



Guide de l'utilisateur

AWS Batch



AWS Batch: Guide de l'utilisateur

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

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

Table of Contents

Qu'est-ce que AWS Batch ?	1
Composants de AWS Batch	1
Tâches	1
Définitions de tâches	2
Files d'attente de tâches	2
Environnement de calcul	2
Démarrage	3
Tableau de bord	3
File d'attente de tâches unique	3
CloudWatch Informations sur les conteneurs	4
Journaux d'emplois	4
Configuration	6
Inscrivez-vous pour un Compte AWS	6
Création d'un utilisateur doté d'un accès administratif	7
Créez des rôles IAM pour vos environnements informatiques et vos instances de conteneur	8
Création d'une paire de clés	9
Création d'un VPC	11
Création d'un groupe de sécurité	12
Installez le AWS CLI	14
Prise en main	15
Prérequis	15
Mise en route - Amazon EC2	15
Création d'un environnement informatique	15
Création d'une file d'attente de tâches	21
Créer une définition de tâche	21
Créez une tâche	26
Vérifier et créer	26
Pour démarrer - Fargate	26
Création d'un environnement informatique	26
Création d'une file d'attente de tâches	28
Créer une définition de tâche	28
Créez une tâche	32
Vérifier et créer	32
AWS Batch sur Amazon EKS	32

Prérequis	33
Étape 1 : Préparation de votre cluster Amazon EKS pour AWS Batch	35
Étape 2 : Création d'un environnement informatique Amazon EKS	38
Étape 3 : créer une file d'attente de tâches et associer l'environnement informatique	41
Étape 4 : Création d'une définition de tâche	41
Étape 5 : Soumettre une offre d'emploi	42
(Facultatif) Soumettre une tâche avec des dérogations	43
AWS Batch sur les clusters privés Amazon EKS	44
Tâches	57
Soumettre une offre d'emploi	57
États des tâches	60
Variables d'environnement de tâche	64
Nouvelles tentatives de travail automatisées	65
Dépendances liées aux emplois	67
Expiration des délais d'exécution des tâches	67
Offres d'emploi Amazon EKS	69
Mappez une tâche en cours d'exécution à un pod et à un nœud	70
Comment faire correspondre un module en cours d'exécution à sa fonction	71
Tâches liées à	72
Exemple de flux de travail sur une matrice	75
Tutoriel : Utilisation de Array Job Index	79
Tâches parallèles sur plusieurs nœuds	85
Variables d'environnement	86
Groupes de nœuds	86
Cycle de vie du travail	87
Considérations relatives à l'environnement informatique	88
Tâches sur GPU	89
Pour créer une tâche basée sur un GPU sur les ressources Amazon EKS	91
Pour créer un Kubernetes cluster basé sur un GPU sur Amazon EKS	92
Pour créer une définition de tâche GPU Amazon EKS	94
Pour exécuter une tâche GPU dans votre cluster Amazon EKS	95
Rechercher et filtrer des AWS Batch offres d'emploi	95
Journaux d'emplois	96
Informations sur le job	97
Définitions des emplois	99
Création d'une définition de tâche à nœud unique	99

Création d'une définition de tâche à nœud unique sur les ressources Amazon EC2	100
Création d'une définition de tâche à nœud unique sur les ressources AWS Fargate	106
Création d'une définition de tâche à nœud unique sur les ressources Amazon EKS	113
Création d'une définition de tâche parallèle sur plusieurs nœuds	117
Création d'une définition de tâche parallèle à nœuds multiples sur les ressources Amazon EC2	118
Création de définitions de tâches à l'aide de ContainerProperties	125
Paramètres de définition du job pour ContainerProperties	133
Création de définitions de tâches à l'aide de EcsProperties	178
ContainerProperties par rapport aux définitions de EcsProperties poste	179
Modifications générales apportées aux AWS Batch API	180
Définitions de tâches multi-conteneurs pour Amazon ECS	180
Définitions de tâches multi-conteneurs pour Amazon EKS	181
AWS Batch scénarios de travail utilisant EcsProperties	182
Utilisation du pilote du journal awslogs	188
Options du pilote du journal awslogs disponibles	189
Spécification d'une configuration de journal dans la définition de votre tâche	191
Spécification des données sensibles	193
Utilisation de Secrets Manager	194
Utilisation du Systems Manager Parameter Store	202
Authentification par registre privé pour les tâches	206
Autorisations IAM requises pour l'authentification de registre privé	208
Utilisation de l'authentification de registre privé	209
Volumes Amazon EFS	210
Considérations relatives aux volumes Amazon EFS	210
Utilisation des points d'accès Amazon EFS	211
Spécification d'un système de fichiers Amazon EFS dans votre définition de tâche	212
Exemples de définitions de tâches	215
Utiliser des variables d'environnement	216
Utilisation de la substitution de paramètres	217
Tester les fonctionnalités du GPU	218
Tâche parallèle sur plusieurs nœuds	219
Files d'attente pour les jobs	220
Création d'une file d'attente de tâches	220
Création d'une file d'attente de tâches Fargate	220
Création d'une file d'attente de tâches Amazon EC2	221

Création d'une file d'attente de tâches Amazon EKS	223
Modèle de file d'attente de tâches	224
Paramètres de la file d'attente de tâches	225
Nom de la file d'attente de tâches	225
État de la file d'attente de tâches, actions de limitation de durée	226
Priorité	226
Politique de planification	226
État	227
Ordre de l'environnement de calcul	227
Balises	228
Affichage de l'état de la file d'attente	229
Affichage des informations relatives à la file d'attente	229
Planification des tâches	231
Partager les identifiants	231
Planification du partage équitable	232
Environnement de calcul	234
Environnements informatiques gérés	234
Prise en compte lors de la création de tâches parallèles à nœuds multiples	237
Environnements informatiques non gérés	237
AMI de ressources de calcul	238
Spécification de l'AMI des ressources de calcul	240
Création d'une AMI de ressources de calcul	242
Utilisation d'une AMI de charge de travail du GPU	245
Obsolète d'Amazon Linux	251
Support de modèle de lancement	252
Données utilisateur Amazon EC2 dans les modèles de lancement	254
Création d'un environnement informatique	258
Pour créer un environnement informatique géré à l'aide des ressources AWS Fargate	259
Pour créer un environnement informatique géré à l'aide des ressources EC2	261
Pour créer un environnement informatique non géré à l'aide des ressources EC2	267
Pour créer un environnement informatique géré à l'aide des ressources Amazon EKS	267
Modèle d'environnement de calcul	271
Paramètres de l'environnement de calcul	273
Nom de l'environnement de calcul	274
Type	274
État	274

Ressources informatiques	275
Configuration d'Amazon EKS	288
Rôle de service	289
Balises	290
Configurations EC2	290
Stratégies d'allocation	291
Mise à jour des environnements informatiques	293
Mise à jour de l'ID AMI	296
Environnements informatiques Amazon EKS	297
Sélection de l'AMI par défaut	298
Versions de Kubernetes prises en charge	299
Mise à jour de la Kubernetes version de l'environnement informatique	299
Responsabilité partagée des Kubernetes nœuds	300
Exécution DaemonSet d'un nœud AWS Batch géré	301
Personnalisation à l'aide de modèles de lancement	302
Gestion de mémoire	305
Réservation de mémoire système	306
Affichage de la mémoire d'une ressource de calcul	307
Considérations relatives à la mémoire et au vCPU pour AWS Batch Amazon EKS	307
Politiques de planification	314
Création d'une politique de planification	314
Modèle de politique de planification	316
Paramètres de la politique de planification	317
Nom de la politique de planification	317
Politique de partage équitable	317
Balises	320
Orchestrez les tâches AWS Batch	321
Affichage des détails de la machine d'état	321
Modification d'une machine d'état	322
Exécution d'une machine d'état	322
AWS Batch sur AWS Fargate	323
Quand utiliser Fargate	323
Définitions de poste sur Fargate	324
Files d'attente pour les emplois sur Fargate	326
Environnements informatiques sur Fargate	326
AWS Batch sur Amazon EKS	328

Elastic Fabric Adapter	331
Politiques, rôles et autorisations IAM	334
Structure d'une politique	335
Syntaxe d'une politique	335
Actions pour AWS Batch	336
Amazon Resource Names (ARN) pour AWS Batch	337
Test des autorisations	338
Autorisations au niveau des ressources prises en charge	339
Clés de condition	350
Exemples de politiques	351
Accès en lecture seule	352
Restreindre l'utilisateur, l'image, les privilèges et le rôle	352
Restreindre la soumission de tâches	354
Restreindre la file des tâches	355
Refuser l'action lorsque la condition est que toutes les clés correspondent à des chaînes ...	356
Refuser l'action lorsque des clés de condition correspondent à des chaînes	357
Utiliser la clé de batch:ShareIdentifier condition	358
Stratégie gérée par AWS Batch	359
AWSBatchFullAccess	359
Création de politiques IAM	361
Rôle d'instance Amazon ECS	361
Rôle de la flotte Amazon EC2 Spot	364
Créez des rôles pour les flottes ponctuelles Amazon EC2 dans AWS Management	
Console	365
Créez des rôles pour les flottes Amazon EC2 Spot avec AWS CLI	366
EventBridge Rôle IAM	368
EventBridge	370
AWS Batch Événements	371
Événements de modification de l'état de tâche	371
Événements bloqués dans la file d'offres d'emploi	373
Utilisation des notifications AWS utilisateur avec AWS Batch	375
AWS Batch les emplois en tant que EventBridge cibles	375
Création d'une tâche planifiée	376
Création d'une règle avec un modèle d'événement	379
Transformateur d'entrée d'événement	382
Tutoriel : Écouter AWS Batch EventBridge	385

Prérequis	385
Étape 1 : Création de la fonction Lambda	385
Étape 2 : Enregistrer une règle d'événement	386
Étape 3 : Testez votre configuration	388
Tutoriel : envoi d'alertes Amazon Simple Notification Service en cas d'échec d'un Job	389
Prérequis	389
Étape 1 : créer une rubrique Amazon SNS et s'y abonner	389
Étape 2 : Enregistrer une règle d'événement	390
Étape 3 : Tester une règle	392
Règle alternative : Batch Job Queue bloquée	392
CloudWatch Journaux	393
Ajouter une politique CloudWatch Logs IAM	393
Installation et configuration de l' CloudWatch agent	395
Afficher les CloudWatch journaux	396
Utiliser CloudWatch les journaux pour surveiller AWS Batch les tâches Amazon EKS	398
Prérequis	398
Installation AWS pour Fluent Bit	398
Activez Fluent Bit pour les AWS Batch nœuds	398
CloudWatch Informations sur les conteneurs	400
Activer Container Insights	400
CloudTrail	402
Informations AWS Batch dans CloudTrail	402
Présentation des entrées des fichiers journaux AWS Batch	403
Création d'un cloud privé virtuel	406
Création d'un VPC	406
Étapes suivantes	407
Sécurité	408
Gestion de l'identité et des accès	408
Public ciblé	409
Authentification par des identités	410
Gestion des accès à l'aide de politiques	414
Comment AWS Batch fonctionne avec IAM	417
Rôle IAM d'exécution	424
Exemples de politiques basées sur l'identité	426
Prévention du cas de figure de l'adjoint désorienté entre services	430
Résolution des problèmes	432

Utilisation des rôles liés à un service	434
AWS politiques gérées	442
Points de terminaison d'un VPC	457
Considérations	458
Création d'un point de terminaison d'interface	459
Création d'une politique de point de terminaison	460
Validation de la conformité	461
Sécurité de l'infrastructure	462
Balisage de vos ressources	464
Principes de base des balises	464
Balisage de vos ressources	465
Restrictions liées aux étiquettes	466
Gestion des étiquettes à l'aide de la console	467
Ajout de balises sur une ressource individuelle lors de la création	467
Ajout et suppression de balises sur une ressource individuelle	467
Gestion des balises à l'aide de la CLI ou de l'API	468
Service Quotas	470
Résolution des problèmes	471
AWS Batch	472
INVALIDenvironnement informatique	472
Offres d'emploi bloquées dans un RUNNABLE statut	474
Instances ponctuelles non étiquetées lors de la création	480
Les instances ponctuelles ne sont pas réduites	480
Impossible de récupérer les secrets de Secrets Manager	482
Impossible de contourner les exigences en ressources liées à la définition des tâches	482
Message d'erreur lorsque vous mettez à jour le <code>desiredvCpus</code> paramètre	484
AWS Batch sur Amazon EKS	484
INVALIDenvironnement informatique	484
AWS Batch sur Amazon EKS, le RUNNABLE statut de la tâche est bloqué	488
Vérifiez que le <code>aws-auth ConfigMap</code> est correctement configuré	489
Les autorisations ou les liaisons RBAC ne sont pas correctement configurées	490
Bonnes pratiques	492
Quand utiliser AWS Batch	492
Liste de contrôle à exécuter à grande échelle	493
Optimisez les conteneurs et les AMI	494
Choisissez la bonne ressource d'environnement informatique	495

Amazon EC2 On-Demand ou Amazon EC2 Spot	496
Utilisez les meilleures pratiques d'Amazon EC2 Spot pour AWS Batch	497
Erreurs courantes et résolution des problèmes	499
Historique du document	503
.....	dx

Qu'est-ce que AWS Batch ?

AWS Batch vous aide à 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.

Rubriques

- [Composants de AWS Batch](#)
- [Démarrage](#)
- [Tableau de bord](#)

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.

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 AWS Fargate les ressources Amazon EC2 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 nom ou par ID, et peuvent dépendre de la réussite de l'exécution d'autres tâches. Pour plus d'informations, consultez [Tâches](#).

Définitions de tâches

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, veuillez consulter la page [Définitions des emplois](#).

Files d'attente de tâches

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.

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 ou EC2) à plusieurs niveaux de détail. Vous pouvez configurer des environnements informatiques qui utilisent un type particulier d'instance EC2, 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 vCPU 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 plus d'informations, consultez [Environnement de calcul](#).

Démarrage

Démarrez avec AWS Batch en créant une définition de tâche, un environnement de calcul et une file d'attente de tâches dans la console AWS Batch.

L'assistant de AWS Batch première exécution vous permet de créer un environnement informatique et une file d'attente de tâches et de soumettre un exemple de tâche Hello World. Si vous avez déjà une image Docker dans laquelle vous souhaitez vous lancer AWS Batch, vous pouvez créer une définition de tâche avec cette image et la soumettre à votre file d'attente à la place. Pour plus d'informations, consultez [Commencer avec AWS Batch](#).

Tableau de bord

Sur le 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 [Environnement de calcul](#).

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

File d'attente de tâches unique

Ce widget affiche des informations détaillées sur une seule file de tâches.

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. Choisissez Ajouter des 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.

CloudWatch Informations sur les conteneurs

Ce widget affiche des métriques agrégées pour les environnements de AWS Batch calcul et les tâches. Pour plus d'informations sur Container Insights, consultez [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. Choisissez Ajouter des 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.

Journaux d'emplois

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 " Journaux d'emplois"](#).

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. Choisissez Ajouter des 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.

Configuration avec AWS Batch

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 console Amazon EC2 pour la plupart de ces étapes.

Effectuez les tâches suivantes pour vous préparer AWS Batch. Si vous avez déjà effectué l'une de ces étapes, vous pouvez passer directement à l'installation du AWS CLI.

Rubriques

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

Inscrivez-vous pour un Compte AWS

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

Pour vous inscrire à un Compte AWS

1. Ouvrez <https://portal.aws.amazon.com/billing/signup>.

2. Suivez les instructions en ligne.

Dans le cadre de la procédure d'inscription, vous recevrez un appel téléphonique 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. Pour des raisons de sécurité, attribuez un accès administratif à un utilisateur et utilisez uniquement l'utilisateur root pour effectuer [les tâches nécessitant un accès utilisateur root](#).

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

Création d'un utilisateur doté d'un accès administratif

Après vous être inscrit à un Compte AWS, sécurisez l'utilisateur racine d'un compte AWS AWS IAM Identity Center, activez et créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

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

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

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

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

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

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

2. Dans IAM Identity Center, accordez 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 l'utilisateur AWS IAM Identity Center.

Connectez-vous en tant qu'utilisateur disposant 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'utilisateur Connexion à AWS.

Attribuer l'accès à des utilisateurs supplémentaires

1. Dans IAM Identity Center, créez un ensemble d'autorisations conforme aux meilleures pratiques en matière d'application des autorisations du moindre privilège.

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

2. Affectez des utilisateurs à un groupe, puis attribuez un accès d'authentification unique au groupe.

Pour obtenir des instructions, consultez la section [Ajouter des 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 APIs AWS en votre nom. Créez un rôle IAM 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

L'environnement de AWS Batch calcul 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 AWS Batch console, vous pouvez passer à la section suivante. Si vous envisagez d'utiliser le à la AWS CLI place, suivez les procédures [Rôle d'instance Amazon ECS](#) avant [Utilisation de rôles liés à un service pour AWS Batch](#) et avant de créer votre premier environnement informatique.

Création d'une paire de clés

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 AWS Batch calcul, 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 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`.

- 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.

- 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 la section [Paires de clés Amazon EC2](#) dans le guide de l'utilisateur Amazon EC2.

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

- Téléchargez PuTTY depuis <http://www.chiark.greenend.org.uk/~sgtatham/putty/>, puis installez-le. Assurez-vous d'installer toute la suite.
- Lancez PuTTYgen (par exemple, dans le menu Démarrer, choisissez Tous les programmes, PuTTY et PuTTYgen).
- 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 ayant 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 sur l'enregistrement de la clé sans une phrase passe. 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 console Amazon EC2](#) 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

Option	Valeur	
Nom	Vous pouvez, si vous le souhaitez, nommer votre VPC.	
Bloc d'adresse CIDR IPv4	Entrée manuelle CIDR IPv4 La taille du bloc d'adresse CIDR doit être comprise entre /16 et /28.	
Bloc d'adresse CIDR IPv6	Pas de bloc d'adresse CIDR IPv6	
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 l'utilisateur Amazon EC2.

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 Security Groups (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, consultez [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, voir [Installation de l'interface de ligne de commande AWS](#) dans le guide de AWS Command Line Interface l'utilisateur.

Commencer avec AWS Batch

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.

Prérequis

Veillez à effectuer les opérations suivantes avant de démarrer l'assistant de AWS Batch première exécution :

- Effectuez les étapes décrites dans [Configuration avec AWS Batch](#).
- Vérifiez que vous Compte AWS disposez des [autorisations requises](#).

Mise en route - Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) offre une capacité de calcul évolutive dans le AWS Cloud. L'utilisation d'Amazon EC2 vous dispense d'investir à l'avance dans du matériel et, par conséquent, vous pouvez développer et déployer les applications plus rapidement.


Vous pouvez utiliser Amazon EC2 pour lancer autant de serveurs virtuels que nécessaire, configurer la sécurité et les réseaux, et gérer le stockage. Amazon EC2 vous permet d'augmenter ou de diminuer l'échelle afin de gérer les modifications en termes d'exigences ou de pics de popularité, et réduire ainsi le besoin de prévoir le trafic.

Création d'un environnement informatique

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


1. Ouvrez l'[assistant de première exécution de la AWS Batch console](#).

2. Pour Sélectionner le type 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 profil d'instance existant auquel sont attachées les autorisations IAM requises. Ce profil d'instance permet aux instances de conteneur Amazon ECS de votre environnement informatique d'appeler les opérations d' AWS API requises. Pour plus d'informations, consultez [Rôle d'instance Amazon ECS](#).
6. (Facultatif) Une balise est une étiquette attribuée à une ressource. Pour ajouter un tag ou un tag Amazon EC2, développez Tags, puis choisissez Ajouter un tag. Entrez une paire clé-valeur, puis choisissez à nouveau Ajouter une balise.

 Important

Si vous choisissez Ajouter un tag, vous devez saisir une paire clé-valeur et choisir à nouveau Ajouter un tag ou choisir Supprimer un tag.

7. (Facultatif) Dans la section Configuration de l'instance pour Utiliser les instances Spot Amazon EC2, activez Activer l'utilisation d'instances Spot.
8. (Spot uniquement) Pour le pourcentage maximum du prix à la demande, entrez le pourcentage maximum du prix à la demande que vous souhaitez payer pour les ressources Spot.
9. (Facultatif) (Spot uniquement) Pour le rôle de flotte Spot, choisissez un rôle IAM Amazon EC2 Spot Fleet existant à appliquer à votre environnement de calcul Spot. Si vous ne possédez pas encore de rôle IAM Amazon EC2 Spot Fleet, vous devez d'abord en créer un. Pour plus d'informations, consultez [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 gérée par Amazon SpotFleetTaggingRole EC2. La politique SpotFleetRole gérée par AmazonEC2 ne dispose pas des autorisations requises pour baliser les instances Spot. Pour plus d'informations, consultez [Instances](#)

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

10. Pour un nombre minimal de vCPU, choisissez le nombre minimal de vCPU EC2 gérés par votre environnement informatique, quelle que soit la demande de files d'attente des tâches.
11. Pour les vCPU souhaités, choisissez le nombre de vCPU EC2 avec lesquels votre environnement de calcul démarre. À mesure que la demande de files d'attente de tâches AWS Batch augmente, le nombre souhaité de vCPU augmente et des instances EC2 sont ajoutées. Le nombre de vCPU peut augmenter jusqu'au nombre maximum de vCPU. À mesure que la demande AWS Batch diminue, le nombre souhaité de vCPU diminue et des instances sont supprimées. Le nombre de diminue jusqu'au nombre minimum de vCPU.
12. Dans Nombre maximal de vCPU, choisissez le nombre maximal de vCPU EC2 que votre environnement de calcul peut accommoder, quelle que soit la demande de la file d'attente de tâches.
13. Pour les types d'instances autorisés, choisissez les types d'instances Amazon EC2 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`. Vous pouvez également choisir de `optimal` sélectionner des types d'instances (parmi les familles d'R4instances `C4M4`, `et`) qui répondent à la demande de vos files d'attente de travail.

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 adapte les GPU 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 de la g5 famille `p2` `p3` `p4p5`, `g3`, `g3s`, `g4`, ou.

 Note

Utilise actuellement des types d'instances issus des familles d'instances C4M4,, et. Si Régions AWS aucun type d'instance n'est issu de ces familles d'instances, les types d'instance des familles C5M5, et d'instances R5 sont utilisés.

14. Développez Additional configuration (Configuration supplémentaire).
15. (Facultatif) Pour Groupe de placement, entrez un nom de groupe de placement pour regrouper les ressources dans l'environnement informatique.
16. (Facultatif) Pour la paire de clés EC2, 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 clés Amazon EC2, consultez la section Paires de [clés Amazon EC2](#) et instances Linux.
17. 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 EC2 On-Demand, et SPOT_CAPACITY_OPTIMIZED pour les environnements de calcul EC2 Spot. Pour plus d'informations, consultez [the section called "Stratégies d'allocation"](#).
18. (Facultatif) Pour la configuration EC2, choisissez Ajouter une configuration EC2. Choisissez les valeurs de remplacement du type d'image et de l'identifiant de l'image AWS Batch pour fournir des informations permettant de sélectionner Amazon Machine Images (AMI) pour les instances de 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

Peut être utilisé pour les familles d'instances autres que le GPU ou AWS Graviton. Le support standard pour Amazon Linux AMI a pris fin. Pour plus d'informations, consultez [AMI Amazon Linux](#).

Note

L'AMI que vous choisissez pour un environnement de calcul doit correspondre à l'architecture des types d'instances que vous souhaitez utiliser pour cet environnement de calcul. Par exemple, si votre environnement informatique utilise des types d'A1instances, l'AMI de ressources de calcul que vous choisissez doit prendre en charge Arm les instances. Amazon ECS vend à la fois des Arm versions x86 et des versions 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.

19. (Facultatif) Pour le modèle de lancement, sélectionnez un modèle de lancement Amazon EC2 existant pour configurer vos ressources de calcul. La version par défaut du modèle est automatiquement renseignée. Pour plus d'informations, consultez [Support de modèle de lancement](#).

Note

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

20. (Facultatif) Pour Version du modèle de lancement, saisissez `$Default`, `$Latest` ou un numéro de version spécifique à utiliser.

 Important

Une fois l'environnement informatique créé, la version du modèle de lancement utilisée n'est pas modifiée même si la `$Latest` version `$Default` ou du modèle de lancement est mise à jour. Pour utiliser une nouvelle version du modèle de lancement, créez d'abord un nouvel environnement de calcul, puis ajoutez-le à la file d'attente de tâches existante. Supprimez ensuite l'ancien environnement informatique de la file d'attente des tâches, puis supprimez l'ancien environnement informatique.

21. Dans la section Configuration du réseau :

- a. Pour l'ID Virtual Private Cloud (VPC), choisissez un Amazon VPC.
- b. Pour les sous-réseaux, les sous-réseaux correspondants Compte AWS sont répertoriés. Si vous souhaitez créer un ensemble personnalisé de sous-réseaux, choisissez Effacer les sous-réseaux, puis choisissez les sous-réseaux souhaités.

 Important

Les ressources de calcul doivent communiquer avec le point de terminaison VPC Amazon ECS via un point de terminaison VPC ou plusieurs adresses IP publiques. Pour plus d'informations, consultez la section [Points de terminaison VPC de l'interface Amazon ECS](#) (). [AWS PrivateLink](#) Si votre instance n'a pas de point de terminaison VPC configuré ou d'adresse IP publique, vous pouvez utiliser la traduction d'adresses réseau (NAT). Pour plus d'informations sur le NAT, consultez les sections [Passerelles NAT](#) et [Création d'un cloud privé virtuel](#) .

- c. Pour les groupes de sécurité, choisissez les groupes de sécurité Amazon EC2 que vous souhaitez associer à l'instance. Si vous souhaitez créer un ensemble personnalisé de groupes de sécurité, choisissez Effacer les groupes de sécurité. Choisissez ensuite les groupes de sécurité que vous souhaitez.


22. Choisissez Suivant.

Création d'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 plus d'informations, consultez [Files d'attente pour les jobs](#).

Pour créer une file d'attente de tâches pour une orchestration Amazon EC2, 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. Pour Priorité, entrez un entier compris entre 0 et 100 pour la file d'attente des tâches.

 Important

Les valeurs entières les plus élevées se voient attribuer une priorité plus élevée par le AWS Batch planificateur.

3. Choisissez Suivant.

Créer 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. Dans la section Configuration générale :
 - a. Dans la section Configuration générale pour Nom, spécifiez un nom unique pour votre environnement informatique. Le nom peut comporter jusqu'à 128 caractères. Le nom peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-) et des traits de soulignement (_).
 - b. (Facultatif) Pour le délai d'exécution, entrez la durée (en secondes) pendant laquelle une tâche inachevée se termine après.

⚠ Important

Le délai minimum est de 60 secondes.

- c. (Facultatif) Une balise est une étiquette attribuée à une ressource. Pour ajouter un tag, développez Tags, puis choisissez Ajouter un tag. Entrez une paire clé-valeur, puis choisissez à nouveau Ajouter une balise.

⚠ Important

Si vous choisissez Ajouter un tag, vous devez saisir une paire clé-valeur et choisir à nouveau Ajouter un tag ou choisir Supprimer un tag.

- d. (Facultatif) Activez les balises Propagate pour propager les balises vers la tâche Amazon Elastic Container Service.

2. Dans la section Configuration du conteneur :


- a. Pour Image, entrez le nom de l'image utilisée pour lancer le conteneur. Par défaut, toutes les images du registre Docker Hub sont disponibles. Vous pouvez également spécifier d'autres référentiels au format repository-url/image:tag. Le paramètre peut comporter jusqu'à 255 caractères. Le paramètre peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-), des traits de soulignement (_), des points (:), des points (.), des barres obliques (/) et des signes numériques (#). Le paramètre correspond Image à la section [Créer un conteneur](#) de l'[API Docker Remote](#) et au IMAGE paramètre de [docker run](#).

📘 Note

Docker l'architecture de l'image doit correspondre à l'architecture du processeur des ressources de calcul sur lesquelles elles sont planifiées. Par exemple, les Docker images Arm basées ne peuvent être exécutées que sur Arm des ressources informatiques basées.

- 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. Dans Commande, précisez la commande à transmettre au conteneur. 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 paramètre Docker `CMD`, consultez <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note

Vous pouvez utiliser des valeurs par défaut de substitution de paramètres et des espaces réservés dans votre commande. Pour plus d'informations, consultez [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. (Facultatif) Pour la configuration du rôle Job, choisissez un rôle IAM autorisé à accéder aux AWS API. 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.

- e. (Facultatif) Vous pouvez ajouter des paramètres à la définition de tâche sous forme de mappages clé-valeur pour remplacer les valeurs par défaut de définition de tâche. Pour ajouter un paramètre :
- Pour Paramètres, choisissez Ajouter un paramètre. Entrez une paire clé-valeur, puis sélectionnez à nouveau Ajouter un paramètre.

 Important

Si vous choisissez Ajouter un paramètre, vous devez configurer au moins un paramètre ou choisir Supprimer le paramètre.

- f. Dans la section Configuration de l'environnement pour les vCPU, spécifiez le nombre de vCPU 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.
- g. Pour Mémoire, spécifiez la limite stricte (en MiB) de mémoire à présenter au conteneur de tâches. 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](#).
- h. Dans Nombre de GPU, choisissez le nombre de GPU à réserver pour le conteneur.
- i. (Facultatif) Pour la configuration des variables d'environnement, choisissez Ajouter des variables d'environnement pour ajouter des variables d'environnement à transmettre au 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](#).
- j. (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 [SecretOptions](#) dans. [Paramètres de définition du job pour ContainerProperties](#)
- k. (Facultatif) Dans la section de configuration Linux :

- i. 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](#).
 - ii. 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](#).
 - iii. Activez `Activer le processus d'initialisation` pour exécuter un `init` processus dans le conteneur. Ce processus transmet les signaux et récolte les processus.
- I. (Facultatif) Dans la section Configuration du système de fichiers :
- i. Activez l'option `Activer le système de fichiers en lecture seule` pour supprimer l'accès en écriture au volume.
 - ii. Pour `Taille de la mémoire partagée`, entrez la taille (en MiB) du `/dev/shm` volume.
 - 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. Pour plus d'informations, consultez [Swappiness in. Paramètres de définition du job pour ContainerProperties](#)
 - v. (Facultatif) Développez la configuration supplémentaire.
 - vi. Pour `Tmpfs`, choisissez `Ajouter des tmpfs` pour ajouter un montage. `tmpfs`
 - vii. 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`.

- viii. (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.

3. Choisissez Suivant.

Créez une tâche

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. Choisissez Suivant.

Vérifier et créer

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.

Pour démarrer - Fargate


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](#).

Création d'un environnement informatique

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 Sélectionner le type 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. (Facultatif) Une balise est une étiquette attribuée à une ressource. Pour ajouter un tag, développez Tags, puis choisissez Ajouter un tag. Entrez une paire clé-valeur, puis choisissez à nouveau Ajouter une balise.

 Important

Si vous choisissez Ajouter un tag, vous devez saisir une paire clé-valeur et choisir à nouveau Ajouter un tag ou choisir Supprimer un tag.

6. (Facultatif) Dans la section Configuration de l'instance pour Utiliser la capacité Fargate Spot, activez Activer l'utilisation des instances Spot.
7. Pour Maximum de vCPU, entrez le nombre maximal de vCPU que l'instance peut utiliser.
8. Dans la section Configuration du réseau :
 - a. Pour l'ID Virtual Private Cloud (VPC), choisissez un Amazon VPC.
 - b. Pour les sous-réseaux, les sous-réseaux correspondants Compte AWS sont répertoriés. Si vous souhaitez créer un ensemble personnalisé de sous-réseaux, choisissez Effacer les sous-réseaux, puis choisissez les sous-réseaux souhaités.

 Important

Les ressources de calcul doivent communiquer avec le point de terminaison VPC Amazon ECS via un point de terminaison VPC ou plusieurs adresses IP publiques. Pour plus d'informations, consultez la section [Points de terminaison VPC de l'interface Amazon ECS](#) (). [AWS PrivateLink](#) Si votre instance n'a pas de point de terminaison VPC configuré ou d'adresse IP publique, vous pouvez utiliser la traduction d'adresses réseau (NAT). Pour plus d'informations sur le NAT, consultez les sections [Passerelles NAT](#) et [Création d'un cloud privé virtuel](#) .

