



AWS ParallelCluster Guide de l'utilisateur (v3)

AWS ParallelCluster



AWS ParallelCluster: AWS ParallelCluster Guide de l'utilisateur (v3)

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

| | |
|---|----|
| Présentation d'AWS ParallelCluster | 1 |
| Tarification | 1 |
| Con AWS ParallelCluster figuration | 2 |
| Configuration d'un Compte AWS | 2 |
| Inscrivez-vous pour un Compte AWS | 2 |
| Création d'un utilisateur doté d'un accès administratif | 3 |
| Création d'une paire de clés | 4 |
| Installation de la AWS ParallelCluster CLI | 4 |
| Installation AWS ParallelCluster dans un environnement virtuel (recommandé) | 5 |
| Installation AWS ParallelCluster dans un environnement non virtuel à l'aide de pip | 7 |
| Installation AWS ParallelCluster en tant qu'application autonome | 8 |
| Étapes à suivre après l'installation | 10 |
| Installation de l'AWS ParallelClusterinterface utilisateur | 11 |
| Installation de l'AWS ParallelClusterinterface utilisateur | 11 |
| Création d'un domaine personnalisé | 14 |
| Options du groupe d'utilisateurs Amazon Cognito | 17 |
| Identifier la version AWS ParallelCluster et la version de AWS ParallelCluster l'interface utilisateur | 20 |
| Mettre à jour l'AWS ParallelClusterinterface utilisateur vers une nouvelle AWS ParallelCluster version | 20 |
| AWS ParallelClusterCoûts d'interface utilisateur | 20 |
| Premiers pas | 21 |
| Configuration et création d'un cluster à l'aide de la AWS ParallelCluster CLI | 21 |
| Configuration et création d'un cluster avec l' AWS ParallelCluster interface utilisateur | 32 |
| Connexion à un cluster | 33 |
| Accès de plusieurs utilisateurs aux clusters | 34 |
| Création d'un Active Directory | 35 |
| Création d'un cluster avec un domaine AD | 36 |
| Connectez-vous à un cluster intégré à un domaine AD | 39 |
| Exécution de tâches MPI | 40 |
| Exemple AWS Managed Microsoft AD de configuration de cluster LDAP (S) | 40 |
| Bonnes pratiques | 45 |
| Bonnes pratiques : sélection du type d'instance du nœud principal | 45 |
| Bonnes pratiques : performances du réseau | 45 |

| | |
|---|-----|
| Bonnes pratiques : alertes budgétaires | 47 |
| Bonnes pratiques : déplacement d'un cluster vers une nouvelle version AWS ParallelCluster mineure ou un correctif | 47 |
| Passer de la version AWS ParallelCluster 2.x à la version 3.x | 48 |
| Actions d'amorçage personnalisées | 48 |
| AWS ParallelCluster 2.x et 3.x utilisent une syntaxe de fichier de configuration différente | 49 |
| Langage inclusif | 55 |
| Support pour les planificateurs | 55 |
| AWS ParallelCluster CLI | 56 |
| Mise à jour de configuration IMDS | 59 |
| Régions prises en charge pour AWS ParallelCluster | 59 |
| En utilisant AWS ParallelCluster | 61 |
| AWS ParallelCluster INTERFACE UTILISATEUR | 62 |
| AWS Lambda Configuration VPC dans AWS ParallelCluster | 64 |
| AWS Identity and Access Management autorisations dans AWS ParallelCluster | 65 |
| AWS ParallelCluster Rôles d'instance Amazon EC2 | 66 |
| AWS ParallelCluster exemples de politiques <code>pcluster</code> utilisateur | 67 |
| AWS ParallelCluster exemples de politiques utilisateur pour la gestion des ressources IAM | 82 |
| AWS ParallelCluster paramètres de configuration pour gérer les autorisations IAM | 88 |
| Configurations réseau | 103 |
| AWS ParallelCluster dans un seul sous-réseau public | 104 |
| AWS ParallelCluster en utilisant deux sous-réseaux | 106 |
| AWS ParallelCluster dans un seul sous-réseau privé connecté à l'aide de AWS Direct Connect | 107 |
| AWS ParallelCluster avec AWS Batch planificateur | 108 |
| AWS ParallelCluster dans un seul sous-réseau sans accès à Internet | 111 |
| Nœuds de connexion | 117 |
| Actions de bootstrap personnalisées | 121 |
| Configuration | 123 |
| Arguments | 127 |
| Exemple de cluster avec des actions bootstrap personnalisées | 127 |
| Exemple de mise à jour d'un script bootstrap personnalisé pour IMDSv2 | 129 |
| Exemple de mise à jour d'une configuration pour IMDSv1 | 130 |
| Utilisation des services dans Amazon S3 | 131 |
| Exemples | 131 |

| | |
|--|-----|
| Utilisation de instances Spot | 132 |
| Scénario 1 : Une instance Spot sans tâches en cours d'exécution est interrompue | 133 |
| Scénario 2 : Une instance Spot exécutant des tâches à nœud unique est interrompue | 133 |
| Scénario 3 : Une instance Spot exécutant des tâches à plusieurs nœuds est interrompue ... | 133 |
| Planificateurs pris en charge par AWS ParallelCluster | 133 |
| Slurm Workload Manager | 134 |
| AWS Batch | 201 |
| Stockage partagé | 209 |
| Configuration du stockage partagé | 212 |
| Utilisation du stockage partagé | 215 |
| Quotas | 219 |
| Identification | 220 |
| Surveillance AWS ParallelCluster et journaux | 223 |
| Intégration avec Amazon CloudWatch Logs avec Amazon Logs | 224 |
| Tableau de CloudWatch bord Amazon | 228 |
| CloudWatchAlarmes Amazon pour les métriques des clusters | 230 |
| AWS ParallelClusterrotation des journaux configurée | 233 |
| pcClusterJournaux de la CLI | 234 |
| Journaux de sortie de la console Amazon EC2 | 235 |
| Récupérez les journauxAWS ParallelCluster d'AWS ParallelClusterinterface utilisateur et d'exécution | 236 |
| Récupération et conservation des journaux | 238 |
| AWS CloudFormation ressource personnalisée | 241 |
| Pile de fournisseurs hébergée par AWS ParallelCluster | 242 |
| Ressource de cluster | 244 |
| Opérations du cluster | 247 |
| Résolution des problèmes liés aux piles qui incluent la ressource AWS ParallelCluster personnalisée | 247 |
| Elastic Fabric Adapter | 248 |
| Activez Intel MPI | 249 |
| AWS ParallelCluster API | 250 |
| AWS ParallelCluster Documentation de l'API | 251 |
| Déployez avec AWS CLI | 252 |
| Mettre à jour l'API | 255 |
| Invocation de AWS ParallelCluster l'API | 255 |
| Accès aux journaux et aux métriques de l'API | 258 |

| | |
|--|-----|
| AWS ParallelCluster pour Terraform | 258 |
| Connectez-vous au nœud principal via NICE DCV | 259 |
| Certificat HTTPS NICE DCV | 259 |
| Licence NICE DCV | 260 |
| Utiliser <code>pcluster update-cluster</code> | 260 |
| Politique de mise à jour : définitions | 260 |
| Exemples <code>pcluster update-cluster</code> | 264 |
| AWS ParallelCluster Personnalisation de l'AMI | 267 |
| AWS ParallelCluster Considérations relatives à la personnalisation de l'AMI | 268 |
| Réaliser des tests de validation de composants personnalisés | 268 |
| Surveillez le processus Image Builder à l'aide de <code>pcluster</code> commandes pour faciliter le débogage | 269 |
| Autres considérations | 270 |
| Lancez des instances avec des réservations de capacité à la demande (ODCR) | 270 |
| Utilisation de l'ODCR avec AWS ParallelCluster | 271 |
| Lancer des instances avec des blocs de capacité (CB) | 280 |
| Utiliser CB avec AWS ParallelCluster | 280 |
| Corriger les AMI et remplacer les instances Amazon EC2 | 282 |
| Mise à jour ou remplacement de l'instance du nœud principal | 283 |
| Enregistrez des données à partir de disques éphémères | 284 |
| Arrêter et démarrer le nœud principal d'un cluster | 284 |
| Systèmes d'exploitation | 286 |
| Considérations relatives au système d'exploitation | 286 |
| Référence pour AWS ParallelCluster | 289 |
| AWS ParallelCluster commandes CLI version 3 | 289 |
| <code>pcluster</code> | 290 |
| <code>pcluster3-config-convert</code> | 334 |
| Fichiers de configuration | 335 |
| Fichier de configuration du cluster | 335 |
| Créez des fichiers de configuration d'image | 472 |
| AWS ParallelCluster Référence d'API | 480 |
| Construire une image | 481 |
| <code>createCluster</code> | 486 |
| Supprimer le cluster | 492 |
| <code>deleteClusterInstances</code> | 495 |
| Supprimer l'image | 497 |

| | |
|---|-----|
| Décrire le cluster | 500 |
| describeClusterInstances | 508 |
| describeComputeFleet | 512 |
| Décrire l'image | 513 |
| getClusterLogÉvènements | 520 |
| getClusterStackÉvènements | 524 |
| getImageLogÉvènements | 528 |
| getImageStackÉvènements | 532 |
| Clusters de listes | 536 |
| listClusterLogStreams | 540 |
| listImageLogStreams | 544 |
| Images de la liste | 548 |
| listOfficialImages | 551 |
| Mettre à jour le cluster | 554 |
| updateComputeFleet | 561 |
| AWS ParallelClusterAPI de bibliothèque Python | 563 |
| AWS ParallelClusterAutorisation de bibliothèque Python | 563 |
| Installez leAWS ParallelClusterbibliothèque Python | 564 |
| Opérations de l'API du cluster | 564 |
| Opérations de l'API du parc de calcul | 568 |
| Opérations de journalisation des clusters et des piles | 570 |
| Opérations de l'API Image | 572 |
| Opérations de journalisation des images et des piles | 575 |
| Exemple | 578 |
| AWS Lambdapour leAWS ParallelClusterbibliothèque Python | 579 |
| Fonctionnement d'AWS ParallelCluster | 581 |
| AWS ParallelCluster processus | 581 |
| clustermgtd | 581 |
| clusterstatusmgtd | 582 |
| computemgtd | 582 |
| AWS services utilisés par AWS ParallelCluster | 583 |
| Amazon API Gateway | 584 |
| AWS Batch | 584 |
| AWS CloudFormation | 584 |
| Amazon CloudWatch | 585 |
| CloudWatch Événements Amazon | 585 |

| | |
|---|-----|
| Amazon CloudWatch Logs | 585 |
| AWS CodeBuild | 585 |
| Amazon DynamoDB | 586 |
| Amazon Elastic Block Store | 586 |
| Amazon Elastic Compute Cloud | 586 |
| Amazon Elastic Container Registry | 587 |
| Amazon EFS | 587 |
| Amazon FSx pour Lustre | 587 |
| Amazon FSx pour ONTAP NetApp | 587 |
| Amazon FSx pour OpenZFS | 588 |
| AWS Identity and Access Management | 588 |
| AWS Lambda | 588 |
| Amazon RDS | 588 |
| Amazon Route 53 | 589 |
| Amazon Simple Notification Service | 589 |
| Amazon Simple Storage Service | 589 |
| Amazon VPC | 590 |
| Elastic Fabric Adapter | 590 |
| EC2 Image Builder | 590 |
| NICE DCV | 590 |
| AWS ParallelClusterRépertoires internes | 590 |
| Didacticiels | 592 |
| Exécution de votre premier job sur AWS ParallelCluster | 592 |
| Vérification de votre installation | 593 |
| Création de votre premier cluster | 593 |
| Connexion à votre nœud principal | 594 |
| Exécution de votre première tâche avec Slurm | 595 |
| Création d'une AWS ParallelCluster AMI personnalisée | 596 |
| Comment personnaliser l' AWS ParallelCluster AMI | 597 |
| Créez une AWS ParallelCluster AMI personnalisée | 597 |
| Modifier une AWS ParallelCluster AMI | 605 |
| Intégration d'Active Directory | 607 |
| Configuration du chiffrement du stockage partagé à l'aide d'une AWS KMS clé | 640 |
| Création de la stratégie | 641 |
| Configuration et création du cluster | 642 |
| Exécution de tâches dans un cluster en mode de files d'attente multiples | 643 |

| | |
|--|-----|
| Configurez votre cluster | 644 |
| Créer votre cluster | 645 |
| Connectez-vous au nœud principal | 646 |
| Exécuter le job en mode file d'attente multiple | 647 |
| Utilisation de l' AWS ParallelCluster API | 651 |
| Création d'un cluster avec Slurm comptabilité | 665 |
| Étape 1 : créer le VPC et les sous-réseaux pour AWS ParallelCluster | 666 |
| Étape 2 : Création de la pile de base de données | 666 |
| Étape 3 : créer un cluster avec la Slurm comptabilité activée | 667 |
| Création d'un cluster avec une Slurmdbd comptabilité externe | 667 |
| Étape 1 : Création de la pile Slurmdbd | 669 |
| Étape 2 : Création d'un cluster avec l'Slurmdbdoption externe activée | 670 |
| Revenir à une version précédente d'un document AWS Systems Manager | 671 |
| Revenir à une version précédente du document SSM | 671 |
| Création d'un cluster avec AWS CloudFormation | 674 |
| Création de clusters à l'aide d'une pile CloudFormation à création rapide | 674 |
| Création de clusters avec l'interface de ligne de commande de AWS CloudFormation (CLI) | 677 |
| Afficher la sortie CloudFormation du cluster | 679 |
| Accédez à votre cluster | 680 |
| Nettoyage | 680 |
| Déploiement ParallelCluster d'une API avec Terraform | 681 |
| Définir un projet Terraform | 681 |
| Déploiement de l'API | 683 |
| Autorisations nécessaires | 683 |
| Création d'un cluster avec Terraform | 687 |
| Définir un projet Terraform | 687 |
| Déploiement du cluster | 693 |
| Autorisations nécessaires | 694 |
| Création d'une AMI personnalisée avec Terraform | 695 |
| Définir un projet Terraform | 696 |
| Déployer l'AMI | 699 |
| Autorisations nécessaires | 699 |
| AWS ParallelClusterIntégration de l'interface utilisateur à Identity Center | 700 |
| Activer IAM Identity Center | 701 |
| Ajouter votre application à IAM Identity Center | 704 |
| AWS ParallelCluster résolution des problèmes | 712 |

| | |
|---|-----|
| Essayer de créer un cluster | 713 |
| failureCode est OnNodeConfiguredExecutionFailure | 713 |
| failureCode est OnNodeConfiguredDownloadFailure | 714 |
| failureCode est OnNodeConfiguredFailure | 714 |
| failureCode est OnNodeStartExecutionFailure | 714 |
| failureCode est OnNodeStartDownloadFailure | 715 |
| failureCode est OnNodeStartFailure | 715 |
| failureCode est EbsMountFailure | 716 |
| failureCode est EfsMountFailure | 716 |
| failureCode est FsxMountFailure | 716 |
| failureCode est RaidMountFailure | 717 |
| failureCode est AmiVersionMismatch | 717 |
| failureCode est InvalidAmi | 717 |
| failureCode porte HeadNodeBootstrapFailure la mention « failureReason Impossible de configurer le nœud principal » | 718 |
| failureCode est que HeadNodeBootstrapFailure le délai de création failureReason du cluster est expiré. | 718 |
| failureCode est HeadNodeBootstrapFailure associé à failureReason Failed to bootstrap the head node. | 719 |
| failureCode est ResourceCreationFailure | 720 |
| failureCode est ClusterCreationFailure | 720 |
| Voir WaitCondition timed out... dans la CloudFormation pile | 720 |
| Voir Resource creation cancelled dans la CloudFormation pile | 721 |
| Erreurs Failed to run cfn-init... visibles ou autres dans la AWS CloudFormation pile | 721 |
| Voir chef-client.log se termine par INFO: Waiting for static fleet capacity provisioning | 721 |
| Voyant Failed to run preinstall or postinstall in cfn-init.log | 721 |
| Voir This AMI was created with xxx, but is trying to be used with xxx... dans la CloudFormation pile | 721 |
| Voir This AMI was not baked by AWS ParallelCluster... dans la CloudFormation pile | 721 |
| Voir que pcluster create-cluster la commande ne s'exécute pas localement | 722 |
| Support supplémentaire | 722 |
| Essayer d'exécuter une tâche | 722 |

| | |
|---|-----|
| srun la tâche interactive échoue avec une erreur srun: error: fwd_tree_thread: can't find address for <host>, check slurm.conf | 722 |
| Job bloqué dans son CF état avec squeue la commande | 723 |
| Exécuter des travaux à grande échelle et voir nfsd: too many open connections, consider increasing the number of threads in /var/log/messages | 723 |
| Exécution d'une tâche MPI | 724 |
| Essayer de mettre à jour un cluster | 725 |
| pcluster update-cluster la commande ne s'exécute pas localement | 725 |
| Voir clusterStatus c'est UPDATE_FAILED avec pcluster describe-cluster commande | 725 |
| Le délai de mise à jour du cluster a expiré | 725 |
| Essayer d'accéder au stockage | 725 |
| Utilisation d'un système de fichiers externe Amazon FSx for Lustre | 725 |
| Utilisation d'un système de fichiers Amazon Elastic File System externe | 725 |
| Essayer de supprimer un cluster | 726 |
| La pcluster delete-cluster commande ne s'exécute pas localement | 726 |
| La pile de clusters ne parvient pas à être supprimée | 726 |
| Essayer de mettre à niveau la pile AWS ParallelCluster d'API | 726 |
| Observation des erreurs lors de l'initialisation des nœuds de calcul | 726 |
| Voir Node bootstrap error dans clustermgtd.log | 726 |
| J'ai configuré des réservations de capacité à la demande (ODCR) ou des instances réservées zonales | 726 |
| Voir An error occurred (VcpuLimitExceeded)slurm_resume.log quand je ne parviens pas à exécuter une tâche, ou quand je ne parviens pas à créer un cluster clustermgtd.log | 728 |
| Voir An error occurred (InsufficientInstanceCapacity)slurm_resume.log quand je ne parviens pas à exécuter une tâche, ou quand je ne parviens pas à créer un cluster clustermgtd.log | 728 |
| Voir que les nœuds sont en DOWN état avec Reason (Code:InsufficientInstanceCapacity)... .. | 728 |
| Voir cannot change locale (en_US.utf-8) because it has an invalid name dans slurm_resume.log | 729 |
| Aucun des scénarios précédents ne s'applique à ma situation | 729 |
| Résolution des problèmes liés aux indicateurs de santé du | 730 |
| Voir le graphique des erreurs de provisionnement des instances | 730 |
| Affichage du graphique des erreurs d'instance non conformes | 732 |

| | |
|---|-----|
| Voir le graphique des temps d'inactivité de la flotte de calcul | 734 |
| Résolution des problèmes de déploiement de clusters | 734 |
| Afficher AWS CloudFormation les événements sur CREATE_FAILED | 735 |
| Utiliser la CLI pour afficher les flux de journaux | 737 |
| Recréez le cluster défaillant avec rollback-on-failure | 740 |
| Résolution des problèmes de déploiement de clusters à l'aide de Terraform | 741 |
| ParallelCluster API introuvable | 741 |
| L'utilisateur n'est pas autorisé à appeler ParallelCluster l'API | 741 |
| Résolution des problèmes de dimensionnement | 742 |
| Journaux clés pour le débogage | 743 |
| Affichage InsufficientInstanceCapacity d'une erreur slurm_resume.log lorsque je ne parviens pas à exécuter une tâche ou clustermgtd.log lorsque je ne parviens pas à créer un cluster | 728 |
| Résolution des problèmes d'initialisation des nœuds | 746 |
| Résolution des problèmes de remplacement et de terminaison inattendus de nœuds | 748 |
| Remplacement, arrêt ou mise hors tension des instances et des nœuds problématiques | 750 |
| InactiveÉtat de la file d'attente (partition) | 751 |
| Résolution d'autres problèmes connus liés aux nœuds et aux tâches | 751 |
| Problèmes liés aux groupes de placement et au lancement d'instances | 751 |
| Répertoires qui ne peuvent pas être remplacés | 752 |
| Résolution des problèmes dans NICE DCV | 752 |
| Logos pour NICE DCV | 752 |
| Problèmes liés à Ubuntu NICE DCV | 752 |
| Résolution des problèmes dans les clusters avec AWS Batch intégration | 753 |
| Problèmes liés au nœud principal | 753 |
| Problèmes de calcul | 754 |
| Échecs au travail | 754 |
| Erreur d'expiration du délai de connexion en cas d'URL du point de terminaison | 754 |
| Résolution des problèmes d'intégration multi-utilisateurs avec Active Directory | 754 |
| Résolution des problèmes spécifiques à Active Directory | 755 |
| Activer le mode de débogage | 756 |
| Comment passer de LDAPS à LDAP | 756 |
| Comment désactiver la vérification du certificat du serveur LDAPS | 757 |
| Comment se connecter avec une clé SSH plutôt qu'un mot de passe | 757 |
| Comment réinitialiser le mot de passe d'un utilisateur et les mots de passe expirés | 757 |
| Comment vérifier le domaine joint | 758 |

| | |
|--|-------------|
| Comment résoudre les problèmes liés aux certificats | 759 |
| Comment vérifier que l'intégration avec Active Directory fonctionne | 761 |
| Comment résoudre les problèmes de connexion aux nœuds de calcul | 761 |
| Problèmes connus liés aux tâches SimCenter StarCCM+ dans un environnement multi-utilisateurs | 762 |
| Problèmes connus liés à la résolution des noms d'utilisateur | 762 |
| Comment résoudre les problèmes de création du répertoire de base | 763 |
| Résolution des problèmes liés aux AMI personnalisées | 764 |
| Résolution d'un délai d'expiration de mise à jour du cluster en cas cfn-hup d'inexécution | 765 |
| Dépannage du réseau | 766 |
| Problèmes liés au cluster dans un seul sous-réseau public | 766 |
| La mise à jour du cluster a échoué lors d'onNodeUpdatedune action personnalisée | 766 |
| Voir les erreurs liées à la Slurm configuration personnalisée | 766 |
| Alarmes de cluster | 767 |
| Support supplémentaire | 768 |
| AWS ParallelCluster politique de support | 769 |
| Sécurité | 770 |
| Informations de sécurité pour les services utilisés par AWS ParallelCluster | 771 |
| Protection des données | 771 |
| Chiffrement des données | 772 |
| Consultez aussi | 774 |
| Identity and Access Management (Gestion des identités et des accès) | 774 |
| Validation de la conformité | 775 |
| Application de TLS 1.2 | 776 |
| Déterminez vos protocoles actuellement pris en charge | 776 |
| Compiler OpenSSL et Python | 778 |
| Notes de mise à jour et historique du document | 780 |
| | dcccclxxxix |

Présentation d'AWS ParallelCluster

AWS ParallelCluster est un outil de gestion de clusters open source AWS compatible qui vous aide à déployer et à gérer des clusters de calcul haute performance (HPC) dans le AWS Cloud. Il configure automatiquement les ressources de calcul, le planificateur et le système de fichiers partagé requis. Vous pouvez l'utiliser AWS ParallelCluster avec AWS Batch et les Slurm planificateurs.

Avec AWS ParallelCluster, vous pouvez rapidement créer et déployer des environnements de calcul HPC de validation de principe et de production. Vous pouvez également créer et déployer un flux de travail de haut niveau par-dessus AWS ParallelCluster, tel qu'un portail de génomique qui automatise l'intégralité d'un flux de travail de séquençage de l'ADN.

Vous pouvez y accéder AWS ParallelCluster en utilisant les méthodes suivantes :

- [AWS ParallelCluster interface de ligne de commande \(CLI\)](#)
- [API AWS ParallelCluster](#)
- [AWS ParallelCluster Interface utilisateur](#) (ajoutée avec la version 3.5.0)
- [AWS ParallelCluster API de bibliothèque Python](#) (ajouté avec la version 3.5.0)
- En tant que [AWS CloudFormation ressource personnalisée](#) (ajouté avec la version 3.6.0)

Tarifification

Lorsque vous utilisez l'interface de ligne de commande (CLI) ou l'API, vous ne payez que pour les AWS ressources créées lorsque vous créez ou mettez à jour AWS ParallelCluster des images et des clusters. Pour plus d'informations, veuillez consulter [AWS services utilisés par AWS ParallelCluster](#).

L'AWS ParallelCluster interface utilisateur repose sur une architecture sans serveur et vous pouvez l'utiliser dans la catégorie AWS Free Tier dans la plupart des cas. Pour plus d'informations, veuillez consulter [AWS ParallelCluster Coûts d'interface utilisateur](#).

Con AWS ParallelCluster figuration

Rubriques

- [Configuration d'un Compte AWS](#)
- [Création d'une paire de clés](#)
- [Installation de l'interface de ligne de AWS ParallelCluster commande \(CLI\)](#)
- [Étapes à suivre après l'installation](#)
- [Installation de l'AWS ParallelClusterinterface utilisateur](#)
- [Commencer avec AWS ParallelCluster](#)
- [Accès de plusieurs utilisateurs aux clusters](#)
- [Bonnes pratiques](#)
- [Passer de la version AWS ParallelCluster 2.x à la version 3.x](#)
- [Régions prises en charge pour AWS ParallelCluster](#)

Configuration d'un Compte AWS

Configurez un AWS compte à utiliser AWS ParallelCluster.

Inscrivez-vous pour un Compte AWS

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

Pour vous inscrire à un Compte AWS

1. Ouvrez <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 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 AWS IAM Identity Center l'utilisateur.

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'Connexion à AWS utilisateur.

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 AWS IAM Identity Center l'utilisateur.

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

Pour obtenir des instructions, voir [Ajouter des groupes](#) dans le guide de AWS IAM Identity Center l'utilisateur.

Création d'une paire de clés

Pour déployer des clusters, AWS ParallelCluster lance des instances Amazon EC2 afin de créer le nœud principal du cluster et les nœuds de calcul. Pour effectuer des tâches de cluster, telles que l'exécution et la surveillance de tâches ou la gestion des utilisateurs, vous devez être en mesure d'accéder au nœud principal du cluster. Pour vérifier que vous pouvez accéder à l'instance du nœud principal via SSH, vous devez utiliser une paire de clés Amazon EC2. Pour savoir comment créer une paire de clés, consultez la section [Créer une paire de clés](#) dans le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux.

Installation de l'interface de ligne de commande (CLI) de AWS ParallelCluster

AWS ParallelCluster est distribué sous forme de package Python et est installé à l'aide du gestionnaire de `pip` packages Python. Pour obtenir des instructions sur l'installation de packages Python, consultez la section [Installation de packages](#) dans le Guide de l'utilisateur des packages Python.

Modes d'installation AWS ParallelCluster :

- [Installation AWS ParallelCluster dans un environnement virtuel \(recommandé\)](#)
- [Installation AWS ParallelCluster dans un environnement non virtuel à l'aide de pip](#)
- [Installation AWS ParallelCluster en tant qu'application autonome](#)

Vous trouverez le numéro de version de la CLI la plus récente sur la [page des versions sur GitHub](#). Dans ce guide, les exemples de commandes supposent que vous avez installé une version de Python ultérieure à la version 3.6. Les exemples de commande pip utilisent la version pip3.

Gérez à la fois AWS ParallelCluster 2 et AWS ParallelCluster 3

Pour les clients qui utilisent à la fois les versions AWS ParallelCluster 2 et AWS ParallelCluster 3 et qui souhaitent gérer les interfaces de ligne de commande des deux packages, nous vous recommandons d'installer les AWS ParallelCluster versions 2 et AWS ParallelCluster 3 dans [des environnements virtuels](#) différents. Cela garantit que vous pouvez continuer à utiliser chaque version de AWS ParallelCluster et toutes les ressources de cluster associées.

Installation AWS ParallelCluster dans un environnement virtuel (recommandé)

Nous vous recommandons d'effectuer l'installation AWS ParallelCluster dans un environnement virtuel afin d'éviter tout conflit de version entre les exigences et les autres pip packages.

Prérequis

- AWS ParallelCluster nécessite Python 3.7 ou version ultérieure. Si ce n'est pas déjà fait, [téléchargez une version compatible](#) pour votre plateforme [sur python.org](#).

Pour effectuer une installation AWS ParallelCluster dans un environnement virtuel

1. S'il virtualenv n'est pas installé, installez-le virtualenv à l'aide de pip3. Si `python3 -m virtualenv help` affiche des informations d'aide, passez à l'étape 2.

```
$ python3 -m pip install --upgrade pip
$ python3 -m pip install --user --upgrade virtualenv
```

Exécutez `exit` pour quitter la fenêtre de terminal actuelle et ouvrir une nouvelle fenêtre de terminal afin de récupérer les modifications apportées à l'environnement.

2. Créez et nommez un environnement virtuel.

```
$ python3 -m virtualenv ~/apc-ve
```

Vous pouvez également utiliser l'option `-p` pour spécifier une version spécifique de Python.

```
$ python3 -m virtualenv -p $(which python3) ~/apc-ve
```

3. Activez votre nouvel environnement virtuel.

```
$ source ~/apc-ve/bin/activate
```

4. Effectuez AWS ParallelCluster l'installation dans votre environnement virtuel.

```
(apc-ve)~$ python3 -m pip install --upgrade "aws-parallelcluster"
```

5. Installez Node Version Manager et la dernière version de Long-Term Support (LTS) Node.js. AWS Cloud Development Kit (AWS CDK) (AWS CDK) nécessite Node.js CloudFormation pour la génération de modèles.

Note

Si votre installation de Node.js ne fonctionne pas sur votre plate-forme, vous pouvez installer une version LTS antérieure à la dernière version LTS. Pour plus d'informations, consultez le [calendrier de publication de Node.js](#) et les conditions requises pour le [AWS CDK](#).

Exemple de commande d'installation de Node.js :

```
$ nvm install --lts=Hydrogen
```

```
$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
$ chmod ug+x ~/.nvm/nvm.sh
$ source ~/.nvm/nvm.sh
$ nvm install --lts
$ node --version
```

6. Vérifiez qu'il AWS ParallelCluster est correctement installé.

```
$ pcluster version
{
  "version": "3.7.0"
}
```

La commande `deactivate` vous permet de quitter l'environnement virtuel. Chaque fois que vous démarrez une session, vous devez [réactiver l'environnement](#).

Pour effectuer une mise à niveau vers la dernière version de AWS ParallelCluster, réexécutez la commande d'installation.

```
(apc-ve)~$ python3 -m pip install --upgrade "aws-parallelcluster"
```

Installation AWS ParallelCluster dans un environnement non virtuel à l'aide de pip

Prérequis

- AWS ParallelCluster nécessite Python 3.7 ou version ultérieure. Si ce n'est pas déjà fait, [téléchargez une version compatible](#) pour votre plateforme [sur python.org](#).

Installer AWS ParallelCluster

1. `pip` à utiliser pour installer AWS ParallelCluster.

```
$ python3 -m pip install "aws-parallelcluster" --upgrade --user
```

Lorsque vous utilisez le `--user` commutateur, `pip` s'installe sur. AWS ParallelCluster
~/.local/bin

2. Installez Node Version Manager et la dernière version de Long-Term Support (LTS) Node.js. AWS Cloud Development Kit (AWS CDK) (AWS CDK) nécessite Node.js CloudFormation pour la génération de modèles.

Note

Si votre installation de Node.js ne fonctionne pas sur votre plate-forme, vous pouvez installer une version LTS antérieure à la dernière version LTS. Pour plus d'informations, consultez le [calendrier de publication de Node.js](#) et les conditions requises pour le [AWS CDK](#).

```
$ nvm install --lts=Gallium
```

```
$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
$ chmod ug+x ~/.nvm/nvm.sh
$ source ~/.nvm/nvm.sh
$ nvm install --lts
$ node --version
```

3. Vérifiez qu'il AWS ParallelCluster est correctement installé.

```
$ pcluster version
{
  "version": "3.7.0"
}
```

4. Pour effectuer une mise à niveau vers la dernière version, exécutez à nouveau la commande d'installation.

```
$ python3 -m pip install "aws-parallelcluster" --upgrade --user
```

Installation AWS ParallelCluster en tant qu'application autonome

Installez-le AWS ParallelCluster en tant qu'application autonome sur votre environnement. Suivez les instructions d'installation AWS ParallelCluster sur un système d'exploitation disponible dans la section suivante.

Prérequis

- Environnement doté d'un système d'exploitation compatible avec une version disponible du programme d'installation.

Note

AWS ParallelCluster nécessite NodeJS. AWS ParallelCluster Le programme d'installation inclut une version groupée de NodeJS (v18), qui est installée si elle n'existe pas déjà. Si votre système n'est pas compatible avec NodeJS v18, vous devez installer NodeJS avant de l'installer. AWS ParallelCluster

Linux

Linux x86 (64-bit)

Effectuez AWS ParallelCluster l'installation sur votre environnement.

1. Téléchargez le dernier programme d'[installation de pcluster](#).
2. Décompressez le bundle d'installation et installez-le AWS ParallelCluster à l'aide des commandes suivantes :

```
$ unzip pcluster-installer-bundle-3.10.1.713-node-v18.20.3-Linux_x86_64-signed.zip
-d pcluster-installer-bundle
$ cd pcluster-installer-bundle
$ chmod +x install_pcluster.sh
```

3. Exécutez le script d'installation suivant.

```
$ bash install_pcluster.sh
```

4. Vérifiez qu'il AWS ParallelCluster est correctement installé.

```
$ pcluster version
{
  "version": "3.10.1"
}
```

Résolution des erreurs `pcluster` d'installation

- Si la AWS ParallelCluster version n'est pas renvoyée à l'étape 4, redémarrez le terminal ou source mettez `bash_profile` à jour la `PATH` variable afin d'inclure le nouveau répertoire binaire, comme indiqué dans l'exemple suivant :

```
$ source ~/.bash_profile
```

- Si vous utilisez votre `pcluster` installation pour créer des clusters avec des ressources HTTPS CustomActions spécifiées, plutôt que des URI S3, vous verrez peut-être un WARNING message indiquant que ces ressources ne sont peut-être pas vérifiées ([SSL : CERTIFICATE_VERIFY_FAILED]). Cela est dû à un problème connu et vous pouvez ignorer cet avertissement si vous avez confiance en l'authenticité des ressources spécifiées.

Versions précédentes du bundle d'installation

- Aucun

Étapes à suivre après l'installation

Vous pouvez vérifier qu'il AWS ParallelCluster est correctement installé en exécutant [pcluster version](#).

```
$ pcluster version
{
  "version": "3.7.0"
}
```

AWS ParallelCluster est régulièrement mis à jour. Pour effectuer la mise à jour vers la dernière version de AWS ParallelCluster, réexécutez la commande d'installation. Pour plus d'informations sur la dernière version de AWS ParallelCluster, consultez les [notes AWS ParallelCluster de publication](#).

```
$ pip3 install aws-parallelcluster --upgrade --user
```

Pour désinstaller AWS ParallelCluster, utilisez `pip3 uninstall`.

```
$ pip3 uninstall aws-parallelcluster
```

Si vous n'avez pas Python ni pip3, utilisez la procédure indiquée pour votre environnement.

Installation de l'AWS ParallelClusterinterface utilisateur

L'AWS ParallelClusterinterface utilisateur est une interface utilisateur Web qui reflète la AWS ParallelCluster `pccluster` CLI, tout en offrant une expérience semblable à celle d'une console. Vous installez et accédez à l'AWS ParallelClusterinterface utilisateur dans votreCompte AWS. Lorsque vous l'exécutez, l'AWS ParallelClusterinterface utilisateur accède à une instance de l'AWS ParallelClusterAPI hébergée sur Amazon API Gateway dans votreCompte AWS. Pour plus d'informations sur l'AWS ParallelClusterinterface utilisateur, consultez [AWS ParallelClusterINTERFACE UTILISATEUR](#).

Prérequis :

- Un Compte AWS
- Accès au AWS Management Console

Rubriques

- [Installation de l'AWS ParallelClusterinterface utilisateur](#)
- [Création d'un domaine personnalisé](#)
- [Options du groupe d'utilisateurs Amazon Cognito](#)
- [Identifier la version AWS ParallelCluster et la version de AWS ParallelCluster l'interface utilisateur](#)
- [Mettre à jour l'AWS ParallelClusterinterface utilisateur vers une nouvelle AWS ParallelCluster version](#)
- [AWS ParallelClusterCoûts d'interface utilisateur](#)

Installation de l'AWS ParallelClusterinterface utilisateur

Pour installer une instance de l'AWS ParallelClusterinterface utilisateur, vous devez choisir un lien de AWS CloudFormation création rapide dans Région AWS lequel vous créez des clusters. L'URL de création rapide vous dirige vers un assistant de création de pile dans lequel vous fournissez des entrées de modèle de pile pour créer rapidement et déployer la pile. Pour plus d'informations sur la CloudFormation création rapide de piles, voir [Création de liens de création rapide pour les piles](#) dans le Guide de l'utilisateur. AWS CloudFormation

Note

Vous ne pouvez créer et modifier des clusters ou créer des images qu'avec la même AWS ParallelCluster version que celle que vous utilisez pour installer l'AWS ParallelClusterinterface utilisateur.

AWS ParallelClusterCréation rapide de liens par région dans l'interface utilisateur

Création rapide de liens par l'interface utilisateur

[us-east-1](#)

[us-east-2](#)

[us-west-1](#)

[us-west-2](#)

[eu-west-1](#)

[eu-west-2](#)

[eu-west-3](#)

[eu-central-1](#)

[eu-north-1](#)

[me-south-1](#)

[sa-east-1](#)

[ca-central-1](#)

[ap-northeast-1](#)

[ap-northeast-2](#)

[ap-south-1](#)

Création rapide de liens par l'interface utilisateur

[ap-southeast-1](#)

[ap-southeast-2](#)

[us-gov-west-1](#)

Utilisez un lien de AWS CloudFormation création rapide pour déployer une pile d'AWS ParallelClusterinterface utilisateur avec des piles Amazon Cognito, API Gateway et Amazon EC2 Systems Manager imbriquées.

1. Connectez-vous à la AWS Management Console.
2. Déployez l'AWS ParallelClusterinterface utilisateur en choisissant un lien de Région AWS création rapide dans le tableau au début de cette section. Cela vous amène à l'assistant CloudFormation Create Stack de la console.
3. Entrez une adresse e-mail valide pour l'adresse e-mail de l'administrateur.

Une fois le déploiement terminé avec succès, l'AWS ParallelClusterinterface utilisateur vous envoie un mot de passe temporaire à cette adresse e-mail. Vous utilisez le mot de passe temporaire pour accéder à l'AWS ParallelClusterinterface utilisateur. Si vous supprimez l'e-mail avant d'enregistrer ou d'utiliser le mot de passe temporaire, vous devez supprimer la pile et réinstaller l'AWS ParallelClusterinterface utilisateur.

4. Laissez le reste du formulaire vide ou entrez des valeurs pour les paramètres (facultatifs) afin de personnaliser la création de l'AWS ParallelClusterinterface utilisateur.
5. Notez le nom de la pile à utiliser dans les étapes ultérieures.
6. Accédez à Capabilities. Acceptez les CloudFormation capacités.
7. Choisissez Créer. Il faut environ 15 minutes pour terminer le déploiement de l'AWS ParallelClusterAPI et de l'AWS ParallelClusterinterface utilisateur.
8. Affichez les détails de la pile au fur et à mesure de sa création.
9. Une fois le déploiement terminé, ouvrez l'e-mail d'administrateur envoyé à l'adresse que vous avez saisie. Il contient un mot de passe temporaire que vous utilisez pour accéder à l'AWS ParallelClusterinterface utilisateur. Si vous supprimez définitivement l'e-mail et que vous n'êtes pas encore connecté à l'AWS ParallelClusterinterface utilisateur, vous devez supprimer

la pile d'AWS ParallelClusterinterface utilisateur que vous avez créée et réinstaller l'AWS ParallelClusterinterface utilisateur.

10. Dans la liste des piles de la AWS CloudFormation console, choisissez le lien vers le nom de la pile que vous avez noté à l'étape précédente.
11. Dans Stack details, choisissez Outputs et sélectionnez le lien de la clé nommée **Stackname** URL pour ouvrir l'AWS ParallelClusterinterface utilisateur. **Stackname** est le nom que vous avez noté à l'étape précédente.
12. Entrez le mot de passe temporaire. Suivez les étapes pour créer votre propre mot de passe et connectez-vous.
13. Vous êtes maintenant sur la page d'accueil de l'AWS ParallelClusterinterface utilisateur Région AWS que vous avez sélectionnée.
14. Pour commencer à utiliser l'AWS ParallelClusterinterface utilisateur, voir [Configuration et création d'un cluster avec l' AWS ParallelCluster interface utilisateur](#).

Note

Les sessions PCUI ont une durée par défaut de 5 minutes, ce qui est la valeur minimale fournie par Cognito à partir du PCUI 2023.12.0. Par conséquent, on s'attend à ce qu'un utilisateur supprimé des groupes d'utilisateurs de Cognito puisse toujours accéder au système jusqu'à l'expiration de la session.

Création d'un domaine personnalisé

Découvrez comment créer un domaine personnalisé pour l'AWS ParallelClusterinterface utilisateur. L'interface utilisateur est hébergée sur Amazon API Gateway dans votreCompte AWS. Vous pouvez créer un domaine personnalisé dans la console API Gateway.

Prérequis :

- Tu as unCompte AWS.
- Vous disposez d'une instance d'AWS ParallelClusterinterface utilisateur à laquelle vous pouvez accéder.
- Vous êtes propriétaire d'un domaine.
- Vous pouvez modifier les paramètres de base du système de noms de domaine (DNS).

Étape 1 : créer un nouveau domaine dans Amazon API Gateway

1. Dans leAWS Management Console, accédez à [API Gateway](#) où vous pouvez voir votre API d'AWS ParallelClusterinterface utilisateur répertoriée.
2. Dans le panneau de navigation, sélectionnez Custom domain names (Noms de domaine personnalisés).
3. Choisissez Créer.
4. Dans Détails du domaine, entrez votre nom de domaine.
5. Dans Configuration du point de terminaison, choisissez un certificat ACM existant ou choisissez Créer un nouveau certificat ACM.

(Facultatif) Créez un certificat

- a. Dans la console ACM, choisissez Request.
- b. Dans Noms de domaine, entrez votre nom de domaine.
- c. Dans Méthode de validation, choisissez une méthode de validation.

Si vous choisissez la validation par e-mail, un e-mail est envoyé à l'adresse e-mail enregistrée auprès du bureau d'enregistrement de domaines.

- d. Sélectionnez J'approuve pour activer le certificat.

Étape 2 : configurer les mappages d'API

1. Dans [API Gateway](#), Noms de domaine personnalisés your-domain-name, sélectionnez Configurer les mappages d'API.
2. Choisissez Noms de domaine personnalisés.
3. Choisissez Ajouter un nouveau mappage.
4. Choisissez l'API AWS ParallelCluster UI, le stage \$default et Save.
5. Dans le nom de domaine API Gateway, copiez la valeur à utiliser lors des étapes suivantes.

Étape 3 : configurer le DNS

- Créez une règle DNS CNAME qui pointe votre domaine vers le domaine API Gateway. Entrez uniquement le domaine. Par exemple, n'ajoutez pas l'étape, telle que beta ouprod. Remplacez *abcde12345* par votre ID d'API Gateway et remplacez *us-east-2* par l'API. Région AWS

| Règle | Source | Destination |
|-------|--------------------|--|
| CNAME | <i>example.com</i> | d- <i>abcde12345</i> <i>.execute-api.us-east-2</i> <i>.amazonaws.com</i> |

Étape 4 : ajouter le domaine à votre groupe d'utilisateurs Amazon Cognito

1. Accédez à la [console Amazon Cognito](#).
2. Sélectionnez le lien de votre groupe d'utilisateurs.
3. Choisissez Intégration des applications.
4. Dans Domaine, choisissez Actions, puis Créer un domaine personnalisé.
5. Entrez votre domaine personnalisé et sélectionnez votre certificat ACM.
6. Choisissez Créer un domaine personnalisé.

Étape 5 : Configuration de l'URL de rappel d'API Gateway

1. Accédez à la [console Amazon Cognito](#).
2. Dans votre groupe d'utilisateurs Amazon Cognito (intégration des applications, clients d'applications et analyses), sélectionnez le lien de l'application.
3. Dans l'interface utilisateur hébergée, choisissez Modifier.
4. Dans URL de rappel autorisées, choisissez Ajouter une autre URL et entrez une URL de rappel telle que `example.com/login`

Étape 6 : Configuration de la fonction Lambda

1. Accédez à la console [Lambda](#).
2. Dans le volet de navigation, choisissez Fonctions.
3. Filtrez la liste des fonctions pour les trouver `ParallelClusterUIFunction` et sélectionnez le lien.
4. Choisissez Configuration, Variables d'environnement.
5. Choisissez Modifier.

6. Pour la `SITE_URL` valeur, entrez votre domaine personnalisé.
7. Accédez à votre domaine, par exemple `example.com`, et authentifiez-vous pour vous connecter à l'AWS ParallelCluster interface utilisateur.

Options du groupe d'utilisateurs Amazon Cognito

Les sections suivantes font référence à la CloudFormation création rapide de liens ou à la création rapide d'URL. L'URL de création rapide vous dirige vers un assistant de création de pile dans lequel vous fournissez des entrées de modèle de pile pour créer rapidement et déployer la pile. Pour plus d'informations sur la CloudFormation création rapide de piles, voir [Création de liens de création rapide pour les piles](#) dans le Guide de l'utilisateur. AWS CloudFormation

Pour gérer un groupe d'utilisateurs Amazon Cognito que vous pouvez utiliser avec plusieurs instances d'AWS ParallelCluster interface utilisateur, envisagez les options suivantes :

- Utilisez une instance d'AWS ParallelCluster interface utilisateur existante qui renvoie à un groupe d'utilisateurs Amazon Cognito créé à partir d'une pile imbriquée CloudFormation . C'est ce qui est créé lorsque vous déployez l'AWS ParallelCluster interface utilisateur à l'aide du lien de création rapide et que vous laissez tous les paramètres Amazon Cognito vides.
- Utilisez un groupe d'utilisateurs Amazon Cognito autonome déployé avant le déploiement de l'AWS ParallelCluster l'interface utilisateur. Déployez ensuite une nouvelle instance d'AWS ParallelCluster interface utilisateur liée au groupe d'utilisateurs autonome Amazon Cognito que vous avez déjà déployé. Ainsi, vous séparez le déploiement d'Amazon Cognito du déploiement de l'AWS ParallelCluster interface utilisateur. De plus, les CloudFormation piles d'AWS ParallelCluster interface utilisateur non imbriquées sont plus faciles à mettre à jour.

Utiliser un groupe d'utilisateurs Amazon Cognito existant avec une nouvelle AWS ParallelCluster instance d'interface utilisateur

1. Dans la CloudFormation console, sélectionnez la pile d'AWS ParallelCluster interface utilisateur contenant le groupe d'utilisateurs Amazon Cognito que vous souhaitez utiliser avec plusieurs instances d'AWS ParallelCluster interface utilisateur.
2. Accédez à la pile imbriquée qui a créé le pool d'utilisateurs Amazon Cognito.
3. Sélectionnez l'onglet Outputs (Sorties).
4. Copiez les valeurs des paramètres suivants :

- `UserPoolId`
 - `UserPoolAuthDomain`
 - `SNSRole`
5. Déployez une nouvelle instance d'AWS ParallelClusterinterface utilisateur à l'aide du lien de création rapide et renseignez tous les External AWS ParallelCluster UI Amazon Cognito paramètres avec les sorties que vous avez copiées. Cela empêche la nouvelle pile d'AWS ParallelClusterinterface utilisateur de créer un nouveau pool et le lie au groupe d'utilisateurs Amazon Cognito existant créé à partir d'une pile imbriquée. Vous pouvez déployer de nouvelles instances d'AWS ParallelClusterinterface utilisateur suivantes qui ont les mêmes valeurs de paramètres, et vous pouvez les lier au groupe d'utilisateurs Amazon Cognito.

Création d'un groupe d'utilisateurs Amazon Cognito autonome

AWS ParallelClusterInterface utilisateur Amazon Cognito : création rapide de liens par région

Interface utilisateur Amazon Cognito : création rapide de liens

[us-east-1](#)

[us-east-2](#)

[us-west-1](#)

[us-west-2](#)

[eu-west-1](#)

[eu-west-2](#)

[eu-west-3](#)

[eu-central-1](#)

[eu-north-1](#)

[me-south-1](#)

[sa-east-1](#)

Interface utilisateur Amazon Cognito : création rapide de liens

[ca-central-1](#)

[ap-northeast-1](#)

[ap-northeast-2](#)

[ap-south-1](#)

[ap-southeast-1](#)

[ap-southeast-2](#)

[us-gov-west-1](#)

1. Lancez une pile réservée à Amazon Cognito en choisissant un lien de création rapide portant le même nom Région AWS dans lequel vous déployez vos instances d'interface utilisateur. AWS ParallelCluster Consultez les liens de création rapide au début de cette section.
2. Une fois la création de la pile terminée, sélectionnez l'onglet Sorties et copiez les valeurs des paramètres suivants :
 - `UserPoolId`
 - `UserPoolAuthDomain`
 - `SNSRole`
3. Déployez une nouvelle instance d'AWS ParallelClusterinterface utilisateur en choisissant un lien de démarrage rapide de l'AWS ParallelClusterinterface utilisateur et en remplissant tous les `External AWS ParallelCluster UI Amazon Cognito` paramètres avec les valeurs que vous avez copiées. La nouvelle instance d'AWS ParallelClusterinterface utilisateur est liée au groupe d'utilisateurs autonome Amazon Cognito et ne crée pas de pile imbriquée ni de nouveau groupe d'utilisateurs. Vous pouvez déployer de nouvelles instances d'AWS ParallelClusterinterface utilisateur suivantes qui ont les mêmes valeurs de paramètres, et vous pouvez les lier au groupe d'utilisateurs autonome Amazon Cognito.

Identifier la version AWS ParallelCluster et la version de AWS ParallelCluster l'interface utilisateur

1. Dans la CloudFormation console, sélectionnez une pile d'AWS ParallelCluster interface utilisateur.
2. Sélectionnez l'onglet Paramètres.
3. La AWS ParallelCluster version est la valeur du paramètre Version.
4. La version de l'AWS ParallelCluster interface utilisateur se trouve à la fin de la `PublicEcrImageUri` valeur. Par exemple, si la valeur est `public.ecr.aws/pcui/parallelcluster-ui-awslambda:2023.02`, la version est `2023.02`.

Mettre à jour l'AWS ParallelCluster interface utilisateur vers une nouvelle AWS ParallelCluster version

Pour mettre à jour l'AWS ParallelCluster interface utilisateur vers la dernière AWS ParallelCluster version, lancez une nouvelle pile en choisissant un lien de [création rapide](#).

AWS ParallelCluster Coûts d'interface utilisateur

L'AWS ParallelCluster interface utilisateur repose sur une architecture sans serveur et vous pouvez l'utiliser dans la catégorie AWS Free Tier dans la plupart des cas. Le tableau suivant répertorie les éléments services AWS dont dépend l'AWS ParallelCluster interface utilisateur et leurs limites en matière de niveau gratuit. Le coût d'utilisation typique est estimé à moins d'un dollar par mois.

| Service | AWS Free Tier |
|------------------------------|--|
| Amazon Cognito | 50 000 utilisateurs actifs par mois |
| Amazon API Gateway | 1 million d'appels d'API REST |
| AWS Lambda | 1 million de requêtes gratuites par mois et 400 000 Go de temps de calcul par mois |
| EC2 Image Builder | Sans frais, sauf EC2 |
| Amazon Elastic Compute Cloud | Création d'une image de conteneur unique en 15 minutes |

| Service | AWS Free Tier |
|--------------------|---|
| AWS CloudFormation | 5 Go de données (ingestion, stockage d'archives et données scannées par les requêtes Logs Insights) |

Commencer avec AWS ParallelCluster

Commencez par configurer et créer un cluster à l'aide de l'interface de ligne de AWS ParallelCluster commande (CLI) ou de l'interface utilisateur Web (UI). L' AWS ParallelCluster interface utilisateur a été ajoutée dans la version 3.5.0.

Rubriques

- [Configuration et création d'un cluster à l'aide de l'interface de ligne de AWS ParallelCluster commande](#)
- [Configuration et création d'un cluster avec l' AWS ParallelCluster interface utilisateur](#)
- [Connexion à un cluster](#)

Configuration et création d'un cluster à l'aide de l'interface de ligne de AWS ParallelCluster commande

Après l'installation AWS ParallelCluster, effectuez les étapes de configuration suivantes.

Vérifiez que votre AWS compte possède un rôle qui inclut les autorisations nécessaires pour exécuter la [pcluster](#) CLI. Pour plus d'informations, consultez [AWS ParallelCluster exemples de politiques pcluster utilisateur](#).

Configurez vos AWS informations d'identification. Pour plus d'informations, consultez [Configuration de l' AWS CLI](#) dans le Guide de l'utilisateur de l'AWS CLI .

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
Default region name [us-east-1]: us-east-1
Default output format [None]:
```

L' Région AWS endroit où le cluster est lancé doit comporter au moins une paire de clés Amazon EC2. Pour plus d'informations, consultez les [paires de clés Amazon Elastic Compute Cloud](#) dans le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux.

Lorsque vous utilisez l'interface de ligne de commande (CLI) de AWS ParallelCluster, vous ne payez que pour les AWS ressources créées lorsque vous créez ou mettez à jour AWS ParallelCluster des images et des clusters. Pour plus d'informations, consultez [AWS services utilisés par AWS ParallelCluster](#).

Configurez et créez votre premier cluster

Créez votre premier cluster à l'aide de la commande `pcluster configure` CLI pour lancer un assistant qui vous invite à saisir toutes les informations nécessaires à la configuration et à la création de votre cluster. Les détails de la séquence diffèrent lorsqu'elle est utilisée AWS Batch comme planificateur par rapport à son utilisation. Slurm

Slurm

```
$ pcluster configure --config config-file.yaml
```

Dans la liste des Région AWS identifiants valides, choisissez l' Région AWS endroit où vous souhaitez que votre cluster s'exécute.

Note

La liste des Régions AWS éléments affichés est basée sur la partition de votre compte et inclut uniquement Régions AWS les partitions activées pour votre compte. Pour plus d'informations sur l'activation Régions AWS de votre compte, consultez [la section Gestion Régions AWS](#) dans le Références générales AWS. L'exemple présenté provient de la partition AWS globale. Si votre compte se trouve dans la AWS GovCloud (US) partition, seule Régions AWS cette partition est répertoriée (`gov-us-east-1` et `gov-us-west-1`). De même, si votre compte se trouve dans la partition AWS Chine, uniquement `cn-north-1` et `cn-northwest-1` s'affiche. Pour la liste complète des produits Régions AWS pris en charge par AWS ParallelCluster, voir [Régions prises en charge pour AWS ParallelCluster](#).

Allowed values for Région AWS ID:

```
1. af-south-1
2. ap-east-1
3. ap-northeast-1
4. ap-northeast-2
5. ap-south-1
6. ap-southeast-1
7. ap-southeast-2
8. ca-central-1
9. eu-central-1
10. eu-north-1
11. eu-south-1
12. eu-west-1
13. eu-west-2
14. eu-west-3
15. me-south-1
16. sa-east-1
17. us-east-1
18. us-east-2
19. us-west-1
20. us-west-2
Région AWS ID [ap-northeast-1]:
```

La paire de clés est sélectionnée parmi les paires de clés enregistrées auprès d'Amazon Elastic Compute Cloud dans la zone sélectionnée Région AWS. Choisissez la paire de clés :

```
Allowed values for Amazon EC2 Key Pair Name:
1. your-key-1
2. your-key-2
Amazon EC2 Key Pair Name [your-key-1]:
```

Choisissez le planificateur à utiliser avec votre cluster.

```
Allowed values for Scheduler:
1. slurm
2. awsbatch
Scheduler [slurm]:
```

Choisissez le système d'exploitation.

```
Allowed values for Operating System:
1. alinux2
2. centos7
```

```
3. ubuntu2204
4. ubuntu2004
5. rhel8
Operating System [alinux2]:
```

Choisissez le type d'instance du nœud principal :

```
Head node instance type [t2.micro]:
```

Choisissez la configuration de la file d'attente. Remarque : Le type d'instance ne peut pas être spécifié pour plusieurs ressources de calcul dans la même file d'attente.

```
Number of queues [1]:
Name of queue 1 [queue1]:
Number of compute resources for queue1 [1]: 2
Compute instance type for compute resource 1 in queue1 [t2.micro]:
Maximum instance count [10]:
```

Permettez à EFA d'exécuter des applications nécessitant des niveaux élevés de communication entre instances à grande échelle AWS , sans frais supplémentaires :

- Choisissez un type d'instance [compatible avec Elastic Fabric Adapter \(EFA\)](#).
- Activez [EFA](#).
- Spécifiez un nom de [groupe de placement](#) existant. Si vous le laissez vide, AWS ParallelCluster créez-en un pour vous.


```
Compute instance type for compute resource 2 in queue1 [t2.micro]: c5n.18xlarge
Enable EFA on c5n.18xlarge (y/n) [y]: y
Maximum instance count [10]:
Placement Group name []:
```

Une fois les étapes précédentes terminées, décidez d'utiliser un VPC existant ou de laisser AWS ParallelCluster créer un VPC pour vous. Si vous ne disposez pas d'un VPC correctement configuré, vous pouvez en créer un nouveau pour vous. Il place soit les nœuds de tête et de calcul dans le même sous-réseau public, soit uniquement le nœud principal dans un sous-réseau public avec tous les nœuds de calcul dans un sous-réseau privé. Si vous autorisez la création d'un VPC, vous devez décider si tous les nœuds

doivent se trouver dans un sous-réseau public. Pour plus d'informations, consultez [Configurations réseau](#).


Si vous configurez votre cluster pour utiliser des types d'instances dotés de plusieurs interfaces réseau ou d'une carte réseau, consultez [Configurations réseau](#) les exigences réseau supplémentaires.

Il est possible d'atteindre votre quota pour le nombre de VPC autorisés dans un Région AWS. Le quota par défaut est de cinq VPC pour un Région AWS. Pour plus d'informations sur ce quota et sur la manière de demander une augmentation, consultez la section [VPC et sous-réseaux](#) dans le guide de l'utilisateur Amazon VPC.

 Important

Les VPC créés par AWS ParallelCluster n'activent pas les journaux de flux VPC par défaut. Les journaux de flux VPC vous permettent de capturer des informations sur le trafic IP à destination et en provenance des interfaces réseau de vos VPC. Pour plus d'informations, consultez la rubrique [Journaux de flux VPC](#) dans le Guide de l'utilisateur Amazon VPC.

Si vous autorisez la AWS ParallelCluster création d'un VPC, assurez-vous de décider si tous les nœuds doivent se trouver dans un sous-réseau public.

 Note

Si vous le souhaitez¹. Head node in a public subnet and compute fleet in a private subnet, AWS ParallelCluster crée une passerelle NAT qui entraîne des coûts supplémentaires, même si vous spécifiez des ressources de niveau gratuit.

```
Automate VPC creation? (y/n) [n]: y
Allowed values for Availability Zone:
1. us-east-1a
2. us-east-1b
3. us-east-1c
4. us-east-1d
5. us-east-1e
6. us-east-1f
Availability Zone [us-east-1a]:
```

Allowed values for Network Configuration:

1. Head node in a public subnet and compute fleet in a private subnet
2. Head node and compute fleet in the same public subnet

Network Configuration [Head node in a public subnet and compute fleet in a private subnet]: 1

Beginning VPC creation. Please do not leave the terminal until the creation is finalized

Si vous ne créez pas de nouveau VPC, vous devez sélectionner un VPC existant.

Si vous choisissez de AWS ParallelCluster créer le VPC, notez l'ID du VPC afin de pouvoir l'utiliser AWS CLI pour le supprimer ultérieurement.

Automate VPC creation? (y/n) [n]: n

Allowed values for VPC ID:

| # | id | name | number_of_subnets |
|---|-----------------------|-----------------------------------|-------------------|
| 1 | vpc-0b4ad9c4678d3c7ad | ParallelClusterVPC-20200118031893 | 2 |
| 2 | vpc-0e87c753286f37eef | ParallelClusterVPC-20191118233938 | 5 |

VPC ID [vpc-0b4ad9c4678d3c7ad]: 1

Une fois le VPC sélectionné, décidez si vous souhaitez utiliser des sous-réseaux existants ou en créer de nouveaux.

Automate Subnet creation? (y/n) [y]: y

Creating CloudFormation stack...

Do not leave the terminal until the process has finished

AWS Batch

```
$ pcluster configure --config config-file.yaml
```

Dans la liste des Région AWS identifiants valides, choisissez l' Région AWS endroit où vous souhaitez que votre cluster s'exécute.

Note

La liste Régions AWS affichée est basée sur la partition de votre compte. Cela inclut uniquement Régions AWS ceux qui sont activés pour votre compte. Pour plus d'informations sur l'activation Régions AWS de votre compte, consultez [la section Gestion](#)

[Régions AWS](#) dans le Références générales AWS. L'exemple présenté provient de la partition AWS globale. Si votre compte se trouve dans la AWS GovCloud (US) partition, seule Régions AWS cette partition est répertoriée (gov-us-east-1 et gov-us-west-1). De même, si votre compte se trouve dans la partition AWS Chine, uniquement cn-north-1 et cn-northwest-1 s'affiche. Pour la liste complète des produits Régions AWS pris en charge par AWS ParallelCluster, voir [Régions prises en charge pour AWS ParallelCluster](#).

Allowed values for Région AWS ID:

1. af-south-1
2. ap-east-1
3. ap-northeast-1
4. ap-northeast-2
5. ap-south-1
6. ap-southeast-1
7. ap-southeast-2
8. ca-central-1
9. eu-central-1
10. eu-north-1
11. eu-south-1
12. eu-west-1
13. eu-west-2
14. eu-west-3
15. me-south-1
16. sa-east-1
17. us-east-1
18. us-east-2
19. us-west-1
20. us-west-2

Région AWS ID [us-east-1]:

La paire de clés est sélectionnée parmi les paires de clés enregistrées auprès d'Amazon EC2 dans le champ sélectionné. Région AWS Choisissez la paire de clés :

Allowed values for Amazon EC2 Key Pair Name:

1. your-key-1
2. your-key-2

Amazon EC2 Key Pair Name [your-key-1]:

Choisissez le planificateur à utiliser avec votre cluster.

Allowed values for Scheduler:

1. slurm
 2. awsbatch
- Scheduler [slurm]: 2

Lorsque `awsbatch` est sélectionné comme planificateur, `alinux2` est utilisé comme système d'exploitation. Le type d'instance du nœud principal est saisi :

Head node instance type [t2.micro]:

Choisissez la configuration de la file d'attente. Le AWS Batch planificateur ne contient qu'une seule file d'attente. La taille maximale du cluster de nœuds de calcul est saisie. Elles sont mesurées dans les vCPU.

Number of queues [1]:
Name of queue 1 [queue1]:
Maximum vCPU [10]:

Décidez si vous souhaitez utiliser des VPC existants ou laisser AWS ParallelCluster créer des VPC pour vous. Si vous n'avez pas de VPC correctement configuré, AWS ParallelCluster peut en créer un nouveau. Il utilise soit les nœuds de tête et de calcul du même sous-réseau public, soit uniquement le nœud principal d'un sous-réseau public avec tous les nœuds d'un sous-réseau privé. Il est possible d'atteindre votre quota en ce qui concerne le nombre de VPC autorisés dans une région. Le nombre par défaut de VPC est de cinq. Pour plus d'informations sur ce quota et sur la manière de demander une augmentation, consultez la section [VPC et sous-réseaux](#) dans le guide de l'utilisateur Amazon VPC.

Important

Les VPC créés par AWS ParallelCluster n'activent pas les journaux de flux VPC par défaut. Les journaux de flux VPC vous permettent de capturer des informations sur le trafic IP à destination et en provenance des interfaces réseau de vos VPC. Pour plus d'informations, consultez la rubrique [Journaux de flux VPC](#) dans le Guide de l'utilisateur Amazon VPC.

Si vous autorisez la AWS ParallelCluster création d'un VPC, assurez-vous de décider si tous les nœuds doivent se trouver dans un sous-réseau public.

Note

Si vous le souhaitez¹. Head node in a public subnet and compute fleet in a private subnet, AWS ParallelCluster crée une passerelle NAT qui entraîne des coûts supplémentaires, même si vous spécifiez des ressources de niveau gratuit.

