé sur les attributs (ABAC) avec AWS Batch

Prise en charge d'ABAC (balises dans les politiques) : Oui

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit les autorisations en fonction des attributs appelés balises. Vous pouvez associer des balises aux entités et aux AWS ressources IAM, puis concevoir des politiques ABAC pour autoriser les opérations lorsque la balise du principal correspond à la balise de la ressource.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'[élément de condition](#) d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur ABAC, consultez [Définition d'autorisations avec l'autorisation ABAC](#) dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez [Utilisation du contrôle d'accès par attributs \(ABAC\)](#) dans le Guide de l'utilisateur IAM.

Utilisez des informations d'identification temporaires avec AWS Batch

Prend en charge les informations d'identification temporaires : oui

Les informations d'identification temporaires fournissent un accès à court terme aux AWS ressources et sont automatiquement créées lorsque vous utilisez la fédération ou que vous changez de rôle. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez la section [Informations d'identification de sécurité temporaires dans IAM](#) et [Services AWS compatibles avec IAM](#) dans le Guide de l'utilisateur IAM.

Autorisations de principal interservices pour AWS Batch

Prend en charge les sessions d'accès direct (FAS) : oui

Les sessions d'accès direct (FAS) utilisent les autorisations du principal appelant et Service AWS, combinées Service AWS à la demande d'envoi de demandes aux services en aval. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez la section [Sessions de transmission d'accès](#).

Rôles de service pour AWS Batch

Prend en charge les rôles de service : oui

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 un rôle de service à partir d'IAM. Pour plus d'informations, consultez [Création d'un rôle pour la délégation d'autorisations à un Service AWS](#) dans le Guide de l'utilisateur IAM.

Warning

La modification des autorisations associées à un rôle de service peut perturber AWS Batch les fonctionnalités. Ne modifiez des rôles de service que quand AWS Batch vous le conseille.

Rôles liés à un service pour AWS Batch

Prend en charge les rôles liés à un service : oui

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS fichier et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus d'informations sur la création ou la gestion des rôles liés à un service, consultez [Services AWS qui fonctionnent avec IAM](#). Recherchez un service dans le tableau qui inclut un Yes dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

Exemples de politiques basées sur l'identité pour AWS Batch

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou modifier les ressources AWS Batch. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez [Création de politiques IAM \(console\)](#) dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par AWS Batch, y compris le format de ARNs pour chacun des types de ressources, voir [Actions, ressources et clés de condition AWS Batch](#) dans la référence d'autorisation de service.

Rubriques

- [Bonnes pratiques en matière de politiques](#)
- [Utilisation de la AWS Batch console](#)
- [Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations](#)

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer AWS Batch des ressources dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez [politiques gérées par AWS](#) ou [politiques gérées par AWS pour les activités professionnelles](#) dans le Guide de l'utilisateur IAM.

- Accordez les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation d'IAM pour appliquer des autorisations, consultez [politiques et autorisations dans IAM](#) dans le Guide de l'utilisateur IAM.
- Utilisez des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que CloudFormation. Pour plus d'informations, consultez [Conditions pour éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.
- Utilisez l'Analyseur d'accès IAM pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : l'Analyseur d'accès IAM valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez [Validation de politiques avec IAM Access Analyzer](#) dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez [Sécurisation de l'accès aux API avec MFA](#) dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez [Bonnes pratiques de sécurité dans IAM](#) dans le Guide de l'utilisateur IAM.

Utilisation de la AWS Batch console

Pour accéder à la AWS Batch console, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher les détails AWS Batch des ressources de votre Compte AWS. Si vous créez une politique basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette politique.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API qu'ils tentent d'effectuer.

Pour garantir que les utilisateurs et les rôles peuvent toujours utiliser la AWS Batch console, associez également la politique AWS Batch ConsoleAccess ou la politique ReadOnly AWS gérée aux entités. Pour plus d'informations, consultez [Ajout d'autorisations à un utilisateur](#) dans le Guide de l'utilisateur IAM.

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or AWS .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
```



```
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS politiques gérées pour AWS Batch

Vous pouvez utiliser des politiques AWS gérées pour simplifier la gestion des identités et des accès pour votre équipe et les AWS ressources allouées. AWS les politiques gérées couvrent une variété de cas d'utilisation courants, sont disponibles par défaut dans votre AWS compte et sont maintenues et mises à jour en votre nom. Vous ne pouvez pas modifier les autorisations dans les politiques AWS gérées. Si vous avez besoin d'une plus grande flexibilité, vous pouvez également choisir de créer des politiques gérées par le client IAM. Ainsi, vous pouvez fournir aux ressources allouées à votre équipe uniquement les autorisations exactes dont elle a besoin.

Pour plus d'informations sur les politiques AWS gérées, voir les [politiques AWS gérées](#) dans le guide de l'utilisateur IAM.

AWS les services maintiennent et mettent à jour les politiques AWS gérées en votre nom. Régulièrement, les AWS services ajoutent des autorisations supplémentaires à une politique AWS gérée. AWS les politiques gérées sont très probablement mises à jour lorsqu'une nouvelle fonctionnalité est lancée ou qu'une nouvelle opération est disponible. Ces mises à jour affectent automatiquement toutes les identités (utilisateurs, groupes et rôles) auxquelles la politique est attachée. Cependant, ils ne suppriment pas les autorisations et n'enfreignent pas vos autorisations existantes.

En outre, AWS prend en charge les politiques gérées pour les fonctions professionnelles qui couvrent plusieurs services. Par exemple, la politique `ReadOnlyAccess` AWS gérée fournit un accès en lecture seule à tous les AWS services et ressources. Lorsqu'un service lance une nouvelle fonctionnalité, il AWS ajoute des autorisations en lecture seule pour les nouvelles opérations et ressources. Pour obtenir la liste des politiques de fonctions professionnelles et leurs descriptions, consultez la page [politiques gérées par AWS pour les fonctions de tâche](#) dans le Guide de l'utilisateur IAM.

AWS politique gérée : BatchServiceRolePolicy

La politique IAM BatchServiceRolePolicygérée est utilisée par le rôle lié au [AWSServiceRoleForBatch](#) service. Cela permet AWS Batch d'effectuer des actions en votre nom. Vous ne pouvez pas attacher cette politique à vos entités IAM. Pour de plus amples informations, veuillez consulter [Utilisation des rôles liés à un service pour AWS Batch](#).

Cette politique permet AWS Batch de réaliser les actions suivantes sur des ressources spécifiques :

- **autoscaling**— Permet de AWS Batch créer et de gérer des ressources Amazon EC2 Auto Scaling. AWS Batch crée et gère des groupes Amazon EC2 Auto Scaling pour la plupart des environnements informatiques.
- **ec2**— Permet AWS Batch de contrôler le cycle de vie des instances Amazon EC2 ainsi que de créer et de gérer des modèles et des balises de lancement. AWS Batch crée et gère les demandes EC2 Spot Fleet pour certains environnements informatiques EC2 Spot.
- **ecs**- Permet AWS Batch de créer et de gérer des clusters Amazon ECS, des définitions de tâches et des tâches pour l'exécution des tâches.
- **eks**- Permet AWS Batch de décrire la ressource du cluster Amazon EKS pour les validations.
- **iam**- Permet AWS Batch de valider et de transmettre les rôles fournis par le propriétaire à Amazon EC2, Amazon EC2 Auto Scaling et Amazon ECS.
- **logs**— Permet AWS Batch de créer et de gérer des groupes de journaux et des flux de journaux pour les AWS Batch tâches.

Pour consulter le JSON de la politique, consultez [BatchServiceRolePolicy](#)le [Guide de référence des politiques AWS gérées](#).

AWS politique gérée : AWSBatchServiceRolePolicyForSageMaker

[AWSServiceRoleForAWSBatchWithSagemaker](#) permet AWS Batch d'effectuer des actions en votre nom. Vous ne pouvez pas attacher cette politique à vos entités IAM. Pour de plus amples informations, veuillez consulter [Utilisation des rôles liés à un service pour AWS Batch](#).

Cette politique permet AWS Batch de réaliser les actions suivantes sur des ressources spécifiques :

- **sagemaker**— Permet de AWS Batch gérer les tâches de formation SageMaker liées à l'IA et les autres ressources de l' SageMaker IA.

- `iam:PassRole`— Permet de AWS Batch transmettre des rôles d'exécution définis par le client à l'Amazon SageMaker IA pour l'exécution des tâches. La contrainte de ressources permet de transmettre des rôles aux services d'Amazon SageMaker IA.

Pour consulter le JSON de la politique, consultez [AWSBatchServiceRolePolicyForSageMaker](#) le [Guide de référence des politiques AWS gérées](#).

AWS stratégie gérée : AWSBatchServiceRolestratégie

La politique d'autorisations de rôle nommée `AWSBatchServiceRole` permet AWS Batch d'effectuer les actions suivantes sur des ressources spécifiques :

La politique IAM `AWSBatchServiceRole` gérée est souvent utilisée par un rôle nommé `AWSBatchServiceRole` et inclut les autorisations suivantes. Conformément aux conseils de sécurité standard pour accorder le moindre privilège, la stratégie gérée par `AWSBatchServiceRole` peut être utilisée comme guide. Si l'une des autorisations qui sont accordées dans la stratégie gérée n'est pas nécessaire pour votre cas d'utilisation, créez une stratégie personnalisée et ajoutez uniquement les autorisations dont vous avez besoin. Cette politique et ce rôle AWS Batch gérés peuvent être utilisés avec la plupart des types d'environnements informatiques, mais l'utilisation de rôles liés à un service est préférable pour une expérience gérée moins sujette aux erreurs, mieux circonscrite et améliorée.

- `autoscaling`— Permet de AWS Batch créer et de gérer des ressources Amazon EC2 Auto Scaling. AWS Batch crée et gère des groupes Amazon EC2 Auto Scaling pour la plupart des environnements informatiques.
- `ec2`— Permet AWS Batch de gérer le cycle de vie des instances Amazon EC2 ainsi que de créer et de gérer des modèles et des balises de lancement. AWS Batch crée et gère les demandes EC2 Spot Fleet pour certains environnements informatiques EC2 Spot.
- `ecs`- Permet AWS Batch de créer et de gérer des clusters Amazon ECS, des définitions de tâches et des tâches pour l'exécution des tâches.
- `iam`- Permet AWS Batch de valider et de transmettre les rôles fournis par le propriétaire à Amazon EC2, Amazon EC2 Auto Scaling et Amazon ECS.
- `logs`— Permet AWS Batch de créer et de gérer des groupes de journaux et des flux de journaux pour les AWS Batch tâches.

Pour consulter le JSON de la politique, consultez [AWSBatchServiceRole](#) le [Guide de référence des politiques AWS gérées](#).

AWS politique gérée : AWSBatchFullAccess

La AWSBatchFullAccess politique accorde aux AWS Batch actions un accès complet aux AWS Batch ressources. Il accorde également l'accès aux actions de description et de liste pour les services Amazon EC2, Amazon ECS CloudWatch, Amazon EKS et IAM. Cela permet aux identités IAM, qu'il s'agisse d'utilisateurs ou de rôles, de consulter les ressources AWS Batch gérées créées en leur nom. Enfin, cette politique permet également de transmettre des rôles IAM sélectionnés à ces services.

Vous pouvez les associer AWSBatchFullAccess à vos entités IAM. AWS Batch associe également cette politique à un rôle de service qui permet AWS Batch d'effectuer des actions en votre nom.

Pour consulter le JSON de la politique, consultez [AWSBatchFullAccess](#) le [Guide de référence des politiques AWS gérées](#).

AWS Batch mises à jour des politiques AWS gérées

Consultez les détails des mises à jour des politiques AWS gérées AWS Batch 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 AWS Batch document.

Modifier	Description	Date
AWSBatchServiceRolePolicyForSageMaker politique ajoutée	Ajout d'une nouvelle politique AWS gérée pour le rôle AWSBatchServiceRolePolicyForSageMaker lié au service qui permet de AWS Batch gérer l' SageMaker IA en votre nom.	31 juillet 2025
BatchServiceRolePolicy politique mise à jour	Mise à jour pour ajouter la prise en charge de la description de l'historique des demandes et des Amazon EC2 Auto Scaling activités de Spot Fleet.	5 décembre 2025
AWSBatchServiceRole politique ajoutée	Mis à jour pour ajouter une déclaration IDs, accorder AWS Batch des autorisations à <code>ec2:DescribeSpotF</code>	5 décembre 2025

Modifier	Description	Date
	etRequestHistory etautoscaling:DescribeScalingActivities .	
BatchServiceRolePolicy politique mise à jour	Mise à jour pour ajouter la prise en charge de la description des clusters Amazon EKS.	20 octobre 2022
AWSBatchFullAccess politique mise à jour	Mise à jour pour ajouter la prise en charge de la liste et de la description des clusters Amazon EKS.	20 octobre 2022
BatchServiceRolePolicy politique mise à jour	Mise à jour pour ajouter la prise en charge des groupes de réservation de capacité Amazon EC2 gérés par. Groupes de ressources AWS Pour plus d'informations, consultez la section Travailler avec des groupes de réservation de capacité dans le guide de l'utilisateur Amazon EC2.	18 mai 2022
BatchServiceRolePolicy et AWSBatchServiceRole politiques mises à jour	Mise à jour pour ajouter la prise en charge de la description de l'état des instances AWS Batch gérées dans Amazon EC2 afin que les instances défectueuses soient remplacées.	6 décembre 2022
BatchServiceRolePolicy politique mise à jour	Mise à jour pour ajouter la prise en charge des groupes de placement, de la réservation de capacité, du GPU élastique et des ressources Elastic Inference dans Amazon EC2.	26 mars 2021

Modifier	Description	Date
BatchServiceRolePolicy politique ajoutée	Avec la politique BatchServiceRolePolicy gérée pour le rôle AWSServiceRoleForBatch lié à un service, vous pouvez utiliser un rôle lié à un service géré par. AWS Batch Avec cette politique, vous n'avez pas besoin de conserver votre propre rôle à utiliser dans vos environnements informatiques.	10 mars 2021
AWSBatchFullAccess - ajouter l'autorisation d'ajouter un rôle lié à un service	Ajoutez des autorisations IAM pour autoriser l'ajout du rôle AWSServiceRoleForBatch lié au service au compte.	10 mars 2021
AWS Batch a commencé à suivre les modifications	AWS Batch a commencé à suivre les modifications apportées AWS à ses politiques gérées.	10 mars 2021

AWS Batch Politiques, rôles et autorisations IAM

Par défaut, les utilisateurs ne sont pas autorisés à créer ou à modifier AWS Batch des ressources ou à effectuer des tâches à l'aide de l' AWS Batch API, de la AWS Batch console ou du AWS CLI. Pour permettre aux utilisateurs d'effectuer ces actions, créez des politiques IAM qui accordent aux utilisateurs l'autorisation d'accéder aux ressources et aux opérations d'API spécifiques. Attachez ensuite les politiques aux utilisateurs ou aux groupes qui ont besoin de ces autorisations.

Lorsque vous associez une politique à un utilisateur ou à un groupe d'utilisateurs, elle autorise ou refuse les autorisations d'effectuer des tâches spécifiques sur des ressources spécifiques. Pour plus d'informations, consultez la section [Permissions et politiques](#) du guide de l'utilisateur IAM. Pour plus d'informations sur la gestion et la création de stratégies IAM personnalisées, consultez [Gestion des stratégies IAM](#).

AWS Batch passe des appels à d'autres Services AWS personnes en votre nom. Par conséquent, vous AWS Batch devez vous authentifier à l'aide de vos informations d'identification. Plus précisément, AWS Batch s'authentifie en créant un rôle et une politique IAM fournissant ces autorisations. Il associe ensuite le rôle à vos environnements informatiques lorsque vous les créez.

Pour plus d'informations [Rôle d'instance Amazon ECS](#), consultez les sections Rôles [IAM](#), [Utilisation de rôles liés à un service](#) et [Création d'un rôle pour déléguer des autorisations à un AWS service](#) dans le Guide de l'utilisateur IAM.

Rubriques

- [Structure des politiques IAM](#)
- [Ressource : exemples de politiques pour AWS Batch](#)
- [Ressource : politique AWS Batch gérée](#)

Structure des politiques IAM

Les rubriques suivantes expliquent la structure d'une politique IAM.

Rubriques

- [Syntaxe d'une politique](#)
- [Actions d'API pour AWS Batch](#)
- [Noms de ressources Amazon pour AWS Batch](#)
- [Vérifiez que les utilisateurs disposent des autorisations requises](#)

Syntaxe d'une politique

Une politique IAM est un document JSON qui se compose d'une ou de plusieurs déclarations. Chaque déclaration est structurée comme suit :

```
{
  "Statement": [{
    "Effect": "effect",
    "Action": "action",
    "Resource": "arn",
    "Condition": {
      "condition": {
        "key": "value"
      }
    }
  ]
}
```

Quatre éléments principaux constituent une déclaration :

- **Effect** : effect peut avoir la valeur `Allow` ou `Deny`. Par défaut, les utilisateurs ne sont pas autorisés à utiliser les ressources et les actions d'API. Toutes les demandes sont donc rejetées. Une autorisation explicite remplace l'autorisation par défaut. Un refus explicite remplace toute autorisation.
- **Action** : L'action est l'action d'API spécifique pour laquelle vous accordez ou refusez l'autorisation. Pour obtenir des instructions sur la manière de spécifier l'action, reportez-vous à la section [Actions d'API pour AWS Batch](#).
- **Resource** : la ressource affectée par l'action. Avec certaines actions d' AWS Batch API, vous pouvez inclure dans votre politique des ressources spécifiques qui peuvent être créées ou modifiées par l'action. Pour spécifier une ressource dans la déclaration, vous utilisez son nom Amazon Resource Name (ARN). Pour plus d'informations, consultez [Autorisations prises en charge au niveau des ressources pour les actions d'API AWS Batch](#) et [Noms de ressources Amazon pour AWS Batch](#). Si l'opération AWS Batch d'API ne prend actuellement pas en charge les autorisations au niveau des ressources, incluez un caractère générique (*) pour indiquer que toutes les ressources peuvent être affectées par l'action.
- **Condition** : les conditions sont facultatives. Elles permettent de contrôler à quel moment votre politique est effective.

Pour plus d'informations sur les exemples de déclarations de politique IAM pour AWS Batch, consultez [Ressource : exemples de politiques pour AWS Batch](#).

Actions d'API pour AWS Batch

Dans une déclaration de politique IAM, vous pouvez spécifier une action d'API à partir de n'importe quel service prenant en charge IAM. Pour AWS Batch, utilisez le préfixe suivant avec le nom de l'action d'API : `batch:` (par exemple, `batch:SubmitJob` et `etbatch:CreateComputeEnvironment`).

Pour spécifier plusieurs actions dans une seule instruction, séparez-les par une virgule.

```
"Action": ["batch:action1", "batch:action2"]
```

Vous pouvez également spécifier plusieurs actions en incluant un caractère générique (*). Par exemple, vous pouvez spécifier toutes les actions avec un nom commençant par le mot « Décrire ».


```
"Action": "batch:Describe"
```

Pour spécifier toutes les actions AWS Batch d'API, incluez un caractère générique (*).

```
"Action": "batch:*"
```

Pour obtenir la liste des AWS Batch actions, consultez la section [Actions](#) dans la référence de l'AWS Batch API.

Noms de ressources Amazon pour AWS Batch

Chaque déclaration de politique IAM s'applique aux ressources que vous spécifiez à l'aide de leur Amazon Resource Names (ARNs).

La syntaxe générale d'un Amazon Resource Name (ARN) est la suivante :

```
arn:aws:[service]:[region]:[account]:resourceType/resourcePath
```

web

Le service (par exemple, batch).

region

Le Région AWS pour la ressource (par exemple, us-east-2).

account

L' Compte AWS identifiant, sans tiret (par exemple, 123456789012).

resourceType

Le type de ressource (par exemple, compute-environment).

chemin de la ressource

Un chemin qui identifie la ressource. Vous pouvez utiliser un caractère générique (*) dans vos tracés.

AWS Batch Les opérations d'API prennent actuellement en charge les autorisations au niveau des ressources pour plusieurs opérations d'API. Pour de plus amples informations, veuillez consulter [Autorisations prises en charge au niveau des ressources pour les actions d'API AWS Batch](#). Pour

spécifier toutes les ressources, ou si une action d'API spécifique n'est pas compatible ARNs, incluez un caractère générique (*) dans l'Resourceélément.

```
"Resource": "*"
```

Vérifiez que les utilisateurs disposent des autorisations requises

Avant de mettre une politique IAM en production, assurez-vous qu'elle accorde aux utilisateurs les autorisations nécessaires pour utiliser les actions et les ressources d'API spécifiques dont ils ont besoin.

Pour ce faire, créez d'abord un utilisateur à des fins de test et associez la politique IAM à l'utilisateur de test. Ensuite, créez une demande en tant qu'utilisateur test. Vous pouvez tester les demandes dans la console ou avec l' AWS CLI.

Note

Vous pouvez également tester vos politiques à l'aide du [simulateur de politiques IAM](#). Pour plus d'informations sur le simulateur de politiques, consultez la section [Utilisation du simulateur de politiques IAM](#) dans le guide de l'utilisateur d'IAM.

Si la stratégie n'accorde pas à l'utilisateur les autorisations que vous escomptiez, ou si elles sont trop excessives, vous pouvez ajuster la stratégie selon vos besoins. Testez-la à nouveau jusqu'à ce que vous obteniez les résultats souhaités.

Important

La propagation des modifications de la politique peut durer plusieurs minutes avant qu'elles ne prennent effet. Par conséquent, nous vous recommandons de laisser passer au moins cinq minutes avant de tester les mises à jour de vos politiques.

Si un contrôle d'autorisation échoue, la demande retourne un message codé avec les informations de diagnostic. Vous pouvez décoder le message à l'aide de l'action `DecodeAuthorizationMessage`. Pour plus d'informations, reportez-vous [DecodeAuthorizationMessage](#) à la référence de l'AWS Security Token Service API et [decode-authorization-message](#) à la référence des AWS CLI commandes.

Ressource : exemples de politiques pour AWS Batch

Vous pouvez créer des politiques IAM spécifiques pour restreindre les appels et les ressources auxquels les utilisateurs de votre compte ont accès. Vous pouvez ensuite associer ces politiques aux utilisateurs.

Lorsque vous associez une politique à un utilisateur ou à un groupe d'utilisateurs, elle autorise ou refuse aux utilisateurs l'autorisation d'exécuter des tâches spécifiques sur des ressources spécifiques. Pour plus d'informations, consultez la section [Permissions et politiques](#) du guide de l'utilisateur IAM. Pour obtenir des instructions sur la façon de gérer et de créer des politiques IAM personnalisées, consultez la section [Gestion des politiques IAM](#).

Les exemples suivants présentent des déclarations de politique que vous pouvez utiliser pour contrôler les autorisations accordées aux utilisateurs AWS Batch.

Exemples

- [Ressource : Accès en lecture seule pour AWS Batch](#)
- [Ressource : Limiter à l'utilisateur POSIX, à l'image Docker, au niveau de privilège et au rôle lors de la soumission des tâches](#)
- [Ressource : Restreindre au préfixe de définition de tâche lors de la soumission de la tâche](#)
- [Ressource : Limiter à une file d'attente de tâches](#)
- [Refuser l'action lorsque toutes les conditions correspondent à des chaînes](#)
- [Ressource : Refuser l'action lorsque des clés de condition correspondent à des chaînes](#)
- [Ressource : Utiliser la clé de batch:ShareIdentifier condition](#)
- [Gérez les ressources d' SageMaker IA avec AWS Batch](#)
- [Ressource : Restreindre la soumission des tâches par des balises de ressource sur la définition des tâches et dans la file d'attente des tâches](#)

Ressource : Accès en lecture seule pour AWS Batch

La politique suivante autorise les utilisateurs à utiliser toutes les actions d' AWS Batch API dont le nom commence par Describe et List.

À moins qu'une autre déclaration ne leur accorde l'autorisation de le faire, les utilisateurs ne sont pas autorisés à effectuer des actions sur les ressources. Par défaut, ils ne sont pas autorisés à utiliser les actions de l'API.

JSON

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

Ressource : Limiter à l'utilisateur POSIX, à l'image Docker, au niveau de privilège et au rôle lors de la soumission des tâches

La politique suivante permet à un utilisateur POSIX de gérer son propre ensemble de définitions de tâches restreintes.

Utilisez les première et deuxième instructions pour enregistrer et désenregistrer tout nom de définition de tâche dont le nom est préfixé par. *JobDefA_*

La première instruction utilise également des clés de contexte de condition pour restreindre l'utilisateur POSIX, le statut privilégié et les valeurs d'images de conteneur au sein des `containerProperties` d'une définition de tâche. Pour plus d'informations, consultez [RegisterJobDefinition](#) dans la Référence d'API AWS Batch . Dans cet exemple, les définitions de tâches ne peuvent être enregistrées que lorsque l'utilisateur POSIX est défini sur. `nobody` Le drapeau privilégié est défini sur `false`. Enfin, l'image est configurée pour figurer `myImage` dans un référentiel Amazon ECR.

Important

Docker résout le `user` paramètre pour cet utilisateur `uid` depuis l'image du conteneur. Dans la plupart des cas, cela se trouve dans le `/etc/passwd` fichier contenu dans l'image du conteneur. Cette résolution de noms peut être évitée en utilisant des `uid` valeurs directes à la

fois dans la définition de la tâche et dans les politiques IAM associées. Les opérations AWS Batch d'API et les clés conditionnelles `batch:User` IAM prennent en charge les valeurs numériques.

Utilisez la troisième déclaration pour limiter la définition d'une tâche à un rôle spécifique.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "arn:aws:batch:us-east-2:999999999999:job-definition/JobDefA_*"
      ],
      "Condition": {
        "StringEquals": {
          "batch:User": [
            "nobody"
          ],
          "batch:Image": [
            "999999999999.dkr.ecr.us-east-2.amazonaws.com/myImage"
          ]
        },
        "Bool": {
          "batch:Privileged": "false"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "batch:DeregisterJobDefinition"
      ],
      "Resource": [
        "arn:aws:batch:us-east-2:999999999999:job-definition/JobDefA_*"
      ]
    }
  ]
}
```

```

    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::999999999999:role/MyBatchJobRole"
      ]
    }
  ]
}

```

Ressource : Restreindre au préfixe de définition de tâche lors de la soumission de la tâche

Utilisez la politique suivante pour soumettre des tâches à n'importe quelle file d'attente de tâches avec un nom de définition de tâche commençant par *JobDefA*.

Important

Lors de la restriction de l'accès au niveau des ressources pour la soumission de tâche, vous devez fournir des types de ressources de file d'attente de tâches et de définition de tâche.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:us-east-2:111122223333:job-definition/JobDefA_*",
        "arn:aws:batch:us-east-2:111122223333:job-queue/*"
      ]
    }
  ]
}

```

```
    ]
  }
}
```

Ressource : Limiter à une file d'attente de tâches

Utilisez la politique suivante pour soumettre des tâches à une file d'attente de tâches spécifique nommée `queue1` avec n'importe quel nom de définition de tâche.

Important

Lors de la restriction de l'accès au niveau des ressources pour la soumission de tâche, vous devez fournir des types de ressources de file d'attente de tâches et de définition de tâche.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:us-east-2:888888888888:job-definition/*",
        "arn:aws:batch:us-east-2:888888888888:job-queue/queue1"
      ]
    }
  ]
}
```

Refuser l'action lorsque toutes les conditions correspondent à des chaînes

La politique suivante refuse l'accès à l'opération d'[RegisterJobDefinition](#) API lorsque la clé de condition `batch:Image` (ID de l'image du conteneur) est « *string1* » et la clé de condition `batch:LogDriver` (pilote du journal du conteneur) est « *string2* ». AWS Batch évalue les clés de condition sur chaque conteneur. Lorsqu'une tâche couvre plusieurs conteneurs, tels qu'une tâche

parallèle à nœuds multiples, il est possible que les conteneurs aient des configurations différentes. Si plusieurs clés de condition sont évaluées dans une instruction, elles sont combinées de manière AND logique. Ainsi, si l'une des multiples clés de condition ne correspond pas à un conteneur, l'effet `Deny` ne s'applique pas à ce conteneur. Au contraire, un conteneur différent pour la même tâche peut être refusé.

Pour la liste des clés de condition pour AWS Batch, voir [Clés de condition pour AWS Batch](#) dans la référence d'autorisation de service. À l'exception `batch:ShareIdentifier`, toutes les clés de batch condition peuvent être utilisées de cette manière. La clé de `batch:ShareIdentifier` condition est définie pour une tâche, et non pour une définition de tâche.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": "batch:RegisterJobDefinition",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "batch:Image": "string1",
          "batch:LogDriver": "string2"
        }
      }
    }
  ]
}
```


Ressource : Refuser l'action lorsque des clés de condition correspondent à des chaînes

La politique suivante refuse l'accès à l'opération d'[RegisterJobDefinition](#) API lorsque la clé de condition `batch:Image` (ID de l'image du conteneur) est « *string1* » ou que la clé de condition `batch:LogDriver` (pilote du journal du conteneur) est « *string2* ». Lorsqu'une tâche couvre plusieurs conteneurs, tels qu'une tâche parallèle à nœuds multiples, il est possible que les conteneurs aient des configurations différentes. Si plusieurs clés de condition sont évaluées dans une instruction, elles sont combinées de manière AND logique. Ainsi, si l'une des multiples clés de condition ne correspond pas à un conteneur, l'effet `Deny` ne s'applique pas à ce conteneur. Au contraire, un conteneur différent pour la même tâche peut être refusé.

Pour la liste des clés de condition pour AWS Batch, voir [Clés de condition pour AWS Batch](#) dans la référence d'autorisation de service. À l'exception de `batch:ShareIdentifier`, toutes les clés de batch condition peuvent être utilisées de cette manière. (La clé de batch `batch:ShareIdentifier` condition est définie pour une tâche, pas pour une définition de tâche.)

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
```

```

        "StringEquals": {
            "batch:Image": [
                "string1"
            ]
        }
    },
    {
        "Effect": "Deny",
        "Action": [
            "batch:RegisterJobDefinition"
        ],
        "Resource": [
            "*"
        ],
        "Condition": {
            "StringEquals": {
                "batch:LogDriver": [
                    "string2"
                ]
            }
        }
    }
]
}

```

Ressource : Utiliser la clé de **batch:ShareIdentifier** condition

Utilisez la politique suivante pour soumettre des tâches utilisant la définition de jobDefA tâche à la file d'attente des jobqueue1 tâches avec l'identifiant de lowCpu partage.

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "batch:SubmitJob"
            ],

```

```

    "Resource": [
      "arn:aws:batch:us-east-2:555555555555:job-definition/JobDefA",
      "arn:aws:batch:us-east-2:555555555555:job-queue/jobqueue1"
    ],
    "Condition": {
      "StringEquals": {
        "batch:ShareIdentifier": [
          "lowCpu"
        ]
      }
    }
  }
}

```

Gérez les ressources d' SageMaker IA avec AWS Batch

Cette politique permet de AWS Batch gérer les ressources de SageMaker l'IA.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": "arn:aws:iam::*:role/*AWSServiceRoleForAWSBatchWithSagemaker",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "sagemaker-queuing.batch.amazonaws.com"
        }
      }
    }
  ]
}

```

```

    }
  },
  {
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": [
          "sagemaker.amazonaws.com"
        ]
      }
    }
  }
]
}

```

Ressource : Restreindre la soumission des tâches par des balises de ressource sur la définition des tâches et dans la file d'attente des tâches

Utilisez la politique suivante pour soumettre des tâches uniquement lorsque la file d'attente de tâches possède la balise `Environment=dev` et que la définition de tâche possède la balise `Project=calc`. Cette politique explique comment utiliser les balises de ressources pour contrôler l'accès aux AWS Batch ressources lors de la soumission des tâches.

Important

Lorsque vous soumettez des tâches avec des politiques qui évaluent les balises de ressources de définition de tâche, vous devez soumettre des tâches en utilisant le format de révision des définitions de tâches (`job-definition:revision`). Si vous soumettez des tâches sans spécifier de révision, les balises de définition des tâches ne seront pas évaluées, ce qui peut contourner les contrôles d'accès prévus. Le `*:*` modèle de l'ARN de la ressource impose que les soumissions doivent inclure une révision, afin de garantir que les politiques relatives aux balises sont toujours appliquées efficacement.

Cette politique utilise deux instructions distinctes car elle applique des conditions de balise différentes aux différents types de ressources. Lors de la restriction de l'accès au niveau des ressources pour

la soumission de tâche, vous devez fournir des types de ressources de file d'attente de tâches et de définition de tâche.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "batch:SubmitJob",
      "Resource": "arn:aws:batch:*:*:job-queue/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Environment": "dev"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "batch:SubmitJob",
      "Resource": "arn:aws:batch:*:*:job-definition/*:*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "calc"
        }
      }
    }
  ]
}
```

Ressource : politique AWS Batch gérée

AWS Batch fournit une politique gérée que vous pouvez associer aux utilisateurs. Cette politique autorise l'utilisation des AWS Batch ressources et des opérations d'API. Vous pouvez appliquer cette stratégie directement ou l'utiliser comme point de départ pour créer vos propres stratégies. Pour plus d'informations sur chaque opération d'API mentionnée dans ces politiques, consultez la section [Actions](#) de la référence AWS Batch d'API.