- c. Pour les groupes de sécurité, choisissez les groupes de sécurité Amazon EC2 que vous souhaitez associer à l'instance. Si vous souhaitez créer un ensemble personnalisé de groupes de sécurité, choisissez Effacer les groupes de sécurité. Choisissez ensuite les groupes de sécurité que vous souhaitez.
9. Choisissez Suivant.

Création d'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. Pour Priorité, entrez un entier compris entre 0 et 100 pour la file d'attente des tâches.

Important

Les valeurs entières les plus élevées se voient attribuer une priorité plus élevée par le AWS Batch planificateur.

3. Choisissez Suivant.

Créer une définition de tâche

Pour créer la définition de tâche, procédez comme suit :

1. Dans la section Configuration générale :
 - a. Dans Nom, entrez un nom de définition de tâche personnalisé.


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 (_).

- b. (Facultatif) Pour le délai d'exécution, entrez la durée (en secondes) pendant laquelle une tâche inachevée se termine après.

 Important

Le délai minimum est de 60 secondes.

- c. (Facultatif) Une balise est une étiquette attribuée à une ressource. Pour ajouter un tag, développez Tags, puis choisissez Ajouter un tag. Entrez une paire clé-valeur, puis choisissez à nouveau Ajouter une balise.

 Important

Si vous choisissez Ajouter un tag, vous devez saisir une paire clé-valeur et choisir à nouveau Ajouter un tag ou choisir Supprimer un tag.

- d. (Facultatif) Activez les balises Propagate pour propager les balises vers la tâche Amazon Elastic Container Service.
2. Dans la section de configuration de la plateforme Fargate :
 - a. (Facultatif) Pour la version de la plateforme Fargate, entrez l'environnement d'exécution spécifique que vous souhaitez.
 - b. Pour la plateforme Runtime, sélectionnez un système LINUX ou Windows.
 - c. (Windows uniquement) Pour Famille de systèmes d'exploitation, sélectionnez un système d'exploitation.
 - d. Pour l'architecture du processeur, sélectionnez l'architecture du processeur que vous souhaitez.
 - e. (Facultatif) Activez l'option Attribuer une adresse IP publique pour attribuer une adresse IP publique.
 - f. Pour Stockage éphémère, entrez la quantité de stockage éphémère que vous souhaitez.

 Note

Par défaut, 20 GiB de stockage éphémère sont utilisés. Pour utiliser un stockage éphémère supplémentaire, entrez une valeur comprise entre 21 GiB et 100 GiB.

- g. 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. Par exemple, vous pouvez choisir ecsTaskExecutionRôle.


3. Dans la section Configuration du conteneur :

- a. Pour Image, entrez le nom de l'image utilisée pour lancer le conteneur. Par défaut, toutes les images du registre Docker Hub sont disponibles. Vous pouvez également spécifier d'autres référentiels au format `repository-url/image:tag`. Le paramètre 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 (#). Le paramètre correspond Image à la section [Créer un conteneur](#) de l'[API Docker Remote](#) et au IMAGE paramètre de [docker run](#).

Note

Docker l'architecture de l'image doit correspondre à l'architecture du processeur des ressources de calcul sur lesquelles elles sont planifiées. Par exemple, les Docker images Arm basées ne peuvent être exécutées que sur Arm des ressources informatiques basées.

- 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. Dans Commande, précisez la commande à transmettre au conteneur. 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 paramètre Docker `CMD`, consultez <https://docs.docker.com/engine/reference/builder/#cmd>.


 Note

Vous pouvez utiliser des valeurs par défaut de substitution de paramètres et des espaces réservés dans votre commande. Pour plus d'informations, consultez [Paramètres](#).

 Tip

Choisissez Info pour consulter les exemples de code Bash et JSON.

- c. (Facultatif) Vous pouvez ajouter des paramètres à la définition de tâche sous forme de mappages clé-valeur pour remplacer les valeurs par défaut de définition de tâche. Pour ajouter un paramètre :
- Pour Paramètres, choisissez Ajouter un paramètre. Entrez une paire clé-valeur, puis sélectionnez à nouveau Ajouter un paramètre.

 Important

Si vous choisissez Ajouter un paramètre, vous devez configurer au moins un paramètre ou choisir Supprimer le paramètre.

- d. (Facultatif) Dans la section Configuration de l'environnement pour la configuration du rôle Job, choisissez un rôle IAM qui autorise l'utilisation des AWS API.
- e. Dans la section Configuration de l'environnement pour les vCPU, spécifiez le nombre de vCPU 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.
- f. Pour Mémoire, spécifiez la limite stricte (en MiB) de mémoire à présenter au conteneur de tâches. 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](#).
- g. (Facultatif) Pour les variables d'environnement, choisissez Ajouter des variables d'environnement pour ajouter des variables d'environnement à transmettre au 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](#).

4. Choisissez Suivant.

Créez une tâche

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. Choisissez Suivant.

Vérifier et créer

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.

Commencer à utiliser AWS Batch 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

- [Prérequis](#)
- [Étape 1 : Préparation de votre cluster Amazon EKS pour AWS Batch](#)
- [Étape 2 : Création d'un environnement informatique Amazon EKS](#)
- [Étape 3 : créer une file d'attente de tâches et associer l'environnement informatique](#)
- [Étape 4 : Création d'une définition de tâche](#)
- [Étape 5 : Soumettre une offre d'emploi](#)
- [\(Facultatif\) Soumettez une tâche avec des dérogations](#)
- [Commencer à utiliser AWS Batch les clusters privés Amazon EKS](#)
 - [Prérequis](#)
 - [Étape 1 : Préparation de votre cluster EKS pour AWS Batch](#)
 - [Étape 2 : Création d'un environnement informatique Amazon EKS](#)
 - [Étape 3 : créer une file d'attente de tâches et associer l'environnement informatique](#)
 - [Étape 4 : Création d'une définition de tâche](#)
 - [Étape 5 : Soumettre une offre d'emploi](#)
 - [\(Facultatif\) Soumettez une tâche avec des dérogations](#)
 - [Résolution des problèmes](#)

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.
- **eksct1**— 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 eksct1](#) 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 AWS 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.
- **Création d'un cluster Amazon EKS** — Pour plus d'informations, consultez [Getting started with Amazon EKS — eksct1](#) dans le guide de l'utilisateur Amazon EKS.

Note

AWS Batch prend uniquement en charge les clusters Amazon EKS dotés de points de terminaison de serveur d'API dotés d'un accès public, accessibles sur Internet public. Par défaut, les points de terminaison du serveur d'API des clusters Amazon EKS ont un accès public. Pour plus d'informations, consultez la section [Contrôle d'accès aux points de terminaison du cluster Amazon EKS](#) dans le guide de l'utilisateur Amazon EKS.

Note

AWS Batch ne fournit pas d'orchestration de nœuds gérés pour CoreDNS ou d'autres pods de déploiement. Si vous avez besoin de CoreDNS, consultez la section [Ajout du module](#)

[complémentaire CoreDNS Amazon EKS dans le guide de l'utilisateur Amazon EKS](#). Ou, `eksctl create cluster create` pour créer le cluster, il inclut CoreDNS par défaut.

- 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. L'utilisation de AWS Management Console pour créer une ressource de calcul à l'aide des ressources Amazon EKS nécessite des autorisations à la fois pour `eks:DescribeCluster` et `eks:ListClusters`.

Étape 1 : Préparation de 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.

Note

Pour plus d'informations sur l'utilisation de l'autorisation RBAC, consultez la section [Utilisation de l'autorisation RBAC](#) dans le guide de l'utilisateur de Kubernetes.

```
$ 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: ["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-cluster-name \
  --arn "arn:aws:iam::<your-account>:role/AWSServiceRoleForBatch" \
  --username aws-batch
```

Sortie :

```
2022-10-25 20:19:57 [#] adding identity "arn:aws:iam::<your-account>: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 2 : 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 ponctuelles EC2.

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.

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:<region>:123456789012:cluster/<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

- Le `serviceRole` paramètre ne doit pas être spécifié, le rôle AWS Batch lié au service sera alors utilisé. AWS Batch sur Amazon, EKS prend uniquement en charge le AWS Batch rôle lié au service.

- Seules `BEST_FIT_PROGRESSIVESPOT_CAPACITY_OPTIMIZED`, et les stratégies `SPOT_PRICE_CAPACITY_OPTIMIZED` d'allocation sont prises en charge pour les environnements informatiques Amazon EKS.

Note

Nous vous recommandons d'utiliser `SPOT_PRICE_CAPACITY_OPTIMIZED` plutôt que `SPOT_CAPACITY_OPTIMIZED` dans la plupart des cas.

- Pour ce faire `instanceRole`, 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](#).
- Un moyen d'obtenir des sous-réseaux fonctionnels pour le `subnets` paramètre consiste à utiliser les sous-réseaux publics des groupes de nœuds gérés par Amazon EKS qui ont été créés `eksctl` lors de la création d'un cluster Amazon EKS. Sinon, utilisez des sous-réseaux dotés d'un chemin réseau permettant d'extraire des images.
- Le `securityGroupIds` paramètre peut utiliser le même groupe de sécurité que le cluster Amazon EKS. Cette commande récupère l'ID du groupe de sécurité du cluster.

```
$ eks describe-cluster \
  --name <cluster-name> \
  --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

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

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 plus d'informations, consultez [Résolution des problèmes AWS Batch](#).

Étape 3 : créer une file d'attente de tâches et associer l'environnement informatique

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

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 4 : Création d'une définition de tâche

```
$ 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

- Seules les tâches liées à un seul conteneur sont prises en charge.
- Il existe des considérations relatives aux memory paramètres cpu et. Pour plus d'informations, consultez [Considérations relatives à la mémoire et au vCPU pour AWS Batch Amazon EKS](#).

Étape 5 : Soumettre une offre d'emploi

```
$ aws batch submit-job --job-queue My-Eks-JQ1 \  
  --job-definition MyJob0nEks_Sleep --job-name My-Eks-Job1  
$ aws batch describe-jobs --job <jobId-from-submit-response>
```

Remarques

- Seules les tâches liées à un seul conteneur sont prises en charge.

- Assurez-vous de connaître toutes les considérations pertinentes relatives aux memory paramètres cpu et. Pour plus d'informations, consultez [Considérations relatives à la mémoire et au vCPU pour AWS Batch Amazon EKS](#).
- Pour plus d'informations sur l'exécution de tâches sur les ressources Amazon EKS, consultez [Offres d'emploi Amazon EKS](#).

(Facultatif) Soumettez une tâche avec des dérogations

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

- 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 plus d'informations, consultez [Utiliser CloudWatch les journaux pour surveiller AWS Batch les tâches Amazon EKS](#).

- 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 plus d'informations, voir [Considérations relatives à la mémoire et au vCPU pour AWS Batch Amazon EKS](#).

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. Support pour les clusters Amazon EKS privés sur le marché AWS Batch est généralement disponible dans le [commerce, Régions AWS là où AWS Batch](#) il est disponible.

Les [clusters privés Amazon EKS ne](#) disposent d'aucun accès Internet entrant/sortant et ne disposent que de sous-réseaux privés. 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 votre identifiant de ressource uniforme (URI) du

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

- [Prérequis](#)
- [Étape 1 : Préparation de votre cluster EKS pour AWS Batch](#)
- [Étape 2 : Création d'un environnement informatique Amazon EKS](#)
- [Étape 3 : créer une file d'attente de tâches et associer l'environnement informatique](#)
- [Étape 4 : Création d'une définition de tâche](#)
- [Étape 5 : Soumettre une offre d'emploi](#)
- [\(Facultatif\) Soumettez une tâche avec des dérogations](#)
- [Résolution des problèmes](#)

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. Vous devez également créer toutes les ressources nécessaires, notamment le VPC, les sous-réseaux, les tables de routage, les points de terminaison VPC et le cluster Amazon EKS. Vous devez utiliser le AWS CLI.

- **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. Pour plus d'informations, consultez la section [Configuration rapide avec aws configure](#) dans le Guide de l'AWS Command Line Interface utilisateur.

- **kubect1**— Un outil de ligne de commande permettant de travailler avec des Kubernetes clusters. 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 conçu pour fonctionner avec les clusters Amazon EKS et 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 requises AWS Identity and Access Management (IAM) : le principal de sécurité IAM que vous utilisez doit être autorisé à utiliser les rôles IAM Amazon EKS et les rôles liés à un service AWS 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.
- Création d'un cluster Amazon EKS — Pour plus d'informations, consultez [Getting started with Amazon EKS — eksctl](#) dans le guide de l'utilisateur Amazon EKS.

 Note

AWS Batch ne fournit pas d'orchestration de nœuds gérés pour CoreDNS ou d'autres pods de déploiement. Si vous avez besoin de CoreDNS, consultez la section [Ajout du module complémentaire CoreDNS Amazon EKS dans le guide de l'utilisateur Amazon EKS](#). Ou, `eksctl create cluster create` pour créer le cluster, il inclut CoreDNS par défaut.

- 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. L'utilisation de AWS Management Console pour créer une ressource de calcul à l'aide des ressources Amazon EKS nécessite des autorisations à la fois pour `eks:DescribeCluster` et `eks:ListClusters`.
- Créez un cluster [EKS privé](#) dans la région us-east-1 à l'aide de l'exemple `eksctl` de fichier de configuration.

```
kind: ClusterConfig
apiVersion: eksctl.io/v1alpha5
availabilityZones:
  - us-east-1a
  - us-east-1b
  - us-east-1d
managedNodeGroups:
  privateNetworking: true
privateCluster:
  enabled: true
```

```
skipEndpointCreation: false
```

Créez vos ressources à l'aide de la commande : `eksctl create cluster -f clusterConfig.yaml`

- Les nœuds gérés par lots doivent être déployés sur des sous-réseaux dotés des points de terminaison d'interface VPC dont vous avez besoin. Pour plus d'informations, consultez la section [Exigences relatives aux clusters privés](#).

Étape 1 : Préparation de votre cluster EKS pour AWS Batch

Toutes les étapes sont obligatoires.

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.

Note

Pour plus d'informations sur l'utilisation de l'autorisation RBAC, consultez la section [Utilisation de l'autorisation RBAC](#) dans la Kubernetes documentation.

```
$ 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: ["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-cluster-name \
  --arn "arn:aws:iam::<your-account>:role/AWSServiceRoleForBatch" \
  --username aws-batch
```

Sortie :

```
2022-10-25 20:19:57 [#] adding identity "arn:aws:iam::<your-account>: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 2 : 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 ponctuelles EC2.

Maintenant que le rôle AWSServiceRoleForBatchlié 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.

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:<region>:123456789012:cluster/<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-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

- Le `serviceRole` paramètre ne doit pas être spécifié, le rôle AWS Batch lié au service sera alors utilisé. AWS Batch sur Amazon, EKS prend uniquement en charge le AWS Batch rôle lié au service.

- Seules `BEST_FIT_PROGRESSIVESPOT_CAPACITY_OPTIMIZED`, et les stratégies `SPOT_PRICE_CAPACITY_OPTIMIZED` d'allocation sont prises en charge pour les environnements informatiques Amazon EKS.

Note

Nous vous recommandons d'utiliser `SPOT_PRICE_CAPACITY_OPTIMIZED` plutôt que `SPOT_CAPACITY_OPTIMIZED` dans la plupart des cas.

- Pour ce faire `instanceRole`, 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](#).
- Un moyen d'obtenir des sous-réseaux fonctionnels pour le `subnets` paramètre consiste à utiliser les sous-réseaux publics des groupes de nœuds gérés par Amazon EKS qui ont été créés `eksctl` lors de la création d'un cluster Amazon EKS. Sinon, utilisez des sous-réseaux dotés d'un chemin réseau permettant d'extraire des images.
- Le `securityGroupIds` paramètre peut utiliser le même groupe de sécurité que le cluster Amazon EKS. Cette commande récupère l'ID du groupe de sécurité pour le cluster.

```
$ eks describe-cluster \
  --name <cluster-name> \
  --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

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

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 plus d'informations, consultez [Résolution des problèmes AWS Batch](#).

Étape 3 : créer une file d'attente de tâches et associer l'environnement informatique

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

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 4 : Création d'une définition de tâche

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": "account-id.dkr.ecr.region.amazonaws.com/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
```

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](#).

Étape 5 : Soumettre une offre d'emploi

```
$ aws batch submit-job - -job-queue My-Eks-JQ1 \  
  - -job-definition MyJobOnEks_Sleep - -job-name My-Eks-Job1  
$ aws batch describe-jobs - -job <jobId-from-submit-response>
```

Remarques

- Seules les tâches liées à un seul conteneur sont prises en charge.

- Assurez-vous de connaître toutes les considérations pertinentes relatives aux memory paramètres cpu et. Pour plus d'informations, consultez [Considérations relatives à la mémoire et au vCPU pour AWS Batch Amazon EKS](#).
- Pour plus d'informations sur l'exécution de tâches sur les ressources Amazon EKS, consultez [Offres d'emploi Amazon EKS](#).

(Facultatif) Soumettez une tâche avec des dérogations

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

- 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 plus d'informations, consultez [Utiliser CloudWatch les journaux pour surveiller AWS Batch les tâches Amazon EKS](#).

- 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 plus d'informations, consultez [Considérations relatives à la mémoire et au vCPU pour AWS Batch Amazon EKS](#).

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

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 plus d'informations, consultez [Paramètres de définition du job pour ContainerProperties](#).

Vous pouvez soumettre un grand nombre de tâches simples indépendantes.

Rubriques

- [Soumettre une offre d'emploi](#)
- [États des tâches](#)
- [AWS Batch variables d'environnement de travail](#)
- [Nouvelles tentatives de travail automatisées](#)
- [Dépendances liées aux emplois](#)
- [Expiration des délais d'exécution des tâches](#)
- [Offres d'emploi Amazon EKS](#)
- [Tâches liées à](#)
- [Tâches parallèles sur plusieurs nœuds](#)
- [Tâches sur GPU](#)
- [Pour créer une tâche basée sur un GPU sur les ressources Amazon EKS](#)
- [Rechercher et filtrer des AWS Batch offres d'emploi](#)
- [Journaux d'emplois](#)
- [Informations sur le job](#)


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/](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 plus d'informations, consultez [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 plus d'informations, consultez [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 plus d'informations, consultez [Dépendances liées aux emplois](#).
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 plus d'informations, consultez [Nouvelles tentatives de travail automatisées](#).
 - c. (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 plus

d'informations, consultez [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. Dans Commande, précisez la commande à transmettre au conteneur. Pour les commandes simples, entrez la commande comme vous le feriez pour une invite de commande. Pour les commandes plus complexes (par exemple avec des caractères spéciaux), utilisez la syntaxe JSON.

Note

Ce paramètre ne peut pas contenir de chaîne vide.

- b. Pour les vCPU, entrez le nombre de vCPU 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 plus d'informations, consultez [Ressource de calcul Gestion de la mémoire](#).

- d. (Facultatif) Dans Nombre de GPU, choisissez le nombre de GPU 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.

États des tâches

Lorsque vous soumettez une tâche à une file d'attente de tâches AWS Batch, 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.

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 plus d'informations, consultez [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 [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.

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 plus d'informations, consultez [Tutoriel : Utilisation de l'index de 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 plus d'informations, consultez [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'UID 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 la section [UID](#) dans 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'adresse IPv4 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. Si l'[option `evaluateOnExit`](#) est spécifiée, elle peut contenir jusqu'à 5 stratégies de nouvelle tentative. Si l'[évaluation `OnExit`](#) 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, le travail est remis en `RUNNABLE` état. Pour plus d'informations, consultez [États des tâches](#).

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 [Réessayer la stratégie](#) sections [Création d'une définition de tâche à nœud unique](#) , [Soumettre une offre d'emploi](#) et [Codes d'erreur des tâches arrêtées](#).

Dépendances liées aux emplois

Lorsque vous soumettez une AWS Batch tâche, vous pouvez spécifier les identifiants de la tâche dont dépend la tâche. Dans ce cas, le planificateur AWS Batch garantit que votre tâche soit exécutée uniquement après que les dépendances spécifiées soient 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 plus d'informations, consultez [Tâches liées à](#).

Pour soumettre une AWS Batch tâche comportant des dépendances, consultez [Soumettre une offre d'emploi](#).

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 qui ne devrait vous 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 AWS Batch 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` 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'expiration 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 [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 PropertiesOverride paramètre [eks](#) 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 plus d'informations, consultez [Utiliser CloudWatch les journaux pour surveiller AWS Batch les tâches Amazon EKS](#).

Mappez 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.

Comment faire correspondre un module en cours d'exécution à sa fonction

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 plus d'informations, consultez [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 liées aux emplois](#)
- [Tâches liées à](#)
- [Expiration des délais d'exécution des tâches](#)
- [Nouvelles tentatives de travail automatisées](#)
- [Planification du partage équitable](#)

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](#)
- [Pour créer une tâche basée sur un GPU sur les ressources Amazon EKS](#)
- [Offres d'emploi bloquées dans un RUNNABLE statut](#)

Tâches liées à

Une tâche de tableau est une tâche qui partage des paramètres communs, comme la définition de tâche, les vCPU 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

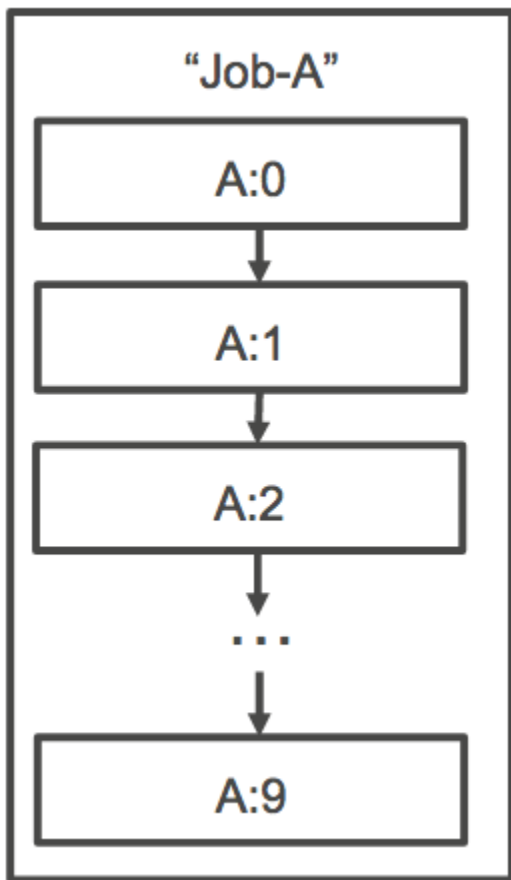
tâche s'exécute et génère 1 000 tâches enfants. La tâche de tableau est une référence ou un pointeur pour gérer toutes les tâches enfants. Ainsi, vous pouvez soumettre des charges de travail importantes en une seule requête. Le délai spécifié dans le `attemptDurationSeconds` paramètre s'applique à chaque tâche enfant. La tâche du tableau parent n'a pas de délai d'expiration.

Lorsque vous soumettez une tâche matricielle, la tâche matricielle 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 `example_job_ID:0` pour le premier travail enfant du tableau.

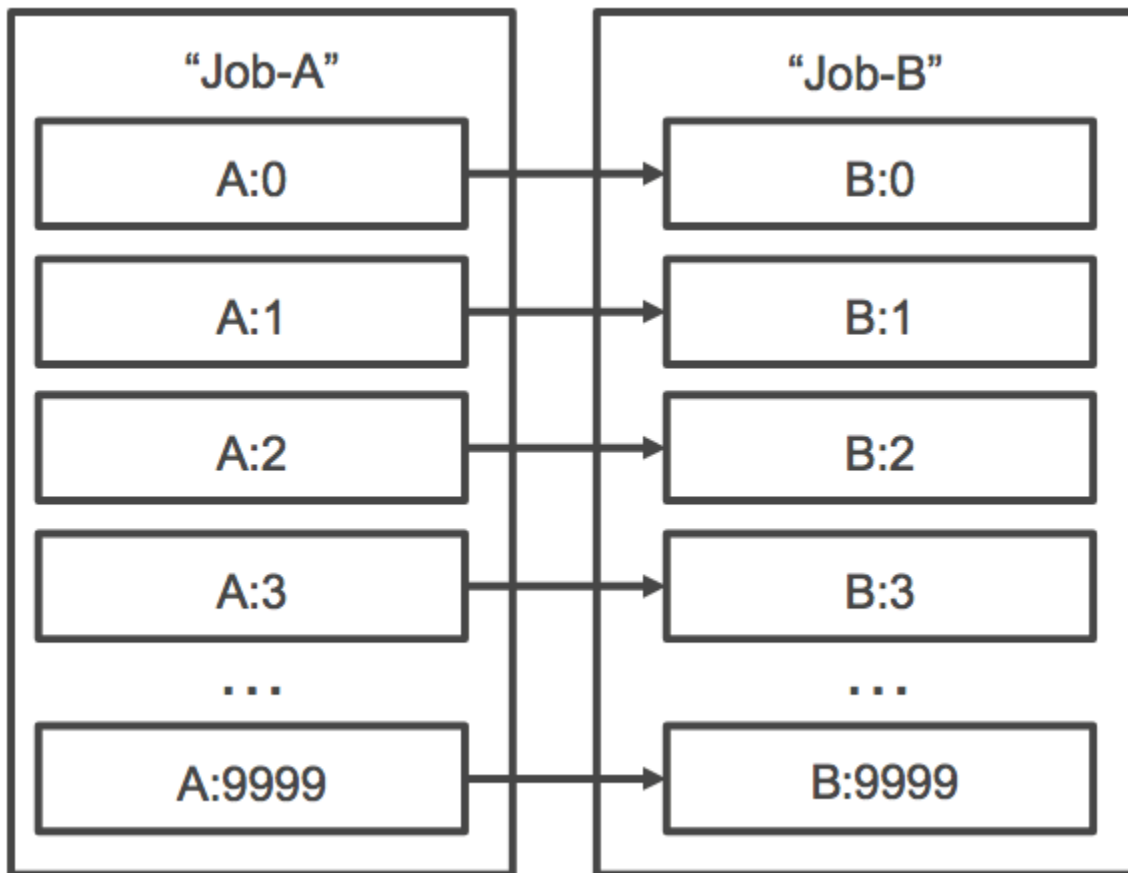
La tâche de matrice parent peut entrer un SUCCEEDED statut SUBMITTED PENDING FAILED,, ou. Une tâche parent de tableau est mise à jour PENDING lorsqu'une tâche enfant est mise à jour vers RUNNABLE. Pour plus d'informations sur les interdépendances entre les emplois, consultez [Dépendances liées aux emplois](#).

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 plus d'informations, consultez [Tutoriel : Utilisation de l'index de 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 suspendez une tâche de tableau parente, toutes les tâches sont annulées ou suspendues avec elle. Vous pouvez annuler ou suspendre des tâches enfants individuelles (leur statut devient alors FAILED) sans affecter les autres tâches enfants. Toutefois, si une tâche de tableau enfant échoue (seule ou en annulant ou en s'arrêtant manuellement), la tâche parent échoue également.

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 [SubmitJobJSON](#) 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 [SubmitJobJSON](#) 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 [SubmitJobJSON](#) 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 [SubmitJobJSON](#) est la suivante.

```

{
  "jobName": "JobE",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobE-Cleanup-and-Notification:1",
  "parameters": {
    "SourceBucket": "s3://JobD-Output-Bucket",
    "Recipient": "pipeline-notifications@mycompany.com"
  },
  "dependsOn": [
    {
      "jobId": "JobD_job_ID"
    }
  ]
}

```

Tutoriel : Utilisation de l'index de 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. Il en résulte que plusieurs AWS Batch tâches peuvent exécuter 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 qui est 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.

Prérequis

Ce didacticiel nécessite la configuration suivante :

- Un environnement AWS Batch informatique. Pour plus d'informations, consultez [Création d'un environnement informatique](#).
- Une file AWS Batch d'attente de tâches et un environnement informatique associé. Pour plus d'informations, consultez [Création d'une file d'attente de tâches](#).
- Ils AWS CLI sont installés sur votre système local. Pour plus d'informations, consultez [Installation d' AWS Command Line Interface](#) dans le Guide de l'utilisateur AWS Command Line Interface .
- Docker installé sur votre système local. Pour en savoir plus, consultez la section [À propos de Docker CE](#) dans la documentation Docker.

Étape 1 : 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
```

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.
```

Étape 2 : Transférez votre image vers 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
```

Étape 3 : créer et enregistrer 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 plus d'informations, consultez [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
```

Étape 4 : Soumettre une tâche AWS Batch de matrice

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 "Prérequis"](#) 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.

Étape 5 : 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/](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.

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, vous pouvez exécuter des applications informatiques hautes performances à grande échelle et des formations 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 entre nœuds basée sur IP. Les exemples incluent Apache MXnet TensorFlow, Caffe2 ou l'interface MPI (Message Passing Interface).

Les tâches parallèles à plusieurs nœuds 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 lancé, 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 plus d'informations, consultez [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 API. AWS Batch [DescribeJobs](#) 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.

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'adresse IPv4 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 dans une seule 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 dans une tâche parallèle à plusieurs nœuds doivent utiliser le même type d'instance.

Cycle de vie du travail

Lorsque vous soumettez une tâche parallèle à nœuds multiples, la tâche entre dans le 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 qui 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. Cela met fin à la tâche ou l'annule.

Considérations relatives à l'environnement informatique

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. Tous les groupes de sécurité appliqués à vos ressources de calcul lui sont également appliqués. Pour plus d'informations, consultez la section [Mise en réseau des tâches avec le mode réseau awsvpc](#) dans le manuel Amazon Elastic Container Service Developer Guide.
- Il est possible que votre environnement informatique ne soit associé qu'à cinq groupes de sécurité au maximum.
- 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. Une communication entre les nœuds doit utiliser l'adresse IP privée ou le nom d'hôte DNS pour le 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 [Création d'un cloud privé virtuel](#).

- 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 de calcul doit disposer de suffisamment de vCPU pour prendre en charge votre tâche parallèle à 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 sur GPU

Les tâches GPU vous permettent d'exécuter des tâches qui utilisent les GPU d'une instance.

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 P2, les instances Amazon EC2 P3, les instances Amazon EC2 P4 et les instances Amazon EC2 P5.](#)

Type d'instance	GPU	Mémoire GPU	vCPU	Mémoire	Bande passante 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

Type d'instance	GPU	Mémoire GPU	vCPU	Mémoire	Bande passante réseau
g3.8xlarge	2	16 GiO	32	244 GiB	10 Gbit/s
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
p2.xlarge	1	12 GiO	4	61 GiB	Élevée
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

Type d'instance	GPU	Mémoire GPU	vCPU	Mémoire	Bande passante réseau
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	4x100 Gbps
p5.48xlarge	8	640 GiB	192	2 Tio	32 x 100 Gbit/s

Note

Seuls les types d'instance prenant en charge un GPU NVIDIA et utilisant une architecture x86_64 sont pris en charge pour les tâches GPU dans AWS Batch. Par exemple, les familles [G4ad](#) et [G5g](#) instances ne sont pas prises en charge.

Le paramètre [ResourceRequirements](#) pour la définition de tâche indique le nombre de GPU à épingler au conteneur. Ce nombre de GPU 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'instances p2 p3 p4 p5 g3g3s,g4,,,, 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 les 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 non 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.

Pour créer une tâche basée sur un GPU sur les ressources Amazon EKS

Cette section explique comment exécuter une charge de travail GPU Amazon EKS sur AWS Batch.