```
Automate VPC creation? (y/n) [n]: y
Allowed values for Availability Zone:
1. us-east-1a
2. us-east-1b
3. us-east-1c
4. us-east-1d
5. us-east-1e
6. us-east-1f
Availability Zone [us-east-1a]:
Allowed values for Network Configuration:
1. Head node in a public subnet and compute fleet in a private subnet
2. Head node and compute fleet in the same public subnet
Network Configuration [Head node in a public subnet and compute fleet in a private
subnet]: *1*
Beginning VPC creation. Please do not leave the terminal until the creation is
finalized
```

Si vous ne créez pas de nouveau VPC, vous devez sélectionner un VPC existant.

Si vous choisissez de AWS ParallelCluster créer le VPC, notez l'ID du VPC afin de pouvoir utiliser le AWS CLI ou AWS Management Console pour le supprimer ultérieurement.

```
Automate VPC creation? (y/n) [n]: n
Allowed values for VPC ID:
#  id                                     name                                     number_of_subnets
---  -----
1  vpc-0b4ad9c4678d3c7ad  ParallelClusterVPC-20200118031893  2
2  vpc-0e87c753286f37eef  ParallelClusterVPC-20191118233938  5
VPC ID [vpc-0b4ad9c4678d3c7ad]: 1
```

Une fois le VPC sélectionné, assurez-vous de décider d'utiliser les sous-réseaux existants ou d'en créer de nouveaux.

```
Automate Subnet creation? (y/n) [y]: y
```

```
Creating CloudFormation stack...  
Do not leave the terminal until the process has finished
```

Lorsque vous avez terminé les étapes précédentes, un simple cluster est lancé dans un VPC. Le VPC utilise un sous-réseau existant qui prend en charge les adresses IP publiques. La table de routage pour le sous-réseau est `0.0.0.0/0 => igw-xxxxxx`. Notez les conditions suivantes :

- Le VPC doit comporter les paramètres DNS Resolution = yes et DNS Hostnames = yes.
- Le VPC doit également disposer d'options DHCP dont la valeur est correcte domain-name pour Région AWS. Le jeu d'options DHCP par défaut spécifie déjà le AmazonProvidedDNS requis. Si vous spécifiez plusieurs serveurs de noms de domaine, consultez les [ensembles d'options DHCP](#) dans le guide de l'utilisateur Amazon VPC. Lorsque vous utilisez des sous-réseaux privés, utilisez une passerelle NAT ou un proxy interne pour permettre l'accès Web aux nœuds de calcul. Pour plus d'informations, consultez [Configurations réseau](#).

Une fois que tous les paramètres contiennent les valeurs valides, vous pouvez lancer le cluster en exécutant la commande create.

```
$ pcluster create-cluster --cluster-name test-cluster --cluster-configuration cluster-  
config.yaml  
{  
  "cluster": {  
    "clusterName": "test-cluster",  
    "cloudformationStackStatus": "CREATE_IN_PROGRESS",  
    "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/test-cluster/  
abcdef0-f678-890a-5abc-021345abcdef",  
    "region": "eu-west-1",  
    "version": "3.7.0",  
    "clusterStatus": "CREATE_IN_PROGRESS"  
  },  
  "validationMessages": []  
}
```

Suivez la progression du cluster :

```
$ pcluster describe-cluster --cluster-name test-cluster
```

or

```
$ pcluster list-clusters --query 'clusters[?clusterName==`test-cluster`]'
```

Une fois que le cluster a atteint le "clusterStatus": "CREATE_COMPLETE" statut, vous pouvez vous y connecter en utilisant les paramètres habituels de votre client SSH. Pour plus d'informations sur la connexion aux instances Amazon EC2, consultez le guide de l'utilisateur [Amazon EC2 dans le guide de l'utilisateur](#) Amazon EC2. Vous pouvez également connecter le cluster via

```
$ pcluster ssh --cluster-name test-cluster -i ~/path/to/keyfile.pem
```

Pour supprimer le cluster, exécutez la commande suivante.

```
$ pcluster delete-cluster --region us-east-1 --cluster-name test-cluster
```

Une fois le cluster supprimé, vous pouvez supprimer les ressources réseau du VPC en supprimant la pile CloudFormation réseau. Le nom de la pile commence par « parallelclusternetworking- » et contient l'heure de création au format « YYYYMMDDHHMMSS ». Vous pouvez répertorier les piles à l'aide de la [list-stacks](#) commande.

```
$ aws --region us-east-1 cloudformation list-stacks \  
  --stack-status-filter "CREATE_COMPLETE" \  
  --query "StackSummaries[].StackName" | \  
  grep -e "parallelclusternetworking-" \  
  "parallelclusternetworking-pubpriv-20191029205804"
```

La pile peut être supprimée à l'aide de la [delete-stack](#) commande.

```
$ aws --region us-east-1 cloudformation delete-stack \  
  --stack-name parallelclusternetworking-pubpriv-20191029205804
```

Le VPC qui [pcluster configure](#) crée pour vous n'est pas créé dans la pile CloudFormation réseau. Vous pouvez supprimer ce VPC manuellement dans la console ou à l'aide du AWS CLI

```
$ aws --region us-east-1 Amazon EC2 delete-vpc --vpc-id vpc-0b4ad9c4678d3c7ad
```

Configuration et création d'un cluster avec l' AWS ParallelCluster interface utilisateur

L' AWS ParallelCluster interface utilisateur est une interface utilisateur Web qui reflète la AWS ParallelCluster `pccluster` CLI, tout en offrant une expérience semblable à celle d'une console. Vous installez et accédez à l' AWS ParallelCluster interface utilisateur dans votre Compte AWS. Lorsque vous l'exécutez, l' AWS ParallelCluster interface utilisateur accède à une instance de l' AWS ParallelCluster API hébergée sur Amazon API Gateway dans votre Compte AWS.

Note

L'assistant d' AWS ParallelCluster interface utilisateur ne dispose peut-être pas d'options d'interface utilisateur pour toutes les fonctionnalités prises en charge dans la dernière AWS ParallelCluster version prise en charge. Vous pouvez modifier manuellement le fichier de configuration selon vos besoins ou utiliser la AWS ParallelCluster CLI.

Dans cette section, nous vous expliquons comment configurer et créer un cluster à l'aide de l' AWS ParallelCluster interface utilisateur.

Prérequis :

- Accès à une instance en cours d'exécution de l' AWS ParallelCluster interface utilisateur. Pour plus d'informations, consultez [Installation de l'AWS ParallelCluster interface utilisateur](#).

Configuration et création d'un cluster

1. Dans la vue AWS ParallelCluster UI Clusters, choisissez Create cluster, Step by step.
2. Dans Cluster, Nom, entrez le nom de votre cluster.
3. Choisissez un VPC doté d'un sous-réseau public pour votre cluster, puis cliquez sur Next.
4. Dans le nœud Head, choisissez Ajouter une session SSM, puis Next.
5. Dans Queues, Ressources de calcul, choisissez 1 pour les nœuds statiques.
6. Pour Type d'instance, supprimez le type d'instance par défaut sélectionné, choisissez t2.micro, puis Next.
7. Dans Stockage, choisissez Next.

8. Dans Configuration du cluster, passez en revue le code YAML de configuration du cluster et choisissez Dry run pour le valider.
9. Choisissez Create pour créer votre cluster, en fonction de la configuration validée.
10. Après quelques secondes, l' AWS ParallelCluster interface utilisateur vous redirige automatiquement vers Clusters, où vous pouvez surveiller l'état de création du cluster et les événements Stack.
11. Choisissez Détails pour voir les détails du cluster, tels que la version et le statut.
12. Choisissez Instances pour voir la liste des instances Amazon EC2 et leur statut.
13. Choisissez Stack events pour afficher les événements de stack de clusters, ainsi qu'un AWS Management Console lien vers la CloudFormation pile qui crée le cluster.
14. Dans Détails, une fois la création du cluster terminée, choisissez View YAML pour afficher ou télécharger le fichier YAML de configuration du cluster.
15. Une fois la création du cluster terminée, choisissez Shell pour accéder au nœud principal du cluster.

Note

Lorsque vous choisissez Shell, vous AWS ParallelCluster ouvrez une session Amazon EC2 Systems Manager et ajoutez ssm-user un /etc/sudoers à. Pour plus d'informations, consultez [Activer ou désactiver les autorisations administratives du ssm-user compte](#) dans le guide de l'utilisateur d'Amazon EC2 Systems Manager.

16. Pour nettoyer, dans la vue Clusters, sélectionnez le cluster, puis choisissez Actions, Supprimer le cluster.

Connexion à un cluster

Lors de l'utilisation AWS ParallelCluster, vous pouvez vous connecter au nœud principal du cluster pour exécuter des tâches, afficher les résultats, gérer les utilisateurs et surveiller le cluster et l'état des tâches. Connectez-vous à l'instance du nœud principal du cluster en utilisant les méthodes suivantes :

- Connectez-vous à l'aide ssh d'une [paire de clés](#). Spécifiez la clé privée dans [HeadNode/KeyName](#) dans la configuration du cluster. Pour plus d'informations, consultez [Connect to Your Linux Instance \(Connexion à votre instance Linux\) à l'aide du protocole SSH](#) dans le Guide de l'utilisateur Amazon EC2 pour instances Linux.

- Connectez-vous à l'aide de la `pcluster ssh` commande de l'interface de ligne de commande (CLI). Spécifiez la clé privée dans la configuration du cluster [HeadNode/KeyName](#). Pour plus d'informations, consultez [pcluster ssh](#).
- Connectez-vous au nœud principal du cluster à l'aide d'une session SSM. Vous devez ajouter la politique `AmazonSSMManagedInstanceCore` gérée à [HeadNode/AdditionalIamPolicies](#) dans la configuration du cluster pour vous connecter à l'aide d'une session SSM. Pour plus d'informations, consultez la section [Gestionnaire de sessions SSM](#) dans le guide de l'utilisateur SSM.
- Connectez-vous au nœud principal du cluster à l'aide de NICE DCV. Pour plus d'informations, consultez [Connectez-vous au nœud principal via NICE DCV](#).
- Lorsque vous utilisez l' AWS ParallelCluster interface utilisateur, vous pouvez également vous connecter au nœud principal du cluster à l'aide d'une commande Amazon EC2 Connect fournie par l'interface utilisateur.

Accès de plusieurs utilisateurs aux clusters

Apprenez à implémenter et à gérer l'accès de plusieurs utilisateurs à un seul cluster.

Dans cette rubrique, un AWS ParallelCluster utilisateur fait référence à un utilisateur du système pour les instances de calcul. Un exemple est celui `ec2-user` d'une instance Amazon EC2.

AWS ParallelCluster le support d'accès multi-utilisateurs est disponible partout Régions AWS où il AWS ParallelCluster est actuellement disponible. Il fonctionne avec d'autres services AWS, notamment [Amazon FSx for Lustre](#) et [Amazon Elastic File System](#).

Vous pouvez utiliser un [AWS Directory Service for Microsoft Active Directory](#) ou [Simple AD](#) pour gérer l'accès au cluster. Assurez-vous de vérifier [Région AWS la disponibilité](#) de ces services. Pour configurer un cluster, spécifiez une [AWS ParallelCluster DirectoryService](#) configuration. AWS Directory Service les répertoires peuvent être connectés à plusieurs clusters. Cela permet une gestion centralisée des identités dans plusieurs environnements et une expérience de connexion unifiée.

Lorsque vous utilisez l'accès AWS Directory Service pour AWS ParallelCluster plusieurs utilisateurs, vous pouvez vous connecter au cluster à l'aide des informations d'identification utilisateur définies dans l'annuaire. Ces informations d'identification se composent d'un nom d'utilisateur et d'un mot de passe. Lorsque vous vous connectez au cluster pour la première fois, une clé SSH utilisateur est automatiquement générée. Vous pouvez l'utiliser pour vous connecter sans mot de passe.

Vous pouvez créer, supprimer et modifier les utilisateurs ou les groupes d'un cluster après le déploiement de votre service d'annuaire. Avec AWS Directory Service, vous pouvez le faire dans AWS Management Console ou à l'aide de l'outil Active Directory Users and Computers. Cet outil est accessible depuis n'importe quelle instance Amazon EC2 jointe à votre Active Directory. Pour plus d'informations, veuillez consulter [Installation des outils d'administration Active Directory](#) (langue française non garantie).

Si vous prévoyez de l'utiliser AWS ParallelCluster dans un seul sous-réseau sans accès à Internet, consultez [AWS ParallelCluster dans un seul sous-réseau sans accès à Internet](#) les exigences supplémentaires.

Rubriques

- [Création d'un Active Directory](#)
- [Création d'un cluster avec un domaine AD](#)
- [Connectez-vous à un cluster intégré à un domaine AD](#)
- [Exécution de tâches MPI](#)
- [Exemple AWS Managed Microsoft AD de configuration de cluster LDAP \(S\)](#)

Création d'un Active Directory

Assurez-vous de créer un Active Directory (AD) avant de créer votre cluster. Pour plus d'informations sur le choix du type d'Active Directory pour votre cluster, voir [Lequel choisir](#) dans le Guide d'AWS Directory Service administration.

Si le répertoire est vide, ajoutez des utilisateurs avec des noms d'utilisateur et des mots de passe. Pour plus d'informations, consultez la documentation spécifique à [AWS Directory Service for Microsoft Active DirectorySimple AD](#).

Note

AWS ParallelCluster exige que tous les annuaires d'utilisateurs Active Directory figurent dans le `/home/$user` répertoire.

Création d'un cluster avec un domaine AD

⚠ Warning

Cette section d'introduction décrit comment configurer AWS ParallelCluster un serveur Active Directory (AD) géré via le protocole LDAP (Lightweight Directory Access Protocol). Le protocole LDAP n'est pas sécurisé. Pour les systèmes de production, nous recommandons vivement l'utilisation de certificats TLS (LDAPS), comme décrit dans la [Exemple AWS Managed Microsoft AD de configuration de cluster LDAP \(S\)](#) section suivante.

Configurez votre cluster pour l'intégrer à un répertoire en spécifiant les informations pertinentes dans la `DirectoryService` section du fichier de configuration du cluster. Pour plus d'informations, consultez la section [DirectoryService](#) de configuration.

Vous pouvez utiliser l'exemple suivant pour intégrer votre cluster à un protocole LDAP (AWS Managed Microsoft AD Over the Lightweight Directory Access Protocol).

Définitions spécifiques requises pour une configuration AWS Managed Microsoft AD via LDAP :

- Vous devez définir le `ldap_auth_disable_tls_never_use_in_production` paramètre sur une valeur `True` inférieure à [DirectoryService/AdditionalSssdConfigs](#).
- Vous pouvez spécifier les noms d'hôte ou les adresses IP des contrôleurs pour [DirectoryService/DomainAddr](#).
- [DirectoryService](#) La `DomainReadOnlyUsers` syntaxe/doit être la suivante :

```
cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
```

Obtenez vos données AWS Managed Microsoft AD de configuration :

```
$ aws ds describe-directories --directory-id "d-abcdef01234567890"
```

```
{
  "DirectoryDescriptions": [
    {
      "DirectoryId": "d-abcdef01234567890",
      "Name": "corp.example.com",
```

```

        "DnsIpAddrs": [
            "203.0.113.225",
            "192.0.2.254"
        ],
        "VpcSettings": {
            "VpcId": "vpc-021345abcdef6789",
            "SubnetIds": [
                "subnet-1234567890abcdef0",
                "subnet-abcdef01234567890"
            ],
            "AvailabilityZones": [
                "region-idb",
                "region-idd"
            ]
        }
    ]
}

```

Configuration du cluster pour AWS Managed Microsoft AD :

```

Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-1234567890abcdef0
  Ssh:
    KeyName: pcluster
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: t2micro
          InstanceType: t2.micro
          MinCount: 1
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
DirectoryService:

```

```
DomainName: dc=corp,dc=example,dc=com
DomainAddr: ldap://203.0.113.225,ldap://192.0.2.254
PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:MicrosoftAD.Admin.Password-1234
DomainReadOnlyUser: cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
AdditionalSssdConfigs:
  ldap_auth_disable_tls_never_use_in_production: True
```

Pour utiliser cette configuration pour un Simple AD, modifiez la valeur de la **DomainReadOnlyUser** propriété dans la **DirectoryService** section :

```
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldap://203.0.113.225,ldap://192.0.2.254
  PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:SimpleAD.Admin.Password-1234
  DomainReadOnlyUser: cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
  AdditionalSssdConfigs:
    ldap_auth_disable_tls_never_use_in_production: True
```

Considérations :

- Nous vous recommandons d'utiliser le protocole LDAP sur TLS/SSL (ou LDAPS) plutôt que le protocole LDAP seul. Le protocole TLS/SSL garantit que la connexion est cryptée.
- La valeur de la [DomainAddr](#) propriété [DirectoryService](#)/correspond aux entrées de la `DnsIpAddrs` liste à partir de la `describe-directories` sortie.
- Nous recommandons que votre cluster utilise des sous-réseaux situés dans la même zone de disponibilité que celle vers laquelle [DomainAddr](#) pointe le [DirectoryService](#)//. Si vous utilisez une [configuration DHCP \(Dynamic Host Configuration Protocol\) personnalisée](#) recommandée pour les VPC d'annuaire et que vos sous-réseaux ne sont pas situés dans la zone [DirectoryService](#)/de [DomainAddr](#) disponibilité, le trafic croisé entre les zones de disponibilité est possible. L'utilisation de configurations DHCP personnalisées n'est pas requise pour utiliser la fonctionnalité d'intégration AD multi-utilisateurs.
- La valeur de la [DomainReadOnlyUser](#) propriété [DirectoryService](#)/indique un utilisateur qui doit être créé dans le répertoire. Cet utilisateur n'est pas créé par défaut. Nous vous recommandons de ne pas autoriser cet utilisateur à modifier les données du répertoire.
- La valeur de la [PasswordSecretArn](#) propriété [DirectoryService](#)/pointe vers un AWS Secrets Manager secret qui contient le mot de passe de l'utilisateur que vous avez spécifié

pour la [DomainReadOnlyUser](#) propriété [DirectoryService](#)/. Si le mot de passe de cet utilisateur change, mettez à jour la valeur secrète et mettez à jour le cluster. Pour mettre à jour le cluster en fonction de la nouvelle valeur secrète, vous devez arrêter le parc de calcul à l'aide de la `pcluster update-compute-fleet` commande. Si vous avez configuré votre cluster pour l'utiliser [LoginNodes](#), arrêtez le [LoginNodes/Pool](#)set mettez-le à jour après avoir défini [LoginNodesPools//Count](#)sur 0. Exécutez ensuite la commande suivante depuis le nœud principal du cluster.

```
sudo /opt/parallelcluster/scripts/directory_service/  
update_directory_service_password.sh
```

Pour un autre exemple, voir également [Intégration d'Active Directory](#).

Connectez-vous à un cluster intégré à un domaine AD

Si vous avez activé la fonctionnalité d'intégration du domaine Active Directory (AD), l'authentification par mot de passe est activée sur le nœud principal du cluster. Le répertoire personnel d'un utilisateur AD est créé lorsque le premier utilisateur se connecte au nœud principal ou lorsqu'il sudo-user passe pour la première fois à l'utilisateur AD sur le nœud principal.

L'authentification par mot de passe n'est pas activée pour les nœuds de calcul du cluster. Les utilisateurs d'AD doivent se connecter aux nœuds de calcul à l'aide de clés SSH.

Par défaut, les clés SSH sont configurées dans le `/${HOME}/.ssh` répertoire des utilisateurs AD lors de la première connexion SSH au nœud principal. Ce comportement peut être désactivé en attribuant à la propriété [DirectoryService/GenerateSshKeysForUsers](#)boolean la valeur `false` dans la configuration du cluster. Par défaut, [DirectoryService/GenerateSshKeysForUsers](#)est défini sur `true`.

Si une AWS ParallelCluster application nécessite le protocole SSH sans mot de passe entre les nœuds du cluster, assurez-vous que les clés SSH sont correctement configurées dans le répertoire personnel de l'utilisateur.

AWS Managed Microsoft AD les mots de passe expirent au bout de 42 jours. Pour plus d'informations, consultez la section [Gérer les politiques de mot de passe](#) AWS Managed Microsoft AD dans le Guide d'AWS Directory Service administration. Si votre mot de passe expire, il doit être réinitialisé pour rétablir l'accès au cluster. Pour plus d'informations, consultez [Comment réinitialiser le mot de passe d'un utilisateur et les mots de passe expirés](#).

Note

Si la fonctionnalité d'intégration AD ne fonctionne pas comme prévu, les journaux SSSD peuvent fournir des informations de diagnostic utiles pour résoudre le problème. Ces journaux se trouvent dans le `/var/log/sss` répertoire des nœuds du cluster. Par défaut, ils sont également stockés dans le groupe de CloudWatch journaux Amazon d'un cluster.

Pour plus d'informations, consultez [Résolution des problèmes d'intégration multi-utilisateurs avec Active Directory](#).

Exécution de tâches MPI

Comme suggéré dans SchedMD, amorcez les tâches MPI en utilisant Slurm la méthode d'amorçage MPI. Pour plus d'informations, reportez-vous à la [Slurmdocumentation](#) officielle ou à la documentation officielle de votre bibliothèque MPI.

Par exemple, dans la [documentation officielle d'IntelMPI](#), vous apprenez que lors de l'exécution d'une tâche StarCCM, vous devez le définir Slurm comme orchestrateur de processus en exportant la variable d'environnement. `I_MPI_HYDRA_BOOTSTRAP=slurm`

Note

Problème connu

Dans le cas où votre application MPI utilise SSH comme mécanisme pour générer des tâches MPI, il est possible qu'un [bogue connu dans Slurm](#) entraîne la mauvaise résolution du nom d'utilisateur du répertoire en « personne ».

Configurez votre application pour l'utiliser Slurm comme méthode de démarrage MPI ou consultez la section Dépannage pour plus de détails et les solutions possibles. [Problèmes connus liés à la résolution des noms d'utilisateur](#)

Exemple AWS Managed Microsoft AD de configuration de cluster LDAP (S)

AWS ParallelCluster prend en charge l'accès de plusieurs utilisateurs en s'intégrant AWS Directory Service au protocole LDAP (Lightweight Directory Access Protocol) ou au protocole LDAP sur TLS/SSL (LDAPS).

Les exemples suivants montrent comment créer des configurations de cluster à intégrer à un réseau AWS Managed Microsoft AD LDAP (S).

AWS Managed Microsoft AD via LDAPS avec vérification des certificats

Vous pouvez utiliser cet exemple pour intégrer votre cluster à un réseau AWS Managed Microsoft AD LDAPS, avec vérification des certificats.

Définitions spécifiques pour une configuration AWS Managed Microsoft AD via LDAPS avec certificats :

- [DirectoryService/LdapTlsReqCert](#) doit être défini sur `hard` (par défaut) pour LDAPS avec vérification des certificats.
- [DirectoryService/LdapTlsCaCert](#) doit spécifier le chemin d'accès à votre certificat de certificat d'autorité (CA).

Le certificat CA est un ensemble de certificats qui contient les certificats de l'ensemble de la chaîne CA qui a émis des certificats pour les contrôleurs de domaine AD.

Votre certificat CA et vos certificats doivent être installés sur les nœuds du cluster.

- Les noms d'hôte des contrôleurs doivent être spécifiés pour [DirectoryService/DomainAddr](#), et non pour les adresses IP.
- [DirectoryService](#) La [DomainReadOnlyUser](#) syntaxe/ doit être la suivante :

```
cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
```

Exemple de fichier de configuration de cluster pour l'utilisation d'AD sur LDAPS :

```
Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-1234567890abcdef0
  Ssh:
    KeyName: pcluster
  Iam:
    AdditionalIamPolicies:
```

```

- Policy: arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
CustomActions:
  OnNodeConfigured:
    Script: s3://aws-parallelcluster/scripts/pcluster-dub-msad-ldaps.post.sh
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: t2micro
          InstanceType: t2.micro
          MinCount: 1
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
      Iam:
        AdditionalIamPolicies:
          - Policy: arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
      CustomActions:
        OnNodeConfigured:
          Script: s3://aws-parallelcluster-pcluster/scripts/pcluster-dub-msad-
ldaps.post.sh
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldaps://win-abcdef01234567890.corp.example.com,ldaps://win-
abcdef01234567890.corp.example.com
  PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:MicrosoftAD.Admin.Password-1234
  DomainReadOnlyUser: cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
  LdapTlsCaCert: /etc/openldap/cacerts/corp.example.com.bundleca.cer
  LdapTlsReqCert: hard

```

Ajoutez des certificats et configurez les contrôleurs de domaine dans le script de post-installation :

```

#!/bin/bash*
set -e

AD_CERTIFICATE_S3_URI="s3://corp.example.com/bundle/corp.example.com.bundleca.cer"
AD_CERTIFICATE_LOCAL="/etc/openldap/cacerts/corp.example.com.bundleca.cer"

AD_HOSTNAME_1="win-abcdef01234567890.corp.example.com"
AD_IP_1="192.0.2.254"

```

```
AD_HOSTNAME_2="win-abcdef01234567890.corp.example.com"
AD_IP_2="203.0.113.225"

# Download CA certificate
mkdir -p $(dirname "${AD_CERTIFICATE_LOCAL}")
aws s3 cp "${AD_CERTIFICATE_S3_URI}" "${AD_CERTIFICATE_LOCAL}"
chmod 644 "${AD_CERTIFICATE_LOCAL}"

# Configure domain controllers reachability
echo "${AD_IP_1} ${AD_HOSTNAME_1}" >> /etc/hosts
echo "${AD_IP_2} ${AD_HOSTNAME_2}" >> /etc/hosts
```

Vous pouvez récupérer les noms d'hôte des contrôleurs de domaine à partir d'instances jointes au domaine, comme indiqué dans les exemples suivants.

À partir d'une instance Windows

```
$ nslookup 192.0.2.254
```

```
Server: corp.example.com
Address: 192.0.2.254

Name: win-abcdef01234567890.corp.example.com
Address: 192.0.2.254
```

Depuis une instance Linux

```
$ nslookup 192.0.2.254
```

```
192.0.2.254.in-addr.arpa name = corp.example.com
192.0.2.254.in-addr.arpa name = win-abcdef01234567890.corp.example.com
```

AWS Managed Microsoft AD via LDAPS sans vérification du certificat

Vous pouvez utiliser cet exemple pour intégrer votre cluster à un réseau LDAPS AWS Managed Microsoft AD supérieur, sans vérification de certificat.

Définitions spécifiques pour une configuration AWS Managed Microsoft AD via LDAPS sans vérification de certificat :

- [DirectoryService/LdapTlsReqCert](#) doit être défini sur `never`.
- Les noms d'hôte ou les adresses IP des contrôleurs peuvent être spécifiés pour [DirectoryService/DomainAddr](#).
- [DirectoryService](#) La [DomainReadOnlyUser](#) syntaxe/ doit être la suivante :

```
cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
```

Exemple de fichier de configuration de cluster à utiliser AWS Managed Microsoft AD via LDAPS sans vérification de certificat :

```
Region: region-id
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-1234567890abcdef0
  Ssh:
    KeyName: pcluster
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: t2micro
          InstanceType: t2.micro
          MinCount: 1
          MaxCount: 10
      Networking:
        SubnetIds:
          - subnet-abcdef01234567890
DirectoryService:
  DomainName: dc=corp,dc=example,dc=com
  DomainAddr: ldaps://203.0.113.225,ldaps://192.0.2.254
  PasswordSecretArn: arn:aws:secretsmanager:region-
id:123456789012:secret:MicrosoftAD.Admin.Password-1234
  DomainReadOnlyUser: cn=ReadOnly,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
  LdapTlsReqCert: never
```

Bonnes pratiques

Bonnes pratiques : sélection du type d'instance du nœud principal

Même si le nœud principal n'exécute aucune tâche, ses fonctions et son dimensionnement sont essentiels aux performances globales du cluster. Lorsque vous choisissez le type d'instance à utiliser pour votre nœud principal, tenez compte des caractéristiques suivantes :

Taille du cluster : le nœud principal orchestre la logique de dimensionnement du cluster et est chargé d'associer les nouveaux nœuds au planificateur. Pour augmenter ou diminuer un cluster comportant un grand nombre de nœuds, fournissez au nœud principal une capacité de calcul supplémentaire.

Systèmes de fichiers partagés : lorsque vous utilisez des systèmes de fichiers partagés, choisissez un type d'instance doté d'une bande passante réseau suffisante et d'une bande passante Amazon EBS suffisante pour gérer vos flux de travail. Assurez-vous que le nœud principal est capable à la fois d'exposer suffisamment de répertoires de serveurs NFS pour le cluster et de gérer les artefacts qui doivent être partagés entre les nœuds de calcul et le nœud principal.

Bonnes pratiques : performances du réseau

Les performances du réseau sont essentielles pour les applications de calcul haute performance (HPC). Sans performances réseau fiables, ces applications ne peuvent pas fonctionner comme prévu. Pour optimiser les performances du réseau, prenez en compte les meilleures pratiques suivantes.

- **Groupe de placement :** si vous utilisez Slurm, pensez à configurer chaque Slurm file d'attente pour utiliser un groupe de placement de clusters. Le groupe de placement d'un cluster est un regroupement logique d'instances au sein d'une seule zone de disponibilité. Pour plus d'informations, consultez la section [Groupes de placement](#) dans le guide de l'utilisateur Amazon EC2. Vous pouvez spécifier un `PlacementGroup` dans la `Networking` section de la file d'attente, chaque ressource de calcul est affectée au groupe de placement de la file d'attente. Lorsque vous spécifiez un `PlacementGroup` dans la `Networking` section de la ressource de calcul, cette ressource de calcul spécifique est affectée à ce groupe de placement. La spécification du groupe de placement des ressources de calcul remplace la spécification de file d'attente pour la ressource de calcul. Pour plus d'informations, consultez [SlurmQueues/Networking/PlacementGroup](#) et [SlurmQueues/ComputeResources/Networking/PlacementGroup](#).

```
Networking:  
  PlacementGroup:
```

```
Enabled: true
Id: your-placement-group-name
```

Vous pouvez également AWS ParallelCluster créer un groupe de placement pour vous.

```
Networking:
  PlacementGroup:
    Enabled: true
```

À partir de AWS ParallelCluster la version 3.3.0, la création et la gestion des groupes de placement sont modifiées. Lorsque vous spécifiez le groupe de placement à activer, sans name ou dans la file d'attenteId, chaque ressource de calcul se voit attribuer son propre groupe de placement géré, au lieu d'un groupe géré pour l'ensemble de la file d'attente. Cela permet de réduire les erreurs liées à une capacité insuffisante. Si vous avez besoin d'un groupe de placement pour l'ensemble de la file d'attente, vous pouvez utiliser un groupe de placement nommé.

[SlurmQueues/Networking/PlacementGroup/Name](#) a été ajouté comme alternative préférée à [SlurmQueues/Networking/PlacementGroup/Id](#).

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

- Mise en réseau améliorée : pensez à choisir un type d'instance qui prend en charge la mise en réseau améliorée. Cette recommandation s'applique à toutes les [instances de la génération actuelle](#). Pour plus d'informations, consultez la section [Mise en réseau améliorée sous Linux](#) dans le guide de l'utilisateur Amazon EC2.
- Adaptateur Elastic Fabric : pour prendre en charge des niveaux élevés de communication évolutive entre instances, pensez à choisir des interfaces réseau EFA pour votre réseau. Le matériel de contournement du système d'exploitation (OS) personnalisé de l'EFA améliore les communications entre instances grâce à l'élasticité et à la flexibilité à la demande du AWS Cloud. Vous pouvez configurer chaque Slurm file d'attente [ComputeResource](#) à utiliser [Efa](#). Pour plus d'informations sur l'utilisation d'EFA avec AWS ParallelCluster, consultez [Elastic Fabric Adapter](#).

```
ComputeResources:
  - Name: your-compute-resource-name
    Efa:
      Enabled: true
```

Pour plus d'informations sur EFA, consultez [Elastic Fabric Adapter](#) dans le guide de l'utilisateur Amazon EC2 pour les instances Linux.

- Bande passante de l'instance : la bande passante évolue en fonction de la taille de l'instance. Pour plus d'informations sur les différents types d'instances, consultez les [instances optimisées pour Amazon EBS](#) et les [types de volumes Amazon EBS](#) dans le guide de l'utilisateur Amazon EC2.

Bonnes pratiques : alertes budgétaires

Pour gérer les coûts des ressources dans AWS ParallelCluster, nous vous recommandons d'utiliser AWS Budgets des actions pour créer un budget. Vous pouvez également créer des alertes de seuils budgétaires définis pour AWS des ressources sélectionnées. Pour plus d'informations, consultez [la section Configuration d'une action budgétaire](#) dans le Guide de AWS Budgets l'utilisateur. De même, vous pouvez également utiliser Amazon CloudWatch pour créer une alarme de facturation. Pour plus d'informations, consultez [Création d'une alarme de facturation pour surveiller vos frais AWS estimés](#).

Bonnes pratiques : déplacement d'un cluster vers une nouvelle version AWS ParallelCluster mineure ou un correctif

Actuellement, chaque version AWS ParallelCluster mineure est autonome avec sa `pccluster` CLI. Pour déplacer un cluster vers une nouvelle version mineure ou corrective, vous devez recréer le cluster à l'aide de la CLI de la nouvelle version.

Pour optimiser le processus de déplacement d'un cluster vers une nouvelle version mineure ou un correctif, nous vous recommandons de procéder comme suit :

- Enregistrez les données personnelles dans des volumes externes créés en dehors du cluster, tels qu'Amazon EFS et FSx for Lustre. De cette manière, vous pourrez facilement déplacer les données d'un cluster à un autre dans le futur.
- Créez des systèmes de stockage partagés à l'aide des types suivants. Vous pouvez créer ces systèmes à l'aide du AWS CLI ou AWS Management Console.
 - [SharedStorage](#) / [EbsSettings](#) / [VolumeId](#)
 - [SharedStorage](#) / [EfsSettings](#) / [FileSystemId](#)
 - [SharedStorage](#) / [FsxLustreSettings](#) / [FileSystemId](#)

Définissez un système de fichiers ou un volume dans une configuration de cluster en tant que système de fichiers ou volume existant. Ils sont ainsi préservés lorsque vous supprimez le cluster et peuvent être attachés à un nouveau cluster.

Nous vous recommandons d'utiliser les systèmes de fichiers Amazon EFS ou FSx for Lustre. Ces deux systèmes peuvent être connectés à plusieurs clusters en même temps. En outre, vous pouvez associer l'un ou l'autre de ces systèmes à un nouveau cluster avant de supprimer le cluster existant.

- Utilisez [des actions d'amorçage personnalisées](#) pour personnaliser vos instances plutôt que d'utiliser une AMI personnalisée. Si vous utilisez plutôt une AMI personnalisée, vous devez supprimer et recréer cette AMI pour chaque nouvelle version publiée.
- Nous vous recommandons d'appliquer les recommandations précédentes dans l'ordre suivant :
 1. Mettez à jour la configuration de cluster existante pour utiliser les définitions de système de fichiers existantes.
 2. Vérifiez la `pcluster` version et mettez-la à jour si nécessaire.
 3. Créez et testez le nouveau cluster. Lorsque vous testez le nouveau cluster, vérifiez les points suivants :
 - Assurez-vous que vos données sont disponibles dans le nouveau cluster.
 - Assurez-vous que votre application fonctionne dans le nouveau cluster.
 4. Une fois que votre nouveau cluster est entièrement testé et opérationnel et que vous n'avez plus besoin du cluster existant, supprimez-le.

Passer de la version AWS ParallelCluster 2.x à la version 3.x

Actions d'amorçage personnalisées

Avec AWS ParallelCluster 3, vous pouvez spécifier différents scripts d'actions bootstrap personnalisés pour le nœud principal et les nœuds de calcul à l'aide des paramètres `OnNodeStart` (`pre_install` dans la AWS ParallelCluster version 2) et `OnNodeConfigured` (`post_install` dans la AWS ParallelCluster version 2) dans les [SlurmQueues](#) sections [HeadNode](#) et [Scheduling](#)/. Pour plus d'informations, consultez [Actions de bootstrap personnalisées](#).

Les scripts d'actions bootstrap personnalisés développés pour AWS ParallelCluster 2 doivent être adaptés pour être utilisés dans AWS ParallelCluster 3 :

- Nous ne recommandons pas d'utiliser `/etc/parallelcluster/cfnconfig` et `cfn_node_type` de différencier les nœuds de tête des nœuds de calcul. Nous vous recommandons plutôt de spécifier deux scripts différents dans le [HeadNode](#) et [Scheduling/SlurmQueues](#).

- Si vous préférez continuer le chargement `/etc/parallelcluster/cfnconfig` pour l'utiliser dans votre script d'actions bootstrap, notez que la valeur de `cfn_node_type` est modifiée de « MasterServer » à « HeadNode » (voir :[Langage inclusif](#)).
- Le AWS ParallelCluster 2, le premier argument d'entrée des scripts d'action bootstrap était l'URL S3 du script et était réservé. En AWS ParallelCluster 3, seuls les arguments configurés dans la configuration sont transmis aux scripts.

Warning

L'utilisation de variables internes fournies par le biais du `/etc/parallelcluster/cfnconfig` fichier n'est pas officiellement prise en charge. Ce fichier sera peut-être supprimé dans le cadre d'une future version.

AWS ParallelCluster 2.x et 3.x utilisent une syntaxe de fichier de configuration différente

AWS ParallelCluster La configuration 3.x utilise la syntaxe YAML. La référence complète se trouve à l'adresse [Fichiers de configuration](#).

En plus de nécessiter un format de fichier YAML, un certain nombre de sections de configuration, de paramètres et de valeurs de paramètres ont été mis à jour dans la version AWS ParallelCluster 3.x. Dans cette section, nous notons les principales modifications apportées à la AWS ParallelCluster configuration ainsi que des side-by-side exemples illustrant ces différences entre chaque version de AWS ParallelCluster.

Exemple de configuration de plusieurs files d'attente de planificateur avec l'hyperthreading activé et désactivé

AWS ParallelCluster 2 :

```
[cluster default]
queue_settings = ht-enabled, ht-disabled
...

[queue ht-enabled]
compute_resource_settings = ht-enabled-i1
disable_hyperthreading = false
```

```
[queue ht-disabled]
compute_resource_settings = ht-disabled-i1
disable_hyperthreading = true

[compute_resource ht-enabled-i1]
instance_type = c5n.18xlarge
[compute_resource ht-disabled-i1]
instance_type = c5.xlarge
```

AWS ParallelCluster 3 :

```
...
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: ht-enabled
      Networking:
        SubnetIds:
          - compute_subnet_id
      ComputeResources:
        - Name: ht-enabled-i1
          DisableSimultaneousMultithreading: true
          InstanceType: c5n.18xlarge
    - Name: ht-disabled
      Networking:
        SubnetIds:
          - compute_subnet_id
      ComputeResources:
        - Name: ht-disabled-i1
          DisableSimultaneousMultithreading: false
          InstanceType: c5.xlarge
```

Exemple de nouvelle configuration du système de fichiers FSx for Lustre

AWS ParallelCluster 2 :

```
[cluster default]
fsx_settings = fsx
...

[fsx fsx]
shared_dir = /shared-fsx
```

```

storage_capacity = 1200
imported_file_chunk_size = 1024
import_path = s3://bucket
export_path = s3://bucket/export_dir
weekly_maintenance_start_time = 3:02:30
deployment_type = PERSISTENT_1
data_compression_type = LZ4

```

AWS ParallelCluster 3 :

```

...
SharedStorage:
  - Name: fsx
    MountDir: /shared-fsx
    StorageType: FsxLustre
    FsxLustreSettings:
      StorageCapacity: 1200
      ImportedFileChunkSize: 1024
      ImportPath: s3://bucket
      ExportPath: s3://bucket/export_dir
      WeeklyMaintenanceStartTime: "3:02:30"
      DeploymentType: PERSISTENT_1
      DataCompressionType: LZ4

```

Exemple de configuration de cluster lors du montage d'un système de fichiers FSx for Lustre existant

AWS ParallelCluster 2 :

```

[cluster default]
fsx_settings = fsx
...

[fsx fsx]
shared_dir = /shared-fsx
fsx_fs_id = fsx_fs_id

```

AWS ParallelCluster 3 :

```

...
SharedStorage:
  - Name: fsx
    MountDir: /shared-fsx
    StorageType: FsxLustre

```



```
FsxLustreSettings:
  FileSystemId: fsx_fs_id
```

Exemple de cluster doté de la pile logicielle Intel HPC Platform Specification

AWS ParallelCluster 2 :

```
[cluster default]
enable_intel_hpc_platform = true
...
```

AWS ParallelCluster 3 :

```
...
AdditionalPackages:
  IntelSoftware:
    IntelHpcPlatform: true
```

Remarques :

- L'installation du logiciel Intel HPC Platform Specification est soumise aux termes et conditions du [contrat de licence utilisateur final Intel](#) applicable.

Exemple de configurations IAM personnalisées, notamment : profil d'instance, rôle d'instance, politiques supplémentaires pour les instances et rôle des fonctions lambda associées au cluster

AWS ParallelCluster 2 :

```
[cluster default]
additional_iam_policies = arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess,arn:aws:iam::aws:policy/AmazonDynamoDBReadOnlyAccess
ec2_iam_role = ec2_iam_role
iam_lambda_role = lambda_iam_role
...
```

AWS ParallelCluster 3 :

```
...
Iam:
  Roles:
    CustomLambdaResources: lambda_iam_role
```

```

HeadNode:
  ...
  Iam:
    InstanceRole: ec2_iam_role
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ...
      Iam:
        InstanceProfile: iam_instance_profile
    - Name: queue2
      ...
      Iam:
        AdditionalIamPolicies:
          - Policy: arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
          - Policy: arn:aws:iam::aws:policy/AmazonDynamoDBReadOnlyAccess

```

Remarques :

- Pour AWS ParallelCluster 2, les paramètres IAM sont appliqués à toutes les instances d'un cluster et ne `additional_iam_policies` peuvent pas être utilisés conjointement avec `ec2_iam_role`.
- Pour AWS ParallelCluster 3, vous pouvez définir différents paramètres IAM pour les nœuds de tête et de calcul et même spécifier des paramètres IAM différents pour chaque file d'attente de calcul.
- Pour AWS ParallelCluster 3, vous pouvez utiliser un profil d'instance IAM comme alternative à un rôle IAM. `InstanceProfile`, `InstanceRole` ou ne `AdditionalIamPolicies` peuvent pas être configurés ensemble.

Exemple d'actions de bootstrap personnalisées

AWS ParallelCluster 2 :

```

[cluster default]
s3_read_resource = arn:aws:s3:::bucket_name/*
pre_install = s3://bucket_name/scripts/pre_install.sh
pre_install_args = 'R curl wget'
post_install = s3://bucket_name/scripts/post_install.sh
post_install_args = "R curl wget"
...

```

AWS ParallelCluster 3 :

```

...
HeadNode:
  ...
  CustomActions:
    OnNodeStart:
      Script: s3://bucket_name/scripts/pre_install.sh
      Args:
        - R
        - curl
        - wget
    OnNodeConfigured:
      Script: s3://bucket_name/scripts/post_install.sh
      Args: ['R', 'curl', 'wget']
  Iam:
    S3Access:
      - BucketName: bucket_name
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
    ...
    CustomActions:
      OnNodeStart:
        Script: s3://bucket_name/scripts/pre_install.sh
        Args: ['R', 'curl', 'wget']
      OnNodeConfigured:
        Script: s3://bucket_name/scripts/post_install.sh
        Args: ['R', 'curl', 'wget']
    Iam:
      S3Access:
        - BucketName: bucket_name

```

Exemple de cluster avec accès en lecture et en écriture aux ressources du compartiment S3

AWS ParallelCluster 2 :

```

[cluster default]
s3_read_resource = arn:aws:s3:::bucket/read_only/*
s3_read_write_resource = arn:aws:s3:::bucket/read_and_write/*
...

```

AWS ParallelCluster 3 :

```
...
HeadNode:
  ...
  Iam:
    S3Access:
      - BucketName: bucket_name
        KeyName: read_only/
        EnableWriteAccess: False
      - BucketName: bucket_name
        KeyName: read_and_write/
        EnableWriteAccess: True
  Scheduling:
    Scheduler: slurm
    SlurmQueues:
      - Name: queue1
      ...
    Iam:
      S3Access:
        - BucketName: bucket_name
          KeyName: read_only/
          EnableWriteAccess: False
        - BucketName: bucket_name
          KeyName: read_and_write/
          EnableWriteAccess: True
```

Langage inclusif

AWS ParallelCluster 3 utilise les mots « nœud principal » aux endroits où « maître » a été utilisé dans AWS ParallelCluster 2. Cela inclut les éléments suivants :

- La variable exportée dans l'environnement de AWS Batch travail a été modifiée : de MASTER_IP à PCLUSTER_HEAD_NODE_IP.
- Toutes les AWS CloudFormation sorties sont passées de Master* à HeadNode*.
- Tous les tags NodeType et tags sont passés de Master à HeadNode.

Support pour les planificateurs

AWS ParallelCluster 3.x ne prend pas en charge le moteur Son of Grid (SGE) et les planificateurs de couple.

Les AWS Batch commandes `sawsbhosts`, `sawsbkill`, `sawsbout`, `sawsbqueuesawsbstat`, et `sawsbsub` sont distribuées sous forme de package `aws-parallelcluster-awsbatch-cli` PyPI distinct. Ce package est installé AWS ParallelCluster sur le nœud principal. Vous pouvez toujours utiliser ces AWS Batch commandes depuis le nœud principal du cluster. Toutefois, si vous souhaitez utiliser des AWS Batch commandes depuis un emplacement autre que le nœud principal, vous devez d'abord installer le package `aws-parallelcluster-awsbatch-cli` PyPI.

AWS ParallelCluster CLI

L'interface de ligne de AWS ParallelCluster commande (CLI) a été modifiée. La nouvelle syntaxe est décrite dans [Commandes de l'interface de ligne de commande AWS ParallelCluster](#). Le format de sortie de la CLI est une chaîne [JSON](#).

Configuration d'un nouveau cluster

La `pcluster` configure commande inclut des paramètres différents en AWS ParallelCluster 3 par rapport à AWS ParallelCluster 2. Pour plus d'informations, consultez [pcluster configure](#).

Notez également que la syntaxe du fichier de configuration a changé depuis AWS ParallelCluster 2. Pour une référence complète des paramètres de configuration du cluster, consultez [Fichier de configuration du cluster](#).

Création d'un nouveau cluster

AWS ParallelCluster La `pcluster create` commande de 2 a été remplacée par la [pcluster create-cluster](#) commande.

Notez que le comportement par défaut de la AWS ParallelCluster version 2.x, sans `-nw` cette option, est d'attendre les événements de création du cluster, tandis que la commande AWS ParallelCluster 3.x revient immédiatement. La progression de la création du cluster peut être surveillée à l'aide de [pcluster describe-cluster](#).

Un fichier de configuration AWS ParallelCluster 3 contient une seule définition de cluster, le `-t` paramètre n'est donc plus nécessaire.

Voici un exemple de fichier de configuration.

```
# AWS ParallelCluster v2
$ pcluster create \
  -r REGION \
  -c V2_CONFIG_FILE \
```

```
-nw \  
-t CLUSTER_TEMPLATE \  
CLUSTER_NAME  
  
# AWS ParallelCluster v3  
$ pcluster create-cluster \  
  --region REGION \  
  --cluster-configuration V3_CONFIG_FILE \  
  --cluster-name CLUSTER_NAME
```

Classer des clusters

La commande `pcluster list` AWS ParallelCluster 2.x doit être remplacée par une [pcluster list-clusters](#) commande.

Remarque : vous avez besoin de la CLI AWS ParallelCluster v2 pour répertorier les clusters créés avec les versions 2.x de AWS ParallelCluster. Découvrez [Installation AWS ParallelCluster dans un environnement virtuel \(recommandé\)](#) comment installer plusieurs versions de l' AWS ParallelCluster utilisation d'environnements virtuels.

```
# AWS ParallelCluster v2  
$ pcluster list -r REGION  
  
# AWS ParallelCluster v3  
$ pcluster list-clusters --region REGION
```

Démarrage et arrêt d'un cluster

Les commandes `pcluster start` et `pcluster stop` AWS ParallelCluster 2.x doivent être remplacées par des [pcluster update-compute-fleet](#) commandes.

Démarrage d'un parc informatique :

```
# AWS ParallelCluster v2  
$ pcluster start \  
  -r REGION \  
  CLUSTER_NAME  
  
# AWS ParallelCluster v3 - Slurm fleets  
$ pcluster update-compute-fleet \  
  --region REGION \  
  --cluster-name CLUSTER_NAME \  
  CLUSTER_NAME
```

```
--status START_REQUESTED

# AWS ParallelCluster v3 - AWS Batch fleets
$ pcluster update-compute-fleet \
  --region REGION \
  --cluster-name CLUSTER_NAME \
  --status ENABLED
```

Arrêter un parc informatique :

```
# AWS ParallelCluster v2
$ pcluster stop \
  -r REGION \
  CLUSTER_NAME

# AWS ParallelCluster v3 - Slurm fleets
$ pcluster update-compute-fleet \
  --region REGION \
  --cluster-name CLUSTER_NAME \
  --status STOP_REQUESTED

# AWS ParallelCluster v3 - AWS Batch fleets
$ pcluster update-compute-fleet \
  --region REGION \
  --cluster-name CLUSTER_NAME \
  --status DISABLED
```

Connexion à un cluster

La commande `pcluster ssh` AWS ParallelCluster 2.x a des noms de paramètres différents dans la version AWS ParallelCluster 3.x. veuillez consulter [pcluster ssh](#).

Connexion à un cluster :

```
# AWS ParallelCluster v2
$ pcluster ssh \
  -r REGION \
  CLUSTER_NAME \
  -i ~/.ssh/id_rsa

# AWS ParallelCluster v3
$ pcluster ssh \
  --region REGION \
```

```
--cluster-name CLUSTER_NAME \  
-i ~/.ssh/id_rsa
```

Mise à jour de configuration IMDS

À partir de la version 3.0.0, la prise en charge de la restriction de l'accès à l'IMDS du nœud principal (et aux informations d'identification du profil d'instance) à un sous-ensemble de superutilisateurs, par défaut, a été introduite. Pour plus d'informations, voir [Propriétés de Imds](#).

Régions prises en charge pour AWS ParallelCluster

AWS ParallelCluster la version 3 est disponible dans les versions suivantes Régions AWS :

| Nom de la région | Région |
|----------------------------|----------------|
| USA Est (Ohio) | us-east-2 |
| USA Est (Virginie du Nord) | us-east-1 |
| US West (N. California) | us-west-1 |
| US West (Oregon) | us-west-2 |
| Afrique (Le Cap) | af-south-1 |
| Asie-Pacifique (Hong Kong) | ap-east-1 |
| Asie-Pacifique (Mumbai) | ap-south-1 |
| Asia Pacific (Seoul) | ap-northeast-2 |
| Asie-Pacifique (Singapour) | ap-southeast-1 |
| Asie-Pacifique (Sydney) | ap-southeast-2 |
| Asia Pacific (Tokyo) | ap-northeast-1 |
| Canada (Central) | ca-central-1 |
| China (Beijing) | cn-north-1 |

| Nom de la région | Région |
|-----------------------------|----------------|
| China (Ningxia) | cn-northwest-1 |
| Europe (Francfort) | eu-central-1 |
| Europe (Irlande) | eu-west-1 |
| Europe (Londres) | eu-west-2 |
| Europe (Milan) | eu-south-1 |
| Europe (Paris) | eu-west-3 |
| Europe (Stockholm) | eu-north-1 |
| Moyen-Orient (Bahreïn) | me-south-1 |
| Amérique du Sud (São Paulo) | sa-east-1 |
| AWS GovCloud (USA Est) | us-gov-east-1 |
| AWS GovCloud (US-Ouest) | us-gov-west-1 |
| Israël (Tel Aviv) | il-central-1 |

En utilisant AWS ParallelCluster

Rubriques

- [AWS ParallelCluster INTERFACE UTILISATEUR](#)
- [AWS Lambda Configuration VPC dans AWS ParallelCluster](#)
- [AWS Identity and Access Management autorisations dans AWS ParallelCluster](#)
- [Configurations réseau](#)
- [Nœuds de connexion](#)
- [Actions de bootstrap personnalisées](#)
- [Utilisation des services dans Amazon S3](#)
- [Utilisation de instances Spot](#)
- [Planificateurs pris en charge par AWS ParallelCluster](#)
- [Stockage partagé](#)
- [AWS ParallelCluster ressources et balisage](#)
- [Surveillance AWS ParallelCluster et journaux](#)
- [AWS CloudFormation ressource personnalisée](#)
- [Elastic Fabric Adapter](#)
- [Activez Intel MPI](#)
- [AWS ParallelCluster API](#)
- [AWS ParallelCluster pour Terraform](#)
- [Connectez-vous au nœud principal via NICE DCV](#)
- [Utiliser pcluster update-cluster](#)
- [AWS ParallelCluster Personnalisation de l'AMI](#)
- [Lancez des instances avec des réservations de capacité à la demande \(ODCR\)](#)
- [Lancer des instances avec des blocs de capacité \(CB\)](#)
- [Corriger les AMI et remplacer les instances Amazon EC2](#)
- [Systèmes d'exploitation](#)

AWS ParallelClusterINTERFACE UTILISATEUR

L'AWS ParallelClusterinterface utilisateur est une interface utilisateur Web qui sert de tableau de bord pour créer, surveiller et gérer des clusters. Vous installez et accédez à l'AWS ParallelClusterinterface utilisateur dans votreCompte AWS. L'AWS ParallelClusterinterface utilisateur a été ajoutée avecAWS ParallelCluster la version 3.5.0.

Pour installer l'AWS ParallelClusterinterface utilisateur et commencer, reportez-vous aux sections [Installation de l'AWS ParallelClusterinterface utilisateur](#) et [Configuration et création d'un cluster avec l' AWS ParallelCluster interface utilisateur](#).

The screenshot displays the AWS ParallelCluster web interface. At the top, the header shows the AWS ParallelCluster logo, the user's email (user@domain.com), and the region (eu-west-1). The main navigation menu on the left includes 'AWS ParallelCluster', 'Clusters', 'Images', and 'Users'. The central area shows a 'Clusters (2)' overview with buttons for 'Shell', 'DCV', 'Stop fleet', 'Actions', and 'Create cluster'. Below this is a search bar and a table of clusters:

| Name | Status | Version |
|---------------|--------------------|---------|
| hpc-cluster-1 | CREATE COMPLETE | 3.5.0 |
| hpc-cluster-2 | DELETE IN PROGRESS | 3.5.0 |

Below the table, the 'Cluster: hpc-cluster-1' details are shown, including tabs for 'Details', 'Instances', 'Storage', 'Job scheduling', and 'Stack events'. The 'Properties' section is expanded, showing:

- Cluster configuration:** [VIEW](#)
- SSH command:** `ssh ec2-user@54.78.245.22`
- EC2 Instance Connect:** `mssh -r eu-west-1 ec2-user@i-0b14dc1a2f5dc048e`
- Cluster status:** CREATE COMPLETE
- Compute fleet status:** RUNNING
- Version:** 3.5.0
- Region:** eu-west-1
- Created time:** March 10, 2023 at 09:39 (UTC+1:00)
- Latest update time:** March 10, 2023 at 09:39 (UTC+1:00)

L'AWS ParallelClusterinterface utilisateur prend en charge :

- L'affiche
 - La liste des clusters que vous avez créésCompte AWS avecAWS ParallelCluster.

- L'état disponible et les détails de vos clusters répertoriés.
- CloudFormation empilez les événements et AWS ParallelCluster les journaux que vous pouvez utiliser à des fins de surveillance.
- État des tâches en cours d'exécution sur vos clusters.
- La liste des images personnalisées que vous pouvez utiliser pour générer des clusters.
- La liste des images officielles que l'interface utilisateur utilise pour créer des clusters.
- La liste des utilisateurs ayant accès à l'AWS ParallelCluster interface utilisateur. Vous pouvez ajouter et supprimer des utilisateurs.
- Fournit step-by-step des instructions pour créer et modifier (mettre à jour) un cluster et sélectionner les fonctionnalités de cluster prises en charge à ajouter, modifier ou supprimer. Les champs de saisie inaccessibles ne peuvent pas être modifiés pour la configuration du cluster en cours de modification. Vous avez la possibilité d'effectuer une validation à sec de la configuration de votre cluster avant le déploiement du cluster.
- Comprend des liens directs vers le shell pour accéder au nœud principal dans la vue Clusters. Choisissez Ajouter une session SSM pendant les step-by-step instructions pour ajouter l'accès direct au shell et la politique SSM Managed Instance Core sur le nœud principal.

Tenez compte des éléments suivants lorsque vous utilisez l'AWS ParallelCluster interface utilisateur pour créer et gérer vos clusters

- Vous pouvez uniquement créer et modifier des clusters ou créer des images avec la même AWS ParallelCluster version que celle utilisée pour créer l'AWS ParallelCluster interface utilisateur. Les clusters ou les images des versions antérieures ne peuvent être visualisés que. Si vous gérez plusieurs versions de clusters et d'images, nous vous recommandons de créer une instance d'AWS ParallelCluster interface utilisateur pour prendre en charge chaque version.
- L'AWS ParallelCluster interface utilisateur est conçue pour refléter les fonctionnalités de `lapcluster` CLI. Il y a quelques différences. Si vous vous conformez aux step-by-step instructions, vous utilisez toutes les fonctionnalités prises en charge. Avant le déploiement, vous avez la possibilité de modifier manuellement la configuration du cluster ou de l'image. Dans ce cas, nous vous recommandons de valider la configuration en choisissant Dry Run pour vérifier que vos modifications sont entièrement prises en charge.

Note

AWS ParallelCluster L'interface utilisateur ne prend pas en charge AWS Batch.

AWS Lambda Configuration VPC dans AWS ParallelCluster

AWS ParallelCluster utilise AWS Lambda pour effectuer des opérations pendant le cycle de vie du cluster. Une [AWS Lambda fonction s'exécute toujours dans un VPC](#) appartenant au service Lambda. Cette fonction Lambda peut également être connectée à des sous-réseaux privés dans un cloud privé virtuel (VPC) pour accéder à des ressources privées.

Note

Les fonctions Lambda ne peuvent pas se connecter directement à un VPC avec la location d'instance dédiée. Pour vous connecter aux ressources d'un VPC dédié, associez le VPC dédié à un second VPC avec une location par défaut qui peut se connecter à un VPC dédié. Pour plus d'informations, consultez les [sections Instances dédiées](#) dans le guide de l'utilisateur Amazon EC2 pour les instances Linux et [Comment connecter une fonction Lambda à un VPC dédié ?](#) depuis le AWS Knowledge Center.

Les fonctions Lambda créées par AWS ParallelCluster peuvent être connectées à un VPC privé. Ces fonctions Lambda doivent y accéder. services AWS Vous pouvez fournir un accès via Internet ou des points de terminaison VPC en utilisant les méthodes suivantes.

- Accès à Internet

Pour accéder à Internet services AWS, une fonction Lambda nécessite une traduction d'adresses réseau (NAT). Acheminez le trafic sortant de votre sous-réseau privé vers une [passerelle NAT](#) dans un sous-réseau public.

- Points de terminaison d'un VPC

Plusieurs AWS services proposent des points de [terminaison VPC](#). Vous pouvez utiliser des points de terminaison VPC pour vous connecter à services AWS un VPC qui n'a pas accès à Internet. [Pour consulter la liste des points de terminaison AWS ParallelCluster VPC, consultez la section Mise en réseau.](#)

Note

Chaque combinaison de sous-réseaux et de groupes de sécurité doit permettre d'accéder à services AWS l'une de ces méthodes. Les sous-réseaux et les groupes de sécurité doivent se trouver dans le même VPC.

Pour plus d'informations, consultez les [points de terminaison VPC](#) dans le guide de l'utilisateur d'Amazon Virtual Private Cloud et l'[accès à Internet et aux services pour les fonctions connectées au VPC](#) dans le manuel du développeur.AWS Lambda

Pour configurer l'utilisation des fonctions Lambda et des VPC, consultez [DeploymentSettings/LambdaFunctionsVpcConfig](#) pour les clusters ou [DeploymentSettings/LambdaFunctionsVpcConfig](#) pour les images.

AWS Identity and Access Management autorisations dans AWS ParallelCluster

AWS ParallelCluster utilise les autorisations IAM pour contrôler l'accès aux ressources lors de la création et de la gestion des clusters.

Pour créer et gérer des clusters dans un AWS compte, des AWS ParallelCluster autorisations à deux niveaux sont nécessaires :

- Autorisations dont l'`pcluster`utilisateur a besoin pour appeler les commandes de la `pcluster` CLI pour créer et gérer des clusters.
- Autorisations dont les ressources du cluster ont besoin pour effectuer des actions de cluster.

AWS ParallelCluster utilise un [profil et un rôle d'instance Amazon EC2 pour fournir des autorisations](#) de ressources de cluster. Pour gérer les autorisations des ressources du cluster, il faut AWS ParallelCluster également des autorisations pour les ressources IAM. Pour plus d'informations, consultez [AWS ParallelCluster exemples de politiques utilisateur pour la gestion des ressources IAM](#).

`pcluster`les utilisateurs ont besoin d'autorisations IAM pour utiliser la `pcluster` CLI afin de créer et de gérer un cluster et ses ressources. Ces autorisations sont incluses dans les politiques IAM qui peuvent être ajoutées à un utilisateur ou à un rôle. Pour plus d'informations sur les rôles

IAM, consultez la section [Création d'un rôle utilisateur](#) dans le Guide de l'AWS Identity and Access Management utilisateur.

Vous pouvez également utiliser [AWS ParallelCluster paramètres de configuration pour gérer les autorisations IAM](#).

Les sections suivantes contiennent les autorisations requises avec des exemples.

Pour utiliser les exemples de politiques, remplacez `<REGION><AWS ACCOUNT ID>`, et les chaînes similaires par les valeurs appropriées.

Les exemples de politiques suivants incluent les Amazon Resource Names (ARN) pour les ressources. Si vous travaillez dans les partitions AWS GovCloud (US) ou AWS China, les ARN doivent être modifiés. Plus précisément, ils doivent être remplacés par « arn:aws » par « arn : aws-us-gov » pour la AWS GovCloud (US) partition ou « arn:aws-cn » pour la partition chinoise. AWS Pour plus d'informations, consultez [Amazon Resource Names \(ARN\) dans AWS GovCloud \(US\) les régions dans](#) le guide de AWS GovCloud (US) l'utilisateur et [ARN pour les AWS services en Chine dans](#) Getting Started with AWS services in China.

Vous pouvez suivre les modifications apportées aux exemples de politiques dans [AWS ParallelCluster la documentation sur GitHub](#).

Rubriques

- [AWS ParallelCluster Rôles d'instance Amazon EC2](#)
- [AWS ParallelCluster exemples de politiques pcluster utilisateur](#)
- [AWS ParallelCluster exemples de politiques utilisateur pour la gestion des ressources IAM](#)
- [AWS ParallelCluster paramètres de configuration pour gérer les autorisations IAM](#)

AWS ParallelCluster Rôles d'instance Amazon EC2

Lorsque vous créez un cluster avec les paramètres de configuration par défaut, AWS ParallelCluster vous utilisez les [profils d'instance](#) Amazon EC2 pour créer automatiquement un rôle d'[instance Amazon EC2](#) par défaut qui fournit les autorisations requises pour créer et gérer le cluster et ses ressources.

Alternatives à l'utilisation du rôle d' AWS ParallelCluster instance par défaut

À la place du rôle d' AWS ParallelCluster instance par défaut, vous pouvez utiliser le paramètre de configuration du `InstanceRole` cluster pour spécifier votre propre rôle IAM existant pour EC2.

Pour plus d'informations, consultez [AWS ParallelCluster paramètres de configuration pour gérer les autorisations IAM](#). Généralement, vous spécifiez les rôles IAM existants pour contrôler entièrement les autorisations accordées à EC2.

Si votre intention est d'ajouter des politiques supplémentaires au rôle d'instance par défaut, nous vous recommandons de transmettre les politiques IAM supplémentaires en utilisant le paramètre de `AdditionalIamPoliciesInstanceProfileconfiguration` plutôt que `InstanceRole` les paramètres. Vous pouvez mettre à jour `AdditionalIamPolicies` lorsque vous mettez à jour votre cluster, mais vous ne pouvez pas le mettre à jour `InstanceRole` lorsque vous mettez à jour votre cluster.

AWS ParallelCluster exemples de politiques `pcluster` utilisateur

Les exemples suivants montrent les politiques utilisateur requises pour créer AWS ParallelCluster et gérer ses ressources à l'aide de la `pcluster` CLI. Vous pouvez associer des politiques à un utilisateur ou à un rôle.

Rubriques

- [Politique AWS ParallelCluster `pcluster` utilisateur de base](#)
- [Politique AWS ParallelCluster `pcluster` utilisateur supplémentaire lors de l'utilisation du AWS Batch planificateur](#)
- [Politique AWS ParallelCluster `pcluster` utilisateur supplémentaire lors de l'utilisation d'Amazon FSx for Lustre](#)
- [AWS ParallelCluster politique `pcluster` utilisateur de création d'images](#)

Politique AWS ParallelCluster `pcluster` utilisateur de base

La politique suivante indique les autorisations requises pour exécuter AWS ParallelCluster `pcluster` des commandes.

La dernière action répertoriée dans la politique est incluse pour valider tous les secrets spécifiés dans la configuration du cluster. Par exemple, un AWS Secrets Manager secret est utilisé pour configurer l'[DirectoryService](#) intégration. Dans ce cas, un cluster n'est créé que s'il existe un secret valide dans le [PasswordSecretArn](#). Si cette action est omise, la validation secrète est ignorée. Pour améliorer votre niveau de sécurité, nous vous recommandons de limiter la portée de cette déclaration de politique en ajoutant uniquement les secrets spécifiés dans la configuration de votre cluster.

Note

Si les systèmes de fichiers Amazon EFS existants sont les seuls systèmes de fichiers utilisés dans votre cluster, vous pouvez appliquer les exemples de déclarations de politique Amazon EFS aux systèmes de fichiers spécifiques référencés dans le fichier [SharedStorage Section](#) de configuration du cluster.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:Describe*"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "EC2Read"
    },
    {
      "Action": [
        "ec2:AllocateAddress",
        "ec2:AssociateAddress",
        "ec2:AttachNetworkInterface",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateFleet",
        "ec2:CreateLaunchTemplate",
        "ec2:CreateLaunchTemplateVersion",
        "ec2:CreateNetworkInterface",
        "ec2:CreatePlacementGroup",
        "ec2:CreateSecurityGroup",
        "ec2:CreateSnapshot",
        "ec2:CreateTags",
        "ec2>DeleteTags",
        "ec2:CreateVolume",
        "ec2>DeleteLaunchTemplate",
        "ec2>DeleteNetworkInterface",
        "ec2>DeletePlacementGroup",
        "ec2>DeleteSecurityGroup",
        "ec2>DeleteVolume",
        "ec2:DisassociateAddress",
```

```

        "ec2:ModifyLaunchTemplate",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2:ModifyVolume",
        "ec2:ModifyVolumeAttribute",
        "ec2:ReleaseAddress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:RunInstances",
        "ec2:TerminateInstances"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "EC2Write"
},
{
    "Action": [
        "dynamodb:DescribeTable",
        "dynamodb:ListTagsOfResource",
        "dynamodb:CreateTable",
        "dynamodb>DeleteTable",
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:Query",
        "dynamodb:TagResource"
    ],
    "Resource": "arn:aws:dynamodb:*:<AWS ACCOUNT ID>:table/parallelcluster-*",
    "Effect": "Allow",
    "Sid": "DynamoDB"
},
{
    "Action": [
        "route53:ChangeResourceRecordSets",
        "route53:ChangeTagsForResource",
        "route53:CreateHostedZone",
        "route53>DeleteHostedZone",
        "route53:GetChange",
        "route53:GetHostedZone",
        "route53:ListResourceRecordSets",
        "route53:ListQueryLoggingConfigs"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "Route53HostedZones"
}