AWSBatchFullAccess

Cette politique autorise l'accès complet de l'administrateur à AWS Batch.

Pour consulter le JSON de la politique, consultez [AWSBatchFullAccess](#) le [Guide de référence des politiques AWS gérées](#).

AWS Batch Rôle d'exécution IAM

Le rôle d'exécution accorde au conteneur Amazon ECS et aux AWS Fargate agents l'autorisation d'effectuer des appels d' AWS API en votre nom.

Note

Le rôle d'exécution est pris en charge par l'agent de conteneur Amazon ECS version 1.16.0 et versions ultérieures.

Le rôle d'exécution IAM est requis en fonction des exigences de votre tâche. Vous pouvez avoir plusieurs rôles d'exécution pour différents objectifs et services associés à votre compte.

Note

Pour plus d'informations sur le rôle d'instance Amazon ECS, consultez [Rôle d'instance Amazon ECS](#). Pour plus d'informations sur les rôles de service, consultez [Comment AWS Batch fonctionne avec IAM](#).

Amazon ECS fournit la politique AmazonECSTaskExecutionRolePolicy gérée. Cette politique contient les autorisations requises pour les cas d'utilisation courants décrits ci-dessus. Il peut être nécessaire d'ajouter des politiques intégrées à votre rôle d'exécution pour les cas d'utilisation particuliers décrits ci-dessous.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
```

```
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
    ],
    "Resource": "*"
}
]
```

Autorisations prises en charge au niveau des ressources pour les actions d'API AWS Batch

Le terme autorisations au niveau des ressources fait référence à la capacité de spécifier les ressources sur lesquelles les utilisateurs sont autorisés à effectuer des actions. AWS Batch prend partiellement en charge les autorisations au niveau des ressources. Pour certaines AWS Batch actions, vous pouvez contrôler le moment où les utilisateurs sont autorisés à utiliser ces actions en fonction des conditions qui doivent être remplies. Vous pouvez également contrôler en fonction des ressources spécifiques que les utilisateurs sont autorisés à utiliser. Par exemple, vous pouvez autoriser les utilisateurs à soumettre des tâches, mais uniquement à une file d'attente spécifique et uniquement avec une définition de tâche spécifique.

Pour plus de détails sur les actions et les types de ressources définis par AWS Batch, y compris le ARNs format de chaque type de ressource, voir Actions, ressources et clés de condition [AWS Batch](#) dans la référence d'autorisation de service.

Tutoriel : Création du rôle d'exécution IAM

Si votre compte ne possède pas encore de rôle d'exécution IAM, suivez les étapes ci-dessous pour créer le rôle.

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le panneau de navigation, choisissez Rôles.
3. Sélectionnez Create role (Créer un rôle).
4. Pour Type d'entité de confiance, sélectionnez Service AWS.
5. Pour Service ou cas d'utilisation, choisissez Elastic Container Service. Choisissez ensuite à nouveau Elastic Container Service Task.

6. Choisissez Suivant.
7. Pour les politiques relatives aux autorisations, recherchez Amazon ECSTask ExecutionRolePolicy.
8. Cochez la case située à gauche de la ECSTask ExecutionRolePolicy politique d'Amazon, puis choisissez Next.
9. Dans Nom du rôle, entrez `ecsTaskExecutionRole` puis choisissez Créer un rôle.

Tutoriel : Vérifiez le rôle d'exécution IAM

Utilisez la procédure suivante pour vérifier que votre compte possède déjà le rôle d'exécution IAM et associez la politique IAM gérée, si nécessaire.

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le panneau de navigation, choisissez Rôles.
3. Recherchez dans la liste des rôles `ecsTaskExecutionRole`. Si vous ne trouvez pas le rôle, consultez [Tutoriel : Création du rôle d'exécution IAM](#). Si vous avez trouvé le rôle, choisissez-le pour afficher les politiques associées.
4. Dans l'onglet Autorisations, vérifiez que la politique ECSTask ExecutionRolePolicy gérée par Amazon est attachée au rôle. Si la politique est attachée, votre rôle d'exécution est correctement configuré. Sinon, suivez les sous-étapes ci-dessous pour attacher la politique.
 - a. Choisissez Ajouter des autorisations, puis choisissez Joindre des politiques.
 - b. Recherchez Amazon ECSTask ExecutionRolePolicy.
 - c. Cochez la case située à gauche de la ECSTask ExecutionRolePolicy politique d'Amazon et choisissez Joindre les politiques.
5. Choisissez Trust Relationships (Relations d'approbation).
6. Vérifiez que la relation d'approbation contient la politique suivante. Si la relation de confiance correspond à la politique ci-dessous, le rôle est correctement configuré. Si la relation d'approbation ne correspond pas, choisissez Modifier la politique de confiance, entrez ce qui suit, puis choisissez Mettre à jour la politique.

JSON

```
{
```



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

Utilisation des rôles liés à un service pour AWS Batch

AWS Batch utilise des AWS Identity and Access Management rôles liés à un [service](#) (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié à. AWS Batch Les rôles liés au service sont prédéfinis par AWS Batch et incluent toutes les autorisations dont le service a besoin pour appeler d'autres AWS services en votre nom.

AWS Batch utilise deux rôles différents liés à un service :

- [AWSServiceRoleForBatch](#)- Pour les AWS Batch opérations, y compris les environnements informatiques.
- [AWSServiceRoleForAWSBatchWithSagemaker](#)- Pour la gestion de la charge de travail liée à l' SageMaker IA et la mise en file d'attente.

Rubriques

- [Utilisation de rôles pour AWS Batch](#)
- [Utiliser les rôles pour utiliser AWS Batch l' SageMaker IA](#)

Utilisation de rôles pour AWS Batch

AWS Batch utilise des AWS Identity and Access Management rôles liés à un [service](#) (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié à. AWS Batch Les rôles liés au service sont prédéfinis par AWS Batch 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 AWS Batch car vous n'avez pas à ajouter manuellement les autorisations nécessaires. AWS Batch définit les autorisations associées à ses rôles liés aux services et, sauf indication contraire, seul AWS Batch peut assumer ses rôles. Les autorisations définies comprennent la politique de confiance et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

 Note

Procédez de l'une des manières suivantes pour spécifier un rôle de service pour un environnement AWS Batch informatique.

- Utilisez une chaîne vide pour le rôle de service. Cela permet de AWS Batch créer le rôle de service.
- Spécifiez le rôle de service au format suivant :`arn:aws:iam::account_number:role/aws-service-role/batch.amazonaws.com/AWSServiceRoleForBatch`.

Pour plus d'informations, consultez [Nom de rôle ou ARN incorrect](#) le guide de AWS Batch l'utilisateur.

Vous pouvez supprimer un rôle lié à un service uniquement après la suppression préalable de ses ressources connexes. Cela protège vos AWS Batch ressources car vous ne pouvez pas supprimer par inadvertance l'autorisation d'accès aux ressources.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés à un service, consultez la section [AWS Services qui fonctionnent avec IAM](#) et recherchez les services dont la valeur est Oui dans la colonne 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 AWS Batch

AWS Batch utilise le rôle lié au service nommé `AWSServiceRoleForBatch`— Permet de AWS Batch créer et de gérer des AWS ressources en votre nom.

Le rôle `AWSService RoleForBatch` lié à un service fait confiance aux services suivants pour assumer le rôle :

- `batch.amazonaws.com`

La politique d'autorisations de rôle nommée [BatchServiceRolePolicy](#) AWS Batch permet d'effectuer les actions suivantes sur les ressources spécifiées :

- **autoscaling**— Permet de AWS Batch créer et de gérer des ressources Amazon EC2 Auto Scaling. AWS Batch crée et gère des groupes Amazon EC2 Auto Scaling pour la plupart des environnements informatiques.
- **ec2**— Permet AWS Batch de contrôler le cycle de vie des instances Amazon EC2 ainsi que de créer et de gérer des modèles et des balises de lancement. AWS Batch crée et gère les demandes EC2 Spot Fleet pour certains environnements informatiques EC2 Spot.
- **ecs**- Permet AWS Batch de créer et de gérer des clusters Amazon ECS, des définitions de tâches et des tâches pour l'exécution des tâches.
- **eks**- Permet AWS Batch de décrire la ressource du cluster Amazon EKS pour les validations.
- **iam**- Permet AWS Batch de valider et de transmettre les rôles fournis par le propriétaire à Amazon EC2, Amazon EC2 Auto Scaling et Amazon ECS.
- **logs**— Permet AWS Batch de créer et de gérer des groupes de journaux et des flux de journaux pour les AWS Batch tâches.

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 AWS Batch

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous créez un environnement informatique dans le AWS Management Console, le ou l' AWS API AWS CLI, vous AWS Batch 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. Si vous utilisiez le AWS Batch service avant le 10 mars 2021, date à laquelle il a commencé à prendre en charge les rôles liés au service, vous avez AWS Batch créé le AWSService RoleForBatch rôle dans votre compte. Pour en savoir plus, voir [Un nouveau rôle est apparu dans mon 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 environnement informatique, il AWS Batch crée à nouveau le rôle lié au service pour vous.

Modification d'un rôle lié à un service pour AWS Batch

AWS Batch ne vous permet pas de modifier le rôle `AWSService RoleForBatch` lié au service. Après avoir 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 plus d'informations, consultez [Modification d'un rôle lié à un service](#) dans le Guide de l'utilisateur IAM.

Pour autoriser une entité IAM à modifier la description du rôle lié à un `AWSService RoleForBatch` service

Ajoutez la déclaration suivante à la politique d'autorisation. Cela permet à l'entité IAM de modifier la description d'un rôle lié à un service.

```
{
  "Effect": "Allow",
  "Action": [
    "iam:UpdateRoleDescription"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/batch.amazonaws.com/
AWSServiceRoleForBatch",
  "Condition": {"StringLike": {"iam:AWSServiceName": "batch.amazonaws.com"}}
}
```

Suppression d'un rôle lié à un service pour AWS Batch

Si vous n'avez plus besoin d'utiliser une fonction ou un service qui nécessite un rôle lié à un service, nous vous recommandons de supprimer ce rôle. De cette façon, vous n'avez aucune entité inutilisée qui n'est pas surveillée ou gérée activement. Cependant, vous devez nettoyer votre rôle lié à un service avant de pouvoir le supprimer manuellement.

Pour autoriser une entité IAM à supprimer le rôle lié à un `AWSService RoleForBatch` service

Ajoutez la déclaration suivante à la politique d'autorisation. Cela permet à l'entité IAM de supprimer un rôle lié à un service.

```
{
  "Effect": "Allow",
```

```
"Action": [
    "iam:DeleteServiceLinkedRole",
    "iam:GetServiceLinkedRoleDeletionStatus"
],
"Resource": "arn:aws:iam::*:role/aws-service-role/batch.amazonaws.com/
AWSServiceRoleForBatch",
"Condition": {"StringLike": {"iam:AWSServiceName": "batch.amazonaws.com"}}
}
```

Nettoyage d'un rôle lié à un service

Avant de pouvoir utiliser IAM pour supprimer un rôle lié à un service, vous devez d'abord confirmer que le rôle n'a aucune session active et supprimer tous les environnements AWS Batch informatiques qui utilisent le rôle dans toutes les AWS régions sur une seule partition.

Pour vérifier si une session est active pour le rôle lié à un service

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le volet de navigation, choisissez Rôles, puis le AWSService RoleForBatch nom (et non la case à cocher).
3. Sur la page Résumé, choisissez Access Advisor et consultez l'activité récente pour le rôle lié à un service.

Note

Si vous ne savez pas si le AWSService RoleForBatch rôle AWS Batch est utilisé, vous pouvez essayer de le supprimer. Si le service utilise le rôle, le rôle ne sera pas supprimé. Vous pouvez consulter les régions dans lesquelles le rôle est utilisé. Si le rôle est utilisé, vous devez attendre que la session se termine avant de pouvoir le supprimer. Vous ne pouvez pas révoquer la session d'un rôle lié à un service.

Pour supprimer les AWS Batch ressources utilisées par le rôle lié à AWSService RoleForBatch un service

Vous devez supprimer tous les environnements AWS Batch informatiques qui utilisent le AWSService RoleForBatch rôle dans toutes les AWS régions avant de pouvoir supprimer le AWSService RoleForBatch rôle.

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.

2. Dans la barre de navigation, sélectionnez la région à utiliser.
3. Dans le panneau de navigation, choisissez Environnements de calcul.
4. Sélectionnez l'environnement informatique.
5. Choisissez Désactiver. Attendez que l'état passe à DÉSACTIVÉ.
6. Sélectionnez l'environnement informatique.
7. Sélectionnez Delete (Supprimer). Confirmez que vous souhaitez supprimer l'environnement de calcul en choisissant Supprimer l'environnement de calcul.
8. Répétez les étapes 1 à 7 pour tous les environnements informatiques qui utilisent le rôle lié à un service dans toutes les régions.

Supprimer un rôle lié à un service dans IAM (console)

Vous pouvez utiliser la console IAM pour supprimer un rôle lié à un service.

Pour supprimer un rôle lié à un service (console)

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à <https://console.aws.amazon.com/iam/> l'adresse.
2. Dans le panneau de navigation de la console IAM, choisissez Rôles. Sélectionnez ensuite la case à cocher située à côté AWSServiceRoleForBatch, et non le nom ou la ligne elle-même.
3. Choisissez Delete role (Supprimer le rôle).
4. Dans la boîte de dialogue de confirmation, vérifiez les dernières données consultées dans le service. Elles indiquent quels rôles, parmi ceux sélectionnés, ont accédé en dernier à un service Service AWS. Cela vous permet de confirmer si le rôle est actif actuellement. Si vous souhaitez continuer, sélectionnez Oui, supprimer pour lancer la tâche de suppression du rôle.
5. Consultez les notifications de la console IAM pour surveiller la progression de la suppression du rôle lié à un service. Dans la mesure où la suppression du rôle lié à un service IAM est asynchrone, une fois que vous soumettez le rôle afin qu'il soit supprimé, la suppression peut réussir ou échouer.
 - Si la tâche réussit, le rôle est supprimé de la liste et une notification de succès s'affiche en haut de la page.
 - Si la tâche échoue, vous pouvez choisir View details (Afficher les détails) ou View Resources (Afficher les ressources) à partir des notifications pour connaître le motif de l'échec de la suppression. Si la suppression échoue car le rôle utilise les ressources du service, alors

la notification comprend une liste de ressources, à condition que le service renvoie ces informations. Vous pouvez alors [nettoyer les ressources](#) et lancer à nouveau la tâche de suppression.

 Note

Vous devrez peut-être répéter ce processus plusieurs fois, en fonction des informations renvoyées par le service. Par exemple, il est possible que votre rôle lié à un service utilise six ressources et que votre service renvoie des informations sur cinq d'entre elles. Si vous nettoyez les cinq ressources et lancez à nouveau la tâche de suppression pour le rôle, la suppression échoue et le service indique la ressource restante. Un service peut renvoyer toutes les ressources, quelques ressources ou n'indiquer aucune ressource.

- Si la tâche échoue et que la notification n'inclut pas de liste des ressources, le service peut ne pas renvoyer cette information. Pour savoir comment nettoyer les ressources pour ce service, veuillez consulter [Services Services AWS fonctionnant avec IAM](#). Identifiez votre service dans le tableau, puis choisissez le lien Yes (Oui) pour afficher la documentation relative au rôle lié à un service pour ce service.

Supprimer un rôle lié à un service dans IAM (AWS CLI)

Vous pouvez utiliser les commandes IAM depuis le AWS Command Line Interface pour supprimer un rôle lié à un service.

Pour supprimer un rôle lié à un service (CLI)

1. Un rôle lié à un service ne pouvant pas être supprimé s'il est utilisé ou si des ressources lui sont associées, vous devez envoyer une demande de suppression. Cette demande peut être refusée si ces conditions ne sont pas remplies. Vous devez capturer le `deletion-task-id` de la réponse afin de vérifier l'état de la tâche de suppression. Saisissez la commande suivante pour envoyer une demande de suppression d'un rôle lié à un service :

```
$ aws iam delete-service-linked-role --role-name AWSServiceRoleForBatch
```

2. Saisissez la commande suivante pour vérifier l'état de la tâche de suppression :

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

L'état de la tâche de suppression peut être NOT_STARTED, IN_PROGRESS, SUCCEEDED ou FAILED. Si la suppression échoue, l'appel renvoie le motif de l'échec, afin que vous puissiez apporter une solution. Si la suppression échoue car le rôle utilise les ressources du service, alors la notification comprend une liste de ressources, à condition que le service renvoie ces informations. Vous pouvez alors [nettoyer les ressources](#) et lancer à nouveau la tâche de suppression.

 Note

Vous devrez peut-être répéter ce processus plusieurs fois, en fonction des informations renvoyées par le service. Par exemple, il est possible que votre rôle lié à un service utilise six ressources et que votre service renvoie des informations sur cinq d'entre elles. Si vous nettoyez les cinq ressources et lancez à nouveau la tâche de suppression pour le rôle, la suppression échoue et le service indique la ressource restante. Un service peut renvoyer toutes les ressources, certaines d'entre elles. Ou bien, il se peut qu'aucune ressource ne soit signalée. Pour savoir comment nettoyer les ressources d'un service qui ne signale aucune ressource, consultez la section [AWS Services qui fonctionnent avec IAM](#). Identifiez votre service dans le tableau, puis choisissez le lien Yes (Oui) pour afficher la documentation relative au rôle lié à un service pour ce service.

Supprimer un rôle lié à un service dans IAM (API)AWS

Vous pouvez utiliser l'API IAM pour supprimer un rôle lié à un service.

Pour supprimer un rôle lié à un service (API)

1. Pour envoyer une demande de suppression pour un rôle lié à un service, appelez [DeleteServiceLinkedRole](#). Dans la demande, spécifiez le nom du AWSService RoleForBatch rôle.

Dans la mesure où un rôle lié à un service ne peut pas être supprimé s'il est utilisé ou si des ressources lui sont associées, vous devez envoyer une demande de suppression. Cette

demande peut être refusée si ces conditions ne sont pas satisfaites. Vous devez capturer le `DeletionTaskId` de la réponse afin de vérifier l'état de la tâche de suppression.

2. Pour vérifier l'état de la suppression, appelez [GetServiceLinkedRoleDeletionStatus](#). Dans la demande, spécifiez le `DeletionTaskId`.

L'état de la tâche de suppression peut être `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` ou `FAILED`. Si la suppression échoue, l'appel renvoie le motif de l'échec, afin que vous puissiez apporter une solution. Si la suppression échoue car le rôle utilise les ressources du service, alors la notification comprend une liste de ressources, à condition que le service renvoie ces informations. Vous pouvez alors [nettoyer les ressources](#) et lancer à nouveau la tâche de suppression.

Note

Vous devrez peut-être répéter ce processus plusieurs fois, en fonction des informations renvoyées par le service. Par exemple, il est possible que votre rôle lié à un service utilise six ressources et que votre service renvoie des informations sur cinq d'entre elles. Si vous nettoyez les cinq ressources et lancez à nouveau la tâche de suppression pour le rôle, la suppression échoue et le service indique la ressource restante. Un service peut renvoyer toutes les ressources, quelques ressources ou n'indiquer aucune ressource. Pour apprendre à nettoyer les ressources pour un service qui n'indique aucune ressource, veuillez consulter [Services Services AWS fonctionnant avec IAM](#). Identifiez votre service dans le tableau, puis choisissez le lien Yes (Oui) pour afficher la documentation relative au rôle lié à un service pour ce service.

Régions prises en charge pour les rôles liés à un service AWS Batch

AWS Batch prend en charge l'utilisation de rôles liés au service dans toutes les régions où le service est disponible. Pour plus d'informations, consultez [Points de terminaison AWS Batch](#).

Utiliser les rôles pour utiliser AWS Batch l' SageMaker IA

AWS Batch utilise des AWS Identity and Access Management rôles liés à un [service](#) (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié à. AWS Batch Les rôles liés au service sont prédéfinis par AWS Batch 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 AWS Batch car vous n'avez pas à ajouter manuellement les autorisations nécessaires. AWS Batch définit les autorisations associées à ses rôles liés aux services et, sauf indication contraire, seul AWS Batch peut assumer ses rôles. Les autorisations définies comprennent la politique de confiance 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 AWS Batch ressources car vous ne pouvez pas supprimer par inadvertance l'autorisation d'accès aux ressources.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés à un service, consultez la section [AWS Services qui fonctionnent avec IAM](#) et recherchez les services dont la valeur est Oui dans la colonne 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 AWS Batch

AWS Batch utilise le rôle lié au service nommé `AWSServiceRoleForAWSBatchWithSageMaker`— Permet de mettre en file AWS Batch d'attente et de gérer les tâches de SageMaker formation en votre nom.

Le rôle `AWSService RoleFor AWSBatch WithSageMaker` lié à un service fait confiance aux services suivants pour assumer le rôle :

- `sagemaker-queuing.batch.amazonaws.com`

La politique d'autorisation des rôles AWS Batch permet d'effectuer les actions suivantes sur les ressources spécifiées :

- `sagemaker`— Permet AWS Batch de gérer les tâches de SageMaker formation, de transformer les tâches et d'autres ressources d' SageMaker IA.
- `iam:PassRole`— Permet de AWS Batch transmettre des rôles d'exécution définis par le client à l' SageMaker IA pour l'exécution des tâches. La contrainte de ressources permet de transmettre des rôles aux services d' SageMaker IA.

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 AWS Batch

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous créez un environnement de service `CreateServiceEnvironment` à l'aide de AWS Management Console, de AWS CLI, ou de l' AWS API, vous AWS Batch créez le rôle lié au service pour vous.

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 environnement de service à l'aide de `CreateServiceEnvironment`, AWS Batch crée à nouveau le rôle lié au service pour vous.

Pour consulter le JSON de la politique, consultez [AWSBatchServiceRolePolicyForSageMaker](#) le [Guide de référence des politiques AWS gérées](#).

Modification d'un rôle lié à un service pour AWS Batch

AWS Batch ne vous permet pas de modifier le rôle `AWSService RoleFor AWSBatch WithSagemaker` lié au service. Après avoir 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 plus d'informations, consultez [Modification d'un rôle lié à un service](#) dans le Guide de l'utilisateur IAM.

Suppression d'un rôle lié à un service pour AWS Batch

Si vous n'avez plus besoin d'utiliser une fonction ou un service qui nécessite un rôle lié à un service, nous vous recommandons de supprimer ce rôle. De cette façon, vous n'avez aucune entité inutilisée qui n'est pas surveillée ou gérée activement. Cependant, vous devez nettoyer votre rôle lié à un service avant de pouvoir le supprimer manuellement.

Nettoyage d'un rôle lié à un service

Avant de pouvoir utiliser IAM pour supprimer un rôle lié à un service, vous devez d'abord confirmer que le rôle n'a aucune session active et supprimer tous les environnements de service qui utilisent le rôle dans toutes les AWS régions sur une seule partition.

Pour vérifier si une session est active pour le rôle lié à un service

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le volet de navigation, choisissez Rôles, puis le `AWSService RoleFor AWSBatch WithSagemaker` nom (pas la case à cocher).

3. Sur la page Résumé, choisissez Access Advisor et consultez l'activité récente pour le rôle lié à un service.

 Note

Si vous ne savez pas si le AWSService RoleFor AWSBatch WithSagemaker rôle AWS Batch est utilisé, vous pouvez essayer de le supprimer. Si le service utilise le rôle, le rôle ne sera pas supprimé. Vous pouvez consulter les régions dans lesquelles le rôle est utilisé. Si le rôle est utilisé, vous devez attendre que la session se termine avant de pouvoir le supprimer. Vous ne pouvez pas révoquer la session d'un rôle lié à un service.

Pour supprimer les AWS Batch ressources utilisées par le rôle lié à AWSService RoleFor AWSBatch WithSagemaker un service

Vous devez dissocier toutes les files d'attente de tâches de tous les environnements de service, puis vous devez supprimer tous les environnements de service qui utilisent le AWSService RoleFor AWSBatch WithSagemaker rôle dans toutes les AWS régions avant de pouvoir supprimer le AWSService RoleFor AWSBatch WithSagemaker rôle.

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation, sélectionnez la région à utiliser.
3. Dans le volet de navigation, choisissez Environments, puis Service environments.
4. Sélectionnez tous les environnements de service.
5. Choisissez Désactiver. Attendez que l'état passe à DÉSACTIVÉ.
6. Sélectionnez l'environnement de service.
7. Sélectionnez Delete (Supprimer). Confirmez que vous souhaitez supprimer l'environnement de service en choisissant Supprimer l'environnement de service.
8. Répétez les étapes 1 à 7 pour tous les environnements de service qui utilisent le rôle lié au service dans toutes les régions.

Supprimer un rôle lié à un service dans IAM (console)

Vous pouvez utiliser la console IAM pour supprimer un rôle lié à un service.

Pour supprimer un rôle lié à un service (console)

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à <https://console.aws.amazon.com/iam/> l'adresse.
2. Dans le panneau de navigation de la console IAM, choisissez Rôles. Sélectionnez ensuite la case à cocher située à côté AWSService RoleFor AWSBatchWithSagemaker, et non le nom ou la ligne elle-même.
3. Choisissez Delete role (Supprimer le rôle).
4. Dans la boîte de dialogue de confirmation, vérifiez les dernières données consultées dans le service. Elles indiquent quels rôles, parmi ceux sélectionnés, ont accédé en dernier à un service Service AWS. Cela vous permet de confirmer si le rôle est actif actuellement. Si vous souhaitez continuer, sélectionnez Oui, supprimer pour lancer la tâche de suppression du rôle.
5. Consultez les notifications de la console IAM pour surveiller la progression de la suppression du rôle lié à un service. Dans la mesure où la suppression du rôle lié à un service IAM est asynchrone, une fois que vous soumettez le rôle afin qu'il soit supprimé, la suppression peut réussir ou échouer.
 - Si la tâche réussit, le rôle est supprimé de la liste et une notification de succès s'affiche en haut de la page.
 - Si la tâche échoue, vous pouvez choisir View details (Afficher les détails) ou View Resources (Afficher les ressources) à partir des notifications pour connaître le motif de l'échec de la suppression. Si la suppression échoue car le rôle utilise les ressources du service, alors la notification comprend une liste de ressources, à condition que le service renvoie ces informations. Vous pouvez alors [nettoyer les ressources](#) et lancer à nouveau la tâche de suppression.

Note

Vous devrez peut-être répéter ce processus plusieurs fois, en fonction des informations renvoyées par le service. Par exemple, il est possible que votre rôle lié à un service utilise six ressources et que votre service renvoie des informations sur cinq d'entre elles. Si vous nettoyez les cinq ressources et lancez à nouveau la tâche de suppression pour le rôle, la suppression échoue et le service indique la ressource restante. Un service peut renvoyer toutes les ressources, quelques ressources ou n'indiquer aucune ressource.

- Si la tâche échoue et que la notification n'inclut pas de liste des ressources, le service peut ne pas renvoyer cette information. Pour savoir comment nettoyer les ressources pour ce service, veuillez consulter [Services Services AWS fonctionnant avec IAM](#). Identifiez votre service dans le tableau, puis choisissez le lien Yes (Oui) pour afficher la documentation relative au rôle lié à un service pour ce service.

Supprimer un rôle lié à un service dans IAM (AWS CLI)

Vous pouvez utiliser les commandes IAM depuis le AWS Command Line Interface pour supprimer un rôle lié à un service.

Pour supprimer un rôle lié à un service (CLI)

1. Un rôle lié à un service ne pouvant pas être supprimé s'il est utilisé ou si des ressources lui sont associées, vous devez envoyer une demande de suppression. Cette demande peut être refusée si ces conditions ne sont pas remplies. Vous devez capturer le `deletion-task-id` de la réponse afin de vérifier l'état de la tâche de suppression. Saisissez la commande suivante pour envoyer une demande de suppression d'un rôle lié à un service :

```
$ aws iam delete-service-linked-role --role-name  
AWSServiceRoleForAWSBatchWithSagemaker
```

2. Saisissez la commande suivante pour vérifier l'état de la tâche de suppression :

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-  
id
```

L'état de la tâche de suppression peut être NOT_STARTED, IN_PROGRESS, SUCCEEDED ou FAILED. Si la suppression échoue, l'appel renvoie le motif de l'échec, afin que vous puissiez apporter une solution. Si la suppression échoue car le rôle utilise les ressources du service, alors la notification comprend une liste de ressources, à condition que le service renvoie ces informations. Vous pouvez alors [nettoyer les ressources](#) et lancer à nouveau la tâche de suppression.

Note

Vous devrez peut-être répéter ce processus plusieurs fois, en fonction des informations renvoyées par le service. Par exemple, il est possible que votre rôle lié à un service

utilise six ressources et que votre service renvoie des informations sur cinq d'entre elles. Si vous nettoyez les cinq ressources et lancez à nouveau la tâche de suppression pour le rôle, la suppression échoue et le service indique la ressource restante. Un service peut renvoyer toutes les ressources, certaines d'entre elles. Ou bien, il se peut qu'aucune ressource ne soit signalée. Pour savoir comment nettoyer les ressources d'un service qui ne signale aucune ressource, consultez la section [AWS Services qui fonctionnent avec IAM](#). Identifiez votre service dans le tableau, puis choisissez le lien Yes (Oui) pour afficher la documentation relative au rôle lié à un service pour ce service.

Supprimer un rôle lié à un service dans IAM (API)AWS

Vous pouvez utiliser l'API IAM pour supprimer un rôle lié à un service.

Pour supprimer un rôle lié à un service (API)

1. Pour envoyer une demande de suppression pour un rôle lié à un service, appelez [DeleteServiceLinkedRole](#). Dans la demande, spécifiez le nom du AWSService RoleForAWSBatch WithSagemaker rôle.

Dans la mesure où un rôle lié à un service ne peut pas être supprimé s'il est utilisé ou si des ressources lui sont associées, vous devez envoyer une demande de suppression. Cette demande peut être refusée si ces conditions ne sont pas satisfaites. Vous devez capturer le DeletionTaskId de la réponse afin de vérifier l'état de la tâche de suppression.

2. Pour vérifier l'état de la suppression, appelez [GetServiceLinkedRoleDeletionStatus](#). Dans la demande, spécifiez le DeletionTaskId.

L'état de la tâche de suppression peut être NOT_STARTED, IN_PROGRESS, SUCCEEDED ou FAILED. Si la suppression échoue, l'appel renvoie le motif de l'échec, afin que vous puissiez apporter une solution. Si la suppression échoue car le rôle utilise les ressources du service, alors la notification comprend une liste de ressources, à condition que le service renvoie ces informations. Vous pouvez alors [nettoyer les ressources](#) et lancer à nouveau la tâche de suppression.

Note

Vous devrez peut-être répéter ce processus plusieurs fois, en fonction des informations renvoyées par le service. Par exemple, il est possible que votre rôle lié à un service

utilise six ressources et que votre service renvoie des informations sur cinq d'entre elles. Si vous nettoyez les cinq ressources et lancez à nouveau la tâche de suppression pour le rôle, la suppression échoue et le service indique la ressource restante. Un service peut renvoyer toutes les ressources, quelques ressources ou n'indiquer aucune ressource. Pour apprendre à nettoyer les ressources pour un service qui n'indique aucune ressource, veuillez consulter [Services Services AWS fonctionnant avec IAM](#). Identifiez votre service dans le tableau, puis choisissez le lien Yes (Oui) pour afficher la documentation relative au rôle lié à un service pour ce service.

Régions prises en charge pour les rôles liés à un service AWS Batch

AWS Batch prend en charge l'utilisation de rôles liés au service dans toutes les régions où le service est disponible. Pour plus d'informations, consultez [Points de terminaison AWS Batch](#).