Table des matières

- [Pour créer un Kubernetes cluster basé sur un GPU sur Amazon EKS](#)
- [Pour créer une définition de tâche GPU Amazon EKS](#)
- [Pour exécuter une tâche GPU dans votre cluster Amazon EKS](#)

Pour créer 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 Amazon EKS](#). En outre, tenez également compte des points suivants :

- AWS Batch prend en charge les types d'instances avec les GPU NVIDIA.
- 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 joindre 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
```

Pour créer 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 [Planifier les GPU](#) dans la [Kubernetesdocumentation](#).

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
```

Pour 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"
}
```

Rechercher et filtrer des AWS Batch offres d'emploi

Vous pouvez répertorier les tâches dans une file d'attente à l'aide de la AWS Batch console.

Toutefois, s'il y a de nombreuses tâches dans la file d'attente, il peut être difficile de trouver une tâche spécifique.

Vous pouvez utiliser la recherche et le filtrage pour répertorier les tâches correspondant aux critères de recherche que vous spécifiez.

1. Ouvrez la [AWS Batch console](#).
2. Choisissez Tâches.
3. Activez la recherche et le filtrage.

Note

Si vous avez plusieurs emplois, ce processus peut prendre quelques minutes.

4. Dans le champ Veuillez sélectionner une file d'attente de tâches, choisissez la file d'attente de tâches que vous souhaitez rechercher.

5. Dans la zone Filtrer les ressources par propriété ou valeur, choisissez l'une des propriétés répertoriées.
6. Choisissez l'opérateur que vous souhaitez utiliser. Par exemple, choisissez Status =.

 Tip

Pour utiliser une autre propriété ou un autre opérateur, fermez les critères actuels. Choisissez ensuite la propriété et l'opérateur souhaités.

7. Entrez ou choisissez une valeur de propriété. Par exemple, entrez tout ou partie du nom d'une tâche ou choisissez Status = RUNNABLE.
8. Choisissez le travail que vous souhaitez dans la liste filtrée.

 Tip

Si vous ne trouvez pas le poste que vous souhaitez, faites défiler la liste filtrée.

Journaux d'emplois

Vous pouvez configurer vos AWS Batch tâches pour envoyer des informations de journal à CloudWatch Logs. Ainsi, vous pouvez consulter les différents journaux de vos tâches en un seul endroit pratique. Pour plus d'informations, consultez [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.
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 plus d'informations, consultez [Rechercher et filtrer des AWS Batch offres d'emploi](#).

4. Dans le champ Statut, choisissez le statut du poste que vous souhaitez.
5. Choisissez le poste que vous souhaitez.
6. Sur la page Détails, faites défiler la page vers le bas jusqu'à Job Logs.
7. Choisissez Retrieve logs.
8. Dans Autorisation requise, entrez **OK**, puis choisissez Autoriser pour accepter les CloudWatch frais Amazon.

Note

Pour révoquer votre autorisation de CloudWatch facturation :

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.

9. Passez en revue les données du journal de la AWS Batch tâche.

Tip

Vous pouvez filtrer le journal en fonction des mots clés, du nombre maximal de résultats et du tri. Vous pouvez également choisir l'un des intervalles de temps par défaut ou créer un intervalle personnalisé pour personnaliser les résultats.

Informations sur le job

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.

i 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 plus d'informations, consultez [Rechercher et filtrer des AWS Batch offres d'emploi](#).

4. Choisissez le poste que vous souhaitez.

i 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](#).

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 au moment de l'exécution.

Table des matières

- [Création d'une définition de tâche à nœud unique](#)
- [Création d'une définition de tâche parallèle sur plusieurs nœuds](#)
- [Création de définitions de tâches à l'aide de ContainerProperties](#)
- [Création de définitions de tâches à l'aide de EcsProperties](#)
- [Utilisation du pilote du journal awslogs](#)
- [Spécification des données sensibles](#)
- [Authentification par registre privé pour les tâches](#)
- [Volumes Amazon EFS](#)
- [Exemples de définitions de tâches](#)

Parmi les attributs spécifiés dans une définition de tâche figurent les suivants :

- L'image Docker à utiliser avec le conteneur dans la tâche
- Le nombre de processeurs virtuels et la quantité de mémoire à utiliser avec le conteneur
- La commande que le conteneur doit exécuter au démarrage
- Les éventuelles variables d'environnement à transmettre au conteneur lorsqu'il démarre
- Les volumes de données à utiliser avec le conteneur
- Quel rôle IAM (le cas échéant) votre travail doit utiliser pour les autorisations AWS

Pour obtenir une description complète des paramètres disponibles dans une définition de tâche, consultez [Paramètres de définition du job pour ContainerProperties](#).

Création d'une définition de tâche à nœud unique

Avant de pouvoir exécuter des tâches dans 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 AWS Batch tâche qui n'est pas une tâche parallèle à plusieurs nœuds.

Vous pouvez créer une définition de tâche parallèle à plusieurs nœuds sur les ressources Amazon Elastic Container Service. Pour plus d'informations, consultez [the section called “Création d'une définition de tâche parallèle sur plusieurs nœuds”](#).

Rubriques

- [Création d'une définition de tâche à nœud unique sur les ressources Amazon EC2](#)
- [Création d'une définition de tâche à nœud unique sur les ressources AWS Fargate](#)
- [Création d'une définition de tâche à nœud unique sur les ressources Amazon EKS](#)

Création d'une définition de tâche à nœud unique sur les ressources Amazon EC2

Pour créer une nouvelle définition de tâche sur les ressources Amazon EC2 :


1. Ouvrez la AWS Batch console à l'[adresse https://console.aws.amazon.com/batch/](https://console.aws.amazon.com/batch/).
2. Dans la barre de navigation, choisissez le Région AWS à utiliser.
3. Dans le volet de navigation de gauche, choisissez Job definitions.
4. Choisissez Créer.
5. Pour le type d'orchestration, choisissez Amazon Elastic Compute Cloud (Amazon EC2).
6. Pour la configuration de la plate-forme EC2, 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) 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 plus d'informations, consultez [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 Ajouter une étiquette.
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 comporter jusqu'à 225 caractères. Il peut contenir des lettres majuscules et minuscules, des chiffres, des traits d'union (-), des traits de soulignement (_), des 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


Docker l'architecture de l'image doit correspondre à l'architecture du processeur des ressources de calcul sur lesquelles elles sont planifiées. Par exemple, les Docker images Arm basées ne peuvent être exécutées que sur Arm des ressources informatiques basées.

- 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 la syntaxe des commandes, choisissez Bash ou JSON.
- c. Dans Commande, précisez la commande à transmettre au conteneur. Pour des commandes plus simples, entrez la commande comme vous le feriez pour une invite de commande. Vérifiez ensuite que le JSON résultat est correct et transmis au Docker daemon. Pour les commandes plus complexes (par exemple, avec des caractères spéciaux), utilisez la syntaxe JSON.

 Tip


Choisissez Info pour afficher Bash et JSON coder des exemples.

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 paramètre Docker `CMD`, veuillez consulter <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 plus d'informations, consultez [Paramètres](#).


- d. (Facultatif) Pour le rôle d'exécution, spécifiez un rôle IAM qui autorise les agents de conteneur Amazon ECS à effectuer des appels d'AWSAPI 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.
- e. Pour la configuration du rôle Job, choisissez un rôle IAM autorisé à accéder aux AWS API. 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 les vCPU, entrez le nombre de vCPU 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 plus d'informations, consultez [Ressource de calcul Gestion de la mémoire](#).

- c. Dans Nombre de GPU, choisissez le nombre de GPU 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 contenant. Pour plus d'informations, consultez [SecretOptions](#) dans [Paramètres de définition du job pour ContainerProperties](#)
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 `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 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 [Swappiness in. Paramètres de définition du job pour ContainerProperties](#)
 - 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 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. (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, consultez [LogDriver](#) dans [Paramètres de définition du job pour ContainerProperties](#).

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 [SecretOptions](#) dans. [Paramètres de définition du job pour ContainerProperties](#)


22. Choisissez Page suivante.
23. 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 AWS Fargate

Pour créer une nouvelle définition de tâche sur les AWS Fargate ressources :

1. Ouvrez la AWS Batch console à l'[adresse https://console.aws.amazon.com/batch/](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 de gauche, choisissez Job definitions.
4. Choisissez Créer.
5. Pour le type d'orchestration, choisissez Fargate. Pour plus d'informations, consultez [AWS Batch sur AWS 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 plus d'informations, consultez [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. Dans Image, choisissez l'image Docker 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 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

Docker l'architecture de l'image doit correspondre à l'architecture du processeur des ressources de calcul sur lesquelles elles sont planifiées. Par exemple, les


Docker images Arm basées ne peuvent être exécutées que sur Arm des ressources informatiques basées.

- 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 la syntaxe des commandes, choisissez Bash ou JSON.
- c. Dans Commande, précisez la commande à transmettre au conteneur. Pour les commandes simples, entrez la commande comme vous le feriez pour une invite de commande, puis vérifiez que le JSON résultat est correct. Il est transmis au Docker daemon. Pour les commandes plus complexes (par exemple, avec des caractères spéciaux), utilisez la syntaxe JSON.

 Tip


Choisissez Info pour afficher Bash et JSON coder des exemples.

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 paramètre Docker `CMD`, veuillez consulter <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 plus d'informations, consultez [Paramètres](#).

- d. (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 :
- 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

- e. Dans la section Configuration de l'environnement :
- i. Pour la configuration du rôle Job, choisissez un rôle IAM autorisé à accéder aux AWS API. 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 les vCPU, entrez le nombre de vCPU 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 plus d'informations, consultez [Ressource de calcul Gestion de la mémoire](#).


- f. (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.
 - g. (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 [SecretOptions](#) dans. [Paramètres de définition du job pour ContainerProperties](#)
 - h. 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 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 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 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, consultez [LogDriver](#) dans [Paramètres de définition du job pour ContainerProperties](#).

 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 [SecretOptions](#) dans [Paramètres de définition du job pour ContainerProperties](#)


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

Pour créer une nouvelle définition de tâche sur les ressources Amazon Elastic Kubernetes Service, procédez comme suit :


1. Ouvrez la AWS Batch console à l'[adresse https://console.aws.amazon.com/batch/](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 de gauche, choisissez Job definitions.
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) 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 plus d'informations, consultez [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 unpod.
 - 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'Kubernetesenvironnement.

- Par défaut : pod hérite de la configuration de résolution de noms du nœud sur lequel il s'exécute.

 Note


Si aucune stratégie DNS n'est spécifiée, Default n'est pas la stratégie DNS par défaut. Au lieu de cela, ClusterFirstest 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 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/`.

- e. Choisissez Page suivante.
- f. Dans la section Configuration du conteneur :
- 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 (-).
 - 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

Dockerl'architecture de l'image doit correspondre à l'architecture du processeur des ressources de calcul sur lesquelles elles sont planifiées. Par exemple,

les Docker images Arm basées ne peuvent être exécutées que sur Arm des ressources informatiques basées.


- 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 Bash ou 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.
- g. (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

- h. Dans la section Configuration de l'environnement :

- i. Pour les vCPU, entrez le nombre de vCPU 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 plus d'informations, consultez [Ressource de calcul Gestion de la mémoire](#).

- i. (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.
- j. (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.
 - iii. Choisissez Lecture seule pour supprimer les autorisations d'écriture sur le volume.
 - iv. Choisissez Ajouter un montage de volume.
- k. (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.

- l. (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.

- m. (Facultatif) Pour accorder au conteneur de votre tâche 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](#).
- n. (Facultatif) Activez le système de fichiers racine en lecture seule pour supprimer l'accès en écriture au système de fichiers racine.
- o. (Facultatif) Activez Exécuter en tant qu'utilisateur non root t pour exécuter les conteneurs dans le 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.

- p. 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.

Création d'une définition de tâche parallèle sur plusieurs nœuds

Avant de pouvoir exécuter des tâches dans 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 nœuds multiples. Pour plus d'informations, consultez [Tâches parallèles sur plusieurs nœuds](#).

Note

AWSFargate ne prend pas en charge les tâches parallèles à nœuds multiples.

Création d'une définition de tâche parallèle à nœuds multiples sur les ressources Amazon EC2

Pour créer une définition de tâche parallèles à nœud unique, consultez [Création d'une définition de tâche à nœud unique](#).

Pour créer une définition de tâche parallèle à plusieurs nœuds sur les ressources Amazon Elastic Compute Cloud :

1. Ouvrez la AWS Batch console à l'[adresse https://console.aws.amazon.com/batch/](https://console.aws.amazon.com/batch/).
2. Dans la barre de navigation, sélectionnez la 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 plus d'informations, consultez [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.

ℹ 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 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

Docker l'architecture de l'image doit correspondre à l'architecture du processeur des ressources de calcul sur lesquelles elles sont planifiées. Par exemple, les Docker images Arm basées ne peuvent être exécutées que sur Arm des ressources informatiques basées.


- 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 la syntaxe des commandes, choisissez Bash ou JSON.
- e. Dans Commande, précisez la commande à transmettre au conteneur. Pour les commandes simples, vous pouvez saisir la commande comme vous le feriez à l'invite de commande dans l'onglet Espace délimité. Vérifiez ensuite que le JSON résultat est correct. Le résultat JSON est transmis au Docker daemon. Pour les commandes plus complexes (par exemple, celles comprenant des caractères spéciaux), vous pouvez ouvrir l'onglet JSON et y saisir l'équivalent du tableau de chaînes.

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 paramètre Docker CMD, veuillez consulter <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 plus d'informations, consultez [Paramètres](#).

- f. Dans `vCPUs`, précisez le nombre de vCPU 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.
- g. 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 plus d'informations, consultez [Ressource de calcul Gestion de la mémoire](#).

- h. (Facultatif) Dans `Nombre de GPU`, spécifiez le nombre de GPU utilisés par votre travail. La tâche s'exécute sur un conteneur avec le nombre spécifié de GPU épinglés à ce conteneur.
- i. (Facultatif) Pour le rôle `Job`, vous pouvez spécifier un rôle IAM qui autorise le conteneur de votre tâche à utiliser les AWS API. 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.

- j. (Facultatif) Pour le rôle d'exécution, spécifiez un rôle IAM qui autorise les agents de conteneur Amazon ECS à effectuer des appels d'AWSAPI 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 les AWS API. 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'rootutilisateur), 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 [SecretOptions](#) dans. [Paramètres de définition du job pour ContainerProperties](#)
21. Dans la section de configuration Linux :
- a. Activez l'option 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 [Swappiness in. Paramètres de définition du job pour ContainerProperties](#)
 - 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. (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, consultez [LogDriver](#) dans [Paramètres de définition du job pour ContainerProperties](#).

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 [SecretOptions](#) dans [Paramètres de définition du job pour ContainerProperties](#)

27. Choisissez Page suivante.
28. 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 de définitions de tâches à l'aide de 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 [Paramètres de définition du job pour ContainerProperties](#).

```
{
  "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
      }
    }
  ]
}
}
```

Note

Vous pouvez générer un modèle de définition de tâche à conteneur unique à l'aide de la commande suivante : AWS CLI

```
$ aws batch register-job-definition --generate-cli-skeleton
```

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
- le type de définition du 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 les tâches exécutées 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
- la priorité de planification par défaut pour la définition de la 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 du job](#)
- [Type](#)
- [Paramètres](#)
- [Propriétés du contenant](#)
- [Propriétés d'Amazon EKS](#)
- [Fonctionnalités de la plateforme](#)
- [Propager les tags](#)
- [Propriétés du nœud](#)

- [Réessayer la stratégie](#)
- [Priorité de planification](#)
- [Balises](#)
- [Timeout](#)

Nom de la définition du job

`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 : chaîne

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 [the section called "Création d'une définition de tâche parallèle sur plusieurs nœuds"](#).

Type : chaîne

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 du nœud](#), pour chaque groupe de nœuds.

command

La commande transmise au conteneur. 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 paramètre Docker CMD, 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 : chaîne

Obligatoire : oui lorsque environment est utilisé.

value

Valeur de la variable d'environnement.

Type : chaîne

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 plus d'informations, consultez [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 des ressources EC2 ne doivent pas spécifier ce paramètre.

Type : objet [FargatePlatformde configuration](#)

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 : chaîne

Par défaut: LATEST

Obligatoire : non

`image`

L'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

Docker l'architecture de l'image doit correspondre à l'architecture du processeur des ressources de calcul sur lesquelles elles sont planifiées. Par exemple, les Docker images Arm basées ne peuvent être exécutées que sur Arm des ressources informatiques basées.

- 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 : chaîne

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 exécutées sur les ressources Fargate.

Type : tableau d'objets [Périphérique](#)

Obligatoire : non

hostPath

Chemin où se trouve le périphérique disponible dans l'instance de conteneur hôte.

Type : chaîne

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, etMKNOD.

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 exécutées sur les 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 exécutées sur les 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

L'échange n'est pas activé par défaut pour les AMI optimisées 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 l'utilisateur Amazon EC2 ou [Comment allouer de la mémoire pour qu'elle fonctionne comme espace de swap dans une instance Amazon EC2 à l'aide d'un fichier d'échange ?](#).

- Les paramètres de l'espace d'échange sont uniquement pris en charge pour les définitions de tâches utilisant des ressources EC2.
- 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 exécutées sur les 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 exécutées sur les ressources Fargate.

Obligatoire : non

`containerPath`

Chemin d'accès absolu du fichier dans le conteneur où le volume `tmpfs` est monté.

Type : chaîne

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 "|atime" » | 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 pilotes `awslogs` 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'ECS_AVAILABLE_LOGGING_DRIVERS environnement. 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 Utilisation du pilote du 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 : chaîne

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 plus d'informations, consultez [Spécification des données sensibles](#).

Type : tableau d'objets

Obligatoire : non

name

Nom de l'option de pilote de journal à définir dans le job.

Type : chaîne

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 : chaîne

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 : chaîne

Obligatoire : oui lorsque `mountPoints` est utilisé.

containerPath

Le chemin sur le conteneur où monter le volume hôte.

Type : chaîne

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 des ressources EC2 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 : chaîne

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 exécutées sur les 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 : chaîne

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"

Nombre de GPU physiques à réserver pour le conteneur. Le nombre de GPU réservés à tous les conteneurs d'une tâche ne peut pas dépasser le nombre de GPU disponibles 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 [Ressource de calcul Gestion de la mémoire](#).

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

VCPU	MEMORY
1 vCPU	2048-8192 MiB par incréments de 1024 MiB
2 vCPU	4096-16384 MiB par incréments de 1024 MiB
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 vCPU réservés pour 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 des ressources EC2, 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 basé sur le nombre de ressources vCPU Fargate à la demande est de 6 vCPU. 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 : chaîne

Obligatoire : oui lorsque `resourceRequirements` est utilisé.

secrets

Les secrets du travail exposés sous forme de variables d'environnement. Pour plus d'informations, consultez [Spécification des 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 : chaîne

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 : chaîne

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 : chaîne

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.

Nombre de processeurs virtuels réservés pour le 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 exécutées sur les 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 plus d'informations, consultez [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 plus d'informations, consultez [Utilisation des points d'accès Amazon EFS](#).

Type : chaîne

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 : chaîne

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 : chaîne

Obligatoire : oui

`value`

Valeur de la variable d'environnement.

Type : chaîne

Obligatoire : non

`image`

Image Docker utilisée pour démarrer le conteneur.

Type : chaîne

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 [Ressource de calcul Gestion de la mémoire](#).

cpu

Nombre de processeurs réservés pour le 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

Nombre de GPU réservés pour le 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 [Ressource de calcul Gestion de la mémoire](#).

cpu

Nombre de processeurs réservés pour le 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

Nombre de GPU réservés pour le 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 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 Kubernetes documentation.

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 d'API de [RegisterJobdéfinition](#), aucune valeur n'est renvoyée `dnsPolicy` par les [DescribeJobdéfinitions](#) ou les 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 [Kubernetesdocumentation](#).

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 : chaîne

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 Kubernetesdocumentation.

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 : chaîne

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 : chaîne

Obligatoire : oui

Fonctionnalités 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 : chaîne

Valeurs valides : EC2 | FARGATE


Obligatoire : non

Propager les tags

`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 du 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 plus d'informations, consultez [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 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.

Type : Tableau d'objets de [NodeRangepropriété](#)

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 plus d'informations, consultez [Propriétés du contenant](#).

Type : objet [ContainerProperties](#)

Obligatoire : non

Réessayer la stratégie

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 plus d'informations, consultez [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 : chaîne

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 n'affecte que les tâches dans les files d'attente de tâches avec 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 plus d'informations, consultez [Balisage de vos ressources AWS Batch](#).

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 plus d'informations, consultez [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 plus d'informations, consultez [Expiration des délais d'exécution des tâches](#).

Type : objet [JobTimeout](#)

Obligatoire : non

`attemptDurationSeconds`

Durée en secondes (mesurée à partir de `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

Création de 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 aux AWS Batch API

Vous trouverez ci-dessous un aperçu de certaines des principales différences entre les types de données `EcsProperties` 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 `pidMode`. Pour plus d'informations, reportez-vous [EcsTaskDetails](#) aux sections 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` telle 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](#) le Guide de référence des AWS Batch API.
- [taskProperties](#) contient les propriétés définies pour la tâche Amazon ECS. Pour plus d'informations, consultez [EcsProperties](#) le Guide de référence des AWS Batch API.
- [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 [ECS:Containers TaskProperties dans le guide](#) de AWS Batch référence des API.
- [essential](#) le paramètre indique comment le conteneur affecte la tâche. Tous les conteneurs essentiels doivent être terminés avec succès (sortie sous la forme 0) pour que la tâche progresse. Si un conteneur marqué comme essentiel échoue (sort avec une valeur différente de 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 [ECSProperties](#).

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 être terminés avec succès 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](#).

AWS Batch scénarios de travail 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 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": true,
          "command": [
            "echo",
            "hello world"
          ],
          "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
          "resourceRequirements": [
            {
              "type": "VCPU",
              "value": "6"
            },
            {
              "type": "MEMORY",
              "value": "12288"
            }
          ]
        }
      ]
    }
  ]
}
```

```

    ]
  }
]
}
}

```

AWS Batch job pour Amazon ECS 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 offre d'emploi 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 multicode parallèle (MNP) AWS Batch 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"
      }
    ]
  }
]
}
}
```

Utilisation du pilote du 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 flux STDOUT et d'STDERR/S. 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 du journal `awslogs` disponibles

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 `strptime` 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 plus d'informations, consultez [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`.

Spécification d'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 plus d'informations, consultez [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 plus d'informations, consultez [Soumettre une offre d'emploi](#).

Spécification des 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

- [Spécification de données sensibles à l'aide de Secrets Manager](#)
- [Spécification de données sensibles à l'aide du Systems Manager Parameter Store](#)

Spécification de données sensibles à l'aide de 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 relatives à la spécification des données sensibles en utilisant 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. Cependant, 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'[CreateSecret](#)API, sont pris en charge. Les secrets qui stockent des données binaires, qui sont des secrets créés avec le SecretBinary paramètre de l'[CreateSecret](#)API, 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 plus d'informations, consultez [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 plus d'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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
```

```
    "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:<secret_name>",  
    "arn:aws:kms:<region>:<aws_account_id>:key/<key_id>"  
  ]  
}  
]  
}
```

Injection de données sensibles sous la forme d'une variable d'environnement

Dans votre définition de tâche, vous pouvez spécifier les éléments suivants :

- L'objet `secrets` 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.

Les ID de version sont utilisés 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 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"
    }]
  }]
}

```

Injection de données sensibles dans une configuration de journal

Dans votre définition de tâche, lorsque `logConfiguration` vous spécifiez un, vous pouvez spécifier `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"
      }]
    }]
  }]
}

```

Création d'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 : /_+=.@-

- (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](#).

- 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.

Spécification de données sensibles à l'aide du 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 relatives à la spécification de données sensibles à l'aide du Systems Manager Parameter Store](#)
- [Autorisations IAM requises pour les secrets AWS Batch](#)
- [Injection de données sensibles sous la forme d'une variable d'environnement](#)
- [Injection de données sensibles dans une configuration de journal](#)
- [Création d'un AWS Systems Manager paramètre Parameter Store](#)

Considérations relatives à la spécification de données sensibles à l'aide du Systems Manager Parameter Store

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. Cependant, 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 plus d'informations, consultez [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 plus d'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 :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetParameters",
```

```

    "secretsmanager:GetSecretValue",
    "kms:Decrypt"
  ],
  "Resource": [
    "arn:aws:ssm:<region>:<aws_account_id>:parameter/<parameter_name>",
    "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:<secret_name>",
    "arn:aws:kms:<region>:<aws_account_id>:key/<key_id>"
  ]
}
]
}

```

Injection de données sensibles sous la forme d'une 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"
    }]
  }]
}

```

Injection de 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 de magasin de paramètres de Systems Manager 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/](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 autres que ceux AWS qui nécessitent une authentification dans vos définitions de tâches. Cette fonctionnalité est prise en charge par les jobs hébergés sur les instances Amazon EC2 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 des instances Amazon EC2, cette fonctionnalité nécessite une version 1.19.0 ou une version ultérieure de l'agent de conteneur. Cependant, 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.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "kms:Decrypt",  
        "secretsmanager:GetSecretValue"  
      ],  
      "Resource": [  
        "arn:aws:secretsmanager:region:123456789012:secret:secret_name",  
        "arn:aws:kms:region:123456789012:key/key_id"  
      ]  
    }  
  ]  
}
```

Utilisation de l'authentification de registre privé

Pour créer un secret basique

AWS Secrets Manager À utiliser pour créer un secret pour vos informations d'identification de registre privées.

1. Ouvrez la AWS Secrets Manager console à l'[adresse https://console.aws.amazon.com/secretsmanager/](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 savoir comment 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 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 des ressources EC2, 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.

Utilisation des 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 AWS Batch tâche, 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.

Spécification d'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 : chaîne

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 : chaîne

Valeurs valides : ENABLED | DISABLED

Obligatoire : non

Détermine s'il faut activer le chiffrement pour les données Amazon EFS en transit entre l'AWS Batchhôte 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 Batchhôte 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 `EFSSVolumeConfiguration`. Pour plus d'informations, consultez [Utilisation des points d'accès Amazon EFS](#) dans le Guide de l'utilisateur Amazon Elastic File System.

iam

Type : chaîne

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 `EFSSVolumeConfiguration`. Si ce paramètre est omis, la valeur par défaut `DISABLED` est utilisée. Pour plus d'informations sur l'exécution des rôles IAM, consultez [AWS Batch rôle d'exécution IAM](#).

Exemples de définitions de tâches

L'exemple de définitions de tâches suivant illustre l'utilisation des modèles courants tels que les variables d'environnement, la substitution de paramètres et les montages de volume.

Utiliser des 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 est tiré du billet de blog [Creating a Simple "Fetch & Run" AWS Batch Job](#). 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://my-batch-scripts/myjob.sh"
      },
      {
        "name": "BATCH_FILE_TYPE",
        "value": "script"
      }
    ],
    "user": "nobody"
  }
}
```

```
}  
}
```

Utilisation de la 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 plus d'informations, consultez [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 [Utilisation d'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 AWSCompute](#).

```
{
  "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"
        }
      }
    ]
  }
}
```

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](#)
- [Paramètres de la file d'attente de tâches](#)
- [Affichage de l'état de la file d'attente](#)

Création d'une file d'attente de tâches


Avant de pouvoir soumettre des tâches dans 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.

Création d'une file d'attente de tâches 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/](https://console.aws.amazon.com/batch/).
2. Dans la barre de navigation, sélectionnez la 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 EC2

Pour créer une file d'attente de tâches Amazon EC2

1. Ouvrez la AWS Batch console à l'[adresse https://console.aws.amazon.com/batch/](https://console.aws.amazon.com/batch/).
2. Dans la barre de navigation, sélectionnez la 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 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

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/](https://console.aws.amazon.com/batch/).
2. Dans la barre de navigation, sélectionnez la 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 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.

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.

Modèle de file d'attente de tâches

Voici 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.

```
{
  "computeEnvironmentOrder": [
    {
      "computeEnvironment": "",
      "order": 0
    }
  ],
  "jobQueueName": "",
  "jobStateTimeLimitActions": [
    {
      "state": "RUNNABLE",
      "action": "CANCEL",
      "maxTimeSeconds": 0,
      "reason": ""
    }
  ],
  "priority": 0,
  "schedulingPolicyArn": "",
  "state": "ENABLED",
```

```
"tags": {  
  "KeyName": ""  
}  
}
```

Note

Vous pouvez générer le modèle de file d'attente de tâches précédent à l'aide de la AWS CLI commande suivante.

```
$ aws batch create-job-queue --generate-cli-skeleton
```

Paramètres de la file d'attente de tâches

Les files d'attente de tâches sont divisées en quatre éléments de base : nom, état, priorité et ordre de l'environnement de calcul. Cette section décrit les composants associés à ces composants.

Rubriques

- [Nom de la file d'attente de tâches](#)
- [État de la file d'attente de tâches, actions de limitation de durée](#)
- [Priorité](#)
- [Politique de planification](#)
- [État](#)
- [Ordre de l'environnement de calcul](#)
- [Balises](#)

Nom de la file d'attente de tâches

[jobQueueName](#)

Nom de votre file d'attente de tâches. Il peut comporter jusqu'à 128 lettres (majuscules et minuscules), des chiffres et des traits de soulignement.

Type : chaîne

Obligatoire : oui

État de la file d'attente de tâches, actions de limitation de durée

[jobStateTimeLimitActions](#)

Ensemble d'actions exécutées sur AWS Batch les tâches qui restent en tête de la file d'attente dans l'état spécifié plus longtemps que la durée spécifiée. AWS Batch exécutera chaque action `maxTimeSeconds` une fois passée. (Remarque : La valeur minimale pour `maxTimeSeconds` est 600 (10 minutes) et sa valeur maximale est 86 400 (24 heures).)

Type : tableau d'objets `JobStateTimeLimitActions`

Obligatoire : non

Priorité

[priority](#)

Priorité de la file d'attente de tâches. Les files d'attente de tâches avec une priorité élevée (ou une valeur d'entier supérieure pour le paramètre `priority`) sont évaluées en premier lorsqu'elles sont associées au même environnement de calcul. 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. Tous les environnements de calcul doivent être Amazon EC2 (EC2 ou SPOT) ou Fargate (ou). FARGATE FARGATE_SPOT Les environnements informatiques Amazon EC2 et Fargate ne peuvent pas être combinés.

Type : entier

Obligatoire : oui

Politique de planification

[schedulingPolicyArn](#)

Le nom de ressource Amazon (ARN) de la politique de planification pour la file d'attente des tâches. Les files d'attente de tâches qui ne sont pas soumises à une politique de planification sont planifiées selon un modèle « premier entré, premier sorti » (FIFO). Une fois qu'une file d'attente de tâches est dotée d'une politique de planification, elle peut être remplacée mais ne

peut pas être supprimée. Une file de tâches sans politique de planification est planifiée en tant que file d'attente de tâches FIFO et aucune politique de planification ne peut être ajoutée. Les files d'attente d'offres d'emploi soumises à une politique de planification peuvent comporter un maximum de 500 identifiants Fair Share actifs. Lorsque la limite est atteinte, les soumissions de toutes les offres d'emploi qui ajoutent un nouvel identifiant de partage équitable échouent.

Type : chaîne

Obligatoire : non

État

state

État de la file d'attente de tâches. Si l'état de la file d'attente des tâches est ENABLED (valeur par défaut), elle peut accepter des tâches. Si l'état de la file d'attente des travaux est DISABLED, les nouvelles tâches ne peuvent pas être ajoutées à la file d'attente, mais les tâches déjà dans la file d'attente peuvent se terminer.

Type : chaîne

Valeurs valides : ENABLED | DISABLED

Obligatoire : non

Ordre de l'environnement de calcul

computeEnvironmentOrder

Ensemble d'environnements de calcul mappés sur une file d'attente de tâches et leur ordre les uns par rapport aux autres. Le planificateur de tâches utilise ce paramètre pour déterminer l'environnement de calcul qui doit exécuter une tâche donnée. Les environnements de calcul doivent être dans l'état VALID pour que vous puissiez les associer à une file d'attente de tâches. Vous pouvez associer jusqu'à trois environnements de calcul à une file d'attente de tâches. Tous les environnements de calcul doivent être Amazon EC2 (EC2ouSP0T) ou Fargate (ou). FARGATE FARGATE_SP0T Les environnements informatiques Amazon EC2 et Fargate ne peuvent pas être combinés.

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.

Type : tableau d'objets [ComputeEnvironmentOrder](#)

Obligatoire : oui

`computeEnvironment`

L'Amazon Resource Name (ARN) de l'environnement de calcul.

Type : chaîne

Obligatoire : oui

`order`

Ordre de l'environnement de calcul. Les environnements de calcul sont essayés par ordre croissant. Par exemple, si deux environnements de calcul sont associés à une file d'attente de tâches, l'environnement de calcul avec une valeur d'entier `order` plus faible est essayé en premier pour le placement de tâche.