```

```

    },
    {
      "Action": [
        "cloudformation:*"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "CloudFormation"
    },
    {
      "Action": [
        "cloudwatch:PutDashboard",
        "cloudwatch:ListDashboards",
        "cloudwatch>DeleteDashboards",
        "cloudwatch:GetDashboard",
        "cloudwatch:PutMetricAlarm",
        "cloudwatch>DeleteAlarms",
        "cloudwatch:DescribeAlarms"
        "cloudwatch:PutCompositeAlarm"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "CloudWatch"
    },
    {
      "Action": [
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:GetPolicy",
        "iam:SimulatePrincipalPolicy",
        "iam:GetInstanceProfile"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:policy/*",
        "arn:aws:iam::aws:policy/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/*"
      ],
      "Effect": "Allow",
      "Sid": "IamRead"
    },
    {
      "Action": [
        "iam:CreateInstanceProfile",

```

```

        "iam:DeleteInstanceProfile",
        "iam:AddRoleToInstanceProfile",
        "iam:RemoveRoleFromInstanceProfile"
    ],
    "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamInstanceProfile"
},
{
    "Condition": {
        "StringEqualsIfExists": {
            "iam:PassedToService": [
                "lambda.amazonaws.com",
                "ec2.amazonaws.com",
                "spotfleet.amazonaws.com"
            ]
        }
    },
    "Action": [
        "iam:PassRole"
    ],
    "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
    ],
    "Effect": "Allow",
    "Sid": "IamPassRole"
},
{
    "Action": [
        "lambda:CreateFunction",
        "lambda:DeleteFunction",
        "lambda:GetFunctionConfiguration",
        "lambda:GetFunction",
        "lambda:InvokeFunction",
        "lambda:AddPermission",
        "lambda:RemovePermission",
        "lambda:UpdateFunctionConfiguration",
        "lambda:TagResource",
        "lambda:ListTags",
        "lambda:UntagResource"
    ],
    "Resource": [

```

```

        "arn:aws:lambda:*:<AWS ACCOUNT ID>:function:parallelcluster-*",
        "arn:aws:lambda:*:<AWS ACCOUNT ID>:function:pcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "Lambda"
},
{
    "Action": [
        "s3:*"
    ],
    "Resource": [
        "arn:aws:s3:::parallelcluster-*",
        "arn:aws:s3:::aws-parallelcluster-*"
    ],
    "Effect": "Allow",
    "Sid": "S3ResourcesBucket"
},
{
    "Action": [
        "s3:Get*",
        "s3:List*"
    ],
    "Resource": "arn:aws:s3:::*-aws-parallelcluster*",
    "Effect": "Allow",
    "Sid": "S3ParallelClusterReadOnly"
},
{
    "Action": [
        "elasticfilesystem:*"
    ],
    "Resource": [
        "arn:aws:elasticfilesystem:*:<AWS ACCOUNT ID>:*"
    ],
    "Effect": "Allow",
    "Sid": "EFS"
},
{
    "Action": [
        "logs:DeleteLogGroup",
        "logs:PutRetentionPolicy",
        "logs:DescribeLogGroups",
        "logs:CreateLogGroup",
        "logs:TagResource",
        "logs:UntagResource",

```

```

        "logs:FilterLogEvents",
        "logs:GetLogEvents",
        "logs>CreateExportTask",
        "logs:DescribeLogStreams",
        "logs:DescribeExportTasks",
        "logs:DescribeMetricFilters",
        "logs:PutMetricFilter",
        "logs>DeleteMetricFilter"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "CloudWatchLogs"
},
{
    "Action": [
        "resource-groups:ListGroupResources"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "ResourceGroupRead"
},
{
    "Sid": "AllowDescribingFileCache",
    "Effect": "Allow",
    "Action": [
        "fsx:DescribeFileCaches"
    ],
    "Resource": "*"
},
{
    "Action": "secretsmanager:DescribeSecret",
    "Resource": "arn:aws:secretsmanager:<REGION>:<AWS ACCOUNT ID>:secret:<SECRET
NAME>",
    "Effect": "Allow"
}
]
}

```

Politique AWS ParallelCluster **pcluster** utilisateur supplémentaire lors de l'utilisation du AWS Batch planificateur

Si vous devez créer et gérer un cluster avec un AWS Batch planificateur, la politique supplémentaire suivante est requise.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Condition": {
        "StringEqualsIfExists": {
          "iam:PassedToService": [
            "ecs-tasks.amazonaws.com",
            "batch.amazonaws.com",
            "codebuild.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IamPassRole"
    },
    {
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": [
            "batch.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:CreateServiceLinkedRole",
        "iam>DeleteServiceLinkedRole"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/aws-service-role/
batch.amazonaws.com/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "codebuild:*"
      ]
    }
  ]
}
```

```
    ],
    "Resource": "arn:aws:codebuild:*:<AWS ACCOUNT ID>:project/pcluster-*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "ecr:*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "ECR"
  },
  {
    "Action": [
      "batch:*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "Batch"
  },
  {
    "Action": [
      "events:*"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "AmazonCloudWatchEvents"
  },
  {
    "Action": [
      "ecs:DescribeContainerInstances",
      "ecs:ListContainerInstances"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Sid": "ECS"
  }
]
}
```


Politique AWS ParallelCluster **pcluster** utilisateur supplémentaire lors de l'utilisation d'Amazon FSx for Lustre

Si vous devez créer et gérer un cluster avec Amazon FSx for Lustre, la politique supplémentaire suivante est requise.

Note

Si les systèmes de fichiers Amazon FSx existants sont les seuls systèmes de fichiers utilisés dans votre cluster, vous pouvez appliquer les exemples de déclarations de politique Amazon FSx aux systèmes de fichiers spécifiques référencés dans le fichier de configuration [SharedStorage Section](#) du cluster.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": [
            "fsx.amazonaws.com",
            "s3.data-source.lustre.fsx.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:CreateServiceLinkedRole",
        "iam>DeleteServiceLinkedRole"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "fsx:*"
      ],
      "Resource": [
        "arn:aws:fsx:*:<AWS ACCOUNT ID>:*"
      ],
      "Effect": "Allow",
    }
  ]
}
```

```

        "Sid": "FSx"
    },
    {
        "Action": [
            "iam:CreateServiceLinkedRole",
            "iam:AttachRolePolicy",
            "iam:PutRolePolicy"
        ],
        "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/aws-service-role/s3.data-
source.lustre.fsx.amazonaws.com/*",
        "Effect": "Allow"
    },
    {
        "Action": [
            "s3:Get*",
            "s3:List*",
            "s3:PutObject"
        ],
        "Resource": "arn:aws:s3:::<S3 NAME>",
        "Effect": "Allow"
    }
]
}

```

AWS ParallelCluster politique **pcluster** utilisateur de création d'images

Les utilisateurs qui ont l'intention de créer des images Amazon EC2 personnalisées AWS ParallelCluster doivent disposer des autorisations suivantes.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:DescribeImages",
        "ec2:DescribeInstanceTypeOfferings",
        "ec2:DescribeInstanceTypes",
        "ec2:DeregisterImage",
        "ec2>DeleteSnapshot"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "EC2"
    }
  ]
}

```

```

    },
    {
      "Action": [
        "iam:CreateInstanceProfile",
        "iam:AddRoleToInstanceProfile",
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:GetInstanceProfile",
        "iam:RemoveRoleFromInstanceProfile"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/ParallelClusterImage*",
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IAM"
    },
    {
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "lambda.amazonaws.com",
            "ec2.amazonaws.com"
          ]
        }
      },
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*",
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IAMPassRole"
    },
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:TagResource",
        "logs:UntagResource",
        "logs>DeleteLogGroup"
      ],

```

```

    "Resource": [
      "arn:aws:logs:*:<AWS ACCOUNT ID>:log-group:/aws/imagebuilder/
ParallelClusterImage-*",
      "arn:aws:logs:*:<AWS ACCOUNT ID>:log-group:/aws/lambda/
ParallelClusterImage-*"
    ],
    "Effect": "Allow",
    "Sid": "CloudWatch"
  },
  {
    "Action": [
      "cloudformation:DescribeStacks",
      "cloudformation:CreateStack",
      "cloudformation>DeleteStack"
    ],
    "Resource": [
      "arn:aws:cloudformation:*:<AWS ACCOUNT ID>:stack/*"
    ],
    "Effect": "Allow",
    "Sid": "CloudFormation"
  },
  {
    "Action": [
      "lambda:CreateFunction",
      "lambda:GetFunction",
      "lambda:AddPermission",
      "lambda:RemovePermission",
      "lambda>DeleteFunction",
      "lambda:TagResource",
      "lambda:ListTags",
      "lambda:UntagResource"
    ],
    "Resource": [
      "arn:aws:lambda:*:<AWS ACCOUNT ID>:function:ParallelClusterImage-*"
    ],
    "Effect": "Allow",
    "Sid": "Lambda"
  },
  {
    "Action": [
      "imagebuilder:Get*"
    ],
    "Resource": "*",
    "Effect": "Allow",

```

```

        "Sid": "ImageBuilderGet"
    },
    {
        "Action": [
            "imagebuilder:CreateImage",
            "imagebuilder:TagResource",
            "imagebuilder:CreateImageRecipe",
            "imagebuilder:CreateComponent",
            "imagebuilder:CreateDistributionConfiguration",
            "imagebuilder:CreateInfrastructureConfiguration",
            "imagebuilder>DeleteImage",
            "imagebuilder>DeleteComponent",
            "imagebuilder>DeleteImageRecipe",
            "imagebuilder>DeleteInfrastructureConfiguration",
            "imagebuilder>DeleteDistributionConfiguration"
        ],
        "Resource": [
            "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:image/parallelclusterimage-*",
            "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:image-recipe/
parallelclusterimage-*",
            "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:component/
parallelclusterimage-*",
            "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:distribution-configuration/
parallelclusterimage-*",
            "arn:aws:imagebuilder:*:<AWS ACCOUNT ID>:infrastructure-configuration/
parallelclusterimage-*"
        ],
        "Effect": "Allow",
        "Sid": "ImageBuilder"
    },
    {
        "Action": [
            "s3:CreateBucket",
            "s3:ListBucket",
            "s3:ListBucketVersions"
        ],
        "Resource": [
            "arn:aws:s3:::parallelcluster-*"
        ],
        "Effect": "Allow",
        "Sid": "S3Bucket"
    },
    {
        "Action": [

```

```

        "sns:GetTopicAttributes",
        "sns:TagResource",
        "sns:CreateTopic",
        "sns:Subscribe",
        "sns:Publish",
        "SNS:DeleteTopic",
        "SNS:Unsubscribe"
    ],
    "Resource": [
        "arn:aws:sns:*:<AWS ACCOUNT ID>:ParallelClusterImage-*"
    ],
    "Effect": "Allow",
    "Sid": "SNS"
},
{
    "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion"
    ],
    "Resource": [
        "arn:aws:s3:::parallelcluster-*/*"
    ],
    "Effect": "Allow",
    "Sid": "S3Objects"
},
{
    "Action": "iam:CreateServiceLinkedRole",
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/aws-service-role/
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": "imagebuilder.amazonaws.com"
        }
    }
}
]
}

```

AWS ParallelCluster exemples de politiques utilisateur pour la gestion des ressources IAM

Lors AWS ParallelCluster de l'utilisation pour créer des clusters ou des AMI personnalisées, des politiques IAM doivent être fournies qui contiennent des autorisations permettant d'accorder l'ensemble d'autorisations requis aux AWS ParallelCluster composants. Ces ressources IAM peuvent être créées automatiquement AWS ParallelCluster ou fournies en entrée lors de la création d'un cluster ou d'une image personnalisée.

Vous pouvez utiliser les modes suivants pour fournir à l' AWS ParallelCluster utilisateur les autorisations nécessaires pour accéder aux ressources IAM en utilisant des politiques IAM supplémentaires dans la configuration.

Rubriques

- [Mode d'accès IAM privilégié](#)
- [Mode d'accès IAM restreint](#)
- [Mode PermissionsBoundary](#)

Mode d'accès IAM privilégié

Avec ce mode, crée AWS ParallelCluster automatiquement toutes les ressources IAM nécessaires. Ces politiques IAM sont limitées pour permettre l'accès aux ressources du cluster uniquement.

Pour activer le mode d'accès privilégié IAM, ajoutez la politique suivante au rôle d'utilisateur.

Note

Si vous configurez [HeadNode](#) les [AdditionalPolicies](#) paramètres [Scheduling/SlurmQueues/AdditionalPolicies](#) ou [Iam///](#), vous devez autoriser l' AWS ParallelCluster utilisateur à associer et détacher des politiques de rôle pour chaque politique supplémentaire, comme indiqué dans la politique suivante. [Iam](#) Ajoutez les ARN de politique supplémentaires à la condition d'attachement et de détachement des politiques de rôle.

⚠ Warning

Ce mode permet à l'utilisateur de disposer des privilèges d'administrateur IAM dans Compte AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:CreateServiceLinkedRole",
        "iam>DeleteRole",
        "iam:TagRole"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IamRole"
    },
    {
      "Action": [
        "iam:CreateRole"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IamCreateRole"
    },
    {
      "Action": [
        "iam:PutRolePolicy",
        "iam>DeleteRolePolicy"
      ],
      "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
      "Effect": "Allow",
      "Sid": "IamInlinePolicy"
    },
    {
      "Condition": {
```



```

    "ArnLike": {
      "iam:PolicyARN": [
        "arn:aws:iam::<AWS ACCOUNT ID>:policy/parallelcluster*",
        "arn:aws:iam::<AWS ACCOUNT ID>:policy/parallelcluster/*",
        "arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy",
        "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore",
        "arn:aws:iam::aws:policy/AWSBatchFullAccess",
        "arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess",
        "arn:aws:iam::aws:policy/service-role/AWSBatchServiceRole",
        "arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role",
        "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy",
        "arn:aws:iam::aws:policy/service-role/
AmazonEC2SpotFleetTaggingRole",
        "arn:aws:iam::aws:policy/EC2InstanceProfileForImageBuilder",
        "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole"
      ]
    },
    "Action": [
      "iam:AttachRolePolicy",
      "iam:DetachRolePolicy"
    ],
    "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
    "Effect": "Allow",
    "Sid": "IamPolicy"
  }
}

```

Mode d'accès IAM restreint

Lorsqu'aucune politique IAM supplémentaire n'est accordée à l'utilisateur, les rôles IAM requis par les clusters ou la création d'images personnalisées doivent être créés manuellement par un administrateur et transmis dans le cadre de la configuration du cluster.

Lors de la création d'un cluster, les paramètres suivants sont requis :

- [Iam / Roles / LambdaFunctionsRole](#)
- [HeadNode / Iam / InstanceRole](#) | [InstanceProfile](#)

- [Scheduling](#) / [SlurmQueues](#) / [Iam](#) / [InstanceRole](#) | [InstanceProfile](#)

Lors de la création d'une image personnalisée, les paramètres suivants sont requis :

- [Build](#) / [Iam](#) / [InstanceRole](#) | [InstanceProfile](#)
- [Build](#) / [Iam](#) / [CleanupLambdaRole](#)

Les rôles IAM transmis dans le cadre des paramètres listés ci-dessus doivent être créés dans le préfixe de `/parallelcluster/` chemin. Si cela n'est pas possible, la politique utilisateur doit être mise à jour pour accorder des `iam:PassRole` autorisations sur les rôles personnalisés spécifiques, comme dans l'exemple suivant.

```
{
  "Condition": {
    "StringEqualsIfExists": {
      "iam:PassedToService": [
        "ecs-tasks.amazonaws.com",
        "lambda.amazonaws.com",
        "ec2.amazonaws.com",
        "spotfleet.amazonaws.com",
        "batch.amazonaws.com",
        "codebuild.amazonaws.com"
      ]
    }
  },
  "Action": [
    "iam:PassRole"
  ],
  "Resource": [
    <list all custom IAM roles>
  ],
  "Effect": "Allow",
  "Sid": "IamPassRole"
}
```

Warning

Actuellement, ce mode n'autorise pas la gestion des AWS Batch clusters car tous les rôles IAM ne peuvent pas être transmis dans la configuration du cluster.

Mode **PermissionsBoundary**

Ce mode délègue AWS ParallelCluster la création de rôles IAM liés à la limite d'autorisations IAM configurée. Pour plus d'informations sur les limites des autorisations IAM, consultez la section [Limites des autorisations pour les entités IAM](#) dans le guide de l'utilisateur IAM.

La politique suivante doit être ajoutée au rôle d'utilisateur.

Dans la politique, remplacez `< permissions-boundary-arn >` par l'ARN de la politique IAM à appliquer comme limite d'autorisations.

Warning

Si vous configurez les [AdditionalPolicies](#) paramètres [HeadNodeIam//AdditionalPolicies](#) ou [Scheduling/SlurmQueues/Iam/](#), vous devez accorder à l'utilisateur l'autorisation d'attacher et de détacher des politiques de rôle pour chaque politique supplémentaire, comme indiqué dans la politique suivante. Ajoutez les ARN de politique supplémentaires à la condition d'attachement et de détachement des politiques de rôle.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:CreateServiceLinkedRole",
        "iam>DeleteRole",
        "iam:TagRole"
      ],
      "Resource": [
        "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
      ],
      "Effect": "Allow",
      "Sid": "IamRole"
    },
    {
      "Condition": {
        "StringEquals": {
          "iam:PermissionsBoundary": [
            <permissions-boundary-arn>
          ]
        }
      }
    }
  ]
}
```

```

    ]
  }
},
"Action": [
  "iam:CreateRole"
],
"Resource": [
  "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*"
],
"Effect": "Allow",
"Sid": "IamCreateRole"
},
{
  "Condition": {
    "StringEquals": {
      "iam:PermissionsBoundary": [
        <permissions-boundary-arn>
      ]
    }
  },
  "Action": [
    "iam:PutRolePolicy",
    "iam>DeleteRolePolicy"
  ],
  "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
  "Effect": "Allow",
  "Sid": "IamInlinePolicy"
},
{
  "Condition": {
    "StringEquals": {
      "iam:PermissionsBoundary": [
        <permissions-boundary-arn>
      ]
    }
  },
  "ArnLike": {
    "iam:PolicyARN": [
      "arn:aws:iam::<AWS ACCOUNT ID>:policy/parallelcluster*",
      "arn:aws:iam::<AWS ACCOUNT ID>:policy/parallelcluster/*",
      "arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy",
      "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore",
      "arn:aws:iam::aws:policy/AWSBatchFullAccess",
      "arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess",
      "arn:aws:iam::aws:policy/service-role/AWSBatchServiceRole",

```

```

        "arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role",
        "arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy",
        "arn:aws:iam::aws:policy/service-role/
AmazonEC2SpotFleetTaggingRole",
        "arn:aws:iam::aws:policy/EC2InstanceProfileForImageBuilder",
        "arn:aws:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole"
    ]
  },
  "Action": [
    "iam:AttachRolePolicy",
    "iam:DetachRolePolicy"
  ],
  "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
  "Effect": "Allow",
  "Sid": "IamPolicy"
}
]
}

```

Lorsque ce mode est activé, vous devez spécifier l'ARN de la limite d'autorisations dans le paramètre [Iam/PermissionsBoundary](#) configuration lors de la création ou de la mise à jour d'un cluster et dans le [PermissionBoundary](#) paramètre [BuildIam](#) lors de la création d'une image personnalisée.

AWS ParallelCluster paramètres de configuration pour gérer les autorisations IAM

AWS ParallelCluster expose une série d'options de configuration permettant de personnaliser et de gérer les autorisations et les rôles IAM utilisés dans un cluster ou lors du processus de création d'une AMI personnalisée.

Rubriques

- [Configuration de cluster](#)
- [Configuration d'image personnalisée](#)

Configuration de cluster

Rubriques

- [Rôle IAM du nœud principal](#)
- [Accès à Amazon S3](#)
- [Politiques IAM supplémentaires](#)
- [AWS Lambda fonctions \(rôle\)](#)
- [Rôle IAM des nœuds de calcul](#)
- [Limite d'autorisations](#)

Rôle IAM du nœud principal

[HeadNode](#) / [Iam](#) / [InstanceRole](#) | [InstanceProfile](#)

Avec cette option, vous remplacez le rôle IAM par défaut attribué au nœud principal du cluster. Pour plus de détails, veuillez vous référer à la [InstanceProfile](#) référence.

Voici l'ensemble minimal de politiques à utiliser dans le cadre de ce rôle lorsque le planificateur est Slurm :

- `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy` politique IAM gérée. Pour plus d'informations, consultez la section [Créer des rôles et des utilisateurs IAM à utiliser avec l' CloudWatch agent](#) dans le guide de l' CloudWatch utilisateur Amazon.
- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` politique IAM gérée. Pour plus d'informations, consultez [les politiques AWS gérées AWS Systems Manager](#) dans le guide de AWS Systems Manager l'utilisateur.
- Politique IAM supplémentaire :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::<REGION>-aws-parallelcluster/*",
        "arn:aws:s3:::dcv-license.<REGION>/*",

```

```

        "arn:aws:s3:::parallelcluster-*~v1-do-not-delete/*"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "dynamodb:GetItem",
        "dynamodb:PutItem",
        "dynamodb:UpdateItem",
        "dynamodb:BatchWriteItem",
        "dynamodb:BatchGetItem"
    ],
    "Resource": "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/parallelcluster-*",
    "Effect": "Allow"
},
{
    "Condition": {
        "StringEquals": {
            "ec2:ResourceTag/parallelcluster:node-type": "Compute"
        }
    },
    "Action": "ec2:TerminateInstances",
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": [
        "ec2:RunInstances",
        "ec2:CreateFleet"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": [
                "ec2.amazonaws.com"
            ]
        }
    },
    "Action": [
        "iam:PassRole"
    ]
}

```

```
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
      "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "ec2:DescribeInstances",
      "ec2:DescribeInstanceStatus",
      "ec2:DescribeVolumes",
      "ec2:DescribeInstanceAttribute",
      "ec2:DescribeCapacityReservations"
    ],
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "ec2:CreateTags",
      "ec2:AttachVolume"
    ],
    "Resource": [
      "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:instance/*",
      "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:volume/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "cloudformation:DescribeStacks",
      "cloudformation:DescribeStackResource",
      "cloudformation:SignalResource"
    ],
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "route53:ChangeResourceRecordSets"
    ],
    "Resource": "*",
    "Effect": "Allow"
  }
```



```

    },
    {
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "arn:aws:secretsmanager:<REGION>:<AWS ACCOUNT
ID>:secret:<SECRET_ID>",
      "Effect": "Allow"
    }
  ]
}

```

Notez que dans le cas où [SchedulingSlurmQueues/Iam//InstanceRole](#) est utilisé pour remplacer le rôle IAM de calcul, la politique du nœud principal décrite ci-dessus doit inclure ce rôle dans la Resource section de l'iam:PassRole autorisation.

Voici l'ensemble minimal de politiques à utiliser dans le cadre de ce rôle lorsque le planificateur est : AWS Batch

- `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy` politique IAM gérée. Pour plus d'informations, consultez la section [Créer des rôles et des utilisateurs IAM à utiliser avec l'CloudWatch agent](#) dans le guide de l'CloudWatch utilisateur Amazon.
- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` politique IAM gérée. Pour plus d'informations, consultez [les politiques AWS gérées AWS Systems Manager](#) dans le guide de AWS Systems Manager l'utilisateur.
- Politique IAM supplémentaire :

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::parallelcluster-*-*v1-do-not-delete/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": "s3:GetObject",

```

```

    "Resource": [
      "arn:aws:s3:::dcv-license.<REGION>/*",
      "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
    ],
    "Effect": "Allow"
  },
  {
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": [
          "batch.amazonaws.com"
        ]
      }
    },
    "Action": [
      "iam:PassRole"
    ],
    "Resource": [
      "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
      "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "batch:DescribeJobQueues",
      "batch:DescribeJobs",
      "batch:ListJobs",
      "batch:DescribeComputeEnvironments"
    ],
    "Resource": "*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "batch:SubmitJob",
      "batch:TerminateJob",
      "logs:GetLogEvents",
      "ecs:ListContainerInstances",
      "ecs:DescribeContainerInstances"
    ],
    "Resource": [
      "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:log-group:/aws/batch/job:log-
stream:PclusterJobDefinition*"

```

```

        "arn:aws:ecs:<REGION>:<AWS ACCOUNT ID>:container-instance/AWSBatch-
PclusterComputeEnviron*",
        "arn:aws:ecs:<REGION>:<AWS ACCOUNT ID>:cluster/AWSBatch-Pcluster*",
        "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-queue/
PclusterJobQueue*",
        "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job-definition/
PclusterJobDefinition*:*",
        "arn:aws:batch:<REGION>:<AWS ACCOUNT ID>:job/*"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeVolumes",
        "ec2:DescribeInstanceAttribute"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": [
        "ec2:CreateTags",
        "ec2:AttachVolume"
    ],
    "Resource": [
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:instance/*",
        "arn:aws:ec2:<REGION>:<AWS ACCOUNT ID>:volume/*"
    ],
    "Effect": "Allow"
},
{
    "Action": [
        "cloudformation:DescribeStackResource",
        "cloudformation:DescribeStacks",
        "cloudformation:SignalResource"
    ],
    "Resource": "*",
    "Effect": "Allow"
},
{
    "Action": "secretsmanager:GetSecretValue",

```

```
    "Resource": "arn:aws:secretsmanager:<REGION>:<AWS ACCOUNT ID>:secret:<SECRET_ID>",
    "Effect": "Allow"
  }
]
```

Accès à Amazon S3

[HeadNode/Iam/S3Access](#) ou [Scheduling/SlurmQueues/S3Access](#)

Dans ces sections de configuration, vous pouvez personnaliser l'accès Amazon S3 en accordant des politiques Amazon S3 supplémentaires aux rôles IAM associés au nœud principal ou aux nœuds de calcul du cluster lorsque ces rôles sont créés par AWS ParallelCluster. Pour plus d'informations, consultez la documentation de référence pour chacun des paramètres de configuration.

Ce paramètre ne peut être utilisé que lorsque l'utilisateur est configuré avec [Mode d'accès IAM privilégié](#) ou [Mode PermissionsBoundary](#).

Politiques IAM supplémentaires

[HeadNode/Iam/AdditionalIamPolicies](#) ou [SlurmQueues/Iam/AdditionalIamPolicies](#)

Utilisez cette option pour associer des politiques IAM gérées supplémentaires aux rôles IAM associés au nœud principal ou aux nœuds de calcul du cluster lorsque ces rôles sont créés par AWS ParallelCluster

Warning

Pour utiliser cette option, assurez-vous que l'[AWS ParallelCluster utilisateur iam:DetachRolePolicy](#) dispose `iam:AttachRolePolicy` des autorisations nécessaires pour les politiques IAM qui doivent être associées.

AWS Lambda fonctions (rôle)

[Iam / Roles / LambdaFunctionsRole](#)

Cette option remplace le rôle attaché à toutes les AWS Lambda fonctions utilisées lors du processus de création du cluster. AWS Lambda doit être configuré en tant que principal autorisé à assumer le rôle.

Note

Si [DeploymentSettings/LambdaFunctionsVpcConfig](#) est défini, `LambdaFunctionsRole` il doit inclure l'[autorisation de AWS Lambda rôle](#) pour définir la configuration du VPC.

Voici l'ensemble minimal de politiques à utiliser dans le cadre de ce rôle :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "route53:ListResourceRecordSets",
        "route53:ChangeResourceRecordSets"
      ],
      "Resource": "arn:aws:route53:::hostedzone/*",
      "Effect": "Allow"
    },
    {
      "Action": ["logs:CreateLogStream", "logs:PutLogEvents"],
      "Effect": "Allow",
      "Resource": "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:log-group:/aws/lambda/
pcluster-*"
    },
    {
      "Action": "ec2:DescribeInstances",
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": "ec2:TerminateInstances",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/parallelcluster:node-type": "Compute"
        }
      },
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

```

    "Action": [
      "s3:DeleteObject",
      "s3:DeleteObjectVersion",
      "s3:ListBucket",
      "s3:ListBucketVersions"
    ],
    "Effect": "Allow",
    "Resource": [
      "arn:aws:s3:::parallelcluster-*-*v1-do-not-delete",
      "arn:aws:s3:::parallelcluster-*-*v1-do-not-delete/*"
    ]
  }
]
}

```

Rôle IAM des nœuds de calcul

[Scheduling](#) / [SlurmQueues](#) / [Iam](#) / [InstanceRole](#) | [InstanceProfile](#)

Cette option permet de remplacer le rôle IAM attribué aux nœuds de calcul du cluster. Pour plus d'informations, consultez [InstanceProfile](#).

Voici l'ensemble minimal de politiques à utiliser dans le cadre de ce rôle :

- `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy` politique IAM gérée. Pour plus d'informations, consultez la section [Créer des rôles et des utilisateurs IAM à utiliser avec l'CloudWatchagent](#) dans le guide de l' CloudWatch utilisateur Amazon.
- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` politique IAM gérée. Pour plus d'informations, consultez [les politiques AWS gérées AWS Systems Manager](#) dans le guide de AWS Systems Manager l'utilisateur.
- Politique IAM supplémentaire :

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "dynamodb:Query",
        "dynamodb:UpdateItem",
        "dynamodb:PutItem",
        "dynamodb:GetItem"
      ],
    }
  ],
}

```

```

        "Resource": "arn:aws:dynamodb:<REGION>:<AWS ACCOUNT ID>:table/
parallelcluster-*",
        "Effect": "Allow"
    },
    {
        "Action": "s3:GetObject",
        "Resource": [
            "arn:aws:s3:::<REGION>-aws-parallelcluster/*"
        ],
        "Effect": "Allow"
    },
    {
        "Action": "ec2:DescribeInstanceAttribute",
        "Resource": "*",
        "Effect": "Allow"
    },
    {
        "Action": "cloudformation:DescribeStackResource",
        "Resource": [
            "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/*/*" ],
        "Effect": "Allow"
    }
]
}

```

Limite d'autorisations

[Iam / PermissionsBoundary](#)

Ce paramètre oblige AWS ParallelCluster à associer la politique IAM donnée `PermissionsBoundary` à tous les rôles IAM créés dans le cadre d'un déploiement de cluster.

Consultez [Mode PermissionsBoundary](#) la liste des politiques requises par l'utilisateur lorsque ce paramètre est défini.

Configuration d'image personnalisée

Rubriques

- [Rôle d'instance pour EC2 Image Builder](#)
- [AWS Lambda rôle de nettoyage](#)

- [Politiques IAM supplémentaires](#)
- [Limite d'autorisations](#)

Rôle d'instance pour EC2 Image Builder

[Build](#) / [Iam](#) / [InstanceRole](#) | [InstanceProfile](#)

Cette option vous permet de remplacer le rôle IAM attribué à l'instance Amazon EC2 lancée par EC2 Image Builder pour créer une AMI personnalisée.

Voici l'ensemble minimal de politiques à utiliser dans le cadre de ce rôle :

- `arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore` politique IAM gérée. Pour plus d'informations, consultez [les politiques AWS gérées AWS Systems Manager](#) dans le guide de AWS Systems Manager l'utilisateur.
- `arn:aws:iam::aws:policy/EC2InstanceProfileForImageBuilder` politique IAM gérée. Pour plus d'informations, consultez la [EC2InstanceProfileForImageBuilder](#) politique dans le guide de l'utilisateur d'Image Builder.
- Politique IAM supplémentaire :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ec2:CreateTags",
        "ec2:ModifyImageAttribute"
      ],
      "Resource": "arn:aws:ec2:<REGION>::image/*",
      "Effect": "Allow"
    }
  ]
}
```

AWS Lambda rôle de nettoyage

[Build](#) / [Iam](#) / [CleanupLambdaRole](#)

Cette option remplace le rôle attaché à toutes les AWS Lambda fonctions utilisées lors du processus de création d'image personnalisée. AWS Lambda doit être configuré en tant que principal autorisé à assumer le rôle.

Note

Si `DeploymentSettings/LambdaFunctionsVpcConfig` est défini, `CleanupLambdaRole` il doit inclure l'[autorisation de AWS Lambda rôle](#) pour définir la configuration du VPC.

Voici l'ensemble minimal de politiques à utiliser dans le cadre de ce rôle :

- `arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole` politique IAM gérée. Pour plus d'informations, consultez la section [Politiques AWS gérées pour les fonctionnalités Lambda](#) dans le Guide du AWS Lambda développeur.
- Politique IAM supplémentaire :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:DetachRolePolicy",
        "iam>DeleteRole",
        "iam>DeleteRolePolicy"
      ],
      "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:role/parallelcluster/*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "iam>DeleteInstanceProfile",
        "iam:RemoveRoleFromInstanceProfile"
      ],
      "Resource": "arn:aws:iam::<AWS ACCOUNT ID>:instance-profile/parallelcluster/*",
      "Effect": "Allow"
    },
    {
      "Action": "imagebuilder:DeleteInfrastructureConfiguration",
      "Resource": "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT ID>:infrastructure-configuration/parallelclusterimage-*",
```

```

    "Effect": "Allow"
  },
  {
    "Action": [
      "imagebuilder:DeleteComponent"
    ],
    "Resource": [
      "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT ID>:component/
parallelclusterimage-*/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": "imagebuilder:DeleteImageRecipe",
    "Resource": "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT ID>:image-recipe/
parallelclusterimage-*/*",
    "Effect": "Allow"
  },
  {
    "Action": "imagebuilder:DeleteDistributionConfiguration",
    "Resource": "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT ID>:distribution-
configuration/parallelclusterimage-*",
    "Effect": "Allow"
  },
  {
    "Action": [
      "imagebuilder:DeleteImage",
      "imagebuilder:GetImage",
      "imagebuilder:CancelImageCreation"
    ],
    "Resource": "arn:aws:imagebuilder:<REGION>:<AWS ACCOUNT ID>:image/
parallelclusterimage-*/*",
    "Effect": "Allow"
  },
  {
    "Action": "cloudformation:DeleteStack",
    "Resource": "arn:aws:cloudformation:<REGION>:<AWS ACCOUNT ID>:stack/*/*",
    "Effect": "Allow"
  },
  {
    "Action": "ec2:CreateTags",
    "Resource": "arn:aws:ec2:<REGION>::image/*",
    "Effect": "Allow"
  },
},

```

```

    {
      "Action": "tag:TagResources",
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "lambda:DeleteFunction",
        "lambda:RemovePermission"
      ],
      "Resource": "arn:aws:lambda:<REGION>:<AWS ACCOUNT
ID>:function:ParallelClusterImage-*",
      "Effect": "Allow"
    },
    {
      "Action": "logs:DeleteLogGroup",
      "Resource": "arn:aws:logs:<REGION>:<AWS ACCOUNT ID>:log-group:/aws/
lambda/ParallelClusterImage-*:*",
      "Effect": "Allow"
    },
    {
      "Action": [
        "SNS:GetTopicAttributes",
        "SNS:DeleteTopic",
        "SNS:GetSubscriptionAttributes",
        "SNS:Unsubscribe"
      ],
      "Resource": "arn:aws:sns:<REGION>:<AWS ACCOUNT ID>:ParallelClusterImage-
*",
      "Effect": "Allow"
    }
  ]
}

```

Politiques IAM supplémentaires

[Build](#) / [Iam](#) / [AdditionalIamPolicies](#)

Vous utilisez cette option pour associer des politiques IAM gérées supplémentaires au rôle associé à l'instance Amazon EC2 utilisée par EC2 Image Builder pour produire l'AMI personnalisée.

⚠ Warning

Pour utiliser cette option, assurez-vous que l'[AWS ParallelCluster utilisateur](#) `iam:DetachRolePolicy` dispose `iam:AttachRolePolicy` des autorisations nécessaires pour les politiques IAM qui doivent être associées.

Limite d'autorisations

[Build](#) / [Iam](#) / [PermissionsBoundary](#)

Ce paramètre oblige AWS ParallelCluster à associer la politique IAM donnée `PermissionsBoundary` à tous les rôles IAM créés dans le cadre de la création d'une AMI personnalisée.

Consultez [Mode PermissionsBoundary](#) la liste des politiques requises pour utiliser ces fonctionnalités.

Configurations réseau

AWS ParallelCluster utilise Amazon Virtual Private Cloud (VPC) pour la mise en réseau. Le VPC fournit une plate-forme réseau flexible et configurable sur laquelle vous pouvez déployer des clusters.

Le VPC doit avoir les options `DNS Resolution = yes`, `DNS Hostnames = yes` et DHCP avec le nom de domaine correct pour la région. Le jeu d'options DHCP par défaut spécifie déjà le `AmazonProvidedDNS` requis. Si vous spécifiez plusieurs serveurs de noms de domaine, consultez les [ensembles d'options DHCP](#) dans le guide de l'utilisateur Amazon VPC.

AWS ParallelCluster prend en charge les configurations de haut niveau suivantes :

- Un sous-réseau pour les nœuds de tête et de calcul.
- Deux sous-réseaux, avec le nœud principal dans un sous-réseau public et des nœuds de calcul dans un sous-réseau privé. Les sous-réseaux peuvent être nouveaux ou existants.

Toutes ces configurations peuvent fonctionner avec ou sans adressage IP public. AWS ParallelCluster peut également être déployé pour utiliser un proxy HTTP pour toutes les AWS requêtes. Les combinaisons de ces configurations se traduisent par de nombreux scénarios de déploiement. Par exemple, vous pouvez configurer un sous-réseau public unique avec tous les accès

via Internet. Vous pouvez également configurer un réseau entièrement privé à l'aide AWS Direct Connect d'un proxy HTTP pour l'ensemble du trafic.

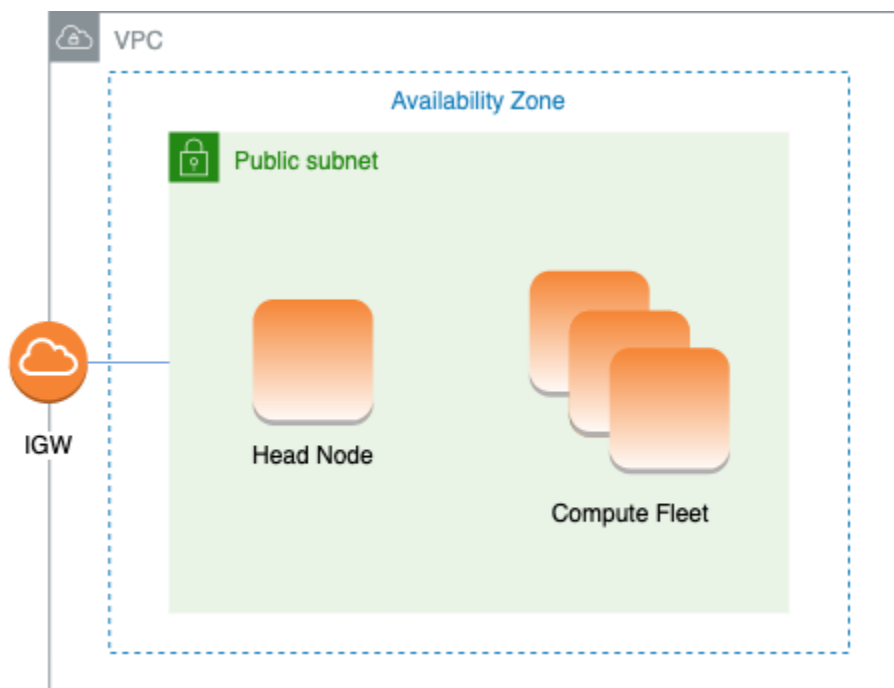
À partir de la AWS ParallelCluster version 3.0.0SecurityGroups, AdditionalSecurityGroups il est possible de configurer différents PlacementGroup paramètres pour chaque file d'attente. Pour plus d'informations, consultez [HeadNode/Networking](#) et [SlurmQueues/Networking](#) et [AwsBatchQueues/Networking](#).

Pour des illustrations de certains scénarios de mise en réseau, reportez-vous aux diagrammes d'architecture suivants.

Rubriques

- [AWS ParallelCluster dans un seul sous-réseau public](#)
- [AWS ParallelCluster en utilisant deux sous-réseaux](#)
- [AWS ParallelCluster dans un seul sous-réseau privé connecté à l'aide de AWS Direct Connect](#)
- [AWS ParallelCluster avec AWS Batch planificateur](#)
- [AWS ParallelCluster dans un seul sous-réseau sans accès à Internet](#)

AWS ParallelCluster dans un seul sous-réseau public



La configuration de cette architecture nécessite les paramètres suivants :

```
# Note that all values are only provided as examples
HeadNode:
  ...
  Networking:
    SubnetId: subnet-12345678 # subnet with internet gateway
    #ElasticIp: true | false | eip-12345678
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - ...
    Networking:
      SubnetIds:
        - subnet-12345678 # subnet with internet gateway
      #AssignPublicIp: true
```

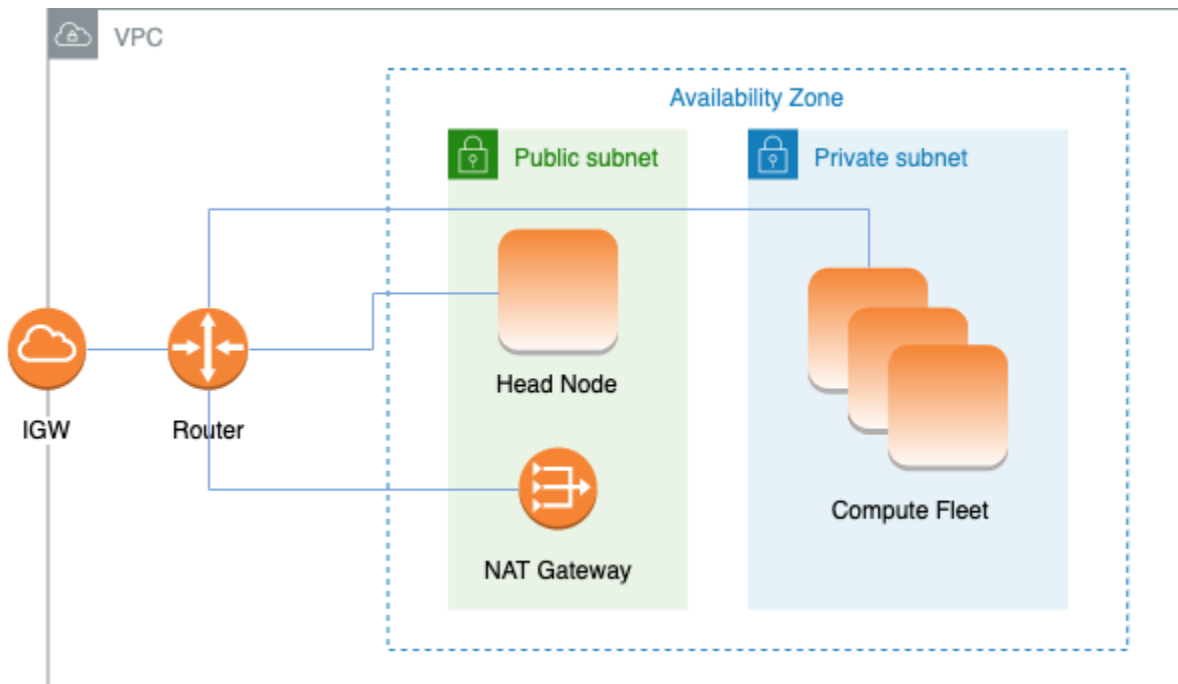
Dans cette configuration, toutes les instances du cluster doivent se voir attribuer une adresse IP publique afin d'accéder à Internet. Pour ce faire, procédez comme suit :

- Assurez-vous qu'une adresse IP publique est attribuée au nœud principal en activant le paramètre « Activer l'attribution automatique de l'adresse IPv4 publique » pour le sous-réseau utilisé dans [HeadNode/Networking/SubnetId](#) ou en attribuant une adresse IP élastique dans [HeadNodeNetworkingElasticIp](#).
- Assurez-vous qu'une adresse IP publique est attribuée aux nœuds de calcul en activant le paramètre « Activer l'attribution automatique de l'adresse IPv4 publique » pour le sous-réseau utilisé dans [Scheduling//SlurmQueuesNetworking/SubnetIds](#) ou en définissant [AssignPublicIp: true](#) dans [Scheduling//SlurmQueuesNetworking](#).
- Si vous définissez un type d'instance p4d, ou un autre type d'instance doté de plusieurs interfaces réseau ou d'une carte d'interface réseau reliée au nœud principal, vous devez définir [HeadNode/Networking](#) sur [ElasticIp>true](#) pour fournir un accès public. AWS les adresses IP publiques ne peuvent être attribuées qu'aux instances lancées avec une seule interface réseau. Dans ce cas, nous vous recommandons d'utiliser une [passerelle NAT](#) pour fournir un accès public aux nœuds de calcul du cluster. Pour plus d'informations sur les adresses IP, consultez la section [Attribuer une adresse IPv4 publique lors du lancement de l'instance](#) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.
- Vous ne pouvez pas définir un type d'instance hp6id ou un autre type d'instance doté de plusieurs interfaces réseau ou d'une carte d'interface réseau pour calculer des nœuds, car les adresses IP AWS publiques ne peuvent être attribuées qu'aux instances lancées avec une seule interface réseau. Pour plus d'informations sur les adresses IP, consultez la section [Attribuer une](#)

[adresse IPv4 publique lors du lancement de l'instance](#) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Pour plus d'informations, consultez la section [Activation de l'accès à Internet](#) dans le guide de l'utilisateur Amazon VPC.

AWS ParallelCluster en utilisant deux sous-réseaux



La configuration pour utiliser un sous-réseau privé existant pour les instances de calcul nécessite les paramètres suivants :

```
# Note that all values are only provided as examples
HeadNode:
  ...
  Networking:
    SubnetId: subnet-12345678 # subnet with internet gateway
    #ElasticIp: true | false | eip-12345678
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - ...
      Networking:
        SubnetIds:
          - subnet-23456789 # subnet with NAT gateway
```

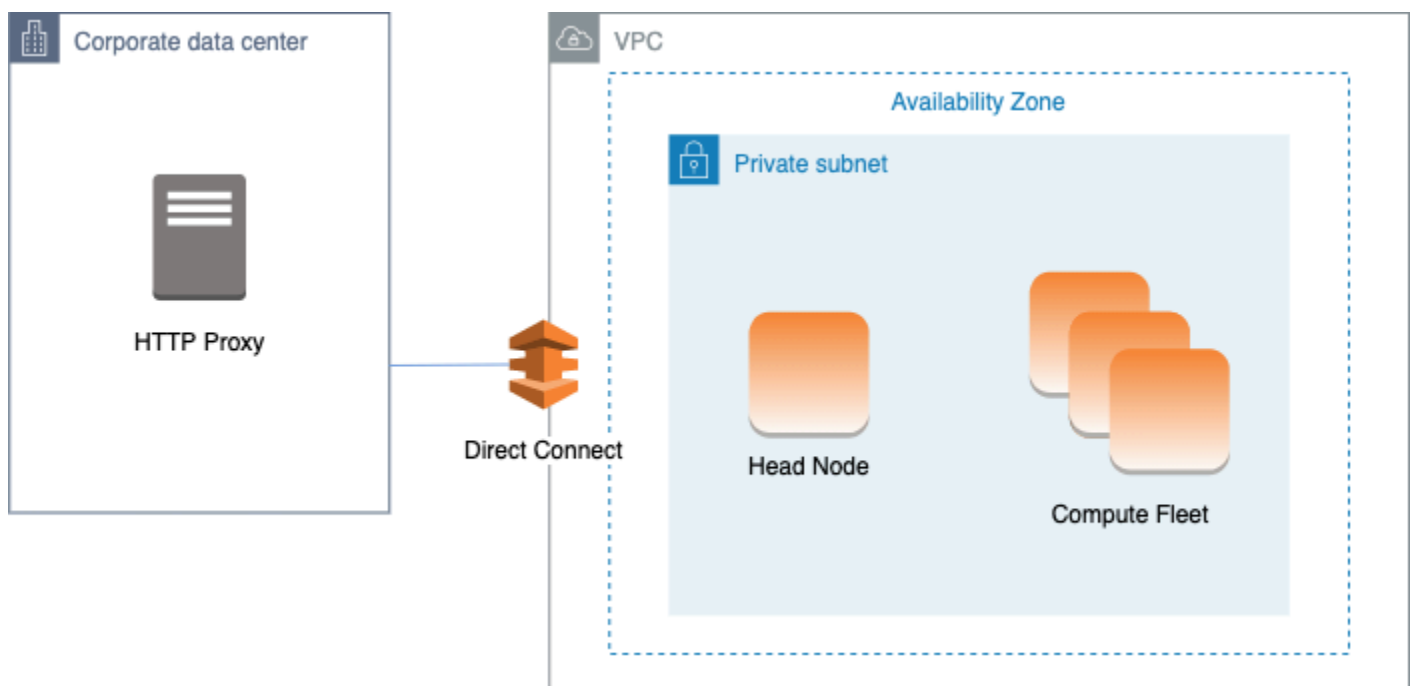
```
#AssignPublicIp: false
```

Dans cette configuration, seul le nœud principal du cluster doit disposer d'une adresse IP publique. Vous pouvez y parvenir en activant le paramètre « Activer l'attribution automatique de l'adresse IPv4 publique » pour le sous-réseau utilisé dans [HeadNode/Networking/SubnetId](#) ou en attribuant une adresse IP élastique dans [HeadNodeNetworkingElasticIp](#).

Si vous définissez un type d'instance p4d ou un autre type d'instance doté de plusieurs interfaces réseau ou d'une carte d'interface réseau reliée au nœud principal, vous devez définir [HeadNode/Networking/ElasticIp](#) sur `true` pour fournir un accès public. AWS les adresses IP publiques ne peuvent être attribuées qu'aux instances lancées avec une seule interface réseau. Pour plus d'informations sur les adresses IP, consultez la section [Attribuer une adresse IPv4 publique lors du lancement de l'instance](#) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Cette configuration nécessite une [passerelle NAT](#) ou un proxy interne dans le sous-réseau utilisé pour les files d'attente, afin de donner accès à Internet aux instances de calcul.

AWS ParallelCluster dans un seul sous-réseau privé connecté à l'aide de AWS Direct Connect



La configuration de cette architecture nécessite les paramètres suivants :

```
# Note that all values are only provided as examples
```



```
HeadNode:
  ...
  Networking:
    SubnetId: subnet-34567890 # subnet with proxy
    Proxy:
      HttpProxyAddress: http://proxy-address:port
  Ssh:
    KeyName: ec2-key-name
Scheduling:
  Scheduler: slurm
  SlurmQueues:
  - ...
    Networking:
      SubnetIds:
        - subnet-34567890 # subnet with proxy
      AssignPublicIp: false
      Proxy:
        HttpProxyAddress: http://proxy-address:port
```

Lorsque [Scheduling/SlurmQueues/Networking/AssignPublicIp](#) est défini sur `false`, les sous-réseaux doivent être correctement configurés pour utiliser le proxy pour tout le trafic. L'accès au Web est requis pour les nœuds de tête et de calcul.

AWS ParallelCluster avec AWS Batch planificateur

Lorsque vous l'utilisez `awsbatch` comme type de planificateur, AWS ParallelCluster crée un environnement informatique AWS Batch géré. L'AWS Batch environnement gère les instances de conteneur Amazon Elastic Container Service (Amazon ECS). Ces instances sont lancées dans le sous-réseau configuré dans le [SubnetIds](#) paramètre [AwsBatchQueues/Networking/](#). AWS Batch Pour fonctionner correctement, les instances de conteneur Amazon ECS ont besoin d'un accès réseau externe pour communiquer avec le point de terminaison du service Amazon ECS. Cela se traduit par les scénarios suivants :

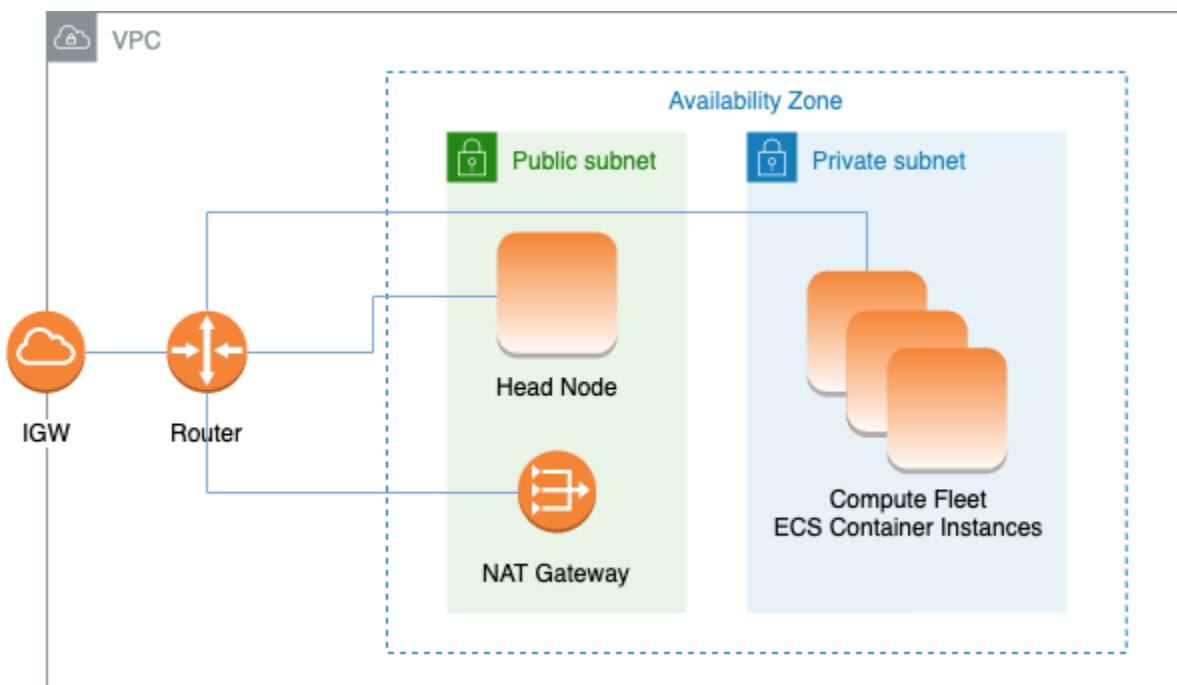
- L'ID de sous-réseau spécifié pour la file d'attente utilise une [passerelle NAT](#) pour accéder à Internet. Nous avons recommandé cette approche.
- Les instances lancées dans le sous-réseau de file d'attente possèdent des adresses IP publiques et peuvent accéder à Internet via une passerelle Internet.

De plus, si vous êtes intéressé par les jobs parallèles sur plusieurs nœuds (voir la [AWS Batch documentation](#)) :

AWS Batch les tâches parallèles à nœuds multiples utilisent le mode `aws-vc` réseau Amazon ECS. Cela 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 Amazon VPC que sa ressource de calcul hôte. Tous les groupes de sécurité appliqués à vos ressources de calcul lui sont également appliqués.

Lorsque vous utilisez Amazon ECS Task Networking, le mode `aws-vc` réseau ne fournit pas d'interfaces réseau élastiques avec des adresses IP publiques pour les tâches utilisant le type de lancement Amazon EC2. Pour accéder à Internet, les tâches utilisant le type de lancement Amazon EC2 doivent être lancées dans un sous-réseau privé configuré pour utiliser une passerelle NAT.