Rôle d'instance Amazon ECS

AWS Batch les environnements de calcul sont alimentés par des instances de conteneur Amazon ECS. Ils exécutent l'agent de conteneur Amazon ECS localement. L'agent de conteneur Amazon ECS appelle diverses opérations AWS d'API en votre nom. Par conséquent, les instances de conteneur qui exécutent l'agent nécessitent une politique et un rôle IAM pour que ces services reconnaissent que l'agent vous appartient. Vous devez créer un rôle IAM et un profil d'instance pour les instances de conteneur à utiliser lors de leur lancement. Sinon, vous ne pouvez pas créer d'environnement de calcul et y lancer des instances de conteneur. Cette exigence s'applique aux instances de conteneur lancées avec ou sans l'AMI optimisée Amazon ECS fournie par Amazon. Pour plus d'informations, consultez [Rôle d'instance Amazon ECS](#) dans le Guide du développeur Amazon Elastic Container Service

Rubriques

- [Vérifiez le rôle d'instance Amazon ECS de votre compte](#)

Vérifiez le rôle d'instance Amazon ECS de votre compte

Le rôle et le profil d'instance Amazon ECS sont automatiquement créés pour vous lors de la première utilisation de la console. Toutefois, vous pouvez suivre ces étapes pour vérifier si votre compte possède déjà le rôle et le profil d'instance Amazon ECS. Les étapes suivantes expliquent également comment associer la politique IAM gérée.

Tutoriel : Vérifiez la présence du **ecsInstanceRole** dans la console IAM

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le panneau de navigation, choisissez Rôles.
3. Recherchez dans la liste des rôles `ecsInstanceRole`. Si le rôle n'existe pas, procédez comme suit pour le créer.
 - a. Choisissez Create Role (Créer un rôle).
 - b. Pour Trusted entity (Entité de confiance), choisissez Service AWS.
 - c. Pour les cas d'utilisation courants, choisissez EC2.
 - d. Choisissez Suivant.
 - e. Pour les politiques d'autorisations, recherchez Amazon EC2 ContainerServicefor EC2 Role.
 - f. Cochez la case à côté d'Amazon EC2 ContainerServicefor EC2 Role, puis choisissez Next.
 - g. Sous Nom du rôle, tapez `ecsInstanceRole`, puis choisissez Créer un rôle.

Note

Si vous utilisez le AWS Management Console pour créer un rôle pour Amazon EC2, la console crée un profil d'instance portant le même nom que le rôle.

Vous pouvez également utiliser le AWS CLI pour créer le rôle `ecsInstanceRole` IAM. L'exemple suivant crée un rôle IAM avec une politique de confiance et une politique AWS gérée.

Tutoriel : Création d'un rôle IAM et d'un profil d'instance ()AWS CLI

1. Créez la politique de confiance suivante et enregistrez-la dans un fichier texte nommé `ecsInstanceRole-role-trust-policy.json`.

JSON

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

```

        "Action": "sts:AssumeRole"
      }
    ]
  }

```

2. Utilisez la commande [create-role](#) pour créer le `ecsInstanceRole` rôle. Spécifiez l'emplacement du fichier de politique de confiance dans le `assume-role-policy-document` paramètre.

```

$ aws iam create-role \
  --role-name ecsInstanceRole \
  --assume-role-policy-document file://ecsInstanceRole-role-trust-policy.json

```

3. Utilisez la [create-instance-profile](#) commande pour créer un profil d'instance nommé `ecsInstanceRole`.

Note

Vous devez créer des rôles et des profils d'instance en tant qu'actions distinctes dans l'AWS API AWS CLI and.

```

$ aws iam create-instance-profile --instance-profile-name ecsInstanceRole

```

Voici un exemple de réponse.

```

{
  "InstanceProfile": {
    "Path": "/",
    "InstanceProfileName": "ecsInstanceRole",
    "InstanceProfileId": "AIPAT46P5RDITREXAMPLE",
    "Arn": "arn:aws:iam::123456789012:instance-profile/ecsInstanceRole",
    "CreateDate": "2022-06-30T23:53:34.093Z",
    "Roles": [],
  }
}

```

4. Utilisez la commande [add-role-to-instance-profile](#) pour ajouter le `ecsInstanceRole` rôle au profil de l'`ecsInstanceRole` instance.

```

aws iam add-role-to-instance-profile \
  --role-name ecsInstanceRole --instance-profile-name ecsInstanceRole

```

5. Utilisez la [attach-role-policy](#) commande pour associer la politique AmazonEC2ContainerServiceforEC2Role AWS gérée au ecsInstanceRole rôle.

```
$ aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role \
    --role-name ecsInstanceRole
```

Rôle de la flotte Amazon EC2 Spot

Si vous créez un environnement informatique géré qui utilise des instances Amazon EC2 Spot Fleet, vous devez créer la AmazonEC2SpotFleetTaggingRole politique. Cette politique autorise Spot Fleet à lancer, étiqueter et résilier des instances en votre nom. Spécifiez le rôle dans votre demande de parc d'instances Ponctuelles. Vous devez également avoir les rôles AWSServiceRoleForEC2Spot et AWSServiceRoleForEC2SpotFleet liés aux services pour Amazon EC2 Spot et Spot Fleet. Suivez les instructions ci-dessous pour créer tous ces rôles. Pour plus d'informations, consultez les sections [Utilisation de rôles liés à un service](#) et [Création d'un rôle pour déléguer des autorisations à un AWS service](#) dans le guide de l'utilisateur IAM.

Rubriques

- [Créez des rôles pour les flottes ponctuelles Amazon EC2 dans AWS Management Console](#)
- [Créez des rôles pour les flottes ponctuelles Amazon EC2 à l'aide du AWS CLI](#)

Créez des rôles pour les flottes ponctuelles Amazon EC2 dans AWS Management Console

Pour créer le rôle lié au service **AmazonEC2SpotFleetTaggingRole** IAM pour Amazon EC2 Spot Fleet

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Pour la gestion des accès, choisissez Rôles,
3. Pour Rôles, choisissez Créer un rôle.
4. Dans Sélectionner une entité de confiance pour le type d'entité de confiance, sélectionnez Service AWS.
5. Pour les autres cas d'utilisation Services AWS, choisissez EC2, puis EC2 - Spot Fleet Tagging.

6. Choisissez Suivant.
7. Dans Politiques d'autorisations pour le nom de la politique, vérifiez `AmazonEC2SpotFleetTaggingRole`.
8. Choisissez Suivant.
9. Pour Nom, passez en revue et créez :
 - a. Dans Nom du rôle, entrez un nom pour identifier le rôle.
 - b. Dans Description, entrez une brève explication de la politique.
 - c. (Facultatif) Pour l'étape 1 : sélectionnez les entités fiables, choisissez Modifier pour modifier le code.
 - d. (Facultatif) Pour l'étape 2 : Ajouter des autorisations, choisissez Modifier pour modifier le code.
 - e. (Facultatif) Pour Ajouter des balises, choisissez Ajouter une balise pour ajouter des balises à la ressource.
 - f. Choisissez Créer un rôle.

Note

Dans le passé, il existait deux politiques gérées pour le rôle Amazon EC2 Spot Fleet.

- Amazon EC2 SpotFleetRole : il s'agit de la politique gérée d'origine pour le rôle Spot Fleet. Cependant, nous ne vous recommandons plus de l'utiliser avec AWS Batch. Cette politique ne prend pas en charge le balisage des instances Spot dans les environnements informatiques, qui est nécessaire pour utiliser le rôle lié à un `AWSServiceRoleForBatch` service. Si vous avez déjà créé un rôle Spot Fleet avec cette politique, appliquez la nouvelle politique recommandée à ce rôle. Pour de plus amples informations, veuillez consulter [Instances ponctuelles non étiquetées lors de la création](#).
- Amazon EC2 SpotFleetTaggingRole : ce rôle fournit toutes les autorisations nécessaires pour baliser les instances Spot Amazon EC2. Utilisez ce rôle pour autoriser le balisage des instances Spot dans vos environnements AWS Batch informatiques.

Créez des rôles pour les flottes ponctuelles Amazon EC2 à l'aide du AWS CLI

Pour créer le rôle Amazon EC2 SpotFleetTaggingRole IAM pour vos environnements informatiques Spot Fleet

1. Exécutez la commande suivante avec AWS CLI.

```
$ aws iam create-role --role-name AmazonEC2SpotFleetTaggingRole \
  --assume-role-policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "spotfleet.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}'
```

2. Pour associer la politique IAM EC2 SpotFleetTaggingRole gérée par Amazon à votre EC2 SpotFleetTaggingRole rôle Amazon, exécutez la commande suivante avec le AWS CLI.

```
$ aws iam attach-role-policy \
  --policy-arn \
    arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetTaggingRole \
  --role-name \
    AmazonEC2SpotFleetTaggingRole
```

Pour créer le rôle lié au service **AWSServiceRoleForEC2Spot** IAM pour Amazon EC2 Spot

Note

Si le rôle lié au service AWSServiceRoleForEC2Spot IAM existe déjà, un message d'erreur semblable au suivant s'affiche.

```
An error occurred (InvalidInput) when calling the CreateServiceLinkedRole
operation:
```

```
Service role name AWSServiceRoleForEC2Spot has been taken in this account,  
please try a different suffix.
```

- Exécutez la commande suivante avec AWS CLI.

```
$ aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Pour créer le rôle lié au service **AWSServiceRoleForEC2SpotFleet** IAM pour Amazon EC2 Spot Fleet

Note

Si le rôle lié au service **AWSServiceRoleForEC2SpotFleet** IAM existe déjà, un message d'erreur semblable au suivant s'affiche.

```
An error occurred (InvalidInput) when calling the CreateServiceLinkedRole  
operation:  
Service role name AWSServiceRoleForEC2SpotFleet has been taken in this account,  
please try a different suffix.
```

- Exécutez la commande suivante avec AWS CLI.

```
$ aws iam create-service-linked-role --aws-service-name spotfleet.amazonaws.com
```

EventBridge Rôle IAM

Amazon EventBridge fournit un flux quasi en temps réel d'événements système décrivant les modifications apportées aux AWS ressources. AWS Batch des emplois sont disponibles en tant que EventBridge cibles. À l'aide de règles simples et rapidement configurées, vous pouvez faire correspondre des événements et leur soumettre des tâches AWS Batch en réponse. Avant de pouvoir soumettre des AWS Batch tâches avec des EventBridge règles et des cibles, vous EventBridge devez disposer des autorisations nécessaires pour exécuter AWS Batch des tâches en votre nom.

 Note

Lorsque vous créez une règle dans la EventBridge console qui spécifie une AWS Batch file d'attente comme cible, vous pouvez créer ce rôle. Pour afficher un exemple de procédure, consultez [AWS Batch les emplois en tant que EventBridge cibles](#). Vous pouvez créer le EventBridge rôle manuellement à l'aide de la console IAM. Pour obtenir des instructions, consultez [la section Création d'un rôle à l'aide de politiques de confiance personnalisées \(console\)](#) dans le guide de l'utilisateur IAM.

La relation de confiance associée à votre rôle EventBridge IAM doit permettre au directeur du `events.amazonaws.com` service d'assumer ce rôle.

JSON

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

Assurez-vous que la politique associée à votre rôle EventBridge IAM autorise `batch:SubmitJob` les autorisations sur vos ressources. Dans l'exemple suivant, AWS Batch fournit la politique `AWSBatchServiceEventTargetRole` gérée pour fournir ces autorisations.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
{
  "Effect": "Allow",
  "Action": [
    "batch:SubmitJob"
  ],
  "Resource": "*"
}
```

Créer un Virtual Private Cloud

Les ressources informatiques de vos environnements informatiques ont besoin d'un accès au réseau externe pour communiquer avec AWS Batch les points de terminaison du service Amazon ECS. Toutefois, il se peut que vous souhaitiez exécuter des tâches dans des sous-réseaux privés. Pour avoir la possibilité d'exécuter des tâches dans un sous-réseau public ou privé, créez un VPC doté de sous-réseaux publics et privés.

Vous pouvez utiliser Amazon Virtual Private Cloud (Amazon VPC) pour lancer AWS des ressources dans un réseau virtuel que vous définissez. Cette rubrique fournit un lien vers l'assistant Amazon VPC et une liste des options à sélectionner.

Création d'un VPC

Pour plus d'informations sur la création d'un Amazon VPC, consultez la section [Créer un VPC uniquement dans](#) le guide de l'utilisateur Amazon VPC et utilisez le tableau suivant pour déterminer les options à sélectionner.

Option	Value	
Ressources à créer	VPC uniquement	
Nom	Vous pouvez, si vous le souhaitez, nommer votre VPC.	
IPv4 Bloc d'adresse CIDR	IPv4 Entrée manuelle CIDR	

Option	Value	
	La taille du bloc d'adresse CIDR doit être comprise entre /16 et /28.	
IPv6 Bloc d'adresse CIDR	Aucun IPv6 bloc CIDR	
Tenancy	Par défaut	

Pour plus d'informations sur Amazon VPC, consultez [Qu'est-ce qu'Amazon VPC ?](#) dans le Guide de l'utilisateur Amazon VPC.

Étapes suivantes

Après avoir créé votre VPC, considérez les étapes suivantes :

- Créez des groupes de sécurité pour vos ressources publiques et privées si celles-ci nécessitent un accès réseau entrant. Pour plus d'informations, consultez la section [Travailler avec les groupes de sécurité](#) dans le Guide de l'utilisateur Amazon VPC.
- Créez un environnement informatique AWS Batch géré qui lance des ressources de calcul dans votre nouveau VPC. Pour de plus amples informations, veuillez consulter [Création d'un environnement informatique](#). Si vous utilisez l'assistant de création d'environnement informatique de la AWS Batch console, vous pouvez spécifier le VPC que vous venez de créer et les sous-réseaux publics ou privés dans lesquels vous souhaitez lancer vos instances.
- Créez une file d'attente de AWS Batch tâches mappée à votre nouvel environnement informatique. Pour de plus amples informations, veuillez consulter [Création d'une file d'attente de tâches](#).
- Créez une définition de tâche avec laquelle exécuter vos tâches. Pour de plus amples informations, veuillez consulter [Création d'une définition de tâche à nœud unique](#).
- Soumettez une tâche avec votre définition à votre nouvelle file d'attente de tâches. Cette tâche aboutit à l'environnement informatique que vous avez créé avec votre nouveau VPC et vos sous-réseaux. Pour de plus amples informations, veuillez consulter [Tutoriel : soumettre une offre d'emploi](#).

Utiliser un point de terminaison d'interface pour accéder AWS Batch

Vous pouvez l'utiliser AWS PrivateLink pour créer une connexion privée entre votre VPC et. AWS Batch Vous pouvez y accéder AWS Batch comme s'il se trouvait dans votre VPC, sans utiliser de passerelle Internet, de périphérique NAT, de connexion VPN ou Direct Connect de connexion. Les instances de votre VPC n'ont pas besoin d'adresses IP publiques pour y accéder. AWS Batch

Vous établissez cette connexion privée en créant un point de terminaison d'interface optimisé par AWS PrivateLink. Nous créons une interface réseau de point de terminaison dans chaque sous-réseau que vous activez pour le point de terminaison d'interface. Il s'agit d'interfaces réseau gérées par le demandeur qui servent de point d'entrée pour le trafic destiné à AWS Batch.

Pour plus d'informations, consultez la section [Points de terminaison VPC de l'interface dans le Guide](#).AWS PrivateLink

Considérations relatives à AWS Batch

Avant de configurer un point de terminaison d'interface pour AWS Batch, consultez les [propriétés et les limites du point de terminaison d'interface](#) dans le AWS PrivateLink Guide.

AWS Batch prend en charge les appels à toutes ses actions d'API via le point de terminaison de l'interface.

Avant de configurer les points de terminaison VPC d'interface pour AWS Batch, tenez compte des points suivants :

- Les tâches utilisant le type de lancement de ressources Fargate ne nécessitent pas les points de terminaison VPC d'interface pour Amazon ECS, mais vous pourriez avoir besoin de points de terminaison VPC d'interface pour AWS Batch Amazon ECR, Secrets Manager ou Amazon Logs décrits dans les points suivants. CloudWatch
- Pour exécuter des tâches, vous devez créer les points de terminaison VPC de l'interface pour Amazon ECS. Pour plus d'informations, consultez [Interface VPC Endpoints \(AWS PrivateLink\)](#) dans le manuel Amazon Elastic Container Service Developer Guide.
- Pour permettre à vos tâches d'extraire des images privées d'Amazon ECR, vous devez créer les points de terminaison VPC d'interface pour Amazon ECR. Pour de plus amples informations, veuillez consulter [Points de terminaison d'un VPC d'interface \(AWS PrivateLink\)](#) dans le Guide de l'utilisateur Amazon Elastic Container Registry.

- Pour permettre à vos tâches d'extraire des données sensibles de Secrets Manager, vous devez créer les points de terminaison VPC d'interface pour Secrets Manager. Pour de plus amples informations, veuillez consulter [Utilisation de Secrets Manager avec des points de terminaison de VPC](#) dans le Guide de l'utilisateur AWS Secrets Manager .
- Si votre VPC ne possède pas de passerelle Internet et que vos tâches utilisent le pilote de journal pour envoyer les informations de awslogs journal aux CloudWatch journaux, vous devez créer un point de terminaison VPC d'interface pour les journaux. CloudWatch Pour plus d'informations, consultez la section [Utilisation CloudWatch des journaux avec les points de terminaison VPC d'interface dans le guide](#) de l'utilisateur Amazon CloudWatch Logs.
- Les tâches utilisant les ressources EC2 nécessitent que les instances de conteneur sur lesquelles elles sont lancées exécutent la version 1.25.1 ou une version ultérieure de l'agent de conteneur Amazon ECS. Pour plus d'informations, consultez les [versions de l'agent de conteneur Amazon ECS Linux](#) dans le manuel du développeur Amazon Elastic Container Service.
- Les points de terminaison d'un VPC ne prennent pas en charge les demandes inter-régions pour le moment. Veillez à créer votre point de terminaison dans la même région que celle dans laquelle vous souhaitez envoyer vos appels d'API à AWS Batch.
- Les points de terminaison d'un VPC prennent uniquement en charge le DNS fourni par Amazon via Amazon Route 53. Si vous souhaitez utiliser votre propre DNS, vous pouvez utiliser le transfert DNS conditionnel. Pour en savoir plus, consultez [Jeux d'options DHCP](#) dans le Guide de l'utilisateur Amazon VPC.
- Le groupe de sécurité attaché au point de terminaison d'un VPC doit autoriser les connexions entrantes sur le port 443 à partir du sous-réseau privé du VPC.
- AWS Batch ne prend pas en charge les points de terminaison d'interface VPC dans les cas suivants : Régions AWS
 - Asie-Pacifique(Osaka) (ap-northeast-3)
 - Asie-Pacifique (Jakarta) (ap-southeast-3)

Créez un point de terminaison d'interface pour AWS Batch

Vous pouvez créer un point de terminaison d'interface pour AWS Batch utiliser la console Amazon VPC ou le AWS Command Line Interface (AWS CLI). Pour plus d'informations, consultez [Création d'un point de terminaison d'interface](#) dans le Guide AWS PrivateLink .

Créez un point de terminaison d'interface pour AWS Batch utiliser les noms de service suivants :

- `com.amazonaws.region.batch`
- `com.amazonaws.region.batch-fips` (pour les points de terminaison conformes à la norme FIPS, voir [points de terminaison et quotas](#))AWS Batch

Par exemple :

```
com.amazonaws.us-east-2.batch
```

```
com.amazonaws.us-east-2.batch-fips
```

Dans la `aws-cn` partition, le format est différent :

```
cn.com.amazonaws.region.batch
```

Par exemple :

```
cn.com.amazonaws.cn-northwest-1.batch
```

Noms DNS privés pour les points de terminaison AWS Batch d'interface

Si vous activez le DNS privé pour le point de terminaison de l'interface, vous pouvez utiliser des noms DNS spécifiques pour AWS Batch vous connecter. Nous proposons les options suivantes :

- `lot.region.amazonaws.com`
- `lot.region.api.aws`

Pour les terminaux conformes à la norme FIPS :

- virements par lots. `region.api.aws`
- `fips.batch.region`Le fichier `.amazonaws.com` n'est pas pris en charge

Pour plus d'informations, consultez la section [Accès à un service via un point de terminaison d'interface](#) dans le AWS PrivateLink Guide.

Création d'une politique de point de terminaison pour votre point de terminaison d'interface

Une politique de point de terminaison est une ressource IAM que vous pouvez attacher à votre point de terminaison d'interface. La politique de point de terminaison par défaut autorise un accès complet AWS Batch via le point de terminaison de l'interface. Pour contrôler l'accès autorisé à AWS Batch depuis votre VPC, attachez une politique de point de terminaison personnalisée au point de terminaison de l'interface.

Une politique de point de terminaison spécifie les informations suivantes :

- Les principaux qui peuvent effectuer des actions (Comptes AWS utilisateurs et rôles IAM).
- Les actions qui peuvent être effectuées.
- La ressource sur laquelle les actions peuvent être effectuées.

Pour plus d'informations, consultez [Contrôle de l'accès aux services à l'aide de politiques de point de terminaison](#) dans le Guide AWS PrivateLink .

Exemple : politique de point de terminaison VPC pour les actions AWS Batch

Voici un exemple de politique de point de terminaison personnalisée. Lorsque vous attachez cette politique au point de terminaison de votre interface, elle accorde l'accès aux AWS Batch actions répertoriées à tous les principaux sur toutes les ressources.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob",
        "batch:ListJobs",
        "batch:DescribeJobs"
      ],
      "Resource": "*"
    }
  ]
}
```

Validation de conformité pour AWS Batch

Pour savoir si un [programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité](#) spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de [AWS conformité Programmes AWS](#) de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir [Téléchargement de rapports dans AWS Artifact](#) .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. Pour plus d'informations sur votre responsabilité en matière de conformité lors de l'utilisation Services AWS, consultez [AWS la documentation de sécurité](#).

Sécurité de l'infrastructure dans AWS Batch

En tant que service géré, AWS Batch il est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section [Sécurité du AWS cloud](#). Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section [Protection de l'infrastructure](#) dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder AWS Batch via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

Vous pouvez appeler ces opérations d'API depuis n'importe quel emplacement réseau, mais AWS Batch elles prennent en charge les politiques d'accès basées sur les ressources, qui peuvent inclure des restrictions basées sur l'adresse IP source. Vous pouvez également utiliser des AWS Batch politiques pour contrôler l'accès depuis des points de terminaison Amazon Virtual Private Cloud (Amazon VPC) spécifiques ou spécifiques. VPCs En fait, cela isole l'accès réseau à une AWS Batch ressource donnée uniquement du VPC spécifique au sein AWS du réseau.

Prévention du problème de l'adjoint confus entre services

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'a pas l'autorisation d'effectuer une action peut contraindre une entité plus privilégiée à effectuer cette action. En AWS, l'usurpation d'identité interservices peut entraîner la confusion des adjoints. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé et ses autorisations utilisées pour agir sur les ressources d'un autre client auxquelles on ne serait pas autorisé à accéder autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés de contexte de condition [aws:SourceAccount](#) globale [aws:SourceArn](#) et les clés contextuelles dans les politiques de ressources afin de limiter les autorisations qui AWS Batch accordent un autre service à la ressource. Si la valeur `aws:SourceArn` ne contient pas l'ID du compte, tel qu'un ARN de compartiment Amazon S3, vous devez utiliser les deux clés de contexte de condition globale pour limiter les autorisations. Si vous utilisez les deux clés de contexte de condition globale et que la valeur `aws:SourceArn` contient l'ID de compte, la valeur `aws:SourceAccount` et le compte dans la valeur `aws:SourceArn` doivent utiliser le même ID de compte lorsqu'ils sont utilisés dans la même instruction de politique. Utilisez `aws:SourceArn` si vous souhaitez qu'une seule ressource soit associée à l'accès entre services. Utilisez `aws:SourceAccount` si vous souhaitez autoriser l'association d'une ressource de ce compte à l'utilisation interservices.

La valeur de `aws:SourceArn` doit être la ressource qui AWS Batch stocke.

Le moyen le plus efficace de se protéger contre le problème de député confus consiste à utiliser la clé de contexte de condition globale `aws:SourceArn` avec l'ARN complet de la ressource. Si vous ne connaissez pas l'ARN complet de la ressource ou si vous spécifiez plusieurs ressources, utilisez la clé de contexte de condition globale `aws:SourceArn` avec des caractères génériques (*) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:service:*:123456789012:*`.

Les exemples suivants montrent comment utiliser les touches de contexte de condition `aws:SourceAccount` globale `aws:SourceArn` et globale AWS Batch pour éviter le problème de confusion des adjoints.

Exemple : rôle permettant d'accéder à un seul environnement informatique

Le rôle suivant ne peut être utilisé que pour accéder à un seul environnement informatique. Le nom de la tâche doit être spécifié * car la file d'attente des tâches peut être associée à plusieurs environnements informatiques.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:batch:us-east-1:123456789012:compute-environment/testCE",
            "arn:aws:batch:us-east-1:123456789012:job/*"
          ]
        }
      }
    }
  ]
}
```

Exemple : rôle pour accéder à plusieurs environnements informatiques

Le rôle suivant peut être utilisé pour accéder à plusieurs environnements informatiques. Le nom de la tâche doit être spécifié * car la file d'attente des tâches peut être associée à plusieurs environnements informatiques.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:batch:us-east-1:123456789012:compute-environment/*",
            "arn:aws:batch:us-east-1:123456789012:job/*"
          ]
        }
      }
    }
  ]
}
```

Journalisation des appels d' AWS Batch API avec AWS CloudTrail

AWS Batch est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans AWS Batch. CloudTrail capture tous les appels d'API AWS Batch sous forme d'événements. Les appels capturés incluent des appels provenant de la AWS Batch console et des appels de code vers les opérations de l' AWS Batch API. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris les événements pour AWS Batch. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite AWS Batch, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des détails supplémentaires.

Pour en savoir plus CloudTrail, consultez le [guide de AWS CloudTrail l'utilisateur](#).

Rubriques

- [AWS Batch informations dans CloudTrail](#)
- [Référence : Comprendre les entrées du fichier AWS Batch journal](#)

AWS Batch informations dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité se produit dans AWS Batch, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre AWS compte. Pour plus d'informations, consultez la section [Affichage des événements avec l'historique des CloudTrail événements](#).

Pour un enregistrement continu des événements de votre AWS compte, y compris des événements pour AWS Batch, créez un parcours. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un parcours dans la console, celui-ci s'applique à toutes les AWS régions. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez les ressources suivantes :

- [Vue d'ensemble de la création d'un journal d'activité](#)
- [CloudTrail Services et intégrations pris en charge](#)
- [Configuration des notifications Amazon SNS pour CloudTrail](#)
- [Réception de fichiers CloudTrail journaux de plusieurs régions](#) et [réception de fichiers CloudTrail journaux de plusieurs comptes](#)

Toutes les AWS Batch actions sont enregistrées CloudTrail et documentées dans la [https:// docs.aws.amazon.com/batch/ dernière/](https://docs.aws.amazon.com/batch/ dernière/) /. APIReference Par exemple, les appels adressés aux sections [SubmitJob](#), [ListJobs](#) et [DescribeJobs](#) génèrent des entrées dans les fichiers journaux CloudTrail .

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec des informations d'identification d'utilisateur root ou IAM.
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour plus d'informations, consultez la section [Élément `userIdentity` CloudTrail](#).

Référence : Comprendre les entrées du fichier AWS Batch journal

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande individuelle à partir d'une source quelconque et comprend des informations sur l'action demandée, sur tous les paramètres, les paramètres de la demande, etc. Les fichiers journaux CloudTrail ne sont pas des séries ordonnées retraçant les appels d'API publics. Ils ne suivent aucun ordre précis.

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'[CreateComputeEnvironment](#) action.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:admin",
    "arn": "arn:aws:sts::012345678910:assumed-role/Admin/admin",
    "accountId": "012345678910",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2017-12-20T00:48:46Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::012345678910:role/Admin",
        "accountId": "012345678910",
        "userName": "Admin"
      }
    }
  },
  }
```

```

"eventTime": "2017-12-20T00:48:46Z",
"eventSource": "batch.amazonaws.com",
"eventName": "CreateComputeEnvironment",
"awsRegion": "us-east-1",
"sourceIPAddress": "203.0.113.1",
"userAgent": "aws-cli/1.11.167 Python/2.7.10 Darwin/16.7.0 botocore/1.7.25",
"requestParameters": {
  "computeResources": {
    "subnets": [
      "subnet-5eda8e04"
    ],
    "tags": {
      "testBatchTags": "CLI testing CE"
    },
    "desiredvCpus": 0,
    "minvCpus": 0,
    "instanceTypes": [
      "optimal"
    ],
    "securityGroupIds": [
      "sg-aba9e8db"
    ],
    "instanceRole": "ecsInstanceRole",
    "maxvCpus": 128,
    "type": "EC2"
  },
  "state": "ENABLED",
  "type": "MANAGED",
  "computeEnvironmentName": "Test"
},
"responseElements": {
  "computeEnvironmentName": "Test",
  "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-environment/
Test"
},
"requestID": "890b8639-e51f-11e7-b038-EXAMPLE",
"eventID": "874f89fa-70fc-4798-bc00-EXAMPLE",
"readOnly": false,
"eventType": "AwsApiCall",
"recipientAccountId": "012345678910"
}

```

Résoudre les problèmes liés à l'IAM AWS Batch

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec AWS Batch IAM.

Rubriques

- [Je ne suis pas autorisé à effectuer une action dans AWS Batch](#)
- [Je ne suis pas autorisé à effectuer iam : PassRole](#)
- [Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes AWS Batch ressources](#)

Je ne suis pas autorisé à effectuer une action dans AWS Batch

S'il vous AWS Management Console indique que vous n'êtes pas autorisé à effectuer une action, vous devez contacter votre administrateur pour obtenir de l'aide. Votre administrateur est la personne qui vous a fourni votre nom d'utilisateur et votre mot de passe.