Balises

[tags](#)

Balises de paire clé-valeur à associer à la file d'attente des tâches. Pour plus d'informations, consultez [Balisage de vos ressources AWS Batch](#).

Type : mappage chaîne/chaîne

Obligatoire : non

Affichage de l'état de la file d'attente

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.

Affichage des 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.

Capture d'écran de la file d'emplois

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 [AWS Batch Fair Share Scheduling \(FSS\)](#), l'ordre des tâches est basé sur la priorité des tâches et l'utilisation du partage.

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.

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.

Balises

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.

Planification des tâches

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 plus d'informations, consultez [Politiques de planification](#).

Partager les identifiants

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 équitable en utilisant la $(T * weightFactor)$ formule, où se T situe 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 équitable 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 la priorité de planification 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, 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.

Note

Les ressources informatiques associées sont réparties de manière égale entre tous les identifiants de partage, sauf dérogation explicite.

Planification du partage équitable

La planification du partage équitable fournit un ensemble de contrôles pour aider à planifier les tâches.

Note

Pour plus d'informations sur les paramètres des politiques de planification, consultez [Paramètres de la politique de planification](#).

- **Secondes de déclin des parts** : période (en secondes) utilisée par le AWS Batch planificateur pour calculer un pourcentage de part équitable pour chaque identifiant de partage équitable. 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 fair share 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 plus d'informations, consultez la section [Affichage de l'état de la file d'attente des tâches](#).

Environnement de calcul

Les files d'attente de tâches sont mappées sur un ou plusieurs environnements de calcul. 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](#)
- [AMI de ressources de calcul](#)
- [Support de modèle de lancement](#)
- [Création d'un environnement informatique](#)
- [Modèle d'environnement de calcul](#)
- [Paramètres de l'environnement de calcul](#)
- [Configurations EC2](#)
- [Stratégies d'allocation](#)
- [Mise à jour des environnements informatiques](#)
- [Environnements informatiques Amazon EKS](#)
- [Ressource de calcul Gestion de la mémoire](#)

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 tâche Windows est soumise à une file d'attente de tâches qui utilise uniquement les environnements informatiques Fargate Spot.

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 [Création d'un cloud privé virtuel](#)

Par défaut, les environnements de calcul 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 plus d'informations, consultez [AMI de ressources de calcul](#).

 Note

AWS Batch ne met pas automatiquement à jour les AMI dans un environnement informatique une fois celui-ci créé. Par exemple, il ne met pas à jour les AMI de votre environnement informatique lorsqu'une nouvelle version de l'AMI optimisée pour 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 plus d'informations, consultez [Mise à jour des environnements informatiques](#). Pour utiliser la mise à jour améliorée des environnements de calcul afin de mettre à jour les AMI, suivez les règles ci-dessous :

- 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 entraînera 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 de calcul non géré, vous gérez vos propres ressources de calcul. Vous devez vérifier que l'AMI que vous utilisez pour vos ressources de calcul est conforme à la spécification AMI de l'instance de conteneur Amazon ECS. Pour plus d'informations, consultez [Spécification de l'AMI des ressources de calcul](#) et [Création d'une AMI de ressources de calcul](#).

Note

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

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
```

AMI de ressources de calcul

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'instances Amazon EC2 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'instances Amazon EC2 pris en charge

Note

Une fois qu'un environnement de calcul est créé, les AMI AWS Batch ne sont pas mises à niveau dans l'environnement de calcul. AWS Batch ne met pas non plus à jour les AMI de votre environnement informatique lorsqu'une version plus récente de l'AMI optimisée pour 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 plus d'informations, consultez [Mise à jour des environnements informatiques](#). Pour utiliser la mise à jour améliorée des environnements de calcul afin de mettre à jour les AMI, suivez les règles ci-dessous :

- 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_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 AWS Batch 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 de l'AMI des ressources de calcul](#)
- [Création d'une AMI de ressources de calcul](#)
- [Utilisation d'une AMI de charge de travail du GPU](#)
- [Obsolète d'Amazon Linux](#)

Spécification de l'AMI des 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 plus d'informations, consultez [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 nœud 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'est pas encore disponible dans ces AMI. Pour plus d'informations, consultez [l'AMI optimisée pour Amazon ECS](#) dans le manuel Amazon Elastic Container Service Developer Guide.

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 de l'AMI des 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 souhaitez utiliser pour cet environnement de calcul. Par exemple, si votre environnement informatique utilise des types d'instances Arm, l'AMI de ressources de calcul que vous choisissez doit prendre en charge les instances Arm. Amazon ECS vend à la fois des versions x86 et des versions 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 instance Amazon Linux 2 EC2](#) 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 plus d'informations, consultez [Utilisation d'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 Store](#) et [Amazon EC2 Instance Store](#) dans le guide de l'utilisateur Amazon EC2.
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'utilisateur Amazon EC2.

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 l'utilisateur Amazon EC2.

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 l'ID d'image boîte de remplacement lorsque vous créez l'environnement de AWS Batch calcul. Pour plus d'informations, voir [the section called "Pour créer un environnement informatique géré à l'aide des ressources EC2"](#).

Note

L'AMI que vous choisissez pour un environnement de calcul doit correspondre à l'architecture des types d'instances que vous souhaitez utiliser pour cet environnement de calcul. Par exemple, si votre environnement informatique utilise des types d'A1 instances, l'AMI de ressources de calcul que vous choisissez doit prendre en charge les instances Arm. Amazon ECS vend à la fois des versions x86 et des versions 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 plus d'informations, consultez [Création d'une 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.

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 [Soumettre une offre d'emploi](#).

Utilisation d'une AMI de charge de travail du GPU

Pour exécuter des charges de travail GPU sur vos ressources de calcul AWS Batch, vous devez utiliser une AMI avec prise en charge GPU. Pour plus d'informations, consultez la section [Utilisation des GPU sur Amazon ECS et des AMI optimisées pour Amazon ECS dans le manuel du développeur Amazon Elastic Container Service](#).

Dans les environnements informatiques gérés, si l'environnement de calcul spécifie des p2 types d'instances ou des familles d'instances, il AWS Batch utilise une AMI optimisée pour le GPU Amazon ECS. p3 p4 p5 g3 g3s g4

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 des AMI optimisées pour le GPU Amazon ECS recommandées.

Note

La famille d'instances p5 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'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](#) dans les ComputeResource propriétés. Pour plus d'informations sur la compatibilité des AMI avec les instances de GPU, consultez la section [Utilisation des GPU 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-eb\", \"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-eks",
        "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-eks  
/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](#).

Support de modèle de lancement

AWS Batch prend en charge l'utilisation des modèles de lancement Amazon EC2 avec vos environnements informatiques EC2. Grâce aux modèles de lancement, vous pouvez modifier la configuration par défaut de vos ressources AWS Batch informatiques sans avoir à créer des AMI personnalisées.

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
- ID de groupe de sécurité
- 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 plus d'informations, consultez [Mise à jour des environnements informatiques](#).

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 pouvez associer un boothook cloud qui configure le démon Docker à un script shell de données utilisateur qui écrit des 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.

Vous trouverez ci-dessous des exemples de fichiers MIME en plusieurs parties que vous pouvez utiliser pour créer les vôtres.

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":
      ""ewogICAgIkxhdW5jaFRlbXBsYXRlTmFtZSI6ICJpbmNyZWZzZS1jb250YWluZXItZm9udW50dW...""
  }
}
```

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 depuis la bibliothèque Extras et y monter un système de fichiers FSx for Lustre existant avec le nom de montage `/scratch` 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 for Lustre. Pour plus d'informations, consultez la section [Montage automatique de votre système de fichiers Amazon FSx](#) dans le guide de l'utilisateur d'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"
  }
],
}
```

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 :

- Sur les conteneurs Amazon Linux dotés d'une architecture ARM64.
- 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.

Table des matières

- [Pour créer un environnement informatique géré à l'aide des ressources AWS Fargate](#)
- [Pour créer un environnement informatique géré à l'aide des ressources EC2](#)
- [Pour créer un environnement informatique non géré à l'aide des ressources EC2](#)
- [Pour créer un environnement informatique géré à l'aide des ressources Amazon EKS](#)

Pour créer un environnement informatique géré à l'aide des ressources AWS Fargate


1. Ouvrez la AWS Batch console à l'[adresse https://console.aws.amazon.com/batch/](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 plus d'informations, consultez [Autorisations de rôle liées à un service pour AWS Batch](#).
 - d. (Facultatif) Développez les balises. Pour ajouter une identification, choisissez Ajouter une identification. 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'[Amazon EC2](#) Spot et de Fargate_spot.
 - b. Pour un maximum de vCPU, choisissez le nombre maximal de vCPU que votre environnement informatique peut atteindre, quelle que soit la demande de files 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 plus d'informations, consultez [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.


Pour créer un environnement informatique géré à l'aide des ressources EC2

1. Ouvrez la AWS Batch console à l'[adresse https://console.aws.amazon.com/batch/](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.
 - 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. (Facultatif) 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 plus d'informations, consultez [Autorisations de rôle liées 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 plus d'informations, consultez [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.
 - g. (Facultatif) Pour les balises EC2, 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.
 - h. (Facultatif) Pour Tags, choisissez Ajouter un tag. Entrez ensuite un nom de clé et une valeur facultative. Choisissez Ajouter une balise.

Pour plus d'informations, consultez [Balisage de vos ressources AWS Batch](#).

- i. Choisissez Page suivante.
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 instance EC2. 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 IAM Amazon EC2 Spot Fleet existant à appliquer à votre environnement de calcul Spot. Si vous ne possédez pas encore de rôle IAM Amazon EC2 Spot Fleet, vous devez d'abord en créer un. Pour plus d'informations, consultez [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 gérée par Amazon SpotFleet TaggingRole EC2. La politique de gestion des SpotFleet rôles AmazonEC2 ne dispose pas des autorisations requises pour baliser les instances Spot. Pour plus d'informations, consultez [Instances ponctuelles non étiquetées lors de la création et the section called "Balisage de vos ressources"](#).

- d. Pour le nombre minimal de vCPU, choisissez le nombre minimal de vCPU que votre environnement informatique gère, quelle que soit la demande de la file d'attente des tâches.
- e. Pour les vCPU souhaités, choisissez le nombre de vCPU avec lesquels votre environnement informatique démarre. Au fur et à mesure que la demande de votre file d'attente de tâches augmente, AWS Batch peut augmenter le nombre souhaité de processeurs virtuels dans votre environnement de calcul et ajouter des instances EC2, jusqu'au nombre maximal de processeurs virtuels. Au fur et à mesure que la demande diminue, AWS Batch peut réduire le nombre souhaité de processeurs virtuels dans votre environnement de calcul et supprimer des instances, jusqu'au nombre minimal de processeurs virtuels.
- f. Pour un maximum de vCPU, choisissez le nombre maximal de vCPU que votre environnement informatique peut atteindre, quelle que soit la demande de files d'attente des tâches.


- g. Pour les types d'instances autorisés, choisissez les types d'instances Amazon EC2 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`. Vous pouvez également choisir de `optimal` sélectionner des types d'instances (parmi les familles d'instances `C4M4`, et) qui répondent à la demande de vos files d'attente de travail.

 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 adaptera les GPU 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 familles `p2 p3 p4p5,g3,g3s,g4`, ou.


 Note

`optimal` utilise actuellement des types d'instances issus des familles d'instances `C4M4`, et. Si Régions AWS aucun type d'instance n'est issu de ces familles d'instances, les types d'instance des familles `C5M5`, et d'instances `R5` sont utilisés.

- h. 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.
- j. (Facultatif) Pour la paire de clés EC2, 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 clés Amazon EC2, consultez la section [Paires de clés Amazon EC2](#) et instances Linux.

- k. 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 EC2 On-Demand, SPOT_CAPACITY_OPTIMIZED et SPOT_PRICE_CAPACITY_OPTIMIZED pour les environnements de calcul EC2 Spot. Pour plus d'informations, consultez [the section called "Stratégies d'allocation"](#).
- l. (Facultatif) Pour la configuration EC2, 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 (AMI) pour les instances de 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

Peut être utilisé pour les familles d'instances autres que le GPU ou AWS Graviton. Le support standard pour Amazon Linux AMI a pris fin. Pour plus d'informations, consultez [AMI Amazon Linux](#).

 Note


L'AMI que vous choisissez pour un environnement de calcul doit correspondre à l'architecture des types d'instances que vous souhaitez utiliser pour cet environnement de calcul. Par exemple, si votre environnement informatique utilise des types d'instances A1, l'AMI de ressources de calcul que vous choisissez doit prendre en charge les instances Arm. Amazon ECS vend à la fois des versions x86 et des versions 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.

- m. (Facultatif) Pour le modèle de lancement, sélectionnez un modèle de lancement Amazon EC2 existant pour configurer vos ressources de calcul. La version par défaut du modèle est automatiquement renseignée. Pour plus d'informations, consultez [Support de modèle de lancement](#).

 Note

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

- n. (Facultatif) Pour Version du modèle de lancement, saisissez `$Default`, `$Latest` ou un numéro de version spécifique à utiliser.

 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 plus d'informations, consultez [the section called "Mise à jour de l'ID AMI"](#).

- o. Choisissez Page suivante.

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 plus d'informations, consultez [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.

ℹ Note

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

- 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.
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 Créer un environnement de calcul.


Pour créer un environnement informatique non géré à l'aide des ressources EC2

1. Ouvrez la AWS Batch console à l'[adresse https://console.aws.amazon.com/batch/](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. (Facultatif) Pour le rôle de service, choisissez un rôle qui permet au AWS Batch service d'appeler les opérations d' AWS API requises en votre nom. Par exemple, sélectionnez AWSBatchServiceRole. Pour plus d'informations, consultez [the section called "Utilisation des rôles liés à un service"](#).
7. Pour un maximum de vCPU, choisissez le nombre maximal de vCPU que votre environnement informatique peut atteindre, quelle que soit la demande de files d'attente des tâches.
8. (Facultatif) Développez les balises. Pour ajouter une identification, choisissez Ajouter une identification. Entrez ensuite un nom de clé et une valeur facultative. Choisissez Ajouter une balise. Pour plus d'informations, consultez [Balisage de vos ressources AWS Batch](#).
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.

Pour créer un environnement informatique géré à l'aide des ressources Amazon EKS

1. Ouvrez la AWS Batch console à l'[adresse https://console.aws.amazon.com/batch/](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 les instances Spot EC2, activez Activer l'utilisation d'instances Spot pour utiliser les instances Spot Amazon EC2.
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 instance EC2. 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 Spot Amazon EC2 pour l'SPOTenvironnement informatique.

⚠ Important

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

15. (Facultatif) Pour le nombre minimal de vCPU, choisissez le nombre minimal de vCPU gérés par votre environnement informatique, quelle que soit la demande de files d'attente des tâches.
16. (Facultatif) Pour le nombre maximal de vCPU, choisissez le nombre maximal de vCPU 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'instances Amazon EC2 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`. Vous pouvez également choisir de sélectionner des types d'instances (parmi les familles d'instances `C4M4`, et) car vous devez répondre à la demande de vos files d'attente de travail.

📘 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 adapte les GPU 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 familles `g5 p2 p3 p4p5,g3,g3s,g4`, ou.

Note

Actuellement, `optimal` utilise les types d'instances des familles d'instances C4, M4 et R4. Si Régions AWS aucun type d'instance n'est issu de ces familles d'instances, les types d'instance des familles C5M5, et d'R5instances sont utilisés.

18. (Facultatif) Développez la configuration supplémentaire.
 - a. (Facultatif) Pour Groupe de placement, entrez un nom de groupe de placement pour regrouper les ressources dans l'environnement informatique.
 - b. Pour la stratégie d'allocation, choisissez `BEST_FIT_PROGRESSIVE`.
 - c. (Facultatif) Pour la configuration Amazon Machine Images (AMI), choisissez Ajouter la configuration Amazon Machine Images (`amis`). Choisissez ensuite un type d'image, entrez un remplacement d'ID d'image et une Kubernetesversion.

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.

Note

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.

[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.

- d. (Facultatif) Pour Modèle de lancement, choisissez un modèle de lancement existant.
 - e. (Facultatif) Pour la version du modèle Launch**\$Default**, entrez**\$Latest**, ou un numéro de version.
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 Créer un environnement de calcul.

Modèle d'environnement de calcul

L'exemple suivant montre un modèle d'environnement de calcul vide. Vous pouvez l'utiliser pour créer un environnement de calcul, que vous pouvez ensuite enregistrer dans un fichier et utiliser avec l'option AWS CLI de `--cli-input-json`. Pour plus d'informations sur ces paramètres, consultez [CreateComputeEnvironment](#) la référence de l'AWS BatchAPI.