Vous devez configurer une [passerelle NAT](#) afin de permettre au cluster d'exécuter des tâches parallèles sur plusieurs nœuds.



Toutes les configurations et considérations précédentes sont également valables pour AWS Batch. Voici un exemple de configuration AWS Batch réseau.

```
# Note that all values are only provided as examples
HeadNode:
  ...
  Networking:
```

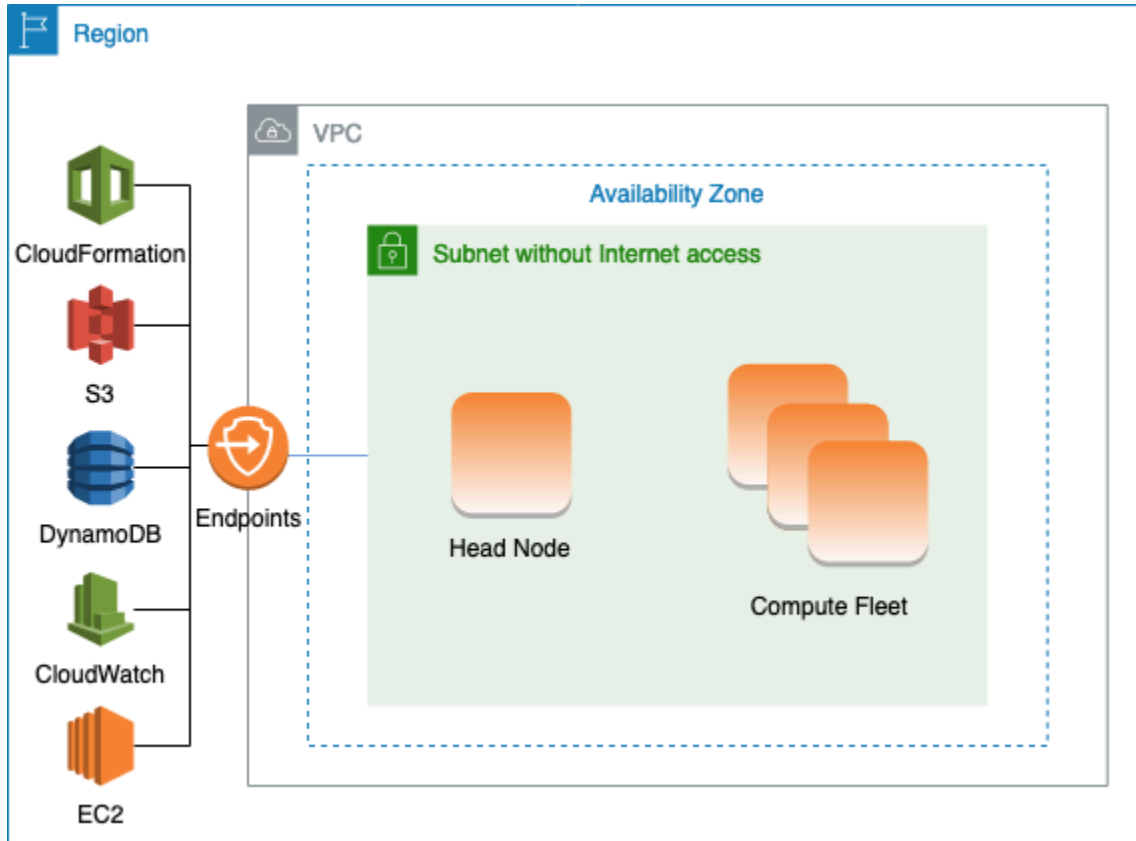
```
SubnetId: subnet-12345678 # subnet with internet gateway, NAT gateway or proxy
#ElasticIp: true | false | eip-12345678
#Proxy:
  #HttpProxyAddress: http://proxy-address:port
Ssh:
  KeyName: ec2-key-name
Scheduling:
  Scheduler: awsbatch
  AwsBatchQueues:
    - ...
  Networking:
    SubnetIds:
      - subnet-23456789 # subnet with internet gateway, NAT gateway or proxy
    #AssignPublicIp: true | false
```

Dans la [Networking](#) section [Scheduling/AwsBatchQueues/](#), [SubnetIds](#) il s'agit d'un type de liste mais, actuellement, un seul sous-réseau est pris en charge.

Pour plus d'informations, consultez les rubriques suivantes :

- [AWS Batch environnements informatiques gérés](#)
- [AWS Batch tâches parallèles sur plusieurs nœuds](#)
- [Mise en réseau des tâches Amazon ECS avec le mode réseau awsvpc](#)

AWS ParallelCluster dans un seul sous-réseau sans accès à Internet



Un sous-réseau sans accès à Internet n'autorise pas les connexions entrantes ou sortantes vers Internet. Cette AWS ParallelCluster configuration peut aider les clients soucieux de la sécurité à renforcer davantage la sécurité de leurs ressources. AWS ParallelCluster les nœuds sont créés à partir d' AWS ParallelCluster AMI qui incluent tous les logiciels nécessaires pour exécuter un cluster sans accès à Internet. De cette façon, AWS ParallelCluster vous pouvez créer et gérer des clusters avec des nœuds qui n'ont pas accès à Internet.

Dans cette section, vous découvrirez comment configurer le cluster. Vous découvrirez également les limites liées à l'exécution de clusters sans accès à Internet.

Configuration des points de terminaison VPC

Pour garantir le bon fonctionnement du cluster, les nœuds du cluster doivent être en mesure d'interagir avec un certain nombre de AWS services.

Créez et configurez les [points de terminaison VPC](#) suivants afin que les nœuds de cluster puissent interagir avec les AWS services, sans accès à Internet :

Commercial and AWS GovCloud (US) partitions

| Service | Nom du service | Type |
|-----------------------|---|-------------|
| Amazon CloudWatch | com.amazonaws. identifiant de <i>région .logs</i> | utilisateur |
| AWS CloudFormation | com.amazonaws. <i>region-id .cloudformation</i> | utilisateur |
| Amazon EC2 | com.amazonaws. identificateur de <i>région .ec2</i> | utilisateur |
| Amazon S3 | com.amazonaws. identificateur de <i>région .s3</i> | Passerelle |
| Amazon DynamoDB | com.amazonaws. identificateur de <i>région .dynamodb</i> | Passerelle |
| AWS Secrets Manager** | com.amazonaws. identifiant de <i>région .secretsmanager</i> | utilisateur |

China partition

| Service | Nom du service | Type |
|--------------------|--|-------------|
| Amazon CloudWatch | com.amazonaws. identifiant de <i>région .logs</i> | utilisateur |
| AWS CloudFormation | cn.com.amazonaws. <i>region-id .cloudformation</i> | utilisateur |
| Amazon EC2 | cn.com.amazonaws. identificateur de <i>région .ec2</i> | utilisateur |

| Service | Nom du service | Type |
|-----------------------|--|-------------|
| Amazon S3 | com.amazonaws. identificateur de <i>région .s3</i> | Passerelle |
| Amazon DynamoDB | com.amazonaws. identificateur de <i>région .dynamodb</i> | Passerelle |
| AWS Secrets Manager** | com.amazonaws. identificateur de <i>région .secretsmanager</i> | utilisateur |

** Ce point de terminaison n'est requis que lorsqu'il [DirectoryService](#) est activé, sinon il est facultatif.

Toutes les instances du VPC doivent disposer de groupes de sécurité appropriés pour communiquer avec les points de terminaison. Vous pouvez le faire en ajoutant des groupes de sécurité [AdditionalSecurityGroups](#) sous [HeadNode](#) et [AdditionalSecurityGroups](#) sous les [SlurmQueues](#) configurations. Par exemple, si les points de terminaison VPC sont créés sans spécifier explicitement de groupe de sécurité, le groupe de sécurité par défaut est associé aux points de terminaison. En ajoutant le groupe de sécurité par défaut à [AdditionalSecurityGroups](#), vous activez la communication entre le cluster et les points de terminaison.

Note

Lorsque vous utilisez des politiques IAM pour restreindre l'accès aux points de terminaison VPC, vous devez ajouter les éléments suivants au point de terminaison VPC Amazon S3 :

```
PolicyDocument:
  Version: 2012-10-17
  Statement:
    - Effect: Allow
      Principal: "*"
      Action:
        - "s3:PutObject"
      Resource:
```

```
- !Sub "arn:${AWS::Partition}:s3:::cloudformation-waitcondition-
${AWS::Region}/*"
```

Désactivez Route 53 et utilisez les noms d'hôte Amazon EC2

Lors de la création d'un Slurm cluster, AWS ParallelCluster crée une zone hébergée Route 53 privée qui est utilisée pour résoudre les noms d'hôte des nœuds de calcul personnalisés, tels que `{queue_name}-{st|dy}-{compute_resource}-{N}`. Route 53 ne prenant pas en charge les points de terminaison VPC, cette fonctionnalité doit être désactivée. En outre, AWS ParallelCluster doit être configuré pour utiliser les noms d'hôte Amazon EC2 par défaut, tels que `ip-1-2-3-4`. Appliquez les paramètres suivants à la configuration de votre cluster :

```
...
Scheduling:
  ...
  SlurmSettings:
    Dns:
      DisableManagedDns: true
      UseEc2Hostnames: true
```

Warning

Pour les clusters créés avec [SlurmSettingsDns//DisableManagedDns](#) et [UseEc2Hostnames](#) définis sur `true`, le DNS Slurm NodeName ne résout pas le problème. Utilisez Slurm NodeHostName plutôt le.

Note

Cette note n'est pas pertinente à partir de AWS ParallelCluster la version 3.3.0. Pour les versions AWS ParallelCluster prises en charge avant la version 3.3.0 : Lorsqu'il `UseEc2Hostnames` est défini sur `true`, le fichier Slurm de configuration est défini avec les `epilog` scripts AWS ParallelCluster `prolog` et :

- `prolog`'exécute pour ajouter des informations `/etc/hosts` sur les nœuds aux nœuds de calcul lorsque chaque tâche est allouée.
- `epilog`'exécute pour nettoyer le contenu écrit par `prolog`.

Pour ajouter des epilog scripts personnalisés prolog ou personnalisés, ajoutez-les respectivement aux `/opt/slurm/etc/pcluster/epilog.d/` dossiers `/opt/slurm/etc/pcluster/prolog.d/` or.

Configuration du cluster

Découvrez comment configurer votre cluster pour qu'il s'exécute dans un sous-réseau sans connexion Internet.

La configuration de cette architecture nécessite les paramètres suivants :

```
# Note that all values are only provided as examples
...
HeadNode:
  ...
  Networking:
    SubnetId: subnet-1234567890abcdef0 # the VPC of the subnet needs to have VPC
    endpoints
    AdditionalSecurityGroups:
      - sg-abcdef01234567890 # optional, the security group that enables the
        communication between the cluster and the VPC endpoints
  Scheduling:
    Scheduler: Slurm # Cluster in a subnet without internet access is supported only when
    the scheduler is Slurm.
    SlurmSettings:
      Dns:
        DisableManagedDns: true
        UseEc2Hostnames: true
    SlurmQueues:
      - ...
        Networking:
          SubnetIds:
            - subnet-1234567890abcdef0 # the VPC of the subnet needs to have VPC
            endpoints attached
          AdditionalSecurityGroups:
            - sg-1abcdef01234567890 # optional, the security group that enables the
            communication between the cluster and the VPC endpoints
```

- [SubnetId\(s\)](#): le sous-réseau sans accès à Internet.

Pour permettre la communication entre AWS ParallelCluster et les AWS services, les points de terminaison du VPC doivent être attachés au VPC du sous-réseau. Avant de créer votre cluster, vérifiez que [l'attribution automatique de l'adresse IPv4 publique est désactivée](#) dans le sous-réseau afin de garantir que les `pcluster` commandes ont accès au cluster.

- [AdditionalSecurityGroups](#): le groupe de sécurité qui permet la communication entre le cluster et les points de terminaison VPC.

Facultatif :

- Si les points de terminaison du VPC sont créés sans spécifier explicitement de groupe de sécurité, le groupe de sécurité par défaut du VPC est associé. Par conséquent, indiquez le groupe de sécurité par défaut à `AdditionalSecurityGroups`.
- Si des groupes de sécurité personnalisés sont utilisés lors de la création du cluster et/ou des points de terminaison VPC, cela n'`AdditionalSecurityGroup`est pas nécessaire tant que les groupes de sécurité personnalisés permettent la communication entre le cluster et les points de terminaison VPC.
- [Scheduler](#): Le planificateur de clusters.

`slurmest` la seule valeur valide. Seul le Slurm planificateur prend en charge un cluster dans un sous-réseau sans accès à Internet.

- [SlurmSettings](#): Les Slurm paramètres.

Consultez la section précédente Désactiver Route53 et utiliser les noms d'hôte Amazon EC2.

Limites

- Connexion au nœud principal via SSH ou NICE DCV : lors de la connexion à un cluster, assurez-vous que le client de la connexion peut atteindre le nœud principal du cluster via son adresse IP privée. Si le client ne se trouve pas dans le même VPC que le nœud principal, utilisez une instance de proxy dans un sous-réseau public du VPC. Cette exigence s'applique à la fois aux connexions SSH et DCV. L'adresse IP publique d'un nœud principal n'est pas accessible si le sous-réseau n'a pas accès à Internet. Les `dcv-connect` commandes `pcluster ssh` et utilisent l'adresse IP publique si elle existe ou l'adresse IP privée. Avant de créer votre cluster, vérifiez que [l'attribution automatique de l'adresse IPv4 publique est désactivée](#) dans le sous-réseau afin de garantir que les `pcluster` commandes ont accès au cluster.

L'exemple suivant montre comment vous connecter à une session DCV exécutée dans le nœud principal de votre cluster. Vous vous connectez via une instance proxy Amazon EC2. L'instance fonctionne comme un serveur DCV NICE pour votre PC et comme client pour le nœud principal du sous-réseau privé.

Connect via DCV via une instance de proxy dans un sous-réseau public :

1. Créez une instance Amazon EC2 dans un sous-réseau public, qui se trouve dans le même VPC que le sous-réseau du cluster.
 2. Assurez-vous que le client et le serveur NICE DCV sont installés sur votre instance Amazon EC2.
 3. Attachez une politique AWS ParallelCluster utilisateur à l'instance proxy Amazon EC2. Pour plus d'informations, consultez [AWS ParallelCluster exemples de politiques `pcluster` utilisateur](#).
 4. Installez-le AWS ParallelCluster sur l'instance proxy Amazon EC2.
 5. Connect via DCV à l'instance proxy Amazon EC2.
 6. Utilisez la `pcluster dcv-connect` commande sur l'instance de proxy pour vous connecter au cluster à l'intérieur du sous-réseau sans accès à Internet.
- Interaction avec d'autres AWS services : seuls les services strictement requis par AWS ParallelCluster sont répertoriés ci-dessus. Si votre cluster doit interagir avec d'autres services, créez les points de terminaison VPC correspondants.

Nœuds de connexion

À partir de la version 3.7.0, les administrateurs de AWS ParallelCluster cluster peuvent configurer des nœuds de connexion qui peuvent être utilisés pour permettre aux utilisateurs d'exécuter des tâches plutôt que d'accéder directement au nœud principal du cluster. Les utilisateurs du cluster disposant des autorisations appropriées peuvent utiliser Active Directory ou leurs informations d'identification SSH pour se connecter, soumettre et gérer leurs tâches. Par conséquent, la gestion du cluster peut être améliorée et les risques d'épuisement des ressources du nœud principal nécessaires Slurm à la gestion du cluster peuvent être minimisés. Les utilisateurs connectés auront également accès à tout le stockage partagé du cluster monté sur les nœuds de connexion. Si un nœud de connexion doit être arrêté, les utilisateurs connectés seront avertis à l'avance par le biais de la session shell active qu'ils utilisent.

Les nœuds de connexion sont spécifiés comme des pools dans lesquels un pool définit un groupe de nœuds de connexion ayant la même configuration de ressources. Tous les nœuds de connexion d'un pool sont configurés pour faire partie d'un [équilibreur de charge réseau](#) qui permet de répartir les sessions entre les nœuds de connexion de manière circulaire. La présente implémentation permet de spécifier un pool de nœuds de connexion contenant plusieurs nœuds de connexion.

Sécurité

Les nœuds de connexion héritent des AllowedIPs paramètres [AllowedIps](#) du nœud principal. De cette manière, les administrateurs du cluster peuvent restreindre le niveau de sécurité du cluster en spécifiant le CIDR source ou une liste de préfixes à partir de laquelle les connexions SSH sont autorisées.

Dans la présente implémentation, l'accès au nœud principal n'est pas automatiquement restreint lors de l'activation des nœuds de connexion. Si nécessaire, un administrateur de cluster peut restreindre cet accès en mettant à jour la configuration ssh des nœuds principaux à l'aide de commandes Linux standard. Cela peut également être accompli en spécifiant des groupes de sécurité personnalisés sur le nœud principal en utilisant le `AdditionalSecurityGroups` paramètre de la section nœud principal du fichier ParallelCluster YAML pour refuser les connexions aux utilisateurs non autorisés.

Réseaux

Les nœuds de connexion sont approvisionnés avec une adresse de connexion unique à l'équilibreur de charge réseau configuré pour le pool de nœuds de connexion. Les paramètres de connectivité de l'adresse sont basés sur le type de sous-réseau spécifié dans la configuration du pool de nœuds de connexion.

- Si le sous-réseau est privé, l'adresse sera privée et, pour autoriser l'accès aux nœuds de connexion, l'administrateur du cluster doit configurer un hôte bastion.
- Si le sous-réseau est public, l'adresse sera publique

Toutes les demandes de connexion sont gérées par le Network Load Balancer à l'aide d'un routage circulaire.

Stockage

Tout le stockage partagé configuré sur le cluster à l'aide ParallelCluster , y compris le stockage géré, sera monté sur tous les nœuds de connexion.

Récupérez les informations des nœuds de connexion

Pour récupérer l'adresse de la connexion unique configurée pour accéder aux nœuds de connexion, l'administrateur du cluster peut exécuter la [describe-cluster](#) commande. La commande fournira également plus d'informations sur l'état des nœuds de connexion.

Les nœuds de connexion sont un nouveau type de nœud pris en charge ParallelCluster qui peut être spécifié à l'aide de la [describe-cluster-instances](#) commande lors de la demande de statut d'un type de nœud spécifique.

La disponibilité d'une adresse de connexion unique au pool de nœuds de connexion n'empêche pas l'accès direct à un nœud de connexion spécifique. Cependant, il n'est pas recommandé d'utiliser la connexion directe pour éviter les avertissements du client SSH. Le client SSH stocke les identifiants d'hôte localement pour chaque adresse cible. Étant donné que l'identifiant d'hôte est spécifique à chaque pool, l'utilisation d'adresses IP différentes et/ou d'une adresse de connexion unique peut avoir le même identifiant d'hôte associé à différentes adresses cibles : cela peut provoquer un avertissement de la part du client ssh puisque le même identifiant d'hôte est associé à plusieurs cibles.

Propriétés de l'Imds

L'accès à l'IMDS du nœud de connexion (et aux informations d'identification du profil d'instance) est limité à l'utilisateur root, à l'utilisateur administratif du cluster (`pc-cluster-admin` par défaut) et à l'utilisateur par défaut spécifique au système d'exploitation (`ec2-users` sur Amazon Linux 2 et RedHat, `ubuntu` sur Ubuntu 18.04, sur `centos` CentOS 7).

Pour restreindre l'accès à l'IMDS, AWS ParallelCluster gère une chaîne de `iptables`.

Note

Toute personnalisation `iptables` ou `ip6tables` règle peut interférer avec le mécanisme utilisé pour restreindre l'accès à l'IMDS sur le nœud de connexion. Voir également. [Imds property setting](#)

Cycle de vie des nœuds de connexion

Il n'existe actuellement aucune commande dédiée permettant d'arrêter et de démarrer les nœuds de connexion dans un pool. Pour arrêter les nœuds de connexion dans un pool, l'administrateur du cluster doit mettre à jour la configuration du cluster en spécifiant zéro sur le nombre de nœuds de connexion (`Count: 0`), puis exécuter une [pcluster.update-cluster-v3](#) commande.

Note

Les utilisateurs connectés sont informés de la résiliation de l'instance spécifique et du délai de grâce correspondant. Pendant la période de grâce, aucune nouvelle connexion ne sera autorisée, à l'exception de celles provenant de [l'utilisateur par défaut du cluster](#). Le message affiché est personnalisable par l'administrateur du cluster depuis le nœud principal ou depuis un nœud de connexion éditant le fichier `/opt/parallelcluster/shared_login_nodes/loginmgt_config.json`.

Pour démarrer le pool de nœuds de connexion, l'administrateur du cluster doit restaurer la Count valeur précédente dans la configuration du cluster, puis exécuter une [update-cluster](#) commande.

Autorisations requises pour exécuter le pool de nœuds de connexion

Pour gérer le pool de nœuds de connexion, l'administrateur du cluster doit disposer des autorisations supplémentaires suivantes :

```

- Action:
  - autoscaling:DeleteAutoScalingGroup
  - autoscaling:DeleteLifecycleHook
  - autoscaling:Describe*
  - autoscaling:PutLifecycleHook
  - autoscaling:UpdateAutoScalingGroup
  - elasticloadbalancing:CreateListener
  - elasticloadbalancing:CreateTargetGroup
  - elasticloadbalancing>DeleteListener
  - elasticloadbalancing>DeleteLoadBalancer
  - elasticloadbalancing>DeleteTargetGroup
  - elasticloadbalancing:Describe*
  - elasticloadbalancing:ModifyLoadBalancerAttributes
Resource: '*'
Condition:
  ForAllValues:StringEquals:
    aws:TagKeys: [ "parallelcluster:cluster-name" ]
- Action:
  - autoscaling:CreateAutoScalingGroup
  - elasticloadbalancing:AddTags
  - elasticloadbalancing:CreateLoadBalancer
Resource: '*'

```

Effect: Allow

Actions de bootstrap personnalisées

Si vous définissez les paramètres de [OnNodeStart](#) configuration [HeadNodeCustomActions//](#), AWS ParallelCluster exécute du code arbitraire immédiatement après le démarrage du nœud. Si vous définissez les paramètres de [OnNodeConfigured](#) configuration [HeadNodeCustomActions//](#), AWS ParallelCluster exécute le code une fois que la configuration du nœud est correctement terminée.

À partir de la AWS ParallelCluster version 3.4.0, le code peut être exécuté après la mise à jour du nœud principal, si vous définissez les paramètres de [OnNodeUpdated](#) configuration [HeadNodeCustomActions//](#).

Dans la plupart des cas, ce code est stocké dans Amazon Simple Storage Service (Amazon S3) et accessible via une connexion HTTPS. Le code est exécuté `root` et peut être exécuté dans n'importe quel langage de script pris en charge par le système d'exploitation du cluster. Le code est souvent en Bash ou en Python.

Note

À partir de AWS ParallelCluster la version 3.7.0, le [ImdsSupport](#) paramètre [Imdscluster/](#) par défaut est `v2.0`.

Lorsque vous créez un nouveau cluster pour effectuer une mise à niveau vers la version 3.7.0 et les versions ultérieures, mettez à jour vos scripts d'action bootstrap personnalisés pour qu'ils soient compatibles avec IMDSv2 ou définissez [Imds/ImdsSupport](#) to `v1.0` dans votre fichier de configuration de cluster.

Warning

Vous êtes responsable de la configuration des scripts et arguments personnalisés, comme décrit dans le [modèle de responsabilité partagée](#). Vérifiez que vos scripts et arguments bootstrap personnalisés proviennent de sources fiables offrant un accès complet aux nœuds de votre cluster.

⚠ Warning

AWS ParallelCluster ne prend pas en charge l'utilisation de variables internes fournies par le biais du `/etc/parallelcluster/cfnconfig` fichier. Ce fichier sera peut-être supprimé dans le cadre d'une future version.

`OnNodeStart` les actions sont appelées avant le lancement de toute action d'amorçage du déploiement d'un nœud, telle que la configuration de NAT, d'Amazon Elastic Block Store (Amazon EBS) ou du planificateur. `OnNodeStart` les actions bootstrap peuvent inclure la modification du stockage, l'ajout d'utilisateurs supplémentaires et l'ajout de packages.

ℹ Note

Si vous [DirectoryService](#) configurez un `OnNodeStart` script [HeadNode/CustomActions](#)/pour votre cluster, AWS ParallelCluster configurez `DirectoryService` et redémarrez `lessd`, avant qu'il n'exécute le `OnNodeStart` script.

`OnNodeConfigured` les actions sont appelées une fois les processus d'amorçage du nœud terminés. `OnNodeConfigured` les actions correspondent aux dernières actions effectuées avant qu'une instance ne soit considérée comme entièrement configurée et complète. Certaines `OnNodeConfigured` actions incluent la modification des paramètres du planificateur, la modification du stockage et la modification des packages. Vous pouvez transmettre des arguments aux scripts en les spécifiant lors de la configuration.

`OnNodeUpdated` les actions sont appelées une fois que la mise à jour du nœud principal est terminée et que le planificateur et le stockage partagé sont alignés sur les dernières modifications de configuration du cluster.

Lorsque `OnNodeStart` les actions `OnNodeConfigured` personnalisées réussissent, le succès est indiqué par le code de sortie zéro (0). Tout autre code de sortie indique que le bootstrap de l'instance a échoué.

Lorsque les actions `OnNodeUpdated` personnalisées réussissent, leur réussite est signalée par le code de sortie zéro (0). Tout autre code de sortie indique que la mise à jour a échoué.

Note

Si vous configurez [OnNodeUpdated](#), vous devez restaurer manuellement les `OnNodeUpdated` actions à leur état précédent en cas d'échec de mise à jour. Si une action `OnNodeUpdated` personnalisée échoue, la mise à jour revient à l'état précédent. Toutefois, l'action `OnNodeUpdated` n'est exécutée qu'au moment de la mise à jour et non au moment de la restauration de la pile.

Vous pouvez spécifier différents scripts pour le nœud principal et pour chaque file d'attente, dans les sections de [CustomActions](#) configuration [HeadNodeSchedulingSlurmQueues/CustomActions](#) et//. [OnNodeUpdated](#) ne peut être configuré que dans la `HeadNode` section.

Note

Avant AWS ParallelCluster la version 3.0, il n'était pas possible de spécifier des scripts différents pour les nœuds de tête et de calcul. Veuillez consulter [Passer de la version AWS ParallelCluster 2.x à la version 3.x](#).

Rubriques

- [Configuration](#)
- [Arguments](#)
- [Exemple de cluster avec des actions bootstrap personnalisées](#)
- [Exemple de mise à jour d'un script bootstrap personnalisé pour IMDSv2](#)
- [Exemple de mise à jour d'une configuration pour IMDSv1](#)

Configuration

Les paramètres de configuration suivants sont utilisés pour définir les [HeadNodeOnNodeConfigured](#) actions `OnNodeStart` et `OnNodeConfigured` les arguments [Scheduling/CustomActions/OnNodeStart](#) & et//. [CustomActionsOnNodeUpdated](#)

```
HeadNode:
  [...]
  CustomActions:
```



```
OnNodeStart:
  # Script URL. This is run before any of the bootstrap scripts are run
  Script: s3://bucket-name/on-node-start.sh
  Args:
    - arg1
OnNodeConfigured:
  # Script URL. This is run after all the bootstrap scripts are run
  Script: s3://bucket-name/on-node-configured.sh
  Args:
    - arg1
OnNodeUpdated:
  # Script URL. This is run after the head node update is completed.
  Script: s3://bucket-name/on-node-updated.sh
  Args:
    - arg1
# Bucket permissions
Iam:
  S3Access:
    - BucketName: bucket_name
      EnableWriteAccess: false
Scheduling:
  Scheduler: slurm
  [...]
SlurmQueues:
  - Name: queue1
  [...]
CustomActions:
  OnNodeStart:
    Script: s3://bucket-name/on-node-start.sh
    Args:
      - arg1
  OnNodeConfigured:
    Script: s3://bucket-name/on-node-configured.sh
    Args:
      - arg1
Iam:
  S3Access:
    - BucketName: bucket_name
      EnableWriteAccess: false
```

En utilisant le Sequence paramètre (ajouté dans la AWS ParallelCluster version 3.6.0) :

```
HeadNode:
```

```
[...]  
CustomActions:  
  OnNodeStart:  
    # Script URLs. The scripts are run in the same order as listed in the  
configuration, before any of the bootstrap scripts are run.  
    Sequence:  
      - Script: s3://bucket-name/on-node-start1.sh  
        Args:  
          - arg1  
      - Script: s3://bucket-name/on-node-start2.sh  
        Args:  
          - arg1  
    [...]  
  OnNodeConfigured:  
    # Script URLs. The scripts are run in the same order as listed in the  
configuration, after all the bootstrap scripts are run.  
    Sequence:  
      - Script: s3://bucket-name/on-node-configured1.sh  
        Args:  
          - arg1  
      - Script: s3://bucket-name/on-node-configured2.sh  
        Args:  
          - arg1  
    [...]  
  OnNodeUpdated:  
    # Script URLs. The scripts are run in the same order as listed in the  
configuration, after the head node update is completed.  
    Sequence:  
      - Script: s3://bucket-name/on-node-updated1.sh  
        Args:  
          - arg1  
      - Script: s3://bucket-name/on-node-updated2.sh  
        Args:  
          - arg1  
    [...]  
# Bucket permissions  
Iam:  
  S3Access:  
    - BucketName: bucket_name  
      EnableWriteAccess: false  
Scheduling:  
  Scheduler: slurm  
  [...]  
SlurmQueues:
```

```

- Name: queue1
  [...]
  CustomActions:
    OnNodeStart:
      # Script URLs. The scripts are run in the same order as listed in the
      configuration, before any of the bootstrap scripts are run
      Sequence:
        - Script: s3://bucket-name/on-node-start1.sh
          Args:
            - arg1
        - Script: s3://bucket-name/on-node-start2.sh
          Args:
            - arg1
        [...]
    OnNodeConfigured:
      # Script URLs. The scripts are run in the same order as listed in the
      configuration, after all the bootstrap scripts are run
      Sequence:
        - Script: s3://bucket-name/on-node-configured1.sh
          Args:
            - arg1
        - Script: s3://bucket-name/on-node-configured2.sh
          Args:
            - arg1
        [...]
  Iam:
    S3Access:
      - BucketName: bucket_name
        EnableWriteAccess: false

```

Le Sequence paramètre est ajouté à partir de la AWS ParallelCluster version 3.6.0. Lorsque vous le spécifiez Sequence, vous pouvez répertorier plusieurs scripts pour une action personnalisée. AWS ParallelCluster continue de prendre en charge la configuration d'une action personnalisée avec un seul script, sans inclure Sequence.

AWS ParallelCluster ne permet pas d'inclure à la fois un seul script et Sequence pour la même action personnalisée. Par exemple, AWS ParallelCluster échoue si vous spécifiez la configuration suivante.

```

[...]
CustomActions:
  OnNodeStart:
    # Script URL. This is run before any of the bootstrap scripts are run

```

```
Script: s3://bucket-name/on-node-start.sh
  Args:
    - arg1
  # Script URLs. The scripts are run in the same order as listed in the
  # configuration, before any of the bootstrap scripts are run.
  Sequence:
    - Script: s3://bucket-name/on-node-start1.sh
      Args:
        - arg1
    - Script: s3://bucket-name/on-node-start2.sh
      Args:
        - arg1
[...]
```

Arguments

Note

Dans la AWS ParallelCluster version 2.x, les \$1 arguments étaient réservés, pour stocker l'URL du script personnalisé. Si vous souhaitez réutiliser les scripts bootstrap personnalisés créés pour AWS ParallelCluster 2.x avec AWS ParallelCluster 3.x, vous devez les adapter en tenant compte du décalage des arguments. Veuillez consulter [Passer de la version AWS ParallelCluster 2.x à la version 3.x](#).

Exemple de cluster avec des actions bootstrap personnalisées

Les étapes suivantes créent un script simple à exécuter une fois le nœud configuré, qui installe les wget packages R, curl et dans les nœuds du cluster.

1. Créez un script.

```
#!/bin/bash
echo "The script has $# arguments"
for arg in "$@"
do
  echo "arg: ${arg}"
done
yum -y install "${@:1}"
```

2. Téléchargez le script avec les autorisations appropriées sur Amazon S3. Si les autorisations de lecture publiques ne vous conviennent pas, utilisez [HeadNode](#) les sections de [SlurmQueues](#) configuration [Scheduling](#)/[S3Access](#)et/. [Iam](#) Pour plus d'informations, consultez [Utilisation des services dans Amazon S3](#).

```
$ aws s3 cp --acl public-read /path/to/myscript.sh s3://<bucket-name>/myscript.sh
```

Important

Si le script a été modifié sous Windows, les fins de ligne doivent passer de CRLF à LF avant que le script ne soit chargé sur Amazon S3.

3. Mettez à jour la AWS ParallelCluster configuration pour inclure la nouvelle `OnNodeConfigured` action.

```
CustomActions:
OnNodeConfigured:
  Script: https://<bucket-name>.s3.<region>.amazonaws.com/myscript.sh
  Args:
    - "R"
    - "curl"
    - "wget"
```

Si le bucket ne dispose pas d'une autorisation de lecture publique, s3 utilisez-le comme protocole URL.

```
CustomActions:
OnNodeConfigured:
  Script: s3://<bucket-name>/myscript.sh
  Args:
    - "R"
    - "curl"
    - "wget"
```

4. Lancement du cluster

```
$ pcluster create-cluster --cluster-name mycluster \
  --region <region> --cluster-configuration config-file.yaml
```

5. Vérifiez la sortie.

- Si vous avez ajouté des actions personnalisées à la HeadNode configuration, connectez-vous au nœud principal et vérifiez le `cfn-init.log` fichier qui s'y trouve `/var/log/cfn-init.log` en exécutant la commande suivante :

```
$ less /var/log/cfn-init.log
2021-09-03 10:43:54,588 [DEBUG] Command run
postinstall output: The script has 3 arguments
arg: R
arg: curl
arg: wget
Loaded plugins: dkms-build-requires, priorities, update-motd, upgrade-helper
Package R-3.4.1-1.52.amzn1.x86_64 already installed and latest version
Package curl-7.61.1-7.91.amzn1.x86_64 already installed and latest version
Package wget-1.18-4.29.amzn1.x86_64 already installed and latest version
Nothing to do
```

- Si vous avez ajouté des actions personnalisées au SlurmQueues paramètre, vérifiez l'`cloud-init.log` emplacement `/var/log/cloud-init.log` dans un nœud de calcul. CloudWatch À utiliser pour consulter ces journaux.

Vous pouvez consulter ces deux journaux dans la CloudWatch console Amazon. Pour plus d'informations, consultez [Intégration avec Amazon CloudWatch Logs avec Amazon Logs](#).

Exemple de mise à jour d'un script bootstrap personnalisé pour IMDSv2

Dans l'exemple suivant, nous mettons à jour un script d'action bootstrap personnalisé qui a été utilisé avec IMDSv1 pour être utilisé avec IMDSv2. Le script IMDSv1 récupère les métadonnées de l'ID AMI de l'instance Amazon EC2.

```
#!/bin/bash
AMI_ID=$(curl http://169.254.169.254/latest/meta-data/ami-id)
echo $AMI_ID >> /home/ami_id.txt
```

Ce qui suit montre le script d'action bootstrap personnalisé modifié pour être compatible avec IMDSv2.

```
#!/bin/bash
AMI_ID=$(TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
```

```
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/
latest/meta-data/ami-id)
echo $AMI_ID >> /home/ami_id.txt
```

Pour plus d'informations, consultez [Récupération des métadonnées d'instance](#) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Exemple de mise à jour d'une configuration pour IMDSv1

Voici un exemple de configuration de cluster qui prend en charge IMDSv1 lors de l'utilisation des AWS ParallelCluster versions 3.7.0 et antérieures.

```
Region: us-east-1
Imds:
  ImdsSupport: v1.0
Image:
  Os: alinux2
HeadNode:
  InstanceType: t2.micro
  Networking:
    SubnetId: subnet-abcdef01234567890
  Ssh
    KeyName: key-name
  CustomActions:
    OnNodeConfigured:
      Script: Script-path
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      CustomActions:
        OnNodeConfigured:
          Script: Script-path
      ComputeResources:
        - Name: t2micro
          Instances:
            - InstanceType: t2.micro
          MinCount: 11
  Networking:
    SubnetIds:
      - subnet-abcdef01234567890
```

Utilisation des services dans Amazon S3

Vous pouvez configurer AWS ParallelCluster l'accès à Amazon S3 via les [S3Access](#) paramètres [HeadNodeIam/Scheduling/S3Access](#) et [SlurmQueues- Name/Iam](#) de la AWS ParallelCluster configuration.

Exemples

L'exemple suivant configure l'accès en lecture seule à tous les objets contenus *firstbucket/read_only/* et l'accès en lecture/écriture à tous les objets contenus dans *secondbucket/read_and_write/*.

```
...
HeadNode:
  ...
  Iam:
    S3Access:
      - BucketName: firstbucket
        KeyName: read_only/*
        EnableWriteAccess: false
      - BucketName: secondbucket
        KeyName: read_and_write/*
        EnableWriteAccess: true
  ...
```

L'exemple suivant configure l'accès en lecture seule à tous les objets du dossier *read_only/* de n'importe quel compartiment (*) du compte.

```
...
HeadNode:
  ...
  Iam:
    S3Access:
      - BucketName: *
        KeyName: read_only/*
        EnableWriteAccess: false
  ...
```

Le dernier exemple configure l'accès en mode lecture seule à tous les compartiments et à tous les objets dans le compte.


```
...
HeadNode:
  ...
  Iam:
    S3Access:
      - BucketName: *
  ...
```

Utilisation de instances Spot

AWS ParallelCluster utilise des instances Spot si vous avez défini [SlurmQueues/CapacityType](#) ou [AwsBatchQueues/CapacityType](#) sur SPOT dans le fichier de configuration du cluster. Les instances ponctuelles sont plus économiques que les instances à la demande, mais elles peuvent être interrompues. Il peut être utile de tirer parti des avis d'interruption des instances Spot, qui fournissent un avertissement de deux minutes avant qu'Amazon EC2 ne doive arrêter ou résilier votre instance Spot. Pour plus d'informations, consultez la section [Interruptions des instances Spot](#) dans le guide de l'utilisateur Amazon EC2. Pour en savoir plus sur le [AwsBatchQueues](#) fonctionnement des instances Spot, consultez la section [Ressources informatiques](#) du guide de AWS Batch l'utilisateur.

Le planificateur AWS ParallelCluster configuré affecte les tâches aux ressources de calcul dans les files d'attente avec des instances ponctuelles de la même manière qu'il affecte les tâches aux ressources de calcul dans les files d'attente avec des instances à la demande.

Lorsque vous utilisez des instances Spot, un rôle `AWSServiceRoleForEC2Spot` lié au service doit exister dans votre compte. Pour créer ce rôle dans votre compte à l'aide de AWS CLI, exécutez la commande suivante :

```
$ aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Pour plus d'informations, consultez la section [Rôle lié au service pour les demandes d'instance Spot](#) dans le guide de l'utilisateur Amazon EC2.

Les sections suivantes décrivent trois scénarios dans lesquels les instances Spot peuvent être interrompues lors de leur utilisation [SlurmQueues](#).

Scénario 1 : Une instance Spot sans tâches en cours d'exécution est interrompue

Lorsque cette interruption se produit, AWS ParallelCluster essaie de remplacer l'instance si la file d'attente du planificateur contient des tâches en attente qui nécessitent des instances supplémentaires, ou si le nombre d'instances actives est inférieur au [SlurmQueues//ComputeResources.MinCount](#). Si AWS ParallelCluster vous ne parvenez pas à approvisionner de nouvelles instances, une demande de nouvelles instances est répétée périodiquement.

Scénario 2 : Une instance Spot exécutant des tâches à nœud unique est interrompue

La tâche échoue avec un code d'état de `NODE_FAIL`, et la tâche est mise en attente (sauf indication contraire `--no-requeue` lors de la soumission de la tâche). Si le nœud est un nœud statique, il est remplacé. Si le nœud est un nœud dynamique, le nœud est arrêté et réinitialisé. Pour plus d'informations `sbatch`, notamment sur le `--no-requeue` paramètre, consultez [sbatch](#) la Slurmdocumentation.

Scénario 3 : Une instance Spot exécutant des tâches à plusieurs nœuds est interrompue

La tâche échoue avec un code d'état de `NODE_FAIL`, et la tâche est mise en attente (sauf si cela `--no-requeue` a été spécifié lors de la soumission de la tâche). Si le nœud est un nœud statique, il est remplacé. Si le nœud est un nœud dynamique, le nœud est arrêté et réinitialisé. Les autres nœuds qui exécutaient les tâches terminées peuvent être alloués à d'autres tâches en attente ou réduits une fois le [ScaledownIdleTime](#) temps configuré [SlurmSettings](#)/dépassé.

Pour plus d'informations sur les instances Spot, consultez la section [Instances Spot](#) dans le guide de l'utilisateur Amazon EC2.

Planificateurs pris en charge par AWS ParallelCluster

Planificateurs pris en charge par AWS ParallelCluster

AWS ParallelCluster supports Slurm et AWS Batch planificateurs, définis à l'aide du [Scheduler](#) paramètre.

Rubriques

- [Slurm Workload Manager \(slurm\)](#)
- [AWS Batch \(awsbatch\)](#)

Slurm Workload Manager (**slurm**)

Taille et mise à jour de la capacité du cluster

La capacité du cluster est définie par le nombre de nœuds de calcul que le cluster peut dimensionner. Les nœuds de calcul sont soutenus par des instances Amazon EC2 définies dans les ressources de calcul de la AWS ParallelCluster configuration (`Scheduling/SlurmQueues/ComputeResources`), et sont organisés en files d'attente (`Scheduling/SlurmQueues`) qui mappent 1:1 aux partitions. Slurm

Au sein d'une ressource de calcul, il est possible de configurer le nombre minimum de nœuds de calcul (instances) qui doivent toujours continuer à fonctionner dans le cluster (`MinCount`), ainsi que le nombre maximum d'instances que la ressource de calcul peut atteindre (`MaxCount3`).

Au moment de la création du cluster, ou lors d'une mise à jour du cluster, AWS ParallelCluster lance autant d'instances Amazon EC2 que configuré `MinCount` pour chaque ressource de calcul (`Scheduling/SlurmQueues/ ComputeResources`) définie dans le cluster. Les instances lancées pour couvrir le nombre minimal de nœuds pour une ressource de calcul dans le cluster sont appelées nœuds statiques. Une fois démarrés, les nœuds statiques sont censés être persistants dans le cluster et le système ne les arrête pas, sauf si un événement ou une condition spécifique se produit. Ces événements incluent, par exemple, l'échec des tests de Slurm santé d'Amazon EC2 et le changement de l'état du Slurm nœud en DRAIN ou DOWN.

Les instances Amazon EC2, de l'ordre de 1 à `'MaxCount - MinCount'` (`MaxCount` moins `MinCount`), lancées à la demande pour faire face à l'augmentation de la charge du cluster, sont appelées nœuds dynamiques. Leur nature est éphémère, ils sont lancés pour exécuter des tâches en attente et sont interrompus une fois qu'ils restent inactifs pendant une période définie `Scheduling/SlurmSettings/ScaledownIdletime` dans la configuration du cluster (par défaut : 10 minutes).

Les nœuds statiques et les nœuds dynamiques sont conformes au schéma de dénomination suivant :

- Nœuds statiques `<Queue/Name>-st-<ComputeResource/Name>-<num>` où `<num> = 1..ComputeResource/MinCount`

- Nœuds dynamiques `<Queue/Name>-dy-<ComputeResource/Name>-<num>` où `<num> = 1..(ComputeResource/MaxCount - ComputeResource/MinCount)`

Par exemple, étant donné la AWS ParallelCluster configuration suivante :

```
Scheduling:
  Scheduler: Slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: c5xlarge
          Instances:
            - InstanceType: c5.xlarge
              MinCount: 100
              MaxCount: 150
```

Les nœuds suivants seront définis dans Slurm

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

Lorsqu'une ressource de calcul l'est `MinCount == MaxCount`, tous les nœuds de calcul correspondants seront statiques et toutes les instances seront lancées au moment de la création/de la mise à jour du cluster et maintenues opérationnelles. Par exemple :

```
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: c5xlarge
          Instances:
            - InstanceType: c5.xlarge
              MinCount: 100
              MaxCount: 100
```

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up        infinite   100    idle queue1-st-c5xlarge-[1-100]
```

Mise à jour des capacités du cluster

La mise à jour de la capacité du cluster inclut l'ajout ou la suppression de files d'attente, de ressources de calcul ou la modification MinCount/MaxCount d'une ressource de calcul. À partir de AWS ParallelCluster la version 3.9.0, la réduction de la taille d'une file d'attente nécessite que le parc de calcul soit arrêté ou [QueueUpdateStrategy](#) défini sur TERMINATE avant qu'une mise à jour du cluster n'ait lieu. Il n'est pas nécessaire d'arrêter le parc informatique ou de le [QueueUpdateStrategy](#) configurer sur TERMINATE lorsque :

- Ajouter de nouvelles files d'attente à la planification/ [SlurmQueues](#)
- Ajouter de nouvelles ressources de calcul Scheduling/SlurmQueues/[ComputeResources](#) à une file d'attente
- Augmenter la valeur [MaxCount](#) d'une ressource de calcul
- Augmentation MinCount d'une ressource de calcul et augmentation MaxCount de la même ressource de calcul d'au moins la même quantité

Considérations et restrictions

Cette section vise à décrire tous les facteurs, contraintes ou limitations importants à prendre en compte lors du redimensionnement de la capacité du cluster.

- Lors de la suppression d'une file d'attente de Scheduling/<https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling-v3.html#Scheduling-v3-SlurmQueues> SlurmQueues tous les nœuds de calcul portant un nom <Queue/Name>-*, à la fois statique et dynamique, sera supprimé de la Slurm configuration et les instances Amazon EC2 correspondantes seront résiliées.
- Lorsque vous supprimez une ressource Scheduling/SlurmQueues/<https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling->

[v3.html#Scheduling-v3-SlurmQueues-ComputeResources](#) ComputeResources

de calcul d'une file d'attente, tous les nœuds de calcul portant un nom `<Queue/Name> - * - <ComputeResource/Name> - *`, qu'ils soient statiques ou dynamiques, sont supprimés de la Slurm configuration et les instances Amazon EC2 correspondantes sont mises hors service.

Lorsque vous modifiez le `MinCount` paramètre d'une ressource de calcul, nous pouvons distinguer deux scénarios différents, s'il `MaxCount` est maintenu égal à `MinCount` (capacité statique uniquement) et s'il `MaxCount` est supérieur à `MinCount` (capacité statique et dynamique mixte).

Changements de capacité avec des nœuds statiques uniquement

- Si `MinCount == MaxCount`, lors de l'augmentation `MinCount` (et `MaxCount`), le cluster est configuré en étendant le nombre de nœuds statiques à la nouvelle valeur de `MinCount` `<Queue/Name>-st-<ComputeResource/Name>-<new_MinCount>` et que le système continue d'essayer de lancer des instances Amazon EC2 pour atteindre la nouvelle capacité statique requise.
- Si `MinCount == MaxCount`, lors de la diminution `MinCount` (et `MaxCount`) du montant `N`, le cluster est configuré en supprimant les `N` derniers nœuds statiques `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<old_MinCount>`] et le système met fin aux instances Amazon EC2 correspondantes.
- État initial `MinCount = MaxCount = 100`

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up        infinite   100   idle queue1-st-c5xlarge-[1-100]
```

- Mise à jour `-30` sur `MinCount` et `MaxCount`: `MinCount = MaxCount = 70`

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up        infinite   70    idle queue1-st-c5xlarge-[1-70]
```

Changements de capacité avec des nœuds mixtes

Si $\text{MinCount} < \text{MaxCount}$, lors d'une augmentation MinCount d'un montant N (en supposant que MaxCount cela restera inchangé), le cluster sera configuré en étendant le nombre de nœuds statiques à la nouvelle valeur de MinCount ($\text{old_MinCount} + N$) : `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>` et le système continuera d'essayer de lancer des instances Amazon EC2 pour atteindre la nouvelle capacité statique requise. De plus, pour respecter la MaxCount capacité de la ressource de calcul, la configuration du cluster est mise à jour en supprimant les N derniers nœuds dynamiques : `<Queue/Name>-dy-<ComputeResource/Name>-[<MaxCount - old_MinCount - N>...<MaxCount - old_MinCount>]` et le système mettra fin aux instances Amazon EC2 correspondantes.

- État initial : $\text{MinCount} = 100$; $\text{MaxCount} = 150$

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*   up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

- Mettre à jour +30 vers MinCount : $\text{MinCount} = 130$ ($\text{MaxCount} = 150$)

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*   up    infinite   20    idle~ queue1-dy-c5xlarge-[1-20]
queue1*   up    infinite  130    idle queue1-st-c5xlarge-[1-130]
```

Si $\text{MinCount} < \text{MaxCount}$, lors de l'augmentation MinCount et MaxCount de la même quantité N , le cluster est configuré en étendant le nombre de nœuds statiques à la nouvelle valeur de MinCount ($\text{old_MinCount} + N$) : `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>` et le système continuera d'essayer de lancer des instances Amazon EC2 pour atteindre la nouvelle capacité statique requise. De plus, aucune modification ne sera apportée au nombre de nœuds dynamiques pour honorer le nouveau

Valeur MaxCount .

- État initial : MinCount = 100; MaxCount = 150

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  100    idle  queue1-st-c5xlarge-[1-100]

```

- Mettre à jour +30 vers MinCount : MinCount = 130 (MaxCount = 180)

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   20    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  130    idle  queue1-st-c5xlarge-[1-130]

```

Si $\text{MinCount} < \text{MaxCount}$, lors de la diminution MinCount du montant N (en supposant qu'il MaxCount restera inchangé), le cluster sera configuré en supprimant les N derniers nœuds statiques (nœuds statiques) `<Queue/Name>-st-<ComputeResource/Name>-[<old_MinCount - N>...<old_MinCount>` et le système mettra fin aux instances Amazon EC2 correspondantes. De plus, pour respecter la MaxCount capacité de la ressource de calcul, la configuration du cluster est mise à jour en augmentant le nombre de nœuds dynamiques afin de combler le vide. `MaxCount - new_MinCount: <Queue/Name>-dy-<ComputeResource/Name>-[1..<MaxCount - new_MinCount>]` Dans ce cas, comme il s'agit de nœuds dynamiques, aucune nouvelle instance Amazon EC2 ne sera lancée à moins que le planificateur n'ait des tâches en attente sur les nouveaux nœuds.

- État initial : MinCount = 100; MaxCount = 150

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  100    idle  queue1-st-c5xlarge-[1-100]

```


- Mise à jour -30 le MinCount : MinCount = 70 (MaxCount = 120)

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   80    idle~ queue1-dy-c5xlarge-[1-80]
queue1*    up    infinite   70    idle  queue1-st-c5xlarge-[1-70]

```

Si $\text{MinCount} < \text{MaxCount}$, lors de la diminution MinCount et MaxCount de la même quantité N, le cluster sera configuré en supprimant les N derniers nœuds statiques `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<oldMinCount>`] et le système mettra fin aux instances Amazon EC2 correspondantes.

De plus, aucune modification ne sera apportée au nombre de nœuds dynamiques pour respecter la nouvelle MaxCount valeur.

- État initial : MinCount = 100; MaxCount = 150

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  100    idle  queue1-st-c5xlarge-[1-100]

```

- Mise à jour -30 le MinCount : MinCount = 70 (MaxCount = 120)

```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   80    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite   70    idle  queue1-st-c5xlarge-[1-70]

```

Si $\text{MinCount} < \text{MaxCount}$, lors de la diminution MaxCount du montant N (en supposant MinCount qu'il reste inchangé), le cluster est configuré en supprimant les N derniers nœuds dynamiques

<Queue/Name>-dy-<ComputeResource/Name>-<old_MaxCount - N...<oldMaxCount>] et le système arrête les instances Amazon EC2 correspondantes dans le cas où elles étaient en cours d'exécution. Aucun impact n'est attendu sur les nœuds statiques.

- État initial : MinCount = 100; MaxCount = 150

-

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

- Mise à jour -30 le MaxCount : MinCount = 100 (MaxCount = 120)

-

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   20    idle~ queue1-dy-c5xlarge-[1-20]
queue1*    up    infinite  100    idle queue1-st-c5xlarge-[1-100]
```

Impacts sur les emplois

Dans tous les cas où des nœuds sont supprimés et des instances Amazon EC2 résiliées, une tâche sbatch exécutée sur les nœuds supprimés sera remise en file d'attente, sauf si aucun autre nœud ne répond aux exigences de la tâche. Dans ce dernier cas, la tâche échouera avec le statut `NODE_FAIL` et disparaîtra de la file d'attente ; dans ce cas, elle devra être soumise à nouveau manuellement.

Si vous prévoyez d'effectuer une mise à jour de redimensionnement du cluster, vous pouvez empêcher les tâches de s'exécuter sur les nœuds qui seront supprimés lors de la mise à jour planifiée. Cela est possible en configurant les nœuds à supprimer lors de la maintenance. Sachez que le fait de configurer un nœud en maintenance n'aura aucune incidence sur les tâches qui sont déjà en cours d'exécution sur le nœud.

Supposons qu'avec la mise à jour prévue pour le redimensionnement du cluster, vous allez supprimer le `queue-st-computeresource-[9-10 [nœud]]`. Vous pouvez créer une Slurm réservation à l'aide de la commande suivante

```
sudo -i scontrol create reservation ReservationName=maint_for_update user=root
starttime=now duration=infinite flags=maint,ignore_jobs nodes=queueu-st-
computeresource-[9-10]
```

Cela créera une Slurm réservation nommée `maint_for_update` sur les nœuds `queueu-st-computeresource-[9-10]`. À partir du moment où la réservation est créée, aucune autre tâche ne peut être exécutée sur les nœuds `queueu-st-computeresource-[9-10]`. Sachez que la réservation n'empêchera pas l'attribution éventuelle de tâches sur les nœuds `queueu-st-computeresource-[9-10]`.

Après la mise à jour du redimensionnement du cluster, si la Slurm réservation a été définie uniquement sur les nœuds supprimés lors de la mise à jour du redimensionnement, la réservation de maintenance sera automatiquement supprimée. Si vous avez plutôt créé une Slurm réservation sur les nœuds qui sont toujours présents après la mise à jour du redimensionnement du cluster, nous souhaiterons peut-être supprimer la réservation de maintenance sur les nœuds une fois la mise à jour du redimensionnement effectuée, à l'aide de la commande suivante

```
sudo -i scontrol delete ReservationName=maint_for_update
```

[Pour plus de détails sur la Slurm réservation, consultez le document officiel de SchedMD ici.](#)

Processus de mise à jour du cluster en cas de modification de capacité

Lors d'un changement de configuration du planificateur, les étapes suivantes sont exécutées pendant le processus de mise à jour du cluster :

- Arrêtez AWS ParallelCluster `clustermgtd` (`supervisorctl stop clustermgtd`)
- Générer une configuration de Slurm partitions mise à jour depuis AWS ParallelCluster la configuration
- Redémarrage `slurmctld` (effectué via la recette du service Chef)
- Vérifier le `slurmctld` statut (`systemctl is-active --quiet slurmctld.service`)
- Recharger la configuration Slurm (`scontrol reconfigure`)
- Démarrage de `clustermgtd` (`supervisorctl start clustermgtd`)

Pour plus d'informations sur Slurm, consultez <https://slurm.schedmd.com>. Pour les téléchargements, consultez le [site https://github.com/SchedMD/slurm/tags](https://github.com/SchedMD/slurm/tags). Pour le code source, consultez <https://github.com/SchedMD/slurm>.

| AWS ParallelCluster version (s) | Version Slurm prise en charge |
|---------------------------------|-------------------------------|
| 3,9.2, 3,9.3, 3.10.0 | 23,11.7 |
| 3.9.0, 3.9.1 | 23,11.4 |
| 3.8.0 | 23,02,7 |
| 3.7.2 | 23,02,6 |
| 3.7.1 | 23,02,5 |
| 3.7.0 | 23,02,4 |
| 3.6.0, 3.6.1 | 23,02,2 |
| 3.5.0, 3.5.1 | 22,05,8 |
| 3.4.0, 3.4.1 | 22,05,7 |
| 3.3.0, 3.3.1 | 22,05,5 |
| 3.1.4, 3.1.5, 3.2.0, 3.2.1 | 21.08.8-2 |
| 3.1.2, 3.1.3 | 21,08,6 |
| 3.1.1 | 21,08,5 |
| 3.0.0 | 20,11,8 |

Rubriques

- [Configuration de plusieurs files d'attente](#)
- [Slurmguide pour le mode de file d'attente multiple](#)
- [Slurmmode protégé par cluster](#)
- [Slurmbasculement rapide d'une capacité insuffisante du cluster](#)
- [Slurmplanification basée sur la mémoire](#)
- [Allocation de plusieurs types d'instances avec Slurm](#)
- [Mise à l'échelle du cluster pour les nœuds dynamiques](#)

- [Slurmcomptabilité avec AWS ParallelCluster](#)
- [Slurmpersonnalisation de la configuration](#)
- [Slurmprolog et epilog](#)
- [Taille et mise à jour de la capacité du cluster](#)

Configuration de plusieurs files d'attente

Configuration de plusieurs files d'attente

Avec AWS ParallelCluster la version 3, vous pouvez configurer plusieurs files d'attente en définissant le [Scheduler](#) à `slurm` et en spécifiant plusieurs files d'attente [SlurmQueues](#) dans le fichier de configuration. Dans ce mode, différents types d'instances coexistent dans les nœuds de calcul spécifiés dans la [ComputeResources](#) section du fichier de configuration. [ComputeResources](#) avec différents types d'instances sont redimensionnés à la hausse ou à la baisse selon les besoins du [SlurmQueues](#).

Quotas de ressources de calcul et files d'attente du cluster

| Ressource | Quota |
|-----------------------------------|---|
| Slurm queues | 50 files d'attente par cluster |
| Compute resources | 50 ressources de calcul par file d'attente 50 ressources de calcul par cluster |

Nombre de nœuds

Chaque ressource de calcul [ComputeResources](#) d'une file d'attente doit avoir un [Name](#) caractère unique [InstanceType](#), [MinCount](#), et [MaxCount](#). [MinCount](#) et [MaxCount](#) ont des valeurs par défaut qui définissent la plage d'instances d'une ressource de calcul dans [ComputeResources](#) une file d'attente. Vous pouvez également spécifier vos propres valeurs pour [MinCount](#) et [MaxCount](#). Chaque ressource de calcul [ComputeResources](#) est composée de nœuds statiques numérotés de 1 à la valeur de [MinCount](#) et de nœuds dynamiques numérotés de la valeur de [MinCount](#) à la valeur de [MaxCount](#).

Exemple de configuration

Voici un exemple de section de [planification](#) pour un fichier de configuration de cluster. Dans cette configuration, deux files d'attente sont nommées `queue1` `queue2` et chacune d'entre elles possède un nom [ComputeResources](#) spécifié. [MaxCount](#)

```
Scheduling:
  Scheduler: slurm
  SlurmQueues:
  - Name: queue1
    ComputeResources:
    - InstanceType: c5.xlarge
      MaxCount: 5
      Name: c5xlarge
    - InstanceType: c4.xlarge
      MaxCount: 5
      Name: c4xlarge
  - Name: queue2
    ComputeResources:
    - InstanceType: c5.xlarge
      MaxCount: 5
      Name: c5xlarge
```

Noms d'hôtes

Les instances lancées dans le parc informatique sont attribuées dynamiquement. Les noms d'hôtes sont générés pour chaque nœud. Par défaut AWS ParallelCluster, le format du nom d'hôte sera le suivant :

```
$HOSTNAME=$QUEUE-$STATDYN-$COMPUTE_RESOURCE-$NODENUM
```

- `$QUEUE` est le nom de la file d'attente. Par exemple, si la [SlurmQueues](#) section comporte une entrée [Name](#) définie sur « `queue-name` », alors « `$QUEUE` » est « `queue-name` ».
- `$STATDYN` est destiné aux nœuds statiques ou `dy` aux nœuds dynamiques.
- `$COMPUTE_RESOURCE` est le [Name](#) de la ressource de [ComputeResources](#) calcul correspondant à ce nœud.
- `$NODENUM` est le numéro du nœud. `$NODENUM` est compris entre un (1) et la valeur de [MinCount](#) pour les nœuds statiques et entre un (1) et [MaxCount](#) - [MinCount](#) pour les nœuds dynamiques.

Dans l'exemple de fichier de configuration ci-dessus, un nœud donné `queue1` et une ressource de calcul `c5xlarge` ont un nom d'hôte `:queue1-dy-c5xlarge-1`.

Les noms d'hôte et les noms de domaine complets (FQDN) sont créés à l'aide des zones hébergées Amazon Route 53. Le FQDN est `$HOSTNAME.$CLUSTERNAME.pcluster`, où `$CLUSTERNAME` est le nom du cluster.

Notez que le même format sera également utilisé pour les noms de Slurm nœuds.

Les utilisateurs peuvent choisir d'utiliser le nom d'hôte Amazon EC2 par défaut de l'instance alimentant le nœud de calcul au lieu du format de nom d'hôte par défaut utilisé par AWS ParallelCluster. Cela peut être fait en réglant le [UseEc2Hostnames](#) paramètre sur `true`. Toutefois, les noms de Slurm nœuds continueront d'utiliser le AWS ParallelCluster format par défaut.

Slurmguide pour le mode de file d'attente multiple

Ici, vous pouvez apprendre comment Slurm gérer AWS ParallelCluster les nœuds de file d'attente (partition) et comment surveiller l'état de la file d'attente et des nœuds.

Présentation

L'architecture de mise à l'échelle est basée sur le [guide Slurm de planification dans le cloud](#) et sur le plugin d'économie d'énergie. Pour plus d'informations sur le plug-in d'économie d'énergie, consultez le [Guide d'économie d'Slurménergie](#). Dans l'architecture, les ressources susceptibles d'être mises à disposition pour un cluster sont généralement prédéfinies dans la Slurm configuration sous forme de nœuds cloud.

Cycle de vie des nœuds cloud

Tout au long de leur cycle de vie, les nœuds cloud entrent dans plusieurs des états suivants `POWER_SAVING`, `POWER_UP` voire dans tous les cas `:`, `ALLOCATED` (`alloc`), `()` et `POWER_DOWN` (`pow_dn`). `pow_up` Dans certains cas, un nœud de cloud peut entrer dans `OFFLINE` cet état. La liste suivante détaille plusieurs aspects de ces états dans le cycle de vie des nœuds de cloud.

- Un nœud dans un **POWER_SAVING** état apparaît avec un `~` suffixe (par exemple `idle~`) dans `sinfo`. Dans cet état, aucune instance EC2 ne sauvegarde le nœud. Cependant, Slurm vous pouvez toujours allouer des tâches au nœud.
- Un nœud en transition vers un **POWER_UP** état apparaît avec un `#` suffixe (par exemple `idle#`) dans `sinfo`. Un nœud passe automatiquement à un `POWER_UP` état lorsqu'il Slurm alloue une tâche à un nœud dans un `POWER_SAVING` état.

Vous pouvez également faire passer les nœuds à l'`POWER_UP` état manuellement en tant qu'utilisateur su root à l'aide de la commande suivante :

```
$ scontrol update nodename=nodename state=power_up
```

À ce stade, le `ResumeProgram` est invoqué, les instances EC2 sont lancées et configurées, et le nœud passe à l'`POWER_UP` état.

- Un nœud actuellement disponible apparaît sans suffixe (par exemple `idle`) dans `sinfo`. Une fois le nœud configuré et rejoint le cluster, il devient disponible pour exécuter des tâches. À ce stade, le nœud est correctement configuré et prêt à être utilisé.

En règle générale, nous recommandons que le nombre d'instances Amazon EC2 soit identique au nombre de nœuds disponibles. Dans la plupart des cas, les nœuds statiques sont disponibles après la création du cluster.

- Un nœud en transition vers un **POWER_DOWN** état apparaît avec un `%` suffixe (par exemple `idle%`) dans `sinfo`. Les nœuds dynamiques entrent automatiquement dans l'`POWER_DOWN` état suivant [ScaledownIdleTime](#). En revanche, dans la plupart des cas, les nœuds statiques ne sont pas hors tension. Cependant, vous pouvez placer les nœuds dans l'`POWER_DOWN` état manuellement en tant qu'utilisateur su root à l'aide de la commande suivante :

```
$ scontrol update nodename=nodename state=down reason="manual draining"
```

Dans cet état, les instances associées à un nœud sont résiliées, et le nœud est remis à son `POWER_SAVING` état et peut être utilisé ultérieurement [ScaledownIdleTime](#).

Le [ScaledownIdleTime](#) paramètre est enregistré dans le `SuspendTimeout` paramètre Slurm de configuration.

- Un nœud hors ligne apparaît avec un `*` suffixe (par exemple `down*`) dans `sinfo`. Un nœud est déconnecté si le Slurm contrôleur ne parvient pas à le contacter ou si les nœuds statiques sont désactivés et que les instances de support sont interrompues.

Examinez les états des nœuds illustrés dans l'`sinfo` exemple suivant.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa        up    infinite     4  idle~ efa-dy-efacompute1-[1-4]
```



```

efa          up    infinite    1    idle efa-st-efacompute1-1
gpu          up    infinite    1    idle% gpu-dy-gpucompute1-1
gpu          up    infinite    9    idle~ gpu-dy-gpucompute1-[2-10]
ondemand    up    infinite    2    mix#  ondemand-dy-ondemandcompute1-[1-2]
ondemand    up    infinite    18   idle~  ondemand-dy-ondemandcompute1-
[3-10],ondemand-dy-ondemandcompute2-[1-10]
spot*       up    infinite    13   idle~  spot-dy-spotcompute1-[1-10],spot-dy-
spotcompute2-[1-3]
spot*       up    infinite    2    idle  spot-st-spotcompute2-[1-2]

```

Les `efa-st-efacompute1-1` nœuds `spot-st-spotcompute2-[1-2]` et ont déjà des instances de sauvegarde configurées et peuvent être utilisées. Les `ondemand-dy-ondemandcompute1-[1-2]` nœuds sont en bon `POWER_UP` état et devraient être disponibles dans quelques minutes. Le `gpu-dy-gpucompute1-1` nœud est dans l'`POWER_DOWN` état, et il passe à `POWER_SAVING` l'état par la suite [ScaledownIdletime](#) (10 minutes par défaut).