L'exemple d'erreur suivant se produit quand l'utilisateur mateojackson tente d'utiliser la console pour afficher des informations détaillées sur une ressource *my-example-widget* fictive, mais ne dispose pas des autorisations batch : *GetWidget* fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
batch: GetWidget on resource: my-example-widget
```

Dans ce cas, Mateo demande à son administrateur de mettre à jour ses politiques pour lui permettre d'accéder à la ressource *my-example-widget* à l'aide de l'action batch : *GetWidget*. Pour plus d'informations sur l'octroi d'autorisations pour transmettre un rôle, consultez la section [Octroi à un utilisateur des autorisations pour transmettre un rôle à un AWS service](#).

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à exécuter iam : PassRole l'action, vos stratégies doivent être mises à jour afin de vous permettre de transmettre un rôle à AWS Batch.

Certains services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, vous devez disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour exécuter une action dans AWS Batch. Toutefois, l'action nécessite que le service ait des autorisations accordées par une fonction de service. Mary n'est pas autorisée à transmettre le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes AWS Batch ressources

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- Pour savoir si ces fonctionnalités sont prises AWS Batch en charge, consultez [Comment AWS Batch fonctionne avec IAM](#).
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section [Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez](#) dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section [Fournir un accès à des ressources Comptes AWS détenues par des tiers](#) dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez [Fournir un accès à des utilisateurs authentifiés en externe \(fédération d'identité\)](#) dans le Guide de l'utilisateur IAM.

- Pour en savoir plus sur la différence entre l'utilisation des rôles et des politiques basées sur les ressources pour l'accès intercompte, consultez [Accès intercompte aux ressources dans IAM](#) dans le Guide de l'utilisateur IAM.

AWS Step Functions

Vous pouvez utiliser la AWS Batch console pour afficher les détails de vos machines d'état Step Functions et des fonctions qu'elles utilisent.

Sections

- [Tutoriel : Afficher les détails de la machine à états](#)
- [Tutoriel : Modifier une machine à états](#)
- [Tutoriel : Exécuter une machine à états](#)

Tutoriel : Afficher les détails de la machine à états

La AWS Batch console affiche une liste de vos machines d'état actuelles Région AWS qui contiennent au moins une étape du flux de travail qui soumet une AWS Batch tâche.

Choisissez une machine d'état pour afficher une représentation graphique du flux de travail. Les étapes surlignées en bleu représentent les AWS Batch emplois. Utilisez les commandes du graphique pour effectuer un zoom avant, un zoom arrière et centrer le graphique.

Note

Lorsqu'une AWS Batch tâche est [référéncée dynamiquement JsonPath](#) dans la définition de la machine à états, les détails de la fonction ne peuvent pas être affichés dans la AWS Batch console. Au lieu de cela, le nom de la tâche est répertorié sous forme de référence dynamique, et les étapes correspondantes dans le graphique sont grisées.

Pour afficher les détails de la machine d'état

1. Ouvrez la [page Workflow orchestration powered by Step Functions](#) de la AWS Batch console.
2. Choisissez une machine d'état.

<result>

La AWS Batch console ouvre la page Détails.

</result>

Pour plus d'informations, consultez [Step Functions](#) dans le Manuel du développeur AWS Step Functions .

Tutoriel : Modifier une machine à états

Lorsque vous souhaitez modifier une machine à états, AWS Batch ouvre la page Modifier la définition de la console Step Functions.

Pour modifier une machine d'état

1. Ouvrez la [page Workflow orchestration powered by Step Functions](#) de la AWS Batch console.
2. Choisissez une machine d'état.
3. Choisissez Modifier.

La console Step Functions ouvre la page Edit definition (Modifier la définition).

4. Modifiez la machine d'état et choisissez Save (Enregistrer).

Pour plus d'informations sur la modification des machines d'état, consultez [Langage de machine d'état Step Functions](#) dans le Manuel du développeur AWS Step Functions .

Tutoriel : Exécuter une machine à états

Lorsque vous souhaitez exécuter une machine à états, AWS Batch ouvre la page Nouvelle exécution de la console Step Functions.

Pour exécuter une machine d'état

1. Ouvrez la [page Workflow orchestration powered by Step Functions](#) de la AWS Batch console.
2. Choisissez une machine d'état.
3. Sélectionnez Execute (Exécuter).

La console Step Functions ouvre la page New execution (Nouvelle exécution).

4. (Facultatif) Modifiez la machine d'état et choisissez Start execution (Démarrer une exécution).

Pour plus d'informations sur l'exécution de machines d'état, consultez [Concepts d'exécution des machines d'état Step Functions](#) dans le Manuel du développeur AWS Step Functions .

AWS Batch flux d'événements pour Amazon EventBridge

Vous pouvez utiliser le flux d' AWS Batch événements pour qu'Amazon EventBridge reçoive des notifications en temps quasi réel concernant l'état actuel des offres d'emploi dans vos files d'attente.

Vous pouvez l'utiliser EventBridge pour obtenir de plus amples informations sur votre AWS Batch service. Plus précisément, vous pouvez l'utiliser pour vérifier l'avancement des tâches, créer des flux de travail AWS Batch personnalisés, générer des rapports ou des statistiques d'utilisation, ou créer vos propres tableaux de bord. Avec AWS Batch et EventBridge, vous n'avez pas besoin d'un code de planification et de surveillance qui interroge en permanence AWS Batch les changements de statut des tâches. Au lieu de cela, vous pouvez gérer les modifications de l'état des AWS Batch tâches de manière asynchrone à l'aide de diverses cibles Amazon EventBridge . Il s'agit AWS Lambda notamment d'Amazon Simple Queue Service, d'Amazon Simple Notification Service ou d'Amazon Kinesis Data Streams.

Les événements du flux d' AWS Batch événements sont assurés d'être diffusés au moins une fois. Dans le cas où des événements dupliqués sont envoyés, l'événement fournit suffisamment d'informations pour identifier les doublons. Ainsi, vous pouvez comparer l'horodatage de l'événement et le statut de la tâche.

AWS Batch des emplois sont disponibles en tant que EventBridge cibles. À l'aide de règles simples, vous pouvez faire correspondre les événements et soumettre AWS Batch des offres d'emploi en réponse à ceux-ci. Pour plus d'informations, voir [Qu'est-ce que c'est EventBridge ?](#) dans le guide de EventBridge l'utilisateur Amazon. Vous pouvez également l'utiliser EventBridge pour planifier des actions automatisées qui se déclenchent automatiquement à certains moments à l'aide cron ou à l'évaluation d'expressions. Pour plus d'informations, consultez [la section Création d'une EventBridge règle Amazon exécutée selon un calendrier](#) dans le guide de EventBridge l'utilisateur Amazon. Pour afficher un exemple de procédure, consultez [AWS Batch les emplois en tant que EventBridge cibles](#). Pour plus d'informations sur l'utilisation du EventBridge planificateur, consultez la section [Configuration d'Amazon EventBridge Scheduler dans](#) le guide de l'utilisateur Amazon. EventBridge

Rubriques

- [AWS Batch événements](#)
- [Tutoriel : Utiliser les notifications AWS utilisateur avec AWS Batch](#)
- [AWS Batch les emplois en tant que EventBridge cibles](#)
- [Tutoriel : écoutez les événements liés à l' AWS Batch emploi en utilisant EventBridge](#)
- [Tutoriel : envoi d'alertes Amazon Simple Notification Service en cas d'échec d'une tâche](#)

AWS Batch événements

AWS Batch envoie les événements de modification du statut de la tâche à EventBridge. AWS Batch suit l'état de vos emplois. Si le statut d'une tâche précédemment soumise change, un événement est invoqué. Par exemple, si une tâche ayant le RUNNING statut passe au FAILED statut. Ces événements sont considérés comme des événements de changement d'état de tâche.

Note

AWS Batch pourrait ajouter d'autres types d'événements, sources et détails à l'avenir. Si vous désérialisez par programmation les données JSON des événements, assurez-vous que votre application est prête à gérer des propriétés inconnues. Cela permet d'éviter des problèmes si et quand ces propriétés supplémentaires sont ajoutées.

Événements liés à la modification de l'état du poste

Chaque fois qu'une tâche existante (précédemment soumise) change d'état, un événement est créé. Pour plus d'informations sur les états des AWS Batch tâches, consultez [États de l'emploi](#).

Note

Les événements ne sont pas créés pour la soumission initiale de la tâche.

Exemple d'événement de modification de l'état de tâche

Les événements de modification de l'état du job sont fournis dans le format suivant. La detail section ressemble à l'[JobDetail](#) objet renvoyé par une opération d'[DescribeJobs](#) API dans la référence AWS Batch d'API. Pour plus d'informations sur EventBridge les paramètres, consultez la section [Événements et modèles d'événements](#) dans le guide de EventBridge l'utilisateur Amazon.

```
{
  "version": "0",
  "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
  "detail-type": "Batch Job State Change",
  "source": "aws.batch",
  "account": "123456789012",
  "time": "2022-01-11T23:36:40Z",
  "region": "us-east-1",
```

```

    "resources": [
      "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
    ],
    "detail": {
      "jobArn": "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
      "jobName": "event-test",
      "jobId": "4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
      "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/PexjEHappyPathCanary2JobQueue",
      "status": "RUNNABLE",
      "attempts": [],
      "createdAt": 1641944200058,
      "retryStrategy": {
        "attempts": 2,
        "evaluateOnExit": []
      },
      "dependsOn": [],
      "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/first-run-job-definition:1",
      "parameters": {},
      "container": {
        "image": "137112412989.dkr.ecr.us-east-1.amazonaws.com/amazonlinux:latest",
        "command": [
          "sleep",
          "600"
        ],
        "volumes": [],
        "environment": [],
        "mountPoints": [],
        "ulimits": [],
        "networkInterfaces": [],
        "resourceRequirements": [
          {
            "value": "2",
            "type": "VCPU"
          }, {
            "value": "256",
            "type": "MEMORY"
          }
        ],
        "secrets": []
      },
      "propagateTags": false,

```

```

    "platformCapabilities": []
  }
}

```

Événements bloqués dans la file d'offres d'emploi

Chaque fois qu'une tâche est AWS Batch détectée dans RUNNABLE cet état et bloque ainsi une file d'attente, un événement est créé dans Amazon CloudWatch Events. Pour plus d'informations sur les causes de blocage des files d'attente prises en charge, consultez [Offres d'emploi bloquées dans un RUNNABLE statut](#). La même raison est également disponible dans le `statusReason` champ de l'action [DescribeJobs](#) API.

Exemple Événement bloqué dans la file d'offres d'emploi

Les événements bloqués dans la file d'attente de tâches sont transmis au format suivant. La `detail` section ressemble à l'[JobDetail](#) objet renvoyé par une opération d'[DescribeJobs](#) API dans la référence AWS Batch d'API. Pour plus d'informations sur EventBridge les paramètres, consultez la section [Événements et modèles d'événements](#) dans le guide de EventBridge l'utilisateur Amazon.

```

{
  "version": "0",
  "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
  "detail-type": "Batch Job Queue Blocked",
  "source": "aws.batch",
  "account": "123456789012",
  "time": "2022-01-11T23:36:40Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
    "arn:aws:batch:us-east-1:123456789012:job-queue/PexjEHappyPathCanary2JobQueue"
  ],
  "detail": {
    "jobArn": "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
    "jobName": "event-test",
    "jobId": "4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
    "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/PexjEHappyPathCanary2JobQueue",
    "status": "RUNNABLE",
    "statusReason": "blocked-reason",

```

```

    "attempts": [],
    "createdAt": 1641944200058,
    "retryStrategy": {
      "attempts": 2,
      "evaluateOnExit": []
    },
    "dependsOn": [],
    "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/first-run-job-definition:1",
    "parameters": {},
    "container": {
      "image": "137112412989.dkr.ecr.us-east-1.amazonaws.com/amazonlinux:latest",
      "command": [
        "sleep",
        "600"
      ],
      "volumes": [],
      "environment": [],
      "mountPoints": [],
      "ulimits": [],
      "networkInterfaces": [],
      "resourceRequirements": [
        {
          "value": "2",
          "type": "VCPU"
        }, {
          "value": "256",
          "type": "MEMORY"
        }
      ],
      "secrets": []
    },
    "propagateTags": false,
    "platformCapabilities": []
  }
}

```

Événements de modification de l'état des tâches de service

Chaque fois qu'une tâche de service existante change d'état, un événement est créé. Pour plus d'informations sur les états des tâches de service, consultez [Mappage AWS Batch de l'état de la tâche du service avec SageMaker le statut de](#).

Note

Les événements ne sont pas créés pour la soumission initiale de la tâche.

Exemple d'événement de changement d'état de Service Job

Les événements de modification de l'état des tâches de service sont fournis dans le format suivant. La `detail` section ressemble à la réponse renvoyée par une opération d'[DescribeServiceJob](#) API dans la référence AWS Batch d'API. Pour plus d'informations sur EventBridge les paramètres, consultez la section [Événements et modèles d'événements](#) dans le guide de EventBridge l'utilisateur Amazon.

Note

Les `serviceRequestPayload` champs `tags` et ne sont pas inclus dans l'événement `detail`.

```
{
  "version": "0",
  "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
  "detail-type": "Batch Service Job State Change",
  "source": "aws.batch",
  "account": "123456789012",
  "time": "2022-01-11T23:36:40Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:batch:us-east-1:123456789012:service-job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
  ],
  "detail": {
    "attempts": [
      {
        "serviceResourceId": {
          "name": "TrainingJobArn",
          "value": "arn:aws:sagemaker:us-east-1:123456789012:training-job/AWSBatchmy-training-job88b610a69aa8380ca5b0a7aba3f81cb8"
        },
        "startedAt": 1641944300058,
        "stoppedAt": 1641944400058,

```

```

    "statusReason": "Received status from SageMaker: Training job completed"
  }
],
"createdAt": 1641944200058,
"jobArn": "arn:aws:batch:us-east-1:123456789012:service-job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
"jobId": "0bb17543-ece6-4480-b1a7-a556d344746b",
"jobName": "event-test",
"jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/HappyPathJobQueue",
"latestAttempt": {
  "serviceResourceId": {
    "name": "TrainingJobArn",
    "value": "arn:aws:sagemaker:us-east-1:123456789012:training-job/AWSBatchmy-training-job88b610a69aa8380ca5b0a7aba3f81cb8"
  }
},
"serviceJobType": "SAGEMAKER_TRAINING",
"startedAt": 1641944300058,
"status": "SUCCEEDED",
"statusReason": "Received status from SageMaker: Training job completed",
"stoppedAt": 1641944400058,
"timeoutConfig": {
  "attemptDurationSeconds": 60
}
}
}

```

Événements bloqués dans la file d'attente des tâches de service

Chaque fois qu'une file d'attente bloquée est AWS Batch détectée, un événement est créé dans Amazon CloudWatch Events. La raison du blocage de la file d'attente est disponible dans le `statusReason` champ de l'action [DescribeServiceJob](#) API.

Exemple Événement bloqué par Service Job Queue

Les événements bloqués dans la file d'attente des tâches de service sont transmis au format suivant. La `detail` section ressemble à la réponse renvoyée par l'opération d'[DescribeServiceJob](#) API dans la référence AWS Batch d'API. Pour plus d'informations sur EventBridge les paramètres, consultez la section [Événements et modèles d'événements](#) dans le guide de EventBridge l'utilisateur Amazon.

Note

Les `serviceRequestPayload` champs `tags` et ne sont pas inclus dans `l'événementdetail`.

```
{
  "version": "0",
  "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
  "detail-type": "Batch Service Job Queue Blocked",
  "source": "aws.batch",
  "account": "123456789012",
  "time": "2022-01-11T23:36:40Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:batch:us-east-1:123456789012:service-job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
  ],
  "detail": {
    "attempts": [],
    "createdAt": 1641944200058,
    "jobArn": "arn:aws:batch:us-east-1:123456789012:service-job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
    "jobId": "6271dfdf-d8a7-41b1-a4d2-55a2224f5375",
    "jobName": "event-test",
    "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/HappyPathJobQueue",
    "serviceJobType": "SAGEMAKER_TRAINING",
    "status": "RUNNABLE",
    "statusReason": "blocked-reason",
    "timeoutConfig": {
      "attemptDurationSeconds": 60
    }
  }
}
```

Tutoriel : Utiliser les notifications AWS utilisateur avec AWS Batch

Vous pouvez utiliser [les notifications aux AWS utilisateurs](#) pour configurer des canaux de diffusion afin d'être informé AWS Batch des événements. Vous recevez une notification lorsqu'un événement correspond à une règle que vous avez spécifiée. Vous pouvez recevoir des notifications d'événements via plusieurs canaux, notamment par e-mail, [Amazon Q Developer](#)

dans les [applications de chat](#), les notifications par chat ou les notifications [AWS Console Mobile Application](#) push. Vous pouvez également consulter les notifications dans le [Centre de notifications de la console](#). Notifications utilisateur prend en charge l'agrégation, ce qui peut réduire le nombre de notifications que vous recevez lors d'événements spécifiques.

Pour configurer les notifications utilisateur dans AWS Batch :

1. Ouvrez la [AWS Batch console](#).
2. Choisissez Tableau de bord.
3. Choisissez Configurer les notifications.
4. Dans Notifications AWS utilisateur, choisissez Créer une configuration de notification.

Pour plus d'informations sur la façon de configurer et d'afficher les notifications utilisateur, voir [Commencer à utiliser les notifications AWS utilisateur](#).

AWS Batch les emplois en tant que EventBridge cibles

Amazon EventBridge fournit un flux d'événements système en temps quasi réel décrivant les modifications apportées aux ressources Amazon Web Services. Généralement, AWS Batch sur Amazon Elastic Container Service, Amazon Elastic Kubernetes Service AWS et Fargate, les jobs sont disponibles en tant que cibles. EventBridge À l'aide de règles simples, vous pouvez faire correspondre les événements et soumettre AWS Batch des offres d'emploi en réponse à ces événements. Pour plus d'informations, voir [Qu'est-ce que c'est EventBridge ?](#) dans le guide de EventBridge l'utilisateur Amazon.

Vous pouvez également l'utiliser EventBridge pour planifier des actions automatisées qui sont invoquées à certains moments à l'aide cron ou à l'évaluation d'expressions. Pour plus d'informations, consultez [la section Création d'une EventBridge règle Amazon exécutée selon un calendrier](#) dans le guide de EventBridge l'utilisateur Amazon.

Pour plus d'informations sur la création d'une règle qui s'exécute lorsqu'un événement correspond à un modèle d'événement, consultez la section [Création de EventBridge règles Amazon qui réagissent aux événements](#) dans le guide de EventBridge l'utilisateur Amazon.

Les cas d'utilisation courants des AWS Batch jobs en tant que EventBridge cible incluent les cas d'utilisation suivants :

- Une tâche planifiée est exécutée à intervalles réguliers. Par exemple, une cron tâche est exécutée uniquement pendant les heures de faible utilisation, lorsque les instances Amazon EC2 Spot sont moins chères.
- Une AWS Batch tâche s'exécute en réponse à une opération d'API connectée CloudTrail. Par exemple, une tâche est soumise chaque fois qu'un objet est chargé dans un compartiment Amazon S3 spécifié. Chaque fois que cela se produit, le transformateur EventBridge d'entrée transmet le nom du compartiment et de la clé de l'objet aux AWS Batch paramètres.

Note

Dans ce scénario, toutes les AWS ressources associées doivent se trouver dans la même région. Cela inclut des ressources telles que le compartiment, la EventBridge règle et les CloudTrail journaux Amazon S3.

Avant de pouvoir soumettre AWS Batch des tâches avec des EventBridge règles et des cibles, le EventBridge service a besoin de plusieurs autorisations pour exécuter AWS Batch des tâches. Lorsque vous créez une règle dans la EventBridge console qui spécifie une AWS Batch tâche comme cible, vous pouvez également créer ce rôle. Pour en savoir plus sur le principal de service et les autorisations IAM nécessaires pour ce rôle, consultez [EventBridge Rôle IAM](#).

Rubriques

- [Tutoriel : Création d'une AWS Batch tâche planifiée](#)
- [Tutoriel : Création d'une règle avec un modèle d'événement](#)
- [Tutoriel : Transmettre les informations d'un événement à une AWS Batch cible selon un calendrier à l'aide du transformateur EventBridge d'entrée](#)

Tutoriel : Création d'une AWS Batch tâche planifiée

La procédure suivante explique comment créer une AWS Batch tâche planifiée et le rôle EventBridge IAM requis.

Pour créer une AWS Batch tâche planifiée avec EventBridge

Note

Cette procédure fonctionne pour toutes AWS Batch les tâches Amazon ECS, Amazon EKS et AWS Fargate.

1. Ouvrez la EventBridge console Amazon à l'adresse <https://console.aws.amazon.com/events/>.
2. Dans la barre de navigation, sélectionnez le Région AWS à utiliser.
3. Dans le volet de navigation, choisissez Règles.
4. Choisissez Créer une règle.
5. Dans Nom, spécifiez un nom unique pour votre environnement informatique. Le nom peut contenir jusqu'à 64 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).

Note

Une règle ne peut pas avoir le même nom qu'une autre règle de la même région et sur le même bus d'événement.

6. (Facultatif) Dans Description, entrez une description pour la règle.
7. Pour Event bus (Bus d'événement), sélectionnez le bus d'événement que vous souhaitez associer à cette règle. Si vous souhaitez que cette règle mette en correspondance les événements en provenance de votre compte, sélectionnez Par défaut. Lorsqu'un événement Service AWS de votre compte est émis, il est toujours redirigé vers le bus d'événements par défaut de votre compte.
8. (Facultatif) Désactivez la règle sur le bus sélectionné si vous ne souhaitez pas l'exécuter immédiatement.
9. Pour Rule type (Type de règle), choisissez Schedule (Planifier).
10. Choisissez Continuer pour créer une règle ou Suivant.
11. Pour Schedule pattern (Planifier le modèle), effectuez l'une des étapes suivantes :
 - Choisissez Un calendrier détaillé qui s'exécute à une heure précise, à 8 h 00 PST le premier lundi de chaque mois, puis entrez une expression cron. Pour plus d'informations, consultez la section [Cron Expressions](#) dans le guide de EventBridge l'utilisateur Amazon.

- Choisissez Un horaire qui fonctionne à un rythme régulier, par exemple toutes les 10 minutes. puis entrez une expression de taux.
12. Choisissez Suivant.
 13. Pour Target types (Types de cibles), choisissez Service AWS.
 14. Pour Sélectionner une cible, choisissez Batch job queue. Configurez ensuite les éléments suivants :
 - Job queue (File d'attente de tâches) : saisissez l'Amazon Resource Name (ARN) de la file d'attente de tâches où programmer votre tâche.
 - Job definition (Définition de tâche) : saisissez le nom et la révision ou l'ARN complet de la définition de tâche à utiliser pour la tâche.
 - Job name (Nom de la tâche) : saisissez un nom pour votre tâche.
 - Array size (Taille du tableau) : (facultatif) saisissez une taille de tableau pour que votre tâche exécute plusieurs copies. Pour de plus amples informations, veuillez consulter [Tâches liées à](#).
 - Job attempts (Tentatives d'exécution de tâche) : (facultatif) saisissez le nombre de nouvelles tentatives pour votre tâche en cas d'échec. Pour de plus amples informations, veuillez consulter [Nouvelles tentatives de travail automatisées](#).
 15. Pour les types de cibles de file d'attente de tâches EventBridge Batch, une autorisation est requise pour envoyer des événements à la cible. EventBridge peut créer le rôle IAM nécessaire à l'exécution de votre règle. Effectuez l'une des actions suivantes :
 - Pour créer un rôle IAM automatiquement, sélectionnez Create a new role for this specific resource.
 - Pour utiliser un rôle IAM que vous avez déjà créé, choisissez Utiliser un rôle existant.
 16. (Facultatif) Développez Additional settings (Paramètres supplémentaires).
 - a. Pour Configurer la saisie cible, choisissez la manière dont le texte d'un événement est traité avant qu'il ne soit transmis à la cible.
 - b. Pour Âge maximal de l'événement, spécifiez l'intervalle de temps pendant lequel les événements non traités sont conservés.
 - c. Pour Réessayer, entrez le nombre de tentatives renouvelées pour un événement.
 - d. Pour la file d'attente de lettres mortes, choisissez une option concernant la manière dont les événements non traités sont gérés. Si nécessaire, spécifiez la file d'attente Amazon SQS à utiliser comme file d'attente de lettres mortes.

17. (Facultatif) Sélectionnez Add another target (Ajouter une autre cible) pour ajouter une nouvelle cible pour cette règle.
18. Choisissez Suivant.
19. (Facultatif) Pour les balises, choisissez Ajouter une nouvelle balise pour ajouter une étiquette de ressource pour la règle. Pour plus d'informations, consultez [Amazon EventBridge tags](#).
20. Choisissez Suivant.
21. Pour Réviser et créer, passez en revue les étapes de configuration. Si vous devez apporter des modifications, choisissez Modifier. Lorsque vous avez terminé, choisissez Créer une règle.

Pour plus d'informations sur la création de règles, consultez [la section Création d'une EventBridge règle Amazon exécutée selon un calendrier](#) dans le guide de EventBridge l'utilisateur Amazon.

Tutoriel : Création d'une règle avec un modèle d'événement

La procédure suivante explique comment créer une règle avec un modèle d'événement.

Pour créer une règle qui envoie l'événement à une cible lorsque l'événement correspond à un modèle défini

Note

Cette procédure fonctionne pour toutes AWS Batch les tâches Amazon ECS, Amazon EKS et AWS Fargate.

1. Ouvrez la EventBridge console Amazon à l'adresse <https://console.aws.amazon.com/events/>.
2. Dans la barre de navigation, sélectionnez le Région AWS à utiliser.
3. Dans le volet de navigation, choisissez Règles.
4. Choisissez Créer une règle.
5. Dans Nom, spécifiez un nom unique pour votre environnement informatique. Le nom peut contenir jusqu'à 64 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).

 Note

Une règle ne peut pas avoir le même nom qu'une autre règle de la même région et sur le même bus d'événement.

6. (Facultatif) Dans Description, entrez une description pour la règle.
7. Pour Event bus (Bus d'événement), sélectionnez le bus d'événement que vous souhaitez associer à cette règle. Si vous souhaitez que cette règle mette en correspondance les événements en provenance de votre compte, sélectionnez Par défaut. Lorsqu'un événement Service AWS de votre compte est émis, il est toujours redirigé vers le bus d'événements par défaut de votre compte.
8. (Facultatif) Désactivez la règle sur le bus sélectionné si vous ne souhaitez pas l'exécuter immédiatement.
9. Pour Type de règle, choisissez Règle avec un modèle d'événement.
10. Choisissez Suivant.
11. Pour Event Source, choisissez un AWS événement ou des événements EventBridge partenaires.
12. (Facultatif) Pour un exemple d'événement :
 - a. Dans Exemple de type d'événement, sélectionnez AWS événements.
 - b. Pour Sample events, choisissez Batch Job State Change.
13. Pour Méthode de création, choisissez Utiliser le formulaire d'événement.
14. Pour le modèle d'événement :
 - a. Pour Event source (Source d'événement), choisissez Services AWS.
 - b. Pour Service AWS, choisissez Batch.
 - c. Pour Type d'événement, choisissez Batch Job State Change.
15. Choisissez Suivant.
16. Pour Target types (Types de cibles), choisissez Service AWS.
17. Pour Sélectionner une cible, choisissez un type de cible. Par exemple, choisissez Batch job queue. Spécifiez ensuite les éléments suivants :
 - Job queue (File d'attente de tâches) : saisissez l'Amazon Resource Name (ARN) de la file d'attente de tâches où programmer votre tâche.

- Job definition (Définition de tâche) : saisissez le nom et la révision ou l'ARN complet de la définition de tâche à utiliser pour la tâche.
 - Job name (Nom de la tâche) : saisissez un nom pour votre tâche.
 - Array size (Taille du tableau) : (facultatif) saisissez une taille de tableau pour que votre tâche exécute plusieurs copies. Pour de plus amples informations, veuillez consulter [Tâches liées à](#).
 - Job attempts (Tentatives d'exécution de tâche) : (facultatif) saisissez le nombre de nouvelles tentatives pour votre tâche en cas d'échec. Pour de plus amples informations, veuillez consulter [Nouvelles tentatives de travail automatisées](#).
18. Pour les types de cibles de file d'attente de tâches EventBridge Batch, une autorisation est requise pour envoyer des événements à la cible. EventBridge peut créer le rôle IAM nécessaire à l'exécution de votre règle. Effectuez l'une des actions suivantes :
- Pour créer un rôle IAM automatiquement, choisissez Create a new role for this specific resource.
 - Pour utiliser un rôle IAM que vous avez créé auparavant, choisissez Utiliser le rôle existant.
19. (Facultatif) Développez Additional settings (Paramètres supplémentaires).
- a. Pour Configurer la saisie cible, choisissez le mode de traitement du texte d'un événement.
 - b. Pour Âge maximal de l'événement, spécifiez l'intervalle de temps pendant lequel les événements non traités sont conservés.
 - c. Pour Réessayer, entrez le nombre de tentatives renouvelées pour un événement.
 - d. Pour la file d'attente de lettres mortes, choisissez une option concernant la manière dont les événements non traités sont gérés. Si nécessaire, spécifiez la file d'attente Amazon SQS à utiliser comme file d'attente de lettres mortes.
20. (Facultatif) Choisissez Ajouter une autre cible pour ajouter une cible supplémentaire.
21. Choisissez Suivant.
22. (Facultatif) Pour les balises, choisissez Ajouter une nouvelle balise pour ajouter une étiquette de ressource. Pour plus d'informations, consultez les [EventBridge balises Amazon](#) dans le guide de EventBridge l'utilisateur Amazon.
23. Choisissez Suivant.
24. Pour Réviser et créer, passez en revue les étapes de configuration. Si vous devez apporter des modifications, choisissez Modifier. Lorsque vous avez terminé, choisissez Créer une règle.

Pour plus d'informations sur la création de règles, consultez [la section Création de EventBridge règles Amazon qui réagissent aux événements](#) dans le guide de EventBridge l'utilisateur Amazon.

Tutoriel : Transmettre les informations d'un événement à une AWS Batch cible selon un calendrier à l'aide du transformateur EventBridge d'entrée

Vous pouvez utiliser le transformateur EventBridge d'entrée pour transmettre des informations sur les événements AWS Batch lors de la soumission d'une tâche. Cela peut être particulièrement utile si vous invoquez des tâches à la suite d'autres informations relatives à un AWS événement. Le téléchargement d'un objet dans un compartiment Amazon S3 en est un exemple. Vous pouvez également utiliser une définition de tâche avec des valeurs de substitution de paramètres dans la commande du conteneur. Le transformateur EventBridge d'entrée peut fournir les valeurs des paramètres en fonction des données d'événement.

Ensuite, vous créez une cible d' AWS Batch événement qui analyse les informations de l'événement qui le lance et les transforme en `parameters` objet. Lorsque la tâche s'exécute, les paramètres de l'événement déclencheur sont transmis à la commande du conteneur de tâches.

Note

Dans ce scénario, toutes les AWS ressources (telles que les compartiments, les EventBridge règles et les CloudTrail journaux Amazon S3) doivent se trouver dans la même région.

Pour créer une AWS Batch cible utilisant le transformateur d'entrée

1. Ouvrez la EventBridge console Amazon à l'adresse <https://console.aws.amazon.com/events/>.
2. Dans la barre de navigation, sélectionnez le Région AWS à utiliser.
3. Dans le volet de navigation, choisissez Règles.
4. Choisissez Créer une règle.
5. Dans Nom, spécifiez un nom unique pour votre environnement informatique. Le nom peut contenir jusqu'à 64 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).

 Note

Une règle ne peut pas porter le même nom qu'une autre règle du même bus Région AWS d'événements ou du même bus d'événements.

6. (Facultatif) Dans Description, entrez une description pour la règle.
7. Pour Event bus (Bus d'événement), sélectionnez le bus d'événement que vous souhaitez associer à cette règle. Si vous souhaitez que cette règle mette en correspondance les événements en provenance de votre compte, sélectionnez Par défaut. Lorsqu'un événement Service AWS de votre compte est émis, il est toujours redirigé vers le bus d'événements par défaut de votre compte.
8. (Facultatif) Désactivez la règle sur le bus sélectionné si vous ne souhaitez pas l'exécuter immédiatement.
9. Pour Rule type (Type de règle), choisissez Schedule (Planifier).
10. Choisissez Continuer pour créer une règle ou Suivant.
11. Pour Schedule pattern (Planifier le modèle), effectuez l'une des étapes suivantes :
 - Choisissez Un calendrier détaillé qui s'exécute à une heure précise, à 8 h 00 PST le premier lundi de chaque mois, puis entrez une expression cron. Pour plus d'informations, consultez la section [Cron Expressions](#) dans le guide de EventBridge l'utilisateur Amazon.
 - Choisissez Un horaire qui fonctionne à un rythme régulier, par exemple toutes les 10 minutes. puis entrez une expression de taux.
12. Choisissez Suivant.
13. Pour Target types (Types de cibles), choisissez Service AWS.
14. Pour Sélectionner une cible, choisissez Batch job queue. Configurez ensuite les éléments suivants :
 - Job queue (File d'attente de tâches) : saisissez l'Amazon Resource Name (ARN) de la file d'attente de tâches où programmer votre tâche.
 - Job definition (Définition de tâche) : saisissez le nom et la révision ou l'ARN complet de la définition de tâche à utiliser pour la tâche.
 - Job name (Nom de la tâche) : saisissez un nom pour votre tâche.
 - Array size (Taille du tableau) : (facultatif) saisissez une taille de tableau pour que votre tâche exécute plusieurs copies. Pour de plus amples informations, veuillez consulter [Tâches liées à](#).

- Job attempts (Tentatives d'exécution de tâche) : (facultatif) saisissez le nombre de nouvelles tentatives pour votre tâche en cas d'échec. Pour de plus amples informations, veuillez consulter [Nouvelles tentatives de travail automatisées](#).
15. Pour les types de cibles de file d'attente de tâches EventBridge Batch, une autorisation est requise pour envoyer des événements à la cible. EventBridge peut créer le rôle IAM nécessaire à l'exécution de votre règle. Effectuez l'une des actions suivantes :
 - Pour créer un rôle IAM automatiquement, sélectionnez Create a new role for this specific resource.
 - Pour utiliser un rôle IAM que vous avez déjà créé, choisissez Utiliser un rôle existant.
 16. (Facultatif) Développez Additional settings (Paramètres supplémentaires).
 17. Dans la section Additional settings (Réglages supplémentaires), pour Configure target input (Configurer l'entrée cible), choisissez Input Transformer (Transformateur d'entrée).
 18. Choisissez Configure input transformer (Configurer le transformateur d'entrée).
 19. (Facultatif) Pour un exemple d'événement :
 - a. Dans Exemple de type d'événement, sélectionnez AWS événements.
 - b. Pour Sample events, choisissez Batch Job State Change.
 20. Dans la section Target input transformer (Transformateur d'entrée cible), pour Input path (Chemin d'entrée), spécifiez les valeurs à analyser depuis l'événement déclencheur. Par exemple, pour analyser l'événement Batch Job State Change, utilisez le format JSON suivant.