```
{
  "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": ""
  }
}
```

Note

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

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

Paramètres de l'environnement de calcul

Les environnements informatiques sont divisés en plusieurs composants de base : le nom, le type et l'état de l'environnement informatique, la définition des ressources de calcul (s'il s'agit d'un environnement informatique géré), la configuration Amazon EKS (s'il utilise les ressources Amazon EKS), le rôle de service à utiliser pour fournir des autorisations IAM et les balises de l'environnement de calcul. AWS Batch

Rubriques

- [Nom de l'environnement de calcul](#)
- [Type](#)
- [État](#)
- [Ressources informatiques](#)
- [Configuration d'Amazon EKS](#)
- [Rôle de service](#)
- [Balises](#)

Nom de l'environnement de calcul

`computeEnvironmentName`

Nom de 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 (_).

Type : chaîne

Obligatoire : oui

Type

`type`

Type d'environnement de calcul. Choisissez MANAGED de AWS Batch gérer les ressources de calcul EC2 ou Fargate que vous définissez. Pour plus d'informations, consultez [Ressources informatiques](#). Choisissez UNMANAGED de gérer vos propres ressources de calcul EC2.

Type : chaîne

Valeurs valides : MANAGED | UNMANAGED

Obligatoire : oui

État


`state`

État de l'environnement de calcul.

Si c'est le casENABLED, le AWS Batch planificateur tente de placer les tâches dans l'environnement. Ces tâches proviennent d'une file d'attente de tâches associée sur les ressources informatiques. Si l'environnement informatique est géré, les instances augmentent ou augmentent automatiquement en fonction de la demande de la file d'attente des tâches.

Si c'est le casDISABLED, le AWS Batch planificateur ne tente pas de placer les tâches dans l'environnement. Les emplois en cours STARTING d'RUNNINGexécution continuent de progresser

normalement. Les environnements informatiques gérés qui sont dans DISABLED cet état ne sont pas évolutifs.

 Note

Les environnements informatiques d'un DISABLED État peuvent continuer à entraîner des frais de facturation. Pour éviter des frais supplémentaires, désactivez puis supprimez l'environnement informatique. Pour plus d'informations, consultez le Guide [DeleteComputeEnvironment](#) de référence des AWS Batch API et [Éviter les frais imprévus](#) dans le Guide de AWS Billing l'utilisateur.

Lorsqu'une instance est inactive, elle est réduite à sa `minvCpus` valeur. Cependant, la taille de l'instance ne change pas. Par exemple, considérez une `c5.8xlarge` instance avec une `minvCpus` valeur de 4 et une `desiredvCpus` valeur de 36. Cette instance ne se réduit pas à une `c5.large` instance.

Type : chaîne

Valeurs valides : ENABLED | DISABLED

Obligatoire : non

Ressources informatiques

`computeResources`

Détails des ressources de calcul gérées par l'environnement de calcul. Pour plus d'informations, consultez [Environnement de calcul](#).

Type : objet [ComputeResource](#)

Obligatoire : ce paramètre est obligatoire pour les environnements informatiques gérés

`type`

Type d'environnement de calcul. Vous pouvez choisir d'utiliser des instances EC2 On-Demand (EC2) et des instances Spot EC2 (SPOT), ou d'utiliser la capacité Fargate () FARGATE et la capacité Fargate Spot () dans votre environnement informatique géré. FARGATE_SPOT Si

vous choisissez SPOT, vous devez également spécifier un rôle pour le parc Spot Amazon EC2 avec le paramètre `spotIamFleetRole`. Pour plus d'informations, consultez [Rôle de la flotte Amazon EC2 Spot](#).

Valeurs valides : EC2 | SPOT | FARGATE | FARGATE_SPOT

Obligatoire : oui

allocationStrategy

La stratégie d'allocation à utiliser pour la ressource de calcul si un nombre insuffisant d'instances du type d'instance EC2 le mieux adapté ne peut être allouée. Cela peut être dû à la disponibilité du type d'instance dans les limites du service Amazon EC2 Région AWS ou à celles du [service Amazon EC2](#). Pour plus d'informations, consultez [Stratégies d'allocation](#).

Note

Ce paramètre ne s'applique pas aux tâches exécutées sur les ressources Fargate.

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 vous atteignez les [limites du service Amazon EC2](#), les tâches supplémentaires ne sont exécutées qu'une fois les tâches en cours d'exécution 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. Les ressources de calcul qui utilisent une stratégie d'allocation ne prennent pas en charge les mises à jour de l'infrastructure et ne peuvent pas mettre à jour certains paramètres. Pour plus d'informations, consultez [Mise à jour des environnements informatiques](#).

Note

BEST_FIT n'est pas pris en charge pour les environnements informatiques utilisant les ressources Amazon EKS.

BEST_FIT_PROGRESSIVE

Utilisez des types d'instances supplémentaires suffisamment grands pour répondre aux exigences des tâches de la file d'attente. Préférez les types d'instance dont le coût par unité de vCPU est inférieur. Si des instances supplémentaires des types d'instance précédemment sélectionnés ne sont pas disponibles, AWS Batch sélectionne de nouveaux types d'instance.

SPOT_CAPACITY_OPTIMIZED

(Disponible uniquement pour les ressources de calcul des instances Spot) Utilisez des types d'instances supplémentaires suffisamment grands pour répondre aux exigences des tâches de la file d'attente. Préférez les types d'instance moins susceptibles d'être interrompus.

SPOT_PRICE_CAPACITY_OPTIMIZED

(Disponible uniquement pour les ressources de calcul des instances ponctuelles) La stratégie d'allocation optimisée en termes de prix et de capacité tient compte à la fois du prix et de la capacité afin de sélectionner les pools d'instances ponctuels les moins susceptibles d'être interrompus et dont le prix est le plus bas possible.

Note

Nous vous recommandons d'utiliser SPOT_PRICE_CAPACITY_OPTIMIZED plutôt que SPOT_CAPACITY_OPTIMIZED dans la plupart des cas.

Avec BEST_FIT_PROGRESSIVE, SPOT_CAPACITY_OPTIMIZED, et les SPOT_PRICE_CAPACITY_OPTIMIZED stratégies utilisant des instances à la demande ou ponctuelles et la BEST_FIT stratégie utilisant des instances ponctuelles, il est possible que le nombre de maxvCpus dépasse vos exigences en matière de capacité. Dans ce cas, le nombre de maxvCpus ne dépasse jamais plus d'une instance.

Valeurs valides : BEST_FIT | BEST_FIT_PROGRESSIVE | SPOT_CAPACITY_OPTIMIZED | SPOT_PRICE_CAPACITY_OPTIMIZED

Obligatoire : non

minvCpus

Le nombre minimal de vCPU qu'un environnement gère, même si c'est le cas d'un environnement informatique. DISABLED

Note

Ce paramètre ne s'applique pas aux tâches exécutées sur les ressources Fargate.

Type : entier

Obligatoire : non

maxvCpus

Nombre maximal de vCPU que l'environnement de AWS Batch calcul peut prendre en charge.

Note

Les stratégies d'`BEST_FIT_PROGRESSIVESPOT_PRICE_CAPACITY_OPTIMIZED` utilisant des instances à la demande ou ponctuelles et la `BEST_FIT` stratégie utilisant des instances ponctuelles AWS Batch peuvent devoir être dépassées `maxvCpus` pour répondre à vos besoins de capacité. `SPOT_CAPACITY_OPTIMIZED` Dans ce cas, AWS Batch ne `maxvCpus` dépassez jamais plus d'une instance. Par exemple, n' AWS Batch utilise qu'une seule instance parmi celles spécifiées dans votre environnement informatique.

Type : entier

Obligatoire : non

desiredvCpus

Nombre de vCPU souhaités dans l'environnement informatique. AWS Batch modifie cette valeur entre les valeurs minimale et maximale en fonction de la demande de la file d'attente des tâches.

Note

Ce paramètre ne s'applique pas aux tâches exécutées sur les ressources Fargate.

Type : entier

Obligatoire : non

instanceTypes

Les types d'instances qui peuvent être lancés. Ce paramètre ne s'applique pas aux tâches exécutées sur les ressources Fargate. Ne le spécifiez pas. 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`). Notez que les types d'instances de métal ne figurent pas dans les familles d'instances (par exemple, `c5` ils ne sont pas inclus) `c5.metal`. Vous pouvez également choisir `optimal` afin de sélectionner des types d'instances (parmi les familles d'instances C4, M4 et R4) qui correspondent à la demande de vos files d'attente de tâches.

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

Actuellement, `optimal` utilise les types d'instances des familles d'instances C4, M4 et R4. Si Régions AWS aucun type d'instance ne provient de ces familles d'instances, les types d'instance des familles d'instances C5, M5 et R5 sont utilisés.

Type : tableau de chaînes

Obligatoire : oui

imageId

Ce paramètre est obsolète.

ID d'Amazon Machine Image (AMI) utilisé pour les instances lancées dans l'environnement de calcul. Ce paramètre est remplacé par le membre `imageIdOverride` de la structure `Ec2Configuration`.

Note

Ce paramètre ne s'applique pas aux tâches exécutées sur les ressources Fargate.

Note

L'AMI que vous choisissez pour un environnement de calcul doit correspondre à l'architecture des types d'instances que vous souhaitez utiliser pour cet environnement de calcul. Par exemple, si votre environnement informatique utilise des types d'instances A1, l'AMI de ressources de calcul que vous choisissez doit prendre en charge les instances Arm. Amazon ECS vend à la fois des versions x86 et des versions 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.

Type : chaîne

Obligatoire : non

subnets

Sous-réseaux VPC dans lesquels les ressources de calcul sont lancées. Ces sous-réseaux doivent se trouver dans le même VPC. Les ressources de calcul de Fargate peuvent contenir un maximum de 16 sous-réseaux. Pour plus d'informations, consultez [VPC et sous-réseaux](#) dans le Guide de l'utilisateur Amazon VPC.

Note

AWS Batch sur Amazon EC2 et sur AWS Batch Amazon EKS supportent les Zones Locales. Pour plus d'informations, consultez les [sections Zones locales](#) dans le guide

de l'utilisateur Amazon EC2, [Amazon EKS et zones AWS locales](#) dans le guide de l'utilisateur Amazon EKS et les [clusters Amazon ECS dans les zones locales, les zones de longueur d'onde et AWS Outposts](#) dans le guide du développeur Amazon Elastic Container Service.

AWS Batch sur Fargate ne prend actuellement pas en charge les Zones Locales.

Lors de la mise à jour des environnements de calcul, si vous fournissez une liste vide de sous-réseaux VPC, le comportement qui en résulte est différent entre les ressources de calcul Fargate et EC2. Pour les ressources de calcul Fargate, le fait de fournir une liste vide est traité comme si ce paramètre n'était pas spécifié et qu'aucune modification n'était apportée. Pour les ressources de calcul EC2, le fait de fournir une liste vide supprime les sous-réseaux VPC de la ressource de calcul. Si vous modifiez les sous-réseaux VPC, une mise à jour de l'infrastructure de l'environnement de calcul est requise. C'est le cas pour les ressources de calcul Fargate et EC2. Pour plus d'informations, consultez [Mise à jour des environnements informatiques](#).

Type : tableau de chaînes

Obligatoire : oui

`securityGroupIds`

Groupe de sécurité Amazon EC2 associé avec les instances lancées dans l'environnement de calcul. Un ou plusieurs groupes de sécurité doivent être spécifiés, dans `securityGroupIds` ou à l'aide d'un modèle de lancement référencé dans `launchTemplate`. Ce paramètre est obligatoire pour les tâches exécutées sur les ressources Fargate et doit contenir au moins un groupe de sécurité. (Fargate ne prend pas en charge les modèles de lancement.) Si les groupes de sécurité sont spécifiés à l'aide de `securityGroupIds` et `launchTemplate`, les valeurs de `securityGroupIds` seront utilisées.


Lors de la mise à jour des environnements informatiques, si vous fournissez une liste vide de groupes de sécurité, le comportement qui en résulte diffère entre les ressources de calcul Fargate et EC2. Pour les ressources de calcul Fargate, le fait de fournir une liste vide est traité comme si ce paramètre n'était pas spécifié et qu'aucune modification n'était apportée. Pour les ressources de calcul EC2, le fait de fournir une liste vide supprime les groupes de sécurité de la ressource de calcul. Si vous modifiez les groupes de sécurité, une mise à jour de l'infrastructure de l'environnement informatique est requise. C'est le cas pour les ressources de calcul Fargate et EC2. Pour plus d'informations, consultez [Mise à jour des environnements informatiques](#).

Type : tableau de chaînes

Obligatoire : oui

`ec2KeyPair`

La paire de clés EC2 utilisée pour les instances lancées dans l'environnement informatique. Vous pouvez utiliser cette paire de clés afin de vous connecter à vos instances avec SSH. Lors de la mise à jour d'un environnement informatique, si vous modifiez la paire de clés EC2, une mise à jour de l'infrastructure de l'environnement informatique est requise. Pour plus d'informations, consultez [Mise à jour des environnements informatiques](#).

 Note

Ce paramètre ne s'applique pas aux tâches exécutées sur les ressources Fargate.

Type : chaîne

Obligatoire : non

`instanceRole`

Le profil d'instance Amazon ECS à associer aux instances Amazon EC2 dans un environnement informatique. Ce paramètre ne s'applique pas aux tâches exécutées sur les ressources Fargate. Ne le spécifiez pas. Vous pouvez spécifier le nom court ou l'Amazon Resource Name (ARN) complet d'un profil d'instance. Par exemple, `ecsInstanceRole` ou `arn:aws:iam::aws_account_id:instance-profile/ecsInstanceRole`. Pour plus d'informations, consultez [Rôle d'instance Amazon ECS](#).

Lors de la mise à jour d'un environnement informatique, si vous modifiez ce paramètre, une mise à jour de l'infrastructure de l'environnement informatique est requise. Pour plus d'informations, consultez [Mise à jour des environnements informatiques](#).

Type : chaîne


Obligatoire : non

`tags`

Balises de paire clé-valeur à appliquer aux instances EC2 lancées dans l'environnement informatique. Par exemple, vous pouvez spécifier `"Name": "AWS Batch Instance - C4OnDemand"` comme balise de manière à ce que chaque instance dans votre environnement

de calcul prene ce nom. Cela est utile pour reconnaître vos AWS Batch instances dans la console Amazon EC2. Ces balises ne sont pas visibles lors de l'utilisation de l'opération AWS Batch [ListTagsForResourceAPI](#).

Lors de la mise à jour d'un environnement informatique, si vous modifiez les balises EC2, une mise à jour de l'infrastructure de l'environnement de calcul est requise. Pour plus d'informations, consultez [Mise à jour des environnements informatiques](#).

 Note


Ce paramètre ne s'applique pas aux tâches exécutées sur les ressources Fargate.

Type : mappage chaîne/chaîne

Obligatoire : non

placementGroup

Groupe de placement Amazon EC2 à associer à vos ressources de calcul. Ce paramètre ne s'applique pas aux tâches exécutées sur les ressources Fargate. Ne le spécifiez pas. Si vous avez l'intention de soumettre des tâches parallèles à nœuds multiples à votre environnement informatique, pensez à créer un groupe de placement de clusters et à l'associer à vos ressources de calcul. Ainsi, votre tâche parallèle à plusieurs nœuds reste sur un groupement logique d'instances au sein d'une zone de disponibilité unique à flux potentiel de réseau élevé. Pour plus d'informations, consultez [Groupes de placement](#) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

 Note

Ce paramètre ne s'applique pas aux tâches exécutées sur les ressources Fargate.

Type : chaîne


Obligatoire : non

bidPercentage

Pourcentage maximum que le prix d'une instance Spot EC2 peut être comparé au prix à la demande pour ce type d'instance avant le lancement des instances. Par exemple, si votre

pourcentage maximum est de 20 %, le prix spot doit être inférieur à 20 % du prix à la demande actuel pour cette instance EC2. 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. Dans la plupart des cas, nous recommandons de laisser ce champ vide.

Lors de la mise à jour d'un environnement informatique, si vous modifiez le pourcentage d'enchère, une mise à jour de l'infrastructure de l'environnement informatique est requise. Pour plus d'informations, consultez [Mise à jour des environnements informatiques](#).


 Note

Ce paramètre ne s'applique pas aux tâches exécutées sur les ressources Fargate.


Obligatoire : non

`spotIamFleetRole`

L'Amazon Resource Name (ARN) du rôle IAM de parc d'instances Spot Amazon EC2 appliqué à un environnement de calcul SPOT. Ce rôle est requis si la stratégie d'allocation est définie sur BEST_FIT, ou si elle n'est pas spécifiée. Pour plus d'informations, consultez [Rôle de la flotte Amazon EC2 Spot](#).

 Note

Ce paramètre ne s'applique pas aux tâches exécutées sur les ressources Fargate.

 Important

Pour étiqueter vos instances Spot lors de leur création, le rôle Spot Fleet IAM spécifié ici doit utiliser la nouvelle politique gérée par AmazonEC2 SpotFleet TaggingRole. La politique de gestion des SpotFleetrôles AmazonEC2 précédemment recommandée ne dispose pas des autorisations requises pour baliser les instances Spot. Pour plus d'informations, consultez [Instances ponctuelles non étiquetées lors de la création](#).

Type : chaîne

Obligatoire : ce paramètre est obligatoire pour les environnements de calcul SPOT.

launchTemplate

Un modèle de lancement facultatif à associer à vos ressources de calcul. Ce paramètre ne s'applique pas aux tâches exécutées sur les ressources Fargate. Ne le spécifiez pas. Tous les autres paramètres de ressources de calcul que vous spécifiez dans une opération [CreateComputeEnvironment](#) ou [UpdateComputeEnvironment](#) API remplacent les mêmes paramètres dans le modèle de lancement. Pour utiliser un modèle de lancement, vous devez spécifier son ID ou son nom dans la demande, mais pas les deux. Pour plus d'informations, consultez [Support de modèle de lancement](#).

Lors de la mise à jour d'un environnement informatique, pour supprimer le modèle de lancement personnalisé et utiliser le modèle de lancement par défaut, définissez la spécification `launchTemplateId` ou le `launchTemplateName` membre de la spécification du modèle de lancement sur une chaîne vide. La suppression du modèle de lancement d'un environnement informatique ne supprime pas l'AMI spécifiée dans le modèle de lancement, si c'est elle qui a été utilisée. Pour mettre à jour l'AMI sélectionnée à partir d'un modèle de lancement, le `updateToLatestImageVersion` paramètre doit être défini sur `true`. Lors de la mise à jour d'un environnement informatique, si vous modifiez le modèle de lancement, une mise à jour de l'infrastructure de l'environnement informatique est requise. Pour plus d'informations, consultez [Mise à jour des environnements informatiques](#).

Type : [LaunchTemplateSpecification](#)

objet

Obligatoire : non

`launchTemplateId`

ID du modèle de lancement.

Type : chaîne

Obligatoire : non

`launchTemplateName`

Le nom du modèle de lancement.

Type : chaîne

Obligatoire : non

version

Numéro de version du modèle de lancement, `$Latest` ou `$Default`.

Si la valeur est `$Latest`, la dernière version du modèle de lancement est utilisée. Si la valeur est `$Default`, la version par défaut du modèle de lancement est utilisée. Lors d'une mise à jour de l'infrastructure, si l'une `$Latest` ou l'autre `$Default` a été spécifiée pour l'environnement informatique, AWS Batch réévalue la version du modèle de lancement et peut utiliser une version différente du modèle de lancement. Et ce, même si le modèle de lancement n'a pas été spécifié dans la mise à jour.

Par défaut: `$Default`.

Type : chaîne

Obligatoire : non

ec2Configuration

Fournit les informations utilisées pour sélectionner Amazon Machine Images (AMI) pour les instances de l'environnement informatique EC2. Si `Ec2Configuration` ce n'est pas spécifié, la valeur par défaut est [Amazon Linux 2](#) (ECS_AL2). Avant le 31 mars 2021, cette valeur par défaut était [Amazon Linux](#) (ECS_AL1) pour les instances autres que le GPU ou AWS Graviton.

Lors de la mise à jour d'un environnement informatique, si vous modifiez ce paramètre, une mise à jour de l'infrastructure de l'environnement informatique est requise. Pour plus d'informations, consultez [Mise à jour des environnements informatiques](#).

Note

Ce paramètre ne s'applique pas aux tâches exécutées sur les ressources Fargate.

Type : tableau d'objets [Ec2Configuration](#)

Obligatoire : non

imageIdOverride

ID d'AMI utilisé pour les instances lancées dans l'environnement informatique correspondant au type d'image. Ce paramètre remplace `imageId`, défini dans l'objet `computeResource`.

Type : chaîne

Obligatoire : non

`imageKubernetesVersion`

KubernetesVersion pour l'environnement de calcul. Si vous ne spécifiez pas de valeur, la dernière version prise en charge par AWS Batch est utilisée.

Type : chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : non

`imageType`

Le type d'image devant correspondre au type d'instance afin de sélectionner une AMI. Les valeurs prises en charge sont différentes pour les ressources ECS et EKS.

ECS

Si le paramètre `imageIdOverride` n'est pas spécifié, une [AMI Amazon Linux 2 optimisée pour Amazon ECS](#) (ECS_AL2) récente est utilisée. Si un nouveau type d'image est spécifié dans une mise à jour, mais qu'aucun `imageIdOverride` paramètre `imageId` n'est spécifié, l'AMI optimisée Amazon ECS la plus récente pour ce type d'image pris en charge par AWS Batch est utilisée.

ECS_AL2

[Amazon Linux 2](#) : par défaut pour toutes les familles d'instance non GPU.

ECS_AL2_NVIDIA

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

ECS_AL1

[Amazon Linux](#). Amazon Linux a atteint le niveau end-of-life de support standard. Pour plus d'informations, consultez [AMI Amazon Linux](#).

EKS

Si le paramètre `imageIdOverride` n'est pas spécifié, une [AMI Amazon Linux optimisée pour Amazon EKS](#) (EKS_AL2) récente est utilisée. Si un nouveau type

d'image est spécifié dans une mise à jour, mais qu'aucun `imageIdOverride` paramètre `imageId` n'est spécifié, l'AMI optimisée Amazon EKS la plus récente pour ce type d'image AWS Batch compatible est utilisée.

EKS_AL2

[Amazon Linux 2](#) : par défaut pour toutes les familles d'instance non GPU.

EKS_AL2_NVIDIA

[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.

Type : chaîne

Contraintes de longueur : longueur minimum de 1. Longueur maximale de 256.

Obligatoire : oui

Configuration d'Amazon EKS

Configuration pour le cluster Amazon EKS qui prend en charge l'environnement AWS Batch informatique. Le cluster doit exister pour permettre la création de l'environnement de calcul.

`eksClusterArn`

Amazon Resource Name (ARN) du cluster Amazon EKS. Par exemple : `arn:aws:eks:us-east-1:123456789012:cluster/ClusterForBatch`.

Type : chaîne

Obligatoire : oui

`kubernetesNamespace`

L'espace de noms du cluster Amazon EKS. AWS Batch gère les pods dans cet espace de noms. La valeur ne peut pas être vide ou null. Elle doit comporter moins de 64 caractères, ne peut pas être définie sur `default`, ne peut pas commencer par «`kube-` » et doit correspondre à cette expression régulière : `^[a-z0-9]([-a-z0-9]*[a-z0-9])?$`. Pour plus d'informations, consultez [Espaces de noms](#) dans la documentation Kubernetes.

Type : chaîne

Obligatoire : oui

Type : [EksConfiguration](#)Objet

Obligatoire : non

Rôle de service

`serviceRole`

Nom de ressource Amazon (ARN) complet du rôle IAM qui permet de passer des appels AWS Batch vers d'autres AWS services en votre nom. Pour plus d'informations, consultez [Utilisation de rôles liés à un service pour AWS Batch](#). Nous vous recommandons de ne pas spécifier le rôle du service. De cette façon, AWS Batch utilise le rôle `AWSServiceRoleForBatch` lié au service.

Important

Si votre compte a déjà créé le rôle AWS Batch lié au service (`AWSServiceRoleForBatch`), ce rôle est utilisé par défaut pour votre environnement informatique, sauf si vous spécifiez un rôle ici. Si le rôle AWS Batch lié au service n'existe pas dans votre compte et qu'aucun rôle n'est spécifié ici, le service essaie de créer le rôle AWS Batch lié au service dans votre compte. Pour de plus amples informations sur le rôle lié à un service `AWSServiceRoleForBatch`, consultez [Autorisations de rôle liées à un service pour AWS Batch](#).

Si l'environnement de calcul est créé à l'aide du rôle `AWSServiceRoleForBatch` lié au service, il ne peut pas être modifié pour utiliser un rôle IAM normal. De même, si l'environnement informatique est créé avec un rôle IAM normal, il ne peut pas être modifié pour utiliser le rôle lié à un `AWSServiceRoleForBatch` service. Pour mettre à jour les paramètres de l'environnement informatique qui nécessitent une mise à jour de l'infrastructure pour être modifiés, le rôle `AWSServiceRoleForBatch` lié au service doit être utilisé. Pour plus d'informations, consultez [Mise à jour des environnements informatiques](#).

Si le rôle que vous avez spécifié possède un chemin autre que `/`, assurez-vous de spécifier l'ARN complet du rôle (recommandé) ou de préfixer le nom du rôle avec le chemin.

Note

Selon la façon dont vous avez créé votre rôle de AWS Batch service, son Amazon Resource Name (ARN) peut contenir le préfixe de `service-role` chemin. Lorsque vous spécifiez uniquement le nom du rôle de service, cela AWS Batch suppose que votre ARN n'utilise pas le préfixe de `service-role` chemin. De ce fait, nous vous recommandons de spécifier l'ARN complet de votre rôle de service lorsque vous créez des environnements de calcul.

Type : chaîne

Obligatoire : non

Balises

tags

Balises de paires clé-valeur à associer à l'environnement informatique. Pour plus d'informations, consultez [Balisage de vos ressources AWS Batch](#).

Type : mappage chaîne/chaîne

Obligatoire : non

Configurations EC2

AWS Batch utilise des AMI optimisées Amazon ECS pour les environnements de calcul EC2 et EC2 Spot. La valeur par défaut est [Amazon Linux 2](#) (ECS_AL2). Avant le 31 mars 2021, cette valeur par défaut était [Amazon Linux](#) (ECS_AL1) pour les instances autres que le GPU ou AWS Graviton.

Note

AWS Batch prend également en charge Amazon Linux 2023.

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](#)

Stratégies d'allocation

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 du service Amazon EC2](#), 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 plus d'informations, consultez [Mise à jour des environnements informatiques](#).

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 AWS Batch 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 présentant un coût 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.

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.

Mise à jour des environnements informatiques

Après avoir créé un environnement informatique qui utilise les ressources EC2, vous pouvez directement mettre à jour de nombreux paramètres de l'environnement informatique. Cependant, la modification de certains paramètres nécessite le AWS Batch remplacement des instances dans l'environnement de calcul.

Pour les environnements de calcul qui utilisent les ressources Fargate, vous pouvez mettre à jour les éléments suivants.

- `securityGroupIds`
- `subnets`
- `desiredvCpus`
- `maxvCpus`
- `minvCpus`

AWS Batch dispose de deux mécanismes de mise à jour. La première est une mise à jour de dimensionnement dans le cadre de laquelle des instances sont ajoutées ou supprimées de l'environnement informatique. La seconde est une mise à jour de l'infrastructure dans le cadre de laquelle les instances de l'environnement informatique sont remplacées. Une mise à jour de l'infrastructure prend beaucoup plus de temps qu'une mise à niveau.

Si vous mettez à jour les environnements informatiques avec AWS Batch, le fait de modifier uniquement ces paramètres entraîne une mise à jour du dimensionnement : `vCPU` souhaités (`desiredvCpus`), nombre maximal de `vCPU` (`maxvCpus`), nombre minimal de `vCPU` (`minvCpus`), rôle de service (`serviceRole`) et état (`state`).

Note

Lorsque vous mettez à jour le `desiredvCpus` paramètre, la valeur doit être comprise entre les `maxvCpus` valeurs `minvCpus` et.

En outre, la `desiredVCpus` valeur mise à jour doit être supérieure ou égale à la `desiredVCpus` valeur actuelle. Pour plus d'informations, consultez [the section called "Message d'erreur lorsque vous mettez à jour le `desiredVCpus` paramètre"](#).

Si l'un des paramètres suivants est modifié dans le cadre d'une action d'[UpdateComputeEnvironment](#) API, AWS Batch lance une mise à jour de l'infrastructure. Une mise à jour de l'infrastructure nécessite que le rôle de service soit défini sur `AWSServiceRoleForBatch` (valeur par défaut) et que la stratégie d'allocation soit `BEST_FIT_PROGRESSIVESPOT_CAPACITY_OPTIMIZED`, ou `SPOT_PRICE_CAPACITY_OPTIMIZED`. `BEST_FIT` n'est pas pris en charge. À l'exception du rôle de service, tous les paramètres pouvant être modifiés pour une mise à jour de dimensionnement peuvent également être modifiés pour une mise à jour de l'infrastructure.

Note

Nous vous recommandons d'utiliser `SPOT_PRICE_CAPACITY_OPTIMIZED` plutôt que `SPOT_CAPACITY_OPTIMIZED` dans la plupart des cas.

Lors d'une mise à jour de l'infrastructure, l'état de l'environnement de calcul passe à `UPDATING`. Les nouvelles instances sont lancées à l'aide des paramètres mis à jour. De nouvelles tâches sont planifiées sur les nouvelles instances. Les tâches en cours d'exécution sont distribuées conformément à la politique de mise à jour de l'infrastructure. Pour plus d'informations, veuillez consulter les sections [UpdateComputeEnvironment](#) et [UpdatePolicy](#) (français non garanti) de la Référence d'API AWS Batch.

Dans le type de `UpdatePolicy` données, considérez les scénarios suivants :

Note

Dans ces scénarios, ce qui suit est vrai. Lorsqu'une instance est arrêtée, les tâches en cours d'exécution sont arrêtées. Par défaut, ces tâches ne sont pas réessayées. Pour réessayer l'une de ces tâches après la fermeture d'une instance, configurez une stratégie de nouvelle tentative de tâche. Pour plus d'informations, consultez [the section called "Nouvelles tentatives de travail automatisées"](#) dans le Guide de l'utilisateur AWS Batch.

- Si le `terminateJobsOnUpdate` paramètre est défini sur `true`, les tâches en cours d'exécution sont interrompues lors d'une mise à jour de l'infrastructure. Le `jobExecutionTimeoutMinutes` paramètre est ignoré.
- Si le `terminateJobsOnUpdate` paramètre est défini sur `false`, les tâches peuvent être exécutées plus longtemps après la mise à jour de l'infrastructure. Ce délai supplémentaire est configuré dans le `jobExecutionTimeoutMinutes` paramètre. Par défaut, le `jobExecutionTimeoutMinutes` réglage est de 30 minutes.

Au fur et à mesure que la capacité devient disponible dans l'environnement informatique, de nouvelles instances sont lancées avec les paramètres mis à jour et des tâches sont lancées sur les nouvelles instances. Au fur et à mesure que toutes les tâches sont terminées sur des instances avec les anciens paramètres, les anciennes instances sont résiliées. La disponibilité de la capacité signifie que le nombre souhaité de vCPU est inférieur au nombre maximum de vCPU d'au moins autant de vCPU que celui requis par le plus petit type d'instance.

Mises à jour de

Une mise à jour de l'infrastructure est nécessaire pour modifier certains paramètres d'un environnement informatique. Si l'un des paramètres suivants est modifié, une mise à jour de l'infrastructure est lancée :

Important

L'environnement informatique doit utiliser le rôle `AWSServiceRoleForBatch` au service pour apporter des modifications nécessitant une mise à jour de l'infrastructure.

Si l'environnement informatique utilise un rôle lié à un service, il ne peut pas être modifié pour utiliser un rôle IAM normal. De même, si l'environnement informatique possède un rôle IAM normal, il ne peut pas être modifié pour utiliser un rôle lié à un service. Par conséquent, vous ne pouvez effectuer des mises à jour d'infrastructure que sur des environnements informatiques créés à l'aide d'un rôle lié à un service.

- La stratégie d'allocation (`allocationStrategy`, doit être soit `BEST_FIT_PROGRESSIVESPOT_CAPACITY_OPTIMIZED`, soit `SPOT_PRICE_CAPACITY_OPTIMIZED`. Si la stratégie d'allocation d'origine l'est `BEST_FIT`, les mises à jour de l'infrastructure ne sont pas prises en charge.)

Note

Nous vous recommandons d'utiliser `SPOT_PRICE_CAPACITY_OPTIMIZED` plutôt que `SPOT_CAPACITY_OPTIMIZED` dans la plupart des cas.

- Pourcentage d'enchère (`bidPercentage`)
- Configuration EC2 () `ec2Configuration`
- Paire de clés (`ec2KeyPair`)
- Identifiant de l'image (`imageId`)
- Rôle d'instance (`instanceRole`)
- Types d'instances (`instanceTypes`)
- Modèle de lancement (`launchTemplate`)
- Groupe de placement (`placementGroup`)
- Groupes de sécurité (`securityGroupIds`)
- Sous-réseaux VPC () `subnets`
- Balises EC2 () `tags`
- Type d'environnement de calcul (`type`, peut être l'un des types suivants : `EC2` ou `SPOT`)
- S'il faut effectuer la mise à jour vers la dernière AMI prise en charge AWS Batch lors d'une mise à jour de l'infrastructure `updateToLatestImageVersion`

Mise à jour de l'ID AMI

Lors d'une mise à jour de l'infrastructure, l'ID d'AMI de l'environnement de calcul peut changer selon que les AMI sont spécifiées ou non dans l'un de ces trois paramètres. Les AMI sont spécifiées dans le `imageId` (`incomputeResources`), `imageIdOverride` (`in ec2Configuration`) ou dans le modèle de lancement spécifié dans `launchTemplate`. Supposons qu'aucun ID d'AMI ne soit spécifié dans aucun de ces paramètres et que le `updateToLatestImageVersion` paramètre soit le `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. La mise à jour doit ensuite supprimer les paramètres qui ont fourni les ID d'AMI. 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 en charge par 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`.

Environnements informatiques Amazon EKS

[Commencer à utiliser AWS Batch Amazon EKS](#) fournit un petit guide pour créer des environnements informatiques EKS. Cette section fournit plus de détails sur les environnements informatiques Amazon EKS.

Rubriques

- [Sélection de l'AMI par défaut](#)
- [Versions de Kubernetes prises en charge](#)

- [Mise à jour de la Kubernetes version de l'environnement informatique](#)
- [Responsabilité partagée des Kubernetes nœuds](#)
- [Exécution DaemonSet d'un nœud AWS Batch géré](#)
- [Personnalisation à l'aide de modèles de lancement](#)

Sélection de l'AMI par défaut

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 les AMI optimisées pour Amazon EKS, consultez la section AMI [Amazon Linux optimisées pour Amazon EKS](#) dans le guide de l'utilisateur Amazon EKS.

Exécutez la commande suivante pour voir quel type d'AMI AWS Batch est 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 1.29 AMI,
# depending on instance types
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1 \
  | jq '.computeEnvironments[].computeResources.ec2Configuration'
[
  {
    "imageType": "EKS_AL2",
    "imageKubernetesVersion": "1.29"
  }
]
```

L'exemple suivant est un type d'instance de GPU.

```
# GPU CE example: indicates Batch has chosen the AL2 x86 EKS Accelerated 1.29 AMI
$ aws batch describe-compute-environments --compute-environments My-Eks-GPU-CE \
  | jq '.computeEnvironments[].computeResources.ec2Configuration'
[
  {
    "imageType": "EKS_AL2_NVIDIA",
    "imageKubernetesVersion": "1.29"
  }
]
```

Versions de Kubernetes prises en charge

AWS Batch sur Amazon, EKS prend actuellement en charge les Kubernetes versions suivantes :

- 1.29
- 1.28
- 1.27
- 1.26
- 1.25
- 1.24
- 1.23

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
```

Mise à jour de la Kubernetes version de l'environnement informatique

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 [UpdateComputeEnvironmentAPI](#).

```
$ aws batch update-compute-environment \
  --compute-environment <compute-environment-name> \
  --compute-resources \
  'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.23}]'
```

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 plus d'informations, consultez [Exécution DaemonSet d'un nœud AWS Batch géré](#).

AWS Batch ne met pas automatiquement à jour les AMI de l'environnement de calcul. Il est de votre responsabilité de les mettre à jour. Exécutez la commande suivante pour mettre à jour vos AMI avec la dernière version d'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 la version **1.23**.

```
$ aws batch update-compute-environment \  
  --compute-environment <compute-environment-name> \  
  --compute-resources \  
    'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.23}]'
```

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 [Mise à jour des environnements informatiques](#) et la politique de mise à jour de l'infrastructure ([UpdatePolicy](#)) définie dans le fonctionnement de l'[UpdateComputeEnvironmentAPI](#).

Exécution DaemonSet d'un nœud AWS Batch géré

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"
```


Personnalisation à l'aide de modèles de lancement

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

AWS Batch court `/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_KUBELET_EXTRA_ARGS` variable dans le `/etc/aws-batch/batch.config` fichier. Consultez l'exemple suivant pour plus de détails.

ℹ Note

Si le modèle de lancement est modifié après [CreateComputeEnvironment](#) son appel, [UpdateComputeEnvironment](#) il 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](#)
- [Montage d'un volume Amazon EFS](#)
- [Prise en charge d'IPv6](#)

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, `--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==--
```

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 [Kubernetes documentation](#).

Note

La variable d'`CONTAINER_RUNTIME` environnement est équivalente à l'`--container-runtime` option de `bootstrap.sh`. Pour plus d'informations, consultez [Options](#) la [Kubernetes documentation](#).

```
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==--
```

Montage d'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.

Prise en charge d'IPv6

AWS Batch prend en charge les clusters Amazon EKS dotés d'adresses IPv6. 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 adresses IPv6 aux pods et aux services](#) dans le guide de l'utilisateur Amazon EKS.

Ressource de calcul Gestion de la mémoire

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 instances Amazon EC2. 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 ressource de calcul AWS Batch par défaut réserve également 32 Mio de mémoire pour l'agent de conteneur Amazon ECS et d'autres processus système critiques. Cette mémoire n'est pas disponible pour l'allocation des tâches. Pour plus d'informations, consultez [Réservation de mémoire système](#).

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.

Réservation de mémoire système

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 ressource de calcul AWS Batch par défaut réserve 32 Mio de mémoire pour l'agent de conteneur Amazon ECS et d'autres processus système critiques.

Affichage de la mémoire d'une ressource 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 partir de l'adresse <https://console.aws.amazon.com/ecs/v2>.
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 mémoire enregistrée 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`

Exemples de 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, un maximum de 2 vCPU est alloué à la tâche. 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 vCPU 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 dimensionnement et la prise de décision de AWS Batch. Une fois les instances en cours d'exécution, AWS Batch ajuste les allocations initiales.

Exemple de réservation du processeur d'un nœud

La valeur de réservation du processeur est calculée en millicores en utilisant le nombre total de vCPU 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 avec 2 vCPU est de 70 m. Ceci est calculé de la manière suivante : $(1*60) + (1*10) = 70$ m.
- La valeur de réservation du processeur pour une `c5.24xlarge` instance comportant 96 vCPU est de 310 m. Ceci est calculé de la manière suivante : $(1*60) + (1*10) + (2*5) + (92*2,5) = 310$ 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 de réservation de mémoire de 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 : $((11 * \text{nombre maximum de pods pris en charge par le type d'instance}) + 255)$. 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 :
 $(\text{Instance_Capacity_in_MiB}) - (11 * (\text{maximum_number_of_pods})) - 255 - (\text{valeur. HardEvictionLimit})$.

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 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

Batchalloue 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.

Politiques 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 des politiques de planification, vous pouvez attribuer différents identifiants de partage équitable aux charges de travail ou aux utilisateurs. AWS Batch attribue à chaque identifiant de partage équitable un pourcentage du total des ressources disponibles pendant une période donnée.

Le pourcentage de juste part est calculé à l'aide `shareDecaySeconds` des `shareDistribution` valeurs et. Vous pouvez ajouter du temps à l'analyse des parts équitables en attribuant une durée de déclin des actions 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 équitable qui ne sont pas actifs en spécifiant une réservation de calcul. Pour plus d'informations, consultez [Paramètres de la politique de planification](#).

Rubriques

- [Création d'une politique de planification](#)
- [Paramètres de la politique de planification](#)

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, vous associez un ou plusieurs identifiants de partage équitable ou préfixes d'identifiant de partage équitable aux pondérations de 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/](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 décroissance du partage défini par la politique de planification. Un temps de

dégradation des parts plus long permettra de prendre en compte l'utilisation des ressources informatiques sur une plus longue période lors de la planification des tâches. Cela peut permettre aux emplois utilisant un identifiant de juste part d'utiliser temporairement plus de ressources de calcul que le poids de cet identifiant de juste part ne le permettrait si cet identifiant de juste part n'avait pas récemment utilisé de ressources de calcul.

6. (Facultatif) Pour Calculer la réservation, entrez une valeur entière pour la réservation informatique de la politique de planification. La réservation de calcul conservera certains vCPU en réserve qui seront utilisés pour des identifiants de partage équitable qui ne sont pas actifs actuellement.

Le ratio réservé est $(computeReservation/100)^{ActiveFairShares}$, où *ActiveFairShares* est le nombre d'identifiants de partage équitable actifs.

Par exemple, une *computeReservation* valeur de 50 indique que l'on AWS Batch devrait réserver 50 % du maximum de processeurs virtuels disponibles s'il n'existe qu'un seul identifiant de juste part, 25 % s'il existe deux identifiants de partage équitable et 12,5 % s'il existe trois identifiants de partage équitable. Une *computeReservation* valeur de 25 indique que 25 % du maximum de processeurs virtuels disponibles AWS Batch devraient être réservés s'il n'existe qu'un seul identifiant de juste part, 6,25 % s'il existe deux identifiants de juste part et 1,56 % s'il existe trois identifiants de juste part.

7. Dans la section Attributs de partage, vous pouvez spécifier l'identifiant de partage équitable et le poids de chaque identifiant de partage équitable à associer à la politique de planification.
 - a. Choisissez Ajouter un identifiant de partage.
 - b. Pour Identifiant de partage, spécifiez l'identifiant de partage équitable. Si la chaîne se termine par « * », cela devient un préfixe d'identifiant de partage équitable utilisé pour faire correspondre les identifiants de partage équitable pour les emplois. Tous les identifiants de partage équitable et les préfixes d'identifiant de partage équitable 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 équitable « UserA* » et l'identifiant de partage équitable « UserA1 » dans la même politique de planification.
 - c. Dans le champ Facteur de pondération, spécifiez le poids relatif pour l'identifiant de la juste part. 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 équitable est utilisé, les tâches dont les identifiants de partage équitable commencent par le préfixe partageront le facteur de pondération. Cela augmente effectivement le facteur de pondération pour ces emplois,

abaissant leur priorité individuelle tout en maintenant le même facteur de pondération pour le préfixe d'identification de la juste part.

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 plus d'informations, consultez [Balisage de vos ressources AWS Batch](#).
9. Choisissez Soumettre pour terminer et créer votre politique de planification.

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.

```
{
  "name": "",
  "fairsharePolicy": {
    "shareDecaySeconds": 0,
    "computeReservation": 0,
    "shareDistribution": [
      {
        "shareIdentifier": "",
        "weightFactor": 0.0
      }
    ]
  },
  "tags": {
    "KeyName": ""
  }
}
```

Note

Vous pouvez générer le modèle de file d'attente de tâches précédent à l'aide de la AWS CLI commande suivante.

```
$ aws batch create-scheduling-policy --generate-cli-skeleton
```

Paramètres de la politique de planification

Les politiques de planification sont divisées en trois éléments de base : le nom, la politique de partage équitable et les balises de la politique de planification.

Rubriques

- [Nom de la politique de planification](#)
- [Politique de partage équitable](#)
- [Balises](#)

Nom de la politique de planification

name

Le nom de votre politique de planification. Il peut comporter jusqu'à 128 lettres (majuscules et minuscules), des chiffres, des tirets ou des traits de soulignement.

Type : chaîne

Obligatoire : oui

Politique de partage équitable

fairsharePolicy

La politique de partage équitable de la politique de planification.

```
"fairsharePolicy": {
  "computeReservation": number,
  "shareDecaySeconds": number,
  "shareDistribution": [
    {
      "shareIdentifier": "string",
      "weightFactor": number
    }
  ]
}
```

Type : objet

Obligatoire : non

`computeReservation`

Une valeur utilisée pour réserver une partie du nombre maximal de processeurs virtuels disponibles pour des identifiants de partage équitables qui n'ont pas encore été utilisés.

Le ratio réservé est $(\text{computeReservation}/100)^{\text{ActiveFairShares}}$, où `ActiveFairShares` est le nombre d'identifiants de partage équitable actifs.

Par exemple, une `computeReservation` valeur de 50 indique que l'on AWS Batch devrait réserver 50 % du maximum de processeurs virtuels disponibles s'il n'existe qu'un seul identifiant de fair share actif, 25 % s'il existe deux identifiants de fair share actifs et 12,5 % s'il existe trois identifiants de fair share actifs. Une `computeReservation` valeur de 25 indique que 25 % du maximum de processeurs virtuels disponibles AWS Batch devraient être réservés s'il n'existe qu'un seul identifiant de juste action actif, 6,25 % s'il existe deux identifiants de fair share actifs et 1,56 % s'il existe trois identifiants de fair share actifs.

Type : entier

Plage valide : valeur minimale de 0. Valeur maximale de 99.

Obligatoire : non

`shareDecaySeconds`

Période à utiliser pour calculer un pourcentage de juste part pour chaque identifiant de juste part utilisé. La valeur zéro (0) indique que seule l'utilisation actuelle doit être mesurée. La décroissance permet aux tâches dont l'exécution est plus récente d'avoir plus de poids que les tâches exécutées auparavant.

Type : entier

Plage valide : valeur minimale de 0. Valeur maximale de 604800 (1 semaine).

Obligatoire : non

`shareDistribution`

Ensemble d'objets contenant les pondérations pour les identificateurs de parts équitables utilisés dans le cadre de la politique de partage équitable. Les identifiants de partage équitable qui ne sont pas inclus ont une pondération par défaut de 1.0.