Tous les autres nœuds sont dans un `POWER_SAVING` état sans qu'aucune instance EC2 ne les soutienne.

Utilisation d'un nœud disponible

Un nœud disponible est soutenu par une instance Amazon EC2. Par défaut, le nom du nœud peut être utilisé pour accéder directement à l'instance en SSH (par exemple `ssh efa-st-efacompute1-1`). L'adresse IP privée de l'instance peut être récupérée à l'aide de la commande :

```
$ scontrol show nodes nodename
```

Vérifiez l'adresse IP dans le `NodeAddr` champ renvoyé.

Pour les nœuds qui ne sont pas disponibles, le `NodeAddr` champ ne doit pas pointer vers une instance Amazon EC2 en cours d'exécution. Il doit plutôt être identique au nom du nœud.

État des offres d'emploi et soumission

Dans la plupart des cas, les tâches soumises sont immédiatement allouées aux nœuds du système ou mises en attente si tous les nœuds sont alloués.

Si les nœuds alloués à une tâche incluent des nœuds dans un `POWER_SAVING` état, la tâche commence par un `CF CONFIGURING` état ou. À ce stade, la tâche attend que les nœuds de l'`POWER_SAVING` état passent à l'`POWER_UP` état et soient disponibles.

Une fois que tous les nœuds alloués à une tâche sont disponibles, la tâche passe à l'état RUNNING (R).

Par défaut, toutes les tâches sont soumises à la file d'attente par défaut (appelée partition inSlurm). Cela est indiqué par un * suffixe après le nom de la file d'attente. Vous pouvez sélectionner une file d'attente à l'aide de l'option de soumission des -p tâches.

Tous les nœuds sont configurés avec les fonctionnalités suivantes, qui peuvent être utilisées dans les commandes de soumission de tâches :

- Un type d'instance (par exemple `c5.xlarge`)
- Type de nœud (il s'agit de l'un `dynamic` ou `static` de l'autre)

Vous pouvez voir les fonctionnalités d'un nœud en particulier à l'aide de la commande :

```
$ scontrol show nodes nodename
```

Au retour, consultez la `AvailableFeatures` liste.

Tenez compte de l'état initial du cluster, que vous pouvez consulter en exécutant la `sinfo` commande.

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
efa        up    infinite   4    idle~ efa-dy-efacompute1-[1-4]
efa        up    infinite   1    idle  efa-st-efacompute1-1
gpu        up    infinite  10    idle~ gpu-dy-gpucompute1-[1-10]
ondemand   up    infinite  20    idle~ ondemand-dy-ondemandcompute1-
[1-10],ondemand-dy-ondemandcompute2-[1-10]
spot*      up    infinite  13    idle~ spot-dy-spotcompute1-[1-10],spot-dy-
spotcompute2-[1-3]
spot*      up    infinite   2    idle  spot-st-spotcompute2-[1-2]
```

Notez qu'il s'agit de la file d'attente par défaut. Il est indiqué par le * suffixe.

Soumettez une tâche à un nœud statique de la file d'attente par défaut (spot).

```
$ sbatch --wrap "sleep 300" -N 1 -C static
```

Soumettez une tâche à un nœud dynamique de la EFA file d'attente.

```
$ sbatch --wrap "sleep 300" -p efa -C dynamic
```

Soumettez une tâche à huit (8) c5.2xlarge nœuds et à deux (2) t2.xlarge nœuds de la ondemand file d'attente.

```
$ sbatch --wrap "sleep 300" -p ondemand -N 10 -C "[c5.2xlarge*8&t2.xlarge*2]"
```

Soumettez une tâche à un nœud GPU de la gpu file d'attente.

```
$ sbatch --wrap "sleep 300" -p gpu -G 1
```

Examinez l'état des tâches à l'aide de la squeue commande.

```
$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
12 ondemand wrap ubuntu CF 0:36 10 ondemand-dy-ondemandcompute1-
[1-8],ondemand-dy-ondemandcompute2-[1-2]
13 gpu wrap ubuntu CF 0:05 1 gpu-dy-gpucompute1-1
7 spot wrap ubuntu R 2:48 1 spot-st-spotcompute2-1
8 efa wrap ubuntu R 0:39 1 efa-dy-efacompute1-1
```

Les tâches 7 et 8 (dans les efa files d'attente spot et) sont déjà en cours d'exécution (R). Les jobs 12 et 13 sont toujours en cours de configuration (CF), attendant probablement que les instances soient disponibles.

```
# Nodes states corresponds to state of running jobs
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
efa up infinite 3 idle~ efa-dy-efacompute1-[2-4]
efa up infinite 1 mix efa-dy-efacompute1-1
efa up infinite 1 idle efa-st-efacompute1-1
gpu up infinite 1 mix~ gpu-dy-gpucompute1-1
gpu up infinite 9 idle~ gpu-dy-gpucompute1-[2-10]
ondemand up infinite 10 mix# ondemand-dy-ondemandcompute1-[1-8],ondemand-
dy-ondemandcompute2-[1-2]
ondemand up infinite 10 idle~ ondemand-dy-ondemandcompute1-[9-10],ondemand-
dy-ondemandcompute2-[3-10]
spot* up infinite 13 idle~ spot-dy-spotcompute1-[1-10],spot-dy-
spotcompute2-[1-3]
spot* up infinite 1 mix spot-st-spotcompute2-1
```

```
spot*      up    infinite    1    idle spot-st-spotcompute2-2
```

État et fonctionnalités du nœud

Dans la plupart des cas, les états des nœuds sont entièrement gérés AWS ParallelCluster conformément aux processus spécifiques du cycle de vie des nœuds de cloud décrits plus haut dans cette rubrique.

Toutefois, remplace ou arrête AWS ParallelCluster également les nœuds défectueux dans les DRAINED états DOWN et les nœuds dont les instances de sauvegarde ne fonctionnent pas correctement. Pour plus d'informations, consultez [clustermgtd](#).

États de partition

AWS ParallelCluster prend en charge les états de partition suivants. Une Slurm partition est une file d'attente dans AWS ParallelCluster.

- UP: indique que la partition est dans un état actif. Il s'agit de l'état par défaut d'une partition. Dans cet état, tous les nœuds de la partition sont actifs et peuvent être utilisés.
- INACTIVE: indique que la partition est inactive. Dans cet état, toutes les instances qui soutiennent les nœuds d'une partition inactive sont arrêtées. Les nouvelles instances ne sont pas lancées pour les nœuds d'une partition inactive.

cluster update-compute-fleet

- Arrêt du parc informatique : lorsque la commande suivante est exécutée, toutes les partitions passent à l'INACTIVE état, et AWS ParallelCluster les processus les maintiennent dans INACTIVE cet état.

```
$ pcluster update-compute-fleet --cluster-name testSlurm \  
  --region eu-west-1 --status STOP_REQUESTED
```

- Démarrage du parc informatique : lorsque la commande suivante est exécutée, toutes les partitions passent initialement à l'UP état. Cependant, AWS ParallelCluster les processus ne maintiennent pas la partition dans un UP état. Vous devez modifier l'état des partitions manuellement. Tous les nœuds statiques sont disponibles au bout de quelques minutes. Notez que le fait de définir une partition sur UP n'augmente aucune capacité dynamique.

```
$ pcluster update-compute-fleet --cluster-name testSlurm \  
  --region eu-west-1 --status STOP_REQUESTED
```

```
--region eu-west-1 --status START_REQUESTED
```

Lorsqu'il `update-compute-fleet` est exécuté, vous pouvez vérifier l'état du cluster en exécutant la `pcluster describe-compute-fleet` commande et en vérifiant le `Status`. La liste suivante répertorie les états possibles :

- `STOP_REQUESTED`: La demande d'arrêt du parc informatique est envoyée au cluster.
- `STOPPING`: Le `pcluster` processus arrête actuellement le parc informatique.
- `STOPPED`: le `pcluster` processus a terminé le processus d'arrêt, toutes les partitions sont en `INACTIVE` état et toutes les instances de calcul sont terminées.
- `START_REQUESTED`: La demande de démarrage de la flotte de calcul est envoyée au cluster.
- `STARTING`: le `pcluster` processus est en train de démarrer le cluster.
- `RUNNING`: le `pcluster` processus de démarrage est terminé, toutes les partitions sont en `UP` état et les nœuds statiques sont disponibles au bout de quelques minutes.
- `PROTECTED`: Cet état indique que certaines partitions présentent des échecs d'amorçage constants. Les partitions concernées sont inactives. Examinez le problème, puis lancez-vous `update-compute-fleet` pour réactiver le parc.

Contrôle manuel des files d'attente

Dans certains cas, vous souhaitez peut-être contrôler manuellement les nœuds ou la file d'attente (connue sous le nom de partition dans Slurm) d'un cluster. Vous pouvez gérer les nœuds d'un cluster à l'aide des procédures courantes suivantes à l'aide de la `scontrol` commande.

- Allumez les nœuds dynamiques en `POWER_SAVING` état

Exécutez la commande en tant qu'utilisateur su root :

```
$ scontrol update nodename=nodename state=power_up
```

Vous pouvez également soumettre une `sleep 1` tâche fictive demandant un certain nombre de nœuds, puis vous en servir Slurm pour alimenter le nombre de nœuds requis.

- Éteignez les nœuds dynamiques avant [ScaledownIdletime](#)

Nous vous recommandons de définir les nœuds dynamiques en `DOWN` tant qu'utilisateur su root à l'aide de la commande suivante :

```
$ scontrol update nodename=nodename state=down reason="manually draining"
```

AWS ParallelCluster arrête et réinitialise automatiquement les nœuds dynamiques tombés en panne.

En général, il est déconseillé de définir les nœuds sur POWER_DOWN directement à l'aide de la `scontrol update nodename=nodename state=power_down` commande. Cela est dû au fait que le processus de mise hors tension est AWS ParallelCluster automatiquement géré.

- Désactiver une file d'attente (partition) ou arrêter tous les nœuds statiques d'une partition spécifique

Définissez une file d'attente spécifique en INACTIVE tant qu'utilisateur su root à l'aide de la commande :

```
$ scontrol update partition=queuename state=inactive
```

Cela met fin à toutes les instances qui soutiennent les nœuds de la partition.

- Activer une file d'attente (partition)

Définissez une file d'attente spécifique pour UP un utilisateur su root à l'aide de la commande :

```
$ scontrol update partition=queuename state=up
```

Comportement de dimensionnement et ajustements

Voici un exemple du flux de travail de dimensionnement normal :

- Le planificateur reçoit une tâche qui nécessite deux nœuds.
- Le planificateur fait passer deux nœuds à un POWER_UP état et appelle ResumeProgram avec les noms des nœuds (par exemple `queue1-dy-spotcompute1-[1-2]`).
- ResumeProgram lance deux instances Amazon EC2 et attribue les adresses IP privées et les noms d'hôte de `queue1-dy-spotcompute1-[1-2]`, en attendant ResumeTimeout (la période par défaut est de 30 minutes) avant de réinitialiser les nœuds.
- Les instances sont configurées et rejoignent le cluster. Une tâche commence à s'exécuter sur des instances.

- La tâche se termine et s'arrête.
- Une fois la configuration `SuspendTime` expirée (qui est définie sur [ScaledownIdleTime](#)), le planificateur définit l'état des instances. `POWER_SAVING` Le planificateur passe ensuite `queue1-dy-spotcompute1-[1-2]` à l'`POWER_DOWN` état et appelle `SuspendProgram` avec les noms des nœuds.
- `SuspendProgram` est appelé pour deux nœuds. Les nœuds restent dans `POWER_DOWN` cet état, par exemple en restant `idle%` pendant un `SuspendTimeout` (la période par défaut est de 120 secondes (2 minutes)). Après avoir `clustermgtd` détecté que les nœuds sont hors tension, il met fin aux instances de sauvegarde. Ensuite, il passe `queue1-dy-spotcompute1-[1-2]` à l'état inactif et réinitialise l'adresse IP privée et le nom d'hôte afin qu'il soit prêt à fonctionner pour de futures tâches.

Si les choses tournent mal et qu'une instance pour un nœud particulier ne peut pas être lancée pour une raison quelconque, voici ce qui se passe :

- Le planificateur reçoit une tâche qui nécessite deux nœuds.
- Le planificateur fait passer deux nœuds éclatants dans le cloud à l'`POWER_UP` état et appelle `ResumeProgram` avec les noms de nœuds (par exemple). `queue1-dy-spotcompute1-[1-2]`
- `ResumeProgram` lance une (1) seule instance Amazon EC2 et la configure `queue1-dy-spotcompute1-1`, avec une (1) instance `queue1-dy-spotcompute1-2`, en échouant au lancement.
- `queue1-dy-spotcompute1-1` n'est pas concerné et se connecte après avoir atteint l'`POWER_UP` état.
- `queue1-dy-spotcompute1-2` passe à l'`POWER_DOWN` état, et la tâche est automatiquement mise en attente car elle Slurm détecte une défaillance du nœud.
- `queue1-dy-spotcompute1-2` devient disponible après `SuspendTimeout` (la valeur par défaut est de 120 secondes (2 minutes)). Entre-temps, la tâche est mise en attente et peut commencer à s'exécuter sur un autre nœud.
- Le processus ci-dessus se répète jusqu'à ce que la tâche puisse s'exécuter sur un nœud disponible sans qu'une défaillance se produise.

Deux paramètres de chronométrage peuvent être ajustés si nécessaire :

- **`ResumeTimeout`** (la valeur par défaut est de 30 minutes) : `ResumeTimeout` contrôle le temps d'attente de Slurm avant de faire passer le nœud à l'état inactif.

- Il peut être utile d'étendre `ResumeTimeout` si votre processus de pré-installation ou de post-installation prend presque autant de temps.
- `ResumeTimeout` est également le temps d'AWS ParallelCluster attente maximal avant de remplacer ou de réinitialiser un nœud en cas de problème. Les nœuds de calcul s'arrêtent automatiquement en cas d'erreur lors du lancement ou de la configuration. AWS ParallelCluster les processus remplacent un nœud lors de la détection d'une instance interrompue.
- **`SuspendTimeout`**(la valeur par défaut est de 120 secondes (2 minutes)) : `SuspendTimeout` contrôle la rapidité avec laquelle les nœuds sont réinsérés dans le système et sont prêts à être réutilisés.
 - Une valeur plus courte `SuspendTimeout` signifie que les nœuds sont réinitialisés plus rapidement et Slurm peuvent essayer de lancer des instances plus fréquemment.
 - Une valeur plus longue `SuspendTimeout` signifie que les nœuds défectueux sont réinitialisés plus lentement. En attendant, Slurm essaie d'utiliser d'autres nœuds. Si cela dure plus de quelques minutes, Slurm essaie de parcourir tous les nœuds du système. Un délai plus long `SuspendTimeout` peut être bénéfique pour les systèmes à grande échelle (plus de 1 000 nœuds) afin de réduire le stress Slurm lorsqu'ils tentent fréquemment de remettre en file d'attente des tâches défectueuses.
 - Notez que `SuspendTimeout` cela ne fait pas référence au temps d'AWS ParallelCluster attente pour mettre fin à une instance de sauvegarde pour un nœud. Les instances de sauvegarde pour `POWER_DOWN` les nœuds sont immédiatement résiliées. Le processus de terminaison est généralement terminé en quelques minutes. Cependant, pendant ce temps, le nœud reste dans cet `POWER_DOWN` état et n'est pas disponible pour l'utilisation du planificateur.

Logs pour l'architecture

La liste suivante contient les journaux clés. Le nom du flux de journal utilisé avec Amazon CloudWatch Logs a le format `{hostname}.{instance_id}.{logIdentifier}` suivant lequel *LogIdentifier* suit les noms des journaux.

- `ResumeProgram`: `/var/log/parallelcluster/slurm_resume.log` (`slurm_resume`)
- `SuspendProgram`: `/var/log/parallelcluster/slurm_suspend.log` (`slurm_suspend`)
- `clustermgtd`: `/var/log/parallelcluster/clustermgtd.log` (`clustermgtd`)
- `computemgtd`: `/var/log/parallelcluster/computemgtd.log` (`computemgtd`)
- `slurmctld`: `/var/log/slurmctld.log` (`slurmctld`)
- `slurmd`: `/var/log/slurmd.log` (`slurmd`)

Problèmes courants et procédure de débogage :

Nœuds qui n'ont pas réussi à démarrer, à démarrer ou à rejoindre le cluster

- Nœuds dynamiques :
 - Consultez le `ResumeProgram` journal pour voir s'il `ResumeProgram` a été appelé avec le nœud. Si ce n'est pas le cas, consultez le `slurmctld` journal pour déterminer si vous avez Slurm essayé `ResumeProgram` d'appeler le nœud. Notez que des autorisations incorrectes `ResumeProgram` peuvent entraîner son échec silencieux.
 - S'`ResumeProgram` est appelé, vérifiez si une instance a été lancée pour le nœud. Si l'instance n'a pas été lancée, un message d'erreur clair devrait s'afficher expliquant pourquoi l'instance n'a pas pu être lancée.
 - Si une instance a été lancée, il se peut qu'il y ait eu un problème pendant le processus de démarrage. Trouvez l'adresse IP privée et l'ID d'instance correspondants dans le `ResumeProgram` journal et consultez les journaux de démarrage correspondants pour l'instance spécifique dans CloudWatch Logs.
- Nœuds statiques :
 - Consultez le `clustermgtd` journal pour voir si des instances ont été lancées pour le nœud. Si les instances ne se sont pas lancées, il devrait y avoir des erreurs claires expliquant pourquoi elles n'ont pas pu être lancées.
 - Si une instance a été lancée, le processus d'amorçage présente un problème. Trouvez l'adresse IP privée et l'ID d'instance correspondants dans le `clustermgtd` journal et consultez les journaux de démarrage correspondants pour l'instance spécifique dans CloudWatch Logs.

Nœuds remplacés ou interrompus de manière inattendue, et défaillances de nœuds

- Nœuds remplacés/interrompus de façon inattendue :
 - Dans la plupart des cas, `clustermgtd` gère toutes les actions de maintenance des nœuds. Pour vérifier si un nœud a été `clustermgtd` remplacé ou résilié, consultez le `clustermgtd` journal.
 - En cas de `clustermgtd` remplacement ou de résiliation du nœud, un message doit apparaître indiquant le motif de l'action. Si la raison est liée au planificateur (par exemple, le nœud l'étaitDOWN), consultez le `slurmctld` journal pour plus de détails. Si la raison est liée à Amazon EC2, utilisez des outils tels qu'Amazon CloudWatch ou la console Amazon EC2, la CLI ou les kits SDK pour vérifier le statut ou les journaux de cette instance. Par exemple, vous pouvez

vérifier si l'instance a connu des événements planifiés ou si les vérifications de l'état de santé d'Amazon EC2 ont échoué.

- Si le nœud `clustermgtd` n'a pas été résilié, vérifiez s'il `computemgtd` a été résilié ou si EC2 a mis fin à l'instance pour récupérer une instance Spot.
- Défaillances des nœuds :
 - Dans la plupart des cas, les tâches sont automatiquement mises en file d'attente en cas de défaillance d'un nœud. Consultez le `slurmctld` journal pour savoir pourquoi une tâche ou un nœud a échoué et évaluez la situation à partir de là.

Défaillance lors du remplacement ou de l'arrêt d'instances, défaillance lors de la mise hors tension des nœuds

- En général, `clustermgtd` gère toutes les actions de fermeture d'instance attendues. Consultez le `clustermgtd` journal pour savoir pourquoi il n'a pas réussi à remplacer ou à mettre fin à un nœud.
- En cas d'échec de nœuds dynamiques [ScaledownIdletime](#), consultez le `SuspendProgram` journal pour voir si `slurmctld` les processus ont effectué des appels avec le nœud spécifique comme argument. Remarque `SuspendProgram` n'effectue en fait aucune action spécifique. Au contraire, il ne se connecte que lorsqu'il est appelé. Toutes les fermetures et `NodeAdd` réinitialisations d'instances sont terminées par `clustermgtd`. Slurm fait passer les nœuds vers les IDLE nœuds suivants `SuspendTimeout`.

Autres problèmes :

- AWS ParallelCluster ne prend pas de décisions relatives à l'attribution des tâches ou à la mise à l'échelle. Il essaie uniquement de lancer, d'arrêter et de maintenir les ressources conformément Slurm aux instructions.

Pour les problèmes liés à l'allocation des tâches, à l'allocation des nœuds et à la décision de dimensionnement, consultez le `slurmctld` journal pour détecter les erreurs.

Slurm mode protégé par cluster

Lorsqu'un cluster s'exécute avec le mode protégé activé, il AWS ParallelCluster surveille et suit les défaillances de démarrage des nœuds de calcul lors du lancement des nœuds de calcul. Il le fait pour détecter si ces défaillances se produisent en continu.

Si les éléments suivants sont détectés dans une file d'attente (partition), le cluster passe à l'état protégé :

1. Les défaillances consécutives du bootstrap du nœud de calcul se produisent continuellement en l'absence de lancement réussi du nœud de calcul.
2. Le nombre de défaillances atteint un seuil prédéfini.

Une fois que le cluster est devenu protégé, AWS ParallelCluster désactive les files d'attente présentant des défaillances égales ou supérieures au seuil prédéfini.

Slurmle mode protégé par cluster a été ajouté dans AWS ParallelCluster la version 3.0.0.

Vous pouvez utiliser le mode protégé pour réduire le temps et les ressources consacrés au cycle de défaillance du bootstrap des nœuds de calcul.

Paramètre du mode protégé

protected_failure_count

`protected_failure_count` indique le nombre de défaillances consécutives dans une file d'attente (partition) qui active le statut de protection du cluster.

La valeur par défaut `protected_failure_count` est 10 et le mode protégé est activé.

S'il `protected_failure_count` est supérieur à zéro, le mode protégé est activé.

S'`protected_failure_count` est inférieur ou égal à zéro, le mode protégé est désactivé.

Vous pouvez modifier la `protected_failure_count` valeur en ajoutant le paramètre dans le fichier de `clustermgtd` configuration situé `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` dans le `HeadNode`.

Vous pouvez mettre à jour ce paramètre à tout moment et vous n'avez pas besoin d'arrêter le parc informatique pour le faire. Si un lancement réussit dans une file d'attente avant que le nombre d'échecs n'atteigne `protected_failure_count`, le nombre d'échecs est remis à zéro.

Vérifier l'état du cluster dans l'état protégé

Lorsqu'un cluster est protégé, vous pouvez vérifier l'état du parc informatique et l'état des nœuds.

Calculer l'état du parc

L'état du parc informatique est celui PROTECTED d'un cluster fonctionnant en état protégé.

```
$ pcluster describe-compute-fleet --cluster-name <cluster-name> --region <region-id>
{
  "status": "PROTECTED",
  "lastStatusUpdateTime": "2022-04-22T00:31:24.000Z"
}
```

État du nœud

Pour savoir quelles files d'attente (partitions) présentent des défaillances de démarrage ayant activé le statut protégé, connectez-vous au cluster et exécutez la `sinfo` commande. Les partitions présentant des défaillances de bootstrap égales ou supérieures `protected_failure_count` sont dans INACTIVE cet état. Les partitions ne présentant pas d'échec du bootstrap égal ou supérieur `protected_failure_count` sont dans l'UPétat et fonctionnent comme prévu.

PROTECTEDle statut n'a aucun impact sur les tâches en cours d'exécution. Si des tâches sont exécutées sur une partition présentant des échecs de démarrage égaux ou supérieurs `protected_failure_count`, la partition est définie sur une INACTIVE fois les tâches en cours d'exécution terminées.

Examinez les états des nœuds illustrés dans l'exemple suivant.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1*  inact infinite 10  down% queue1-dy-c5xlarge-[1-10]
queue1*  inact infinite 3490 idle~ queue1-dy-c5xlarge-[11-3500]
queue2  up  infinite 10  idle~ queue2-dy-c5xlarge-[1-10]
```

queue1La partition est INACTIVE due au fait que 10 défaillances consécutives du bootstrap du nœud de calcul ont été détectées.

Les instances situées derrière les nœuds ont `queue1-dy-c5xlarge-[1-10]` été lancées mais n'ont pas réussi à rejoindre le cluster en raison d'un état défectueux.

Le cluster est protégé.

La partition `queue2` n'est pas affectée par les échecs de bootstrap dans `queue1`. Il est dans l'UPétat et peut toujours exécuter des tâches.

Comment désactiver le statut protégé

Une fois l'erreur de démarrage résolue, vous pouvez exécuter la commande suivante pour sortir le cluster de son statut protégé.

```
$ pcluster update-compute-fleet --cluster-name <cluster-name> \  
  --region <region-id> \  
  --status START_REQUESTED
```

Défaillances du bootstrap qui activent le statut protégé

Les erreurs de démarrage qui activent le statut protégé sont subdivisées en trois types suivants. Pour identifier le type et le problème, vous pouvez vérifier si des journaux AWS ParallelCluster ont été générés. Si des journaux ont été générés, vous pouvez vérifier les détails des erreurs. Pour plus d'informations, consultez [Récupération et conservation des journaux](#).

1. Erreur de démarrage qui provoque l'arrêt automatique d'une instance.

Une instance échoue au début du processus de démarrage, par exemple une instance qui s'arrête automatiquement en raison d'erreurs dans le script [SlurmQueues\CustomActions\OnNodeStart](#)]. [OnNodeConfigured](#)

Pour les nœuds dynamiques, recherchez les erreurs similaires aux suivantes :

```
Node bootstrap error: Node ... is in power up state without valid backing instance
```

Pour les nœuds statiques, recherchez dans le `clustermgtd` journal (`/var/log/parallelcluster/clustermgtd`) les erreurs similaires aux suivantes :

```
Node bootstrap error: Node ... is in power up state without valid backing instance
```

2. Nœuds `resume_timeout` ou `node_replacement_timeout` expirations.

Une instance ne peut pas rejoindre le cluster au sein du `resume_timeout` (pour les nœuds dynamiques) ou `node_replacement_timeout` (pour les nœuds statiques). Il ne s'arrête pas automatiquement avant le délai imparti. Par exemple, la mise en réseau n'est pas correctement configurée pour le cluster et le nœud est réglé à l'`DOWN`état une Slurm fois le délai expiré.

Pour les nœuds dynamiques, recherchez les erreurs similaires aux suivantes :

```
Node bootstrap error: Resume timeout expires for node
```

Pour les nœuds statiques, recherchez dans le `clustermgtd` journal (`/var/log/parallelcluster/clustermgtd`) les erreurs similaires aux suivantes :

```
Node bootstrap error: Replacement timeout expires for node ... in replacement.
```

3. Le contrôle de santé des nœuds échoue.

Une instance située derrière le nœud échoue à une vérification de l'état d'Amazon EC2 ou à une vérification de l'état d'un événement planifié, et les nœuds sont traités comme des nœuds de défaillance du bootstrap. Dans ce cas, l'instance se termine pour une raison indépendante de la volonté de AWS ParallelCluster.

Recherchez dans le `clustermgtd` journal (`/var/log/parallelcluster/clustermgtd`) les erreurs similaires aux suivantes :

```
Node bootstrap error: Node %s failed during bootstrap when performing health check.
```

4. L'Slurmenregistrement des nœuds de calcul échoue.

L'enregistrement du `slurmd` démon auprès du démon de Slurm contrôle (`slurmctld`) échoue et fait passer l'état du nœud de calcul à cet état. `INVALID_REG` Des nœuds de Slurm calcul mal configurés peuvent provoquer cette erreur, par exemple des nœuds calculés configurés avec des erreurs de spécification de nœuds de [CustomSlurmSettings](#)calcul.

Consultez le fichier `slurmctld` journal (`/var/log/slurmctld.log`) du nœud principal ou le fichier `slurmd` journal (`/var/log/slurmd.log`) du nœud de calcul défaillant pour détecter des erreurs similaires aux suivantes :

```
Setting node %s to INVALID with reason: ...
```

Comment déboguer le mode protégé

Si votre cluster est protégé et s'il a AWS ParallelCluster généré des `clustermgtd` journaux à partir des nœuds de calcul problématiques `HeadNode` et des `cloud-init-output` journaux à partir de nœuds de calcul, vous pouvez consulter les journaux pour obtenir des informations détaillées sur les

erreurs. Pour plus d'informations sur la façon de récupérer les journaux, consultez [Récupération et conservation des journaux](#).

clustermgtdlog (/var/log/parallelcluster/clustermgtd) sur le nœud principal

Les messages du journal indiquent quelles partitions présentent des échecs d'amorçage et le nombre d'échecs d'amorçage correspondant.

```
[slurm_plugin.clustermgtd:_handle_protected_mode_process] - INFO - Partitions bootstrap failure count: {'queue1': 2}, cluster will be set into protected mode if protected failure count reach threshold.
```

Dans le `clustermgtd` journal, recherchez le `Found the following bootstrap failure nodes` nœud qui n'a pas pu démarrer.

```
[slurm_plugin.clustermgtd:_handle_protected_mode_process] - WARNING - Found the following bootstrap failure nodes: (x2) ['queue1-st-c5large-1(192.168.110.155)', 'broken-st-c5large-2(192.168.65.215)']
```

Dans le `clustermgtd` journal, recherchez `Node bootstrap error` la raison de l'échec.

```
[slurm_plugin.clustermgtd:_is_node_bootstrap_failure] - WARNING - Node bootstrap error: Node broken-st-c5large-2(192.168.65.215) is currently in replacement and no backing instance
```

cloud-init-outputlog (/var/log/cloud-init-output.log) sur les nœuds de calcul

Après avoir obtenu l'adresse IP privée du nœud de défaillance du bootstrap dans le `clustermgtd` journal, vous pouvez trouver le journal du nœud de calcul correspondant en vous connectant au nœud de calcul ou en suivant les instructions [Récupération et conservation des journaux](#) pour récupérer les journaux. Dans la plupart des cas, le `/var/log/cloud-init-output` journal du nœud problématique indique l'étape à l'origine de l'échec du bootstrap du nœud de calcul.

Slurmbasculement rapide d'une capacité insuffisante du cluster

À partir de AWS ParallelCluster la version 3.2.0, les clusters s'exécutent avec le mode de basculement rapide en cas de capacité insuffisante activé par défaut. Cela permet de réduire le temps passé à essayer à nouveau de mettre une tâche en file d'attente lorsque des erreurs liées à une capacité insuffisante d'Amazon EC2 sont détectées. Cela est particulièrement efficace lorsque vous configurez votre cluster avec plusieurs types d'instances.

Amazon EC2 a détecté des défaillances de capacité insuffisantes :

- `InsufficientInstanceCapacity`
- `InsufficientHostCapacity`
- `InsufficientReservedInstanceCapacity`
- `MaxSpotInstanceCountExceeded`
- `SpotMaxPriceTooLow`: Activé si le prix de la demande Spot est inférieur au prix minimum requis pour le traitement des demandes Spot.
- `Unsupported`: Activé avec l'utilisation d'un type d'instance qui n'est pas pris en charge dans un environnement spécifique Région AWS.

En mode de défaillance rapide en cas d'insuffisance de capacité, si une erreur de capacité insuffisante est détectée lorsqu'une tâche est affectée à un [SlurmQueues/compute_resource](#), procédez comme suit AWS ParallelCluster :

1. Il met la ressource de calcul à l'état désactivé (DOWN) pendant une période prédéfinie.
2. Il est utilisé `POWER_DOWN_FORCE` pour annuler les tâches du nœud défaillant sur les ressources de calcul et pour suspendre le nœud défaillant. Il place le nœud défaillant à l'`POWER_DOWN (!)` état IDLE et, ensuite, à `POWERING_DOWN (%)`.
3. Il confie la tâche à une autre ressource de calcul.

Les nœuds statiques et sous tension de la ressource de calcul désactivée ne sont pas affectés. Les tâches peuvent être effectuées sur ces nœuds.

Ce cycle se répète jusqu'à ce que la tâche soit correctement assignée à un ou plusieurs nœuds de ressources informatiques. Pour plus d'informations sur les états des nœuds, consultez le [Slurmguide pour le mode de file d'attente multiple](#).

Si aucune ressource de calcul n'est trouvée pour exécuter la tâche, la tâche est réglée sur l'`PENDING` état jusqu'à ce que la période prédéfinie soit écoulée. Dans ce cas, vous pouvez modifier la période prédéfinie comme décrit dans la section suivante.

Paramètre de temporisation de capacité insuffisant

`insufficient_capacity_timeout`

`insufficient_capacity_timeout` indique la durée (en secondes) pendant laquelle la ressource de calcul est maintenue dans l'état désactivé (down) lorsqu'une erreur de capacité insuffisante est détectée.

Par défaut, `insufficient_capacity_timeout` est activé.

La valeur par défaut `insufficient_capacity_timeout` est de 600 secondes (10 minutes).

Si la `insufficient_capacity_timeout` valeur est inférieure ou égale à zéro, le mode de défaillance rapide en cas d'insuffisance de capacité est désactivé.

Vous pouvez modifier la `insufficient_capacity_timeout` valeur en ajoutant le paramètre dans le fichier de `clustermgtd` configuration situé `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` dans le `HeadNode`.

Le paramètre peut être mis à jour à tout moment sans arrêter le parc informatique.

Par exemple :

- `insufficient_capacity_timeout=600`:

Si une erreur de capacité insuffisante est détectée, la ressource de calcul est définie sur Disabled (DOWN). Au bout de 10 minutes, son nœud défaillant passe à l'état `idle~` (POWER_SAVING).

- `insufficient_capacity_timeout=60`:

Si une erreur de capacité insuffisante est détectée, la ressource de calcul est désactivée (DOWN). Au bout d'une minute, son nœud défaillant est réglé sur `idle~` cet état.

- `insufficient_capacity_timeout=0`:

Le mode de défaillance rapide en cas d'insuffisance de capacité est désactivé. La ressource de calcul n'est pas désactivée.

Note

Il peut s'écouler jusqu'à une minute entre le moment où les nœuds tombent en panne en raison d'erreurs de capacité insuffisante et le moment où le démon de gestion du cluster détecte les défaillances des nœuds. Cela est dû au fait que le démon de gestion du cluster vérifie les défaillances de capacité insuffisante des nœuds et définit les ressources de calcul à down intervalles d'une minute.

État du mode de basculement rapide en cas de capacité insuffisante

Lorsqu'un cluster est en mode de basculement rapide en capacité insuffisante, vous pouvez vérifier son état et l'état de ses nœuds.

États des nœuds

Lorsqu'une tâche est soumise à un nœud dynamique de ressources informatiques et qu'une erreur de capacité insuffisante est détectée, le nœud est placé dans `down#` cet état avec raison.

```
(Code:InsufficientInstanceCapacity)Failure when resuming nodes.
```

Ensuite, les nœuds éteints (nœuds en `idle~` état) sont définis `down~` avec raison.

```
(Code:InsufficientInstanceCapacity)Temporarily disabling node due to insufficient capacity.
```

La tâche est mise en file d'attente vers les autres ressources de calcul de la file d'attente.

Les nœuds statiques des ressources de calcul et les nœuds qui UP ne sont pas affectés par le mode de basculement rapide en capacité insuffisante.

Examinez les états des nœuds illustrés dans l'exemple suivant.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1*   up    infinite   30  idle~ queue1-dy-c-1-[1-15],queue1-dy-c-2-[1-15]
queue2    up    infinite   30  idle~ queue2-dy-c-1-[1-15],queue2-dy-c-2-[1-15]
```

Nous soumettons à `queue1` une tâche qui nécessite un nœud.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1*   up    infinite   1   down# queue1-dy-c-1-1
queue1*   up    infinite  15  idle~ queue1-dy-c-2-[1-15]
queue1*   up    infinite  14  down~ queue1-dy-c-1-[2-15]
queue2    up    infinite  30  idle~ queue2-dy-c-1-[1-15],queue2-dy-c-2-[1-15]
```

`queue1-dy-c-1-1`Le nœud est lancé pour exécuter la tâche. Cependant, l'instance n'a pas pu être lancée en raison d'une erreur de capacité insuffisante. `queue1-dy-c-1-1`Le nœud est défini

surdown. Le nœud dynamique hors tension au sein de la ressource de calcul (queue2-dy-c-1) est défini surdown.

Vous pouvez vérifier la raison du nœud avec `scontrol show nodes`.

```
$ scontrol show nodes queue1-dy-c-1-1
NodeName=broken-dy-c-2-1 Arch=x86_64 CoresPerSocket=1
CPUAlloc=0 CPUTot=96 CPULoad=0.00
...
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
Reason=(Code:InsufficientInstanceCapacity)Failure when resuming nodes
[root@2022-03-10T22:17:50]

$ scontrol show nodes queue1-dy-c-1-2
NodeName=broken-dy-c-2-1 Arch=x86_64 CoresPerSocket=1
CPUAlloc=0 CPUTot=96 CPULoad=0.00
...
ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
Reason=(Code:InsufficientInstanceCapacity)Temporarily disabling node due to
insufficient capacity [root@2022-03-10T22:17:50]
```

La tâche est mise en file d'attente vers un autre type d'instance dans les ressources de calcul de la file d'attente.

Une fois ce `insufficient_capacity_timeout` délai écoulé, les nœuds de la ressource de calcul sont remis à l'`idle` état.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1*   up    infinite    30  idle~ queue1-dy-c-1-[1-15],queue1-dy-c-2-[1-15]
queue2    up    infinite    30  idle~ queue2-dy-c-1-[1-15],queue2-dy-c-2-[1-15]
```

Une fois que les nœuds de la ressource de calcul se sont `insufficient_capacity_timeout` écoulés et que leur `idle` état est rétabli, le Slurm planificateur leur attribue une priorité inférieure.

Le planificateur continue de sélectionner des nœuds parmi d'autres ressources de calcul de file d'attente avec des pondérations plus élevées, sauf si l'une des situations suivantes se produit :

- Les exigences de soumission d'une tâche correspondent à la ressource de calcul récupérée.
- Aucune autre ressource de calcul n'est disponible car elle est à pleine capacité.
- `slurmctld` est redémarré.

- Le parc AWS ParallelCluster informatique est arrêté et démarre pour mettre hors tension et alimenter tous les nœuds.

Journaux connexes

Les journaux relatifs aux erreurs de capacité insuffisante et au mode de basculement rapide en cas d'insuffisance de capacité se trouvent dans Slurm le `resume` journal et dans le `clustermgtd` journal du nœud principal.

Slurm `resume` (`/var/log/parallelcluster/slurm_resume.log`)

Messages d'erreur lorsqu'un nœud ne parvient pas à démarrer en raison d'une capacité insuffisante.

```
[slurm_plugin.instance_manager:_launch_ec2_instances] - ERROR - Failed RunInstances request: dcd0c252-90d4-44a7-9c79-ef740f7ecd87
[slurm_plugin.instance_manager:add_instances_for_nodes] - ERROR - Encountered exception when launching instances for nodes (x1) ['queue1-dy-c-1-1']: An error occurred
(InsufficientInstanceCapacity) when calling the RunInstances operation (reached max retries: 1): We currently do not have sufficient p4d.24xlarge capacity in the Availability Zone you requested (us-west-2b). Our system will be working on provisioning additional capacity. You can currently get p4d.24xlarge capacity by not specifying an Availability Zone in your request or choosing us-west-2a, us-west-2c.
```

Slurm `clustermgtd` (`/var/log/parallelcluster/clustermgtd`)

La ressource de calcul c-1 de la file d'attente 1 est désactivée en raison d'une capacité insuffisante.

```
[slurm_plugin.clustermgtd:_reset_timeout_expired_compute_resources] - INFO - The following compute resources are in down state due to insufficient capacity: {'queue1': {'c-1': ComputeResourceFailureEvent(timestamp=datetime.datetime(2022, 4, 14, 23, 0, 4, 769380, tzinfo=datetime.timezone.utc), error_code='InsufficientInstanceCapacity')}}}, compute resources are reset after insufficient capacity timeout (600 seconds) expired
```

Une fois le délai d'expiration de la capacité insuffisante expiré, la ressource de calcul est réinitialisée et les nœuds des ressources de calcul sont définis sur `idle~`

```
[root:_reset_insufficient_capacity_timeout_expired_nodes] - INFO - Reset the
following compute resources because insufficient capacity
timeout expired: {'queue1': ['c-1']}
```

Slurmplanification basée sur la mémoire

À partir de la version 3.2.0, AWS ParallelCluster prend en charge la planification Slurm basée sur la mémoire avec le paramètre de configuration [SlurmSettings/EnableMemoryBasedScheduling](#)cluster.

Note

À partir de AWS ParallelCluster la version 3.7.0, elle `EnableMemoryBasedScheduling` peut être activée si vous configurez plusieurs types d'[instances dans Instances](#). Pour AWS ParallelCluster les versions 3.2.0 à 3.6. **x**, ne `EnableMemoryBasedScheduling` peut pas être activé si vous configurez plusieurs types d'[instances dans Instances](#).

Warning

Lorsque vous spécifiez plusieurs types d'instances dans une ressource de calcul de Slurm file d'attente avec `EnableMemoryBasedScheduling` Activé, la `RealMemory` valeur est la quantité minimale de mémoire disponible pour tous les types d'instances. Cela peut entraîner d'importantes quantités de mémoire inutilisées si vous spécifiez des types d'instances dotés de capacités de mémoire très différentes.

Avec `EnableMemoryBasedScheduling: true`, le Slurm planificateur suit la quantité de mémoire requise par chaque tâche sur chaque nœud. Le Slurm planificateur utilise ensuite ces informations pour planifier plusieurs tâches sur le même nœud de calcul. La quantité totale de mémoire requise par les tâches sur un nœud ne peut pas être supérieure à la mémoire disponible du nœud. Le planificateur empêche une tâche d'utiliser plus de mémoire que ce qui a été demandé lors de la soumission de la tâche.

Avec `EnableMemoryBasedScheduling: false`, les tâches peuvent se disputer la mémoire sur un nœud partagé et provoquer des échecs et `out-of-memory` des événements.

⚠ Warning

Slurm utilise une notation de puissance 2 pour ses étiquettes, telles que MB ou GB. Lisez ces étiquettes sous la forme MiB et GiB, respectivement.

Slurmconfiguration et planification basée sur la mémoire

Avec `EnableMemoryBasedScheduling: true`, Slurm définit les paramètres Slurm de configuration suivants :

- [SelectTypeParameters=CR_CPU_Memory](#) dans le `slurm.conf`. Cette option configure la mémoire du nœud comme une ressource consommable dans Slurm
- [ConstrainRAMSpace=yes](#) dans le `Slurmgroup.conf`. Avec cette option, l'accès à la mémoire d'une tâche est limité à la quantité de mémoire demandée lors de son envoi.

i Note

Plusieurs autres paramètres Slurm de configuration peuvent avoir un impact sur le comportement du Slurm planificateur et du gestionnaire de ressources lorsque ces deux options sont définies. Pour de plus amples d'informations, consultez [la documentation Slurm](#).

Slurmplanificateur et planification basée sur la mémoire

EnableMemoryBasedScheduling: false(par défaut)

Par défaut, `EnableMemoryBasedScheduling` est défini sur `false`. Lorsque la valeur est fausse, Slurm elle n'inclut pas la mémoire en tant que ressource dans son algorithme de planification et ne suit pas la mémoire utilisée par les tâches. Les utilisateurs peuvent spécifier l'option `--mem MEM_PER_NODE` permettant de définir la quantité minimale de mémoire par nœud requise par une tâche. Cela oblige le planificateur à choisir des nœuds avec une `RealMemory` valeur d'au moins égale à `MEM_PER_NODE` lors de la planification de la tâche.

Supposons, par exemple, qu'un utilisateur soumette deux tâches avec `--mem=5GB`. Si les ressources demandées, telles que des processeurs ou des GPU, sont disponibles, les tâches peuvent être exécutées simultanément sur un nœud doté de 8 GiB de mémoire. Les deux tâches ne sont pas planifiées sur des nœuds de calcul de moins de 5 GiB de `RealMemory`

⚠ Warning

Lorsque la planification basée sur la mémoire est désactivée, Slurm elle ne suit pas la quantité de mémoire utilisée par les tâches. Les tâches exécutées sur le même nœud peuvent se disputer les ressources de mémoire et entraîner l'échec de l'autre tâche. Lorsque la planification basée sur la mémoire est désactivée, nous recommandons aux utilisateurs de ne pas spécifier les options `--mem-per-cpu` ou `--mem-per-gpu`. Ces options peuvent entraîner un comportement différent de celui décrit dans la [Slurmdocumentation](#).

EnableMemoryBasedScheduling: true

Lorsque `EnableMemoryBasedScheduling` ce paramètre est défini sur `true`, Slurm suit l'utilisation de la mémoire de chaque tâche et empêche les tâches d'utiliser plus de mémoire que ce qui est demandé avec les options de `--mem` soumission.

Dans l'exemple précédent, un utilisateur soumet deux tâches avec `--mem=5GB`. Les tâches ne peuvent pas être exécutées en même temps sur un nœud doté de 8 GiB de mémoire. Cela est dû au fait que la quantité totale de mémoire requise est supérieure à la mémoire disponible sur le nœud.

Lorsque la planification basée sur la mémoire est activée, `--mem-per-cpu` `--mem-per-gpu` comportez-vous conformément à ce qui est décrit dans la Slurm documentation. Par exemple, une tâche est soumise avec `--ntasks-per-node=2 -c 1 --mem-per-cpu=2GB`. Dans ce cas, Slurm attribue à la tâche un total de 4 GiB pour chaque nœud.

⚠ Warning

Lorsque la planification basée sur la mémoire est activée, nous recommandons aux utilisateurs d'inclure une `--mem` spécification lors de la soumission d'une tâche. Avec la Slurm configuration par défaut incluse dans AWS ParallelCluster, si aucune option de mémoire n'est incluse (`--mem`, `--mem-per-cpu`, ou `--mem-per-gpu`), Slurm affecte la totalité de la mémoire des nœuds alloués à la tâche, même si elle ne demande qu'une partie des autres ressources, telles que les processeurs ou les GPU. Cela empêche efficacement le partage de nœuds tant que la tâche n'est pas terminée, car aucune mémoire n'est disponible pour les autres tâches. Cela se produit car Slurm définit la mémoire par nœud pour la tâche comme [DefMemPerNode](#) si aucune spécification de mémoire n'est fournie au moment de

la soumission de la tâche. La valeur par défaut de ce paramètre est 0 et indique un accès illimité à la mémoire d'un nœud.

Si plusieurs types de ressources informatiques avec différentes quantités de mémoire sont disponibles dans la même file d'attente, une tâche soumise sans options de mémoire peut se voir attribuer différentes quantités de mémoire sur différents nœuds. Cela dépend des nœuds que le planificateur met à la disposition de la tâche. Les utilisateurs peuvent définir une valeur personnalisée pour les options, telles que `DefMemPerNode` ou [DefMemPerCPU](#), au niveau du cluster ou de la partition dans les fichiers Slurm de configuration afin d'empêcher ce comportement.

Slurm RealMemory et AWS ParallelCluster SchedulableMemory

Avec la Slurm configuration fournie avec AWS ParallelCluster, est Slurm interprétée comme [RealMemory](#) étant la quantité de mémoire disponible pour les tâches par nœud. À partir de la version 3.2.0, la valeur par défaut AWS ParallelCluster est définie `RealMemory` sur 95 % de la mémoire répertoriée dans les types d'[instances Amazon EC2](#) et renvoyée par l'API Amazon EC2. [DescribeInstanceTypes](#)

Lorsque la planification basée sur la mémoire est désactivée, le Slurm planificateur filtre les nœuds lorsque `RealMemory` les utilisateurs soumettent une tâche avec une valeur spécifiée. `--mem`

Lorsque la planification basée sur la mémoire est activée, le Slurm planificateur interprète comme étant `RealMemory` la quantité maximale de mémoire disponible pour les tâches exécutées sur le nœud de calcul.

Le paramètre par défaut n'est peut-être pas optimal pour tous les types d'instances :

- Ce paramètre peut être supérieur à la quantité de mémoire à laquelle les nœuds peuvent réellement accéder. Cela peut se produire lorsque les nœuds de calcul sont de petits types d'instances.
- Ce paramètre peut être inférieur à la quantité de mémoire à laquelle les nœuds peuvent réellement accéder. Cela peut se produire lorsque les nœuds de calcul sont des types d'instances de grande taille et peuvent entraîner une quantité importante de mémoire inutilisée.

Vous pouvez utiliser [SlurmQueues/ComputeResources/SchedulableMemory](#) pour affiner la valeur de `RealMemory` configuré par AWS ParallelCluster pour les nœuds de calcul. Pour remplacer

la valeur par défaut, définissez une valeur personnalisée `SchedulableMemory` spécifiquement adaptée à la configuration de votre cluster.

Pour vérifier la mémoire réellement disponible d'un nœud de calcul, exécutez la `/opt/slurm/sbin/slurmd -C` commande sur le nœud. Cette commande renvoie la configuration matérielle du nœud, y compris la [RealMemory](#) valeur. Pour plus d'informations, consultez [slurmd -C](#).

Assurez-vous que les processus du système d'exploitation du nœud de calcul disposent de suffisamment de mémoire. Pour ce faire, limitez la mémoire disponible pour les tâches en définissant `SchedulableMemory` une valeur inférieure à la `RealMemory` valeur renvoyée par la `slurmd -C` commande.

Allocation de plusieurs types d'instances avec Slurm

À partir de AWS ParallelCluster la version 3.3.0, vous pouvez configurer votre cluster pour qu'il alloue à partir de l'ensemble de types d'instances définis d'une ressource de calcul. L'allocation peut être basée sur des stratégies de faible coût ou de capacité optimale de la flotte Amazon EC2.

Cet ensemble de types d'instances définis doit tous avoir le même nombre de vCPU ou, si le multithreading est désactivé, le même nombre de cœurs. De plus, cet ensemble de types d'instances doit comporter le même nombre d'accélérateurs du même fabricant. Si `Efa/Enabled` est défini sur `true`, les instances doivent être compatibles avec l'EFA. Pour plus d'informations et pour connaître les exigences, consultez [SchedulingSlurmQueues//AllocationStrategy](#) et [ComputeResources/Instances](#).

Vous pouvez [AllocationStrategy](#) choisir `lowest-price` ou `capacity-optimized` en fonction de votre [CapacityType](#) configuration.

Dans [Instances](#), vous pouvez configurer un ensemble de types d'instances.

Note

À partir de AWS ParallelCluster la version 3.7.0, elle `EnableMemoryBasedScheduling` peut être activée si vous configurez plusieurs types d'[instances dans Instances](#).

Pour AWS ParallelCluster les versions 3.2.0 à 3.6. x, `EnableMemoryBasedScheduling` peut pas être activé si vous configurez plusieurs types d'[instances dans Instances](#).

Les exemples suivants montrent comment interroger les types d'instances pour les vCPU, le support EFA et l'architecture.

Requête InstanceTypes avec 96 vCPU et architecture x86_64.

```
$ aws ec2 describe-instance-types --region region-id \
  --filters "Name=vcpu-info.default-vcpus,Values=96" "Name=processor-info.supported-
architecture,Values=x86_64" \
  --query "sort_by(InstanceTypes[*]).
{InstanceType:InstanceType,MemoryMiB:MemoryInfo.SizeInMiB,CurrentGeneration:CurrentGeneration,V
&InstanceType)" \
  --output table
```

Requête InstanceTypes avec 64 cœurs, support EFA et architecture arm64.

```
$ aws ec2 describe-instance-types --region region-id \
  --filters "Name=vcpu-info.default-cores,Values=64" "Name=processor-
info.supported-architecture,Values=arm64" "Name=network-info.efa-
supported,Values=true" --query "sort_by(InstanceTypes[*]).
{InstanceType:InstanceType,MemoryMiB:MemoryInfo.SizeInMiB,CurrentGeneration:CurrentGeneration,V
&InstanceType)" \
  --output table
```

L'exemple d'extrait de configuration de cluster suivant montre comment vous pouvez les utiliser InstanceType ainsi AllocationStrategy que leurs propriétés.