```
{
  "instance": "$.detail.jobId",
  "state": "$.detail.status"
}
```

21. Pour Modèle, entrez ce qui suit.

```
{
  "instance": <jobId> ,
  "status": <status>
}
```

22. Choisissez Confirmer.
23. Pour Âge maximal de l'événement, spécifiez l'intervalle de temps pendant lequel les événements non traités sont conservés.

24. Pour Réessayer, entrez le nombre de tentatives renouvelées pour un événement.
25. Pour la file d'attente de lettres mortes, choisissez une option concernant la manière dont les événements non traités sont gérés. Si nécessaire, spécifiez la file d'attente Amazon SQS à utiliser comme file d'attente de lettres mortes.
26. (Facultatif) Choisissez Ajouter une autre cible pour ajouter une cible supplémentaire.
27. Choisissez Suivant.
28. (Facultatif) Pour les balises, choisissez Ajouter une nouvelle balise pour ajouter une étiquette de ressource. Pour plus d'informations, consultez les [EventBridge balises Amazon](#) dans le guide de EventBridge l'utilisateur Amazon.
29. Choisissez Suivant.
30. Pour Réviser et créer, passez en revue les étapes de configuration. Si vous devez apporter des modifications, choisissez Modifier. Lorsque vous avez terminé, choisissez Créer une règle.

Tutoriel : écoutez les événements liés à l' AWS Batch emploi en utilisant EventBridge

Dans ce didacticiel, vous allez configurer une AWS Lambda fonction simple qui écoute les événements liés aux AWS Batch tâches et les enregistre dans un flux de log CloudWatch Logs.

Prérequis

Ce didacticiel suppose que vous disposez d'un environnement de calcul en fonctionnement et une file d'attente prête à accepter des tâches. Si vous ne disposez pas d'un environnement informatique en cours d'exécution ni d'une file d'attente de tâches à partir desquels capturer les événements, suivez les étapes [Commencer à utiliser les AWS Batch didacticiels](#) décrites pour en créer une. À la fin de ce didacticiel, vous pouvez éventuellement soumettre une tâche à cette file d'attente de tâches pour vérifier que vous avez correctement configuré votre fonction Lambda.

Rubriques

- [Tutoriel : Création de la fonction Lambda](#)
- [Tutoriel : Enregistrer la règle de l'événement](#)
- [Tutoriel : Testez votre configuration](#)

Tutoriel : Création de la fonction Lambda

Dans cette procédure, vous créez une fonction Lambda simple qui servira de cible aux messages du flux d' AWS Batch événements.

Pour créer une fonction Lambda cible

1. Ouvrez la AWS Lambda console à l'adresse <https://console.aws.amazon.com/lambda/>.
2. Sélectionnez Create function (Créer une fonction), puis Author from scratch (Créer à partir de zéro).
3. Sous Nom de la fonction, saisissez batch-event-stream-handler.
4. Pour Environnement d'exécution, sélectionnez Python 3.8.
5. Sélectionnez Create function (Créer une fonction).
6. Dans la section Source du code, modifiez l'exemple de code pour qu'il corresponde à l'exemple suivant :

```
import json

def lambda_handler(event, _context):
    # _context is not used
    del _context
    if event["source"] != "aws.batch":
        raise ValueError("Function only supports input from events with a source
type of: aws.batch")

    print(json.dumps(event))
```

Il s'agit d'une simple fonction Python 3.8 qui affiche les événements envoyés par AWS Batch. Si tout est correctement configuré, à la fin de ce didacticiel, les détails de l'événement apparaissent dans le flux du journal CloudWatch des journaux associé à cette fonction Lambda.

7. Choisissez Déployer.

Tutoriel : Enregistrer la règle de l'événement

Dans cette section, vous allez créer une règle d' EventBridge événement qui capture les événements liés aux tâches qui proviennent de vos AWS Batch ressources. Cette règle capture tous les

événements AWS Batch provenant du compte où elle est définie. Les messages de travail eux-mêmes contiennent des informations sur la source de l'événement, y compris la file d'attente des tâches dans laquelle il a été soumis. Vous pouvez utiliser ces informations pour filtrer et trier les événements par programmation.

 Note

Si vous utilisez la règle AWS Management Console pour créer un événement, la console ajoute automatiquement les autorisations IAM EventBridge pour appeler votre fonction Lambda. Toutefois, si vous créez une règle d'événement à l'aide du AWS CLI, vous devez accorder des autorisations de manière explicite. Pour plus d'informations, consultez la section [Événements et modèles d'événements](#) dans le guide de EventBridge l'utilisateur Amazon.

Pour créer votre EventBridge règle

1. Ouvrez la EventBridge console Amazon à l'adresse <https://console.aws.amazon.com/events/>.
2. Dans le volet de navigation, choisissez Règles.
3. Choisissez Créer une règle.
4. Saisissez un nom et une description pour la règle.

Une règle ne peut pas avoir le même nom qu'une autre règle de la même région et sur le même bus d'événement.

5. Pour Event bus (Bus d'événement), sélectionnez le bus d'événement que vous souhaitez associer à cette règle. Si vous souhaitez que cette règle corresponde aux événements provenant de votre compte, sélectionnez Bus d'événements par défaut AWS . Lorsqu'un AWS service de votre compte émet un événement, celui-ci est toujours redirigé vers le bus d'événements par défaut de votre compte.
6. Pour Type de règle, choisissez Règle avec un modèle d'événement.
7. Choisissez Suivant.
8. Pour Event source (Source de l'événement), choisissez Other (Autres).
9. Pour Modèle d'événement, sélectionnez Modèles personnalisés (éditeur JSON).
10. Collez le modèle d'événement suivants dans la zone de texte.

```
{
```

```
"source": [  
  "aws.batch"  
]  
}
```

Cette règle s'applique à tous vos AWS Batch groupes et à tous les AWS Batch événements. Autrement, vous pouvez créer une règle plus spécifique pour filtrer certains résultats.

11. Choisissez Suivant.
12. Pour Types de cibles, choisissez service AWS .
13. Pour Sélectionner une cible, choisissez la fonction Lambda, puis sélectionnez votre fonction Lambda.
14. (Facultatif) Pour Additional settings (Paramètres supplémentaires), procédez comme suit :
 - a. Pour Maximum age of event (Âge maximal de l'événement), saisissez une valeur comprise entre une minute (00:01) et 24 heures (24:00).
 - b. Pour Retry attempts (Nouvelles tentatives), saisissez un nombre compris entre 0 et 185.
 - c. Pour la file d'attente de lettres mortes, choisissez si vous souhaitez utiliser une file d'attente Amazon SQS standard comme file d'attente de lettres mortes. EventBridge envoie les événements qui correspondent à cette règle à la file d'attente des lettres mortes s'ils ne sont pas correctement transmis à la cible. Effectuez l'une des actions suivantes :
 - Choisissez None (Aucune) pour ne pas utiliser de file d'attente de lettres mortes.
 - Choisissez Sélectionnez une file d'attente Amazon SQS dans le AWS compte courant à utiliser comme file d'attente de lettres mortes, puis sélectionnez la file d'attente à utiliser dans le menu déroulant.
 - Choisissez Sélectionnez une file d'attente Amazon SQS dans un autre AWS compte en tant que file d'attente de lettres mortes, puis entrez l'ARN de la file d'attente à utiliser. Vous devez associer à la file d'attente une politique basée sur les ressources qui EventBridge autorise l'envoi de messages. Pour plus d'informations, consultez la section [Octroi d'autorisations à la file d'attente de lettres mortes](#) dans le guide de EventBridge l'utilisateur Amazon.
15. Choisissez Suivant.
16. (Facultatif) Saisissez une ou plusieurs balises pour la règle. Pour plus d'informations, consultez les [EventBridge balises Amazon](#) dans le guide de EventBridge l'utilisateur Amazon.
17. Choisissez Suivant.
18. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

Tutoriel : Testez votre configuration

Vous pouvez désormais tester votre EventBridge configuration en soumettant une tâche à votre file d'attente de tâches. Si tout est correctement configuré, votre fonction Lambda est déclenchée et elle écrit les données de l'événement dans un flux de log CloudWatch Logs pour la fonction.

Pour tester votre configuration

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Soumettez une nouvelle AWS Batch offre d'emploi. Pour de plus amples informations, veuillez consulter [Tutoriel : soumettre une offre d'emploi](#).
3. Ouvrez la CloudWatch console à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
4. Dans le volet de navigation, choisissez Logs et sélectionnez le groupe de journaux pour votre fonction Lambda (par exemple, **my-function**/aws/lambda/).
5. Sélectionnez un flux de journaux pour afficher les données d'événement.

Tutoriel : envoi d'alertes Amazon Simple Notification Service en cas d'échec d'une tâche

Dans ce didacticiel, vous allez configurer une règle d' EventBridge événement Amazon qui capture uniquement les événements liés aux tâches lorsque le statut de la tâche est passé à un FAILED statut. À la fin de ce didacticiel, vous pouvez également éventuellement soumettre une tâche à cette file d'attente de tâches. Cela permet de vérifier que vous avez correctement configuré vos alertes Amazon SNS.

Prérequis

Ce didacticiel suppose que vous disposez d'un environnement de calcul en fonctionnement et une file d'attente prête à accepter des tâches. Si vous ne disposez pas d'un environnement informatique en cours d'exécution ni d'une file d'attente de tâches à partir desquels capturer les événements, suivez les étapes [Commencer à utiliser les AWS Batch didacticiels](#) décrites pour en créer une.

Rubriques

- [Tutoriel : créer une rubrique Amazon SNS et s'y abonner](#)
- [Tutoriel : Enregistrer une règle d'événement](#)
- [Tutoriel : Testez votre règle](#)

- [Règle alternative : file d'attente des tâches Batch bloquée](#)

Tutoriel : créer une rubrique Amazon SNS et s'y abonner

Pour ce didacticiel, vous configurez une rubrique Amazon SNS à utiliser comme une cible de l'événement pour votre nouvelle règle d'événement.

Pour créer une rubrique Amazon SNS

1. [Ouvrez la console Amazon SNS à l'adresse v3/home. https://console.aws.amazon.com/sns/](https://console.aws.amazon.com/sns/)
2. Choisissez Topics (Rubriques), Create topic (Créer une rubrique).
3. Pour Type, choisissez Standard.
4. Dans le champ Nom, entrez **JobFailedAlert** et choisissez Créer un sujet.
5. Sur l'JobFailedAlertécran, choisissez Créer un abonnement.
6. Pour Protocole, choisissez E-mail.
7. Dans le champ Endpoint (Point de terminaison), saisissez l'adresse e-mail à laquelle vous avez actuellement accès et choisissez Create subscription (Créer un abonnement).
8. Vérifiez votre compte de messagerie et attendez de recevoir un e-mail de confirmation de l'abonnement. Lorsque vous le recevez, choisissez Confirm subscription (Confirmer l'abonnement).

Tutoriel : Enregistrer une règle d'événement

Ensuite, enregistrez une règle d'événement qui capture uniquement les événements d'échec de tâche.

Pour enregistrer votre EventBridge règle

1. Ouvrez la EventBridge console Amazon à l'adresse <https://console.aws.amazon.com/events/>.
2. Dans le volet de navigation, choisissez Règles.
3. Choisissez Créer une règle.
4. Saisissez un nom et une description pour la règle.

Une règle ne peut pas avoir le même nom qu'une autre règle de la même région et sur le même bus d'événement.

5. Pour Event bus (Bus d'événement), sélectionnez le bus d'événement que vous souhaitez associer à cette règle. Si vous souhaitez que cette règle corresponde aux événements provenant de votre compte, sélectionnez Bus d'événements par défaut AWS . Lorsqu'un AWS service de votre compte émet un événement, celui-ci est toujours redirigé vers le bus d'événements par défaut de votre compte.
6. Pour Type de règle, choisissez Règle avec un modèle d'événement.
7. Choisissez Suivant.
8. Pour Event source (Source de l'événement), choisissez Other (Autres).
9. Pour Modèle d'événement, sélectionnez Modèles personnalisés (éditeur JSON).
10. Collez le modèle d'événement suivants dans la zone de texte.

```
{
  "detail-type": [
    "Batch Job State Change"
  ],
  "source": [
    "aws.batch"
  ],
  "detail": {
    "status": [
      "FAILED"
    ]
  }
}
```

Ce code définit une EventBridge règle qui correspond à tout événement correspondant au statut de la tâche `FAILED`. Pour plus d'informations sur les modèles d'événements, consultez la section [Événements et modèles d'événements](#) dans le guide de EventBridge l'utilisateur Amazon.

11. Choisissez Suivant.
12. Pour Types de cibles, choisissez service AWS .
13. Pour Sélectionner une cible, choisissez le sujet SNS, et pour le sujet, choisissez JobFailedAlert.
14. (Facultatif) Pour Additional settings (Paramètres supplémentaires), procédez comme suit :
 - a. Pour Maximum age of event (Âge maximal de l'événement), saisissez une valeur comprise entre une minute (00:01) et 24 heures (24:00).
 - b. Pour Retry attempts (Nouvelles tentatives), saisissez un nombre compris entre 0 et 185.

- c. Pour la file d'attente de lettres mortes, choisissez si vous souhaitez utiliser une file d'attente Amazon SQS standard comme file d'attente de lettres mortes. EventBridge envoie les événements qui correspondent à cette règle à la file d'attente des lettres mortes s'ils ne sont pas correctement transmis à la cible. Effectuez l'une des actions suivantes :
 - Choisissez None (Aucune) pour ne pas utiliser de file d'attente de lettres mortes.
 - Choisissez Sélectionnez une file d'attente Amazon SQS dans le AWS compte courant à utiliser comme file d'attente de lettres mortes, puis sélectionnez la file d'attente à utiliser dans le menu déroulant.
 - Choisissez Sélectionnez une file d'attente Amazon SQS dans un autre AWS compte en tant que file d'attente de lettres mortes, puis entrez l'ARN de la file d'attente à utiliser. Vous devez associer à la file d'attente une politique basée sur les ressources qui EventBridge autorise l'envoi de messages. Pour plus d'informations, consultez la section [Octroi d'autorisations à la file d'attente de lettres mortes](#) dans le guide de EventBridge l'utilisateur Amazon.
15. Choisissez Suivant.
16. (Facultatif) Saisissez une ou plusieurs balises pour la règle. Pour plus d'informations, consultez les [EventBridge balises Amazon](#) dans le guide de EventBridge l'utilisateur Amazon.
17. Choisissez Suivant.
18. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

Tutoriel : Testez votre règle

Pour tester votre règle, soumettre une tâche qui s'arrête peu après avoir été lancée avec un code de sortie différent de 0. Si votre règle d'événement est correctement configurée, vous devriez recevoir un e-mail contenant le texte de l'événement dans les minutes qui suivent.

Pour tester une règle

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Soumettez une nouvelle AWS Batch offre d'emploi. Pour de plus amples informations, veuillez consulter [Tutoriel : soumettre une offre d'emploi](#). Pour la commande de tâche, remplacez cette commande pour quitter le conteneur avec un code de sortie de 1.

```
/bin/sh, -c, 'exit 1'
```

3. Vérifiez vos e-mails pour confirmer que vous avez reçu une alerte par e-mail concernant la notification d'échec de l'offre d'emploi.

Règle alternative : file d'attente des tâches Batch bloquée

Pour créer une règle d'événement qui surveille les files d'attente de tâches par lots bloquées, répétez ces didacticiels avec les modifications suivantes :

1. Dans [Tutoriel : créer une rubrique Amazon SNS et s'y abonner](#), utilisez *BlockedJobQueue* comme nom de rubrique.
2. Dans [Tutoriel : Enregistrer une règle d'événement](#), utilisez le modèle suivant dans l'éditeur JSON :

```
{
  "detail-type": [
    "Batch Job Queue Blocked"
  ],
  "source": [
    "aws.batch"
  ]
}
```

Elastic Fabric Adapter

Un périphérique EFA (Elastic Fabric Adapter) est un périphérique réseau qui permet d'accélérer les applications de calcul haute performance (HPC, High Performance Computing). AWS Batch prend en charge les applications EFA si les conditions suivantes sont réunies.

- Pour obtenir la liste des types d'instances compatibles EFAs, consultez la section [Types d'instances pris en charge](#) dans le guide de EC2 l'utilisateur Amazon.

Tip

Pour voir la liste des types d'instances compatibles EFAs avec un Région AWS, exécutez la commande suivante. Ensuite, croisez la liste renvoyée avec la liste des types d'instances disponibles dans la AWS Batch console.

```
$ aws ec2 describe-instance-types --region us-east-1 --filters Name=network-info.efa-supported,Values=true --query "InstanceTypes[*].[InstanceType]" --output text | sort
```

- Pour obtenir la liste des systèmes d'exploitation compatibles EFA, consultez la section [Systèmes d'exploitation pris en charge](#).
- Le pilote EFA est chargé sur l'AMI.
- Le groupe de sécurité pour l'EFA doit autoriser tout le trafic entrant et sortant vers et depuis le groupe de sécurité proprement dit.
- Toutes les instances qui utilisent un EFA doivent appartenir au même groupe de placement de clusters.
- La définition de la tâche doit inclure un membre `devices` avec `hostPath` défini sur `/dev/infiniband/uverbs0` pour autoriser la transmission par le périphérique EFA jusqu'au conteneur. S'il `containerPath` est spécifié, il doit également être défini sur `/dev/infiniband/uverbs0`. Si `permissions` est spécifié, il doit être défini sur `READ | WRITE | MKNOD`.

L'emplacement des [LinuxParameters](#) membres est différent pour les tâches parallèles à nœuds multiples et pour les tâches de conteneur à nœud unique. Les exemples suivants montrent les différences, mais les valeurs obligatoires sont absentes.

Exemple Exemple de tâche parallèle à plusieurs nœuds

```
{
  "jobDefinitionName": "EFA-MNP-JobDef",
  "type": "multinode",
  "nodeProperties": {
    ...
    "nodeRangeProperties": [
      {
        ...
        "container": {
          ...
          "linuxParameters": {
            "devices": [
              {
                "hostPath": "/dev/infiniband/uverbs0",
                "containerPath": "/dev/infiniband/uverbs0",
                "permissions": [
                  "READ", "WRITE", "MKNOD"
                ]
              },
            ],
          },
        },
      },
    ],
  },
},
}
```

Exemple Exemple de tâche de conteneurs à un seul nœud

```
{
  "jobDefinitionName": "EFA-Container-JobDef",
  "type": "container",
  ...
  "containerProperties": {
    ...
    "linuxParameters": {
      "devices": [
        {
          "hostPath": "/dev/infiniband/uverbs0",
        },
      ],
    },
  },
}
```

```
    ],  
    },  
  },  
}
```

Pour plus d'informations sur l'EFA, consultez [Elastic Fabric Adapter](#) dans le guide de EC2 l'utilisateur Amazon.

Moniteur AWS Batch

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité AWS Batch et des performances de votre AWS solution.

Nous vous encourageons vivement à collecter des données de surveillance provenant de toutes les parties de votre AWS solution afin de faciliter le débogage d'une défaillance multipoint, le cas échéant. Commencez par créer un plan de surveillance qui répond aux questions suivantes. Si vous ne savez pas comment répondre à ces questions, vous pouvez toujours utiliser Amazon CloudWatch Logs pour établir vos références de performance.

- Quels sont les objectifs de la surveillance ?
- Quelles sont les ressources à surveiller ?
- A quelle fréquence les ressources doivent-elles être surveillées ?
- Quels outils de surveillance utiliser ?
- Qui exécute les tâches de supervision ?
- Qui doit être informé en cas de problème ?

L'étape suivante consiste à établir une base de référence de AWS Batch performance normale dans votre environnement en mesurant les performances à différents moments et dans différentes conditions de charge. Pendant que vous surveillez AWS Batch, conservez les données de surveillance historiques afin de pouvoir les comparer aux données de performance actuelles. Cela vous aide à identifier les modèles de performances normaux et les anomalies de performances, et à élaborer des méthodes pour résoudre ces problèmes.

Les rubriques de cette section peuvent vous aider à démarrer la journalisation et la surveillance de AWS Batch.

Rubriques

- [Utilisation CloudWatch des journaux avec AWS Batch](#)
- [AWS Batch CloudWatch Informations sur les conteneurs](#)
- [Utiliser CloudWatch les journaux pour surveiller AWS Batch les tâches Amazon EKS](#)

Utilisation CloudWatch des journaux avec AWS Batch

Vous pouvez configurer vos AWS Batch tâches sur les EC2 ressources pour envoyer des informations de journal et des métriques détaillées à CloudWatch Logs. Ainsi, vous pouvez consulter les différents journaux de vos tâches en un seul endroit pratique. Pour plus d'informations sur CloudWatch les journaux, consultez [Qu'est-ce qu'Amazon CloudWatch Logs ?](#) dans le guide de CloudWatch l'utilisateur Amazon.

Note

Par défaut, CloudWatch Logs est activé pour les conteneurs AWS Fargate.

Pour activer et personnaliser la journalisation CloudWatch des journaux, passez en revue les tâches de configuration ponctuelles suivantes :

- Pour les environnements AWS Batch informatiques basés sur les EC2 ressources, ajoutez une politique IAM au `ecsInstanceRole` rôle. Pour de plus amples informations, veuillez consulter [the section called “Tutoriel : Ajouter une politique CloudWatch Logs IAM”](#).
- Créez un modèle de EC2 lancement Amazon incluant une CloudWatch surveillance détaillée, puis spécifiez le modèle lorsque vous créez votre environnement AWS Batch informatique. Vous pouvez également installer l' CloudWatchagent sur une image existante, puis spécifier l'image dans l'assistant de AWS Batch première exécution.
- (Facultatif) Configurez le pilote awslogs. Vous pouvez ajouter des paramètres qui modifient le comportement par défaut des ressources à la fois EC2 et de Fargate. Pour de plus amples informations, veuillez consulter [the section called “Utiliser le pilote de journal awslogs”](#).

Rubriques

- [Tutoriel : Ajouter une politique CloudWatch Logs IAM](#)
- [Installation et configuration de l' CloudWatch agent](#)
- [Tutoriel : Afficher CloudWatch les journaux](#)

Tutoriel : Ajouter une politique CloudWatch Logs IAM

Avant que vos tâches puissent envoyer des données de journal et des métriques détaillées à CloudWatch Logs, vous devez créer une politique IAM qui utilise les CloudWatch journaux APIs. Après avoir créé la politique IAM, associez-la au `ecsInstanceRole` rôle.

Note

Si la `ECS-CloudWatchLogs` politique n'est pas associée au `ecsInstanceRole` rôle, les métriques de base peuvent toujours être envoyées à CloudWatch Logs. Toutefois, les mesures de base n'incluent pas les données du journal ni les mesures détaillées telles que l'espace disque disponible.

AWS Batch les environnements informatiques utilisent les EC2 ressources Amazon. Lorsque vous créez un environnement informatique à l'aide de l'assistant de AWS Batch première exécution, vous AWS Batch créez le `ecsInstanceRole` rôle et configurez l'environnement à l'aide de celui-ci.

Si vous n'utilisez pas l'assistant de première exécution, vous pouvez spécifier le `ecsInstanceRole` rôle lorsque vous créez un environnement de calcul dans l' AWS Batch API AWS Command Line Interface or. Pour plus d'informations, consultez la [référence de AWS CLI commande](#) ou la [référence AWS Batch d'API](#).

Pour créer la stratégie IAM **ECS-CloudWatchLogs**

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le panneau de navigation, choisissez Politiques.
3. Choisissez Create Policy (Créer une politique).
4. Choisissez JSON, puis entrez la politique suivante :

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```

        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
    ],
    "Resource": [
        "arn:aws:logs:*:*:*"
    ]
}
]
}

```

5. Choisissez Suivant : Balises.
6. (Facultatif) Pour Ajouter des balises, choisissez Ajouter une balise pour ajouter une balise à la politique.
7. Choisissez Suivant : Vérification.
8. Sur la page Révision de la politique, pour Nom **ECS-CloudWatchLogs**, entrez, puis entrez une description facultative.
9. Choisissez Create Policy (Créer une politique).

Pour attacher la politique **ECS-CloudWatchLogs** à **ecsInstanceRole**

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le panneau de navigation, choisissez Rôles.
3. Sélectionnez **ecsInstanceRole**. Si le rôle n'existe pas, suivez les procédures décrites [Rôle d'instance Amazon ECS](#) pour le créer.
4. Choisissez Ajouter des autorisations, puis choisissez Joindre des politiques.
5. Choisissez la CloudWatchLogs stratégie ECS-, puis choisissez Attach policy.

Installation et configuration de l' CloudWatch agent

Vous pouvez créer un modèle de EC2 lancement Amazon qui inclut CloudWatch la surveillance. Pour plus d'informations, consultez [Lancer une instance à partir d'un modèle de lancement](#) et [Informations avancées](#) dans le guide de EC2 l'utilisateur Amazon.

Vous pouvez également installer l' CloudWatch agent sur une EC2 AMI Amazon existante, puis spécifier l'image dans l'assistant de AWS Batch première exécution. Pour plus d'informations,

consultez les [sections Installation de l' CloudWatch agent](#) et [Commencer à utiliser les AWS Batch didacticiels](#).

 Note

Les modèles de lancement ne sont pas pris en charge sur AWS Fargate les ressources.

Tutoriel : Afficher CloudWatch les journaux

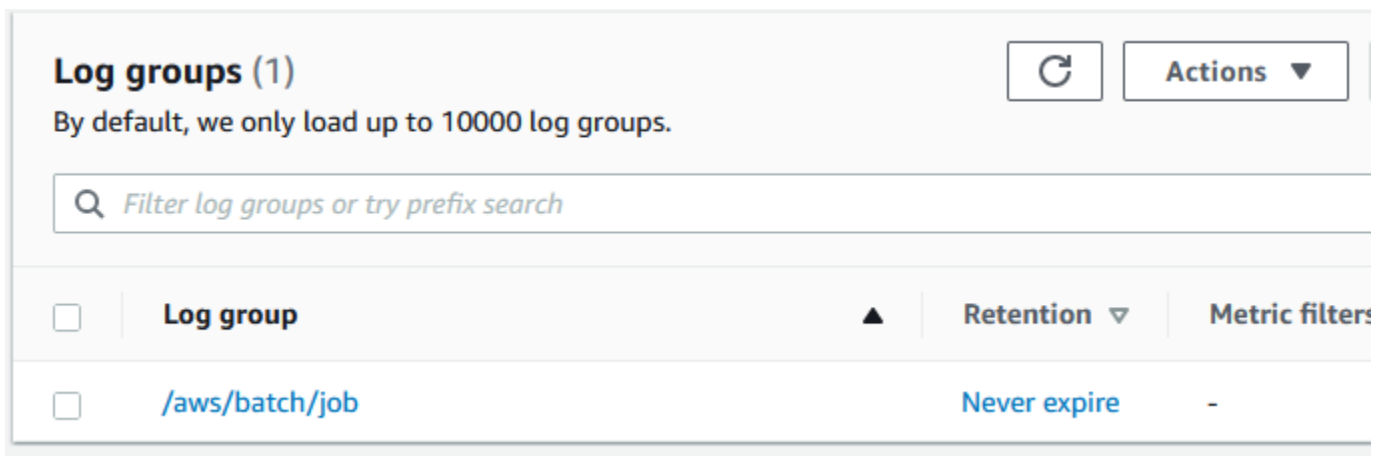
Vous pouvez consulter et rechercher CloudWatch les journaux dans le AWS Management Console.

 Note

L'affichage des données dans les CloudWatch journaux peut prendre quelques minutes.

Pour consulter les données de vos CloudWatch journaux

1. Ouvrez la CloudWatch console à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. Dans le volet de navigation de gauche, choisissez Logs, puis Log groups.



3. Choisissez un groupe de journaux à afficher.

Log streams (9)			Delete	Create log stream	Search all
Filter log streams or try prefix search		< 1 >			
☐	Log stream	Last event time			
☐	Test-jd/default/6622fe43-b2a3-4805-a0a6-3828329cc32b	2020-08-18T19:50:19.311Z			
☐	first-run-job-definition/default/86ed75ac-4f3f-4044-8fb0-dfd9c85ae6b2	2020-08-18T02:07:42.738Z			
☐	Test-jd/default/48f4a9dd-be07-4b43-8696-f0995eefe28b	2020-08-14T00:18:19.395Z			
☐	first-run-job-definition/default/d7d5ccf4-a0a0-44f1-bf36-35f2b3632912	2020-08-13T22:39:06.936Z			
☐	gpuJD/default/6ecf8ffb-ee03-4041-aa18-ab5e7a6dff0d	2019-03-26T08:48:39.637Z			

4. Sélectionnez un flux de journal à afficher. Par défaut, les flux sont identifiés par les 200 premiers caractères du nom de la tâche et par l'ID de tâche Amazon ECS.

Tip

Pour télécharger les données du flux de log, sélectionnez Actions.

Log events

Actions

Create Metric Filter

Filter events

Clear

1m

30m

1h

12h

Custom

Timestamp

Message

There are older events to load. [Load more](#).

2020-08-17T19:07:42.738-07:00...

'hello world'

No newer events at this moment. *Auto retry paused.* [Resume](#)

AWS Batch CloudWatch Informations sur les conteneurs

CloudWatch Container Insights collecte, agrège et résume les métriques et les journaux de vos environnements AWS Batch informatiques et de vos tâches. Les métriques incluent l'utilisation

du processeur, de la mémoire, du disque et du réseau. Vous pouvez ajouter ces indicateurs aux CloudWatch tableaux de bord.

Les données d'exploitation sont collectées en tant qu'événements de journaux de performances. Ces entrées utilisent un schéma JSON structuré qui permet aux données à haute cardinalité d'être intégrées et stockées à grande échelle. À partir de ces données, il CloudWatch crée des métriques agrégées de niveau supérieur au niveau de l'environnement informatique et au niveau du travail sous forme CloudWatch de métriques. Pour plus d'informations, consultez [Container Insights Structured Logs for Amazon ECS](#) dans le guide de CloudWatch l'utilisateur Amazon.

 Important

CloudWatch Les Container Insights sont facturés en tant que métriques personnalisées par CloudWatch. Pour plus d'informations, consultez les [tarifs d'Amazon CloudWatch Events](#)

Rubriques

- [Activer Container Insights](#)

Activer Container Insights

Procédez comme suit pour activer Container Insights pour les environnements AWS Batch informatiques.

1. Ouvrez la [AWS Batch console](#).
2. Choisissez Environnements.
3. Choisissez l'environnement informatique de votre choix.
4. Dans l'onglet Container Insights, activez Container Insights pour l'environnement informatique.

 Tip

Vous pouvez sélectionner un intervalle par défaut pour agréger les mesures ou créer un intervalle personnalisé.

Par défaut, les mesures suivantes sont affichées. Pour obtenir la liste complète des métriques Amazon ECS Container Insights, consultez [Amazon ECS Container Insights Metrics](#) dans le guide de CloudWatch l'utilisateur Amazon.

- **JobCount**— Le nombre de tâches exécutées dans l'environnement informatique.
- **ContainerInstanceCount**— Le nombre d'instances Amazon Elastic Compute Cloud qui exécutent l'agent Amazon ECS et sont enregistrées dans l'environnement informatique.
- **MemoryReserved**— Mémoire réservée par les tâches de l'environnement informatique. Cette métrique est collectée uniquement pour les tâches dont la définition de tâche comporte une réservation de mémoire définie.
- **MemoryUtilized**— La mémoire utilisée par les tâches de l'environnement informatique. Cette métrique est collectée uniquement pour les tâches dont la définition de tâche comporte une réservation de mémoire définie.
- **CpuReserved**— Les unités de processeur réservées par les tâches de l'environnement de calcul. Cette métrique est collectée uniquement pour les tâches dont la définition de tâche comporte une réservation de processeur définie.
- **CpuUtilized**— Les unités de processeur utilisées par les tâches dans l'environnement informatique. Cette métrique est collectée uniquement pour les tâches dont la définition de tâche comporte une réservation de processeur définie.
- **NetworkRxBytes**- Le nombre d'octets reçus. Cette métrique n'est disponible que pour les conteneurs dans les tâches qui utilisent le mode réseau awsvpc ou bridge.
- **NetworkTxBytes**— Le nombre d'octets transmis. Cette métrique n'est disponible que pour les conteneurs dans les tâches qui utilisent le mode réseau awsvpc ou bridge.
- **StorageReadBytes**— Le nombre d'octets lus depuis le stockage.
- **StorageWriteBytes**— Le nombre d'octets écrits dans le stockage.