```
"shareDistribution": [  
  {  
    "shareIdentifier": "string",  
    "weightFactor": number  
  }  
]
```

Type : Array

Obligatoire : non

shareIdentifier

Un identifiant de partage équitable ou un préfixe d'identifiant de partage équitable. Si la chaîne se termine par « * », cette chaîne spécifie un préfixe d'identifiant de partage équitable pour les identifiants de partage équitable commençant par ce préfixe. Par exemple, si la valeur est `UserA*` égale `weightFactor` à 1 et que deux identifiants de juste action commencent par `deuxUserA`, chacun de ces identifiants de juste action aura une pondération de 2 ; s'il existe cinq identifiants de juste action de ce type, chacun aura une pondération de 5.

La liste des identifiants d'actions équitables et des préfixes des identifiants d'actions équitables dans une politique de partage équitable ne peut pas se chevaucher. Par exemple, vous ne pouvez pas avoir de préfixe d'identifiant de juste part `UserA*` et d'identifiant de juste part `UserA-1` dans la même politique de partage équitable.

Type : chaîne

Obligatoire : oui

weightFactor

Facteur de pondération pour l'identifiant de partage équitable. La valeur par défaut est 1.0. Une valeur inférieure a une priorité plus élevée pour les ressources de calcul. Par exemple, les tâches qui utilisent un identifiant de partage avec un facteur de pondération de 0,125 (1/8) obtiennent 8 fois les ressources de calcul des tâches qui utilisent un identifiant de partage avec un facteur de pondération de 1.

La plus petite valeur prise en charge est 0,0001 et la plus grande valeur prise en charge est 999,9999.

Type : flottant

Obligatoire : non

Balises

tags

Balises de paires clé-valeur à associer à la politique de planification. Pour plus d'informations, consultez [Balisage de vos ressources AWS Batch](#).

Type : mappage chaîne/chaîne

Obligatoire : non

Orchestrez AWS Batch des tâches avec les machines d'état Step Functions dans la console AWS Batch

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

- [Affichage des détails de la machine d'état](#)
- [Modification d'une machine d'état](#)
- [Exécution d'une machine d'état](#)

Affichage des détails de la machine d'état

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 console AWS Batch ouvre la page Détails .

</result>

Pour plus d'informations, consultez [Step Functions](#) dans le Manuel du développeur AWS Step Functions.

Modification d'une machine d'état

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.

Exécution d'une machine d'état

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 sur AWS Fargate

AWS 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 AWS 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.

Table des matières

- [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 vCPU
- Plus de 120 Gibioctets (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 [AWS de service Fargate dans le manuel Amazon Elastic Container Service Developer Guide](#).

Définitions de poste sur Fargate

AWS Batch les tâches sur 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. Par conséquent, la limite de mémoire doit 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 plus d'informations, consultez [Utilisation du pilote du 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 plus d'informations, consultez [AWS Batch rôle d'exécution IAM](#).

Files d'attente pour les emplois sur Fargate

AWS Batch les files d'attente de tâches sur 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 sur Fargate 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'computeResourcesobjet

`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

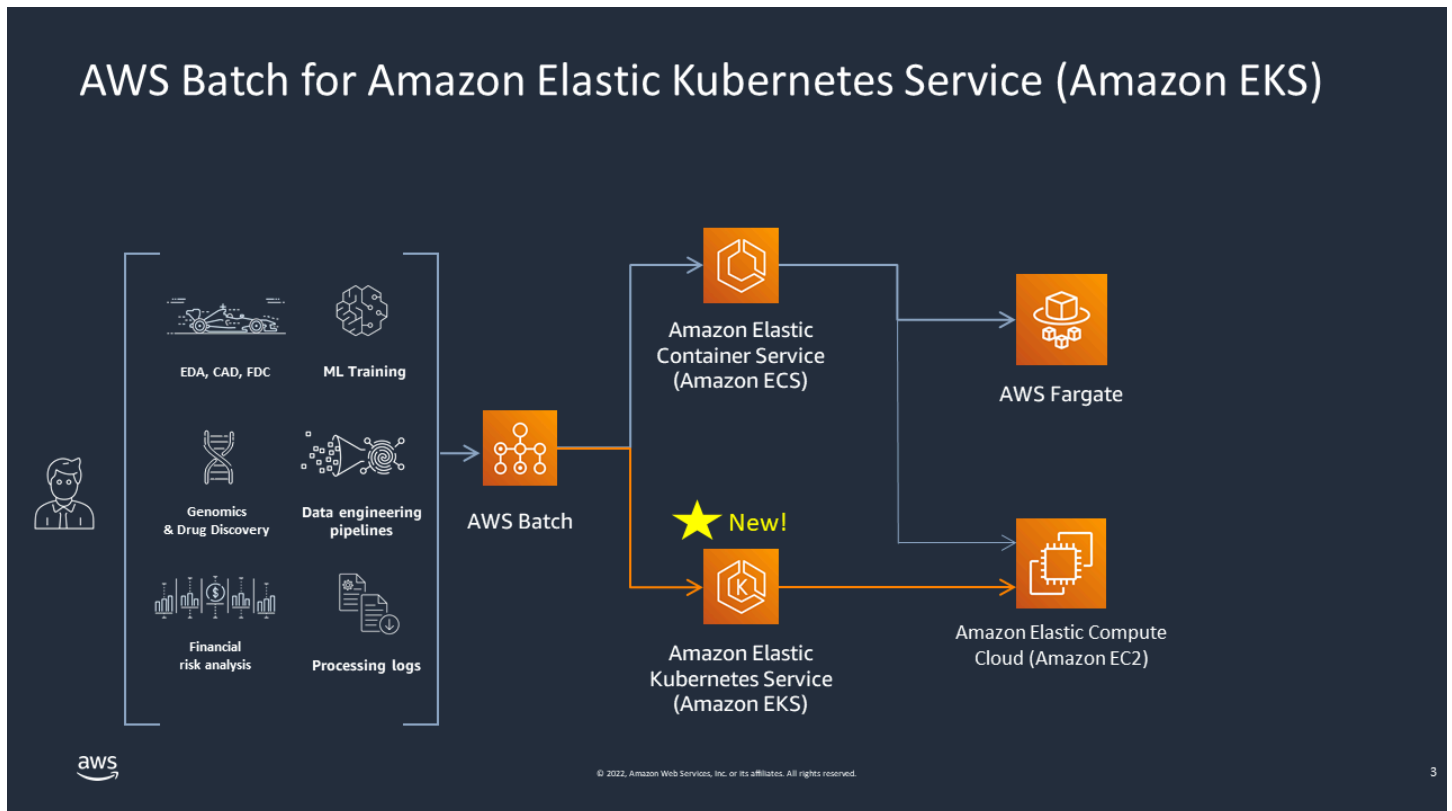
Cela ne s'applique pas aux environnements informatiques Fargate et ne peut pas être fourni. Pour spécifier des balises pour les environnements de calcul Fargate, utilisez `tags` le paramètre qui ne figure pas dans l'objet `computeResources`

type

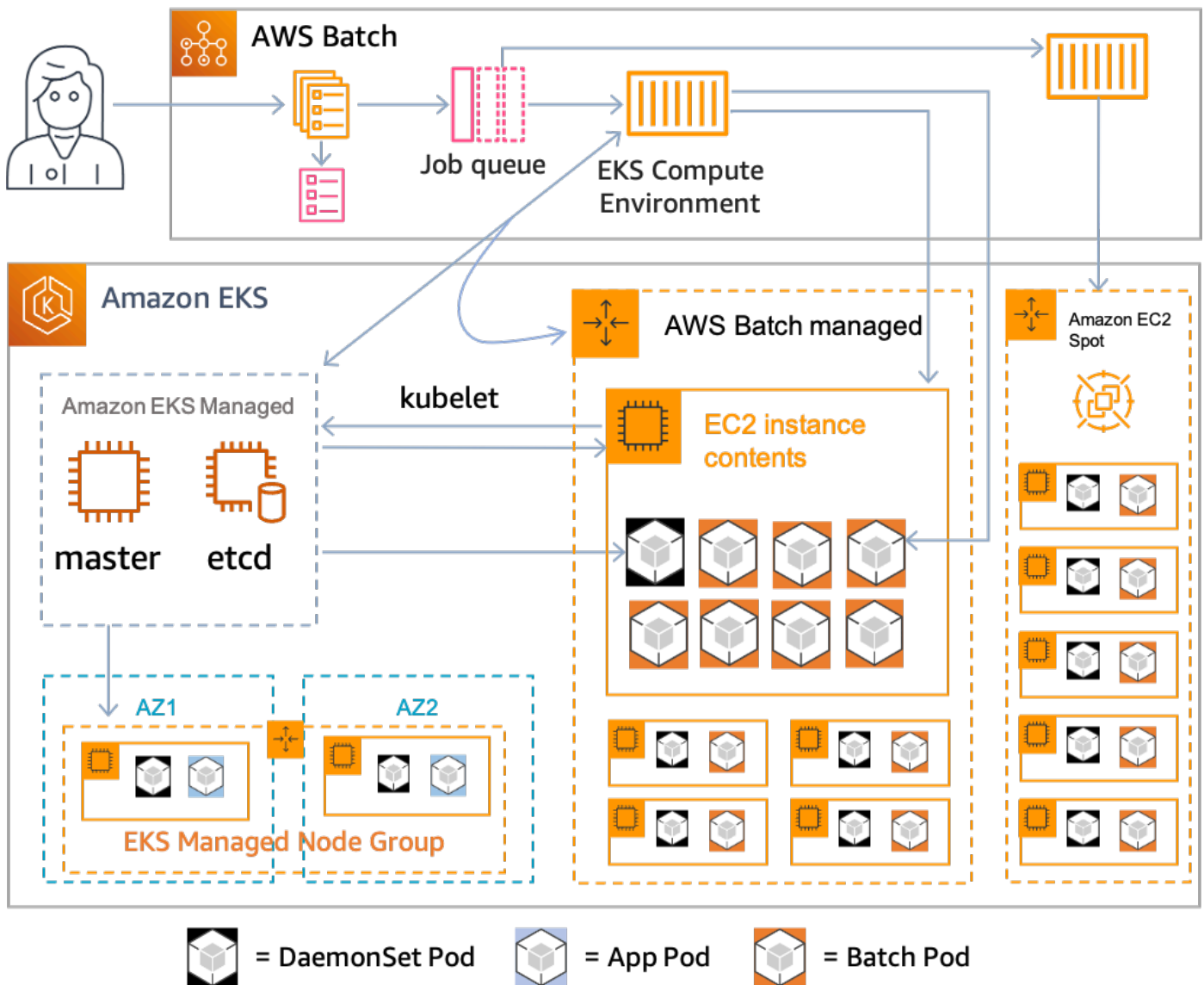
Ce doit être `FARGATE` ou `FARGATE_SPOT`.

```
"type": "FARGATE_SPOT"
```

AWS Batch sur 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 AWS Batch 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 composant Kubernetes (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 Kubernetes. AWS Batch appelle l'API Kubernetes pour créer, surveiller et supprimer des pods et des nœuds.

AWS Batch dispose d'une logique de dimensionnement intégrée pour dimensionner 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 réduit le nombre de nœuds.

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 plus d'informations, consultez [Commencer à utiliser AWS Batch Amazon EKS](#).

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 avec les EFA, consultez la section [Types d'instances pris en charge](#) dans le guide de l'utilisateur Amazon EC2.

Tip

Pour voir la liste des types d'instances qui prennent en charge les EFA dans 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 EFA, consultez [Elastic Fabric Adapter](#) dans le guide de l'utilisateur Amazon EC2.

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 à des ressources et à des 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.

Prise en main

Une politique IAM doit accorder ou refuser l'autorisation d'utiliser une ou plusieurs AWS Batch actions.

Rubriques

- [Structure d'une politique](#)
- [Autorisations au niveau des ressources prises en charge pour les opérations d'API Amazon AWS Batch](#)
- [Exemples de politiques](#)
- [Stratégie gérée par AWS Batch](#)
- [Création de AWS Batch politiques IAM](#)
- [Rôle d'instance Amazon ECS](#)
- [Rôle de la flotte Amazon EC2 Spot](#)

- [EventBridge Rôle IAM](#)

Structure d'une politique

Les rubriques suivantes expliquent la structure d'une politique IAM.

Rubriques

- [Syntaxe d'une politique](#)
- [Actions pour AWS Batch](#)
- [Amazon Resource Names \(ARN\) pour AWS Batch](#)
- [Vérification que les utilisateurs ont les 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"
      }
    }
  ]
}
```

Une déclaration se compose de différents éléments :

- **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 pour AWS Batch](#).
- **Ressource** : 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 au niveau des ressources prises en charge pour les opérations d'API Amazon AWS Batch](#) et [Amazon Resource Names \(ARN\) 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 [Création de AWS Batch politiques IAM](#).

Actions 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 `batch: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 BatchAPI.

Amazon Resource Names (ARN) pour AWS Batch

Chaque déclaration de politique IAM s'applique aux ressources que vous spécifiez à l'aide de leurs Amazon Resource Names (ARN).

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

ID Compte AWS du compte sans trait d'union (123456789012, par exemple).

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 BatchLes opérations d'API prennent actuellement en charge les autorisations au niveau des ressources pour plusieurs opérations d'API. Pour plus d'informations, consultez [Autorisations au niveau des ressources prises en charge pour les opérations d'API Amazon AWS Batch](#). Pour spécifier toutes les ressources, ou si une action d'API spécifique ne prend pas en charge les ARN, incluez un caractère générique (*) dans l'Resourceélément.

```
"Resource": "*"
```

Vérification que les utilisateurs ont les 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.

Autorisations au niveau des ressources prises en charge pour les opérations d'API Amazon 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.

La liste suivante décrit les actions d'AWS Batch API qui prennent actuellement en charge les autorisations au niveau des ressources. La liste décrit également les ressources prises en charge, les ARN des ressources et les clés de condition pour chaque action.

Important

Si une action d'AWS Batch API ne figure pas dans cette liste, cela signifie qu'elle ne prend pas en charge les autorisations au niveau des ressources. Si une action d'AWS Batch API ne prend pas en charge les autorisations au niveau des ressources, vous pouvez autoriser les utilisateurs à utiliser l'action. Toutefois, vous devez inclure un caractère générique (*) pour l'élément ressource de votre déclaration de politique.

Actions

[CancelJob](#), [CreateComputeEnvironment](#), [CreateJobQueue](#), [CreateSchedulingPolicy](#), [DeleteComputeEnvironment](#), [DeleteJobQueue](#), [DeleteSchedulingPolicy](#), [DeregisterJobDefinition](#), [ListTagsForResource](#), [RegisterJobDefinition](#), [SubmitJob](#), [TagResource](#), [TerminateJob](#), [UntagResource](#), [UpdateComputeEnvironment](#), [UpdateSchedulingPolicy](#), [UpdateJobQueue](#)

[CancelJob](#)

Annule une tâche dans une AWS Batch file d'attente.

Ressource

Job

arn:aws:batch : region : account:job/ JobID

Clés de condition

aws:ResourceTag/\${TagKey} (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

CreateComputeEnvironment

Crée un environnement AWS Batch informatique.

Ressource

Environnement informatique

arn:aws:batch : region : account:compute-environment/ compute-environment-name

Clés de condition

aws:ResourceTag/\${TagKey} (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Clés de condition

aws:RequestTag/\${TagKey} (chaîne)

Filtre les actions en fonction des balises qui sont transmises dans la demande.

aws:TagKeys (chaîne)

Filtre les actions en fonction des clés de balise qui sont transmises dans la demande.

CreateJobQueue

Crée une file AWS Batch d'attente de tâches.

Ressource

Environnement informatique

arn:aws:batch : region : account:compute-environment/ compute-environment-name

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

File d'attente de jobs

`arn:aws:batch : region : account:job-queue/queue-name`

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Politique de planification

`arn:aws:batch : region : account:scheduling-policy/ scheduling-policy-name`

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Clés de condition

`aws:RequestTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises qui sont transmises dans la demande.

`aws:TagKeys` (chaîne)

Filtre les actions en fonction des clés de balise qui sont transmises dans la demande.

DeleteComputeEnvironment

Supprime tout environnement AWS Batch informatique.

Ressource

Environnement informatique

`arn:aws:batch : region : account:compute-environment/ compute-environment-name`

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

[CreateSchedulingPolicy](#)

Crée une politique de AWS Batch planification.

Ressource

Politique de planification

arn:aws:batch : region : account:scheduling-policy/ scheduling-policy-name

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Clés de condition

`aws:RequestTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises qui sont transmises dans la demande.

`aws:TagKeys` (chaîne)

Filtre les actions en fonction des clés de balise qui sont transmises dans la demande.

[DeleteJobQueue](#)

Supprime la file d'attente de tâches spécifiée. La suppression de la file d'attente des tâches entraîne finalement la suppression de toutes les tâches de la file d'attente. Les tâches sont supprimées à un rythme d'environ 16 tâches par seconde.

Ressource

File d'attente de jobs

arn:aws:batch : region : account:job-queue/queue-name

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

[DeleteSchedulingPolicy](#)

Supprime la politique de planification spécifiée.

Ressource

Politique de planification

arn:aws:batch : region : account:scheduling-policy/ scheduling-policy-name

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

[DeregisterJobDefinition](#)

Désenregistre une AWS Batch définition de poste.

Ressource

Définition du job

arn:aws:batch : region : account:job-definition/ definition-name : revision

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

[ListTagsForResource](#)

Répertorie les balises pour la ressource spécifiée.

Ressource

Environnement informatique

arn:aws:batch : region : account:compute-environment/ compute-environment-name

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Job

arn:aws:batch : region : account:job/ JobID

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Définition du job

`arn:aws:batch : region : account:job-definition/ definition-name : revision`

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

File d'attente de jobs

`arn:aws:batch : region : account:job-queue/queue-name`

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Politique de planification

`arn:aws:batch : region : account:scheduling-policy/ scheduling-policy-name`

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

[RegisterJobDefinition](#)

Registres et AWS Batch définitions.

Ressource

Définition du job

`arn:aws:batch : region : account:job-definition/ definition-name`

Clés de condition

`batch:AWSLogsCreateGroup`(Booléen)

Lorsque ce paramètre est vrai, le `awslogs-group` est créé pour les journaux.

`batch:AWSLogsGroup` (chaîne)

Le `awslogs` groupe dans lequel se trouvent les journaux.

`batch:AWSLogsRegion` (chaîne)

Région vers laquelle les journaux sont envoyés.

`batch:AWSLogsStreamPrefix` (chaîne)

Le préfixe du flux de `awslogs` log.

`batch:Image` (chaîne)

L'image Docker utilisée pour démarrer une tâche.

`batch:LogDriver` (chaîne)

Le pilote de journal utilisé pour la tâche.

`batch:Privileged`(Booléen)

Lorsque ce paramètre est vrai, le conteneur de la tâche reçoit des autorisations élevées sur l'instance de conteneur hôte.

`batch:User` (chaîne)

Le nom d'utilisateur ou l'UID numérique à utiliser dans le conteneur pour la tâche.

`aws:RequestTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises qui sont transmises dans la demande.

`aws:TagKeys` (chaîne)

Filtre les actions en fonction des clés de balise qui sont transmises dans la demande.

[SubmitJob](#)

Soumet une AWS Batch tâche à partir d'une définition de tâche.

Ressource

Job

`arn:aws:batch : region : account:job/ JobID`

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Définition du job

arn:aws:batch : region : account:job-definition/ definition-name [: revision]

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Note

Cette clé ne peut être utilisée que lorsque la définition de tâche Amazon Resource Name (ARN) est au format `arn:aws:batch:region:account_number:job-definition/definition-name:revision`.

File d'attente de jobs

arn:aws:batch : region : account:job-queue/queue-name

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

TagResource

Balise la ressource spécifiée.

Ressource

Environnement informatique

arn:aws:batch : region : account:compute-environment/ compute-environment-name

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Job

arn:aws:batch : region : account:job/ JobID

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Définition du job

arn:aws:batch : region : account:job-definition/ definition-name : revision

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

File d'attente de jobs

arn:aws:batch : region : account:job-queue/queue-name

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Politique de planification

arn:aws:batch : region : account:scheduling-policy/ scheduling-policy-name

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Clés de condition

`aws:RequestTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises qui sont transmises dans la demande.

`aws:TagKeys` (chaîne)

Filtre les actions en fonction des clés de balise qui sont transmises dans la demande.

TerminateJob

Met fin à une tâche dans une file d'attente de AWS Batch tâches.

Ressource

Job

arn:aws:batch : region : account:job/ JobID

Clés de condition

aws:ResourceTag/\${TagKey} (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

UntagResource

Débalise la ressource spécifiée.

Ressource

Environnement informatique

arn:aws:batch : region : account:compute-environment/ compute-environment-name

Clés de condition

aws:ResourceTag/\${TagKey} (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Job

arn:aws:batch : region : account:job/ JobID

Clés de condition

aws:ResourceTag/\${TagKey} (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Définition du job

arn:aws:batch : region : account:job-definition/ definition-name : revision

Clés de condition

aws:ResourceTag/\${TagKey} (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

File d'attente de jobs

arn:aws:batch : region : account:job-queue/queue-name

Clés de condition

aws:ResourceTag/\${TagKey} (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Politique de planification

arn:aws:batch : region : account:scheduling-policy/ scheduling-policy-name

Clés de condition

aws:ResourceTag/\${TagKey} (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Clés de condition

aws:TagKeys (chaîne)

Filtre les actions en fonction des clés de balise qui sont transmises dans la demande.

UpdateComputeEnvironment

Met à jour un environnement AWS Batch informatique.

Ressource

Environnement informatique

arn:aws:batch : region : account:compute-environment/ compute-environment-name

Clés de condition

aws:ResourceTag/\${TagKey} (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

UpdateJobQueue

Met à jour une file d'attente de tâches.

Ressource

File d'attente de jobs

arn:aws:batch : region : account:job-queue/queue-name

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Politique de planification

arn:aws:batch : region : account:scheduling-policy/ scheduling-policy-name

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

UpdateSchedulingPolicy

Met à jour une politique de planification.

Ressource

Politique de planification

arn:aws:batch : region : account:scheduling-policy/ scheduling-policy-name

Clés de condition

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

Clés de condition pour les actions AWS Batch d'API

AWS Batch définit les clés de condition suivantes qui sont utilisées dans l'élément d'une politique IAM. Vous pouvez utiliser ces touches pour affiner les conditions auxquelles s'applique la déclaration de politique. Pour consulter les clés de condition globales disponibles pour tous les services, consultez les [clés de condition globales disponibles](#) dans le guide de l'utilisateur IAM.

`batch:AWSLogsCreateGroup`(Booléen)

Lorsque ce paramètre est vrai, le `awslogs-group` est créé pour les journaux.

`batch:AWSLogsGroup` (chaîne)

Le `awslogs` groupe dans lequel se trouvent les journaux.

`batch:AWSLogsRegion` (chaîne)

L'Région AWS où les journaux sont envoyés.

`batch:AWSLogsStreamPrefix` (chaîne)

Le préfixe du flux de `awslogs` log.

`batch:Image` (chaîne)

L'image Docker utilisée pour démarrer une tâche.

`batch:LogDriver` (chaîne)

Le pilote de journal utilisé pour la tâche.

`batch:Privileged` (Booléen)

Lorsque ce paramètre est vrai, le conteneur de la tâche reçoit des autorisations élevées sur l'instance de conteneur hôte (comme pour l'utilisateur `root`).

`aws:ResourceTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises associées à la ressource.

`aws:RequestTag/${TagKey}` (chaîne)

Filtre les actions en fonction des balises qui sont transmises dans la demande.

`batch:ShareIdentifier` (chaîne)

Filtre les actions en fonction du `shareIdentifier` paramètre envoyé à [SubmitJob](#).

`aws:TagKeys` (chaîne)

Filtre les actions en fonction des clés de balise qui sont transmises dans la demande.

`batch:User` (chaîne)

Le nom d'utilisateur ou l'ID utilisateur numérique (`uid`) à utiliser dans le conteneur pour la tâche.

Exemples de politiques

Les exemples suivants présentent des déclarations de politique que vous pouvez utiliser pour contrôler les autorisations accordées aux utilisateurs AWS Batch.

Exemples

- [Accès en lecture seule](#)
- [Limitation à l'utilisateur POSIX, à l'image Docker, au niveau de privilège et au rôle lors de la soumission des tâches](#)
- [Restreindre au préfixe de définition de tâche lors de la soumission de la tâche](#)
- [Limiter à la file d'attente des tâches](#)
- [Refuser l'action lorsque la condition est que toutes les clés correspondent à des chaînes](#)
- [Refuser l'action lorsque des clés de condition correspondent à des chaînes](#)
- [Utiliser la clé de batch:ShareIdentifier condition](#)

Accès en lecture seule

La politique suivante autorise les utilisateurs à utiliser toutes les actions d'AWS BatchAPI dont le nom commence par Describe etList.

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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:Describe*",
        "batch:List*"
      ],
      "Resource": "*"
    }
  ]
}
```

Limitation 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 A_. JobDef

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 à `uid` partir de 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "arn:aws:batch:<aws_region>:<aws_account_id>;job-definition/JobDefA_*"
      ],
      "Condition": {
        "StringEquals": {
          "batch:User": [
            "nobody"
          ]
        }
      }
    }
  ]
}
```

```

        "batch:Image": [
            "<aws_account_id>.dkr.ecr.<aws_region>.amazonaws.com/myImage"
        ]
    },
    "Bool": {
        "batch:Privileged": "false"
    }
}
},
{
    "Effect": "Allow",
    "Action": [
        "batch:DeregisterJobDefinition"
    ],
    "Resource": [
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/JobDefA_*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "iam:PassRole"
    ],
    "Resource": [
        "arn:aws:iam::<aws_account_id>:role/MyBatchJobRole"
    ]
}
]
}

```

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 dont le nom de définition de tâche commence 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/JobDefA_*",
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-queue/*"
      ]
    }
  ]
}
```

Limiter à la file d'attente des 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/*",
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-queue/queue1"
      ]
    }
  ]
}
```

```
}
```

Refuser l'action lorsque la condition est que toutes les clés 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 à la fois « *string1* » et que 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 de `batch:ShareIdentifier`, toutes les clés de `batch:condition` peuvent être utilisées de cette manière. La clé de `batch:ShareIdentifier` est définie pour une tâche, et non pour une définition de tâche.

```
{
  "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"
        }
      }
    }
  ]
}
```

```

    }
  }
}
]
}

```

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:ShareIdentifier` condition est définie pour une tâche, pas pour une définition de tâche.)

```

{
  "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"
        ]
      }
    }
  }
}
]
}

```

Utiliser la clé de **batch:ShareIdentifïer** condition

Utilisez la politique suivante pour soumettre les tâches qui utilisent la définition de jobDefA tâche à la file d'attente des jobqueue1 tâches avec l'identifiant de lowCpu partage.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws::batch:<aws_region>:<aws_account_id>;job-definition/JobDefA",

```

```

    "arn:aws::batch:<aws_region>:<aws_account_id>:job-queue/jobqueue1"
  ],
  "Condition": {
    "StringEquals": {
      "batch:ShareIdentifier": [
        "lowCpu"
      ]
    }
  }
}
]
}

```

Stratégie gérée par AWS Batch

AWS Batch fournit une politique gérée que vous pouvez associer aux utilisateurs et qui 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 stratégie permet un accès administrateur complet à AWS Batch.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:*",
        "cloudwatch:GetMetricStatistics",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeVpcs",
        "ec2:DescribeImages",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions",
        "ecs:DescribeClusters",
        "ecs:Describe*",
      ]
    }
  ]
}

```

```

    "ecs:List*",
    "eks:DescribeCluster",
    "eks:ListClusters",
    "logs:Describe*",
    "logs:Get*",
    "logs:TestMetricFilter",
    "logs:FilterLogEvents",
    "iam:ListInstanceProfiles",
    "iam:ListRoles"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "iam:PassRole"
  ],
  "Resource": [
    "arn:aws:iam::*:role/AWSBatchServiceRole",
    "arn:aws:iam::*:role/service-role/AWSBatchServiceRole",
    "arn:aws:iam::*:role/ecsInstanceRole",
    "arn:aws:iam::*:instance-profile/ecsInstanceRole",
    "arn:aws:iam::*:role/iaws-ec2-spot-fleet-role",
    "arn:aws:iam::*:role/aws-ec2-spot-fleet-role",
    "arn:aws:iam::*:role/AWSBatchJobRole*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "iam:CreateServiceLinkedRole"
  ],
  "Resource": "arn:aws:iam::*:role/*Batch*",
  "Condition": {
    "StringEquals": {
      "iam:AWSServiceName": "batch.amazonaws.com"
    }
  }
}
]
}

```

Création de AWS Batch politiques IAM

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](#).

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 le [rôle IAM de l'instance de conteneur Amazon ECS](#) dans le guide du développeur Amazon Elastic Container Service.

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.

Pour vérifier le rôle **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, sélectionnez 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 AmazonEC2 EC2Role. ContainerServicefor
- f. Cochez la case à côté d'AmazonEC2 ContainerServicefor EC2Role, 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.

Pour créer un rôle IAM et 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`.

```
{
  "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
```

Voici un exemple de réponse.

```
{
  "Role": {
    "Path": "/",
    "RoleName": "ecsInstanceRole",
    "RoleId": "AROAT46P5RDIY4EXAMPLE",
    "Arn": "arn:aws:iam::123456789012:role/ecsInstanceRole",
    "CreateDate": "2022-12-12T23:46:37.247Z",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "ec2.amazonaws.com"
          },
          "Action": "sts:AssumeRole",
        }
      ]
    }
  }
}
```

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'AWSAPI 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'`ecsInstanceRoleinstance`.

```
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 `AWSServiceRoleForEC2SpotFleet` liés aux services `AWSServiceRoleForEC2Spot` et 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 Amazon EC2 Spot avec 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. Sélectionnez Créer un rôle.

Note

Dans le passé, il existait deux politiques gérées pour le rôle Amazon EC2 Spot Fleet.

- AmazonEC2 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 plus d'informations, consultez [Instances ponctuelles non étiquetées lors de la création](#).

- `AmazonEC2SpotFleetTaggingRole` : ce rôle fournit toutes les autorisations nécessaires pour baliser les instances Amazon EC2 Spot. Utilisez ce rôle pour autoriser l'utilisation de balises d'instances Spot dans des environnements informatiques AWS Batch.

Créez des rôles pour les flottes Amazon EC2 Spot avec AWS CLI

Pour créer le rôle `SpotFleetTaggingRole` IAM AmazonEC2 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 `SpotFleetTaggingRole` gérée par AmazonEC2 à votre `SpotFleetTaggingRole` rôle AmazonEC2, 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, veuillez consulter [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.

```
{
  "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.

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

AWS Batch Event Stream 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 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 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, veuillez consulter [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](#)
- [Utilisation des notifications AWS utilisateur avec AWS Batch](#)
- [AWS Batch les emplois en tant que EventBridge cibles](#)
- [Tutoriel : Écouter AWS Batch EventBridge](#)

- [Tutoriel : envoi d'alertes Amazon Simple Notification Service en cas d'échec d'un Job](#)

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 de modification de l'état de tâche

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 des tâches](#).

Note

Les événements ne sont pas créés pour la soumission initiale de la tâche.

Exemple 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": [],
  },
  "tags": {
    "resourceArn": "arn:aws:batch:us-
east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
  },
  "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 des [exemples de messages de file d'attente de tâches bloquées](#). La même raison est également disponible dans le `statusReason` champ de l'action [DescribeJobs](#) API.

Exemple 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 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": []
    },
    "tags": {
      "resourceArn": "arn:aws:batch:us-
east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
    },
    "propagateTags": false,
    "platformCapabilities": []
```

```
}  
}
```

Utilisation des 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 relatives à des événements via plusieurs canaux, notamment des e-mails, des notifications de chat [AWS Chatbot](#) ou des notifications push [AWS Console Mobile Application](#). 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 Spot Amazon EC2 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](#).

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/](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 plus d'informations, consultez [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 plus d'informations, consultez [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 le mode de traitement du texte d'un événement 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.

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 lorsqu'il 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/](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 plus d'informations, consultez [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 plus d'informations, consultez [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.

Transmission des informations d'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/](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 plus d'informations, consultez [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 plus d'informations, consultez [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 : Écouter AWS Batch 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 avec AWS Batch](#) 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.

Étape 1 : 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 console AWS Lambda 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 Deploy (Déployer).

Étape 2 : Enregistrer une règle d'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/](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 service AWS de votre compte émet un événement, il accède toujours au bus d'événement 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 Sélectionner une file d'attente Amazon SQS du compte AWS actuel à utiliser en tant que file d'attente de lettres mortes et sélectionnez la file d'attente à utiliser dans la liste déroulante.
 - Choisissez Sélectionner une file d'attente Amazon SQS d'un autre compte AWS en tant que file d'attente de lettres mortes et saisissez 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 l'utilisateur Amazon EventBridge.
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 l'utilisateur Amazon EventBridge.
17. Choisissez Suivant.
18. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

Étape 3 : 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/](https://console.aws.amazon.com/batch/).
2. Soumettez une nouvelle tâche AWS Batch. Pour plus d'informations, consultez [Soumettre une offre d'emploi](#).
3. Ouvrez la CloudWatch console à l'[adresse https://console.aws.amazon.com/cloudwatch/](https://console.aws.amazon.com/cloudwatch/).
4. Dans le panneau de navigation, choisissez Logs (Journaux) et sélectionnez le groupe de journaux pour votre fonction Lambda (par exemple, `/aws/lambda/my-fonction`).
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'un Job

Dans ce didacticiel, vous allez configurer une règle d' EventBridge événement qui capture uniquement les événements de travail lorsque le travail a atteint 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 avec AWS Batch](#) décrites pour en créer une.

Étape 1 : 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 partir de l'adresse <https://console.aws.amazon.com/sns/v3/home>.
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).

Étape 2 : 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/](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).

Étape 3 : Tester une 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/](https://console.aws.amazon.com/batch/).
2. Soumettez une nouvelle AWS Batch offre d'emploi. Pour plus d'informations, consultez [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 : Batch Job Queue bloquée

Pour créer une règle d'événement qui surveille le blocage de Batch Job Queue, répétez les étapes de ce didacticiel avec les modifications suivantes :

1. À l'étape 1, *BlockedJobQueue* utilisez-le comme nom de rubrique.
2. À l'étape 2, utilisez le modèle suivant dans l'éditeur JSON :

```
{
  "detail-type": [
    "Batch Job Queue Blocked"
  ],
  "source": [
    "aws.batch"
  ]
}
```

Utilisation CloudWatch des journaux avec AWS Batch

Vous pouvez configurer vos AWS Batch tâches sur les ressources EC2 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 de AWS Batch calcul basés sur les ressources EC2, ajoutez une politique IAM au `ecsInstanceRole` rôle. Pour plus d'informations, consultez [the section called "Ajouter une politique CloudWatch Logs IAM"](#).
- Créez un modèle de lancement Amazon EC2 qui inclut 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 sur les ressources EC2 et Fargate. Pour plus d'informations, consultez [the section called "Utilisation du pilote du journal awslogs"](#).

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 API CloudWatch Logs. 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 ressources Amazon EC2. Lorsque vous créez un environnement informatique à l'aide de l'assistant de AWS Batch première exécution, vous 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 des [AWS CLI commandes](#) ou la [référence](#) de [AWS Batch l'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 (Politiques).
3. Sélectionnez Create policy (Créer une politique).
4. Choisissez JSON, puis entrez la politique suivante :

```
{
  "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 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, sélectionnez 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 politique `CloudWatchECS-Logs`, puis choisissez Attach policy.