```
...
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue-1
      CapacityType: ONDEMAND
      AllocationStrategy: lowest-price
      ...
      ComputeResources:
        - Name: computeresource1
          Instances:
            - InstanceType: r6g.2xlarge
            - InstanceType: m6g.2xlarge
            - InstanceType: c6g.2xlarge
          MinCount: 0
          MaxCount: 500
        - Name: computeresource2
          Instances:
            - InstanceType: m6g.12xlarge
```

```
- InstanceType: x2gd.12xlarge
MinCount: 0
MaxCount: 500
...
```

Mise à l'échelle du cluster pour les nœuds dynamiques

ParallelCluster les méthodes Slurm de supports pour dimensionner dynamiquement les clusters en utilisant Slurm le plugin d'économie d'énergie. Pour plus d'informations, consultez le [guide de planification dans le cloud](#) et le [guide d'économie d'Slurménergie](#) dans la Slurm documentation.

À partir de ParallelCluster la version 3.8.0, ParallelCluster utilise la reprise au niveau du travail ou le dimensionnement au niveau du travail comme stratégie d'allocation dynamique de nœuds par défaut pour dimensionner le cluster : ParallelCluster augmente le cluster en fonction des exigences de chaque tâche, du nombre de nœuds alloués à la tâche et des nœuds devant être repris. ParallelCluster obtient ces informations à partir de la variable d'environnement `SLURM_RESUME_FILE`.

Le dimensionnement pour les nœuds dynamiques est un processus en deux étapes, qui implique le lancement des instances EC2 et l'attribution des instances Amazon EC2 lancées aux Slurm nœuds. Chacune de ces deux étapes peut être réalisée en utilisant une all-or-nothing logique basée sur le meilleur effort.

Pour le lancement des instances Amazon EC2 :

- all-or-nothing appelle l'API de lancement Amazon EC2 avec une cible minimale égale à la capacité cible totale
- best-effort appelle le lancement de l'API Amazon EC2 avec une cible minimale égale à 1 et une capacité cible totale égale à la capacité demandée

Pour l'attribution des instances Amazon EC2 aux Slurm nœuds :

- all-or-nothing attribue des instances Amazon EC2 Slurm à des nœuds uniquement s'il est possible d'attribuer une instance Amazon EC2 à chaque nœud demandé
- tout est fait pour attribuer les instances Amazon EC2 Slurm aux nœuds, même si tous les nœuds demandés ne sont pas couverts par la capacité des instances Amazon EC2

Les combinaisons possibles des stratégies ci-dessus se traduisent par les stratégies de ParallelCluster lancement.

Exemple

<caption>The available ParallelCluster stratégies de lancement that can be set into the [ScalingStrategy](#) cluster configuration to be used with mise à l'échelle au niveau des tâches are:</caption>

all-or-nothingmise à l'échelle :

Cette stratégie implique de AWS ParallelCluster lancer un appel d'API d'instance de lancement Amazon EC2 pour chaque tâche, ce qui nécessite le lancement réussi de toutes les instances nécessaires au lancement des nœuds de calcul demandés. Cela garantit que le cluster évolue uniquement lorsque la capacité requise par tâche est disponible, évitant ainsi de laisser des instances inactives à la fin du processus de dimensionnement.

La stratégie utilise une all-or-nothinglogique pour le lancement des instances Amazon EC2 pour chaque tâche et une all-or-nothinglogique pour l'attribution des instances Amazon EC2 aux nœuds. Slurm

Les groupes de stratégie lancent les demandes par lots, un pour chaque ressource de calcul demandée et jusqu'à 500 nœuds chacun. Pour les demandes couvrant plusieurs ressources informatiques ou dépassant 500 nœuds, traite plusieurs lots de ParallelCluster manière séquentielle.

La défaillance du lot d'une ressource unique entraîne l'arrêt de toutes les capacités inutilisées associées, garantissant ainsi qu'aucune instance inactive ne sera laissée à la fin du processus de dimensionnement.

Limites

- Le temps nécessaire à la mise à l'échelle est directement proportionnel au nombre de tâches soumises par exécution du programme de Slurm CV.
- L'opération de dimensionnement est limitée par la limite du compte de RunInstances ressources, fixée à 1 000 instances par défaut. Cette limitation est conforme aux politiques AWS de limitation des API EC2. Pour plus de détails, consultez la documentation relative à la limitation des API [Amazon EC2](#)
- Lorsque vous soumettez une tâche dans une ressource de calcul avec un seul type d'instance, dans une file d'attente qui couvre plusieurs zones de disponibilité, l'appel d'API de lancement all-or-nothingEC2 ne réussit que si toute la capacité peut être fournie dans une seule zone de disponibilité.
- Lorsque vous soumettez une tâche dans une ressource de calcul comportant plusieurs types d'instances, dans une file d'attente avec une seule zone de disponibilité, l'appel d'API de lancement

all-or-nothing Amazon EC2 ne réussit que si toute la capacité peut être fournie par un seul type d'instance.

- Lorsque vous soumettez une tâche dans une ressource de calcul comportant plusieurs types d'instances, dans une file d'attente couvrant plusieurs zones de disponibilité, l'appel d'API de lancement d'all-or-nothing Amazon EC2 n'est pas pris en charge et ParallelCluster effectue plutôt un dimensionnement optimal.

greedy-all-or-nothing mise à l'échelle :

Cette variante de la all-or-nothing stratégie garantit toujours que le cluster évolue uniquement lorsque la capacité requise par tâche est disponible, évitant ainsi les instances inactives à la fin du processus de dimensionnement, mais elle implique de ParallelCluster lancer un appel d'API d'instance de lancement Amazon EC2 qui vise une capacité cible minimale de 1, dans le but de maximiser le nombre de nœuds lancés jusqu'à la capacité demandée. La stratégie utilise une logique basée sur le meilleur effort pour le lancement des instances EC2 pour toutes les tâches, ainsi que la all-or-nothing logique d'attribution des instances Amazon EC2 Slurm aux nœuds pour chaque tâche.

Les groupes de stratégie lancent les demandes par lots, un pour chaque ressource de calcul demandée et jusqu'à 500 nœuds chacun. Pour les demandes couvrant plusieurs ressources de calcul ou dépassant 500 nœuds, ParallelCluster traite plusieurs lots de manière séquentielle.

Cela garantit qu'aucune instance inactive ne sera laissée à la fin du processus de dimensionnement, en maximisant le débit au prix d'un surdimensionnement temporaire pendant le processus de dimensionnement.

Limites

- Un surdimensionnement temporaire est possible, ce qui entraîne des coûts supplémentaires pour les instances qui passent à l'état actif avant la fin du dimensionnement.
- La même limite d'instance que dans la all-or-nothing stratégie s'applique, sous réserve de AWS la limite du compte de RunInstances ressources.

mise à l'échelle optimale :

Cette stratégie appelle l'API de l'instance de lancement Amazon EC2 en ciblant une capacité minimale de 1 et en visant à atteindre la capacité totale demandée au prix de laisser les instances inactives après l'exécution du processus de dimensionnement si toutes les capacités demandées ne sont pas disponibles. La stratégie utilise une logique du meilleur effort pour le lancement des

instances Amazon EC2 pour toutes les tâches, ainsi que la logique du meilleur effort pour l'attribution des instances Amazon EC2 aux nœuds Slurm pour chaque tâche.

Les groupes de stratégie lancent les demandes par lots, un pour chaque ressource de calcul demandée et jusqu'à 500 nœuds chacun. Pour les demandes couvrant plusieurs ressources informatiques ou dépassant 500 nœuds, traite plusieurs lots de ParallelCluster manière séquentielle.

Cette stratégie permet une mise à l'échelle bien au-delà de la limite par défaut de 1 000 instances lors de l'exécution de plusieurs processus de dimensionnement, au prix d'instances inactives au cours des différents processus de dimensionnement.

Limites

- Instances inactives possibles à la fin du processus de dimensionnement, dans le cas où il n'est pas possible d'allouer tous les nœuds demandés par les tâches.

L'exemple suivant montre le comportement de la mise à l'échelle des nœuds dynamiques à l'aide des différentes stratégies de ParallelCluster lancement. Supposons que vous ayez soumis deux tâches demandant 20 nœuds chacune, pour un total de 40 nœuds du même type, mais que seules 30 instances Amazon EC2 soient disponibles pour couvrir la capacité demandée sur EC2.

all-or-nothing mise à l'échelle :

- Pour la première tâche, une API d'instance de lancement all-or-nothing Amazon EC2 est appelée, demandant 20 instances. Un appel réussi entraîne le lancement de 20 instances
- all-or-nothing l'attribution des 20 instances lancées aux Slurm nœuds pour la première tâche est réussie
- Une autre API d'instance de lancement all-or-nothing Amazon EC2 est appelée, demandant 20 instances pour la deuxième tâche. L'appel n'aboutit pas, car il n'y a de capacité que pour 10 autres instances. Aucune instance n'est lancée pour le moment

greedy-all-or-nothing mise à l'échelle :

- Une API d'instance de lancement Amazon EC2 est appelée, demandant 40 instances, ce qui correspond à la capacité totale demandée par toutes les tâches. Cela se traduit par le lancement de 30 instances
- Une all-or-nothing affectation de 20 des instances lancées aux Slurm nœuds pour la première tâche est réussie

- Une autre all-or-nothing attribution des instances lancées restantes aux Slurm nœuds pour la deuxième tâche est tentée, mais comme il n'y a que 10 instances disponibles sur un total de 20 demandées par la tâche, l'attribution échoue
- Les 10 instances lancées non assignées sont résiliées

mise à l'échelle optimale :

- Une API d'instance de lancement Amazon EC2 est appelée, demandant 40 instances, ce qui correspond à la capacité totale demandée par toutes les tâches. Cela se traduit par le lancement de 30 instances.
- Une affectation optimale de 20 des instances lancées à des Slurm nœuds pour la première tâche est réussie.
- Une autre affectation optimale des 10 instances lancées restantes aux Slurm nœuds pour la deuxième tâche est couronnée de succès, même si la capacité totale demandée était de 20. Mais comme la tâche demandait les 20 nœuds et qu'il était possible d'attribuer des instances Amazon EC2 à seulement 10 d'entre eux, la tâche ne peut pas démarrer et les instances restent inactives, jusqu'à ce que la capacité soit trouvée suffisante pour démarrer les 10 instances manquantes lors d'un appel ultérieur du processus de dimensionnement, ou que le planificateur planifie la tâche sur d'autres nœuds de calcul déjà en cours d'exécution.

Slurmstratégies d'allocation dynamique de nœuds dans la version 3.7.x

ParallelCluster utilise deux types de stratégies d'allocation dynamique de nœuds pour dimensionner le cluster :

- Allocation basée sur les informations de nœud demandées disponibles :
 - Reprise de tous les nœuds ou mise à l'échelle de la liste des nœuds :

ParallelCluster augmente le cluster en fonction uniquement des noms Slurm de liste de nœuds demandés lors Slurm de son `ResumeProgram` exécution. Il alloue les ressources de calcul aux nœuds uniquement par nom de nœud. La liste des noms de nœuds peut couvrir plusieurs tâches.

- CV au niveau du poste ou mise à l'échelle du poste :

ParallelCluster fait évoluer le cluster en fonction des exigences de chaque tâche, du nombre actuel de nœuds alloués à la tâche et des nœuds devant être réactivés. ParallelCluster obtient ces informations à partir de la variable d'environnement `SLURM_RESUME_FILE`.

- Allocation avec une stratégie de lancement d'Amazon EC2 :

- Mise à l'échelle optimale :

ParallelCluster augmente le cluster en utilisant un appel d'API d'instance de lancement Amazon EC2 avec une capacité cible minimale égale à 1, pour lancer certaines instances, mais pas nécessairement toutes, nécessaires pour prendre en charge les nœuds demandés.

- Une *all-or-nothing* mise à l'échelle :

ParallelCluster augmente le cluster en utilisant un appel d'API d'instance de lancement Amazon EC2 qui ne réussit que si toutes les instances nécessaires pour prendre en charge les nœuds demandés sont lancées. Dans ce cas, il appelle l'API de l'instance de lancement Amazon EC2 avec une capacité cible minimale égale à la capacité totale demandée.

Par défaut, ParallelCluster utilise le dimensionnement des listes de nœuds avec une stratégie de lancement optimale d'Amazon EC2 pour lancer certaines instances, mais pas nécessairement toutes, nécessaires pour prendre en charge les nœuds demandés. Il essaie de fournir autant de capacité que possible pour répondre à la charge de travail soumise.

À partir de ParallelCluster la version 3.7.0, ParallelCluster utilise le dimensionnement au niveau des tâches avec une stratégie de lancement *all-or-nothing* EC2 pour les tâches soumises en mode exclusif. Lorsque vous soumettez une tâche en mode exclusif, la tâche dispose d'un accès exclusif aux nœuds qui lui sont alloués. Pour plus d'informations, consultez [EXCLUSIVE](#) dans la Slurm documentation.

Pour soumettre une offre d'emploi en mode exclusif :

- Passez le drapeau exclusif lorsque vous soumettez une Slurm tâche au cluster. Par exemple, `sbatch ... --exclusive`.

OU

- Soumettez une tâche à une file d'attente de cluster configurée avec la valeur [JobExclusiveAllocation](#) définie sur `true`.

Lorsque vous soumettez une offre d'emploi en mode exclusif :

- ParallelCluster traite actuellement par lots les demandes de lancement pour inclure jusqu'à 500 nœuds. Si une tâche demande plus de 500 nœuds, ParallelCluster effectue une demande de all-or-nothinglancement pour chaque ensemble de 500 nœuds et une demande de lancement supplémentaire pour le reste des nœuds.
- Si l'allocation de nœuds se fait dans une seule ressource de calcul, ParallelCluster effectue une demande de all-or-nothinglancement pour chaque ensemble de 500 nœuds et une demande de lancement supplémentaire pour le reste des nœuds. Si une demande de lancement échoue, ParallelCluster met fin à la capacité inutilisée créée par toutes les demandes de lancement.
- Si l'allocation de nœuds couvre plusieurs ressources de calcul, vous ParallelCluster devez effectuer une demande de all-or-nothinglancement pour chaque ressource de calcul. Ces demandes sont également groupées. Si une demande de lancement échoue pour l'une des ressources de calcul, ParallelCluster met fin à la capacité inutilisée créée par toutes les demandes de lancement de ressources de calcul.

mise à l'échelle au niveau des tâches avec limites connues de la stratégie de all-or-nothinglancement :

- Lorsque vous soumettez une tâche dans une ressource de calcul avec un seul type d'instance, dans une file d'attente qui couvre plusieurs zones de disponibilité, l'appel d'API de lancement all-or-nothingEC2 ne réussit que si toute la capacité peut être fournie dans une seule zone de disponibilité.
- Lorsque vous soumettez une tâche dans une ressource de calcul comportant plusieurs types d'instances, dans une file d'attente avec une seule zone de disponibilité, l'appel d'API de lancement all-or-nothingAmazon EC2 ne réussit que si toute la capacité peut être fournie par un seul type d'instance.
- Lorsque vous soumettez une tâche dans une ressource de calcul comportant plusieurs types d'instances, dans une file d'attente couvrant plusieurs zones de disponibilité, l'appel d'API de lancement d'all-or-nothingAmazon EC2 n'est pas pris en charge et ParallelCluster effectue plutôt un dimensionnement optimal.

Slurmstratégies d'allocation dynamique de nœuds dans les versions 3.6.x et antérieures

AWS ParallelCluster utilise un seul type de stratégie d'allocation dynamique de nœuds pour dimensionner le cluster :

- Allocation basée sur les informations de nœud demandées disponibles :
 - Reprise de tous les nœuds ou dimensionnement de la liste de nœuds ParallelCluster : augmente le cluster en fonction des noms de liste Slurm de nœuds demandés lors Slurm de son exécution. ResumeProgram Il alloue les ressources de calcul aux nœuds uniquement par nom de nœud. La liste des noms de nœuds peut couvrir plusieurs tâches.
- Allocation avec une stratégie de lancement d'Amazon EC2 :
 - Mise à l' ParallelCluster échelle optimale : augmentez le cluster en utilisant un appel d'API d'instance de lancement Amazon EC2 avec une capacité cible minimale égale à 1, pour lancer certaines instances, mais pas nécessairement toutes, nécessaires pour prendre en charge les nœuds demandés.

ParallelCluster utilise le dimensionnement des listes de nœuds avec une stratégie de lancement optimale d'Amazon EC2 pour lancer certaines instances, mais pas nécessairement toutes, nécessaires pour prendre en charge les nœuds demandés. Il essaie de fournir autant de capacité que possible pour répondre à la charge de travail soumise.

Limites

- Instances inactives possibles à la fin du processus de dimensionnement, dans le cas où il n'est pas possible d'allouer tous les nœuds demandés par les tâches.

Slurmcomptabilité avec AWS ParallelCluster


À partir de la version 3.3.0, AWS ParallelCluster prend en charge la Slurm comptabilité avec le paramètre de configuration du cluster [SlurmSettings/Database](#).

À partir de la version 3.10.0, AWS ParallelCluster prend en charge la Slurm comptabilité avec un Slurmdbd externe avec le paramètre de configuration du cluster/. [SlurmSettingsExternalSlurmdbd](#) L'utilisation d'un Slurmdbd externe est recommandée si plusieurs clusters partagent la même base de données.

Avec la Slurm comptabilité, vous pouvez intégrer une base de données comptable externe pour effectuer les opérations suivantes :

- Gérez les utilisateurs du cluster ou les groupes d'utilisateurs et les autres entités. Grâce à cette fonctionnalité, vous pouvez utiliser Slurm les fonctionnalités les plus avancées, telles que l'application des limites de ressources, le fairshare et la qualité de service.

- Collectez et enregistrez des données de travail, telles que l'utilisateur qui a exécuté le travail, la durée du travail et les ressources qu'il utilise. Vous pouvez consulter les données enregistrées à l'aide de l'[sacctutilitaire](#).

 Note

AWS ParallelCluster prend en charge Slurm la comptabilisation des [serveurs de base de données MySQL Slurm pris](#) en charge.


Utilisation de la Slurm comptabilité à l'aide d'un outil externe Slurmdbd dans les AWS ParallelCluster versions 3.10.0 et ultérieures

Avant de configurer la gestion des Slurm comptes, vous devez disposer d'un serveur de Slurmdbd base de données externe existant, qui se connecte à un serveur de base de données externe existant.

Pour configurer cela, définissez les éléments suivants :

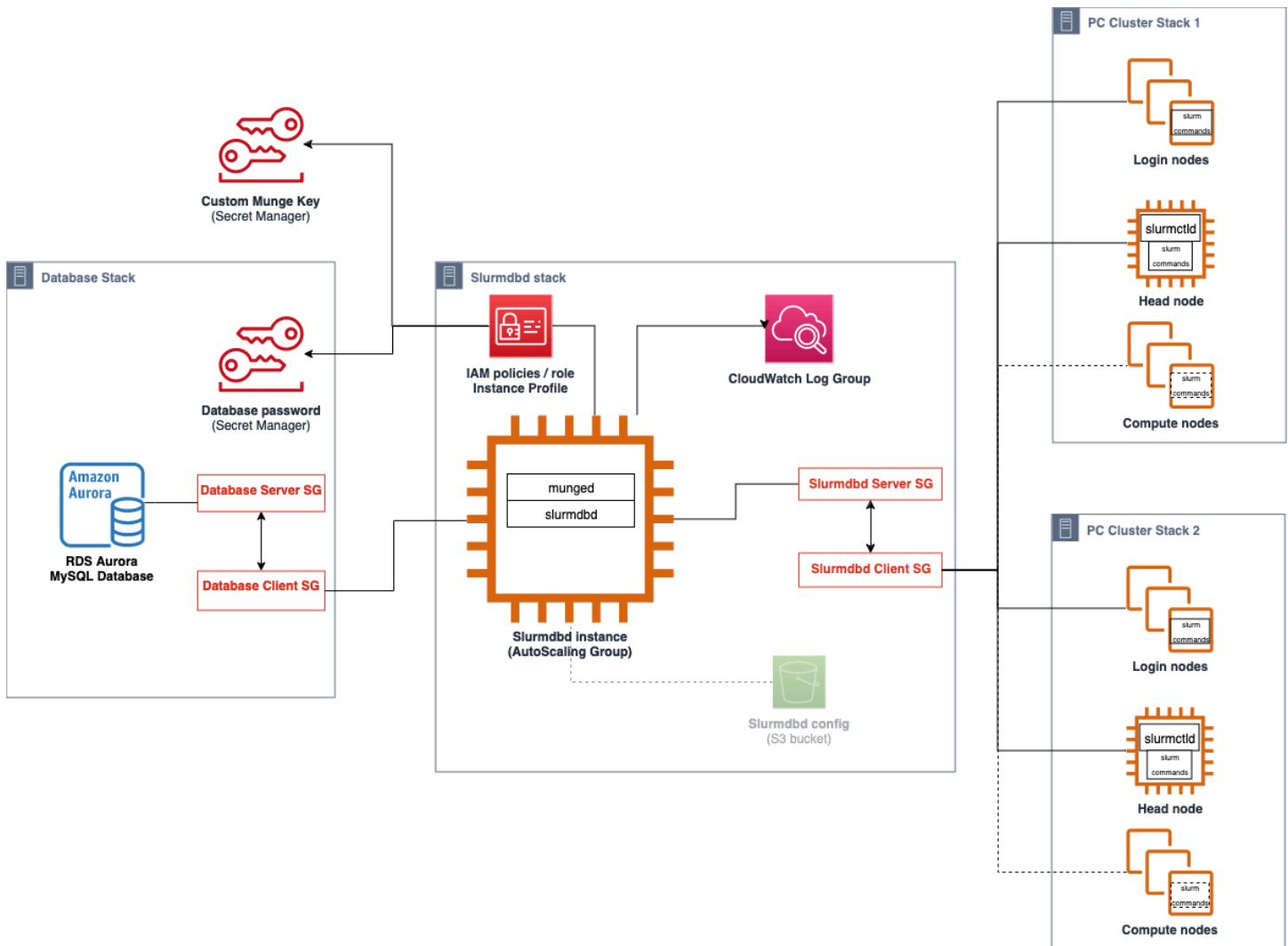
- Adresse du Slurmdbd serveur externe dans [ExternalSlurmdbd/Host](#). Le serveur doit exister et être accessible depuis le nœud principal.
- La touche munge permettant de communiquer avec le Slurmdbd serveur externe.
[MungeKeySecretArn](#)

Pour suivre un didacticiel étape par étape, voir [Création d'un cluster avec une Slurmdbd comptabilité externe](#).

 Note

Vous êtes responsable de la gestion des entités comptables Slurm de la base de données.

L'architecture de la fonctionnalité de SlurmDB support AWS ParallelCluster externe permet à plusieurs clusters de partager SlurmDB la même base de données.



⚠ Warning

Le trafic entre AWS ParallelCluster et l'extérieur n'est pas crypté. Il est recommandé d'exécuter le cluster et l'interface externe SlurmDB sur un réseau sécurisé.

Utilisation de la Slurm comptabilité à l'aide du nœud principal Slurmdbd dans les AWS ParallelCluster versions 3.3.0 et ultérieures

Avant de configurer la gestion des Slurm comptes, vous devez disposer d'un serveur de base de données externe existant et d'une base de données utilisant mysql le protocole.

Pour configurer la Slurm comptabilité avec AWS ParallelCluster, vous devez définir les éléments suivants :

- L'URI du serveur de base de données externe dans [Database/Uri](#). Le serveur doit exister et être accessible depuis le nœud principal.
- Informations d'identification pour accéder à la base de données externe définies dans [Base de données/PasswordSecretArnet](#) [Base de données/UserName](#). AWS ParallelCluster utilise ces informations pour configurer la comptabilité au Slurm niveau et le `slurmdbd` service sur le nœud principal. `slurmdbd` est le daemon qui gère les communications entre le cluster et le serveur de base de données.

Pour suivre un didacticiel étape par étape, voir [Création d'un cluster avec Slurm comptabilité](#).

Note

AWS ParallelCluster exécute un bootstrap de base de données de Slurm comptabilité en définissant l'utilisateur du cluster par défaut comme administrateur de base de données dans la Slurm base de données. AWS ParallelCluster n'ajoute aucun autre utilisateur à la base de données de comptabilité. Le client est responsable de la gestion des entités comptables de la Slurm base de données.

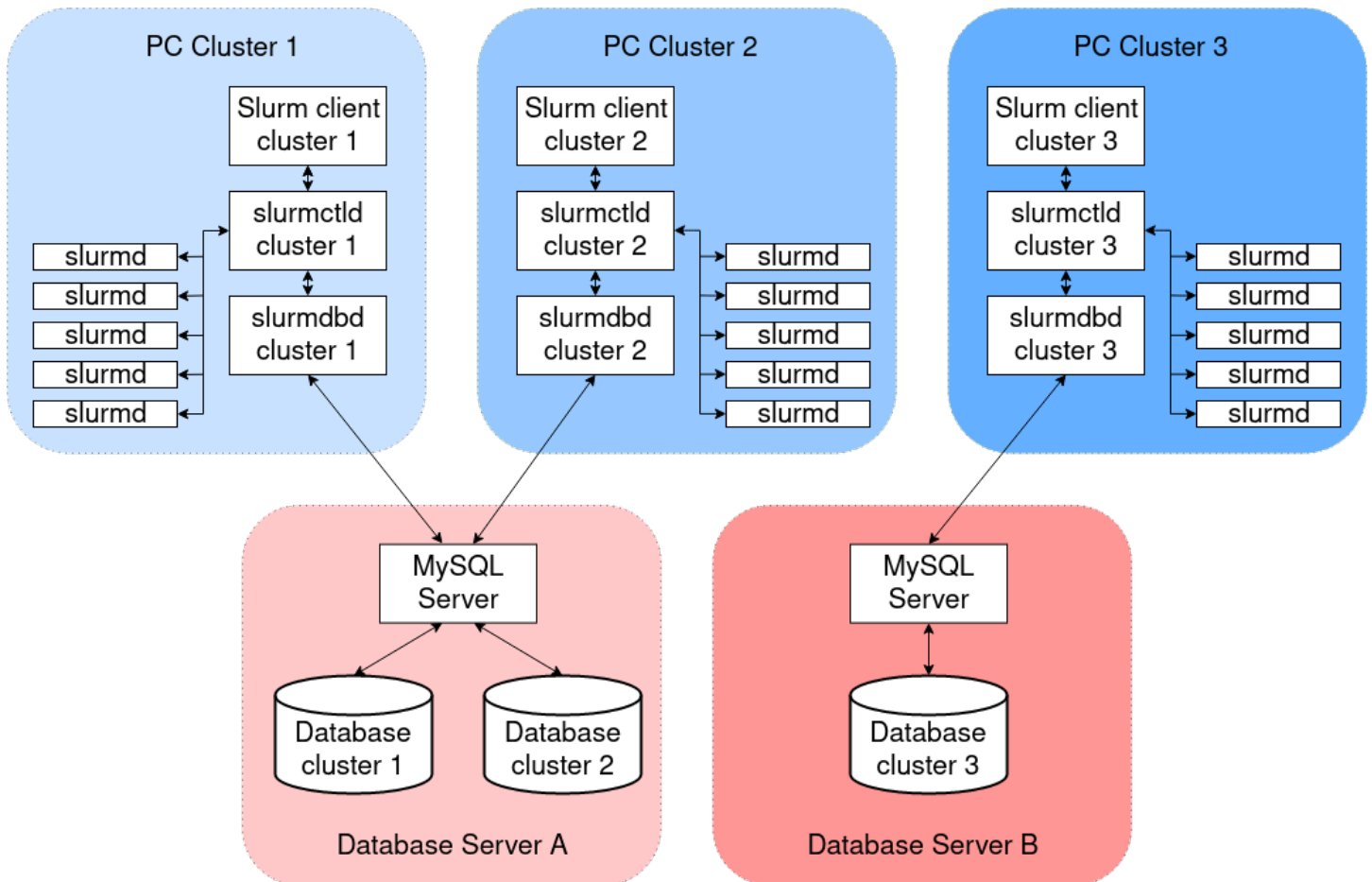
AWS ParallelCluster `slurmdbd` se configure pour garantir qu'un cluster possède sa propre Slurm base de données sur le serveur de base de données. Le même serveur de base de données peut être utilisé sur plusieurs clusters, mais chaque cluster possède sa propre base de données distincte. AWS ParallelCluster utilise le nom du cluster pour définir le nom de la base de données dans le [StorageLoc](#) paramètre du fichier de `slurmdbd` configuration. Considérez la situation suivante. Une base de données présente sur le serveur de base de données inclut un nom de cluster qui ne correspond pas à un nom de cluster actif. Dans ce cas, vous pouvez créer un nouveau cluster portant ce nom de cluster pour le mapper à cette base de données. Slurm réutilise la base de données pour le nouveau cluster.

Warning

- Il est déconseillé de configurer plusieurs clusters pour utiliser la même base de données à la fois. Cela peut entraîner des problèmes de performances ou même des situations de blocage de la base de données.
- Si la gestion des Slurm comptes est activée sur le nœud principal d'un cluster, nous vous recommandons d'utiliser un type d'instance doté d'un processeur puissant, de davantage

de mémoire et d'une bande passante réseau plus importante. Slurm la comptabilité peut alourdir la charge sur le nœud principal du cluster.

Dans l'architecture actuelle de la fonctionnalité de AWS ParallelCluster Slurm comptabilité, chaque cluster possède sa propre instance du `slurmdbd` démon, comme le montrent les exemples de configuration du schéma suivant.



Si vous ajoutez des fonctionnalités personnalisées de Slurm multi-clusters ou de fédération à votre environnement de cluster, tous les clusters doivent faire référence à la même `slurmdbd` instance. Pour cette alternative, nous vous recommandons d'activer la gestion des AWS ParallelCluster Slurm comptes sur un cluster et de configurer manuellement les autres clusters pour `slurmdbd` qu'ils se connectent à ceux hébergés sur le premier cluster.

Si vous utilisez des AWS ParallelCluster versions antérieures à la version 3.3.0, reportez-vous à la méthode alternative pour implémenter la Slurm comptabilité décrite dans ce billet de [blog sur le HPC](#).

Slurmconsidérations comptables

Base de données et cluster sur différents VPC

Pour activer la Slurm comptabilité, un serveur de base de données est nécessaire pour servir de backend pour les opérations de lecture et d'écriture effectuées par le `slurmdbd` démon. Avant que le cluster ne soit créé ou mis à jour pour activer la Slurm comptabilité, le nœud principal doit pouvoir accéder au serveur de base de données.

Si vous devez déployer le serveur de base de données sur un VPC autre que celui utilisé par le cluster, tenez compte des points suivants :

- Pour permettre la communication entre le `slurmdbd` côté cluster et le serveur de base de données, vous devez configurer la connectivité entre les deux VPC. Pour plus d'informations, consultez [VPC Peering dans le guide de l'utilisateur d'Amazon Virtual Private Cloud](#).
- Vous devez créer le groupe de sécurité que vous souhaitez associer au nœud principal sur le VPC du cluster. Une fois que les deux VPC ont été comparés, la liaison croisée entre les groupes de sécurité côté base de données et côté cluster est disponible. Pour plus d'informations, consultez les [règles des groupes de sécurité](#) dans le guide de l'utilisateur d'Amazon Virtual Private Cloud.

Configuration du chiffrement TLS entre `slurmdbd` et le serveur de base de données

Avec la configuration de Slurm comptabilité par défaut qui AWS ParallelCluster fournit, `slurmdbd` établit une connexion cryptée TLS avec le serveur de base de données, si celui-ci prend en charge le chiffrement TLS. AWS les services de base de données tels qu'Amazon RDS Amazon Aurora prennent en charge le chiffrement TLS par défaut.

Vous pouvez exiger des connexions sécurisées côté serveur en définissant le `require_secure_transport` paramètre sur le serveur de base de données. Ceci est configuré dans le CloudFormation modèle fourni.

Conformément aux meilleures pratiques de sécurité, nous vous recommandons d'activer également la vérification de l'identité du serveur sur le `slurmdbd` client. Pour ce faire, configurez le [StorageParameters](#) dans les `slurmdbd.conf`. Téléchargez le certificat CA du serveur sur le nœud principal du cluster. Définissez ensuite l'option [SSL_CA](#) StorageParameters in `slurmdbd.conf` sur le chemin du certificat CA du serveur sur le nœud principal. Cela permet de vérifier l'identité du serveur sur le `slurmdbd` côté. Après avoir apporté ces modifications, redémarrez le `slurmdbd` service pour rétablir la connectivité au serveur de base de données avec la vérification d'identité activée.

Mise à jour des identifiants de base de données

Pour mettre à jour les valeurs [de Database/Username](#) ou [PasswordSecretArn](#), vous devez d'abord arrêter le parc informatique. Supposons que la valeur secrète stockée dans le AWS Secrets Manager secret soit modifiée et que son ARN ne soit pas modifié. Dans ce cas, le cluster ne met pas automatiquement à jour le mot de passe de la base de données à la nouvelle valeur. Pour mettre à jour le cluster en fonction de la nouvelle valeur secrète, exécutez la commande suivante depuis le nœud principal.

```
$ sudo /opt/parallelcluster/scripts/slurm/update_slurm_database_password.sh
```

Warning

Pour éviter de perdre des données comptables, nous vous recommandons de ne modifier le mot de passe de la base de données que lorsque le parc informatique est arrêté.

Surveillance des bases de données

Nous vous recommandons d'activer les fonctionnalités de surveillance des services de AWS base de données. Pour plus d'informations, consultez la documentation relative à la [surveillance Amazon RDS](#) ou à la documentation relative à la [surveillance Amazon Aurora](#).

Slurmpersonnalisation de la configuration

Commençant par AWS ParallelCluster version 3.6.0, vous pouvez personnaliser les `slurm.conf` Slurm configuration dans un AWS ParallelCluster configuration du cluster.

Dans la configuration du cluster, vous pouvez personnaliser Slurm paramètres de configuration à l'aide des paramètres de configuration de cluster suivants :

- Personnaliser Slurm paramètres pour l'ensemble du cluster en utilisant soit le [SlurmSettings/CustomSlurmSettings](#) ou le [CustomSlurmSettingsIncludeFile](#) paramètre. AWS ParallelCluster échoue si vous spécifiez les deux.
- Personnaliser Slurm paramètres d'une file d'attente en utilisant [SlurmQueues/CustomSlurmSettings](#) (mappé à `SlurmClosures`).
- Personnaliser Slurm paramètres d'une ressource de calcul en utilisant [SlurmQueues/ComputeResources/CustomSlurmSettings](#) (mappé à `SlurmNodes`).

Slurmconfiguration, personnalisation, limites et considérations lors de l'utilisationAWS ParallelCluster

- PourCustomSlurmSettingsetCustomSlurmSettingsIncludeFileparamètres, vous pouvez uniquement spécifier et mettre à jourslurm.confparamètres inclus dans[Slurmversion](#)qui est soutenu par leAWS ParallelClusterversion que vous utilisez pour configurer un cluster.
- Si vous spécifiez une option personnaliséeSlurmconfigurations dans l'un desCustomSlurmSettingsparamètres,AWS ParallelClustereffectue des contrôles de validation et empêche le réglage ou la mise à jourSlurmparamètres de configuration en conflit avecAWS ParallelClusterlogique. LeSlurmparamètres de configuration connus pour entrer en conflit avecAWS ParallelClustersont identifiés dans les listes de refus. Les listes de refus peuvent changer à l'avenirAWS ParallelClusterversions d'autresSlurmdes fonctionnalités sont ajoutées. Pour plus d'informations, veuillez consulter [Listé par DenySlurmparamètres de configuration pourCustomSlurmSettings](#).
- AWS ParallelClustervérifie uniquement si un paramètre figure dans une liste de refus.AWS ParallelClusterne valide pas votre personnalisationSlurmsyntaxe ou sémantique des paramètres de configuration. Vous êtes responsable de la validation de votre personnalisationSlurmparamètres de configuration. Personnalisation non valideSlurmls paramètres de configuration peuvent provoquerSlurmdéfaillances de daemon pouvant entraîner des échecs de création et de mise à jour de clusters.
- Si vous spécifiez une option personnaliséeSlurmconfigurations dansCustomSlurmSettingsIncludeFile,AWS ParallelClusterneeffectue aucune validation.
- Vous pouvez mettre à jourCustomSlurmSettingsetCustomSlurmSettingsIncludeFilesans arrêter ni démarrer le parc informatique. Dans ce cas,AWS ParallelClusterredémarre leslurmctlddaemon et exécute lescontrol reconfigurecommande.

CertainsSlurmls paramètres de configuration peuvent nécessiter différentes opérations avant qu'une modification ne soit enregistrée dans l'ensemble du cluster. Par exemple, ils peuvent nécessiter le redémarrage de tous les démons du cluster. Il est de votre responsabilité de vérifier siAWS ParallelClusterles opérations sont suffisantes pour propager votre personnalisationSlurmparamètres de configuration lors des mises à jour. Si tu trouves çaAWS ParallelClusterles opérations ne sont pas suffisantes, il est de votre responsabilité de fournir les actions supplémentaires nécessaires pour propager les paramètres mis à jour, comme recommandé dans le[Slurmdocumentation](#).

Listé par DenySlurmparamètres de configuration pourCustomSlurmSettings

Les tableaux suivants répertorient les paramètres avecAWS ParallelClusterversions qui refusent leur utilisation, à commencer par la version 3.6.0.CustomSlurmSettingsn'est pas pris en charge pourAWS ParallelClusterversions antérieures à la version 3.6.0.

Paramètres refusés au niveau du cluster :

| Slurm paramètre | Listé par Deny dansAWS ParallelClusterversions |
|-------------------------|--|
| CommunicationParameters | 3.6.0 |
| Epilogue | 3.6.0 |
| GresTypes | 3.6.0 |
| LaunchParameters | 3.6.0 |
| Prologue | 3.6.0 |
| ReconfigFlags | 3.6.0 |
| ResumeFailProgram | 3.6.0 |
| ResumeProgram | 3.6.0 |
| ResumeTimeout | 3.6.0 |
| SlurmctldHost | 3.6.0 |
| SlurmctldLogFile | 3.6.0 |
| SlurmctldParameters | 3.6.0 |
| SlurmdLogfile | 3.6.0 |
| SlurmUser | 3.6.0 |
| SuspendExcNodes | 3.6.0 |
| SuspendProgram | 3.6.0 |

| Slurm paramètre | Listé par Deny dansAWS ParallelClusterver sions |
|-----------------|--|
| SuspendTime | 3.6.0 |
| TaskPlugin | 3.6.0 |
| TreeWidth | 3.6.0 |

Refuser les paramètres listés au niveau du cluster lorsque [natifSlurmintégration de la comptabilité](#) est configuré dans la configuration du cluster :

| Slurm paramètre | Listé par Deny dansAWS ParallelClusterver sions |
|-----------------------|--|
| AccountingStorageType | 3.6.0 |
| AccountingStorageHost | 3.6.0 |
| AccountingStoragePort | 3.6.0 |
| AccountingStorageUser | 3.6.0 |
| JobAcctGatherType | 3.6.0 |

Paramètres refusés au niveau de la file d'attente (partition) pour les files d'attente gérées parAWS ParallelCluster:

| Slurm paramètre | Listé par Deny dansAWS ParallelClusterver sions |
|-----------------|--|
| Nœuds | 3.6.0 |
| PartitionName | 3.6.0 |
| ResumeTimeout | 3.6.0 |
| État | 3.6.0 |

| Slurm paramètre | Listé par Deny dansAWS ParallelClusterver sions |
|-----------------|--|
| SuspendTime | 3.6.0 |

Paramètres listés par refus au niveau de la ressource de calcul (nœud) pour les ressources de calcul gérées parAWS ParallelCluster:

| Slurm paramètre | Listé par Deny dansAWS ParallelClusterversion et versions ultérieures |
|-----------------|--|
| Processeurs | 3.6.0 |
| Fonctions | 3.6.0 |
| Gres | 3.6.0 |
| NodeAddr | 3.6.0 |
| NodeHostname | 3.6.0 |
| NodeName | 3.6.0 |
| Weight | 3.7.0 |

Slurm **prolog** et **epilog**

À partir de AWS ParallelCluster la version 3.6.0, la Slurm configuration déployée AWS ParallelCluster inclut les paramètres de Epilog configuration Prolog et :

```
# PROLOG AND EPILOG
Prolog=/opt/slurm/etc/scripts/prolog.d/*
Epilog=/opt/slurm/etc/scripts/epilog.d/*
SchedulerParameters=nohold_on_prolog_fail
BatchStartTimeout=180
```

Pour plus d'informations, consultez le [guide Prolog et Epilog](#) dans la Slurm documentation.

AWS ParallelCluster inclut les scripts prolog et epilog suivants :

- `90_plcluster_health_check_manager`(dans le Prolog dossier)
- `90_pcluster_noop`(dans le Epilog dossier)

Note

Le Epilog dossier Prolog et doit contenir au moins un fichier.

Vous pouvez utiliser vos propres scripts prolog ou vos propres epilog scripts en les ajoutant aux Epilog dossiers Prolog et correspondants.

Warning

Slurm exécute tous les scripts des dossiers, dans l'ordre alphabétique inverse.

La durée d'exécution des epilog scripts prolog et a un impact sur le temps nécessaire à l'exécution d'une tâche. Mettez à jour le paramètre `BatchStartTimeout` de configuration lors de l'exécution de plusieurs prolog scripts ou de longue durée. La valeur par défaut est de 3 minutes.

Si vous utilisez prolog des epilog scripts personnalisés, localisez-les dans les Epilog dossiers Prolog et correspondants. Nous vous recommandons de conserver le `90_plcluster_health_check_manager` script qui s'exécute avant chaque script personnalisé. Pour plus d'informations, voir [Slurmpersonnalisation de la configuration](#).

Taille et mise à jour de la capacité du cluster

La capacité du cluster est définie par le nombre de nœuds de calcul que le cluster peut dimensionner. Les nœuds de calcul sont soutenus par des instances Amazon EC2 définies dans les ressources de calcul de la AWS ParallelCluster configuration (`Scheduling/SlurmQueues/ComputeResources`), et sont organisés en files d'attente (`Scheduling/SlurmQueues`) qui mappent 1:1 aux partitions. Slurm

Au sein d'une ressource de calcul, il est possible de configurer le nombre minimum de nœuds de calcul (instances) qui doivent toujours continuer à fonctionner dans le cluster ([MinCount](#)), ainsi que le nombre maximum d'instances que la ressource de calcul peut atteindre ([MaxCount3](#)).

Au moment de la création du cluster, ou lors d'une mise à jour du cluster, AWS ParallelCluster lance autant d'instances Amazon EC2 que configuré `MinCount` pour chaque ressource de calcul

(`Scheduling/SlurmQueues/ ComputeResources`) définie dans le cluster. Les instances lancées pour couvrir le nombre minimal de nœuds pour les ressources de calcul du cluster sont appelées nœuds statiques. Une fois démarrés, les nœuds statiques sont censés être persistants dans le cluster et le système ne les arrête pas, sauf si un événement ou une condition spécifique se produit. Ces événements incluent, par exemple, l'échec des tests de Slurm santé d'Amazon EC2 et le changement de l'état du Slurm nœud en DRAIN ou DOWN.

Les instances Amazon EC2, de l'ordre de 1 à 'MaxCount - MinCount' (MaxCount moins MinCount), lancées à la demande pour faire face à l'augmentation de la charge du cluster, sont appelées nœuds dynamiques. Leur nature est éphémère, ils sont lancés pour exécuter des tâches en attente et sont interrompus une fois qu'ils restent inactifs pendant une période définie `Scheduling/SlurmSettings/ScaledownIdletime` dans la configuration du cluster (par défaut : 10 minutes).

Les nœuds statiques et les nœuds dynamiques sont conformes au schéma de dénomination suivant :

- Nœuds statiques `<Queue/Name>-st-<ComputeResource/Name>-<num>` où `<num> = 1..ComputeResource/MinCount`
- Nœuds dynamiques `<Queue/Name>-dy-<ComputeResource/Name>-<num>` où `<num> = 1..(ComputeResource/MaxCount - ComputeResource/MinCount)`

Par exemple, étant donné la AWS ParallelCluster configuration suivante :

```
Scheduling:
  Scheduler: Slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: c5xlarge
          Instances:
            - InstanceType: c5.xlarge
              MinCount: 100
              MaxCount: 150
```

Les nœuds suivants seront définis dans Slurm

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
```

```
queue1*      up    infinite    50  idle~ queue1-dy-c5xlarge-[1-50]
queue1*      up    infinite    100 idle queue1-st-c5xlarge-[1-100]
```

Lorsqu'une ressource de calcul l'estMinCount == MaxCount, tous les nœuds de calcul correspondants seront statiques et toutes les instances seront lancées au moment de la création/mise à jour du cluster et maintenues opérationnelles. Par exemple :

```
Scheduling:
  Scheduler: slurm
  SlurmQueues:
    - Name: queue1
      ComputeResources:
        - Name: c5xlarge
          Instances:
            - InstanceType: c5.xlarge
          MinCount: 100
          MaxCount: 100
```

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*      up    infinite    100  idle queue1-st-c5xlarge-[1-100]
```

Mise à jour des capacités du cluster

La mise à jour de la capacité du cluster inclut l'ajout ou la suppression de files d'attente, de ressources de calcul ou la modification MinCount/MaxCount d'une ressource de calcul. À partir de AWS ParallelCluster la version 3.9.0, la réduction de la taille d'une file d'attente nécessite que le parc de calcul soit arrêté ou [QueueUpdateStrategy](#) défini sur TERMINATE avant qu'une mise à jour du cluster n'ait lieu. Il n'est pas nécessaire d'arrêter le parc informatique ou de le [QueueUpdateStrategy](#) configurer sur TERMINATE lorsque :

- Ajouter de nouvelles files d'attente à la planification/ [SlurmQueues](#)
- Ajouter de nouvelles ressources de calcul Scheduling/SlurmQueues/[ComputeResources](#) à une file d'attente

- Augmenter la valeur [MaxCount](#) d'une ressource informatique
- Augmentation MinCount d'une ressource de calcul et augmentation MaxCount de la même ressource de calcul d'au moins la même quantité

Considérations et restrictions

Cette section vise à décrire tous les facteurs, contraintes ou limitations importants à prendre en compte lors du redimensionnement de la capacité du cluster.

- Lors de la suppression d'une file d'attente de Scheduling/<https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling-v3.html#Scheduling-v3-SlurmQueues> SlurmQueues tous les nœuds de calcul portant un nom `<Queue/Name>-*`, statique ou dynamique, seront supprimés de la Slurm configuration et les instances Amazon EC2 correspondantes seront résiliées.
- Lorsque vous supprimez une ressource Scheduling/SlurmQueues/<https://docs.aws.amazon.com/parallelcluster/latest/ug/Scheduling-v3.html#Scheduling-v3-SlurmQueues-ComputeResources> ComputeResources de calcul d'une file d'attente, tous les nœuds de calcul portant un nom `<Queue/Name>-*` - `<ComputeResource/Name>-*`, qu'ils soient statiques ou dynamiques, sont supprimés de la Slurm configuration et les instances Amazon EC2 correspondantes sont mises hors service.

Lorsque vous modifiez le MinCount paramètre d'une ressource de calcul, nous pouvons distinguer deux scénarios différents, s'il MaxCount est maintenu égal à MinCount (capacité statique uniquement) et s'il MaxCount est supérieur à MinCount (capacité statique et dynamique mixte).

Changements de capacité avec des nœuds statiques uniquement

- Si `MinCount == MaxCount`, lors de l'augmentation MinCount (etMaxCount), le cluster est configuré en étendant le nombre de nœuds statiques à la nouvelle valeur de MinCount `<Queue/Name>-st-<ComputeResource/Name>-<new_MinCount>` et que le système continue d'essayer de lancer des instances Amazon EC2 pour atteindre la nouvelle capacité statique requise.
- Si `MinCount == MaxCount`, lors de la diminution MinCount (etMaxCount) du montant N, le cluster est configuré en supprimant les N derniers nœuds statiques `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount - N>...<old_MinCount>`] et le système met fin aux instances Amazon EC2 correspondantes.

- État initial $\text{MinCount} = \text{MaxCount} = 100$

-

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up        infinite   100    idle queue1-st-c5xlarge-[1-100]
```

- Mise à jour -30 sur MinCount et MaxCount : $\text{MinCount} = \text{MaxCount} = 70$

-

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up        infinite    70    idle queue1-st-c5xlarge-[1-70]
```

Changements de capacité avec des nœuds mixtes

Si $\text{MinCount} < \text{MaxCount}$, lors d'une augmentation MinCount d'un montant N (en supposant que MaxCount cela restera inchangé), le cluster sera configuré en étendant le nombre de nœuds statiques à la nouvelle valeur de MinCount ($\text{old_MinCount} + N$): `<Queue/Name>-st-<ComputeResource/Name>-<old_MinCount + N>` et le système continuera d'essayer de lancer des instances Amazon EC2 pour atteindre la nouvelle capacité statique requise. De plus, pour respecter la MaxCount capacité de la ressource de calcul, la configuration du cluster est mise à jour en supprimant les N derniers nœuds dynamiques : `<Queue/Name>-dy-<ComputeResource/Name>-[<MaxCount - old_MinCount - N>...<MaxCount - old_MinCount>]` et le système mettra fin aux instances Amazon EC2 correspondantes.

- État initial : $\text{MinCount} = 100$; $\text{MaxCount} = 150$

-

```
$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up        infinite    50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up        infinite   100    idle queue1-st-c5xlarge-[1-100]
```

- Mettre à jour +30 vers MinCount : $\text{MinCount} = 130$ ($\text{MaxCount} = 150$)

- ```

$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 20 idle~ queue1-dy-c5xlarge-[1-20]
queue1* up infinite 130 idle queue1-st-c5xlarge-[1-130]

```

Si  $\text{MinCount} < \text{MaxCount}$ , lors de l'augmentation  $\text{MinCount}$  et  $\text{MaxCount}$  de la même quantité  $N$ , le cluster est configuré en étendant le nombre de nœuds statiques à la nouvelle valeur de  $\text{MinCount}$  ( $\text{old\_MinCount} + N$ ): `<Queue/Name>-st-<ComputeResource/Name>-<old\_MinCount + N>` et le système continuera d'essayer de lancer des instances Amazon EC2 pour atteindre la nouvelle capacité statique requise. De plus, aucune modification ne sera apportée au nombre de nœuds dynamiques pour honorer le nouveau

Valeur  $\text{MaxCount}$ .

- État initial :  $\text{MinCount} = 100$ ;  $\text{MaxCount} = 150$

- ```

$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
queue1*    up    infinite   50    idle~ queue1-dy-c5xlarge-[1-50]
queue1*    up    infinite  100    idle queue1-st-c5xlarge-[1-100]

```

- Mettre à jour +30 vers MinCount : $\text{MinCount} = 130$ ($\text{MaxCount} = 180$)

- ```

$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 20 idle~ queue1-dy-c5xlarge-[1-50]
queue1* up infinite 130 idle queue1-st-c5xlarge-[1-130]

```

Si  $\text{MinCount} < \text{MaxCount}$ , lors de la diminution  $\text{MinCount}$  du montant  $N$  (en supposant qu'il  $\text{MaxCount}$  restera inchangé), le cluster sera configuré en supprimant les  $N$  derniers nœuds statiques (nœuds statiques) `<Queue/Name>-st-<ComputeResource/Name>-[<old\_MinCount -`

$N > \dots <old\_MinCount>$  et le système mettra fin aux instances Amazon EC2 correspondantes. De plus, pour respecter la MaxCount capacité de la ressource de calcul, la configuration du cluster est mise à jour en augmentant le nombre de nœuds dynamiques afin de combler le vide. MaxCount - new\_MinCount :  $<Queue/Name>-dy-<ComputeResource/Name>-[1..<MaxCount - new\_MinCount>]$  Dans ce cas, comme il s'agit de nœuds dynamiques, aucune nouvelle instance Amazon EC2 ne sera lancée à moins que le planificateur n'ait des tâches en attente sur les nouveaux nœuds.

- État initial : MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 50 idle~ queue1-dy-c5xlarge-[1-50]
queue1* up infinite 100 idle queue1-st-c5xlarge-[1-100]
```

- Mise à jour -30 le MinCount : MinCount = 70 (MaxCount = 120)

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 80 idle~ queue1-dy-c5xlarge-[1-80]
queue1* up infinite 70 idle queue1-st-c5xlarge-[1-70]
```

Si  $MinCount < MaxCount$ , lors de la diminution MinCount et MaxCount de la même quantité  $N$ , le cluster sera configuré en supprimant les  $N$  derniers nœuds statiques  $<Queue/Name>-st-<ComputeResource/Name>-<old\_MinCount - N> \dots <oldMinCount>$ ] et le système mettra fin aux instances Amazon EC2 correspondantes.

De plus, aucune modification ne sera apportée au nombre de nœuds dynamiques pour respecter la nouvelle MaxCount valeur.

- État initial : MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
```

```
queue1* up infinite 50 idle~ queue1-dy-c5xlarge-[1-50]
queue1* up infinite 100 idle queue1-st-c5xlarge-[1-100]
```

- Mise à jour -30 le MinCount : MinCount = 70 (MaxCount = 120)

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 80 idle~ queue1-dy-c5xlarge-[1-50]
queue1* up infinite 70 idle queue1-st-c5xlarge-[1-70]
```

Si  $\text{MinCount} < \text{MaxCount}$ , lors de la diminution  $\text{MaxCount}$  du montant  $N$  (en supposant  $\text{MinCount}$  qu'il reste inchangé), le cluster est configuré en supprimant les  $N$  derniers nœuds dynamiques `<Queue/Name>-dy-<ComputeResource/Name>-<old_MaxCount - N...<oldMaxCount>]` et le système arrête les instances Amazon EC2 correspondantes dans le cas où elles étaient en cours d'exécution. Aucun impact n'est attendu sur les nœuds statiques.

- État initial : MinCount = 100; MaxCount = 150

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 50 idle~ queue1-dy-c5xlarge-[1-50]
queue1* up infinite 100 idle queue1-st-c5xlarge-[1-100]
```

- Mise à jour -30 le MaxCount : MinCount = 100 (MaxCount = 120)

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 20 idle~ queue1-dy-c5xlarge-[1-20]
queue1* up infinite 100 idle queue1-st-c5xlarge-[1-100]
```

## Impacts sur les emplois

Dans tous les cas où des nœuds sont supprimés et des instances Amazon EC2 arrêtées, une tâche sbatch exécutée sur les nœuds supprimés sera remise en file d'attente, sauf si aucun autre nœud ne répond aux exigences de la tâche. Dans ce dernier cas, la tâche échouera avec le statut `NODE_FAIL` et disparaîtra de la file d'attente ; dans ce cas, elle devra être soumise à nouveau manuellement.

Si vous prévoyez d'effectuer une mise à jour de redimensionnement du cluster, vous pouvez empêcher les tâches de s'exécuter sur les nœuds qui seront supprimés lors de la mise à jour planifiée. Cela est possible en configurant les nœuds à supprimer lors de la maintenance. Sachez que le fait de configurer un nœud en maintenance n'aura aucune incidence sur les tâches qui sont déjà en cours d'exécution sur le nœud.

Supposons qu'avec la mise à jour prévue pour le redimensionnement du cluster, vous allez supprimer le `queue-st-computeresource-[9-10 [nœud]]`. Vous pouvez créer une Slurm réservation avec la commande suivante

```
sudo -i scontrol create reservation ReservationName=maint_for_update user=root
starttime=now duration=infinite flags=maint,ignore_jobs nodes=queue-st-
computeresource-[9-10]
```

Cela créera une Slurm réservation nommée `maint_for_update` sur les nœuds `queue-st-computeresource-[9-10]`. À partir du moment où la réservation est créée, aucune autre tâche ne peut être exécutée sur les nœuds `queue-st-computeresource-[9-10]`. Sachez que la réservation n'empêchera pas l'attribution éventuelle de tâches sur les nœuds `queue-st-computeresource-[9-10]`.

Après la mise à jour du redimensionnement du cluster, si la Slurm réservation a été définie uniquement sur les nœuds supprimés lors de la mise à jour du redimensionnement, la réservation de maintenance sera automatiquement supprimée. Si vous avez plutôt créé une Slurm réservation sur les nœuds qui sont toujours présents après la mise à jour du redimensionnement du cluster, nous souhaiterons peut-être supprimer la réservation de maintenance sur les nœuds une fois la mise à jour du redimensionnement effectuée, à l'aide de la commande suivante

```
sudo -i scontrol delete ReservationName=maint_for_update
```

[Pour plus de détails sur la Slurm réservation, consultez le document officiel de SchedMD ici.](#)

## Processus de mise à jour du cluster en cas de modification de capacité

Lors d'un changement de configuration du planificateur, les étapes suivantes sont exécutées pendant le processus de mise à jour du cluster :

- Arrêtez AWS ParallelCluster `clustermgtd` (`supervisorctl stop clustermgtd`)
- Générer une configuration de Slurm partitions mise à jour depuis AWS ParallelCluster la configuration
- Redémarrage `slurmctld` (effectué via la recette du service Chef)
- Vérifier le `slurmctld` statut (`systemctl is-active --quiet slurmctld.service`)
- Recharger la configuration Slurm (`scontrol reconfigure`)
- Démarrage de `clustermgtd` (`supervisorctl start clustermgtd`)

## AWS Batch (**awsbatch**)

Pour de plus amples informations sur AWS Batch, consultez [AWS Batch](#). Pour la documentation, consultez le [Guide de AWS Batch l'utilisateur](#).

### AWS ParallelClusterCommandes CLI pour AWS Batch

Lorsque vous utilisez le `awsbatch` planificateur, les commandes AWS ParallelCluster CLI pour AWS Batch sont automatiquement installées dans le nœud principalAWS ParallelCluster. La CLI utilise des opérations AWS Batch d'API et autorise les opérations suivantes :

- Envoyer et gérer les tâches.
- Surveiller les tâches, les files d'attente et les hôtes.
- Mettre en miroir les commandes traditionnelles du planificateur.

#### Important

AWS ParallelClusterne prend pas en charge les tâches GPU pourAWS Batch. Pour plus d'informations, consultez la section [Tâches GPU](#).

Cette interface de ligne de commande est distribuée sous la forme d'un package distinct. Pour plus d'informations, veuillez consulter [Support pour les planificateurs](#).

## Rubriques

- [awsbsub](#)
- [awsbstat](#)
- [awsbout](#)
- [awsbkill](#)
- [awsbqueues](#)
- [awsbhosts](#)

## awsbsub

Soumet les tâches à la file d'attente des tâches du cluster.

```
awsbsub [-h] [-jn JOB_NAME] [-c CLUSTER] [-cf] [-w WORKING_DIR]
 [-pw PARENT_WORKING_DIR] [-if INPUT_FILE] [-p VCPUS] [-m MEMORY]
 [-e ENV] [-eb ENV_DENYLIST] [-r RETRY_ATTEMPTS] [-t TIMEOUT]
 [-n NODES] [-a ARRAY_SIZE] [-d DEPENDS_ON]
 [command] [arguments [arguments ...]]
```

### Important

AWS ParallelCluster ne prend pas en charge les tâches GPU pour AWS Batch. Pour plus d'informations, consultez la section [Tâches GPU](#).

## Arguments positionnels

### ***command***

Soumet la tâche (la commande spécifiée doit être disponible sur les instances de calcul) ou le nom du fichier à transférer. Voir aussi `--command-file`.

### **arguments**

(Facultatif) Spécifie les arguments de la commande ou du fichier de commandes.

## Arguments nommés

**-jn *JOB\_NAME*, --job-name *JOB\_NAME***

Nom de la tâche. Le premier caractère doit être une lettre ou un chiffre. Le nom du travail peut contenir des lettres (majuscules et minuscules), des chiffres, des traits d'union et des traits de soulignement, et comporter jusqu'à 128 caractères.

**-c *CLUSTER*, --cluster *CLUSTER***

Spécifie le cluster à utiliser.

**-cf, --command-file**

Indique que la commande est un fichier à transférer aux instances de calcul.

Par défaut : false

**-w *WORKING\_DIR*, --working-dir *WORKING\_DIR***

Spécifie le dossier à utiliser en tant que répertoire de travail des tâches. Si aucun répertoire de travail n'est spécifié, la tâche est exécutée dans le job-*<AWS\_BATCH\_JOB\_ID>* sous-dossier du répertoire personnel de l'utilisateur. Vous pouvez utiliser ce paramètre ou le paramètre `--parent-working-dir`.

**-pw *PARENT\_WORKING\_DIR*, --parent-working-dir *PARENT\_WORKING\_DIR***

Spécifie le dossier parent du répertoire de travail de la tâche. Si aucun répertoire de travail parent n'est spécifié, il s'agit par défaut du répertoire personnel de l'utilisateur. Un sous-dossier nommé job-*<AWS\_BATCH\_JOB\_ID>* est créé dans le répertoire de travail parent. Vous pouvez utiliser ce paramètre ou le paramètre `--working-dir`.

**-if *INPUT\_FILE*, --input-file *INPUT\_FILE***

Spécifie le fichier à transférer vers les instances de calcul, dans le répertoire de travail de la tâche. Vous pouvez spécifier plusieurs paramètres de fichiers d'entrée.

**-p *VCPUS*, --vcpus *VCPUS***

Spécifie le nombre de processeurs virtuels à réserver pour le conteneur. Lorsqu'il est utilisé conjointement avec `--nodes`, il identifie le nombre de vCPU pour chaque nœud.

Valeur par défaut : 1



**-m *MEMORY*, --memory *MEMORY***

Spécifie la limite stricte de la mémoire (en Mio) à fournir pour la tâche. Si votre tâche tente de dépasser la limite de mémoire spécifiée ici, elle est terminée.

Valeur par défaut : 128

**-e *ENV*, --env *ENV***

Spécifie une liste de noms de variables d'environnement séparés par des virgules à exporter vers l'environnement de la tâche. Pour exporter toutes les variables d'environnement, spécifiez « all ». Notez qu'une liste de « toutes » les variables d'environnement n'inclut pas celles répertoriées dans le `--env-blacklist` paramètre, ni les variables commençant par le `AWS_*` préfixe `PCLUSTER_*` ou.

**-eb *ENV\_DENYLIST*, --env-blacklist *ENV\_DENYLIST***

Spécifie une liste de noms de variable d'environnement séparés par des virgules à ne pas exporter vers l'environnement de la tâche. Par défaut, HOME, PWD, USER, PATH, LD\_LIBRARY\_PATH, TERM et TERMCAP ne sont pas exportées.

**-r *RETRY\_ATTEMPTS*, --retry-attempts *RETRY\_ATTEMPTS***

Spécifie le nombre de fois qu'une tâche doit passer au `RUNNABLE` statut. Vous pouvez indiquer entre 1 et 10 tentatives. Si la valeur des tentatives est supérieure à 1, la tâche est réessayée en cas d'échec, jusqu'à ce qu'elle passe à un `RUNNABLE` statut pour le nombre de fois spécifié.

Valeur par défaut : 1

**-t *TIMEOUT*, --timeout *TIMEOUT***

Spécifie la durée en secondes (mesurée à partir de `startedAt` horodatage de la tentative de tâche) après laquelle votre tâche s' AWS Batch arrête si elle n'est pas terminée. La valeur du délai d'expiration doit être au moins égal à 60 secondes.

**-n *NODES*, --nodes *NODES***

Spécifie le nombre de nœuds à réserver pour la tâche. Spécifiez une valeur pour ce paramètre afin d'activer la soumission parallèle sur plusieurs nœuds.

**Note**

Lorsque le `CapacityType` paramètre `Scheduler/AwsBatchQueues` est défini sur `SPOT`, les tâches parallèles à nœuds multiples ne sont pas prises en charge. En outre, votre

compte doit comporter un rôle `AWSServiceRoleForEC2Spot` lié au service. Vous pouvez créer ce rôle à l'aide de la AWS CLI commande suivante :

```
$ aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Pour plus d'informations, consultez la section [Rôle lié au service pour les demandes d'instance Spot](#) dans le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux.

### **-a *ARRAY\_SIZE*, --array-size *ARRAY\_SIZE***

Indique la taille du tableau. Vous pouvez spécifier une valeur comprise entre 2 et 10 000. Si vous spécifiez les propriétés de tableau pour une tâche, elle devient une tâche de tableau.

### **-d *DEPENDS\_ON*, --depends-on *DEPENDS\_ON***

Spécifie une liste de dépendances séparées par un point-virgule pour une tâche. Une tâche peut compter jusqu'à 20 tâches au plus. Vous pouvez spécifier une dépendance `SEQUENTIAL` de type sans spécifier d'ID de tâche pour les tâches matricielles. Une dépendance séquentielle permet à chaque tâche de tableau enfant de se terminer de manière séquentielle, à partir de l'index 0. Vous pouvez également spécifier une dépendance de type `N_TO_N` avec un ID de tâche pour les tâches de tableau. Une dépendance `N_TO_N` signifie que 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 syntaxe de ce paramètre est « `jobID=<string>, type= <string>;...` ».

## **awsbstat**

Affiche les tâches soumises dans la file d'attente des tâches du cluster.

```
awsbstat [-h] [-c CLUSTER] [-s STATUS] [-e] [-d] [job_ids [job_ids ...]]
```

Arguments positionnels

### ***job\_ids***

Spécifie la liste des ID de tâche séparés par un espace à afficher dans la sortie. Si la tâche est un tableau de tâches, toutes les tâches enfants sont affichées. Si une tâche unique est demandée, elle apparaît dans une version détaillée.

## Arguments nommés

**-c *CLUSTER*, --cluster *CLUSTER***

Indique le cluster à utiliser.

**-s *STATUS*, --status *STATUS***

Spécifie la liste des statuts de tâche séparés par une virgule à inclure. Par défaut, le statut de la tâche est « active ». Les valeurs acceptées sont : SUBMITTED, PENDING, RUNNABLE, STARTING, RUNNING, SUCCEEDED, FAILED et ALL.

Par défaut : « SUBMITTED,PENDING,RUNNABLE,STARTING,RUNNING »

**-e, --expand-children**

Étend les tâches avec enfants (tableau et parallèle à plusieurs nœuds).

Par défaut : false

**-d, --details**

Affiche les détails des tâches.

Par défaut : false

## awsbout

Affiche la sortie d'une tâche donnée.

```
awsbout [-h] [-c CLUSTER] [-hd HEAD] [-t TAIL] [-s] [-sp STREAM_PERIOD] job_id
```

## Arguments positionnels

***job\_id***

Spécifie l'ID de tâche.

## Arguments nommés

**-c *CLUSTER*, --cluster *CLUSTER***

Indique le cluster à utiliser.

**-hd *HEAD*, --head *HEAD***

Permet d'obtenir les premières lignes *HEAD* de la sortie de tâche.

**-t *TAIL*, --tail *TAIL***

Permet d'obtenir les dernières lignes <tail> de la sortie de tâche.

**-s, --stream**

Permet d'obtenir la sortie de tâche, puis attend qu'une sortie supplémentaire soit produite. Cet argument peut être utilisé avec -tail pour démarrer à partir des dernières lignes <tail> de la sortie de tâche.

Par défaut : false

**-sp *STREAM\_PERIOD*, --stream-period *STREAM\_PERIOD***

Définit la période de streaming.

Par défaut: 5

## awsbkill

Annule ou met fin à des tâches soumises dans le cluster.

```
awsbkill [-h] [-c CLUSTER] [-r REASON] job_ids [job_ids ...]
```

Arguments positionnels

***job\_ids***

Spécifie la liste des ID de tâche séparés par un espace à annuler ou à terminer.

Arguments nommés

**-c *CLUSTER*, --cluster *CLUSTER***

Indique le nom du cluster à utiliser.

**-r *REASON*, --reason *REASON***

Indique le message que vous souhaitez attacher à une tâche, en expliquant la raison de l'annulation.

Par défaut : « Terminated by the user »

## awsbqueues

Affiche la file d'attente de tâches associée au cluster.

```
awsbqueues [-h] [-c CLUSTER] [-d] [job_queues [job_queues ...]]
```

Arguments positionnels

### *job\_queues*

Spécifie la liste des noms de file d'attente séparés par des espaces à afficher. Si une seule file d'attente est demandée, elle est affichée dans une version détaillée.

Arguments de nom

**-c *CLUSTER*, --cluster *CLUSTER***

Spécifie le nom du cluster à utiliser.

**-d, --details**

Indique s'il convient d'afficher les détails des files d'attente.

Par défaut : false

## awsbhosts

Affiche les hôtes qui appartiennent à l'environnement de calcul du cluster.

```
awsbhosts [-h] [-c CLUSTER] [-d] [instance_ids [instance_ids ...]]
```

Arguments positionnels

### *instance\_ids*

Spécifie une liste séparée par des espaces d'ID d'instances. Si une seule instance est demandée, elle apparaît dans une version détaillée.

## Arguments nommés

**-c *CLUSTER*, --cluster *CLUSTER***

Spécifie le nom du cluster à utiliser.

**-d, --details**

Indique s'il convient d'afficher les détails des hôtes.

Par défaut : false

## Stockage partagé

AWS ParallelCluster [prend en charge l'utilisation d'Amazon EBS, FSx for ONTAP et FSx pour les volumes de stockage partagés OpenZFS, d'Amazon EFS et FSx pour les systèmes de fichiers de stockage partagé Lustre ou de caches de fichiers](#). Nous vous recommandons de suivre les directives du [pilier de fiabilité AWS du framework bien conçu](#) et de sauvegarder vos volumes et vos systèmes de fichiers.