Utiliser CloudWatch les journaux pour surveiller AWS Batch les tâches Amazon EKS

Vous pouvez utiliser Amazon CloudWatch Logs pour surveiller, stocker et consulter tous vos fichiers journaux au même endroit. À l'aide CloudWatch des journaux, vous pouvez rechercher, filtrer et analyser les données des journaux provenant de plusieurs sources.

Vous pouvez télécharger un AWS formulaire Fluent Bit image qui inclut un plugin permettant de surveiller AWS Batch les tâches Amazon EKS dans CloudWatch Logs. Fluent Bit est un processeur

de journal et un redirecteur open source qui est à la fois Docker et Kubernetes compatible. Nous vous recommandons d'utiliser Fluent Bit comme routeur de journalisation, car il consomme moins de ressources que Fluentd. Pour plus d'informations, consultez [Installer l' CloudWatch agent avec le module complémentaire Amazon CloudWatch Observability EKS ou le graphique Helm](#).

Prérequis

- Associez la CloudWatchAgentServerPolicy politique à AWS Identity and Access Management celle de vos nœuds de travail. Pour plus d'informations, voir [Vérifier les prérequis](#).

Installez le module complémentaire

Pour obtenir des instructions sur l'installation AWS de Fluent Bit et créez les CloudWatch groupes, consultez [Installer l' CloudWatch agent avec le module complémentaire Amazon CloudWatch Observability EKS ou le graphique Helm](#).

Vous devez fournir les [données de configuration supplémentaires](#) suivantes lors de l'installation du module complémentaire :

- Si vous installez le module complémentaire avec le, AWS Management Console vous devez fournir les tolérances suivantes dans les valeurs de configuration :

```
{
  "tolerations": [
    {
      "key": "batch.amazonaws.com/batch-node",
      "operator": "Exists"
    }
  ]
}
```

- Si vous installez le module complémentaire avec le AWS CLI , ajoutez les arguments suivants :

```
--configuration-values '{"tolerations":[{"key":"batch.amazonaws.com/batch-node","operator":"Exists"}]}'
```


 Tip

N'oubliez pas que Fluent Bit utilise 0,5 processeur et 100 Mo de mémoire sur les AWS Batch nœuds. Cela réduit la capacité totale disponible pour les AWS Batch emplois. Tenez-en compte lorsque vous déterminez la taille de vos tâches.

Marquez vos AWS Batch ressources

Pour vous aider à gérer vos AWS Batch ressources, vous pouvez attribuer vos propres métadonnées à chaque ressource sous forme de balises. Cette rubrique décrit les identifications et explique comment les créer.

Rubriques

- [Principes de base des étiquettes](#)
- [Baliser vos ressources](#)
- [Restrictions liées aux étiquettes](#)
- [Tutoriel : Gérer les tags à l'aide de la console](#)
- [Gérez les balises à l'aide de la CLI ou de l'API](#)

Principes de base des étiquettes

Une étiquette est une étiquette que vous attribuez à une AWS ressource. Chaque balise est constituée d'une clé et d'une valeur facultative que vous définissez.

Les balises vous permettent de classer vos AWS ressources en fonction, par exemple, de leur objectif, de leur propriétaire ou de leur environnement. Lorsque vous avez de nombreuses ressources de même type, vous pouvez rapidement identifier une ressource spécifique en fonction des balises que vous lui avez attribuées. Par exemple, vous pouvez définir un ensemble de balises pour vos AWS Batch services afin de suivre le propriétaire et le niveau de pile de chaque service. Nous vous recommandons de concevoir un ensemble cohérent de clés de balise pour chaque type de ressource.

Les balises ne sont pas automatiquement affectées à vos ressources. Une fois que vous avez ajouté une balise, vous pouvez modifier les clés et valeurs de balise ou supprimer les balises d'une ressource à tout moment. Si vous supprimez une ressource, ses balises sont également supprimées.

Les balises n'ont aucune signification sémantique AWS Batch et sont interprétées strictement comme des chaînes de caractères. Vous pouvez définir la valeur d'une balise sur une chaîne vide, mais vous ne pouvez pas définir la valeur d'une balise sur null. Si vous ajoutez une balise ayant la même clé qu'une balise existante sur cette ressource, la nouvelle valeur remplace l'ancienne valeur.

Vous pouvez travailler avec des balises à l'aide de l'API AWS Management Console AWS CLI, du et de l' AWS Batch API.

Si vous utilisez AWS Identity and Access Management (IAM), vous pouvez contrôler quels utilisateurs de votre AWS compte sont autorisés à créer, modifier ou supprimer des tags.

Baliser vos ressources

Vous pouvez étiqueter des environnements AWS Batch informatiques, des tâches, des définitions de tâches, des files d'attente de tâches et des politiques de planification nouveaux ou existants.

Si vous utilisez la AWS Batch console, vous pouvez appliquer des balises aux nouvelles ressources lors de leur création ou aux ressources existantes à tout moment à l'aide de l'onglet Tags de la page de ressources correspondante.

Si vous utilisez l' AWS Batch API, le ou un AWS SDK AWS CLI, vous pouvez appliquer des balises aux nouvelles ressources à l'aide du tags paramètre de l'action d'API correspondante ou aux ressources existantes à l'aide de l'action d'TagResourceAPI. Pour de plus amples informations, veuillez consulter [TagResource](#).

En outre, certaines actions de création de ressources vous permettent de spécifier des balises pour une ressource lors de la création de cette dernière. Si des balises ne peuvent pas être appliquées au cours de la création de ressources, le processus de création de ressources échoue. Cela garantit que les ressources que vous vouliez baliser lors de la création sont créées avec des balises spécifiées ou ne sont pas créées du tout. Si vous balisez des ressources au moment de la création, vous n'avez pas besoin d'exécuter de scripts de balisage personnalisés après la création des ressources.

Le tableau suivant décrit les AWS Batch ressources qui peuvent être balisées et les ressources qui peuvent être balisées lors de leur création.

Tag : support pour les AWS Batch ressources

Ressource	Prend en charge les étiquettes	Prend en charge la propagation des étiquettes	Supporte le balisage lors de la création (AWS Batch API AWS CLI, AWS SDK)
AWS Batch environnements informatiques	Oui	Non. Les balises d'environnement de calcul ne se propagent à aucune autre ressource. Les	Oui

Ressource	Prend en charge les étiquettes	Prend en charge la propagation des étiquettes	Supporte le balisage lors de la création (AWS Batch API AWS CLI, AWS SDK)
		balises des ressources sont spécifiées dans les balises membres de l'objet ComputerResources transmises lors de l'opération d' CreateComputeEnvironment API.	
AWS Batch emplois	Oui	Oui	Oui
AWS Batch définitions de poste	Oui	Non	Oui
AWS Batch files d'attente pour les emplois	Oui	Non	Oui
AWS Batch politiques de planification	Oui	Non	Oui

Restrictions liées aux étiquettes

Les restrictions de base suivantes s'appliquent aux balises :

- Nombre maximal de balises par ressource : 50
- Pour chaque ressource, chaque clé de balise doit être unique, et chaque clé de balise peut avoir une seule valeur.
- Longueur de clé maximale : 128 caractères Unicode en UTF-8
- Longueur de valeur maximale : 256 caractères Unicode en UTF-8
- Si votre schéma de balisage est utilisé pour plusieurs AWS services et ressources, n'oubliez pas que d'autres services peuvent imposer des restrictions quant aux caractères autorisés. Les

caractères généralement autorisés sont les lettres, les chiffres et les espaces représentables en UTF-8, ainsi que les caractères suivants : + - = . _ : / @.

- Les clés et valeurs de balise sont sensibles à la casse.
- N'utilisez pas `aws:AWS:`, ni aucune combinaison majuscules ou minuscules, comme un préfixe pour les clés ou les valeurs, car il est réservé à l'usage. AWS Vous ne pouvez pas modifier ni supprimer des clés ou valeurs d'étiquette ayant ce préfixe. Les balises comportant ce préfixe ne sont pas prises en compte dans votre tags-per-resource limite.

Tutoriel : Gérer les tags à l'aide de la console

À l'aide de la AWS Batch console, vous pouvez gérer les balises associées aux environnements informatiques, aux tâches, aux définitions de tâches et aux files d'attente de tâches nouveaux ou existants.

Ajouter des balises sur une ressource individuelle lors de sa création

Vous pouvez ajouter des balises aux environnements AWS Batch informatiques, aux tâches, aux définitions de tâches, aux files d'attente de tâches et aux politiques de planification lorsque vous les créez.

Ajouter et supprimer des balises pour une ressource individuelle

AWS Batch vous permet d'ajouter ou de supprimer des balises associées à vos clusters directement depuis la page de la ressource.

Pour ajouter ou supprimer une balise sur une ressource individuelle

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation, choisissez la région à utiliser.
3. Dans le volet de navigation, choisissez un type de ressource (par exemple, Job Queues).
4. Choisissez une ressource spécifique, puis choisissez Modifier les balises.
5. Ajoutez ou supprimez vos tags selon les besoins.
 - Pour ajouter une balise, spécifiez la clé et la valeur dans les zones de texte vides à la fin de la liste.
 - Pour supprimer un tag,

Delete icon

cliquez sur le bouton situé à côté du tag.

- Répétez cette procédure pour chaque balise que vous souhaitez ajouter ou supprimer, puis choisissez Modifier les balises pour terminer.

Gérez les balises à l'aide de la CLI ou de l'API

Utilisez les AWS CLI commandes ou opérations d' AWS Batch API suivantes pour ajouter, mettre à jour, répertorier et supprimer les balises de vos ressources.

Tag : support pour les AWS Batch ressources

Tâche	Action d'API	AWS CLI	AWS Tools for Windows PowerShell
Ajouter ou remplacer une ou plusieurs étiquettes.	TagResource	tag-resource	Ajouter une BATResource étiquette
Supprimer une ou plusieurs étiquettes.	UntagResource	untag-resource	Supprimer- BATResource Tag
Répertorie les balises d'une ressource.	ListTagsForResource	list-tags-for-resource	Obtenir le BATResource tag

Les exemples suivants montrent comment ajouter ou supprimer les étiquettes d'une ressource à l'aide de l' AWS CLI.

Exemple 1 : Baliser une ressource existante

La commande suivante permet de baliser une ressource existante.

```
aws batch tag-resource --resource-arn resource_ARN --tags team=devs
```

Exemple 2 : Supprimer la balise d'une ressource existante

La commande suivante permet de supprimer une balise d'une ressource existante.

```
aws batch untag-resource --resource-arn resource_ARN --tag-keys tag_key
```

Exemple 3 : Afficher la liste des balises d'une ressource

La commande suivante permet de répertorier l'ensemble des étiquettes associées à une ressource existante.

```
aws batch list-tags-for-resource --resource-arn resource_ARN
```

Certaines actions de création de ressources vous permettent de spécifier des étiquettes lorsque vous créez la ressource. Les actions suivantes prennent en charge l'identification lors de la création.

Tâche	Action d'API	AWS CLI	AWS Tools for Windows PowerShell
Création d'un environnement informatique	CreateComputeEnvironment	create-compute-environment	Nouvel BATCompute environnement
Création d'une file d'attente de tâches	CreateJobQueue	create-job-queue	Nouveau- BATJob File d'attente
Création d'une politique de planification	CreateSchedulingPolicy	create-scheduling-policy	Nouveau- BATScheduling Politique
Enregistrer une définition de tâche	RegisterJobDefinition	register-job-definition	S'inscrire - BATJob Définition
Envoi d'une tâche	SubmitJob	soumet-job	Soumettre- BATJob
Création d'une ressource consommable	CreateConsumableResource	create-consumable-resource	Créer une BATConsumable ressource

Les meilleures pratiques pour AWS Batch

Vous pouvez l'utiliser AWS Batch pour exécuter diverses charges de travail de calcul exigeantes à grande échelle sans avoir à gérer une architecture complexe. AWS Batch les jobs peuvent être utilisés dans un large éventail de cas d'utilisation dans des domaines tels que l'épidémiologie, les jeux vidéo et l'apprentissage automatique.

Cette rubrique décrit les meilleures pratiques à prendre en compte lors de l'utilisation AWS Batch et fournit des conseils sur la manière d'exécuter et d'optimiser vos charges de travail lors de l'utilisation AWS Batch.

Rubriques

- [Quand utiliser AWS Batch](#)
- [Liste de contrôle à exécuter à grande échelle](#)
- [Optimisez les conteneurs et AMIs](#)
- [Choisissez la bonne ressource d'environnement informatique](#)
- [Amazon EC2 On-Demand ou Amazon EC2 Spot](#)
- [Utilisez les meilleures pratiques d'Amazon EC2 Spot pour AWS Batch](#)
- [Erreurs courantes et résolution des problèmes](#)

Quand utiliser AWS Batch

AWS Batch exécute des tâches à grande échelle et à faible coût, et fournit des services de mise en file d'attente et une mise à l'échelle optimisée en termes de coûts. Cependant, toutes les charges de travail ne sont pas adaptées à l'exécution AWS Batch.

- **Tâches courtes** : si une tâche ne s'exécute que quelques secondes, la planification de la tâche par lots peut prendre plus de temps que l'exécution de la tâche elle-même. Pour contourner le problème, binpack vos tâches ensemble avant de les soumettre AWS Batch. Configurez ensuite vos AWS Batch tâches pour qu'elles soient itérées sur les tâches. Par exemple, placez les arguments de tâche individuels dans une table Amazon DynamoDB ou sous forme de fichier dans un compartiment Amazon S3. Envisagez de regrouper les tâches de manière à ce qu'elles s'exécutent de 3 à 5 minutes chacune. Après avoir binpack les tâches, parcourez les groupes de tâches au sein de votre AWS Batch tâche.

- Les tâches qui doivent être exécutées immédiatement AWS Batch peuvent être traitées rapidement. Cependant, il AWS Batch s'agit d'un planificateur qui optimise les performances en termes de coûts, de priorité des tâches et de débit. AWS Batch le traitement de vos demandes peut prendre du temps. Si vous avez besoin d'une réponse en moins de quelques secondes, une approche basée sur les services utilisant Amazon ECS ou Amazon EKS est plus adaptée.

Liste de contrôle à exécuter à grande échelle

Avant d'exécuter une charge de travail importante sur 50 000 v ou plus CPUs, considérez la liste de contrôle suivante.

Note

Si vous prévoyez d'exécuter une charge de travail importante sur un million de V ou plus CPUs ou si vous avez besoin de conseils pour exécuter à grande échelle, contactez votre AWS équipe.

- Vérifiez vos EC2 quotas Amazon — Vérifiez vos EC2 quotas Amazon (également appelés limites) dans le panneau Service Quotas du AWS Management Console. Si nécessaire, demandez une augmentation du quota pour votre nombre maximal d' EC2 instances Amazon. N'oubliez pas que les instances Amazon EC2 Spot et Amazon On-Demand ont des quotas distincts. Pour plus d'informations, consultez [Getting started with Service Quotas](#).
- Vérifiez votre quota Amazon Elastic Block Store pour chaque région : chaque instance utilise un GP3 volume GP2 ou pour le système d'exploitation. Par défaut, le quota pour chacun Région AWS est de 300 TiB. Cependant, chaque instance utilise des nombres dans le cadre de ce quota. Assurez-vous donc d'en tenir compte lorsque vous vérifiez votre quota Amazon Elastic Block Store pour chaque région. Si votre quota est atteint, vous ne pouvez pas créer d'autres instances. Pour plus d'informations, consultez la section [Points de terminaison et quotas Amazon Elastic Block Store](#)
- Utilisez Amazon S3 pour le stockage : Amazon S3 fournit un débit élevé et permet d'éliminer les incertitudes quant à la quantité de stockage à allouer en fonction du nombre de tâches et d'instances dans chaque zone de disponibilité. Pour plus d'informations, consultez la section [Modèles de conception des meilleures pratiques : optimisation des performances d'Amazon S3](#).
- Rédimensionnez progressivement pour identifier les goulets d'étranglement à un stade précoce : pour une tâche exécutée sur un million de volts ou plus CPUs, commencez par le bas puis

augmentez progressivement afin de pouvoir identifier les goulets d'étranglement à un stade précoce. Par exemple, commencez par exécuter sur 50 000 CPUs v. Ensuite, augmentez le nombre à 200 000 vCPUs, puis à 500 000 vCPUs, et ainsi de suite. En d'autres termes, continuez à augmenter progressivement le nombre de vCPU jusqu'à atteindre le nombre de vCPU souhaité. CPUs

- Surveillez pour identifier rapidement les problèmes potentiels : pour éviter les pannes et les problèmes potentiels lors de l'exécution à grande échelle, veillez à surveiller à la fois votre application et votre architecture. Des ruptures peuvent se produire même lors d'une mise à l'échelle de 1 000 à 5 000 CPUs v. Vous pouvez utiliser Amazon CloudWatch Logs pour consulter les données des journaux ou utiliser des métriques CloudWatch intégrées à l'aide d'une bibliothèque cliente. Pour plus d'informations, consultez les sections [Référence de l'agent CloudWatch Logs](#) et [aws-embedded-metrics](#)

Optimisez les conteneurs et AMIs

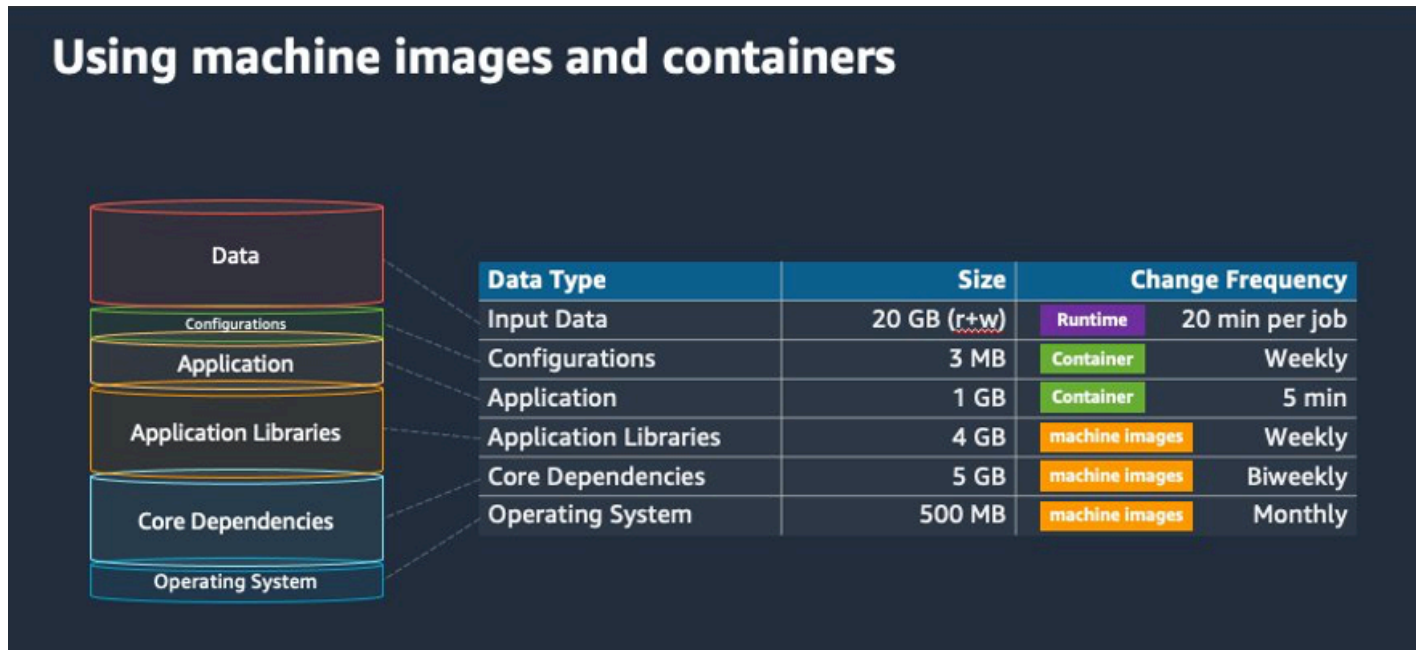
La taille et la structure du conteneur sont importantes pour le premier ensemble de tâches que vous exécutez. Cela est particulièrement vrai si le conteneur est supérieur à 4 Go. Les images de conteneur sont intégrées en couches. Les couches sont récupérées en parallèle par Docker à l'aide de trois threads simultanés. Vous pouvez augmenter le nombre de threads simultanés à l'aide du `max-concurrent-downloads` paramètre. Pour plus d'informations, consultez la documentation [Dockerd](#).

Bien que vous puissiez utiliser des conteneurs plus grands, nous vous recommandons d'optimiser la structure et la taille des conteneurs pour accélérer les temps de démarrage.

- Les petits conteneurs sont récupérés plus rapidement — Les petits conteneurs peuvent accélérer le démarrage des applications. Pour réduire la taille du conteneur, déchargez les bibliothèques ou les fichiers rarement mis à jour vers Amazon Machine Image (AMI). Vous pouvez également utiliser des supports de liaison pour donner accès à vos conteneurs. Pour plus d'informations, consultez la section [Lier les montages](#).
- Créez des couches de taille uniforme et divisez les grandes couches : chaque couche est récupérée par un thread. Une couche importante peut donc avoir un impact significatif sur le temps de démarrage de votre tâche. Nous recommandons une taille de couche maximale de 2 Go comme bon compromis entre une taille de conteneur plus grande et des temps de démarrage plus rapides. Vous pouvez exécuter la `docker history your_image_id` commande pour vérifier la

structure de votre image de conteneur et la taille de la couche. Pour plus d'informations, consultez la [documentation Docker](#).

- Utilisez Amazon Elastic Container Registry comme référentiel de conteneurs : lorsque vous exécutez des milliers de tâches en parallèle, un référentiel autogéré peut échouer ou ralentir le débit. Amazon ECR fonctionne à grande échelle et peut gérer des charges de travail allant jusqu'à plus d'un million de v. CPUs



Choisissez la bonne ressource d'environnement informatique

AWS Fargate nécessite moins d'installation et de configuration initiales qu'Amazon EC2 et est probablement plus facile à utiliser, en particulier si c'est la première fois. Avec Fargate, vous n'avez pas besoin de gérer les serveurs, de gérer la planification de la capacité ou d'isoler les charges de travail des conteneurs pour des raisons de sécurité.

Si vous avez les exigences suivantes, nous vous recommandons d'utiliser les instances Fargate :

- Vos tâches doivent démarrer rapidement, en particulier en moins de 30 secondes.
- Les exigences de vos tâches sont de 16 v CPUs ou moins, non GPUs, et de 120 GiB de mémoire ou moins.

Pour de plus amples informations, veuillez consulter [Quand utiliser Fargate](#).

Si vous répondez aux exigences suivantes, nous vous recommandons d'utiliser des EC2 instances Amazon :

- Vous avez besoin d'un contrôle accru sur la sélection des instances ou de l'utilisation de types d'instances spécifiques.
- Vos tâches nécessitent des ressources qui AWS Fargate ne sont pas en mesure de fournir GPUs, telles que de la mémoire supplémentaire, une AMI personnalisée ou l'adaptateur Amazon Elastic Fabric.
- Vous avez besoin d'un haut niveau de débit ou de simultanéité.
- Vous devez personnaliser votre AMI, votre modèle Amazon EC2 Launch ou accéder à des paramètres Linux spéciaux.

Avec Amazon EC2, vous pouvez ajuster plus finement votre charge de travail en fonction de vos besoins spécifiques et l'exécuter à grande échelle si nécessaire.

Amazon EC2 On-Demand ou Amazon EC2 Spot

La plupart des AWS Batch clients utilisent les instances Amazon EC2 Spot en raison des économies réalisées par rapport aux instances à la demande. Toutefois, si votre charge de travail s'exécute pendant plusieurs heures et ne peut pas être interrompue, les instances à la demande sont peut-être plus adaptées à vos besoins. Vous pouvez toujours essayer d'abord les instances Spot et passer à On-Demand si nécessaire.

Si vous avez les exigences et les attentes suivantes, utilisez les instances Amazon EC2 On-Demand :

- L'exécution de vos tâches est supérieure à une heure et vous ne pouvez tolérer aucune interruption de votre charge de travail.
- Vous avez un SLO (objectif de niveau de service) strict pour votre charge de travail globale et vous ne pouvez pas augmenter le temps de calcul.
- Les instances dont vous avez besoin sont plus susceptibles de connaître des interruptions.

Si vous avez les exigences et les attentes suivantes, utilisez les instances Amazon EC2 Spot :

- La durée d'exécution de vos tâches est généralement de 30 minutes ou moins.

- Vous pouvez tolérer les interruptions potentielles et le report des tâches dans le cadre de votre charge de travail. Pour plus d'informations, consultez [Spot Instance Advisor](#).
- Les tâches de longue durée peuvent être redémarrées depuis un point de contrôle en cas d'interruption.

Vous pouvez combiner les deux modèles d'achat en soumettant d'abord sur une instance Spot, puis en utilisant l'instance On-Demand comme option de secours. Par exemple, soumettez vos tâches sur une file d'attente connectée à des environnements informatiques exécutés sur des instances Amazon EC2 Spot. Si une tâche est interrompue, repérez l'événement sur Amazon EventBridge et corrélerez-le à une réclamation d'instance Spot. Soumettez ensuite à nouveau le travail à une file d'attente à la demande à l'aide d'une AWS Lambda fonction ou AWS Step Functions. Pour plus d'informations [Tutoriel : envoi d'alertes Amazon Simple Notification Service en cas d'échec d'une tâche](#), consultez les [meilleures pratiques pour gérer les interruptions des instances Amazon EC2 Spot et Manage AWS Batch with Step Functions](#).

Important

Utilisez différents types d'instances, tailles et zones de disponibilité pour votre environnement de calcul à la demande afin de maintenir la disponibilité du pool d'instances Amazon EC2 Spot et de réduire le taux d'interruption.

Utilisez les meilleures pratiques d'Amazon EC2 Spot pour AWS Batch

Lorsque vous choisissez les instances Amazon Elastic Compute Cloud (EC2) Spot, vous pouvez probablement optimiser votre flux de travail afin de réduire les coûts, parfois de manière significative. Pour plus d'informations, consultez [Bonnes pratiques pour Amazon EC2 Spot](#).

Pour optimiser votre flux de travail afin de réduire les coûts, prenez en compte les bonnes pratiques Amazon EC2 Spot suivantes pour AWS Batch :

- Choisissez la stratégie **SPOT_CAPACITY_OPTIMIZED** d'allocation : AWS Batch choisit les EC2 instances Amazon parmi les pools de capacités Amazon EC2 Spot les plus complets. Si les interruptions vous inquiètent, c'est un choix approprié. Pour de plus amples informations, veuillez consulter [Stratégies d'allocation des types d'instance pour AWS Batch](#).

- Diversifier les types d'instances : pour diversifier vos types d'instances, considérez les tailles et les familles compatibles, puis laissez le AWS Batch choisir en fonction du prix ou de la disponibilité. Par exemple, considérez `c5.24xlarge` comme une alternative à `c5.12xlarge` ou `c5a`, `c5n`, `c5dm5`, et `m5d` aux familles. Pour plus d'informations, voir [Faire preuve de flexibilité en ce qui concerne les types d'instances et les zones de disponibilité](#).
- Réduisez le temps d'exécution ou le point de contrôle des tâches : nous vous déconseillons d'exécuter des tâches qui prennent une heure ou plus lorsque vous utilisez des instances Amazon EC2 Spot afin d'éviter les interruptions. Si vous divisez ou contrôlez vos tâches en plus petites parties d'une durée de 30 minutes ou moins, vous pouvez réduire considérablement les risques d'interruptions.
- Utiliser des tentatives automatisées : pour éviter d'interrompre les AWS Batch tâches, configurez des tentatives automatisées pour les tâches. Les tâches par lots peuvent être interrompues pour l'une des raisons suivantes : un code de sortie différent de zéro est renvoyé, une erreur de service se produit ou une restauration d'instance se produit. Vous pouvez configurer jusqu'à 10 tentatives automatisées. Pour commencer, nous vous recommandons de définir au moins 1 à 3 nouvelles tentatives automatiques. Pour plus d'informations sur le suivi des interruptions Amazon EC2 Spot, consultez le tableau de [bord des interruptions ponctuelles](#).

En AWS Batch effet, si vous définissez le paramètre de nouvelle tentative, la tâche est placée en tête de la file d'attente des tâches. C'est-à-dire que le travail est prioritaire. Lorsque vous créez la définition de tâche ou que vous soumettez la tâche dans le AWS CLI, vous pouvez configurer une stratégie de nouvelle tentative. Pour plus d'informations, consultez [submit-job](#).

```
$ aws batch submit-job --job-name MyJob \  
  --job-queue MyJQ \  
  --job-definition MyJD \  
  --retry-strategy attempts=2
```

- Utiliser des tentatives personnalisées : vous pouvez configurer une stratégie de relance des tâches en fonction d'un code de sortie d'application ou d'une restauration d'instance spécifique. Dans l'exemple suivant, si l'hôte est à l'origine de l'échec, la tâche peut être réessayée jusqu'à cinq fois. Toutefois, si la tâche échoue pour une autre raison, elle se termine et son statut est défini sur **FAILED**.

```
"retryStrategy": {  
  "attempts": 5,  
  "evaluateOnExit":  
    [{
```

```
    "onStatusReason" : "Host EC2*",
    "action": "RETRY"
  }, {
    "onReason" : "*",
    "action": "EXIT"
  }
}
```

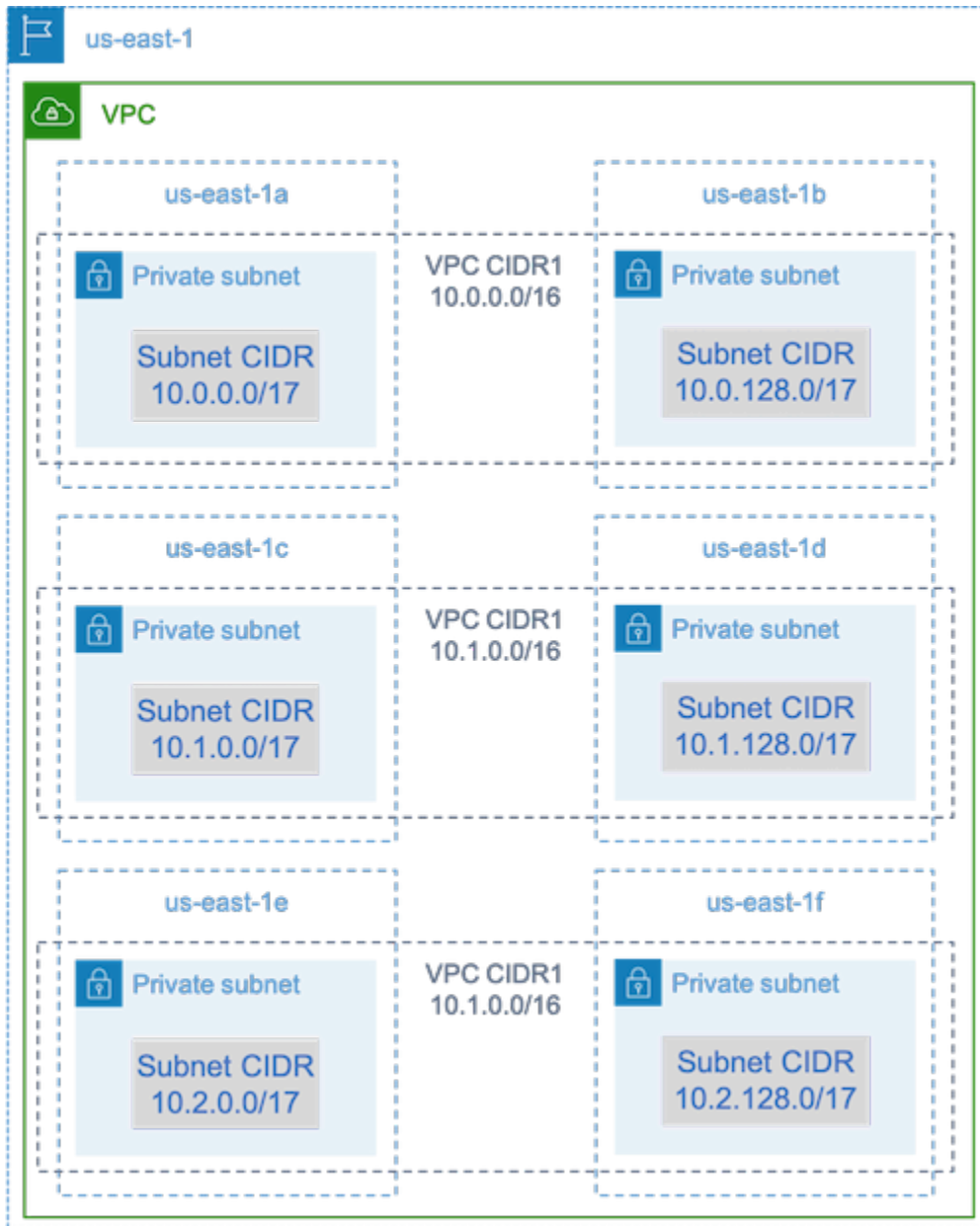
- Utilisez le tableau de bord des interruptions ponctuelles : vous pouvez utiliser le tableau de bord des interruptions ponctuelles pour suivre les interruptions ponctuelles. L'application fournit des statistiques sur les instances Amazon EC2 Spot qui sont récupérées et sur les zones de disponibilité dans lesquelles se trouvent les instances Spot. Pour plus d'informations, voir [Tableau de bord Spot Interruption](#)

Erreurs courantes et résolution des problèmes

Les erreurs se produisent AWS Batch souvent au niveau de l'application ou sont causées par des configurations d'instance qui ne répondent pas aux exigences spécifiques de votre tâche. Parmi les autres problèmes, citons le blocage des tâches dans le `RUNNABLE` statut ou le blocage des environnements informatiques dans un `INVALID` état. Pour plus d'informations sur la résolution des problèmes liés au blocage `RUNNABLE` des tâches, consultez [Offres d'emploi bloquées dans un `RUNNABLE` statut](#). Pour plus d'informations sur le dépannage des environnements informatiques dans un `INVALID` état, consultez [INVALID environnement informatique](#).