Installation et configuration de l' CloudWatch agent

Vous pouvez créer un modèle de lancement Amazon EC2 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 l'utilisateur Amazon EC2.

Vous pouvez également installer l' CloudWatch agent sur une AMI Amazon EC2 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 [Démarrage avec AWS Batch](#).

Note

Les modèles de lancement ne sont pas pris en charge sur AWS Fargate les ressources.

Afficher les CloudWatch 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/](https://console.aws.amazon.com/cloudwatch/).
2. Dans le volet de navigation de gauche, choisissez Logs, puis Log groups.

<input type="checkbox"/>	Log group	Retention	Metric filters
<input type="checkbox"/>	/aws/batch/job	Never expire	-

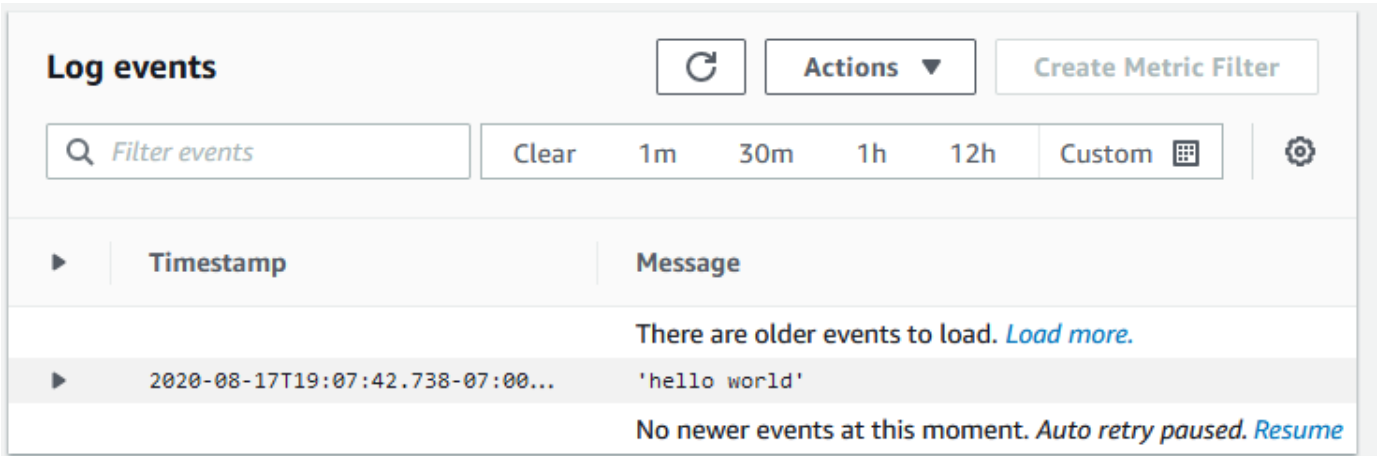
3. Choisissez un groupe de journaux à afficher.



<input type="checkbox"/>	Log stream	Last event time
<input type="checkbox"/>	Test-jd/default/6622fe43-b2a3-4805-a0a6-3828329cc32b	2020-08-18T19:50:19.311Z
<input type="checkbox"/>	first-run-job-definition/default/86ed75ac-4f3f-4044-8fb0-dfd9c85ae6b2	2020-08-18T02:07:42.738Z
<input type="checkbox"/>	Test-jd/default/48f4a9dd-be07-4b43-8696-f0995eefe28b	2020-08-14T00:18:19.395Z
<input type="checkbox"/>	first-run-job-definition/default/d7d5ccf4-a0a0-44f1-bf36-35f2b3632912	2020-08-13T22:39:06.936Z
<input type="checkbox"/>	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**

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. <i>Auto retry paused.</i> Resume

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 une Fluent Bit image contenant un plugin AWS permettant de surveiller les tâches AWS Batch Amazon EKS dans CloudWatch Logs. Fluent Bit est un processeur de journal et un redirecteur open source à la fois Docker et compatible. Kubernetes Nous vous recommandons de l'utiliser Fluent Bit comme routeur de journalisation, car il consomme moins de ressources que Fluentd. Pour plus d'informations, voir [Utilisation de l'image AWS pour Fluent Bit](#).

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](#).

Installation AWS pour Fluent Bit

Pour obtenir des instructions sur l'installation AWS Fluent Bit et la création des CloudWatch groupes, reportez-vous aux [sections Configuration Fluent Bit](#) ou [Démarrage rapide avec l' CloudWatchagent et Fluent Bit](#).

Tip

N'oubliez pas que cela Fluent Bit utilise 0,5 processeur et 100 Mo de mémoire sur AWS Batch les nœuds. Cela réduit la capacité totale disponible pour les AWS Batch emplois. Tenez-en compte lorsque vous déterminez la taille de vos tâches.

Activez Fluent Bit pour les AWS Batch nœuds

Pour garantir l' DaemonSet exécution de la Fluent Bit journalisation sur les nœuds AWS Batch gérés, modifiez les Fluent Bit DaemonSet tolérances :

```
tolerations:  
- key: "batch.amazonaws.com/batch-node"  
  operator: "Exists"
```

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 de calcul 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](#)

Activer Container Insights

Vous pouvez activer Container Insights pour les environnements AWS Batch informatiques.

1. Ouvrez la [AWS Batch console](#).
2. Choisissez Compute Environments.
3. Choisissez l'environnement informatique de votre choix.
4. Pour Container Insights, activez Container Insights pour le calcul environnement.

Tip

Vous pouvez sélectionner un intervalle par défaut pour agréger les mesures ou créer un intervalle personnalisé intervalle.

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**— La 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 utilisant le mode réseau awsvpc ou le mode réseau bridge.
- **NetworkTxBytes**— Le nombre d'octets transmis. Cette métrique n'est disponible que pour les conteneurs dans les tâches utilisant le mode réseau awsvpc ou le mode réseau bridge.
- **StorageReadBytes**— Le nombre d'octets lus depuis le stockage.
- **StorageWriteBytes**— Le nombre d'octets écrits dans le stockage.

Journalisation des AWS Batch appels d'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 de la console AWS Batch et les appels de code vers les opérations d'API AWS Batch. 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](#).

Informations AWS Batch 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 afficher, rechercher et télécharger les événements récents dans votre compte AWS. Pour plus d'informations, consultez la section [Affichage des événements avec l'historique des CloudTrail événements](#).

Pour enregistrer en continu les événements dans votre compte AWS, y compris les événements d'AWS Batch, créez un journal d'activité. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal de suivi dans la console, il s'applique à toutes les régions AWS. Le journal d'activité consigne les événements de toutes les Régions dans la partition AWS et livre les fichiers journaux dans le compartiment Amazon S3 de votre choix. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour en savoir plus, consultez les ressources suivantes :

- [Présentation de la création d'un journal d'activité](#)
- [CloudTrail Services et intégrations pris en charge](#)
- [Configuration des notifications Amazon SNS pour CloudTrail](#)

- [Réception de fichiers CloudTrail journaux de plusieurs régions](#) et [réception de fichiers CloudTrail journaux de plusieurs comptes](#)

Toutes les AWS Batch actions sont enregistrées CloudTrail et documentées sur le site <https://docs.aws.amazon.com/batch/latest/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 effectuée par un autre service AWS.

Pour plus d'informations, consultez la section [Élément userIdentity CloudTrail](#) .

Présentation des entrées des fichiers journaux AWS Batch

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"
}
```


Création d'un cloud privé virtuel

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

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 de calcul géré AWS Batch capable de lancer des ressources de calcul dans votre nouveau VPC. Pour plus d'informations, consultez [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 tâches AWS Batch mappée à votre nouvel environnement informatique. Pour plus d'informations, consultez [Création d'une file d'attente de tâches](#).
- Créez une définition de tâche avec laquelle exécuter vos tâches. Pour plus d'informations, consultez [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 nouveaux sous-réseaux. Pour plus d'informations, consultez [Soumettre une offre d'emploi](#).

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](#)
- [Accès AWS Batch via un point de terminaison d'interface](#)
- [Validation de conformité pour AWS Batch](#)
- [Sécurité de l'infrastructure dans 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 des accès à l'aide de politiques](#)
- [Comment AWS Batch fonctionne avec IAM](#)
- [AWS Batch rôle d'exécution IAM](#)
- [Exemples de politiques basées sur l'identité pour AWS Batch](#)
- [Prévention du cas de figure de l'adjoint désorienté entre services](#)
- [Résolution des problèmes AWS Batch d'identité et d'accès](#)
- [Utilisation de rôles liés à un service 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 du travail que vous effectuez. AWS Batch

Utilisateur du service : si vous utilisez le AWS Batch service pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de nouvelles AWS Batch fonctionnalités pour effectuer votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité dans AWS Batch, consultez [Résolution des problèmes AWS Batch d'identité et d'accès](#).

Administrateur du service — Si vous êtes responsable des AWS Batch ressources de votre entreprise, vous avez probablement un accès complet à AWS Batch. C'est à vous de déterminer les AWS Batch fonctionnalités et les ressources auxquelles les utilisateurs de votre service doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier

les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser IAM avec AWS Batch, voir [Comment AWS Batch fonctionne avec IAM](#).

Administrateur IAM – Si vous êtes un administrateur IAM, vous souhaitez peut-être en savoir plus sur la façon d'écrire des politiques pour gérer l'accès à AWS Batch. Pour consulter des exemples de politiques AWS Batch basées sur l'identité que vous pouvez utiliser dans IAM, consultez. [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 (connecté à AWS) en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez [Comment vous connecter à votre compte Compte AWS dans](#) le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer vous-même les demandes, consultez la section [Signature des demandes AWS d'API](#) dans le guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour en savoir plus, consultez [Authentification multifactorielle](#) dans le Guide de l'utilisateur AWS IAM Identity Center

et [Utilisation de l'authentification multifactorielle \(MFA\) dans l'interface AWS](#) dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur root, consultez [Tâches nécessitant des informations d'identification d'utilisateur root](#) dans le Guide de l'utilisateur IAM.

Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris ceux qui ont besoin d'un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder à l'aide Services AWS d'informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède à l'aide des informations d'identification fournies Services AWS par le biais d'une source d'identité. Lorsque des identités fédérées y accèdent Comptes AWS, elles 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. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité afin de les utiliser dans toutes vos applications Comptes AWS et applications. Pour obtenir des informations sur IAM Identity Center, 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 au sein de votre Compte AWS qui possède des autorisations spécifiques pour une seule personne ou application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme tels que les clés d'accès.

Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, consultez [Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification](#) dans le Guide de l'utilisateur IAM.

Un [groupe IAM](#) est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez avoir un groupe nommé IAMAdmins et accorder à ce groupe les autorisations d'administrer des ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, consultez [Quand créer un utilisateur IAM \(au lieu d'un rôle\)](#) dans le Guide de l'utilisateur IAM.

Rôles IAM

Un [rôle IAM](#) est une identité au sein de votre Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Vous pouvez assumer temporairement un rôle IAM dans le en AWS Management Console [changeant de rôle](#). Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez [Utilisation de rôles IAM](#) dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré – Pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez [Création d'un rôle pour un fournisseur d'identité tiers \(fédération\)](#) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez la rubrique [Jeux d'autorisations](#) dans le Guide de l'utilisateur AWS IAM Identity Center .
- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.

- **Accès intercompte** : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez [Différence entre les rôles IAM et les politiques basées sur les ressources](#) dans le Guide de l'utilisateur IAM.
- **Accès multiservices** — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
- **Sessions d'accès direct (FAS)** : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez [Sessions de transmission d'accès](#).
- **Rôle de service** : il s'agit d'un [rôle IAM](#) attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du 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.
- **Rôle lié à un service** — 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 répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.
- **Applications exécutées sur Amazon EC2** : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une instance EC2 et qui envoient des demandes d'API. AWS CLI AWS Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un AWS rôle à une instance EC2 et le mettre à la disposition de toutes ses applications, vous devez créer un profil d'instance attaché

à l'instance. Un profil d'instance contient le rôle et permet aux programmes qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez [Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2](#) dans le Guide de l'utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, consultez [Quand créer un rôle IAM \(au lieu d'un utilisateur\)](#) dans le Guide de l'utilisateur IAM.

Gestion des 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 est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez [Vue d'ensemble des politiques JSON](#) dans le Guide de l'utilisateur IAM.

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.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

Politiques basées sur l'identité

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 [Création de politiques IAM](#) dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique 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. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment. 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. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez [spécifier un principal](#) dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

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.

Listes de contrôle d'accès (ACL)

Les listes de contrôle d'accès (ACL) vérifie quels principaux (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et Amazon VPC sont des exemples de services qui prennent en charge les ACL. AWS WAF Pour en savoir plus sur les listes de contrôle d'accès, consultez [Vue d'ensemble des listes de contrôle d'accès \(ACL\)](#) dans le Guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- **Limite d'autorisations** : une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez [Limites d'autorisations pour des entités IAM](#) dans le Guide de l'utilisateur IAM.
- **Politiques de contrôle des services (SCP)** — Les SCP sont des politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans AWS Organizations. AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée les comptes AWS multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer les politiques de contrôle des services (SCP) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités figurant dans les comptes des membres, y compris chacune Utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les organisations et les SCP, consultez [Fonctionnement des SCP](#) dans le Guide de l'utilisateur AWS Organizations .
- **Politiques de séance** : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations, consultez [politiques de séance](#) 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

Fonction 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 d'une politique	Oui
ACL	Non
ABAC (étiquettes dans les politiques)	Oui
Informations d'identification temporaires	Oui
Autorisations de principal	Oui
Fonctions 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 [Création de politiques IAM](#) dans le Guide de l'utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité car celle-ci s'applique à l'utilisateur ou au rôle auquel elle est attachée. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez [Références des éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.

Exemples de stratégies 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. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations

nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une stratégie afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des AWS Batch actions, voir [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. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son [Amazon Resource Name \(ARN\)](#). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

Pour consulter la liste des types de AWS Batch ressources et de leurs ARN, consultez la section [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](#).

Pour consulter des exemples de politiques AWS Batch basées sur l'identité, consultez. [Exemples de politiques basées sur l'identité pour 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` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des [opérateurs de condition](#), tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération AND logique. Si vous spécifiez

plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une OR opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez [Éléments d'une politique IAM : variables et identifications](#) dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. 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](#).

Pour consulter des exemples de politiques AWS Batch basées sur l'identité, consultez. [Exemples de politiques basées sur l'identité pour AWS Batch](#)

Contrôle d'accès basé sur les attributs (ABAC) avec AWS Batch

Prend en charge ABAC (étiquettes dans les politiques)	Oui
---	-----

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises aux entités IAM (utilisateurs ou rôles) et à de nombreuses AWS ressources. L'étiquetage des entités et des ressources est la première étape d'ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l'identification du principal correspond à celle de la ressource à laquelle il tente d'accéder.

L'ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

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 l'ABAC, consultez [Qu'est-ce que le contrôle d'accès basé sur les attributs \(ABAC\) ?](#) 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.

Utilisation d'informations d'identification temporaires avec AWS Batch

Prend en charge les informations d'identification temporaires	Oui
---	-----

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation [d'IAM](#) dans le guide de l'utilisateur d'IAM.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, consultez [Changement de rôle \(console\)](#) dans le Guide de l'utilisateur IAM.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. 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 [Informations d'identification de sécurité temporaires dans IAM](#).

Autorisations de principal interservices pour AWS Batch

Prend en charge les sessions d'accès direct (FAS)	Oui
---	-----

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez [Transmission des sessions d'accès](#).

Rôles de service pour AWS Batch

Prend en charge les fonctions du service	Oui
--	-----

Une fonction de service est un [rôle IAM](#) qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. 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 fonctions du 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 répertoire 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.

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 IAM d'exécution 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
```

```
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
    ],
    "Resource": "*"
}
]
```

Vous pouvez utiliser la procédure suivante pour vérifier que votre compte possède déjà le rôle d'exécution et pour associer la politique IAM gérée, si nécessaire.

Pour vérifier le rôle **ecsTaskExecutionRole** dans la console IAM

1. Ouvrez la console IAM à l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le panneau de navigation, sélectionnez Rôles.
3. Recherchez dans la liste des rôles `ecsTaskExecutionRole`. Si vous ne trouvez pas le rôle, consultez [Création du rôle IAM d'exécution](#). Si vous avez trouvé le rôle, choisissez-le pour afficher les politiques associées.
4. Dans l'onglet Autorisations, vérifiez que la politique `TaskExecution RolePolicy` gérée par AmazonECS 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 AmazonECS TaskExecution RolePolicy.
 - c. Cochez la case située à gauche de la politique d'AmazonECS et choisissez Joindre les TaskExecution RolePolicy 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
    "Sid": "",
    "Effect": "Allow",
    "Principal": {
      "Service": "ecs-tasks.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  }
]
```

Création du rôle IAM d'exécution

Si votre compte ne possède pas encore de rôle d'exécution, suivez les étapes ci-dessous pour créer le rôle.

Pour créer le rôle **ecsTaskExecutionRole** IAM

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 EC2. Choisissez ensuite à nouveau EC2.
6. Choisissez Suivant.
7. Pour les politiques d'autorisations, recherchez AmazonECS TaskExecution RolePolicy.
8. Cochez la case située à gauche de la TaskExecution RolePolicy politique d'AmazonECS, puis choisissez Next.
9. Dans Nom du rôle, entrez `ecsTaskExecutionRole` puis choisissez Créer un rôle.

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. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l'API AWS Management Console, AWS Command Line Interface (AWS CLI) ou de l'API AWS. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Pour 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 dans l'onglet JSON](#) 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 des ARN 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.
- Accorder 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 de IAM pour appliquer des autorisations, consultez [politiques et autorisations dans IAM](#) dans le Guide de l'utilisateur IAM.
- Utiliser 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 AWS CloudFormation. Pour plus d'informations, consultez [Conditions pour éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.

- Utilisez IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : IAM Access Analyzer 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 politique 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 le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez [Configuration de l'accès aux API protégé par 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 stratégie 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 stratégie.

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:ListGroupsWithUser",
        "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": "*"
    }
  ]
}
```


Prévention du cas de figure de l'adjoint désorienté entre services

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. En AWS, l'usurpation d'identité interservices peut entraîner un problème de confusion chez les adjoints. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé et ses autorisations utilisées pour agir sur les ressources d'un autre client auxquelles on ne serait pas autorisé d'accéder autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés de contexte de condition [aws:SourceAccount](#) globale [aws:SourceArn](#) et les clés contextuelles dans les politiques de ressources afin de limiter les autorisations qui AWS 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 1 : 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.

```
{
  "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 2 : 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.

```
{
  "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/*"
      ]
    }
  }
}
```

Résolution des problèmes AWS Batch d'identité et d'accès

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 vos 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, un utilisateur doit 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 ne dispose pas des autorisations nécessaires pour transférer 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 (ACL), vous pouvez utiliser ces politiques pour donner l'accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si ces fonctionnalités sont prises en charge par AWS Batch, consultez [Comment AWS Batch fonctionne avec IAM](#).
- Pour savoir comment fournir l'accès à vos ressources sur celles des 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 des Comptes AWS, consultez la section [Fournir un accès à des ressources des 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 découvrir quelle est la différence entre l'utilisation des rôles et l'utilisation des politiques basées sur les ressources pour l'accès entre comptes, consultez [Différence entre les rôles IAM et les politiques basées sur les ressources](#) dans le Guide de l'utilisateur IAM.

Utilisation de 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 un service. Les rôles liés au service sont prédéfinis par AWS Batch et incluent toutes les autorisations dont le service a besoin pour appeler d'autres services AWS 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 d'approbation 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 à AWS Batch de 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 [the section called “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. Vos ressources AWS Batch sont ainsi protégées, car vous ne pouvez pas involontairement supprimer l'autorisation d'accéder aux ressources.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés à un service, consultez [Services AWS qui fonctionnent avec IAM](#) et recherchez les services où Oui figure dans la colonne Rôle lié à un service. Choisissez un Yes (oui) ayant un lien permettant de consulter les détails du rôle pour ce service.

Autorisations de rôle liées à un service pour AWS Batch

AWS Batch utilise le rôle lié au service nommé. AWSServiceRoleForBatch Le AWSServiceRoleForBatchrôle permet AWS Batch de créer et de gérer AWS des ressources en votre nom.

Le rôle AWSServiceRoleForBatchlié au service fait confiance au principal du batch.amazonaws.com service pour assumer le rôle.

La stratégie IAM nommée [BatchServiceRolePolicy](#) permet d' AWS Batch effectuer 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.

Vous devez configurer les autorisations de manière à permettre à une entité IAM (comme un utilisateur, un groupe ou un rôle) de créer, modifier ou supprimer un rôle lié à un service. Pour plus d'informations, consultez [Service-Linked Role Permissions \(autorisations du rôle lié à un service\)](#) dans le IAM User Guide (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 utilisez `CreateComputeEnvironment` l'API AWS Management Console AWS CLI, le ou l' AWS API, et que vous ne spécifiez pas de valeur pour le `serviceRole` paramètre, 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. De plus, 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 `AWSServiceRoleForBatch` rôle dans votre compte. Pour plus d'informations, consultez [A New Role Appeared in My IAM Account](#) (Un nouveau rôle est apparu dans mon compte IAM).

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 `CreateComputeEnvironment` AWS Batch créez à nouveau le rôle lié au service pour vous.

Modification d'un rôle lié à un service pour AWS Batch

Avec AWS Batch, vous ne pouvez pas modifier le rôle `AWSServiceRoleForBatch` lié au service. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas changer le nom du rôle, car plusieurs entités peuvent faire référence au 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 AWSServiceRoleForBatch 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"}}
}
```

Supprimer un rôle lié à un service pour AWS Batch

Si vous n'avez plus besoin d'utiliser une fonctionnalité ou un service nécessitant un rôle lié à un service, nous vous recommandons de supprimer ce rôle. De cette façon, vous n'avez pas d'entité inutilisée qui n'est pas activement surveillée ou maintenue. Cependant, vous devez nettoyer les ressources de votre rôle lié à un service avant de pouvoir les supprimer manuellement.

Pour autoriser une entité IAM à supprimer le rôle lié à un AWSServiceRoleForBatch 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"}}
}
```


Nettoyer 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 AWSServiceRoleForBatch 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 AWSServiceRoleForBatch 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 AWSServiceRoleForBatch un service

Vous devez supprimer tous les environnements AWS Batch informatiques qui utilisent le AWSServiceRoleForBatch rôle dans toutes les AWS régions avant de pouvoir supprimer le AWSServiceRoleForBatch 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 l'adresse <https://console.aws.amazon.com/iam/>.
2. Dans le panneau de navigation de la console IAM, sélectionnez Rôles (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 soumettre une demande de suppression pour un rôle lié à un service, appelez [DeleteServiceLinkedRole](#). Dans la demande, spécifiez le nom du AWSServiceRoleForBatch 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 AWS Batch liés à un service

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](#).

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. De cette façon, 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 `BatchServiceRolePolicy` gérée est utilisée par le rôle lié au [`AWSBatchServiceRoleForBatch`](#) service. Cela permet AWS Batch d'effectuer des actions en votre nom. Vous ne pouvez pas attacher cette politique à vos entités IAM. Pour plus d'informations, consultez [Utilisation de 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSBatchPolicyStatement1",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstanceAttribute",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeImages",
        "ec2:DescribeImageAttribute",
        "ec2:DescribeSpotInstanceRequests",
        "ec2:DescribeSpotFleetInstances",
        "ec2:DescribeSpotFleetRequests",
        "ec2:DescribeSpotPriceHistory",
        "ec2:DescribeSpotFleetRequestHistory",
        "ec2:DescribeVpcClassicLink",
        "ec2:DescribeLaunchTemplateVersions",
        "ec2:RequestSpotFleet",
        "autoscaling:DescribeAccountLimits",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeLaunchConfigurations",
        "autoscaling:DescribeAutoScalingInstances",
        "autoscaling:DescribeScalingActivities",
        "eks:DescribeCluster",
        "ecs:DescribeClusters",
        "ecs:DescribeContainerInstances",
        "ecs:DescribeTaskDefinition",
```

```

        "ecs:DescribeTasks",
        "ecs:ListClusters",
        "ecs:ListContainerInstances",
        "ecs:ListTaskDefinitionFamilies",
        "ecs:ListTaskDefinitions",
        "ecs:ListTasks",
        "ecs:DeregisterTaskDefinition",
        "ecs:TagResource",
        "ecs:ListAccountSettings",
        "logs:DescribeLogGroups",
        "iam:GetInstanceProfile",
        "iam:GetRole"
    ],
    "Resource": "*"
},
{
    "Sid": "AWSBatchPolicyStatement2",
    "Effect": "Allow",
    "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/batch/job*"
},
{
    "Sid": "AWSBatchPolicyStatement3",
    "Effect": "Allow",
    "Action": [
        "logs:PutLogEvents"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/batch/job*:log-stream:*"
},
{
    "Sid": "AWSBatchPolicyStatement4",
    "Effect": "Allow",
    "Action": [
        "autoscaling:CreateOrUpdateTags"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "aws:RequestTag/AWSBatchServiceTag": "false"
        }
    }
}
}

```



```
    },
    {
      "Sid": "AWSBatchPolicyStatement5",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "ec2.amazonaws.com",
            "ec2.amazonaws.com.cn",
            "ecs-tasks.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "AWSBatchPolicyStatement6",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": [
            "spot.amazonaws.com",
            "spotfleet.amazonaws.com",
            "autoscaling.amazonaws.com",
            "ecs.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "AWSBatchPolicyStatement7",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateLaunchTemplate"
      ],
      "Resource": "*",
      "Condition": {
        "Null": {
          "aws:RequestTag/AWSBatchServiceTag": "false"
        }
      }
    }
  ]
}
```

```

    }
  },
  {
    "Sid": "AWSBatchPolicyStatement8",
    "Effect": "Allow",
    "Action": [
      "ec2:TerminateInstances",
      "ec2:CancelSpotFleetRequests",
      "ec2:ModifySpotFleetRequest",
      "ec2>DeleteLaunchTemplate"
    ],
    "Resource": "*",
    "Condition": {
      "Null": {
        "aws:ResourceTag/AWSBatchServiceTag": "false"
      }
    }
  },
  {
    "Sid": "AWSBatchPolicyStatement9",
    "Effect": "Allow",
    "Action": [
      "autoscaling:CreateLaunchConfiguration",
      "autoscaling>DeleteLaunchConfiguration"
    ],
    "Resource":
"arn:aws:autoscaling:*:*:launchConfiguration:*:launchConfigurationName/AWSBatch*"
  },
  {
    "Sid": "AWSBatchPolicyStatement10",
    "Effect": "Allow",
    "Action": [
      "autoscaling:CreateAutoScalingGroup",
      "autoscaling:UpdateAutoScalingGroup",
      "autoscaling:SetDesiredCapacity",
      "autoscaling>DeleteAutoScalingGroup",
      "autoscaling:SuspendProcesses",
      "autoscaling:PutNotificationConfiguration",
      "autoscaling:TerminateInstanceInAutoScalingGroup"
    ],
    "Resource":
"arn:aws:autoscaling:*:*:autoScalingGroup:*:autoScalingGroupName/AWSBatch*"
  },

```

```
{
  "Sid": "AWSBatchPolicyStatement11",
  "Effect": "Allow",
  "Action": [
    "ecs:DeleteCluster",
    "ecs:DeregisterContainerInstance",
    "ecs:RunTask",
    "ecs:StartTask",
    "ecs:StopTask"
  ],
  "Resource": "arn:aws:ecs:*:*:cluster/AWSBatch*"
},
{
  "Sid": "AWSBatchPolicyStatement12",
  "Effect": "Allow",
  "Action": [
    "ecs:RunTask",
    "ecs:StartTask",
    "ecs:StopTask"
  ],
  "Resource": "arn:aws:ecs:*:*:task-definition/*"
},
{
  "Sid": "AWSBatchPolicyStatement13",
  "Effect": "Allow",
  "Action": [
    "ecs:StopTask"
  ],
  "Resource": "arn:aws:ecs:*:*:task/*/*"
},
{
  "Sid": "AWSBatchPolicyStatement14",
  "Effect": "Allow",
  "Action": [
    "ecs:CreateCluster",
    "ecs:RegisterTaskDefinition"
  ],
  "Resource": "*",
  "Condition": {
    "Null": {
      "aws:RequestTag/AWSBatchServiceTag": "false"
    }
  }
},
```

```

{
  "Sid": "AWSBatchPolicyStatement15",
  "Effect": "Allow",
  "Action": "ec2:RunInstances",
  "Resource": [
    "arn:aws:ec2:*:*:image/*",
    "arn:aws:ec2:*:*:snapshot/*",
    "arn:aws:ec2:*:*:subnet/*",
    "arn:aws:ec2:*:*:network-interface/*",
    "arn:aws:ec2:*:*:security-group/*",
    "arn:aws:ec2:*:*:volume/*",
    "arn:aws:ec2:*:*:key-pair/*",
    "arn:aws:ec2:*:*:launch-template/*",
    "arn:aws:ec2:*:*:placement-group/*",
    "arn:aws:ec2:*:*:capacity-reservation/*",
    "arn:aws:ec2:*:*:elastic-gpu/*",
    "arn:aws:elastic-inference:*:*:elastic-inference-accelerator/*",
    "arn:aws:resource-groups:*:*:group/*"
  ]
},
{
  "Sid": "AWSBatchPolicyStatement16",
  "Effect": "Allow",
  "Action": "ec2:RunInstances",
  "Resource": "arn:aws:ec2:*:*:instance/*",
  "Condition": {
    "Null": {
      "aws:RequestTag/AWSBatchServiceTag": "false"
    }
  }
},
{
  "Sid": "AWSBatchPolicyStatement17",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags"
  ],
  "Resource": [
    "*"
  ],
  "Condition": {
    "StringEquals": {
      "ec2:CreateAction": [
        "RunInstances",

```

```
        "CreateLaunchTemplate",  
        "RequestSpotFleet"  
    ]  
  }  
}  
]  
}
```

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 au conseil de sécurité standard consistant à accorder le moindre privilège, la politique `AWSBatchServiceRole` gérée 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 aux services est préférable pour une *less-error-prone* expérience gérée plus étendue et améliorée.