Sélectionnez un système de stockage qui répond aux exigences d'E/S de votre application HPC. Vous pouvez optimiser chaque système de fichiers en fonction de votre cas d'utilisation spécifique. Pour plus d'informations, consultez la [présentation des options de stockage](#).

Les volumes Amazon EBS sont attachés au nœud principal et partagés avec les nœuds de calcul via NFS. Cette option peut être rentable, mais les performances dépendent des ressources du nœud principal à mesure que les besoins de stockage évoluent. Cela peut devenir un goulot d'étranglement à mesure que de nouveaux nœuds de calcul sont ajoutés au cluster et que la demande de débit augmente.

Les systèmes de fichiers Amazon EFS évoluent en fonction de l'évolution des besoins de stockage. Vous pouvez configurer ces systèmes de fichiers pour différents cas d'utilisation. Utilisez les systèmes de fichiers Amazon EFS pour exécuter des applications parallélisées et sensibles à la latence sur votre cluster.

Les systèmes de fichiers FSx for Lustre peuvent traiter des ensembles de données massifs avec un débit pouvant atteindre des centaines de gigaoctets par seconde, des millions d'IOPS et des latences inférieures à la milliseconde. Utilisez les systèmes de fichiers FSx for Lustre pour les environnements de calcul exigeants à hautes performances.

Dans le [SharedStorage Section](#), vous pouvez définir un stockage externe ou AWS ParallelCluster géré :

- Le stockage externe fait référence à un volume ou à un système de fichiers existant que vous gérez. AWS ParallelCluster ne crée ni ne supprime ce stockage.
- Le stockage géré fait référence à un volume ou à un système de fichiers AWS ParallelCluster créé et pouvant être supprimé.

## Stockage externe

Vous pouvez configurer AWS ParallelCluster pour associer un stockage externe au cluster lors de sa création ou de sa mise à jour. De même, vous pouvez le configurer pour détacher le stockage externe du cluster lorsque celui-ci est supprimé ou mis à jour. Vos données sont préservées et vous pouvez les utiliser pour un stockage partagé permanent à long terme en dehors du cycle de vie du cluster.

### Note

Les versions AWS ParallelCluster antérieures à 3.8 n'autorisent pas le montage de systèmes de fichiers gérés en externe. /home À partir de la version 3.8 AWS ParallelCluster , vous pouvez l'utiliser /home comme point de montage pour un système de fichiers géré externe. Vous pouvez monter un système de fichiers géré en externe /home en spécifiant /home comme valeur le [MountDir](#) paramètre situé sous le [SharedStorage Section](#).

Amazon File Cache ne peut pas être utilisé comme /home répertoire système et n'est donc pas pris en charge pour le moment pour le montage /home.

Lorsque vous spécifiez un /home répertoire sous [SharedStorage Section](#) l'option [SharedStorageType](#) de configuration, celle-ci sera remplacée, ce qui signifie que les paramètres ci-dessous [SharedStorage Section](#) seront utilisés à la place.

Lorsque vous montez un système de fichiers externe /home dans le répertoire AWS ParallelCluster , le /home contenu du nœud principal est copié sur le système de fichiers externe, sans remplacer les fichiers existants sur le stockage externe. Cela inclut le transfert de la clé SSH du cluster pour l'utilisateur par défaut, si elle est absente du système de fichiers externe. Pour plus d'informations, reportez-vous à [AWS ParallelCluster considérations relatives au stockage partagé](#).

## AWS ParallelCluster stockage géré

AWS ParallelCluster le stockage géré dépend du cycle de vie du cluster par défaut dans la configuration. Le paramètre `SharedStorage DeletionPolicy` de configuration est défini sur `Delete` par défaut.

Par défaut, un système de fichiers ou un volume AWS ParallelCluster géré et ses données sont supprimés si l'une des conditions suivantes est vraie.

- Vous supprimez le cluster.
- Vous modifiez la configuration du stockage partagé géréName.
- Vous supprimez le stockage partagé géré de la configuration.

Réglez `DeletionPolicy` sur `Retain` pour conserver votre système de fichiers partagé géré ou votre volume et vos données. Nous vous recommandons de sauvegarder régulièrement vos données afin d'éviter toute perte de données. Vous pouvez l'utiliser [AWS Backup](#) pour gérer de manière centralisée les sauvegardes de toutes vos options de stockage.

Vous pouvez supprimer la dépendance au cycle de vie à l'aide des paramètres de configuration. Pour plus d'informations, consultez [Convertir le stockage AWS ParallelCluster géré en stockage externe](#).

Pour plus d'informations sur les quotas de stockage partagés, consultez [Quotas pour le stockage partagé](#).

Pour plus d'informations sur le stockage partagé et le passage aux nouvelles AWS ParallelCluster versions, consultez [Bonnes pratiques : déplacement d'un cluster vers une nouvelle version AWS ParallelCluster mineure ou un correctif](#).

Vous pouvez configurer AWS ParallelCluster pour associer un stockage externe au cluster lors de sa création ou de sa mise à jour. De même, vous pouvez le configurer pour détacher le stockage externe du cluster lorsque celui-ci est supprimé ou mis à jour. Vos données sont préservées et vous pouvez les utiliser pour des solutions de stockage partagé permanent à long terme, indépendantes du cycle de vie du cluster.

Par défaut, le stockage géré dépend du cycle de vie du cluster. Vous pouvez supprimer la dépendance à l'aide des paramètres de configuration décrits dans [Convertir le stockage AWS ParallelCluster géré en stockage externe](#).

Avec des paramètres spécifiques, vous pouvez optimiser chacune des solutions de stockage prises en charge en fonction de vos cas d'utilisation.



Pour les quotas de stockage partagés, voir [Quotas pour le stockage partagé](#).

Pour plus d'informations sur le stockage partagé et le passage aux nouvelles AWS ParallelCluster versions, consultez [Bonnes pratiques : déplacement d'un cluster vers une nouvelle version AWS ParallelCluster mineure ou un correctif](#).

## Rubriques

- [Configuration du stockage partagé](#)
- [Utilisation du stockage partagé dans AWS ParallelCluster](#)
- [Quotas pour le stockage partagé](#)

## Configuration du stockage partagé

Découvrez les paramètres de configuration que vous pouvez utiliser pour définir le stockage partagé pour votre cluster.

## Rubriques

- [Amazon Elastic Block Store](#)
- [Amazon Elastic File System](#)
- [Amazon FSx pour Lustre](#)
- [Configuration de FSx pour ONTAP, de FSx pour OpenZFS et du stockage partagé File Cache](#)

## Amazon Elastic Block Store

Pour utiliser un volume Amazon EBS externe existant pour un stockage permanent à long terme indépendant du cycle de vie du cluster, spécifiez [EbsSettings/VolumeId](#).

Si vous ne le spécifiez pas [VolumeId](#), AWS ParallelCluster crée par défaut un volume EBS géré à partir de [EbsSettings](#) la création de votre cluster. AWS ParallelCluster supprime également le volume et les données lorsque le cluster est supprimé ou que le volume est retiré de la configuration du cluster.

Pour un volume EBS AWS ParallelCluster géré, vous pouvez utiliser [EbsSettings/DeletionPolicy](#) pour donner des instructions AWS ParallelCluster à `DeleteRetain`, ou utiliser Snapshot le volume lorsque le cluster est supprimé ou lorsque le volume est retiré de la configuration du cluster. Par défaut, la propriété `DeletionPolicy` a la valeur `Delete`.

**⚠ Warning**

Pour le stockage partagé AWS ParallelCluster géré, `DeletionPolicy` est défini sur `Delete` par défaut.

Cela signifie que, si l'une des conditions suivantes est vraie, un volume géré et ses données sont supprimés :

- Vous supprimez le cluster.
- Vous modifiez la configuration du stockage partagé géré [SharedStorage/Name](#).
- Vous supprimez le stockage partagé géré de la configuration.

Nous vous recommandons de sauvegarder régulièrement votre stockage partagé avec des instantanés afin d'éviter toute perte de données. Pour plus d'informations sur les instantanés Amazon EBS, consultez les instantanés [Amazon EBS dans](#) le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux. Pour savoir comment gérer les sauvegardes de données entre elles services AWS, consultez la section [AWS Sauvegarde](#) du manuel du AWS Backup développeur.

## Amazon Elastic File System

Pour utiliser un système de fichiers Amazon EFS externe existant pour un stockage permanent à long terme en dehors du cycle de vie du cluster, spécifiez [EfsSettings/FileSystemId](#), par défaut, AWS ParallelCluster crée un système de fichiers Amazon EFS géré à partir du [EfsSettings](#) moment où il crée le cluster. AWS ParallelCluster supprime également le système de fichiers et les données lorsque le cluster est supprimé ou lorsque le système de fichiers est supprimé de la configuration du cluster.

Pour un système de fichiers Amazon EFS AWS ParallelCluster géré, vous pouvez utiliser le [EfsSettings/DeletionPolicy](#) pour indiquer AWS ParallelCluster le moment où le cluster est supprimé ou lorsque le système de fichiers est retiré de la configuration du cluster. `Delete Retain` Par défaut, la propriété `DeletionPolicy` a la valeur `Delete`.

**⚠ Warning**

Pour le stockage partagé AWS ParallelCluster géré, `DeletionPolicy` est défini sur `Delete` par défaut.

Cela signifie que, si l'une des conditions suivantes est vraie, un système de fichiers géré et ses données sont supprimés :

- Vous supprimez le cluster.
- Vous modifiez la configuration du stockage partagé géré [SharedStorage/Name](#).
- Vous supprimez le stockage partagé géré de la configuration.

Nous vous recommandons de sauvegarder régulièrement votre espace de stockage partagé afin d'éviter toute perte de données. Pour plus d'informations sur la façon de sauvegarder des volumes Amazon EFS individuels, consultez la section [Sauvegarde de vos systèmes de fichiers Amazon EFS](#) dans le manuel Amazon Elastic File System User Guide. Pour savoir comment gérer les sauvegardes de données entre elles services AWS, consultez la section [AWS Sauvegarde](#) du manuel du AWS Backup développeur.

## Amazon FSx pour Lustre

Pour utiliser un système de fichiers externe FSx for Lustre existant pour un stockage permanent à long terme en dehors du cycle de vie du cluster, [FsxLustreSettings](#) spécifiez [FileSystemId](#).

Si vous ne spécifiez pas [FsxLustreSettings/FileSystemId](#), AWS ParallelCluster crée par défaut un système de fichiers géré FSx for Lustre [FsxLustreSettings](#) à partir de la création du cluster. AWS ParallelCluster supprime également le système de fichiers et les données lorsque le cluster est supprimé ou lorsque le système de fichiers est supprimé de la configuration du cluster.

Pour un système de fichiers FSx for Lustre AWS ParallelCluster géré, vous pouvez utiliser [FsxLustreSettings](#) le [DeletionPolicy](#) pour AWS ParallelCluster indiquer Delete au Retain système de fichiers que le cluster est supprimé ou lorsque le système de fichiers est supprimé de la configuration du cluster. Par défaut, la propriété DeletionPolicy a la valeur Delete.

### Warning

Pour le stockage partagé AWS ParallelCluster géré, DeletionPolicy est défini sur Delete par défaut.

Cela signifie que, si l'une des conditions suivantes est vraie, un système de fichiers géré et ses données sont supprimés :

- Vous supprimez le cluster.

- Vous modifiez la configuration du stockage partagé géré [SharedStorage/Name](#).
- Vous supprimez le stockage partagé géré de la configuration.

Nous vous recommandons de sauvegarder régulièrement votre espace de stockage partagé afin d'éviter toute perte de données. Vous pouvez définir des sauvegardes dans votre cluster avec [SharedStorage/FsxLustreSettings/AutomaticBackupRetentionDays](#) et [DailyAutomaticBackupStartTime](#). Pour savoir comment gérer les sauvegardes de données entre elles services AWS, consultez la section [AWS Sauvegarde](#) du manuel du AWS Backup développeur.

## Configuration de FSx pour ONTAP, de FSx pour OpenZFS et du stockage partagé File Cache

Pour FSx pour ONTAP, FSx pour OpenZFS et File Cache, vous pouvez utiliser [FsxOntapSettings/VolumeId](#), [FsxOpenZfsSettings/et](#) pour spécifier le montage d'un volume externe existant ou d'un cache de fichiers [FileCacheId](#) pour votre cluster. [VolumeIdFileCacheSettings](#)

AWS ParallelCluster le stockage partagé géré n'est pas pris en charge pour FSx pour ONTAP, FSx pour OpenZFS et File Cache.

## Utilisation du stockage partagé dans AWS ParallelCluster

Découvrez comment travailler avec le stockage partagé AWS ParallelCluster et comment le partager.

### Rubriques

- [AWS ParallelCluster considérations relatives au stockage partagé](#)
- [Convertir le stockage AWS ParallelCluster géré en stockage externe](#)

## AWS ParallelCluster considérations relatives au stockage partagé

Tenez compte des points suivants lorsque vous travaillez avec un stockage partagé dans AWS ParallelCluster.

- Sauvegardez les données de votre système de fichiers à l'aide [AWS Backup](#) d'une autre méthode pour gérer les sauvegardes de tous vos systèmes de stockage.

- Pour ajouter du stockage partagé, vous devez ajouter une section de stockage partagé à votre fichier de configuration et créer ou mettre à jour le cluster.
- Pour supprimer le stockage partagé, vous devez supprimer la section de stockage partagé de votre fichier de configuration et mettre à jour le cluster.
- Pour remplacer le stockage partagé AWS ParallelCluster géré existant par un nouveau stockage géré, modifiez la valeur de [SharedStorage/Name](#) et mettez à jour le cluster.

#### Warning

Par défaut, le stockage AWS ParallelCluster géré et les données existants sont supprimés lorsque vous effectuez la mise à jour du cluster avec un nouveau Name paramètre. Si vous devez modifier Name et conserver les données de stockage partagé géré existantes, assurez-vous de les DeletionPolicy définir Retain ou de les sauvegarder avant de mettre à jour le cluster.

- Si vous ne sauvegardez pas les données de stockage AWS ParallelCluster géré et si DeletionPolicy c'est le casDelete, vos données sont supprimées soit lorsque votre cluster est supprimé, soit lorsque votre stockage géré est retiré de la configuration du cluster et que le cluster est mis à jour.
- Si vous ne sauvegardez pas les données de stockage AWS ParallelCluster gérées et si DeletionPolicy c'est le casRetain, votre système de fichiers est détaché avant que le cluster ne soit supprimé et peut être rattaché à un autre cluster en tant que système de fichiers externe. Vos données sont préservées.
- Si le stockage AWS ParallelCluster géré est supprimé de la configuration du cluster et si c'DeletionPolicyest le casRetain, il peut être rattaché au cluster en tant que système de fichiers externe tout en préservant les données de votre cluster.
- À partir de AWS ParallelCluster la version 3.4.0, vous pouvez améliorer la sécurité des montages du système de fichiers Amazon EFS en configurant [SharedStorage/EfsSettings/EncryptionInTransit](#) et [IamAuthorization](#) les paramètres.
- Lorsque vous montez un système de fichiers externe dans le répertoire /home, AWS ParallelCluster copie le contenu du répertoire /home du nœud principal vers le système de fichiers externe. Il copie les données existantes dans le répertoire /home sans remplacer les fichiers ou répertoires existants sur le stockage externe. Cela inclut la clé SSH du cluster pour l'utilisateur par défaut au cas où elle n'existe pas déjà sur le système de fichiers externe. Par conséquent, tous les autres clusters qui montent le même système de fichiers externe dans leur répertoire /home respectif auront également la même clé SSH pour leur utilisateur par défaut du cluster.

- Dans un environnement multi-clusters qui monte le même système de fichiers externe dans les répertoires `/home` des clusters, les clés SSH qui accordent l'accès aux nœuds de calcul, créées sur le nœud principal par AWS ParallelCluster, ne sont générées qu'une seule fois lorsque le premier cluster monte le système de fichiers externe sur `/home`. Tous les autres clusters utilisent la même clé SSH. Par conséquent, toute personne possédant la clé SSH de l'utilisateur par défaut de ces clusters partagés peut accéder à n'importe quel cluster. Tous les nœuds de calcul autorisent les connexions à l'aide de la clé initialement générée.

## Convertir le stockage AWS ParallelCluster géré en stockage externe

Découvrez comment convertir le stockage AWS ParallelCluster géré en stockage externe.

Les procédures sont basées sur l'exemple d'extrait de fichier de configuration suivant.

```
...
- MountDir: /fsx
 Name: fsx
 StorageType: FsxLustre
 FsxLustreSettings:
 StorageCapacity: 1200
 DeletionPolicy: Delete
...
```

## Convertir le stockage AWS ParallelCluster géré en stockage externe

1. Définissez le `DeletionPolicy` à `Retain` dans le fichier de configuration du cluster.

```
...
- MountDir: /fsx
 Name: fsx
 StorageType: FsxLustre
 FsxLustreSettings:
 StorageCapacity: 1200
 DeletionPolicy: Retain
...
```

2. Pour définir la `DeletionPolicy` modification, exécutez la commande suivante.

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

3. Supprimez la `SharedStorage` section du fichier de configuration du cluster.

```
...
...
```

4. Pour transformer le géré SharedStorage en externe SharedStorage et le détacher du cluster, exécutez la commande suivante.

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

5. Votre stockage partagé est désormais externe et détaché du cluster.
6. Pour associer votre système de fichiers externe au cluster d'origine ou à un autre cluster, procédez comme suit.
  - a. Obtenez l'ID du système de fichiers FSx for Lustre.
    - i. Pour utiliser la AWS CLI commande suivante, recherchez le système de fichiers dont le nom inclut le nom de votre cluster d'origine et notez l'ID du système de fichiers.

```
aws fsx describe-file-systems
```

- ii. Pour utiliser le AWS Management Console, connectez-vous et accédez au [site https://console.aws.amazon.com/fsx/](https://console.aws.amazon.com/fsx/). Dans la liste des systèmes de fichiers, recherchez le système de fichiers dont le nom inclut le nom de votre cluster d'origine et notez l'ID du système de fichiers.
- b. Mettez à jour les règles du groupe de sécurité du système de fichiers afin de fournir un accès vers et depuis les sous-réseaux du système de fichiers et du cluster. Vous trouverez le nom et l'ID du groupe de sécurité du système de fichiers dans la console Amazon FSx.

Ajoutez des règles au groupe de sécurité du système de fichiers qui autorisent le trafic TCP entrant et sortant depuis et vers le nœud principal et les plages ou préfixes IP CIDR du nœud de calcul. Spécifiez les ports TCP 988, 1021, 1022 et 1023 pour le trafic TCP entrant et sortant.

Pour plus d'informations, consultez

[SharedStorage/FsxLustreSettings/FileSystemId](#) et [Création, configuration et suppression de groupes de sécurité pour Amazon EC2](#) dans le guide de l'AWS Command Line Interface utilisateur de la version 2.

- c. Ajoutez la SharedStorage section à la configuration du cluster.

```

...
- MountDir: /fsx
 Name: fsx-external
 StorageType: FsxLustre
 FsxLustreSettings:
 FileSystemId: fs-02e5b4b4abd62d51c
...

```

- d. Pour ajouter le stockage partagé externe au cluster, exécutez la commande suivante.

```
pcluster update-cluster -n cluster-name -c cluster-config.yaml
```

## Quotas pour le stockage partagé

Configurez le cluster `SharedStorage` pour monter le stockage de fichiers partagé existant et créer un nouveau stockage de fichiers partagé en fonction des quotas répertoriés dans le tableau suivant.

Les quotas de stockage de fichiers montés pour chaque cluster

| Type de stockage partagé de fichiers | AWS ParallelCluster stockage géré | Stockage externe | Total net des quotas |
|--------------------------------------|-----------------------------------|------------------|----------------------|
| Amazon EBS                           | 5                                 | 5                | 5                    |
| RAID                                 | 1                                 | 0                | 1                    |
| Amazon EFS                           | 1                                 | 20               | 21                   |
| Amazon FSx †                         | 1 FSx pour Lustre                 | 20               | 21                   |

### Note

Ce tableau des quotas a été ajouté dans la AWS ParallelCluster version 3.2.0.

† prend AWS ParallelCluster uniquement en charge le montage des systèmes Amazon FSx pour NetApp ONTAP, Amazon FSx pour OpenZFS et File Cache existants. Il ne prend pas en charge la création de nouveaux systèmes FSx pour ONTAP, FSx pour OpenZFS et File Cache.



**Note**

Si vous l'utilisez AWS Batch comme planificateur, FSx for Lustre n'est disponible que sur le nœud principal du cluster.

Les caches de fichiers ne prennent pas en charge les AWS Batch planificateurs.

## AWS ParallelCluster ressources et balisage

Avec AWS ParallelCluster vous pouvez créer des balises pour suivre et gérer vos AWS ParallelCluster ressources. Vous définissez les balises que vous souhaitez AWS CloudFormation créer et propager à toutes les ressources du cluster dans le fichier [Tags Section](#) de configuration du cluster. Vous pouvez également utiliser des balises AWS ParallelCluster générées automatiquement pour suivre et gérer vos ressources.

Lorsque vous créez un cluster, le cluster et ses ressources sont étiquetés avec les balises AWS ParallelCluster et AWS systems définies dans cette section.

AWS ParallelCluster applique des balises aux instances, aux volumes et aux ressources du cluster. Pour identifier la pile de clusters, AWS CloudFormation applique des balises AWS système aux instances de cluster. Pour identifier les modèles de lancement du cluster Amazon EC2, Amazon EC2 applique des balises système aux instances. Vous pouvez utiliser ces balises pour afficher et gérer vos AWS ParallelCluster ressources.

Vous ne pouvez pas modifier les balises AWS système. Afin d'éviter tout impact sur les AWS ParallelCluster fonctionnalités, ne modifiez pas les AWS ParallelCluster balises.

Voici un exemple de balise AWS système pour une AWS ParallelCluster ressource. Vous ne pouvez pas les modifier.

```
"aws:cloudformation:stack-name"="clustername"
```

Voici un exemple de AWS ParallelCluster balise appliquée à une ressource. Ne les modifiez pas.

```
"parallelcluster:cluster-name"="clustername"
```

Vous pouvez consulter ces balises dans la section Amazon EC2 du. AWS Management Console

## Affichage des balises

1. [Accédez à la console Amazon EC2 à l'adresse `https://console.aws.amazon.com/ec2/`.](https://console.aws.amazon.com/ec2/)
2. Pour afficher toutes les balises de cluster, choisissez Tags dans le volet de navigation.
3. Pour afficher les balises de cluster par instance, choisissez Instances dans le volet de navigation.
4. Sélectionnez une instance de cluster.
5. Choisissez l'onglet Gérer les balises dans les détails de l'instance et consultez les balises.
6. Choisissez l'onglet Stockage dans les détails de l'instance.
7. Sélectionnez l'ID du volume.
8. Dans Volumes, choisissez le volume.
9. Choisissez l'onglet Tags dans les détails du volume et visualisez les tags.

### AWS ParallelCluster balises d'instance du nœud principal

| Clé                                        | Valeur de balise                                                                                                         |
|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <code>parallelcluster:cluster-name</code>  | <i>clustername</i>                                                                                                       |
| Name                                       | HeadNode                                                                                                                 |
| <code>aws:ec2launchtemplate:id</code>      | <i>lt-1234567890abcdef0</i>                                                                                              |
| <code>aws:ec2launchtemplate:version</code> | <i>1</i>                                                                                                                 |
| <code>parallelcluster:node-type</code>     | HeadNode                                                                                                                 |
| <code>aws:cloudformation:stack-name</code> | <i>clustername</i>                                                                                                       |
| <code>aws:cloudformation:logical-id</code> | HeadNode                                                                                                                 |
| <code>aws:cloudformation:stack-id</code>   | <i>arn:aws:cloudformation: <b>region-id</b> :ACCOUNTID :stack/clustername /1234abcd-12ab-12ab-12ab-1234567890abcdef0</i> |
| <code>parallelcluster:version</code>       | <i>3.7.0</i>                                                                                                             |

## AWS ParallelCluster balises de volume racine du nœud principal

| Clé de balise                             | Valeur de balise   |
|-------------------------------------------|--------------------|
| <code>parallelcluster:cluster-name</code> | <i>clustername</i> |
| <code>parallelcluster:node-type</code>    | HeadNode           |
| <code>parallelcluster:version</code>      | <i>3.7.0</i>       |

## AWS ParallelCluster balises d'instance de nœud de calcul

| Clé                                                | Valeur de balise             |
|----------------------------------------------------|------------------------------|
| <code>parallelcluster:cluster-name</code>          | <i>clustername</i>           |
| <code>parallelcluster:compute-resource-name</code> | <i>compute-resource-name</i> |
| <code>aws:ec2launchtemplate:id</code>              | <i>lt-1234567890abcdef0</i>  |
| <code>aws:ec2launchtemplate:version</code>         | <i>1</i>                     |
| <code>parallelcluster:node-type</code>             | Compute                      |
| <code>parallelcluster:queue-name</code>            | <i>queue-name</i>            |
| <code>parallelcluster:version</code>               | <i>3.7.0</i>                 |

## AWS ParallelCluster balises de volume racine du nœud de calcul

| Clé de balise                                      | Valeur de balise             |
|----------------------------------------------------|------------------------------|
| <code>parallelcluster:cluster-name</code>          | <i>clustername</i>           |
| <code>parallelcluster:compute-resource-name</code> | <i>compute-resource-name</i> |
| <code>parallelcluster:node-type</code>             | Compute                      |

| Clé de balise                           | Valeur de balise  |
|-----------------------------------------|-------------------|
| <code>parallelcluster:queue-name</code> | <i>queue-name</i> |
| <code>parallelcluster:version</code>    | <i>3.7.0</i>      |

#### AWS ParallelCluster Balises d'interface utilisateur

| Clé de balise                   | Valeur de balise  |
|---------------------------------|-------------------|
| <code>parallelcluster-ui</code> | <code>true</code> |

## Surveillance AWS ParallelCluster et journaux

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité AWS ParallelCluster et des performances de vos autres AWS solutions. AWS fournit les outils de surveillance suivants pour surveiller AWS ParallelCluster, signaler tout problème et prendre des mesures automatiques le cas échéant :

- Amazon CloudWatch surveille vos AWS ressources et les applications que vous utilisez AWS en temps réel. Vous pouvez collecter et suivre les métriques, créer des tableaux de bord personnalisés, et définir des alarmes qui vous informent ou prennent des mesures lorsqu'une métrique spécifique atteint un seuil que vous spécifiez. Par exemple, vous pouvez CloudWatch suivre l'utilisation du processeur ou d'autres indicateurs de vos instances Amazon EC2 et lancer automatiquement de nouvelles instances en cas de besoin. Pour plus d'informations, consultez le [guide de CloudWatch l'utilisateur Amazon](#).
- Amazon CloudWatch Logs vous permet de surveiller, de stocker et d'accéder à vos fichiers journaux à partir d'instances Amazon EC2 et d'autres sources. CloudTrail CloudWatch Les journaux peuvent surveiller les informations contenues dans les fichiers journaux et vous avertir lorsque certains seuils sont atteints. Vous pouvez également archiver vos données de journaux dans une solution de stockage hautement durable. Pour plus d'informations, consultez le [guide de l'utilisateur d'Amazon CloudWatch Logs](#).
- AWS CloudTrail capture les appels d'API et les événements associés créés par votre Compte AWS ou au nom de celui-ci et livre les fichiers journaux dans un compartiment Amazon S3 que vous spécifiez. Vous pouvez identifier les utilisateurs et les comptes qui ont appelé AWS, l'adresse IP

source à partir de laquelle les appels ont été émis, ainsi que le moment où les appels ont eu lieu. Pour de plus amples informations, veuillez consulter le [Guide de l'utilisateur AWS CloudTrail](#).

- Amazon EventBridge est un service de bus d'événements sans serveur qui permet de connecter facilement vos applications à des données provenant de diverses sources. EventBridge fournit un flux de données en temps réel à partir de vos propres applications, applications software-as-a-Service (SaaS) et AWS services et achemine ces données vers des cibles telles que Lambda. Cela vous permet de surveiller les événements qui se produisent dans les services et de créer des architectures basées sur les événements. Pour plus d'informations, consultez le [guide de EventBridge l'utilisateur Amazon](#).

## Rubriques

- [Intégration avec Amazon CloudWatch Logs avec Amazon Logs](#)
- [Tableau de CloudWatch bord Amazon](#)
- [CloudWatchAlarmes Amazon pour les métriques des clusters](#)
- [AWS ParallelClusterrotation des journaux configurée](#)
- [pclusterJournaux de la CLI](#)
- [Journaux de sortie de la console Amazon EC2](#)
- [Récupérez les journauxAWS ParallelCluster d'AWS ParallelClusterinterface utilisateur et d'exécution](#)
- [Récupération et conservation des journaux](#)

## Intégration avec Amazon CloudWatch Logs avec Amazon Logs

Pour plus d'informations sur les CloudWatch journaux, consultez le [guide de l'utilisateur d'Amazon CloudWatch Logs](#). Pour configurer l'intégration CloudWatch des journaux, consultez la [Monitoring](#) section. Pour savoir comment ajouter des journaux personnalisés à la CloudWatch configuration à l'aide de `append-config`, consultez les [fichiers de configuration de plusieurs CloudWatch agents](#) dans le Guide de l'CloudWatchutilisateur Amazon.

### CloudWatchJournaux du cluster Amazon Logs

Un groupe de journaux est créé pour chaque cluster avec un nom `/aws/parallelcluster/cluster-name-<timestamp>` (par exemple, `/aws/parallelcluster/testCluster-202202050215`). Chaque journal (ou ensemble de journaux si le chemin contient un `*`) de chaque nœud possède un flux

de journaux nommé `{hostname}.{instance_id}.{logIdentifieur}`. (Par exemple `ip-172-31-10-46.i-02587cf29cc3048f3.nodewatcher`.) Les données du journal sont envoyées CloudWatch par l'[CloudWatchagent](#), qui s'exécute comme root sur toutes les instances du cluster.

Un Amazon CloudWatch Dashs est créé lors de la création du cluster. Ce tableau de bord vous permet de consulter les journaux stockés dans CloudWatch Logs. Pour plus d'informations, veuillez consulter [Tableau de CloudWatch bord Amazon](#).

Cette liste contient le *LogIdentifieur* et le chemin des flux de journaux disponibles pour les plateformes, les planificateurs et les nœuds.

Flux de journaux disponibles pour les plateformes, les planificateurs et les nœuds

| Plateformes                                    | Schedulers        | Nœuds               | Flux de journaux                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------------------------------|-------------------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| amazon<br>centos<br>chapeau<br>rouge<br>ubuntu | awsbatch<br>slurm | HeadNode            | authentificateur DCV : <code>/var/log/parallelcluster/parallelcluster_dcv_authenticator.log</code><br>dcv-ext-authenticator: <code>/var/log/parallelcluster/parallelcluster_dcv_connect.log</code><br>agent DCV : <code>/var/log/dcv/agent.*.log</code><br>session CV-X : <code>/var/log/dcv/dcv-xsession.*.log</code><br>serveur DCV : <code>/var/log/dcv/server.log</code><br>dcv-session-launcher: <code>/var/log/dcv/sessionlauncher.log</code><br>XCV : <code>/var/log/dcv/Xdcv.*.log</code><br>cfn-init cfn : <code>/var/log/cfn-init.log</code><br>chef-client : <code>/var/log/chef-client.log</code> |
| amazon<br>centos                               | awsbatch<br>slurm | Compute<br>HeadNode | cloud-init : <code>/var/log/cloud-init.log</code><br>supervisé : <code>/var/log/supervisord.log</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

| Plateformes                                    | Schedulers | Nœuds         | Flux de journaux                                                                                                                                                                                                     |
|------------------------------------------------|------------|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| chapeau<br>rouge<br>ubuntu                     |            |               |                                                                                                                                                                                                                      |
| amazon<br>centos<br>chapeau<br>rouge<br>ubuntu | slurm      | Compute<br>et | cloud-init-output: /var/log/cloud-init-output.log<br>computemgtd : /var/log/parallelcluster/computemgtd<br>bouse : /var/log/slurmd.log<br>slurm_prolog_epilog : /var/log/parallelcluster/sl<br>urm_prolog_epilog.log |

| Plateformes                                    | Schedulers        | Nœuds                   | Flux de journaux                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------------|-------------------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| amazon<br>centos<br>chapeau<br>rouge<br>ubuntu | slurm             | HeadNode                | <pre> sssd : /var/log/sssds/sssds.log  sssd_domain_default : /var/log/sssds/sssds_default.log  générateur pam_ssh_key_generator : /var/log/parallelcluster/pam_ssh_key_generator.log  gestion de l'état du cluster : /var/log/parallelcluster/clusterstatusmgtd  clustermgtd : /var/log/parallelcluster/clustermgtd  sortie_de la console de calcul : /var/log/parallelcluster/compute_console_output  slurm_resume : /var/log/parallelcluster/slurm_resume.log  slurm_suspend : /var/log/parallelcluster/slurm_suspend.log  slurmctld : /var/log/slurmctld.log  slurm_fleet_status_manager : /var/log/parallelcluster/slurm_fleet_status_manager.log </pre> |
| amazon<br>centos<br>chapeau<br>rouge<br>ubuntu | awsbatch<br>slurm | ComputeNode<br>HeadNode | <pre> messages système : /var/log/messages </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| amazon<br>centos<br>chapeau<br>rouge<br>ubuntu | awsbatch<br>slurm | ComputeNode<br>HeadNode | <pre> Syslog : /var/log/syslog </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |



Les tâches des clusters qui utilisent AWS Batch stockent les résultats des tâches ayant atteint un état de `RUNNING` ou `SUCCEEDED`, ou `FAILED` dans CloudWatch des journaux. Le groupe de journaux est `/aws/batch/job` et le format du nom du flux de journal est `jobDefinitionName/default/ecs_task_id`. Par défaut, ces journaux sont configurés pour ne pas expirer, mais vous pouvez modifier la période de conservation. Pour plus d'informations, consultez la section [Modification de la rétention des données de CloudWatch journaux dans Logs](#) du Amazon CloudWatch Logs User Guide.

## Amazon CloudWatch Logs crée des journaux d'images

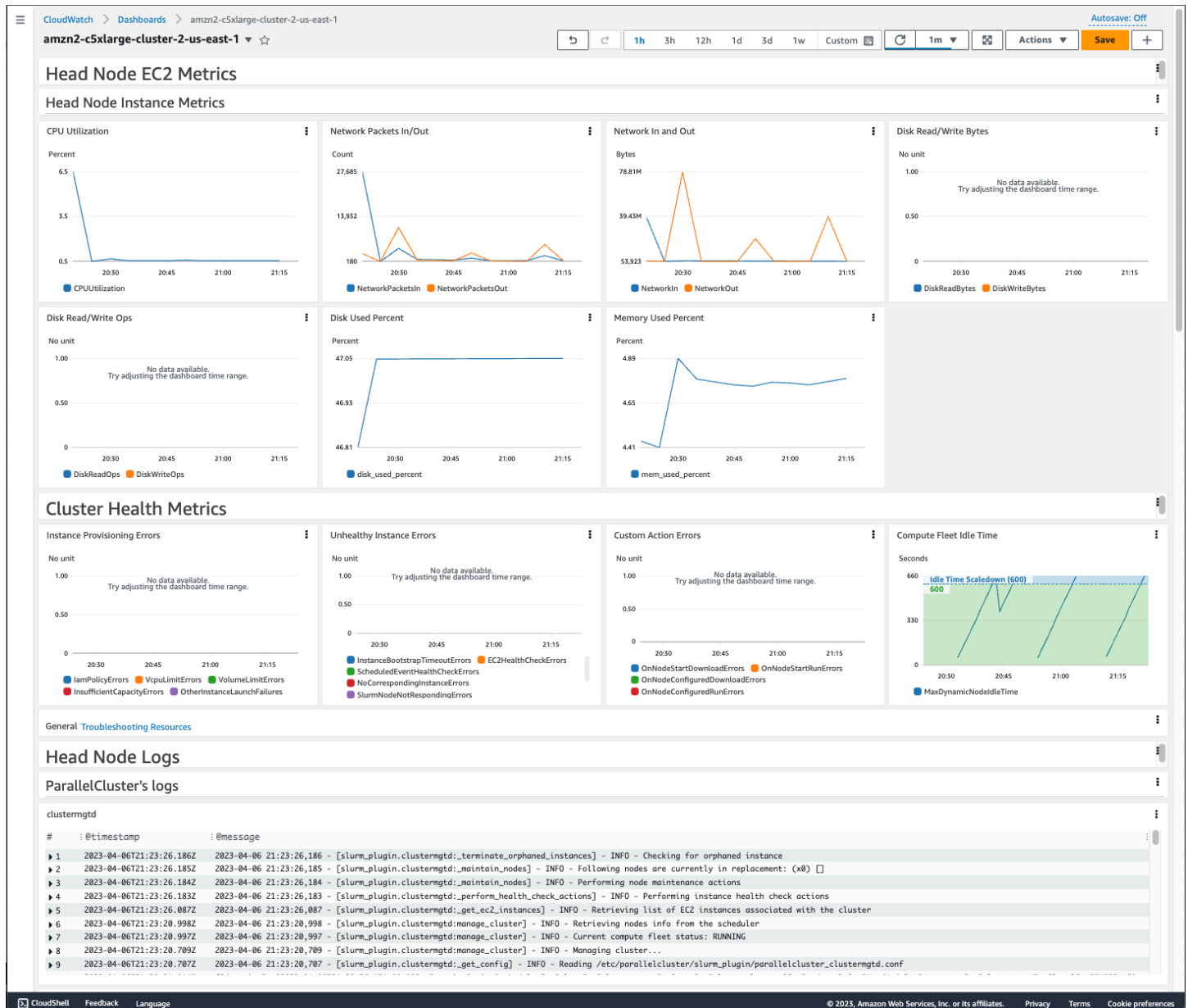
Un groupe de journaux est créé pour chaque image de construction personnalisée avec un nom, `/aws/imagebuilder/ParallelClusterImage-<image-id>`. Un flux de journal unique portant le nom `{pcluster-version}/1` contient le résultat du processus de création de l'image.

Vous pouvez accéder aux journaux à l'aide des commandes `pcluster` d'image. Pour plus d'informations, veuillez consulter [AWS ParallelCluster Personnalisation de l'AMI](#).

## Tableau de CloudWatch bord Amazon

Un CloudWatch tableau de bord Amazon est créé lors de la création d'un cluster. Cela facilite la surveillance des nœuds de votre cluster et l'affichage des journaux stockés dans Amazon CloudWatch Logs. Le nom du tableau de bord est `ClusterName-Region`. `ClusterName` est le nom de votre cluster et `Region` est Région AWS le nom du cluster. Vous pouvez accéder au tableau de bord dans la console ou en l'ouvrant `https://console.aws.amazon.com/cloudwatch/home?region=Region#dashboards:name=ClusterName-Region`.

L'image suivante montre un exemple de CloudWatch tableau de bord pour un cluster.



## Métriques relatives aux instances Head Node

La première section du tableau de bord affiche les graphiques des métriques Amazon EC2 du nœud principal.

Si votre cluster dispose d'un stockage partagé, la section suivante présente les métriques de stockage partagé.

## Indicateurs de santé du cluster

Si votre cluster l'utilise Slurm pour la planification, les graphiques des métriques de santé du cluster indiquent les erreurs des nœuds de calcul du cluster en temps réel. Pour plus d'informations,

consultez [Résolution des problèmes liés aux indicateurs de santé du](#). Les métriques de santé du cluster sont ajoutées au tableau de bord à partir de AWS ParallelCluster la version 3.6.0.

## Journaux du nœud principal

La dernière section répertorie les journaux des nœuds principaux regroupés par journaux, journaux AWS ParallelCluster du planificateur, journaux d'intégration NICE DCV et journaux du système.

Pour plus d'informations sur les CloudWatch tableaux de bord Amazon, consultez la section [Utilisation des CloudWatch tableaux de bord Amazon](#) dans le guide de CloudWatch l'utilisateur Amazon.

Si vous ne souhaitez pas créer le tableau de CloudWatch bord Amazon, vous pouvez le désactiver en réglant [Monitoring/Dashboards/CloudWatch/Enabled](#) sur `false`.

### Note

Si vous désactivez la création du tableau de CloudWatch bord Amazon, vous désactivez également Amazon CloudWatch `disk_used_percent` et les `memory_used_percent` alarmes pour votre cluster. Pour plus d'informations, consultez [CloudWatch Alarmes Amazon pour les métriques des clusters](#).

Les `memory_used_percent` alarmes `disk_used_percent` et sont ajoutées à partir de AWS ParallelCluster la version 3.6.

## CloudWatch Alarmes Amazon pour les métriques des clusters

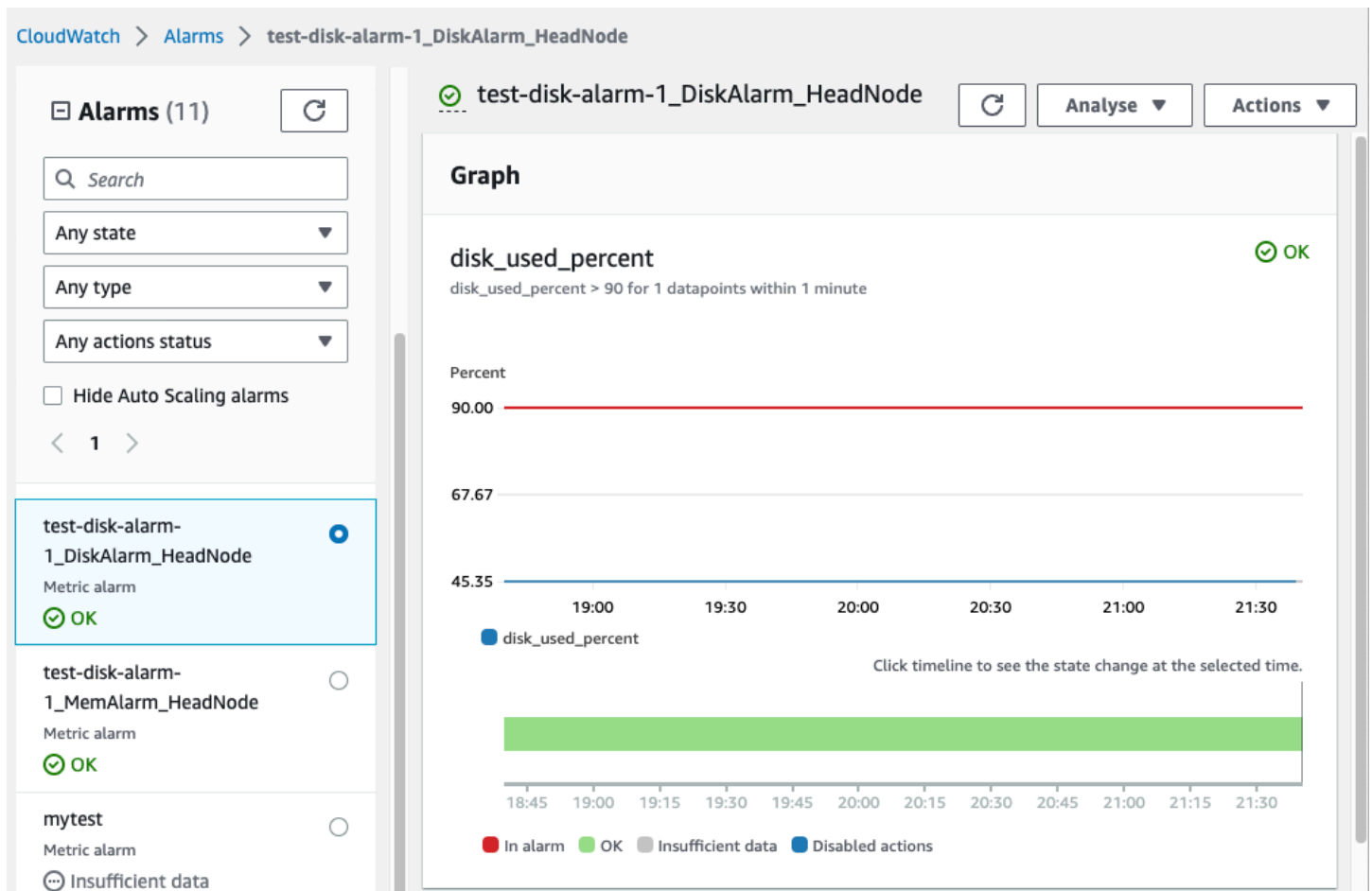
À partir de AWS ParallelCluster la version 3.6, vous pouvez configurer votre cluster avec des CloudWatch alarmes Amazon pour surveiller le nœud principal. Une alarme surveille le volume `disk_used_percent`. L'autre alarme surveille la `mem_used_percent` métrique. Pour plus d'informations, consultez [les statistiques collectées par l'CloudWatchagent](#) dans le Guide de CloudWatch l'utilisateur Amazon.

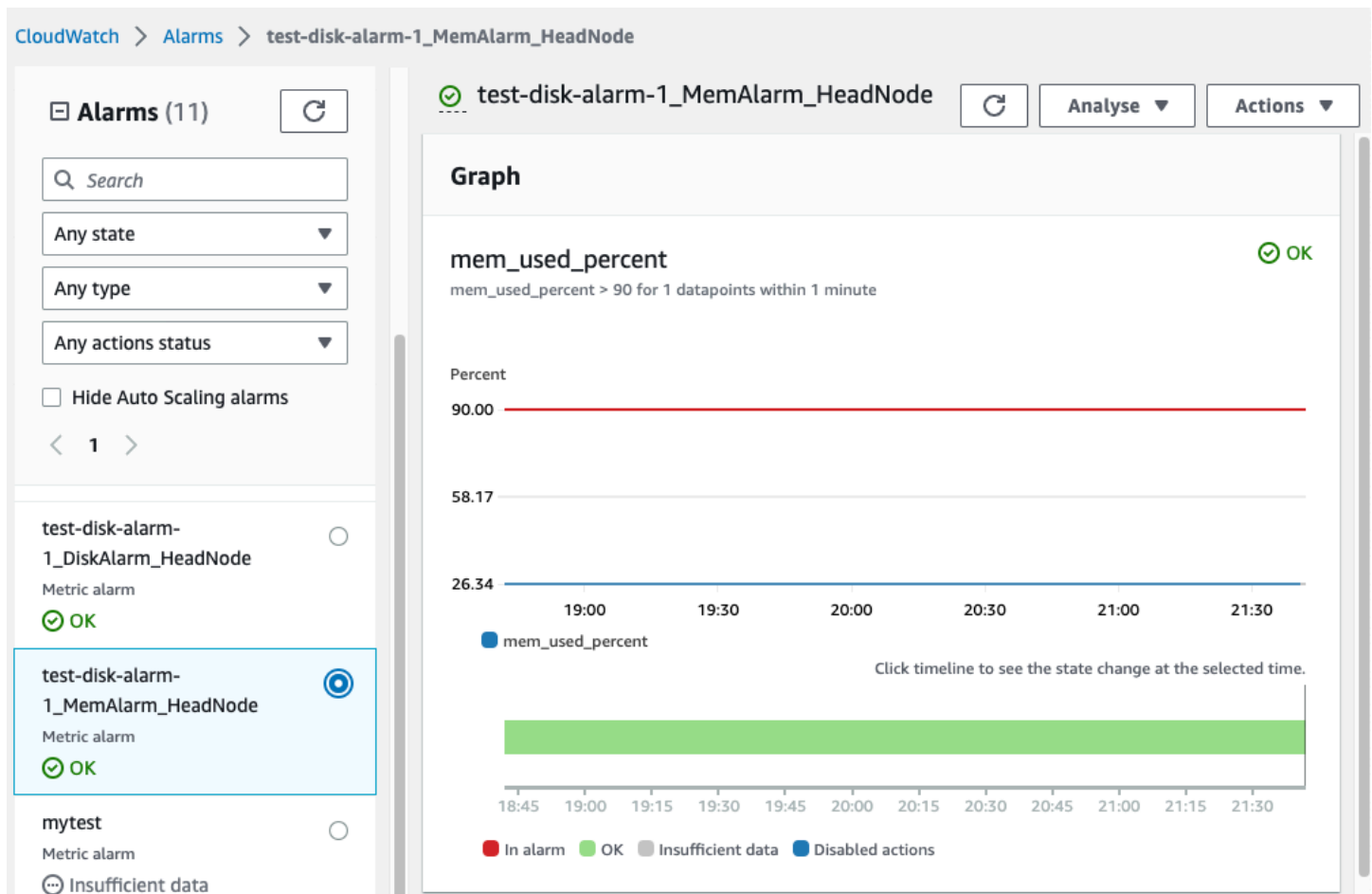
Les alarmes sont nommées comme suit :

- `cluster-name_DiskAlarm_HeadNode`
- `cluster-name_MemAlarm_HeadNode`

par le *nom de votre cluster*.

Accédez aux alarmes dans la CloudWatch console en choisissant Alarmes dans le volet de navigation. Les images suivantes montrent l'alarme d'utilisation du disque et l'alarme d'utilisation de la mémoire pour un cluster.





L'alarme d'utilisation du disque est activée ALARM lorsque le pourcentage d'utilisation du disque est supérieur à 90 % pour un point de données, sur une période d'une minute.

L'alarme d'utilisation de la mémoire est activée ALARM lorsque le pourcentage d'utilisation de la mémoire est supérieur à 90 % pour 1 point de données, sur une période d'une minute.

### Note

AWS ParallelCluster ne configure pas les actions d'alarme par défaut. Pour plus d'informations sur la configuration des actions d'alarme, telles que l'envoi de notifications, voir [Actions d'alarme](#). Pour plus d'informations sur les CloudWatch alarmes Amazon, consultez la section [Utilisation des CloudWatch alarmes Amazon](#) dans le Guide de CloudWatch l'utilisateur Amazon.

Si vous ne souhaitez pas créer ces CloudWatch alarmes Amazon, désactivez-les en réglant [Monitoring//DashboardsCloudWatch/Enabled](#) sur `false` dans la configuration du cluster. Cela

désactive également la création du tableau de CloudWatch bord Amazon. Pour plus d'informations, veuillez consulter [Tableau de CloudWatch bord Amazon](#).

### Note

Si vous désactivez la création du tableau de CloudWatch bord Amazon, vous désactivez également Amazon CloudWatch `disk_used_percent` et les `memory_used_percent` alarmes pour votre cluster.

## AWS ParallelCluster rotation des journaux configurée

Les configurations de rotation des journaux se trouvent dans `/etc/logrotate.d/parallelcluster*_log_rotation` fichiers. Lorsqu'un journal configuré change, le contenu actuel du journal est conservé dans une sauvegarde unique et le journal vidé reprend la journalisation.

Une seule sauvegarde est conservée pour chaque journal configuré.

AWS ParallelCluster configure un journal en croissance rapide pour qu'il effectue une rotation lorsqu'il atteint 50 Mo. Les grumes à croissance rapide sont liées à la mise à l'échelle et Slurm, y compris `/var/log/parallelcluster/clustermgtd`, `/var/log/parallelcluster/slurm_resume.log`, et `/var/log/slurmctld.log`.

AWS ParallelCluster configure un journal à croissance lente pour qu'il effectue une rotation lorsqu'il atteint 10 Mo.

Vous pouvez consulter les journaux antérieurs qui sont conservés pendant le nombre de jours défini dans la configuration du cluster. [Logs/CloudWatch/RetentionInDays](#) réglage avec CloudFormation journalisation activée. Consultez le `RetentionInDays` paramètres pour voir si le nombre de jours doit être augmenté pour votre cas d'utilisation.

AWS ParallelCluster configure et fait pivoter les journaux suivants :

Journaux du nœud principal

```
/var/log/cloud-init.log
/var/log/supervisord.log
/var/log/cfn-init.log
/var/log/chef-client.log
```

```
/var/log/dcv/server.log
/var/log/dcv/sessionlauncher.log
/var/log/dcv/agent.*.log
/var/log/dcv/dcv-xsession.*.log
/var/log/dcv/Xdcv.*.log
/var/log/parallelcluster/pam_ssh_key_generator.log
/var/log/parallelcluster/clustermgtd
/var/log/parallelcluster/clusterstatusmgtd
/var/log/parallelcluster/slurm_fleet_status_manager.log
/var/log/parallelcluster/slurm_resume.log
/var/log/parallelcluster/slurm_suspend.log
/var/log/slurmctld.log
/var/log/slurmdbd.log
/var/log/parallelcluster/compute_console_output.log
```

## Journaux des nœuds de calcul

```
/var/log/cloud-init.log
/var/log/supervisord.log
/var/log/cloud-init-output.log
/var/log/parallelcluster/computemgtd
/var/log/slurmd.log
```

## Journaux du nœud de connexion

```
/var/log/cloud-init.log
/var/log/cloud-init.log
/var/log/cloud-init-output.log
/var/log/supervisord.log
/var/log/parallelcluster/pam_ssh_key_generator.log
```

## **pcluster** Journaux de la CLI

La `pcluster` CLI enregistre les journaux de vos commandes dans `pcluster.log`. # des fichiers/  
`home/user/.parallelcluster/`.

Pour chaque commande, les journaux incluent généralement la commande avec les entrées, une copie de la version de l'API CLI utilisée pour créer la commande, la réponse, ainsi que les messages d'information et d'erreur. Pour une commande de création et de génération, les journaux incluent également le fichier de configuration, les opérations de validation du fichier de configuration, le CloudFormation modèle et les commandes de pile.

Vous pouvez utiliser ces journaux pour vérifier les erreurs, les entrées, les versions et les commandes `pcluster` CLI. Ils peuvent également servir d'enregistrement du moment où les commandes ont été effectuées.

## Journaux de sortie de la console Amazon EC2

Lorsqu'elle AWS ParallelCluster détecte qu'une instance de nœud de calcul statique se termine de manière inattendue, elle tente de récupérer la sortie de la console Amazon EC2 à partir de l'instance de nœud terminée après un certain temps. Ainsi, si le nœud de calcul n'a pas pu communiquer avec Amazon CloudWatch, des informations de dépannage utiles expliquant pourquoi le nœud s'est arrêté peuvent toujours être extraites de la sortie de la console. Cette sortie de console est enregistrée dans le `/var/log/parallelcluster/compute_console_output` journal du nœud principal. Pour plus d'informations sur la sortie de la console Amazon EC2, consultez la section [Sortie de la console d'instance](#) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Par défaut, extrait AWS ParallelCluster uniquement la sortie de la console à partir d'un sous-ensemble d'échantillons de nœuds terminés. Cela évite que le nœud principal du cluster ne soit submergé par plusieurs demandes de sortie de console causées par un grand nombre de résiliations. Par défaut, AWS ParallelCluster attend 5 minutes entre la détection de la terminaison et la récupération de la sortie de console pour laisser à Amazon EC2 le temps de récupérer la sortie de console finale depuis les nœuds.

Vous pouvez modifier la taille de l'échantillon et les valeurs des paramètres de temps d'attente dans le `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` fichier situé sur le nœud principal.

Cette fonctionnalité est ajoutée dans la AWS ParallelCluster version 3.5.0.

## Paramètres de sortie de la console Amazon EC2

Vous pouvez modifier les valeurs des paramètres de sortie de la console Amazon EC2 suivants dans le `/etc/parallelcluster/slurm_plugin/parallelcluster_clustermgtd.conf` fichier situé sur le nœud principal.

### **`compute_console_logging_enabled`**

Pour désactiver la collecte des journaux de sortie de la console, définissez `compute_console_logging_enabled` sur `false`. L'argument par défaut est `true`.

Vous pouvez mettre à jour ce paramètre à tout moment, sans arrêter le parc informatique.



## **compute\_console\_logging\_max\_sample\_size**

`compute_console_logging_max_sample_size` définit le nombre maximum de nœuds de calcul à partir desquels sont AWS ParallelCluster collectées les sorties de console chaque fois qu'il détecte une interruption inattendue. Si cette valeur est inférieure à 1, AWS ParallelCluster extrait la sortie de console de tous les nœuds terminés. La valeur par défaut est 1.

Vous pouvez mettre à jour ce paramètre à tout moment, sans arrêter le parc informatique.

## **compute\_console\_wait\_time**

`compute_console_wait_time` définit le temps, en secondes, qui s'écoule entre AWS ParallelCluster la détection d'une défaillance d'un nœud et la collecte de la sortie de console à partir de ce nœud. Vous pouvez augmenter le temps d'attente si vous déterminez qu'Amazon EC2 a besoin de plus de temps pour collecter le résultat final du nœud arrêté. La valeur par défaut est de 300 secondes (5 minutes).

Vous pouvez mettre à jour ce paramètre à tout moment, sans arrêter le parc informatique.

## Récupérez les journaux AWS ParallelCluster d'AWS ParallelCluster interface utilisateur et d'exécution

Découvrez comment récupérer l'AWS ParallelCluster interface utilisateur et les journaux AWS ParallelCluster d'exécution à des fins de résolution des problèmes. Pour commencer, recherchez l'AWS ParallelCluster interface utilisateur et les noms de AWS ParallelCluster pile pertinents. Utilisez le nom de la pile pour localiser les groupes de journaux d'installation. Pour terminer, exportez les journaux. Ces journaux sont spécifiques à l'AWS ParallelCluster environnement d'exécution. Pour les journaux de cluster, reportez-vous à la section [Récupération et conservation des journaux](#).

### Prérequis

- Le AWS CLI est installé.
- Vous disposez des informations d'identification nécessaires pour exécuter Compte AWS des AWS CLI commandes sur l'AWS ParallelCluster interface utilisateur.
- Vous pouvez accéder à la CloudWatch console Amazon à Compte AWS partir de l'AWS ParallelCluster interface utilisateur.

## Étape 1 : Localisez les noms des piles concernées

Dans l'exemple suivant, remplacez le texte surligné en rouge par vos valeurs réelles.

Répertoriez les piles en utilisant l'emplacement Région AWS où vous avez installé l'AWS ParallelClusterinterface utilisateur :

```
$ aws cloudformation list-stacks --region aws-region-id
```

Notez les noms des piles suivantes :

- Nom de la pile qui a déployé l'AWS ParallelClusterinterface utilisateur sur votre compte. Vous avez saisi ce nom lors de l'installation de l'AWS ParallelClusterinterface utilisateur ; par exemple, `pcluster-ui`.
- LaAWS ParallelCluster pile qui est préfixée par le nom de pile que vous avez saisi ; par exemple, `pcluster-ui-ParallelClusterApi-ABCD1234EFGH`.

## Étape 2 : localiser les groupes de journaux

Réserver les groupes de journaux de la pile d'AWS ParallelClusterinterface utilisateur, comme montré dans l'exemple suivant :

```
$ aws cloudformation describe-stack-resources \
 --region aws-region-id \
 --stack-name pcluster-ui \
 --query "StackResources[?ResourceType == 'AWS::Logs::LogGroup' &&
(LogicalResourceId == 'ApiGatewayAccessLog' || LogicalResourceId ==
'ParallelClusterUILambdaLogGroup')].PhysicalResourceId" \
 --output text
```

Réserver les groupes de journaux de la pile d'AWS ParallelClusterAPI, comme montré dans l'exemple suivant :

```
$ aws cloudformation describe-stack-resources \
 --region aws-region-id \
 --stack-name pcluster-ui-ParallelCluster-Api-ABCD1234EFGH \
 --query "StackResources[?ResourceType == 'AWS::Logs::LogGroup' && LogicalResourceId
== 'ParallelClusterFunctionLogGroup'].PhysicalResourceId" \
 --output text
```

Observer la liste des groupes de journaux à utiliser à l'étape suivante.

### Étape 3 : Exporter les journaux

Pour collecter et exporter les journaux, procédez comme suit :

1. Connectez-vous à AWS Management Console, puis accédez à la CloudWatch console [Amazon](#) sur Compte AWS laquelle se trouve l'AWS ParallelCluster interface utilisateur.
2. Choisissez Logs, Logs Insights dans le volet de navigation.
3. Sélectionnez tous les groupes de journaux listés à l'étape précédente.
4. Choisissez une plage de temps, par exemple 12 heures.
5. Exécutez la requête suivante :

```
$ fields @timestamp, @message
| sort @timestamp desc
| limit 10000
```

6. Choisissez Exporter les résultats, Télécharger le tableau (JSON).

## Récupération et conservation des journaux

AWS ParallelCluster crée des métriques Amazon EC2 pour HeadNode et calcule les instances et le stockage. Vous pouvez consulter les statistiques dans les tableaux de bord personnalisés de la CloudWatch console. AWS ParallelCluster crée également des flux de CloudWatch journaux de cluster dans des groupes de journaux. Vous pouvez consulter ces journaux dans les tableaux de bord personnalisés ou les groupes de journaux de la CloudWatch console. La section Configuration du cluster de [surveillance](#) décrit comment modifier les CloudWatch journaux et le tableau de bord du cluster. Pour plus d'informations, consultez [Intégration avec Amazon CloudWatch Logs avec Amazon Logs](#) et [Tableau de CloudWatch bord Amazon](#).

Les journaux constituent une ressource utile pour résoudre les problèmes. Par exemple, si vous souhaitez supprimer un cluster défaillant, il peut être utile de créer d'abord une archive des journaux du cluster. Suivez les étapes décrites [Journaux d'archivage](#) pour créer une archive.

### Rubriques

- [Les journaux du cluster ne sont pas disponibles dans CloudWatch](#)
- [Journaux d'archivage](#)

- [Bûches préservées](#)
- [Journaux des nœuds interrompus](#)

## Les journaux du cluster ne sont pas disponibles dans CloudWatch

Si les journaux de cluster ne sont pas disponibles dans CloudWatch, assurez-vous de ne pas avoir remplacé la configuration des AWS ParallelCluster CloudWatch journaux lorsque vous ajoutez des journaux personnalisés à la configuration.

Pour ajouter des journaux personnalisés à la CloudWatch configuration, veillez à les ajouter à la configuration plutôt que de les récupérer et de les remplacer. Pour plus d'informations sur `fetch-config` et `append-config`, consultez la section [Fichiers de configuration de plusieurs CloudWatch agents](#) dans le Guide de CloudWatch l'utilisateur.