- Vérifiez les quotas de vCPU Amazon EC2 Spot : vérifiez que vos quotas de service actuels répondent aux exigences du poste. Supposons, par exemple, que votre quota de service actuel soit de 256 V CPUs et que la tâche nécessite 10 000 CPUs V. Dans ce cas, le quota de service ne répond pas aux exigences du poste. Pour plus d'informations et des instructions de résolution des problèmes, consultez les [sections Quotas de EC2 service Amazon](#) et [Comment augmenter le quota de service de mon Amazon EC2resources ?](#).
- Les tâches échouent avant l'exécution de l'application : certaines tâches peuvent échouer en raison d'une `DockerTimeoutError` erreur ou d'une `CannotPullContainerError` erreur. Pour plus d'informations sur le dépannage, voir [Comment résoudre l'erreur DockerTimeoutError « » dans AWS Batch ?](#).
- Adresses IP insuffisantes : le nombre d'adresses IP de votre VPC et de vos sous-réseaux peut limiter le nombre d'instances que vous pouvez créer. Utilisez Classless Inter-Domain Routings (CIDRs) pour fournir plus d'adresses IP que ce qui est nécessaire pour exécuter vos charges

de travail. Si nécessaire, vous pouvez également créer un VPC dédié avec un grand espace d'adressage. Par exemple, vous pouvez créer un VPC avec plusieurs entrées $10.x.0.0/16$ et un sous-réseau CIDRs dans chaque zone de disponibilité avec un CIDR de $10.x.y.0/17$. Dans cet exemple, x est compris entre 1 et 4 et y est égal à 0 ou 128. Cette configuration fournit 36 000 adresses IP dans chaque sous-réseau.



- Vérifiez que les instances sont enregistrées auprès d'Amazon EC2 — Si vous voyez vos instances dans la EC2 console Amazon, mais aucune instance de conteneur Amazon Elastic Container Service dans votre cluster Amazon ECS, l'agent Amazon ECS n'est peut-être pas installé sur une Amazon Machine Image (AMI). L'agent Amazon ECS, les EC2 données Amazon de votre AMI ou le modèle de lancement peuvent également ne pas être correctement configurés. Pour isoler

la cause première, créez une EC2 instance Amazon distincte ou connectez-vous à une instance existante via SSH. Pour plus d'informations, consultez les [sections Configuration de l'agent de conteneur Amazon ECS](#), [Emplacements des fichiers journaux Amazon ECS](#) et [Ressource de calcul AMIs](#).

- Passez en revue le AWS tableau de AWS bord : examinez le tableau de bord pour vérifier que la tâche attendue se présente et que l'environnement de calcul évolue comme prévu. Vous pouvez également consulter les connexions aux tâches CloudWatch.
- Vérifiez que votre instance est créée : si une instance est créée, cela signifie que votre environnement informatique a été dimensionné comme prévu. Si vos instances ne sont pas créées, recherchez les sous-réseaux associés à modifier dans votre environnement informatique. Pour plus d'informations, consultez [Vérifier une activité de dimensionnement pour un groupe Auto Scaling](#).

Nous vous recommandons également de vérifier que vos instances répondent aux exigences professionnelles associées. Par exemple, une tâche peut nécessiter 1 TiB de mémoire, mais l'environnement de calcul utilise un type d'instance C5 limité à 192 Go de mémoire.

- Vérifiez que vos instances sont demandées par AWS Batch — Consultez l'historique du groupe Auto Scaling pour vérifier que vos instances sont demandées par AWS Batch. Cela indique comment Amazon EC2 essaie d'acquérir des instances. Si vous recevez un message d'erreur indiquant qu'Amazon EC2 Spot ne peut pas acquérir une instance dans une zone de disponibilité spécifique, cela peut être dû au fait que la zone de disponibilité ne propose pas de famille d'instances spécifique.
- Vérifiez que les instances sont enregistrées auprès d'Amazon ECS — Si vous voyez des instances dans la EC2 console Amazon, mais aucune instance de conteneur Amazon ECS dans votre cluster Amazon ECS, il est possible que l'agent Amazon ECS ne soit pas installé sur l'Amazon Machine Image (AMI). De plus, il est possible que l'agent Amazon ECS, EC2 les données Amazon de votre AMI ou le modèle de lancement ne soient pas correctement configurés. Pour isoler la cause première, créez une EC2 instance Amazon distincte ou connectez-vous à une instance existante via SSH. Pour plus d'informations, consultez [le fichier de configuration de l'CloudWatch agent : section Journaux](#), [Emplacements des fichiers journaux Amazon ECS](#) et [Ressource de calcul AMIs](#).
- Ouvrez un ticket d'assistance : si vous rencontrez toujours des problèmes après un dépannage et que vous disposez d'un plan d'assistance, ouvrez un ticket d'assistance. Dans le ticket d'assistance, assurez-vous d'inclure des informations sur le problème, les spécificités de la charge de travail, la configuration et les résultats des tests. Pour plus d'informations, voir [Comparer les Support forfaits](#).
- Consultez les forums AWS Batch et HPC — Pour plus d'informations, consultez les forums [AWS Batch](#) et [HPC](#).

- Consultez le tableau de bord de surveillance du temps AWS Batch d'exécution : ce tableau de bord utilise une architecture sans serveur pour capturer les événements d'Amazon ECS et d'Amazon EC2 afin de fournir des informations sur les tâches et les instances. AWS Batch Pour plus d'informations, consultez la section [Solution de tableaux de bord de surveillance des temps d'AWS Batch exécution](#).

Résolution des problèmes AWS Batch

Vous devrez peut-être résoudre des problèmes liés à vos environnements informatiques, à vos files d'attente de tâches, à vos définitions de tâches ou à vos tâches. Ce chapitre explique comment résoudre ces problèmes dans votre AWS Batch environnement.

AWS Batch utilise les politiques, les rôles et les autorisations IAM et s'exécute sur l'infrastructure Amazon EC2, Amazon ECS et Amazon Elastic Kubernetes Service. AWS Fargate Pour résoudre les problèmes liés à ces services, consultez les rubriques suivantes :

- [Résolution des problèmes liés à l'IAM](#) dans le guide de l'utilisateur IAM
- [Résolution des problèmes liés à Amazon ECS](#) dans le manuel du développeur Amazon Elastic Container Service
- [Résolution des problèmes liés à Amazon EKS](#) dans le guide de l'utilisateur Amazon EKS
- [Résoudre les problèmes liés aux EC2 instances](#) dans le guide de l' EC2 utilisateur Amazon

Table des matières

- [AWS Batch](#)
 - [Configuration optimale du type d'instance pour recevoir des mises à jour automatiques de la famille d'instances](#)
 - [INVALIDenvironnement informatique](#)
 - [Nom de rôle ou ARN incorrect](#)
 - [Réparation d'un environnement INVALID informatique](#)
 - [Offres d'emploi bloquées dans un RUNNABLE statut](#)
 - [Instances ponctuelles non étiquetées lors de la création](#)
 - [Les instances ponctuelles ne sont pas réduites](#)
 - [Associez la politique EC2 SpotFleetTaggingRole gérée par Amazon à votre rôle Spot Fleet dans le AWS Management Console](#)
 - [Associez la politique EC2 SpotFleetTaggingRole gérée par Amazon à votre rôle de Spot Fleet avec AWS CLI](#)
 - [Impossible de récupérer les secrets de Secrets Manager](#)
 - [Impossible de contourner les exigences en ressources liées à la définition des tâches](#)
 - [Message d'erreur lorsque vous mettez à jour le desiredvCpus paramètre](#)

- [AWS Batch sur Amazon EKS](#)
 - [INVALIDenvironnement informatique](#)
 - [Version non prise en charge Kubernetes](#)
 - [Le profil d'instance n'existe pas](#)
 - [Espace de Kubernetes noms non valide](#)
 - [Environnement de calcul supprimé](#)
 - [Les nœuds ne rejoignent pas le cluster Amazon EKS](#)
 - [AWS Batch sur Amazon EKS, le RUNNABLE statut de la tâche est bloqué](#)
 - [AWS Batch sur Amazon EKS, le STARTING statut de la tâche est bloqué](#)
 - [Scénario : échec persistant de l'attache ou du montage d'un volume réclamé](#)
 - [Vérifiez que le aws-auth ConfigMap est correctement configuré](#)
 - [Les autorisations ou les liaisons RBAC ne sont pas correctement configurées](#)

AWS Batch

Consultez les rubriques suivantes pour découvrir les processus de révision et les solutions potentielles aux problèmes courants que vous pouvez rencontrer lors de l'utilisation AWS Batch.

Rubriques

- [Configuration optimale du type d'instance pour recevoir des mises à jour automatiques de la famille d'instances](#)
- [INVALIDenvironnement informatique](#)
- [Offres d'emploi bloquées dans un RUNNABLE statut](#)
- [Instances ponctuelles non étiquetées lors de la création](#)
- [Les instances ponctuelles ne sont pas réduites](#)
- [Impossible de récupérer les secrets de Secrets Manager](#)
- [Impossible de contourner les exigences en ressources liées à la définition des tâches](#)
- [Message d'erreur lorsque vous mettez à jour le desiredvCpus paramètre](#)

Configuration optimale du type d'instance pour recevoir des mises à jour automatiques de la famille d'instances

Note

À partir du 01/11/2025, le comportement optimal de sera modifié pour correspondre. `default_x86_64` Pendant le changement, vos familles d'instances peuvent être mises à jour vers une génération plus récente. Vous n'avez aucune action à effectuer pour que la mise à niveau ait lieu.

AWS Batch supportait une seule option dans `InstanceTypes optimal` pour répondre à la demande de vos files d'attente de travail. Nous avons introduit deux nouvelles options de type d'instance : `default_x86_64` et `default_arm64`. Nous l'utiliserons `default_x86_64` si vous ne sélectionnez aucun type d'instance. Ces nouvelles options sélectionneront automatiquement des types d'instances rentables issus de différentes familles et générations en fonction des exigences de votre file d'attente, ce qui vous permettra d'exécuter rapidement vos charges de travail.

Dès qu'une capacité suffisante de nouveaux types d'instances devient disponible dans un Région AWS, le pool par défaut correspondant sera automatiquement mis à jour avec le nouveau type d'instance. L'option existante continuera d'être prise en charge et n'est pas obsolète, car elle sera prise en charge par les pools par défaut sous-jacents afin de fournir des instances mises à jour à l'avenir. Si vous utilisez 'optimal', aucune action de votre part n'est requise.

Sachez toutefois que seuls `ENABLED` les environnements de `VALID` calcul (CEs) seront automatiquement mis à jour avec les nouveaux types d'instances. Si vous en avez `INVALID` CEs, ils recevront `DISABLED` des mises à jour une fois qu'ils seront réactivés et qu'ils auront atteint un `VALID` état.

INVALID environnement informatique

Il est possible que vous ayez mal configuré un environnement informatique géré. Si c'est le cas, l'environnement informatique entre dans un `INVALID` état et ne peut pas accepter de postes à des fins de placement. Les sections suivantes décrivent les causes possibles et la procédure de dépannage en fonction de la cause.

 Important

AWS Batch crée et gère plusieurs AWS ressources en votre nom et au sein de votre compte, notamment les modèles Amazon EC2 Launch, les groupes Amazon EC2 Auto Scaling, les flottes Amazon EC2 Spot et les clusters Amazon ECS. Ces ressources gérées sont configurées spécifiquement pour garantir un AWS Batch fonctionnement optimal. La modification manuelle de ces ressources gérées par lots, sauf indication explicite dans la AWS Batch documentation, peut entraîner un comportement inattendu entraînant un environnement de INVALID calcul, un comportement de dimensionnement des instances sous-optimal, un retard dans le traitement de la charge de travail ou des coûts imprévus. Ces modifications manuelles ne peuvent pas être prises en charge de manière déterministe par le AWS Batch service. Utilisez toujours le Batch compatible APIs ou la console Batch pour gérer vos environnements informatiques.

Nom de rôle ou ARN incorrect

La raison la plus courante pour laquelle un environnement informatique entre dans un INVALID état est que le rôle de AWS Batch service ou le rôle Amazon EC2 Spot Fleet porte un nom ou un nom de ressource Amazon (ARN) incorrect. Cela est plus courant dans les environnements informatiques créés à l'aide du AWS CLI ou du AWS SDKs. Lorsque vous créez un environnement informatique dans le AWS Management Console, AWS Batch cela vous aide à choisir le service ou les rôles Spot Fleet appropriés. Supposons toutefois que vous saisissiez manuellement le nom ou l'ARN et que vous ne les saisissiez pas correctement. Ensuite, l'environnement de calcul qui en résulte l'est égalementINVALID.

Supposons toutefois que vous saisissiez manuellement le nom ou l'ARN d'une ressource IAM dans une AWS CLI commande ou dans le code de votre SDK. Dans ce cas, AWS Batch impossible de valider la chaîne. AWS Batch Il faut plutôt accepter la mauvaise valeur et tenter de créer l'environnement. Si la création de l'environnement AWS Batch échoue, celui-ci passe à un INVALID état et les erreurs suivantes s'affichent.

Pour un rôle de service non valide :

```
CLIENT_ERROR - Not authorized to perform sts:AssumeRole (Service:
AWSSecurityTokenService; Status Code: 403; Error Code: AccessDenied;
Request ID: dc0e2d28-2e99-11e7-b372-7fcc6fb65fe7)
```

Pour un rôle de parc d'instances Spot non valide :

```
CLIENT_ERROR - Parameter: SpotFleetRequestConfig.IamFleetRole
is invalid. (Service: AmazonEC2; Status Code: 400; Error Code:
InvalidSpotFleetRequestConfig; Request ID: 331205f0-5ae3-4cea-
bac4-897769639f8d) Parameter: SpotFleetRequestConfig.IamFleetRole is
invalid
```

L'une des causes fréquentes de ce problème est le scénario suivant. Vous ne spécifiez le nom d'un rôle IAM que lorsque vous utilisez le AWS CLI ou le AWS SDKs, au lieu du nom complet de la ressource Amazon (ARN). Selon la façon dont vous avez créé le rôle, l'ARN peut contenir un préfixe de `aws-service-role` chemin. Par exemple, si vous créez manuellement le rôle de AWS Batch service à l'aide des procédures décrites dans [Utilisation des rôles liés à un service pour AWS Batch](#), l'ARN de votre rôle de service peut ressembler à ce qui suit.

```
arn:aws:iam::123456789012:role/AWSBatchServiceRole
```

Toutefois, si vous avez créé le rôle de service dans le cadre de l'assistant de première exécution de la console aujourd'hui, l'ARN de votre rôle de service peut ressembler à ce qui suit.

```
arn:aws:iam::123456789012:role/aws-service-role/AWSBatchServiceRole
```

Ce problème peut également se produire si vous associez la politique de AWS Batch niveau de service (`AWSBatchServiceRole`) à un rôle non lié au service. Par exemple, vous pouvez recevoir un message d'erreur semblable au suivant dans ce scénario :

```
CLIENT_ERROR - User: arn:aws:sts::account_number:assumed-role/batch-replacement-role/
aws-batch is not
authorized to perform: action on resource ...
```

Pour résoudre ce problème, effectuez l'une des opérations suivantes.

- Utilisez une chaîne vide pour le rôle de service lorsque vous créez l'environnement AWS Batch informatique.
- Spécifiez le rôle de service au format suivant : `arn:aws:iam::account_number:role/aws-service-role/batch.amazonaws.com/AWSServiceRoleForBatch`

Lorsque vous spécifiez uniquement le nom d'un rôle IAM lorsque vous utilisez le AWS CLI ou le AWS SDKs, cela AWS Batch suppose que votre ARN n'utilise pas le préfixe de `aws-service-role`

chemin. C'est pourquoi nous vous recommandons de spécifier l'ARN complet pour vos rôles IAM lorsque vous créez des environnements informatiques.

Pour réparer un environnement informatique mal configuré de cette façon, consultez [Réparation d'un environnement INVALID informatique](#).

Réparation d'un environnement **INVALID** informatique

Lorsqu'un environnement informatique est dans un INVALID état, mettez-le à jour pour réparer le paramètre non valide. Dans le cas [Nom de rôle ou ARN incorrect](#) contraire, mettez à jour l'environnement de calcul en utilisant le rôle de service approprié.

Pour réparer un environnement de calcul mal configuré

1. Ouvrez la AWS Batch console à l'adresse <https://console.aws.amazon.com/batch/>.
2. Dans la barre de navigation, sélectionnez le Région AWS à utiliser.
3. Dans le panneau de navigation, choisissez Environnements de calcul.
4. Sur la page Environnements de calcul, cochez la case en regard de l'environnement de calcul à modifier, puis choisissez Modifier.
5. Sur la page Mettre à jour l'environnement de calcul, pour Rôle de service, choisissez le rôle IAM à utiliser avec votre environnement de calcul. La console AWS Batch affiche uniquement les rôles ayant la relation d'approbation appropriée pour les environnements de calcul.

Tip

Pour savoir comment créer un rôle lié à un service, consultez [Utilisation de rôles pour AWS Batch](#).

6. Choisissez Enregistrer pour mettre à jour votre environnement de calcul.

Offres d'emploi bloquées dans un **RUNNABLE** statut

Supposons que votre environnement informatique contienne des ressources informatiques, mais que vos tâches ne progressent pas au-delà de RUNNABLE leur statut. Il est alors probable que quelque chose empêche le placement des tâches sur une ressource informatique et bloque vos files d'attente de tâches. Voici comment savoir si votre tâche attend son tour ou si elle est bloquée et bloque la file d'attente.

Si vous AWS Batch détectez que vous avez une `RUNNABLE` tâche en tête et que vous bloquez la file d'attente, vous recevrez un [Événements bloqués dans la file d'offres d'emploi](#) événement d'Amazon CloudWatch Events indiquant le motif. La même raison est également mise à jour dans le `statusReason` champ dans le cadre [ListJobs](#) d'appels d'[DescribeJobs](#) API.

Vous pouvez éventuellement configurer le `jobStateLimitActions` paramètre via [CreateJobQueue](#) des actions [UpdateJobQueue](#) d'API.

 Note

À l'heure actuelle, la seule action que vous pouvez utiliser `jobStateLimitActions.action` est d'annuler une tâche.

Le `jobStateLimitActions` paramètre est utilisé pour spécifier un ensemble d'actions exécutées AWS Batch sur des tâches dans un état spécifique. Vous pouvez définir un seuil de temps en secondes via le `maxTimeSeconds` champ.

Lorsqu'une tâche est dans un `RUNNABLE` état défini `statusReason`, AWS Batch exécute l'action spécifiée après son `maxTimeSeconds` expiration.

Par exemple, vous pouvez définir le `jobStateLimitActions` paramètre pour qu'il attende jusqu'à 4 heures pour toute tâche dont l'`RUNNABLE` état attend la disponibilité d'une capacité suffisante. Vous pouvez le faire en `statusReason` réglant sur `144000` avant d'annuler la tâche et de laisser la tâche suivante passer en tête de file d'attente. `CAPACITY: INSUFFICIENT_INSTANCE_CAPACITY` `maxTimeSeconds`

Voici les raisons invoquées lorsqu' AWS Batch il détecte qu'une file d'attente de tâches est bloquée. Cette liste fournit les messages renvoyés par les actions `ListJobs` et `DescribeJobs` API. Il s'agit également des mêmes valeurs que vous pouvez définir pour le `jobStateLimitActions.statusReason` paramètre.

1. Raison : tous les environnements informatiques connectés présentent des erreurs de capacité insuffisante. Sur demande, AWS Batch détecte les EC2 instances Amazon présentant des erreurs de capacité insuffisante. L'annulation manuelle de la tâche permettra à la tâche suivante de passer en tête de file d'attente, mais si le ou les problèmes liés au rôle de service ne sont pas résolus, il est probable que la tâche suivante soit également bloquée. Il est préférable d'étudier et de résoudre ce problème manuellement.

- **statusReason** message lorsque la tâche est bloquée :
CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY - Service cannot fulfill the capacity requested for instance type [instanceTypeName]
- **reason** utilisé pour **jobStateTimeLimitActions** :
CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY
- **statusReason** message après l'annulation de la tâche : Canceled by JobStateTimeLimit action due to reason: CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY

Remarque :

- a. Le rôle AWS Batch de service nécessite une `autoscaling:DescribeScalingActivities` autorisation pour que cette détection fonctionne. Si vous utilisez le rôle [Utilisation des rôles liés à un service pour AWS Batch](#) lié à un service (SLR) ou la politique [AWS stratégie gérée : AWSBatchServiceRolestratégie](#) gérée, vous n'avez aucune action à effectuer car leurs politiques d'autorisation sont mises à jour.
 - b. Si vous utilisez le SLR ou la politique gérée, vous devez ajouter les `ec2:DescribeSpotFleetRequestHistory` autorisations `autoscaling:DescribeScalingActivities` et afin de pouvoir recevoir les événements de file d'attente de tâches bloqués et le statut des tâches mis à jour lorsque vous êtes connecté RUNNABLE. En outre, AWS Batch a besoin de ces autorisations pour effectuer des `cancellation` actions via le `jobStateTimeLimitActions` paramètre, même si elles sont configurées dans la file d'attente des tâches.
 - c. Dans le cas d'une tâche multinode parallèle (MNP), si l'environnement de EC2 calcul Amazon hautement prioritaire associé rencontre des `insufficient capacity` erreurs, il bloque la file d'attente même si un environnement de calcul de priorité inférieure rencontre cette erreur.
2. Raison : Tous les environnements informatiques ont un `maxvCpus` paramètre inférieur aux exigences de la tâche. L'annulation de la tâche, manuellement ou en activant le `jobStateTimeLimitActions` paramètre `statusReason`, permet à la tâche suivante de passer en tête de file d'attente. Vous pouvez éventuellement augmenter le `maxvCpus` paramètre de l'environnement informatique principal pour répondre aux besoins de la tâche bloquée.
 - **statusReason** message lorsque la tâche est bloquée :
MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE - CE(s) associated with the job queue cannot meet the CPU requirement of the job.
 - **reason** utilisé pour **jobStateTimeLimitActions** :
MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE

- **statusReason**message après l'annulation de la tâche : `Canceled by JobStateTimeLimit action due to reason: MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE`
3. Raison : aucun environnement informatique ne possède d'instance répondant aux exigences du poste. Lorsqu'une tâche demande des ressources, AWS Batch détecte qu'aucun environnement informatique associé n'est en mesure d'accueillir la tâche entrante. L'annulation de la tâche, manuellement ou en activant le `jobStateTimeLimitActions` paramètre `statusReason`, permet à la tâche suivante de passer en tête de file d'attente. Vous pouvez éventuellement redéfinir les types d'instances autorisés dans l'environnement de calcul pour ajouter les ressources de travail nécessaires.
- **statusReason**message lorsque la tâche est bloquée :
`MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT - The job resource requirement (vCPU/memory/GPU) is higher than that can be met by the CE(s) attached to the job queue.`
 - **reason**utilisé pour **jobStateTimeLimitActions** :
`MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT`
 - **statusReason**message après l'annulation de la tâche : `Canceled by JobStateTimeLimit action due to reason: MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT`
4. Raison : Tous les environnements informatiques présentent des problèmes de rôle de service. Pour résoudre ce problème, comparez les autorisations de votre rôle de service aux autorisations [AWS politiques gérées pour AWS Batch](#) et corrigez les éventuelles lacunes. Remarque : vous ne pouvez pas configurer d'action programmable via le `jobStateTimeLimitActions` paramètre pour résoudre cette erreur.

Il est recommandé d'utiliser le [Utilisation des rôles liés à un service pour AWS Batch](#) pour éviter des erreurs similaires.

L'annulation de la tâche, manuellement ou en activant le `jobStateTimeLimitActions` paramètre `statusReason`, permet à la tâche suivante de passer en tête de file d'attente. Si les problèmes liés au rôle de service ne sont pas résolus, il est probable que la prochaine tâche soit également bloquée. Il est préférable d'étudier et de résoudre ce problème manuellement.

- **statusReason**message lorsque la tâche est bloquée :
`MISCONFIGURATION:SERVICE_ROLE_PERMISSIONS - Batch service role has a permission issue.`
5. Raison : votre environnement informatique possède une configuration de type d'instance non prise en charge. Cela peut se produire lorsque les types d'instances ne sont pas disponibles dans les

zones de disponibilité que vous avez sélectionnées, ou lorsque votre modèle de lancement ou votre configuration de lancement contient des paramètres incompatibles avec les types d'instances spécifiés. Pour résoudre ce problème, vérifiez que vos types d'instances sont pris en charge dans les zones spécifiées Région AWS et dans les zones de disponibilité, vérifiez que les paramètres de votre modèle de lancement sont compatibles avec vos types d'instances et envisagez de passer à des types d'instances de nouvelle génération. Pour plus d'informations sur la recherche des types d'instances pris en charge, consultez la section [Trouver un type d' EC2 instance Amazon](#) dans le guide de EC2 l'utilisateur Amazon.

- **statusReason**message lorsque la tâche est bloquée :
MISCONFIGURATION:EC2_INSTANCE_CONFIGURATION_UNSUPPORTED - Your compute environment associated with this job queue has an unsupported instance type configuration.
6. Raison : Tous les environnements informatiques ne sont pas valides. Pour de plus amples informations, veuillez consulter [INVALIDenvironnement informatique](#). Remarque : Vous ne pouvez pas configurer une action programmable via le `jobStateTimeLimitActions` paramètre pour résoudre cette erreur.
- **statusReason**message lorsque la tâche est bloquée : ACTION_REQUIRED - CE(s) associated with the job queue are invalid.
7. Raison : AWS Batch a détecté une file d'attente bloquée, mais n'est pas en mesure d'en déterminer la raison. Remarque : Vous ne pouvez pas configurer une action programmable via le `jobStateTimeLimitActions` paramètre pour résoudre cette erreur. Pour plus d'informations sur le dépannage, consultez [Pourquoi mon AWS Batch travail est-il bloqué dans RUNNABLE ou dans AWS Re:post](#).
- **statusReason**message lorsque la tâche est bloquée : UNDETERMINED - Batch job is blocked, root cause is undetermined.

Si vous n'avez pas reçu d'événement de la part d' CloudWatch Events ou si vous avez reçu un événement pour une raison inconnue, voici quelques causes courantes de ce problème.

Le pilote de **awslogs** journal n'est pas configuré sur vos ressources informatiques

AWS Batch les tâches envoient leurs informations de journal à CloudWatch Logs. Dans ce cas, vous devez configurer vos ressources de calcul de manière à ce qu'elles utilisent le pilote de journal **awslogs**. Supposons que vous basiez votre AMI de ressources de calcul sur l'AMI optimisée pour Amazon ECS (ou Amazon Linux). Ce pilote est ensuite enregistré par défaut dans le `ecs-init` package. Supposons maintenant que vous utilisiez une autre AMI de base.

Vous devez ensuite vérifier que le pilote de `awslogs` journal est spécifié comme pilote de journal disponible avec la variable d'`ECS_AVAILABLE_LOGGING_DRIVERS` environnement lorsque l'agent de conteneur Amazon ECS est démarré. Pour plus d'informations, consultez [Spécification d'AMI de ressources de calcul](#) et [Tutoriel : Création d'une AMI de ressources de calcul](#).

Ressources insuffisantes

Si vos définitions de tâches spécifient plus de ressources de processeur ou de mémoire que ce que vos ressources de calcul peuvent allouer, vos tâches ne sont jamais placées. Supposons, par exemple, que votre tâche spécifie 4 GiB de mémoire et que vos ressources de calcul soient inférieures à celles disponibles. Il arrive alors que la tâche ne puisse pas être placée sur ces ressources informatiques. Dans ce cas, vous devez réduire la mémoire spécifiée dans la définition de tâche ou ajouter des ressources de calcul à votre environnement. Une partie de la mémoire est réservée à l'agent de conteneur Amazon ECS et à d'autres processus critiques du système. Pour de plus amples informations, veuillez consulter [Gestion de la mémoire des ressources informatiques](#).

Pas d'accès à Internet pour les ressources informatiques

Les ressources de calcul ont besoin de communiquer avec le point de terminaison de service Amazon ECS service. Cela peut être via un point de terminaison d'un VPC d'interface ou via vos ressources de calcul ayant des adresses IP publiques.

Pour plus d'informations sur les points de terminaison d'un VPC d'interface, veuillez consulter [Points de terminaison d'un VPC d'interface Amazon ECS AWS PrivateLink](#) dans le Guide du développeur Amazon Elastic Container Service.

Si vous n'avez pas de point de terminaison d'un VPC d'interface configuré et que vos ressources de calcul n'ont pas d'adresses IP publiques, elles doivent utiliser la traduction d'adresse réseau (NAT) pour fournir cet accès. Pour de plus amples informations, veuillez consulter [Passerelles NAT](#) dans le Guide de l'utilisateur Amazon VPC. Pour de plus amples informations, veuillez consulter [the section called "Création d'un VPC"](#).

Limite EC2 d'instances Amazon atteinte

Le nombre d' EC2 instances Amazon dans lesquelles votre compte peut être lancé Région AWS est déterminé par votre quota d' EC2 instances. Certains types d'instances sont également soumis à un per-instance-type quota. Pour plus d'informations sur le quota d' EC2 instances Amazon de votre compte, notamment sur la manière de demander une augmentation de limite, consultez [Amazon EC2 Service Limits](#) dans le guide de EC2 l'utilisateur Amazon.

L'agent de conteneur Amazon ECS n'est pas installé

L'agent de conteneur Amazon ECS doit être installé sur l'Amazon Machine Image (AMI) pour permettre l'AWS Batch exécution de tâches. L'agent de conteneur Amazon ECS est installé par défaut sur Amazon ECS optimisé AMIs. Pour plus d'informations sur l'agent de conteneur Amazon ECS, consultez la section relative à l'[agent de conteneur Amazon ECS](#) dans le guide du développeur Amazon Elastic Container Service.

Pour plus d'informations, voir [Pourquoi mon AWS Batch travail est-il RUNNABLE bloqué ?](#) dans Re:post.

Instances ponctuelles non étiquetées lors de la création

Le balisage des instances Spot pour les ressources de AWS Batch calcul est pris en charge depuis le 25 octobre 2017. Auparavant, la politique gérée par IAM recommandée (AmazonEC2SpotFleetRole) pour le rôle Amazon EC2 Spot Fleet ne contenait pas d'autorisations permettant de baliser les instances Spot lors du lancement. La nouvelle politique gérée par IAM recommandée s'appelle AmazonEC2SpotFleetTaggingRole. Il prend en charge le balisage des instances Spot au lancement.

Pour corriger le balisage des instances Spot lors de leur création, suivez la procédure suivante pour appliquer la politique de gestion IAM actuellement recommandée à votre rôle Amazon EC2 Spot Fleet. Ainsi, toutes les futures instances Spot créées avec ce rôle seront autorisées à appliquer des balises d'instance lors de leur création.