- `autoscaling`— Permet à AWS Batch de 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 à 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 à 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 à 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 à AWS Batch de créer et de gérer des groupes de journaux et des flux de journaux pour les tâches AWS Batch.

```
{  
  "Version": "2012-10-17",
```

```
"Statement": [  
  {  
    "Sid": "AWSBatchPolicyStatement1",  
    "Effect": "Allow",  
    "Action": [  
      "ec2:DescribeAccountAttributes",  
      "ec2:DescribeInstances",  
      "ec2:DescribeInstanceStatus",  
      "ec2:DescribeInstanceAttribute",  
      "ec2:DescribeSubnets",  
      "ec2:DescribeSecurityGroups",  
      "ec2:DescribeKeyPairs",  
      "ec2:DescribeImages",  
      "ec2:DescribeImageAttribute",  
      "ec2:DescribeSpotInstanceRequests",  
      "ec2:DescribeSpotFleetInstances",  
      "ec2:DescribeSpotFleetRequests",  
      "ec2:DescribeSpotPriceHistory",  
      "ec2:DescribeSpotFleetRequestHistory",  
      "ec2:DescribeVpcClassicLink",  
      "ec2:DescribeLaunchTemplateVersions",  
      "ec2:CreateLaunchTemplate",  
      "ec2>DeleteLaunchTemplate",  
      "ec2:RequestSpotFleet",  
      "ec2:CancelSpotFleetRequests",  
      "ec2:ModifySpotFleetRequest",  
      "ec2:TerminateInstances",  
      "ec2:RunInstances",  
      "autoscaling:DescribeAccountLimits",  
      "autoscaling:DescribeAutoScalingGroups",  
      "autoscaling:DescribeLaunchConfigurations",  
      "autoscaling:DescribeAutoScalingInstances",  
      "autoscaling:DescribeScalingActivities",  
      "autoscaling:CreateLaunchConfiguration",  
      "autoscaling:CreateAutoScalingGroup",  
      "autoscaling:UpdateAutoScalingGroup",  
      "autoscaling:SetDesiredCapacity",  
      "autoscaling>DeleteLaunchConfiguration",  
      "autoscaling>DeleteAutoScalingGroup",  
      "autoscaling:CreateOrUpdateTags",  
      "autoscaling:SuspendProcesses",  
      "autoscaling:PutNotificationConfiguration",  
      "autoscaling:TerminateInstanceInAutoScalingGroup",  
      "ecs:DescribeClusters",
```

```

        "ecs:DescribeContainerInstances",
        "ecs:DescribeTaskDefinition",
        "ecs:DescribeTasks",
        "ecs:ListAccountSettings",
        "ecs:ListClusters",
        "ecs:ListContainerInstances",
        "ecs:ListTaskDefinitionFamilies",
        "ecs:ListTaskDefinitions",
        "ecs:ListTasks",
        "ecs:CreateCluster",
        "ecs>DeleteCluster",
        "ecs:RegisterTaskDefinition",
        "ecs:DeregisterTaskDefinition",
        "ecs:RunTask",
        "ecs:StartTask",
        "ecs:StopTask",
        "ecs:UpdateContainerAgent",
        "ecs:DeregisterContainerInstance",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "iam:GetInstanceProfile",
        "iam:GetRole"
    ],
    "Resource": "*"
},
{
    "Sid": "AWSBatchPolicyStatement2",
    "Effect": "Allow",
    "Action": "ecs:TagResource",
    "Resource": [
        "arn:aws:ecs:*:*:task/*_Batch_*"
    ]
},
{
    "Sid": "AWSBatchPolicyStatement3",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": [
        "*"
    ],
    "Condition": {
        "StringEquals": {

```

```
        "iam:PassedToService": [
            "ec2.amazonaws.com",
            "ec2.amazonaws.com.cn",
            "ecs-tasks.amazonaws.com"
        ]
    }
},
{
    "Sid": "AWSBatchPolicyStatement4",
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "iam:AWSServiceName": [
                "spot.amazonaws.com",
                "spotfleet.amazonaws.com",
                "autoscaling.amazonaws.com",
                "ecs.amazonaws.com"
            ]
        }
    }
},
{
    "Sid": "AWSBatchPolicyStatement5",
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": [
        "*"
    ],
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction": "RunInstances"
        }
    }
}
]
```


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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:*",
        "cloudwatch:GetMetricStatistics",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeVpcs",
        "ec2:DescribeImages",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions",
        "ecs:DescribeClusters",
        "ecs:Describe*",
        "ecs:List*",
        "eks:DescribeCluster",
        "eks:ListClusters",
        "logs:Describe*",
        "logs:Get*",
        "logs:TestMetricFilter",
        "logs:FilterLogEvents",
        "iam:ListInstanceProfiles",
        "iam:ListRoles"
      ],
      "Resource": "*"
    }
  ]
}
```

```
"Effect": "Allow",
"Action": [
  "iam:PassRole"
],
"Resource": [
  "arn:aws:iam::*:role/AWSBatchServiceRole",
  "arn:aws:iam::*:role/service-role/AWSBatchServiceRole",
  "arn:aws:iam::*:role/ecsInstanceRole",
  "arn:aws:iam::*:instance-profile/ecsInstanceRole",
  "arn:aws:iam::*:role/iaws-ec2-spot-fleet-role",
  "arn:aws:iam::*:role/aws-ec2-spot-fleet-role",
  "arn:aws:iam::*:role/AWSBatchJobRole*"
]
},
{
  "Effect": "Allow",
  "Action": [
    "iam:CreateServiceLinkedRole"
  ],
  "Resource": "arn:aws:iam::*:role/*Batch*",
  "Condition": {
    "StringEquals": {
      "iam:AWSServiceName": "batch.amazonaws.com"
    }
  }
}
]
```

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.

Modification	Description	Date
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 2022
AWSBatchServiceRole politique ajoutée	Mis à jour pour ajouter des identifiants de relevé, accorder AWS Batch des autorisations à <code>ec2:DescribeSpotFleetRequestHistory</code> et <code>autoscaling:DescribeScalingActivities</code> .	5 décembre 2022
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 AWS Resource Groups. 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

Modification	Description	Date
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
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

Accès AWS Batch via un point de terminaison d'interface

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 AWS 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 le nom de service suivant :

```
com.amazonaws.region.batch
```

Par exemple :

```
com.amazonaws.us-east-2.batch
```

Dans la aws-cn partition, le format est différent :

```
cn.com.amazonaws.region.batch
```

Par exemple :

```
cn.com.amazonaws.cn-northwest-1.batch
```

Si vous activez le DNS privé pour le point de terminaison de l'interface, vous pouvez envoyer des demandes d'API AWS Batch en utilisant son nom DNS régional par défaut. Par exemple, `batch.us-east-1.amazonaws.com`.

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 AWS Batch depuis votre VPC, associez une politique de point de terminaison personnalisée au point de terminaison de l'interface.

Une politique de point de terminaison spécifie les informations suivantes :

- Les principaux qui peuvent effectuer des actions (Comptes AWS utilisateurs 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. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- [Guides de démarrage rapide sur la sécurité et la conformité](#) : ces guides de déploiement abordent les considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de base axés sur AWS la sécurité et la conformité.
- [Architecture axée sur la sécurité et la conformité HIPAA sur Amazon Web Services](#) : ce livre blanc décrit comment les entreprises peuvent créer des applications AWS conformes à la loi HIPAA.

Note

Tous ne Services AWS sont pas éligibles à la loi HIPAA. Pour plus d'informations, consultez le [HIPAA Eligible Services Reference](#).

- AWS Ressources de <https://aws.amazon.com/compliance/resources/> de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- [AWS Guides de conformité destinés aux clients](#) — Comprenez le modèle de responsabilité partagée sous l'angle de la conformité. Les guides résument les meilleures pratiques en matière

de sécurisation Services AWS et décrivent les directives relatives aux contrôles de sécurité dans plusieurs cadres (notamment le National Institute of Standards and Technology (NIST), le Payment Card Industry Security Standards Council (PCI) et l'Organisation internationale de normalisation (ISO)).

- [Évaluation des ressources à l'aide des règles](#) du guide du AWS Config développeur : le AWS Config service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- [AWS Security Hub](#)— Cela Service AWS fournit une vue complète de votre état de sécurité interne AWS. Security Hub utilise des contrôles de sécurité pour évaluer vos ressources AWS et vérifier votre conformité par rapport aux normes et aux bonnes pratiques du secteur de la sécurité. Pour obtenir la liste des services et des contrôles pris en charge, consultez [Référence des contrôles Security Hub](#).
- [Amazon GuardDuty](#) — Cela Service AWS détecte les menaces potentielles qui pèsent sur vos charges de travail Comptes AWS, vos conteneurs et vos données en surveillant votre environnement pour détecter toute activité suspecte et malveillante. GuardDuty peut vous aider à répondre à diverses exigences de conformité, telles que la norme PCI DSS, en répondant aux exigences de détection des intrusions imposées par certains cadres de conformité.
- [AWS Audit Manager](#)— Cela vous Service AWS permet d'auditer en permanence votre AWS utilisation afin de simplifier la gestion des risques et la conformité aux réglementations et aux normes du secteur.

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.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser [AWS Security Token Service](#) (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

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 des VPC spécifiques. En fait, cela isole l'accès réseau à une AWS Batch ressource donnée uniquement du VPC spécifique au sein AWS du réseau.

Balisage de vos ressources AWS Batch

Pour vous aider à gérer vos ressources AWS Batch, vous pouvez attribuer vos propres métadonnées à chaque ressource sous la forme de balises. Cette rubrique décrit les balises et vous explique comment les créer.

Table des matières

- [Principes de base des balises](#)
- [Balisage de vos ressources](#)
- [Restrictions liées aux étiquettes](#)
- [Gestion des étiquettes à l'aide de la console](#)
- [Gestion des balises à l'aide de la CLI ou de l'API](#)

Principes de base des balises

Une balise est une étiquette que vous affectez à une ressource AWS. Chaque balise est constituée d'une clé et d'une valeur facultative que vous définissez.

Les balises vous permettent de classer vos ressources AWS par catégorie, objectif, propriétaire ou environnement, par exemple. 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 services AWS Batch afin de vous aider à 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 pas de signification sémantique pour 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 gérer les balises à l'aide de la AWS Management Console, de l'AWS CLI et de l'API AWS Batch.

Si vous utilisez AWS Identity and Access Management (IAM), vous pouvez contrôler quels utilisateurs de votre compte AWS sont autorisés à créer, modifier ou supprimer des étiquettes.

Balisage de 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 console AWS Batch, vous pouvez appliquer des balises aux nouvelles ressources au moment de leur création ou aux ressources existantes à l'aide de l'onglet Tags (Balises) de la page de ressources correspondante à tout moment.

Si vous utilisez l'API AWS Batch, l'AWS CLI ou un kit AWS SDK, vous pouvez appliquer les balises aux nouvelles ressources à l'aide du paramètre `tags` sur l'action d'API correspondante ou utiliser l'action d'API `TagResource`. Pour plus d'informations, consultez [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 ressources AWS Batch qui peuvent porter des balises, et les ressources qui peuvent porter des balises dès la création.

Prise en charge du balisage pour les ressources AWS Batch

Ressource	Prend en charge les étiquettes	Prend en charge la propagation des étiquettes	Prend en charge le balisage au moment de la création (API AWS Batch, AWS CLI, kit 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	Prend en charge le balisage au moment de la création (API AWS Batch, AWS CLI, kit 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 tâches	Oui	Oui	Oui
AWS Batch définitions de poste	Oui	Non	Oui
Files d'attente de tâches AWS Batch	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 services et ressources AWS, n'oubliez pas que d'autres services peuvent avoir des restrictions concernant les 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:`, ou n'importe quelle combinaison de majuscules ou minuscules comme préfixe pour des clés ou des valeurs, car il est réservé à 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.

Gestion des étiquettes 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.

Ajout de balises sur une ressource individuelle lors de la 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.

Ajout et suppression de balises sur une ressource individuelle

AWS Batch permet d'ajouter ou de supprimer des balises associées à vos clusters directement à partir de 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/](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 si nécessaire.
 - 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.

Gestion des balises à l'aide de la CLI ou de l'API

Utilisez les commandes AWS CLI suivantes ou les opérations d'API AWS Batch pour ajouter, mettre à jour, répertorier et supprimer les balises de vos ressources.

Prise en charge du balisage pour les ressources AWS Batch

Tâche	Action d'API	AWS CLI	AWS Tools for Windows PowerShell
Ajouter ou remplacer une ou plusieurs étiquettes.	TagResource	tag-resource	Ajouter un chauve-souris ResourceTag
Supprimer une ou plusieurs étiquettes.	UntagResource	untag-resource	Supprimer le chauve-souris ResourceTag
Répertorie les balises d'une ressource.	ListTagsForResource	list-tags-for-resource	Get-bat ResourceTag

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	Nouvelle chauve-souris ComputeEnvironment
Création d'une file d'attente de tâches	CreateJobQueue	create-job-queue	Nouvelle chauve-souris JobQueue
Création d'une politique de planification	CreateSchedulingPolicy	create-scheduling-policy	Nouvelle chauve-souris SchedulingPolicy
Enregistrer une définition de tâche	RegisterJobDefinition	register-job-definition	Enregistrer un BAT JobDefinition
Envoi d'une tâche	SubmitJob	soumet-job	Soumets-BatJob

Quotas de service AWS Batch

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 plus d'informations, consultez Files d'attente pour les jobs .	50
Nombre maximal d'environnements informatiques sur Amazon ECS et Amazon EKS. Pour plus d'informations, consultez Environnement de calcul .	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

Selon votre mode d'utilisation AWS Batch, des quotas supplémentaires peuvent s'appliquer. Pour en savoir plus sur les quotas Amazon EC2, consultez la section Quotas de [service Amazon EC2](#) 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](#).

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 AWS Fargate Elastic Kubernetes Service. 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 instances EC2](#) dans le guide de l'utilisateur Amazon EC2

Table des matières

- [AWS Batch](#)
 - [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 SpotFleet TaggingRole gérée par AmazonEC2 à votre rôle Spot Fleet dans AWS Management Console](#)
 - [Associez la politique SpotFleet TaggingRole gérée par Amazon EC2 à votre rôle Spot Fleet à l'aide du 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 informatique supprimé](#)
- [Les nœuds ne rejoignent pas le cluster Amazon EKS](#)
- [AWS Batch sur Amazon EKS, le RUNNABLE 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](#)

AWS Batch

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.

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 des AWS SDK. 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 les AWS SDK, 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 de 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 de AWS Batch calcul.

- 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 les AWS SDK, 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/](https://console.aws.amazon.com/batch/).
2. Dans la barre de navigation, sélectionnez la 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.
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. Dans ce cas, il est 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.

S'il AWS Batch détecte que vous avez une `RUNNABLE` tâche en tête et que vous bloquez la file d'attente, vous recevrez un événement de [blocage de la part d'Amazon CloudWatch Events](#) indiquant le motif du blocage. La même raison est également mise à jour dans le `statusReason` champ dans le cadre [ListJobs](#) d'appels d'[DescribeJobsAPI](#).

Vous pouvez éventuellement configurer le `jobStateLimitActions` paramètre par le biais [CreateJobQueue](#) d'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 instances Amazon EC2 présentant des erreurs de capacité insuffisante. L'annulation de la tâche, manuellement ou en activant le `jobStateLimitActions` paramètre `statusReason`, permet à la tâche suivante de passer en tête de file d'attente.

- **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 [AWSServiceRoleForBatch](#) lié à un service (SLR) ou la politique [AWSBatchServiceRolePolicy](#) 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 calcul Amazon EC2 à priorité élevée 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 de prendre en charge 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 vos autorisations de rôle de service aux autorisations de [rôle de service AWS Batch géré](#) et corrigez les éventuelles lacunes.

Il est recommandé d'utiliser le [AWS Batch SLR dans les environnements informatiques](#) afin d'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.`
- **reason**utilisé pour **jobStateTimeLimitActions** : `MISCONFIGURATION:SERVICE_ROLE_PERMISSIONS`
- **statusReason**message après l'annulation de la tâche : `Canceled by JobStateTimeLimit action due to reason: MISCONFIGURATION:SERVICE_ROLE_PERMISSIONS`

5. Raison : Tous les environnements informatiques ne sont pas valides. Pour plus d'informations, consultez la section [Environnement de INVALID calcul](#). 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.`
6. 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 de l'AMI des ressources de calcul](#) et [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 plus d'informations, consultez [Ressource de calcul Gestion de la mémoire](#).

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 plus d'informations, consultez [the section called "Création d'un VPC"](#).

Limite d'instances Amazon EC2 atteinte

Le nombre d'instances Amazon EC2 dans lesquelles votre compte peut être lancé Région AWS est déterminé par votre quota d'instances EC2. Certains types d'instances sont également soumis à un per-instance-type quota. Pour plus d'informations sur le quota d'instances Amazon EC2 de votre compte, notamment sur la manière de demander une augmentation de limite, consultez les limites de [service Amazon EC2](#) dans le guide de l'utilisateur Amazon EC2.

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 les AMI optimisées Amazon ECS. 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 une politique).
4. Sélectionnez l'AmazonEC2, SpotFleet TaggingRole puis choisissez Attach policy.
5. Choisissez à nouveau votre rôle dans Amazon EC2 Spot Fleet pour supprimer la politique précédente.
6. Sélectionnez le x à droite de la politique de SpotFleet rôle d'AmazonEC2, puis choisissez Détacher.

Les instances ponctuelles ne sont pas réduites

AWS Batch a introduit le rôle AWSServiceRoleForBatchlié 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 à un service soit utilisé dans un environnement de calcul EC2 Spot, mais que le rôle Spot utilisé n'inclut pas la politique gérée par SpotFleetTaggingRoleAmazonEC2. 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 SpotFleet TaggingRole gérée par AmazonEC2 à votre rôle Spot Fleet dans AWS Management Console](#)
- [Associez la politique SpotFleet TaggingRole gérée par Amazon EC2 à votre rôle Spot Fleet à l'aide du AWS CLI](#)

Associez la politique SpotFleet TaggingRole gérée par AmazonEC2 à votre rôle Spot Fleet dans 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 une politique).
4. Sélectionnez l'AmazonEC2, SpotFleet TaggingRole puis choisissez Attach policy.
5. Choisissez à nouveau votre rôle dans Amazon EC2 Spot Fleet pour supprimer la politique précédente.
6. Sélectionnez le x à droite de la politique de SpotFleetrôle d'AmazonEC2, puis choisissez Détacher.

Associez la politique SpotFleet TaggingRole gérée par Amazon EC2 à votre rôle Spot Fleet à l'aide du AWS CLI

Les exemples de commandes supposent que votre rôle Amazon EC2 Spot Fleet s'appelle *Amazon SpotFleet* EC2 Role. Si votre rôle utilise un nom différent, ajustez les commandes en conséquence.

Pour associer la politique SpotFleet TaggingRole gérée par AmazonEC2 à votre rôle Spot Fleet

1. Pour associer la politique IAM SpotFleet TaggingRole gérée par AmazonEC2 à votre *SpotFleetrôle AmazonEC2*, 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 gérée par SpotFleet rôle AmazonEC2 de votre rôle *AmazonEC2*, exécutez la commande *SpotFleet* 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`, qui ont été 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 vCPU (`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](#)
- [Vérifiez que le aws-auth ConfigMap est correctement configuré](#)
- [Les autorisations ou les liaisons RBAC ne sont pas correctement configurées](#)

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 :

- 1.29
- 1.28
- 1.27
- 1.26
- 1.25
- 1.24
- 1.23

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 plus d'informations, consultez la section [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 plus d'informations, consultez [Commencer à utiliser AWS Batch Amazon EKS](#).

Environnement informatique 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 enINVALID. 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 `kubectl`. 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 plus d'informations, consultez [Étape 1 : Préparation de votre cluster Amazon EKS pour AWS Batch](#). Vous pouvez également 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.

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 plus d'informations, consultez [Étape 1 : Préparation de 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]

```

```
$ 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 plus d'informations, voir [Étape 1 : Préparation de votre cluster Amazon EKS pour AWS Batch](#).

Bonnes 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 les AMI](#)
- [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 regroupez vos tâches 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 terminé 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 Batchle 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 vCPU ou plus, considérez la liste de contrôle suivante.

Note

Si vous prévoyez d'exécuter une charge de travail importante sur un million de vCPU ou plus ou si vous avez besoin de conseils pour exécuter à grande échelle, contactez votre AWS équipe.

- Vérifiez vos quotas Amazon EC2 — Vérifiez vos quotas Amazon EC2 (également appelés limites) dans le panneau Service Quotas du. AWS Management Console Si nécessaire, demandez une augmentation du quota pour votre nombre maximal d'instances Amazon EC2. N'oubliez pas que les instances Amazon EC2 Spot et Amazon On-Demand disposent de 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 volume GP2 ou GP3 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](#).
- Évoluez progressivement pour identifier rapidement les goulots d'étranglement : pour une tâche exécutée sur un million de vCPU ou plus, commencez par une baisse puis augmentez

progressivement afin de pouvoir identifier les goulots d'étranglement à un stade précoce. Par exemple, commencez par exécuter sur 50 000 vCPU. Ensuite, augmentez le nombre à 200 000 vCPU, puis à 500 000 vCPU, etc. En d'autres termes, continuez à augmenter progressivement le nombre de vCPU jusqu'à atteindre le nombre de vCPU souhaité.

- 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 interruptions peuvent se produire même lors du passage de 1 000 à 5 000 vCPU. 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](#)

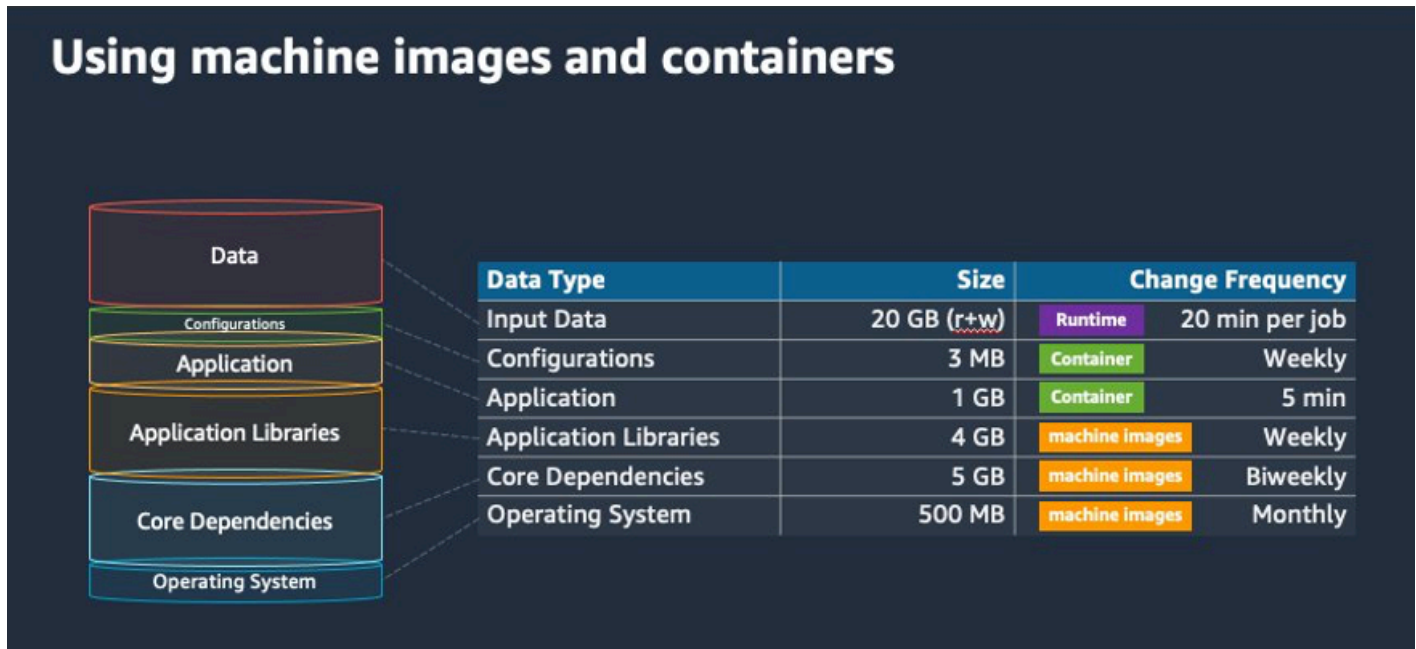
Optimisez les conteneurs et les AMI

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. Ainsi, une couche importante peut 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 avec plus d'un million de vCPU.



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 vCPU ou moins, aucun GPU et 120 GiB de mémoire ou moins.

Pour plus d'informations, consultez [Quand utiliser Fargate](#).

Si vous avez les exigences suivantes, nous vous recommandons d'utiliser des instances Amazon EC2 :

- 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 AWS Fargate impossibles à fournir, telles que des GPU, 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, le modèle de lancement Amazon EC2 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 :

- La durée d'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 la replanification 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 à partir d'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'un Job](#), 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 Spot Amazon Elastic Compute Cloud (EC2), 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 meilleures pratiques Amazon EC2 Spot suivantes pour : AWS Batch

- Choisissez la stratégie **SPOT_CAPACITY_OPTIMIZED** d'allocation : AWS Batch choisit les instances Amazon EC2 parmi les pools de capacité Amazon EC2 Spot les plus profonds. Si vous êtes préoccupé par les interruptions, c'est un choix approprié. Pour plus d'informations, consultez [Stratégies d'allocation](#).
- Diversifier les types d'instances : pour diversifier vos types d'instances, considérez les tailles et les familles compatibles, puis laissez le AWS Batch choix 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 ponctuelles d'Amazon EC2, 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"  
  }]  
}
```

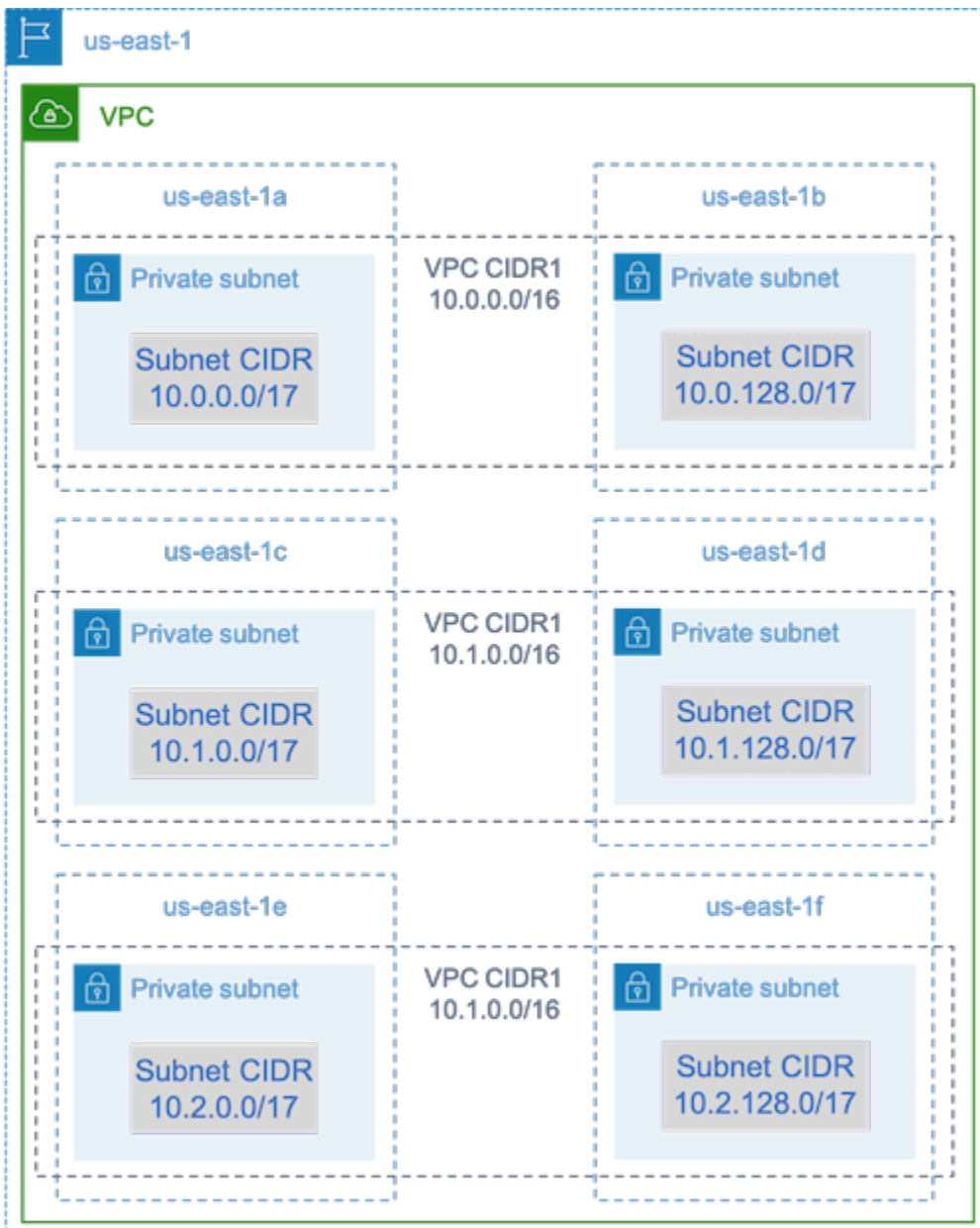
```
}
```

- 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 Spot Amazon EC2 qui sont récupérées et sur les zones de disponibilité dans lesquelles se trouvent les instances Spot. Pour plus d'informations, consultez le 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 vCPU et que la tâche nécessite 10 000 vCPU. Dans ce cas, le quota de service ne répond pas aux exigences du poste. Pour plus d'informations et des instructions de dépannage, consultez les rubriques [Quotas de service Amazon EC2](#) et [Comment augmenter le quota de service de mes ressources Amazon EC2 ?](#).
- 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 les routages interdomaines sans classe (CIDR) 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 CIDR `10.x.0.0/16` et un sous-réseau 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 console Amazon EC2, 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 données Amazon EC2 de votre AMI ou le modèle de lancement peuvent également ne pas être correctement configurés. Pour isoler la cause première, créez une instance Amazon EC2 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 AMI de ressources de calcul](#).

- 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 de calcul 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 console Amazon EC2, mais aucune instance de conteneur Amazon ECS dans votre cluster Amazon ECS, l'agent Amazon ECS n'est peut-être pas installé sur l'Amazon Machine Image (AMI). De plus, il est possible que l'agent Amazon ECS, les données Amazon EC2 de votre AMI ou le modèle de lancement ne soient pas correctement configurés. Pour isoler la cause première, créez une instance Amazon EC2 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 [AMI de ressources de calcul](#).
- 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, consultez la section [Comparer AWS Support les 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 Batchexécution](#).

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
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		5 décembre 2023

BatchServiceRolePolicy

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 des identifiants de relevé, 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. AWS 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 du partage 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

Compatibilité avec Amazon Linux 2	AWS Batch ajoute la prise en charge de la sélection automatique des AMI Amazon Linux 2 dans l'environnement informatique à l'aide des paramètres de configuration EC2.	24 novembre 2020
Stratégie de nouvelle tentative améliorée	AWS Batch améliore la stratégie de nouvelle tentative pour les offres d'emploi. Désormais, les tâches peuvent être réessayées ou arrêter de nouvelles tentatives en faisant correspondre la <code>ExitCode</code> tâche ou en faisant correspondre une tâche présentant <code>StatusReason</code> des modèles. <code>Reason</code>	20 octobre 2020
Étiquette des ressources	AWS Batch permet d'ajouter des 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 GPU requis par chaque tâche et AWS Batch dimensionner les instances en conséquence.	4 avril 2019
Tâches parallèles sur plusieurs nœuds	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 instances Amazon EC2.	19 novembre 2018
Autorisations de niveau ressource	AWS Batch prend en charge les autorisations au niveau des ressources sur plusieurs opérations d'API.	12 novembre 2018
Support du modèle Amazon EC2 Launch	AWS Batch ajoute la prise en charge de l'utilisation de modèles de lancement avec des environnements informatiques.	12 novembre 2018

[AWS Batch délais d'expiration des tâches](#)

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.

1 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 Marquage 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 Spot Amazon EC2 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.