Pour appliquer la politique gérée par IAM actuelle à votre rôle Amazon EC2 Spot Fleet

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Choisissez Rôles, puis choisissez votre rôle Amazon EC2 Spot Fleet.
3. Choisissez Attach policy (Attacher la politique).
4. Sélectionnez Amazon, EC2 SpotFleetTaggingRole puis choisissez Attach policy.
5. Choisissez à nouveau votre rôle Amazon EC2 Spot Fleet pour supprimer la politique précédente.
6. Sélectionnez le x à droite de la EC2 SpotFleetRole politique d'Amazon, puis choisissez Detach.

Les instances ponctuelles ne sont pas réduites

AWS Batch a introduit le rôle `AWSServiceRoleForBatch` au service le 10 mars 2021. Si aucun rôle n'est spécifié dans le `serviceRole` paramètre de l'environnement informatique, ce rôle lié au service est utilisé comme rôle de service. Supposons toutefois que le rôle lié au service soit utilisé dans un environnement informatique EC2 Spot, mais que le rôle Spot utilisé n'inclut pas la politique `EC2 SpotFleetTaggingRole` gérée par Amazon. Dans ce cas, l'instance Spot n'est pas réduite. Par conséquent, vous recevrez un message d'erreur avec le message suivant : « Vous n'êtes pas autorisé à effectuer cette opération ». Procédez comme suit pour mettre à jour le rôle de flotte ponctuelle que vous utilisez dans le `spotIamFleetRole` paramètre. Pour plus d'informations, consultez les sections [Utilisation de rôles liés à un service](#) et [Création d'un rôle pour déléguer des autorisations à un AWS service](#) dans le Guide de l'utilisateur IAM.

Rubriques

- [Associez la politique `EC2 SpotFleetTaggingRole` gérée par Amazon à votre rôle Spot Fleet dans le AWS Management Console](#)
- [Associez la politique `EC2 SpotFleetTaggingRole` gérée par Amazon à votre rôle de Spot Fleet avec AWS CLI](#)

Associez la politique `EC2 SpotFleetTaggingRole` gérée par Amazon à votre rôle Spot Fleet dans le AWS Management Console

Pour appliquer la politique gérée par IAM actuelle à votre rôle Amazon EC2 Spot Fleet

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Choisissez Rôles, puis choisissez votre rôle Amazon EC2 Spot Fleet.
3. Choisissez Attach policy (Attacher la politique).
4. Sélectionnez Amazon, `EC2 SpotFleetTaggingRole` puis choisissez Attach policy.
5. Choisissez à nouveau votre rôle Amazon EC2 Spot Fleet pour supprimer la politique précédente.
6. Sélectionnez le x à droite de la `EC2 SpotFleetRole` politique d'Amazon, puis choisissez Detach.

Associez la politique EC2 SpotFleetTaggingRole gérée par Amazon à votre rôle de Spot Fleet avec AWS CLI

Les exemples de commandes supposent que votre rôle Amazon EC2 Spot Fleet est nommé *AmazonEC2SpotFleetRole*. Si votre rôle utilise un nom différent, ajustez les commandes en conséquence.

Pour associer la politique EC2 SpotFleetTaggingRole gérée par Amazon à votre rôle Spot Fleet

1. Pour associer la politique IAM EC2 SpotFleetTaggingRole gérée par Amazon à votre *AmazonEC2SpotFleetRole* rôle, exécutez la commande suivante à l'aide du AWS CLI.

```
$ aws iam attach-role-policy \  
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetTaggingRole \  
    --role-name AmazonEC2SpotFleetRole
```

2. Pour dissocier la politique IAM EC2 SpotFleetRole gérée par Amazon de votre *AmazonEC2SpotFleetRole* rôle, exécutez la commande suivante à l'aide du AWS CLI

```
$ aws iam detach-role-policy \  
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetRole \  
    --role-name AmazonEC2SpotFleetRole
```

Impossible de récupérer les secrets de Secrets Manager

Si vous utilisez une AMI avec un agent Amazon ECS antérieur à la version

1.16.0-1, vous devez utiliser la variable de configuration de l'agent Amazon ECS

ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE=true pour utiliser cette fonctionnalité.

Vous pouvez l'ajouter au `./etc/ecs/ecs.config` fichier d'une nouvelle instance de conteneur lorsque vous créez cette instance. Vous pouvez également l'ajouter à une instance existante. Si vous l'ajoutez à une instance existante, vous devez redémarrer l'agent ECS après l'avoir ajouté. Pour plus d'informations, consultez [Configuration de l'agent du conteneur Amazon ECS](#) dans le Manuel du développeur Amazon Elastic Container Service.

Impossible de contourner les exigences en ressources liées à la définition des tâches

Les remplacements de mémoire et de vCPU spécifiés dans la structure `memory` et les `vcpus` membres de la structure `ContainerOverrides`, transmis à `SubmitJob`, ne peuvent pas remplacer les exigences en matière de mémoire et de vCPU spécifiées dans la structure `ResourceRequirements` de la définition de tâche.

Si vous essayez de contourner ces exigences en matière de ressources, le message d'erreur suivant peut s'afficher :

« Cette valeur a été soumise dans une clé obsolète et peut entrer en conflit avec la valeur fournie par les exigences en ressources de la définition de tâche. »

Pour corriger cela, spécifiez les exigences en matière de mémoire et de vCPU dans le membre `ResourceRequirements` de `ContainerOverrides`. Par exemple, si vos remplacements de mémoire et de vCPU sont spécifiés dans les lignes suivantes.

```
"containerOverrides": {  
  "memory": 8192,  
  "vcpus": 4  
}
```

Modifiez-les comme suit :

```
"containerOverrides": {  
  "resourceRequirements": [  
    {  
      "type": "MEMORY",  
      "value": "8192"  
    },  
    {  
      "type": "VCPU",  
      "value": "4"  
    }  
  ],  
}
```

Procédez de la même manière aux exigences en matière de mémoire et de vCPU spécifiées dans l'objet `ContainerProperties` dans la définition de la tâche. Par exemple, si vos exigences en matière de mémoire et de vCPU sont spécifiées dans les lignes suivantes.

```
{
  "containerProperties": {
    "memory": 4096,
    "vcpus": 2,
  }
}
```

Modifiez-les comme suit :

```
"containerProperties": {
  "resourceRequirements": [
    {
      "type": "MEMORY",
      "value": "4096"
    },
    {
      "type": "VCPU",
      "value": "2"
    }
  ],
}
```

Message d'erreur lorsque vous mettez à jour le **desiredVcpus** paramètre

Le message d'erreur suivant s'affiche lorsque vous utilisez l' AWS Batch API pour mettre à jour le paramètre v CPUs (desiredVcpus) souhaité.

Manually scaling down compute environment is not supported. Disconnecting job queues from compute environment will cause it to scale-down to minVcpus.

Ce problème se produit si la desiredVcpus valeur mise à jour est inférieure à la desiredVcpus valeur actuelle. Lorsque vous mettez à jour la desiredVcpus valeur, les deux conditions suivantes doivent être vraies :

- La desiredVcpus valeur doit être comprise entre les maxVcpus valeurs minVcpus et.
- La desiredVcpus valeur mise à jour doit être supérieure ou égale à la desiredVcpus valeur actuelle.

AWS Batch sur Amazon EKS

Rubriques

- [INVALIDEnvironnement informatique](#)
- [AWS Batch sur Amazon EKS, le RUNNABLE statut de la tâche est bloqué](#)
- [AWS Batch sur Amazon EKS, le STARTING statut de la tâche est bloqué](#)
- [Vérifiez que le aws-auth ConfigMap est correctement configuré](#)
- [Les autorisations ou les liaisons RBAC ne sont pas correctement configurées](#)

Consultez les rubriques suivantes pour découvrir les processus de révision et les solutions potentielles aux problèmes courants que vous pouvez rencontrer lors de l'utilisation d' AWS Batch Amazon Elastic Kubernetes Service.

INVALIDEnvironnement informatique

Il est possible que vous ayez mal configuré un environnement informatique géré. Si c'est le cas, l'environnement informatique entre dans un INVALID état et ne peut pas accepter de postes à des fins de placement. Les sections suivantes décrivent les causes possibles et la procédure de dépannage en fonction de la cause.

Version non prise en charge Kubernetes

Un message d'erreur semblable au suivant peut s'afficher lorsque vous utilisez l'opération `CreateComputeEnvironmentAPI` ou l'opération `UpdateComputeEnvironmentAPI` pour créer ou mettre à jour un environnement de calcul. Ce problème se produit si vous spécifiez une Kubernetes version non prise en charge dans `EC2Configuration`.

At least one imageKubernetesVersion in EC2Configuration is not supported.

Pour résoudre ce problème, supprimez l'environnement informatique, puis recréez-le avec une Kubernetes version prise en charge.

Vous pouvez effectuer une mise à niveau de version mineure sur votre cluster Amazon EKS. Par exemple, vous pouvez mettre à niveau le cluster de 1.xx à 1.yy même si la version mineure n'est pas prise en charge.

Toutefois, l'état de l'environnement de calcul peut changer `INVALID` après une mise à jour majeure de la version. Par exemple, si vous effectuez une mise à niveau d'une version majeure de `1.xx` vers `2.yy`. Si la version majeure n'est pas prise en charge par AWS Batch, un message d'erreur semblable au suivant s'affiche.

```
reason=CLIENT_ERROR - ... EKS Cluster version [2.yy] is unsupported
```

Pour résoudre ce problème, spécifiez une Kubernetes version prise en charge lorsque vous utilisez une opération d'API pour créer ou mettre à jour un environnement informatique.

AWS Batch sur Amazon, EKS prend actuellement en charge les Kubernetes versions suivantes :

- 1.34
- 1.33
- 1.32
- 1.31
- 1.30
- 1.29

Le profil d'instance n'existe pas

Si le profil d'instance spécifié n'existe pas, le statut de l'environnement de calcul AWS Batch sur Amazon EKS est remplacé par `INVALID`. Vous voyez une erreur définie dans le `statusReason` paramètre qui ressemble à ce qui suit.

```
CLIENT_ERROR - Instance profile arn:aws:iam::...:instance-profile/<name> does not exist
```

Pour résoudre ce problème, spécifiez ou créez un profil d'instance de travail. Pour de plus amples informations, veuillez consulter [Rôle IAM de nœud Amazon EKS](#) dans le Guide de l'utilisateur Amazon EKS.

Espace de Kubernetes noms non valide

Si AWS Batch sur Amazon EKS ne parvient pas à valider l'espace de noms de l'environnement de calcul, le statut de l'environnement de calcul est remplacé par `INVALID`. Par exemple, ce problème peut se produire si l'espace de noms n'existe pas.

Un message d'erreur semblable au suivant s'affiche dans le `statusReason` paramètre.

```
CLIENT_ERROR - Unable to validate Kubernetes Namespace
```

Ce problème peut se produire si l'une des conditions suivantes est vraie :

- La chaîne d'espaces de noms contenue dans l'`CreateComputeEnvironment` appel n'existe pas. Pour de plus amples informations, veuillez consulter [CreateComputeEnvironment](#).
- Les autorisations de contrôle d'accès basé sur les rôles (RBAC) requises pour gérer l'espace de noms ne sont pas correctement configurées.
- AWS Batch n'a pas accès au point de terminaison du serveur Kubernetes d'API Amazon EKS.

Pour résoudre ce problème, consultez [Vérifiez que le aws-auth ConfigMap est correctement configuré](#). Pour de plus amples informations, veuillez consulter [Commencer à utiliser AWS Batch sur Amazon EKS](#).

Environnement de calcul supprimé

Supposons que vous supprimiez un cluster Amazon EKS avant de supprimer le cluster associé AWS Batch dans l'environnement informatique Amazon EKS. Ensuite, l'état de l'environnement de calcul est changé en `INVALID`. Dans ce scénario, l'environnement de calcul ne fonctionne pas correctement si vous recréez le cluster Amazon EKS portant le même nom.

Pour résoudre ce problème, supprimez puis recréez l'environnement de calcul AWS Batch sur Amazon EKS.

Les nœuds ne rejoignent pas le cluster Amazon EKS

AWS Batch sur Amazon EKS réduit un environnement de calcul s'il détermine que tous les nœuds n'ont pas rejoint le cluster Amazon EKS. Lorsque AWS Batch sur Amazon EKS réduit l'environnement de calcul, le statut de l'environnement de calcul est modifié en `INVALID`.

Note

AWS Batch ne modifie pas immédiatement l'état de l'environnement informatique afin que vous puissiez résoudre le problème.

Un message d'erreur semblable à l'un des suivants s'affiche dans le `statusReason` paramètre :

Your compute environment has been **INVALIDATED** and scaled down because none of the instances joined the underlying ECS Cluster. Common issues preventing instances joining are the following: VPC/Subnet configuration preventing communication to ECS, incorrect Instance Profile policy preventing authorization to ECS, or customized AMI or LaunchTemplate configurations affecting ECS agent.

Your compute environment has been **INVALIDATED** and scaled down because none of the nodes joined the underlying Amazon EKS Cluster. Common issues preventing nodes joining are the following: networking configuration preventing communication to Amazon EKS Cluster, incorrect Amazon EKS Instance Profile or Kubernetes RBAC policy preventing authorization to Amazon EKS Cluster, customized AMI or LaunchTemplate configurations affecting Amazon EKS/Kubernetes node bootstrap.

Lorsque vous utilisez une AMI Amazon EKS par défaut, les causes les plus fréquentes de ce problème sont les suivantes :

- Le rôle d'instance n'est pas configuré correctement. Pour de plus amples informations, veuillez consulter [Rôle IAM de nœud Amazon EKS](#) dans le Guide de l'utilisateur Amazon EKS.
- Les sous-réseaux ne sont pas correctement configurés. Pour plus d'informations, consultez les [exigences et considérations relatives aux VPC et aux sous-réseaux Amazon EKS](#) dans le guide de l'utilisateur Amazon EKS.
- Le groupe de sécurité n'est pas configuré correctement. Pour plus d'informations, consultez les [exigences et considérations relatives aux groupes de sécurité Amazon EKS](#) dans le guide de l'utilisateur Amazon EKS.

 Note

Vous pouvez également voir une notification d'erreur dans le Personal Health Dashboard (PHD).

AWS Batch sur Amazon EKS, le **RUNNABLE** statut de la tâche est bloqué

Un `aws-auth` ConfigMap est automatiquement créé et appliqué à votre cluster lorsque vous créez un groupe de nœuds gérés ou un groupe de nœuds à l'aide de `eksctl`. Un `aws-auth` ConfigMap est initialement créé pour permettre aux nœuds de rejoindre votre cluster. Toutefois, vous pouvez

également utiliser le `aws-auth ConfigMap` pour ajouter un accès de contrôle d'accès basé sur les rôles (RBAC) aux utilisateurs et aux rôles.

Pour vérifier que le `aws-auth ConfigMap` est correctement configuré, procédez comme suit :

1. Récupérez les rôles mappés dans : `aws-auth ConfigMap`

```
$ kubectl get configmap -n kube-system aws-auth -o yaml
```

2. Vérifiez que le `roleARN` est configuré comme suit.

```
roleARN: arn:aws:iam::aws_account_number:role/AWSServiceRoleForBatch
```

Note

Vous pouvez également consulter les journaux du plan de contrôle Amazon EKS. Pour plus d'informations, consultez la section [Connexion au plan de contrôle Amazon EKS](#) dans le guide de l'utilisateur Amazon EKS.

Pour résoudre un problème lié au blocage d'une tâche dans un `RUNNABLE` statut, nous vous recommandons de `kubectl` réappliquer le manifeste. Pour de plus amples informations, veuillez consulter [Étape 2 : Préparez votre cluster Amazon EKS pour AWS Batch](#). Vous pouvez également l'utiliser `kubectl` pour modifier manuellement le `aws-authConfigMap`. Pour plus d'informations, consultez la section [Activation de l'accès des utilisateurs et des rôles IAM à votre cluster](#) dans le guide de l'utilisateur Amazon EKS.

AWS Batch sur Amazon EKS, le **STARTING** statut de la tâche est bloqué

Un Job peut conserver `STARTING` son statut lorsque le Pod est bloqué `ContainerCreating` pour des demandes de longue durée provenant du kubelet (`pull`, `logexec`, `etattach`) jusqu'à ce que le problème de démarrage du pod soit résolu ou que le Job soit terminé. `PENDING` Dans les scénarios de qualification ci-dessous, AWS Batch vous mettez fin à la tâche en votre nom, sinon la tâche doit être interrompue manuellement à l'aide de l'[TerminateJob API](#).

Pour vérifier la raison pour laquelle un Job peut être bloqué `STARTING`, utilisez [Tutoriel : mapper une tâche en cours d'exécution à un pod et à un nœud](#) pour trouver le `podName` Pod et le décrire :

```
% kubectl describe pod aws-batch.000c8190-87df-31e7-8819-176fe017a24a -n my-aws-batch-namespace
```

```

Name:          aws-batch.000c8190-87df-31e7-8819-176fe017a24a
Namespace:     my-aws-batch-namespace
...
Containers:
  default:
  ...
    State:      Waiting
    Reason:     ContainerCreating
    Ready:      False
  ...
Conditions:
  Type              Status
  PodReadyToStartContainers  False
  Initialized        True
  Ready              False
  ContainersReady    False
  PodScheduled       True
  ...
Events:
  Type    Reason      Age    From    Message
  ----    -
  Warning  FailedMount  2m32s  kubelet  Unable to attach or mount volumes: ...

```

Envisagez de configurer votre cluster EKS pour [envoyer les journaux du plan de contrôle aux CloudWatch journaux](#) pour une visibilité complète.

Scénario : échec persistant de l'attache ou du montage d'un volume réclamé

Les tâches utilisant des demandes de volume persistantes pour lesquelles le volume ne parvient pas à être attaché ou monté sont susceptibles d'être résiliées. Cela peut être dû à une définition de Job mal configurée. Pour plus d'informations, consultez [Création d'une définition de tâche à nœud unique sur les ressources Amazon EKS](#).

Vérifiez que le **aws-auth ConfigMap** est correctement configuré

Pour vérifier que le aws-auth ConfigMap est correctement configuré, procédez comme suit :

1. Récupérez les rôles mappés dans le aws-authConfigMap.

```
$ kubectl get configmap -n kube-system aws-auth -o yaml
```

2. Vérifiez que le roleARN est configuré comme suit.


```
rolearn: arn:aws:iam::aws_account_number:role/AWSServiceRoleForBatch
```

Note

Le chemin `aws-service-role/batch.amazonaws.com/` a été supprimé de l'ARN du rôle lié au service. Cela est dû à un problème avec la carte `aws-auth` de configuration. Pour plus d'informations, consultez la section [Les rôles dotés de chemins ne fonctionnent pas lorsque le chemin est inclus dans leur ARN dans le aws-authconfigmap](#).

Note

Vous pouvez également consulter les journaux du plan de contrôle Amazon EKS. Pour plus d'informations, consultez la section [Connexion au plan de contrôle Amazon EKS](#) dans le guide de l'utilisateur Amazon EKS.

Pour résoudre un problème lié au blocage d'une tâche dans un `RUNNABLE` statut, nous vous recommandons de `kubectl` réappliquer le manifeste. Pour de plus amples informations, veuillez consulter [Étape 2 : Préparez votre cluster Amazon EKS pour AWS Batch](#). Vous pouvez également l'utiliser `kubectl` pour modifier manuellement le `aws-authConfigMap`. Pour plus d'informations, consultez la section [Activation de l'accès des utilisateurs et des rôles IAM à votre cluster](#) dans le guide de l'utilisateur Amazon EKS.

Les autorisations ou les liaisons RBAC ne sont pas correctement configurées

Si vous rencontrez des problèmes d'autorisation RBAC ou de liaison, vérifiez que le `aws-batch` Kubernetes rôle peut accéder à l'espace de Kubernetes noms :

```
$ kubectl get namespace namespace --as=aws-batch
```

```
$ kubectl auth can-i get ns --as=aws-batch
```

Vous pouvez également utiliser la **kubectl describe** commande pour afficher les autorisations associées à un rôle de cluster ou à un espace de Kubernetes noms.

```
$ kubectl describe clusterrole aws-batch-cluster-role
```

Voici un exemple de sortie.

```
Name:          aws-batch-cluster-role
Labels:        <none>
Annotations:   <none>
PolicyRule:
  Resources                                Non-Resource URLs  Resource Names
  Verbs
  -----
  -----
  configmaps                               []                 []
[get list watch]
  nodes                                   []                 []
[get list watch]
  pods                                    []                 []
[get list watch]
  daemonsets.apps                          []                 []
[get list watch]
  deployments.apps                         []                 []
[get list watch]
  replicaset.apps                          []                 []
[get list watch]
  statefulsets.apps                       []                 []
[get list watch]
  clusterrolebindings.rbac.authorization.k8s.io []                 []
[get list]
  clusterroles.rbac.authorization.k8s.io  []                 []
[get list]
  namespaces                               []                 []
[get]
  events                                   []                 []
[list]
```

```
$ kubectl describe role aws-batch-compute-environment-role -n my-aws-batch-namespace
```

Voici un exemple de sortie.

```

Name:      aws-batch-compute-environment-role
Labels:    <none>
Annotations: <none>
PolicyRule:
  Resources                                Non-Resource URLs  Resource Names      Verbs
  -----                                -
  pods                                    []                  []                  [create
get list watch delete patch]
  serviceaccounts                         []                  []                  [get list]
  rolebindings.rbac.authorization.k8s.io  []                  []                  [get list]
  roles.rbac.authorization.k8s.io         []                  []                  [get list]

```

Pour résoudre ce problème, réappliquez les autorisations et rolebinding les commandes RBAC. Pour de plus amples informations, veuillez consulter [Étape 2 : Préparez votre cluster Amazon EKS pour AWS Batch](#).

Ressource : quotas AWS Batch de service

Le tableau suivant indique les quotas de service AWS Batch qui ne peuvent pas être modifiés. Chaque quota est spécifique à une région.

Ressource	Quota
Nombre maximal de files d'attente de tâches Pour de plus amples informations, veuillez consulter Files d'attente pour les jobs .	50
Nombre maximal d'environnements informatiques sur Amazon ECS et Amazon EKS. Pour de plus amples informations, veuillez consulter Environnements informatiques pour AWS Batch .	50
Nombre maximum d'environnements informatiques par cluster Amazon EKS.	5
Nombre maximal d'environnements informatiques pour chaque file d'attente de tâches	3
Nombre maximal de dépendances de tâches pour une tâche	20
Taille maximale de définition de tâche (pour les opérations d'API RegisterJobDefinition)	24 Kio
Taille maximale de charge utile (pour les opérations d'API SubmitJob)	30 Kio
Taille maximum des tâches de tableau	10 000
Nombre maximal de tâches dans l'état SUBMITTED	1000000
Nombre maximum de transactions par seconde (TPS) pour chaque compte pour les SubmitJob opérations	50
Nombre maximum de ressources consommables	50 000
Nombre maximal d'environnements de service. Pour de plus amples informations, veuillez consulter Environnements de service .	50
Nombre maximal d'environnements de service pour chaque file d'attente de tâches	1

Ressource	Quota
Taille maximale de la SubmitServiceJob demande	30 Kio
Taille maximale de la charge utile d'une demande de service de travail (pour les opérations SubmitServiceJob d'API)	10 Kio
Nombre maximum de transactions par seconde (TPS) pour chaque compte pour les SubmitServiceJob opérations	5
Nombre maximal de tentatives avec une stratégie de nouvelle tentative pour une tâche de service	10

Selon votre mode d'utilisation AWS Batch, des quotas supplémentaires peuvent s'appliquer. Pour en savoir plus sur EC2 les quotas Amazon, consultez [Amazon EC2 Service Quotas](#) dans le Références générales AWS. Pour plus d'informations sur les quotas Amazon ECS, consultez la section [Amazon ECS Service Quotas](#) dans le Références générales AWS. Pour plus d'informations sur les quotas Amazon EKS, consultez la section [Amazon EKS Service Quotas](#) dans le Références générales AWS.

Historique du document

Le tableau suivant décrit les modifications importantes apportées à la documentation depuis la publication initiale de AWS Batch. Nous mettons aussi la documentation à jour régulièrement pour prendre en compte les commentaires qui nous sont envoyés.

Modification	Description	Date
Ajout de default_x86_64 et default_arm64	Ajout d'un nouveau type <code>default_x86_64</code> et <code>default_arm64</code> pour les types d'instances autorisés.	15 août 2025
Environnements de service et tâches de service ajoutés	Ajout d'environnements de service et de tâches de service à utiliser AWS Batch avec l' SageMaker IA.	30 juillet 2025
Ajouté AWSService RoleFor AWSBatch WithSagemaker et AWSBatch ServiceRolePolicyForSageMaker	Ajout d'un nouveau rôle AWS lié au service <code>AWSServiceRoleForAWSBatchWithSagemaker</code> et d'une nouvelle politique gérée <code>AWSBatchServiceRolePolicyForSageMaker</code> qui permettent AWS Batch de gérer l' SageMaker IA en votre nom.	30 juillet 2025
Ajoute le support pour EKS AL2023 AMIs	Comment passer d'EKS AL2 à EKS AL2 023	24 juin 2025
Ajoute la prise en charge FireLens de la commande ECS Exec	Ajoute la prise en charge FireLens de la commande ECS Exec.	15 avril 2025

Ajoute la prise en charge de la planification basée sur les ressources pour AWS Batch	Ajoute la prise en charge de la planification axée sur les ressources pour AWS Batch Amazon Elastic Container Service, Amazon Elastic Kubernetes Service et. AWS Fargate	27 février 2025
Versions Amazon EKS AWS Batch prises en charge mises à jour	Mise à jour des versions d'Amazon EKS qui prennent AWS Batch en charge la suppression de la version 1.22.	11 mars 2024
Versions Amazon EKS AWS Batch prises en charge mises à jour	Mise à jour des versions prises AWS Batch en charge par Amazon EKS pour inclure la version 1.29.	29 février 2024
Nouvelles tentatives de travail automatisées	Corrigé l'exemple de code.	29 février 2024
Ajoute la prise en charge des tâches multi-conteneurs pour AWS Batch	Ajoute la prise en charge des tâches multi-conteneurs AWS Batch pour Amazon Elastic Container Service, Amazon Elastic Kubernetes Service et. AWS Fargate	28 février 2024
Versions Amazon EKS AWS Batch prises en charge mises à jour	Mise à jour des versions prises AWS Batch en charge par Amazon EKS pour inclure la version 1.28	27 janvier 2024

[Mis à jour BatchServiceRolePolicy et AWSBatchServiceRole](#)

BatchServiceRolePolicy

5 décembre 2023

Mise à jour pour ajouter la prise en charge de la description de l'historique des demandes et des Amazon EC2 Auto Scaling activités de Spot Fleet.

AWSBatchServiceRole

Mis à jour pour ajouter une déclaration IDs, accorder AWS Batch des autorisations à `ec2:DescribeSpotFleetRequestsHistory` et `autoscaling:DescribeScalingActivities`.

[AWS Batch sur Amazon EKS](#)

AWS Batch ajoute la prise en charge de l'exécution de tâches sur des clusters Amazon EKS.

25 octobre 2022

[Prévention interservices confuse des adjoints pour AWS Batch](#)

AWS Batch fournit désormais une solution au problème confus de sécurité des adjoints, qui survient lorsqu'une entité (un service ou un compte) est contrainte par une autre entité d'effectuer une action.

6 juin 2022

Points de terminaison VPC d'interface ()AWS PrivateLink	Ajout de la prise en charge de la configuration des points de terminaison VPC d'interface alimentés par. AWS PrivateLink Cela signifie que vous pouvez créer une connexion privée entre votre VPC AWS Batch sans avoir besoin d'y accéder via une instance NAT, une connexion VPN ou. Direct Connect	15 avril 2022
Mises à jour améliorées de l'environnement informatique	AWS Batch mises à jour de support améliorées pour les environnements informatiques.	14 avril 2022
AWS mises à jour des politiques gérées - Mise à jour des politiques existantes	AWS Batch politiques gérées existantes mises à jour.	6 décembre 2021
Planification équitable	AWS Batch ajoute la prise en charge de l'ajout de politiques de planification aux files d'attente de tâches.	9 novembre 2021
Amazon EFS	AWS Batch ajoute la prise en charge de l'ajout de systèmes de fichiers Amazon EFS à vos définitions de tâches.	1 avril 2021
Ajout d'un rôle lié au service	AWS Batch ajoute le rôle AWSServiceRoleForBatchlié au service.	10 mars 2021
AWS Fargate soutien	AWS Batch ajoute la prise en charge de l'exécution de tâches sur les ressources Fargate.	3 décembre 2020

Étiquette des ressources	AWS Batch ajoute la prise en charge de l'ajout de balises de métadonnées à vos environnements informatiques, à vos définitions de tâches, à vos files d'attente de tâches et à vos tâches.	7 octobre 2020
Secrets	AWS Batch ajoute la prise en charge de la transmission de secrets aux emplois.	1er octobre 2020
Journalisation	AWS Batch ajoute la prise en charge de la spécification de pilotes de journal supplémentaires pour les tâches.	1er octobre 2020
Stratégies d'allocation	AWS Batch ajoute la prise en charge de plusieurs stratégies pour choisir les types d'instances.	16 octobre 2019
Soutien à l'EFA	AWS Batch ajoute le support pour les appareils Elastic Fabric Adapter (EFA).	2 août 2019
Planification du GPU	AWS Batch ajoute la planification du GPU. Grâce à cette fonctionnalité, vous pouvez spécifier le nombre de GPUs tâches requises et augmenter le nombre AWS Batch d'instances en conséquence.	4 avril 2019

<u>Tâches parallélisées sur plusieurs nœuds</u>	AWS Batch ajoute la prise en charge des tâches parallèles à nœuds multiples. Vous pouvez utiliser cette fonctionnalité pour exécuter des tâches uniques qui s'étendent sur plusieurs EC2 instances Amazon.	19 novembre 2018
<u>Autorisations de niveau ressource</u>	AWS Batch prend en charge les autorisations au niveau des ressources sur plusieurs opérations d'API.	12 novembre 2018
<u>Support des modèles Amazon EC2 Launch</u>	AWS Batch ajoute la prise en charge de l'utilisation de modèles de lancement avec des environnements informatiques.	12 novembre 2018
<u>AWS Batch délais d'expiration des tâches</u>	AWS Batch ajoute la prise en charge du délai d'expiration des tâches. Grâce à cette prise en charge, vous pouvez configurer un délai d'expiration spécifique pour vos tâches, de sorte que si une tâche s'exécute plus longtemps qu'elle ne le devrait, AWS Batch elle soit interrompue.	5 avril 2018

[AWS Batch les emplois
comme EventBridge cibles](#)

AWS Batch des emplois sont disponibles sous forme d'EventBridge objectifs. En créant des règles simples, vous pouvez faire correspondre les événements et soumettre des AWS Batch offres d'emploi en réponse à ces événements.

1er mars 2018

[CloudTrail audit pour AWS
Batch](#)

CloudTrail peut auditer les appels effectués aux actions de AWS Batch l'API.

10 janvier 2018

[Tâches liées à](#)

AWS Batch ajoute la prise en charge des tâches matricielles. Vous pouvez utiliser des tâches matricielles pour le balayage des paramètres et les charges de travail Monte Carlo.

28 novembre 2017

[AWS Batch Balisage étendu](#)

AWS Batch étend la prise en charge de la fonction de balisage. Vous pouvez utiliser cette fonction pour spécifier des balises pour les instances Amazon EC2 Spot lancées dans des environnements informatiques gérés.

26 octobre 2017

[AWS Batch diffusion
d'événements pour EventBridge](#)

AWS Batch ajoute le flux d'événements pour EventBridge. Vous pouvez utiliser le flux d' AWS Batch événements pour recevoir des notifications en temps quasi réel concernant l'état des tâches soumises à vos files d'attente.

24 octobre 2017

[Nouvelles tentatives de travail automatisées](#)

AWS Batch ajoute la prise en charge des nouvelles tentatives de travail. Avec cette mise à jour, vous pouvez appliquer une stratégie de nouvelle tentative à vos tâches et aux définitions de tâches qui permet à vos tâches d'être automatiquement réessayées en cas d'échec.

28 mars 2017

[AWS Batch disponibilité générale](#)

AWS Batch est introduit, conçu pour vous permettre d'exécuter des charges de travail de calcul par lots sur le AWS Cloud.

5 janvier 2017

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.