Pour restaurer la configuration du AWS ParallelCluster CloudWatch journal, vous pouvez exécuter les commandes suivantes dans un AWS ParallelCluster nœud :

```
$ PLATFORM="$(ohai platform | jq -r ".[]")"
LOG_GROUP_NAME="$(cat /etc/chef/dna.json | jq -r ".cluster.log_group_name")"
SCHEDULER="$(cat /etc/chef/dna.json | jq -r ".cluster.scheduler")"
NODE_ROLE="$(cat /etc/chef/dna.json | jq -r ".cluster.node_type")"
CONFIG_DATA_PATH="/usr/local/etc/cloudwatch_agent_config.json"
/opt/parallelcluster/pyenv/versions/cookbook_virtualenv/bin/python /usr/local/bin/
write_cloudwatch_agent_json.py --platform $PLATFORM --config $CONFIG_DATA_PATH --log-
group $LOG_GROUP_NAME --scheduler $SCHEDULER --node-role $NODE_ROLE
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2
-c file:/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json -s
```

## Journaux d'archivage

Vous pouvez archiver les journaux dans Amazon S3 ou dans un fichier local (selon le `--output-file` paramètre).

### Note

Ajoutez des autorisations à la politique du compartiment Amazon S3 pour accorder CloudWatch l'accès. Pour plus d'informations, consultez la section [Définir des autorisations sur un compartiment Amazon S3](#) dans le Guide de l'utilisateur CloudWatch des journaux.

```
$ pcluster export-cluster-logs --cluster-name mycluster --region eu-west-1 \
 --bucket bucketname --bucket-prefix logs
{
 "url": "https://bucketname.s3.eu-west-1.amazonaws.com/export-log/mycluster-
logs-202109071136.tar.gz?..."
}

use the --output-file parameter to save the logs locally
$ pcluster export-cluster-logs --cluster-name mycluster --region eu-west-1 \
 --bucket bucketname --bucket-prefix logs --output-file /tmp/archive.tar.gz
{
 "path": "/tmp/archive.tar.gz"
}
```

L'archive contient les flux Amazon CloudWatch Logs et les événements de AWS CloudFormation pile provenant du nœud principal et des nœuds de calcul au cours des 14 derniers jours, sauf indication explicite dans la configuration ou dans les paramètres de la `export-cluster-logs` commande. Le temps nécessaire à la fin de la commande dépend du nombre de nœuds du cluster et du nombre de flux de CloudWatch journaux disponibles dans Logs. Pour plus d'informations sur les flux de journaux disponibles, consultez [Intégration avec Amazon CloudWatch Logs avec Amazon Logs](#).

## Bûches préservées

À partir de la version 3.0.0, AWS ParallelCluster préserve CloudWatch les journaux par défaut lorsqu'un cluster est supprimé. Si vous souhaitez supprimer un cluster et conserver ses journaux, assurez-vous que [Monitoring//LogsCloudWatch/DeletionPolicy](#)n'est pas défini sur Delete dans la configuration du cluster. Sinon, remplacez la valeur de ce champ par et exécutez la `pcluster update-cluster` commande. Retain Exécutez ensuite `pcluster delete-cluster --cluster-name <cluster_name>` pour supprimer le cluster, mais conservez le groupe de journaux stocké sur Amazon CloudWatch.

## Journaux des nœuds interrompus

Si un nœud de calcul statique se termine de manière inattendue et ne CloudWatch possède aucun journal, vérifiez s'il AWS ParallelCluster a enregistré la sortie de console pour ce nœud de calcul sur le nœud principal du `/var/log/parallelcluster/compute_console_output` journal. Pour plus d'informations, consultez [Journaux clés pour le débogage](#).

Si le `/var/log/parallelcluster/compute_console_output` journal n'est pas disponible ou ne contient pas la sortie du nœud, utilisez le AWS CLI pour récupérer la sortie de console du

nœud défaillant. Connectez-vous au nœud principal du cluster et récupérez le nœud `instance-id` défaillant dans le `/var/log/parallelcluster/slurm_resume.log` fichier.

Récupérez la sortie de la console à l'aide de la commande suivante avec `instance-id` :

```
$ aws ec2 get-console-output --instance-id i-abcdef01234567890
```

Si un nœud de calcul dynamique s'arrête automatiquement après son lancement et ne CloudWatch possède aucun journal, soumettez une tâche qui active une action de dimensionnement du cluster. Attendez que l'instance échoue et récupérez le journal de la console de l'instance.

Connectez-vous au nœud principal du cluster et récupérez le nœud `instance-id` de calcul depuis le `/var/log/parallelcluster/slurm_resume.log` fichier.

Récupérez le journal de la console de l'instance à l'aide de la commande suivante :

```
$ aws ec2 get-console-output --instance-id i-abcdef01234567890
```

Le journal de sortie de la console peut vous aider à résoudre la cause première d'une défaillance d'un nœud de calcul lorsque le journal du nœud de calcul n'est pas disponible.

## AWS CloudFormation ressource personnalisée

À partir de AWS ParallelCluster la version 3.6.0, vous pouvez utiliser une ressource AWS ParallelCluster CloudFormation personnalisée dans une AWS CloudFormation pile. La ressource personnalisée est une pile AWS ParallelCluster hébergée. De cette façon, vous pouvez CloudFormation configurer et gérer vos clusters. Par exemple, vous pouvez configurer les ressources externes du cluster telles que le réseau, le stockage partagé et l'infrastructure des groupes de sécurité dans une CloudFormation pile. De plus, vous pouvez gérer votre cluster à l'aide d'une CloudFormation infrastructure sous forme de pipeline de code.

Ajoutez une ressource AWS ParallelCluster personnalisée à votre CloudFormation modèle en procédant comme suit :

1. Ajoutez une pile de fournisseurs de ressources personnalisée détenue et hébergée par AWS ParallelCluster.
2. Référez la pile de fournisseurs dans votre CloudFormation modèle en tant que ressource personnalisée.

La pile de fournisseurs de ressources personnalisés gère les CloudFormation demandes et y répond. Par exemple, lorsque vous déployez votre CloudFormation stack, vous configurez et créez également un cluster. Pour mettre à jour un cluster, vous devez mettre à jour votre CloudFormation pile. Vous supprimez un cluster lorsque vous supprimez votre pile. Pour plus d'informations sur les ressources CloudFormation personnalisées, voir [Ressources personnalisées](#) dans le Guide de AWS CloudFormation l'utilisateur.

#### Warning

CloudFormation ne détecte pas la dérive des ressources personnalisées. CloudFormation À utiliser uniquement pour mettre à jour la configuration du cluster et pour supprimer un cluster. Vous pouvez utiliser la [pcluster](#) CLI ou le [AWS ParallelClusterINTERFACE UTILISATEUR](#) pour surveiller l'état du cluster ou pour mettre à jour le parc informatique, mais vous ne devez pas les utiliser pour mettre à jour la configuration du cluster ou pour supprimer le cluster.

#### Note

Nous vous recommandons d'ajouter une [protection contre la résiliation](#) à votre pile afin d'éviter tout retrait accidentel.

## Pile de fournisseurs hébergée par AWS ParallelCluster

La pile de fournisseurs de ressources personnalisée est formatée comme indiqué dans l'extrait de CloudFormation modèle suivant :

```
PclusterClusterProvider:
 Type: AWS::CloudFormation::Stack
 Properties:
 Parameters:
 CustomLambdaRole: # (Optional) RoleARN to override default
 AdditionalIamPolicies: # (Optional) comma-separated list of IAM policies to add
 TemplateURL: !Sub
 - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.${AWS::URLSuffix}/
parallelcluster/${Version}/templates/custom_resource/cluster.yaml
 - { Version: 3.7.0 }
```

## Propriétés

Paramètres :

CustomLambdaRole (facultatif) :

Rôle personnalisé autorisé à exécuter le cluster AWS Lambda qui crée et gère le cluster. Par défaut, le rôle utilise les mêmes politiques définies par défaut dans la [AWS ParallelCluster documentation](#).

AdditionalIamPolicies (facultatif) :

Liste séparée par des virgules des noms de ressources Amazon (ARN) supplémentaires relatifs à la politique IAM à ajouter au rôle utilisé par le Lambda. Ceci n'est utilisé que si a CustomLambdaRole n'est pas spécifié et peut être laissé vide.

Si vous avez besoin de politiques supplémentaires pour le nœud principal, les nœuds de calcul ou pour l'accès à un compartiment Amazon S3, ajoutez-les à la AdditionalIamPolicy propriété CustomLambdaRole or.

Pour plus d'informations sur les politiques par défaut, consultez [AWS Identity and Access Management autorisations dans AWS ParallelCluster](#).

TemplateURL (obligatoire) :

URL du fichier de ressources AWS ParallelCluster personnalisé.

Sorties :

ServiceToken:

Une valeur qui peut être utilisée comme ServiceToken propriété de ressource personnalisée. Une ressource personnalisée ServiceToken indique où les demandes sont AWS CloudFormation envoyées. Il s'agit d'une entrée obligatoire pour une ressource de cluster que vous incluez dans votre AWS CloudFormation modèle.

LogGroupArn:

L'ARN du fichier CloudWatch LogGroup auquel la ressource sous-jacente se connecte.

LambdaLayerArn:

L'ARN de la couche Lambda utilisée pour exécuter AWS ParallelCluster des opérations.



## Ressource de cluster

La ressource du CloudFormation cluster est formatée comme indiqué dans l'extrait de CloudFormation modèle suivant :

```
PclusterCluster:
 Type: Custom::PclusterCluster
 Properties:
 ServiceToken: !GetAtt [PclusterClusterProvider , Outputs.ServiceToken]
 ClusterName: !Sub 'c-${AWS::StackName}' # Must be different from StackName
 ClusterConfiguration:
 # Your Cluster Configuration
```

### Propriétés

#### ServiceToken:

La ServiceToken sortie de la pile du AWS ParallelCluster fournisseur.

#### ClusterName:

Nom du cluster à créer et à gérer. Le nom ne doit pas correspondre au nom de la CloudFormation pile. Le nom ne peut pas être modifié une fois le cluster créé.

#### ClusterConfiguration:

Le fichier YAML de configuration du cluster, comme décrit dans [Fichier de configuration du cluster](#). Cependant, vous pouvez utiliser les CloudFormation constructions habituelles, telles que les [fonctions intrinsèques](#).

#### DeletionPolicy:

Définit s'il faut supprimer le cluster lorsque la pile racine est supprimée. L'argument par défaut est Delete.

#### Conserver :

Conservez le cluster si la ressource personnalisée est supprimée.

#### Note

Pour que le cluster conservé continue de fonctionner, les ressources dépendantes du cluster, telles que le stockage et le réseau, doivent avoir une politique de suppression définie pour être conservées.

## Supprimer :

Supprimez le cluster si la ressource personnalisée est supprimée.

## **Fn::GetAtt** valeurs de retour :

La fonction `Fn::GetAtt` intrinsèque renvoie une valeur pour un attribut spécifié d'un type. Pour plus d'informations sur l'utilisation de `Fn::GetAtt` intrinsèque cette fonction, consultez [Fn::GetAtt](#).

### ClusterProperties:

Les valeurs issues de l'[pcluster describe-cluster](#) opération.

### Messages de validation :

Chaîne contenant tous les messages de validation survenus lors de la dernière opération de création ou de mise à jour.

### logGroupName:

Nom du groupe de journaux utilisé pour la journalisation des opérations du cluster Lambda. Les événements du journal sont conservés pendant 90 jours et le groupe de journaux est conservé après la suppression du cluster.

## Exemple **Fn::GetAtt** :

```
Provide the public IP address of the head node as an output of a stack
Outputs:
 HeadNodeIp:
 Description: The public IP address of the head node
 Value: !GetAtt [PclusterCluster, headNode.publicIpAddress]
```

Exemple : CloudFormation modèle simple et complet avec une ressource AWS ParallelCluster personnalisée :

```
AWSTemplateFormatVersion: '2010-09-09'
Description: > AWS ParallelCluster CloudFormation Template

Parameters:
 HeadNodeSubnet:
 Description: Subnet where the HeadNode will run
 Type: AWS::EC2::Subnet::Id
```

```
ComputeSubnet:
 Description: Subnet where the Compute Nodes will run
 Type: AWS::EC2::Subnet::Id

KeyName:
 Description: KeyPair to login to the head node
 Type: AWS::EC2::KeyPair::KeyName

Resources:
 PclusterClusterProvider:
 Type: AWS::CloudFormation::Stack
 Properties:
 TemplateURL: !Sub
 - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.
 ${AWS::URLSuffix}/parallelcluster/${Version}/templates/custom_resource/cluster.yaml
 - { Version: 3.7.0 }

 PclusterCluster:
 Type: Custom::PclusterCluster
 Properties:
 ServiceToken: !GetAtt [PclusterClusterProvider , Outputs.ServiceToken]
 ClusterName: !Sub 'c-${AWS::StackName}'
 ClusterConfiguration:
 Image:
 Os: alinux2
 HeadNode:
 InstanceType: t2.medium
 Networking:
 SubnetId: !Ref HeadNodeSubnet
 Ssh:
 KeyName: !Ref KeyName
 Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: queue0
 ComputeResources:
 - Name: queue0-cr0
 InstanceType: t2.micro
 Networking:
 SubnetIds:
 - !Ref ComputeSubnet

Outputs:
```

```
HeadNodeIp:
 Description: The Public IP address of the HeadNode
 Value: !GetAtt [PclusterCluster, headNode.publicIpAddress]
ValidationMessages:
 Description: Any warnings from cluster create or update operations.
 Value: !GetAtt PclusterCluster.validationMessages
```

Pour en savoir plus sur l'utilisation de la ressource CloudFormation AWS ParallelCluster personnalisée, consultez [Création d'un cluster avec AWS CloudFormation](#).

## Opérations du cluster

Lorsqu'une ressource personnalisée de cluster est ajoutée à une CloudFormation pile, CloudFormation vous pouvez effectuer les opérations de cluster suivantes :

- CloudFormation crée un cluster dans une nouvelle pile séparée lorsqu'il déploie une pile qui inclut la ressource AWS ParallelCluster personnalisée.
- Si vous mettez à jour la configuration du cluster définie dans la pile, conformément aux politiques de mise à jour de la configuration, CloudFormation met à jour le cluster. Le fournisseur de ressources AWS ParallelCluster personnalisées n'arrête pas le parc informatique avant de mettre à jour le cluster. Nous vous recommandons d'utiliser ce [QueueUpdateStrategy](#) paramètre pour les mises à jour du cluster. Ainsi, vous pouvez éviter de passer des `pcluster update-compute-fleet` appels explicites avant et après les mises à jour lorsque vous utilisez la ressource AWS ParallelCluster personnalisée.
- Si vous supprimez la pile, le cluster est supprimé.

## Résolution des problèmes liés aux piles qui incluent la ressource AWS ParallelCluster personnalisée

À l'aide d'une ressource AWS ParallelCluster personnalisée, CloudFormation déploie un cluster à partir d'une nouvelle pile distincte. Vous pouvez surveiller la création de clusters en suivant les étapes suivantes :

1. Accédez CloudFormation à AWS Management Console et choisissez Stacks dans le volet de navigation.
2. Choisissez la pile portant le nom que vous avez défini pour le nom du cluster.

3. Si l'état de la pile est défini `ROLLBACK_COMPLETE`, une erreur s'est produite lors de la création du cluster.
4. Choisissez `Stack details`, puis cliquez sur l'onglet `Events`.
5. Recherchez dans `Events` on Logical ID le nom que vous avez défini pour le nom du cluster. Il a un `Status reason` qui donne la raison d'un problème.
6. Vous pouvez également choisir le menu déroulant `Piles`, puis `Supprimer` pour voir la liste des piles supprimées. Sélectionnez la pile portant le nom du cluster et consultez les événements pour plus de détails.
7. Pour afficher le résultat du fournisseur de ressources personnalisées qui gère le cluster, sélectionnez la pile dont la description est « Ressource personnalisée du AWS ParallelCluster cluster ». Choisissez l'onglet `Ressources`, recherchez la ressource avec un ID **`PclusterCfnFunctionLogGroup`** logique et suivez le lien indiqué. Affichez les flux de journaux qui affichent la sortie de débogage Lambda.
8. Pour résoudre les problèmes liés au cluster, voir [AWS ParallelCluster résolution des problèmes](#).

## Elastic Fabric Adapter

Elastic Fabric Adapter (EFA) est un périphérique réseau doté de fonctionnalités de contournement du système d'exploitation pour les communications réseau à faible latence avec d'autres instances sur le même sous-réseau. EFA est exposé à l'aide de Libfabric et peut être utilisé par des applications utilisant l'interface MPI (Messaging Passing Interface).

Pour utiliser EFA avec AWS ParallelCluster un Slurm planificateur, définissez [SlurmQueues//ComputeResourcesEfa/Enabled](#) sur `true`

Pour consulter la liste des instances Amazon EC2 qui prennent en charge l'EFA, consultez la section [Types d'instances pris en charge](#) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Nous vous recommandons d'exécuter vos instances compatibles EFA dans un groupe de placement. Les instances sont ainsi lancées dans un groupe à faible latence dans une seule zone de disponibilité. Pour plus d'informations sur la configuration des groupes de placement avec AWS ParallelCluster, consultez [SlurmQueues/Networking/PlacementGroup](#).

Pour plus d'informations, consultez [Elastic Fabric Adapter](#) dans le guide de l'utilisateur Amazon EC2 et [Scale HPC workloads with Elastic Fabric Adapter](#) sur le AWS blog AWS ParallelCluster Open Source.

**Note**

Elastic Fabric Adapter (EFA) n'est pas pris en charge dans différentes zones de disponibilité. Pour plus d'informations, consultez [Scheduling/SlurmQueues/Networking/SubnetIds](#).

**Note**

Par défaut, les Ubuntu distributions activent la protection ptrace (suivi des processus). ptracela protection est désactivée afin que Libfabric fonctionne correctement. Pour plus d'informations, consultez la section [Désactiver la protection ptrace](#) dans le guide de l'utilisateur Amazon EC2.

## Activez Intel MPI

Intel MPI est disponible sur le AWS ParallelCluster AMIs.

**Note**

Pour utiliser IntelMPI, vous devez connaître et accepter les termes de la [licence logicielle simplifiée Intel](#).

Par défaut, Open MPI est placé sur le chemin. Pour activer Intel MPI au lieu d'OpenMPI, vous devez d'abord charger le MPI module Intel. Ensuite, vous devez installer la dernière version en utilisant `module load intelmpi`. Le nom exact du module change avec chaque mise à jour. Pour voir quels modules sont disponibles, exécutez `module avail`. La sortie est la suivante.

```
$ module avail
-----/usr/share/Modules/modulefiles

dot modules
libfabric-aws/1.16.0~amzn3.0 null
module-git openmpi/4.1.4
module-info use.own

-----/opt/intel/mpi/2021.6.0/modulefiles

```

```
intelmpi
```

Pour charger un module, exécutez `module load modulename`. Vous pouvez ajouter ceci au script utilisé pour exécuter `mpirun`.

```
$ module load intelmpi
```

Pour savoir quels modules sont chargés, exécutez `module list`.

```
$ module list
Currently Loaded Modulefiles:
 1) intelmpi
```

Pour vérifier qu'Intel MPI est activé, exécutez `mpirun --version`.

```
$ mpirun --version
Intel(R) MPI Library for Linux* OS, Version 2021.6 Build 20220227 (id: 28877f3f32)
Copyright 2003-2022, Intel Corporation.
```

Une fois le MPI module Intel chargé, plusieurs chemins sont modifiés pour utiliser les MPI outils Intel. Pour exécuter du code compilé par les MPI outils Intel, chargez d'abord le MPI module Intel.

#### Note

Intel MPI n'est pas compatible avec les instances AWS basées sur Graviton.

#### Note

Avant AWS ParallelCluster la version 2.5.0, Intel MPI n'était pas disponible AWS ParallelCluster AMIs dans les régions de Chine (Pékin) et de Chine (Ningxia).

## AWS ParallelCluster API

Qu'est-ce que AWS ParallelCluster l'API ?

AWS ParallelCluster L'API est une application sans serveur qui, une fois déployée sur votre ordinateur Compte AWS, fournit un accès programmatique aux AWS ParallelCluster fonctionnalités via une API.

AWS ParallelCluster L'API est distribuée sous la forme d'un [AWS CloudFormation](#) modèle autonome qui inclut un point de terminaison [Amazon API Gateway](#), qui expose les AWS ParallelCluster fonctionnalités, et une [AWS Lambda](#) fonction, qui prend en charge le traitement des fonctionnalités invoquées.

L'image suivante montre un schéma d'architecture de haut niveau de l'infrastructure d' AWS ParallelCluster API.

## AWS ParallelCluster Documentation de l'API

Le fichier de spécification OpenAPI décrivant l' AWS ParallelCluster API peut être téléchargé à l'adresse suivante :

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/
parallelcluster/<VERSION>/api/ParallelCluster.openapi.yaml
```

[À partir du fichier de spécification OpenAPI, vous pouvez générer de la documentation pour l' AWS ParallelCluster API en utilisant l'un des nombreux outils disponibles tels que Swagger UI ou Redoc.](#)

### Comment déployer une AWS ParallelCluster API

Pour déployer AWS ParallelCluster l'API, vous devez être administrateur du Compte AWS.

Le modèle utilisé pour déployer l'API est disponible à l'adresse URL suivante :

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/
parallelcluster/<VERSION>/api/parallelcluster-api.yaml
```

où se **<REGION>** trouve l' Région AWS endroit où l'API doit être déployée et quelle **<VERSION>** est la AWS ParallelCluster version (par exemple 3.7.0).

AWS Lambda traite les fonctionnalités invoquées par l'API en utilisant une interface de couche Lambda avec le [AWS ParallelClusterAPI de bibliothèque Python](#)



**⚠ Warning**

Tout utilisateur du Compte AWS, qui dispose d'un accès privilégié aux AWS Lambda services Amazon API Gateway, hérite automatiquement des autorisations nécessaires pour administrer les ressources de AWS ParallelCluster l'API.

## Déployez avec AWS CLI

Configurez les AWS informations d'identification à utiliser avec la CLI si ce n'est déjà fait.

```
$ aws configure
```

Exécutez les commandes suivantes pour déployer l'API :

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name> # This can be any name
$ VERSION=3.7.0
$ aws cloudformation create-stack \
 --region ${REGION} \
 --stack-name ${API_STACK_NAME} \
 --template-url https://${REGION}-aws-parallelcluster.s3.${REGION}.amazonaws.com/
parallelcluster/${VERSION}/api/parallelcluster-api.yaml \
 --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND
$ aws cloudformation wait stack-create-complete --stack-name ${API_STACK_NAME} --region
${REGION}
```


### Personnalisez votre déploiement

Vous pouvez personnaliser le déploiement de l'API en utilisant les AWS CloudFormation paramètres exposés par le modèle. Pour configurer la valeur d'un paramètre lors du déploiement via la CLI, l'option suivante peut être utilisée : `--parameters ParameterKey=KeyName,ParameterValue=Value`

Les paramètres suivants sont facultatifs :

- Région : utilisez le `Region` paramètre pour spécifier si l'API est capable de contrôler les ressources dans toutes Régions AWS (par défaut) ou dans une seule Région AWS. Définissez cette valeur sur Région AWS l'API sur laquelle l'API est déployée afin de restreindre l'accès.


- `ParallelClusterFunctionRole`- Cela remplace le rôle IAM attribué à la AWS Lambda fonction AWS ParallelCluster implémentant les fonctionnalités. Le paramètre accepte l'ARN d'un rôle IAM. Ce rôle doit être configuré pour être utilisé AWS Lambda en tant que principal IAM.
- `CustomDomainName`, `CustomDomainCertificate`, `CustomDomainHostedZoneId` - Utilisez ces paramètres pour définir un domaine personnalisé pour le point de terminaison Amazon API Gateway. `CustomDomainName` est le nom du domaine à utiliser, `CustomDomainCertificate` l'ARN d'un certificat AWS géré pour ce nom de domaine et `CustomDomainHostedZoneId` l'ID de la zone hébergée [Amazon Route 53](#) dans laquelle vous souhaitez créer des enregistrements.

 Warning

Vous pouvez configurer des paramètres de domaine personnalisés pour appliquer une version minimale du protocole TLS (Transport Layer Security) pour l'API. Pour plus d'informations, consultez [Choisir une version minimale de TLS pour un domaine personnalisé dans API Gateway](#).

- `EnableIamAdminAccess`- Par défaut, les opérations de l' AWS ParallelCluster API de traitement des AWS Lambda fonctions sont configurées avec un rôle IAM qui empêche tout accès IAM privilégié (`EnableIamAdminAccess=false`). Cela rend l'API incapable de traiter les opérations qui nécessitent la création de rôles ou de politiques IAM. De ce fait, la création de clusters ou d'images personnalisées n'est réussie que lorsque des rôles IAM sont fournis en entrée dans le cadre de la configuration des ressources.

Lorsque cette `EnableIamAdminAccess` option est définie `true` sur l' AWS ParallelCluster API, des autorisations sont accordées pour gérer la création des rôles IAM nécessaires au déploiement de clusters ou à la génération d'AMI personnalisées.

 Warning

La définition de ce paramètre sur `true` accorde des privilèges d'administrateur IAM aux AWS ParallelCluster opérations de traitement des AWS Lambda fonctions.

Reportez-vous à [AWS ParallelCluster exemples de politiques utilisateur pour la gestion des ressources IAM](#) pour plus de détails sur les fonctionnalités qui peuvent être déverrouillées lors de l'activation de ce mode.

- **PermissionsBoundaryPolicy**- Ce paramètre facultatif accepte un ARN de politique IAM existant qui sera défini comme limite d'autorisations pour tous les rôles IAM créés par l'infrastructure API PC et comme condition des autorisations IAM administratives afin que seuls les rôles dotés de cette politique puissent être créés par l'API PC.

Reportez-vous à [Mode PermissionsBoundary](#) pour plus de détails sur les restrictions imposées par ce mode.

- **CreateApiUserRole**- Par défaut, le déploiement de l' AWS ParallelCluster API inclut la création d'un rôle IAM défini comme le seul rôle autorisé à appeler l'API. Le point de terminaison Amazon API Gateway est configuré avec une politique basée sur les ressources pour accorder l'autorisation d'appel uniquement à l'utilisateur créé. Pour modifier cela, définissez `CreateApiUserRole=false` puis accordez l'accès à l'API aux utilisateurs IAM sélectionnés. Pour plus d'informations, consultez la section [Contrôler l'accès pour appeler une API](#) dans le manuel Amazon API Gateway Developer Guide.

#### Warning

Lorsque `CreateApiUserRole=true` accès au point de terminaison de l'API n'est pas limité par les politiques de ressources d'Amazon API Gateway, tous les rôles IAM disposant d'`execute-api:Invoke` autorisations illimitées peuvent accéder aux fonctionnalités. AWS ParallelCluster Pour plus d'informations, consultez la section [Contrôle de l'accès à une API à l'aide des politiques de ressources](#) d'API Gateway dans le Guide du développeur d'API Gateway.

#### Warning

`ParallelClusterApiUserRole` est autorisé à appeler toutes les opérations AWS ParallelCluster d'API. Pour restreindre l'accès à un sous-ensemble de ressources d'API, consultez la [méthode Control who can call an API Gateway API gateway avec des politiques IAM](#) dans le guide du développeur d'API Gateway.

- **IAM RoleAndPolicyPrefix** - Ce paramètre facultatif accepte une chaîne de 10 caractères maximum qui sera utilisée comme préfixe pour les rôles IAM et les politiques créés dans le cadre de l'infrastructure d'API PC.

## Mettre à jour l'API

Mise à niveau vers une AWS ParallelCluster version plus récente

Option 1 : supprimez l'API existante en supprimant la AWS CloudFormation pile correspondante et en déployant la nouvelle API comme indiqué ci-dessus.

Option 2 : mettez à jour l'API existante en exécutant les commandes suivantes :

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name> # This needs to correspond to the existing API stack
name
$ VERSION=3.7.0
$ aws cloudformation update-stack \
 --region ${REGION} \
 --stack-name ${API_STACK_NAME} \
 --template-url https://${REGION}-aws-parallelcluster.s3.${REGION}.amazonaws.com/
parallelcluster/${VERSION}/api/parallelcluster-api.yaml \
 --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND
$ aws cloudformation wait stack-update-complete --stack-name ${API_STACK_NAME} --region
${REGION}
```

## Invocation de AWS ParallelCluster l'API

Le point de terminaison AWS ParallelCluster Amazon API Gateway est configuré avec un [type AWS\\_IAM d'autorisation](#) et nécessite que toutes les demandes soient signées au format SigV4 avec des informations d'identification IAM valides ([référence API : envoi de requêtes http](#)).

Lorsqu'elles sont déployées avec les paramètres par défaut, les autorisations d'appel d'API ne sont accordées qu'à l'utilisateur IAM par défaut créé avec l'API.

Pour récupérer l'ARN de l'utilisateur IAM par défaut, exécutez :

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name>
$ aws cloudformation describe-stacks --region ${REGION} --stack-name ${API_STACK_NAME}
 --query "Stacks[0].Outputs[?OutputKey=='ParallelClusterApiUserRole'].OutputValue" --
output text
```

Pour obtenir des informations d'identification temporaires pour l'utilisateur IAM par défaut, exécutez la AssumeRole commande [STS](#).

Vous pouvez récupérer le point de terminaison de l' AWS ParallelCluster API en exécutant la commande suivante :

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name>
$ aws cloudformation describe-stacks --region ${REGION} --stack-name ${API_STACK_NAME}
 --query "Stacks[0].Outputs[?OutputKey=='ParallelClusterApiInvokeUrl'].OutputValue" --
output text
```

L' AWS ParallelCluster API peut être invoquée par n'importe quel client HTTP conforme aux spécifications OpenAPI disponibles ici :

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/
parallelcluster/<VERSION>/api/ParallelCluster.openapi.yaml
```

Les demandes doivent être signées au format SigV4, comme indiqué [ici](#).

À l'heure actuelle, nous ne proposons aucune implémentation client API officielle. [Cependant, les clients d'API peuvent être facilement générés à partir du modèle OpenAPI à l'aide d'OpenAPI Generator](#). Une fois le client généré, la signature SigV4 doit être ajoutée si elle n'est pas fournie par défaut.

Une implémentation de référence pour un client d'API Python se trouve dans le [AWS ParallelCluster référentiel](#). Pour en savoir plus sur l'utilisation du client API Python, consultez le [Utilisation de l' AWS ParallelCluster API](#) didacticiel.

[Pour mettre en œuvre des mécanismes de contrôle d'accès plus avancés, tels qu'Amazon Cognito ou Lambda Authorizers, ou pour protéger davantage l'API avec AWS WAF des clés d'API, consultez la documentation Amazon API Gateway.](#)

#### Warning

Un utilisateur IAM autorisé à invoquer l' AWS ParallelCluster API peut contrôler indirectement toutes les AWS ressources gérées par AWS ParallelCluster le Compte AWS. Cela inclut la création de AWS ressources que l'utilisateur ne peut pas contrôler directement en raison des restrictions de la politique IAM de l'utilisateur. Par exemple, la création d'un AWS ParallelCluster cluster, en fonction de sa configuration, peut inclure le déploiement d'instances Amazon EC2, d'Amazon Route 53, de systèmes de fichiers Amazon Elastic File System, de systèmes de fichiers Amazon FSx, de rôles IAM et de ressources provenant

d'autres services AWS applications sur AWS ParallelCluster lesquelles l'utilisateur n'a peut-être aucun contrôle direct.

### Warning

Lors de la création d'un cluster avec des paramètres `AdditionalIamPolicies` spécifiés dans la configuration, les politiques supplémentaires doivent correspondre à l'un des modèles suivants :

```
- !Sub arn:${AWS::Partition}:iam::${AWS::AccountId}:policy/parallelcluster*
- !Sub arn:${AWS::Partition}:iam::${AWS::AccountId}:policy/parallelcluster/*
- !Sub arn:${AWS::Partition}:iam::aws:policy/CloudWatchAgentServerPolicy
- !Sub arn:${AWS::Partition}:iam::aws:policy/AmazonSSMManagedInstanceCore
- !Sub arn:${AWS::Partition}:iam::aws:policy/AWSBatchFullAccess
- !Sub arn:${AWS::Partition}:iam::aws:policy/AmazonS3ReadOnlyAccess
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/AWSBatchServiceRole
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
AmazonEC2SpotFleetTaggingRole
- !Sub arn:${AWS::Partition}:iam::aws:policy/EC2InstanceProfileForImageBuilder
- !Sub arn:${AWS::Partition}:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole
```

Si vous avez besoin d'autres politiques supplémentaires, vous pouvez effectuer l'une des opérations suivantes :

- Modifiez le `DefaultParallelClusterIamAdminPolicy` dans :

```
https://<REGION>-aws-parallelcluster.s3.<REGION>.amazonaws.com/
parallelcluster/<VERSION>/api/parallelcluster-api.yaml
```

Ajoutez la politique dans la `ArnLike/iam:PolicyARN` section.

- Omettez de spécifier des politiques `AdditionalIamPolicies` dans le fichier de configuration et ajoutez manuellement des politiques au rôle d' AWS ParallelCluster instance créé dans le cluster.

## Accès aux journaux et aux métriques de l'API

Les journaux d'API sont publiés sur Amazon CloudWatch avec une durée de conservation de 30 jours. Pour récupérer le LogGroup nom associé à un déploiement d'API, exécutez la commande suivante :

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name>
$ aws cloudformation describe-stacks --region ${REGION} --
stack-name ${API_STACK_NAME} --query "Stacks[0].Outputs[?
OutputKey=='ParallelClusterLambdaLogGroup'].OutputValue" --output text
```

Les métriques Lambda, les journaux et les journaux de [AWS X-Ray](#) suivi sont également accessibles via la console Lambda. Pour récupérer l'ARN de la fonction Lambda associée à un déploiement d'API, exécutez la commande suivante :

```
$ REGION=<region>
$ API_STACK_NAME=<stack-name>
$ aws cloudformation describe-stacks --region ${REGION} --stack-name ${API_STACK_NAME}
--query "Stacks[0].Outputs[?OutputKey=='ParallelClusterLambdaArn'].OutputValue" --
output text
```

## AWS ParallelCluster pour Terraform

À partir de la AWS ParallelCluster version 3.8.0, vous pouvez déployer des clusters et des images personnalisées à l'aide de [Terraform](#). Pour commencer à utiliser cette fonctionnalité, consultez [Terraform Provider AWS ParallelCluster depuis](#) le registre Terraform.

### Note

[ParallelCluster L'API](#) doit être déployée dans votre compte pour utiliser le fournisseur.

Utilisez le tableau suivant pour déterminer la compatibilité entre le fournisseur et les AWS ParallelCluster versions :

| Version du fournisseur | AWS ParallelCluster version |
|------------------------|-----------------------------|
| 1.0.0                  | 3,8,0+                      |

Consultez des [exemples](#) d'utilisation du fournisseur.

Pour une expérience encore plus fluide, utilisez le [module Terraform officiel de Terraform AWS ParallelCluster Registry](#). Le module permet de déployer :

1. ParallelCluster API
2. ParallelCluster clusters définis avec le fichier de configuration YAML et HCL
3. Infrastructure réseau requise par un ParallelCluster cluster

Consultez des [exemples](#) d'utilisation du module.

## Connectez-vous au nœud principal via NICE DCV

NICE DCV est une technologie de visualisation à distance qui permet aux utilisateurs de se connecter en toute sécurité à des applications 3D gourmandes en graphismes hébergées sur un serveur distant à hautes performances. Pour plus d'informations, consultez [NICE DCV](#).

Le logiciel NICE DCV est automatiquement installé sur le nœud principal et peut être activé en utilisant la [Dcv](#) section de configuration. [HeadNode](#)

```
HeadNode:
 Dcv:
 Enabled: true
```

De cette façon, AWS ParallelCluster place le nœud principal `/home/<DEFAULT_AMI_USER>` dans le [dossier de stockage du serveur DCV](#). Pour plus d'informations sur les paramètres de configuration de NICE DCV, consultez [HeadNode/](#). [Dcv](#) Pour vous connecter à la session DCV NICE, utilisez la [pcluster dcv-connect](#) commande.

## Certificat HTTPS NICE DCV

NICE DCV génère automatiquement un certificat auto-signé pour sécuriser le trafic entre le client NICE DCV et le serveur NICE DCV.

Pour remplacer le certificat NICE DCV autosigné par défaut par un autre certificat, connectez-vous d'abord au nœud principal. Ensuite, copiez le certificat et la clé dans le dossier `/etc/dcv` avant d'exécuter la commande [pcluster dcv-connect](#).



Pour plus d'informations, consultez la section [Modification du certificat TLS](#) dans le guide de l'administrateur de NICE DCV.

## Licence NICE DCV

Le serveur NICE DCV ne nécessite pas de serveur de licences lorsqu'il est exécuté sur des instances Amazon EC2. Cependant, le serveur NICE DCV doit se connecter régulièrement à un compartiment Amazon S3 pour déterminer si une licence valide est disponible.

AWS ParallelCluster ajoute automatiquement les autorisations requises à la politique IAM du nœud principal. Lorsque vous utilisez une politique d'instance IAM personnalisée, utilisez les autorisations décrites dans [NICE DCV sur Amazon EC2](#) dans le guide de l'administrateur NICE DCV.

Pour obtenir des conseils de dépannage, veuillez consulter [Résolution des problèmes dans NICE DCV](#).

## Utiliser `pcluster update-cluster`

Dans AWS ParallelCluster 3.x, [pcluster update-cluster](#) analyse les paramètres utilisés pour créer le cluster actuel et les paramètres du fichier de configuration pour détecter les problèmes. Si des problèmes sont découverts, ils sont signalés et les étapes à suivre pour les résoudre sont affichées. Par exemple, si le calcul [InstanceType](#) est modifié, le parc informatique doit être arrêté avant qu'une mise à jour puisse avoir lieu. Ce problème est signalé lorsqu'il est découvert. Si aucun problème de blocage n'est découvert, le processus de mise à jour est lancé et les modifications sont signalées.

Vous pouvez utiliser le [pcluster update-cluster --dryrun](#) option pour voir les modifications avant leur exécution. Pour plus d'informations, consultez [Exemples pcluster update-cluster](#).

Pour obtenir des conseils de résolution des problèmes, consultez [AWS ParallelCluster résolution des problèmes](#).

## Politique de mise à jour : définitions

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

Après avoir modifié ce paramètre, le cluster peut être mis à jour à l'aide de [pcluster update-cluster](#).

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

Une fois ce paramètre modifié, le cluster ne peut pas être mis à jour. Vous devez rétablir les paramètres du cluster d'origine et créer un nouveau cluster avec les paramètres mis à jour. Vous pouvez supprimer le cluster d'origine ultérieurement. Pour créer le nouveau cluster, utilisez [pcluster create-cluster](#). Pour supprimer le cluster d'origine, utilisez [pcluster delete-cluster](#).

Politique de mise à jour : ce paramètre n'est pas analysé lors d'une mise à jour.

Ces paramètres peuvent être modifiés et le cluster mis à jour à l'aide de [pcluster update-cluster](#).

Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.

Ces paramètres ne peuvent pas être modifiés tant que le parc informatique existe. La modification doit être annulée ou le parc informatique doit être arrêté (utilisation [pcluster update-compute-fleet](#)). Une fois le parc informatique arrêté, vous pouvez mettre à jour le cluster ([pcluster update-cluster](#)) pour activer les modifications. Par exemple, si vous utilisez un Slurm planificateur avec [SlurmQueues/- ComputeResourcesName/MinCount](#) > 0, une flotte de calcul est démarrée.

Politique de mise à jour : le parc informatique et les nœuds de connexion doivent être arrêtés pour que ce paramètre soit modifié pour une mise à jour.

Ces paramètres ne peuvent pas être modifiés tant que le parc informatique existe ou si les nœuds de connexion sont utilisés. Soit la modification doit être annulée, soit le parc de calcul et les nœuds de connexion doivent être arrêtés (le parc de calcul peut être arrêté en utilisant [pcluster update-compute-fleet](#)). Une fois le parc informatique et les nœuds de connexion arrêtés, vous pouvez mettre à jour le cluster ([pcluster update-cluster](#)) pour activer les modifications.

Politique de mise à jour : ce paramètre ne peut pas être réduit lors d'une mise à jour.


Ces paramètres peuvent être modifiés, mais ils ne peuvent pas être diminués. Si ces paramètres doivent être réduits, vous devez rétablir les paramètres du cluster d'origine et créer un nouveau cluster avec les paramètres mis à jour. Vous pouvez supprimer le cluster d'origine ultérieurement. Pour créer le nouveau cluster, utilisez [pcluster create-cluster](#). Pour supprimer le cluster d'origine, utilisez [pcluster delete-cluster](#).

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée. Si vous forcez la mise à jour, la nouvelle valeur sera ignorée et l'ancienne valeur sera utilisée.

Une fois ce paramètre modifié, le cluster ne peut pas être mis à jour. Vous devez rétablir les paramètres du cluster d'origine et créer un nouveau cluster avec les paramètres mis à jour. Vous pouvez supprimer le cluster d'origine ultérieurement. Pour créer le nouveau cluster, utilisez [pcluster create-cluster](#). Pour supprimer le cluster d'origine, utilisez [pcluster delete-cluster](#).

Politique de mise à jour : le parc informatique doit être arrêté ou [QueueUpdateStrategy](#) défini pour que ce paramètre soit modifié en vue d'une mise à jour.

Ces paramètres peuvent être modifiés. Le parc informatique doit être arrêté (utilisé [pcluster update-compute-fleet](#)) ou [QueueUpdateStrategy](#) doit être défini. Une fois le parc informatique arrêté ou [QueueUpdateStrategy](#) défini, vous pouvez mettre à jour le cluster ([pcluster update-cluster](#)) pour activer les modifications.

 Note

Cette politique de mise à jour est prise en charge à partir de AWS ParallelCluster la version 3.2.0.

Politique de mise à jour : pour ce paramètre de valeurs de liste, une nouvelle valeur peut être ajoutée lors d'une mise à jour ou le parc informatique doit être arrêté lors de la suppression d'une valeur existante.

Une nouvelle valeur pour ces paramètres peut être ajoutée lors d'une mise à jour. Après avoir ajouté une nouvelle valeur à la liste, le cluster peut être mis à jour à l'aide de ([pcluster update-cluster](#)).

Pour supprimer une valeur existante de la liste, le parc informatique doit être arrêté (en utilisant [pcluster update-compute-fleet](#)).

Par exemple, si vous utilisez un Slurm planificateur et que vous ajoutez un nouveau type d'instance à [Instances/ InstanceType](#), vous pouvez mettre à jour le cluster sans arrêter le parc informatique. Pour supprimer un type d'instance existant dans [Instances/ InstanceType](#), le parc de calcul doit d'abord être arrêté (à l'aide de [update-compute-fleetpcluster](#)).

**Note**

Cette politique de mise à jour est prise en charge à partir de AWS ParallelCluster la version 3.2.0.

Politique de mise à jour : la réduction de la taille d'une file d'attente nécessite l'arrêt du parc informatique ou [QueueUpdateStrategy](#) doit être définie sur TERMINATE pour que ce paramètre soit modifié pour une mise à jour.

Ces paramètres peuvent être modifiés, mais si la modification a pour effet de réduire la taille de la file d'attente, le parc de calcul doit être arrêté (à l'aide de `pcluster update-compute-fleet`) ou [QueueUpdateStrategy](#) doit être défini sur TERMINATE. Une fois le parc informatique arrêté ou [QueueUpdateStrategy](#) configuré sur TERMINATE, vous pouvez mettre à jour le cluster ([pcluster update-cluster](#)) pour activer les modifications.

Le paramètre TERMINATE défini lors du redimensionnement de la capacité du cluster ne mettra fin qu'aux nœuds figurant au bas de la liste des nœuds et laissera intacts tous les autres nœuds de la même partition.

Par exemple, si la capacité initiale du cluster est égale à `MinCount = 5` et `MaxCount = 10`, les nœuds sont `st-[1-5]` ; `dy-[1-5]`. Lors du redimensionnement du cluster sur `MinCount = 3` et `MaxCount = 5`, la nouvelle capacité du cluster sera composée par les nœuds `st-[1-3]` ; `dy-[1-2]`, qui ne seront pas modifiés lors de la mise à jour. Seuls les nœuds `st-[4-5]` ; `dy-[3-5]` seront interrompus lors de la mise à jour.

Les modifications suivantes sont prises en charge et ne nécessitent pas l'arrêt du parc de calcul ni le [QueueUpdateStrategy](#) réglage sur TERMINATE :

- Un nouveau [SlurmQueue](#) est ajouté
- Un nouveau [ComputeResource](#) est ajouté
- [MaxCount](#) est augmenté
- [MinCount](#) est augmenté et [MaxCount](#) est augmenté d'au moins le même montant

Remarque : Cette politique de mise à jour est prise en charge à partir de AWS ParallelCluster la version 3.9.0.

Politique de mise à jour : pour ce paramètre de valeurs de liste, le parc de calcul [QueueUpdateStrategy](#) doit être arrêté ou configuré pour ajouter une nouvelle valeur ; le parc de calcul doit être arrêté lors de la suppression d'une valeur existante.

Une nouvelle valeur pour ces paramètres peut être ajoutée lors d'une mise à jour. Le parc informatique doit être arrêté (utilisé [pcluster update-compute-fleet](#)) ou [QueueUpdateStrategy](#) doit être défini. Une fois le parc informatique arrêté ou [QueueUpdateStrategy](#) défini, vous pouvez mettre à jour le cluster ([pcluster update-cluster](#)) pour activer les modifications.

Pour supprimer une valeur existante de la liste, le parc informatique doit être arrêté (en utilisant [pcluster update-compute-fleet](#)).

#### Note

Cette politique de mise à jour est prise en charge à partir de AWS ParallelCluster la version 3.3.0.

Politique de mise à jour : tous les nœuds de calcul doivent être arrêtés pour la suppression d'un groupe de placement géré. Le parc informatique doit être arrêté ou [QueueUpdateStrategy](#) défini pour que ce paramètre soit modifié en vue d'une mise à jour.

Le parc informatique doit être arrêté (utilisé [pcluster update-compute-fleet](#)) afin de supprimer un groupe de placement géré. Si vous exécutez une mise à jour du cluster pour supprimer un groupe de placement géré avant d'arrêter le parc informatique, un message de configuration non valide est renvoyé et la mise à jour ne se poursuit pas. L'arrêt du parc informatique garantit qu'aucune instance n'est en cours d'exécution.

## Exemples [pcluster update-cluster](#)

Ces paramètres peuvent être modifiés, mais si la modification a pour effet de réduire la taille de la file d'attente, le parc de calcul doit être arrêté (à l'aide de [pcluster update-compute-fleet](#)) ou [QueueUpdateStrategy](#) doit être défini sur TERMINATE. Une fois le parc informatique arrêté ou [QueueUpdateStrategy](#) configuré sur TERMINATE, vous pouvez mettre à jour le cluster ([pcluster update-cluster](#)) pour activer les modifications.

- Cet exemple illustre une mise à jour avec certaines modifications autorisées et la mise à jour est lancée directement.

```
$ pcluster update-cluster --cluster-name cluster_name --cluster-config
~/.parallelcluster/test_cluster --region us-east-1
{
 "cluster": {
 "clusterName": cluster_name,
 "cloudformationStackStatus": "UPDATE_IN_PROGRESS",
 "cloudformationStackArn": stack_arn,
 "region": "us-east-1",
 "version": "3.7.0",
 "clusterStatus": "UPDATE_IN_PROGRESS"
 },
 "changeSet": [
 {
 "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
 "requestedValue": [
 "sg-0cd61884c4ad11234"
],
 "currentValue": [
 "sg-0cd61884c4ad16341"
]
 }
]
}
```

- Cet exemple illustre une mise à jour à sec avec certaines modifications autorisées. Dryrun est utile pour signaler le changement défini sans démarrer la mise à jour.

```
$ pcluster update-cluster --cluster-name cluster_name --cluster-config
~/.parallelcluster/test_cluster --region us-east-1 --dryrun true
{
 "message": "Request would have succeeded, but DryRun flag is set.",
 "changeSet": [
 {
 "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
 "requestedValue": [
 "sg-0cd61884c4ad11234"
],
 "currentValue": [
 "sg-0cd61884c4ad16341"
]
 }
]
}
```

```
}

```

- Cet exemple illustre une mise à jour avec certaines modifications qui bloquent la mise à jour.

```
$ pcluster update-cluster --cluster-name cluster_name --cluster-config
~/.parallelcluster/test_cluster --region us-east-1
{
 "message": "Update failure",
 "updateValidationErrors": [
 {
 "parameter": "HeadNode.Ssh.KeyName",
 "requestedValue": "mykey_2",
 "message": "Update actions are not currently supported for the 'KeyName'
parameter. Restore 'KeyName' value to 'jenkinsjun'. If you need this change, please
consider creating a new cluster instead of updating the existing one.",
 "currentValue": "mykey_1"
 },
 {
 "parameter": "Scheduling.SlurmQueues[queue1].ComputeResources[queue1-
t2micro].InstanceType",
 "requestedValue": "c4.xlarge",
 "message": "All compute nodes must be stopped. Stop the compute fleet with the
pcluster update-compute-fleet command",
 "currentValue": "t2.micro"
 },
 {
 "parameter": "SharedStorage[ebs1].MountDir",
 "requestedValue": "/my/very/very/long/shared_dir",
 "message": "Update actions are not currently supported for the 'MountDir'
parameter. Restore 'MountDir' value to '/shared'. If you need this change, please
consider creating a new cluster instead of updating the existing one.",
 "currentValue": "/shared"
 }
],
 "changeSet": [
 {
 "parameter": "HeadNode.Networking.AdditionalSecurityGroups",
 "requestedValue": [
 "sg-0cd61884c4ad11234"
],
 "currentValue": [
 "sg-0cd61884c4ad16341"
]
 }
],
}
```

```
{
 "parameter": "HeadNode.Ssh.KeyName",
 "requestedValue": "mykey_2",
 "currentValue": "mykey_1"
},
{
 "parameter": "Scheduling.SlurmQueues[queue1].ComputeResources[queue1-
t2micro].InstanceType",
 "requestedValue": "c4.xlarge",
 "currentValue": "t2.micro"
},
{
 "parameter": "SharedStorage[ebs1].MountDir",
 "requestedValue": "/my/very/very/long/shared_dir",
 "currentValue": "/shared"
}
]
}
```

## AWS ParallelCluster Personnalisation de l'AMI

Dans certains scénarios, il AWS ParallelCluster est nécessaire de créer une AMI personnalisée pour. Cette section décrit les éléments à prendre en compte lors de la création d'une AWS ParallelCluster AMI personnalisée.

Vous pouvez créer une AWS ParallelCluster AMI personnalisée à l'aide de l'une des méthodes suivantes :

1. [Créez un fichier de configuration d'image](#) de génération, puis utilisez la `pcluster` CLI pour créer l'image avec EC2 Image Builder. Ce processus est automatisé, reproductible et prend en charge la surveillance. Pour plus d'informations, consultez les commandes [pcluster](#)d'image.
2. Créez une instance à partir d'une AWS ParallelCluster AMI, puis connectez-vous à celle-ci et apportez des modifications manuelles. Enfin, utilisez Amazon EC2 pour créer une nouvelle AMI à partir de l'instance modifiée. Ce processus prend moins de temps. Cependant, il n'est ni automatisé ni reproductible, et il ne prend pas en charge l'utilisation des commandes de surveillance d'image de la `pcluster` CLI.

Pour plus d'informations sur ces méthodes, consultez [Création d'une AWS ParallelCluster AMI personnalisée](#).



## AWS ParallelCluster Considérations relatives à la personnalisation de l'AMI

Quelle que soit la manière dont vous créez votre image personnalisée, nous vous recommandons d'effectuer des tests de validation préliminaires et d'inclure des dispositions pour contrôler l'état de l'image en cours de création.

Pour créer une AMI personnalisée à l'aide de `pcluster`, vous devez [créer un fichier de configuration d'image](#) de génération avec une `Image` section `Build` et qu'`EC2 Image Builder` utilise pour créer votre image personnalisée. La `Build` section indique ce dont Image Builder a besoin pour créer l'image. Cela inclut `ParentImage` (image de base), et `Components`. Un [composant Image Builder](#) définit une séquence d'étapes nécessaires pour personnaliser une instance avant la création d'une image ou pour tester une instance lancée par l'image créée. Pour des exemples de AWS ParallelCluster composants, consultez la section [AMI personnalisées](#). La `Image` section définit les propriétés de l'image.

Lorsqu'il est appelé depuis `pcluster build-image` pour créer une image personnalisée, Image Builder utilise la configuration de l'image de construction avec le AWS ParallelCluster livre de recettes pour démarrer sur votre `ParentImage`. Image Builder télécharge les composants, exécute les phases de construction et de validation, crée l'AMI, lance une instance à partir de l'AMI et exécute des tests. Une fois le processus terminé, Image Builder produit une nouvelle image ou un message d'arrêt.

### Réaliser des tests de validation de composants personnalisés

Avant d'inclure un composant Image Builder dans une configuration, testez-le et validez-le à l'aide de l'une des méthodes suivantes. Le processus Image Builder pouvant prendre jusqu'à 1 heure, nous vous recommandons de tester les composants au préalable. Cela peut vous faire gagner un temps considérable.

#### Étui de script

Testez le script dans une instance en cours d'exécution, en dehors du processus de création de l'image, et vérifiez qu'il se termine avec le code de sortie 0.

#### Affaire Amazon Resource Name (ARN)

Testez le document du composant dans une instance en cours d'exécution, en dehors du processus de génération de l'image. Pour obtenir la liste des exigences, consultez la section [Gestionnaire de composants](#) dans le guide de l'utilisateur d'Image Builder.

Une fois la validation réussie, ajoutez le composant à la configuration de votre image de construction

Après avoir vérifié que le composant personnalisé fonctionne, ajoutez-le au [fichier de configuration de l'image Build](#).

## Surveillez le processus Image Builder à l'aide de **pcluster** commandes pour faciliter le débogage

### [describe-image](#)

Utilisez cette commande pour surveiller l'état de l'image de construction.

### [list-image-log-streams](#)

Utilisez cette commande pour obtenir les identifiants des flux de journaux que vous pouvez utiliser pour récupérer les événements du journal [get-image-log-events](#).

### [get-image-log-events](#)

Utilisez cette commande pour obtenir le flux journal des événements du processus de création d'image.

Par exemple, vous pouvez suivre les événements de création d'image à l'aide de la commande suivante.

```
$ watch -n 1 'pcluster get-image-log-events -i <image-id> \
 --log-stream-name/1 <pcluster-version> \
 --query "events[*].message" | tail -n 50'
```

### [get-image-stack-events](#)

Utilisez cette commande pour récupérer les événements de pile d'images pour la pile créée par Image Builder.

### [export-image-logs](#)

Utilisez cette commande pour enregistrer les journaux d'images.

Pour plus d'informations sur AWS ParallelCluster les journaux et Amazon CloudWatch, consultez [Amazon CloudWatch Logs crée des journaux d'images](#) et [Tableau de CloudWatch bord Amazon](#).

## Autres considérations

### Nouvelles AWS ParallelCluster versions et AMI personnalisées

Si vous créez et utilisez une AMI personnalisée, vous devez répéter les étapes que vous avez suivies pour créer votre AMI personnalisée à chaque nouvelle AWS ParallelCluster version.

### Actions de bootstrap personnalisées

Consultez la [Actions de bootstrap personnalisées](#) section pour déterminer si les modifications que vous souhaitez apporter peuvent être scriptées et prises en charge dans les futures AWS ParallelCluster versions.

### Utilisation d'AMI personnalisées

Vous pouvez spécifier des AMI personnalisées dans la configuration du cluster dans les [CustomAmi](#) sections [ImageScheduling/CustomAmi](#) et [SlurmQueues/Name/- Image//](#).

Pour résoudre les problèmes liés aux avertissements de validation d'AMI personnalisés, consultez [Résolution des problèmes liés aux AMI personnalisées](#).

## Lancez des instances avec des réservations de capacité à la demande (ODCR)

Avec les [réservations de capacité à la demande \(ODCR\)](#), vous pouvez réserver de la capacité pour les instances Amazon EC2 de votre cluster dans une zone de disponibilité spécifique. Ainsi, vous pouvez créer et gérer des réservations de capacité indépendamment des comptes de facturation proposés par [Savings Plans](#) ou [les instances réservées régionales](#).

Vous pouvez configurer open ou targeted ODCR. Open ODCR couvre toutes les instances qui correspondent aux attributs ODCR. Ces attributs sont le type d'instance, la plateforme et la zone de disponibilité. Vous devez définir explicitement l'ODCR ciblé dans la configuration du cluster. Pour déterminer si un ODCR est open ou targeted, exécutez la commande AWS CLI Amazon [describe-capacity-reservation](#)EC2.

Vous pouvez également créer un ODCR dans un groupe de placement de clusters appelé groupe de [placement de clusters à la demande \(CPG ODCR\)](#).

Plusieurs ODCR peuvent être regroupés dans un groupe de ressources. Cela peut être défini dans le fichier de configuration du cluster. Pour plus d'informations sur les groupes de ressources, voir [Que sont les groupes de ressources ?](#) dans le guide de l'utilisateur de Resource Groups and Tags.

## Utilisation de l'ODCR avec AWS ParallelCluster

AWS ParallelCluster prend en charge l'ODCR ouvert. Lorsque vous utilisez un ODCR ouvert, il n'est pas nécessaire de spécifier quoi que ce soit dans AWS ParallelCluster. Les instances sont automatiquement sélectionnées pour le cluster. Vous pouvez spécifier un groupe de placement existant ou en AWS ParallelCluster créer un nouveau pour vous.

### ODCR dans la configuration du cluster

À partir de AWS ParallelCluster la version 3.3.0, vous pouvez définir des ODCR dans le fichier de configuration du cluster, sans avoir à spécifier les remplacements des instances d'exécution Amazon EC2.

Vous commencez par créer des [réservations de capacité](#) et [des groupes de ressources](#) en utilisant les méthodes décrites dans la documentation associée pour chacun d'eux. Vous devez utiliser les AWS CLI méthodes pour créer des groupes de réservation de capacité. Si vous utilisez le AWS Management Console, vous ne pouvez créer que des groupes de ressources basés sur des balises ou des piles. Les groupes de ressources basés sur des balises et des piles ne sont pas pris en charge par AWS ParallelCluster ou AWS CLI lors du lancement d'instances avec des réservations de capacité.

Une fois les réservations de capacité et les groupes de ressources créés, spécifiez-les dans [SlurmQueues/CapacityReservationTarget](#) ou [SlurmQueues/ComputeResources/CapacityReservationTarget](#) comme indiqué dans l'exemple de configuration de cluster suivant. Remplacez *les valeurs* surlignées en rouge par vos valeurs valides.

```
Image:
 Os: os
HeadNode:
 InstanceType: head_node_instance
Networking:
 SubnetId: public_subnet_id
Ssh:
 KeyName: key_name
```

```
Scheduling:
 Scheduler: scheduler
 SlurmQueues:
 - Name: queue1
 Networking:
 SubnetIds:
 - private_subnet_id
 ComputeResources:
 - Name: cr1
 Instances:
 - InstanceType: instance
 MaxCount: max_queue_size
 MinCount: max_queue_size
 Efa:
 Enabled: true
 CapacityReservationTarget:
 CapacityReservationResourceGroupArn: capacity_reservation_arn
```

## OBSOLÈTE/DÉCONSEILLÉ - ODCR ciblé avec remplacements d'instances Amazon EC2

### Warning

- À partir de AWS ParallelCluster la version 3.3.0, nous ne recommandons pas cette méthode. Cette section reste une référence pour les implémentations utilisant des versions antérieures.
- Cette méthode n'est pas compatible avec l'allocation de types d'instances multiples avec Slurm.

Support pour les targeted ODCR a été ajouté dans AWS ParallelCluster la version 3.1.1. Dans cette version, un mécanisme a été introduit qui remplace les RunInstances paramètres EC2 et transmet les informations relatives à la réservation à utiliser pour chaque ressource de calcul configurée dans. AWS ParallelCluster Ce mécanisme est compatible avec l'targetedODCR. Toutefois, lorsque vous utilisez targeted ODCR, vous devez spécifier la configuration de run-instances remplacement. Les ODCR ciblés doivent être définis explicitement dans la commande Amazon AWS CLI [run-instances](#)EC2. Pour déterminer si un ODCR est open un ODCR, targeted exécutez la commande AWS CLI Amazon [describe-capacity-reservation](#)EC2.

Plusieurs ODCR peuvent être regroupés dans un groupe de ressources. Cela peut être utilisé dans le cadre du remplacement des instances d'exécution pour cibler plusieurs ODCR en même temps.

Si vous utilisez un `targeted` ODCR, vous pouvez spécifier un groupe de placement. Toutefois, vous devez également spécifier une configuration de `run-instances` remplacement.

Supposons que vous ayez AWS créé un `targeted` ODCR pour vous ou que vous disposiez d'un ensemble spécifique d'instances réservées. Dans ce cas, vous ne pouvez pas spécifier de groupe de placement. Les règles configurées par AWS peuvent entrer en conflit avec le paramètre du groupe de placement. Ainsi, si un groupe de placement est requis pour votre application, utilisez un [ODCR CPG](#). Dans les deux cas, vous devez également spécifier la configuration de `run-instances` remplacement.

Si vous utilisez un ODCR CPG, vous devez spécifier la configuration de `run-instances` remplacement et vous devez spécifier le même groupe de placement dans la configuration du cluster.

## Utilisation d'instances réservées avec AWS ParallelCluster

[Les instances réservées sont différentes des](#) réservations de capacité (ODCR). Il existe [deux types](#) d'instances réservées. Une instance réservée régionale ne réserve pas de capacité. Une instance réservée zonale réserve de la capacité dans la zone de disponibilité spécifiée.

Si vous avez des instances réservées régionales, il n'y a aucune réservation de capacité et vous pouvez obtenir des erreurs de capacité insuffisante. Si vous avez des instances réservées zonales, vous pouvez réserver des capacités, mais il n'existe aucun paramètre d'`run-instancesAPI` que vous pouvez utiliser pour les spécifier.

Les instances réservées sont prises en charge par toutes les AWS ParallelCluster versions. Il n'est pas nécessaire de spécifier quoi que ce soit AWS ParallelCluster et les instances sont automatiquement sélectionnées.

Lorsque vous utilisez des instances réservées zonales, vous pouvez éviter d'éventuelles erreurs de capacité insuffisante en omettant la spécification du groupe de placement dans la configuration du cluster.

**OBSOLÈTE/DÉCONSEILLÉ** - Utilisation de la **RunInstances** personnalisation en AWS ParallelCluster 3 pour les réservations de capacité **targeted** à la demande (ODCR)

### Warning

- À partir de AWS ParallelCluster la version 3.3.0, nous ne recommandons pas cette méthode. Cette section reste une référence pour les implémentations utilisant des versions antérieures.

- Cette méthode n'est pas compatible avec l'allocation de types d'instances multiples avec Slurm.

Vous pouvez remplacer les paramètres Amazon RunInstances EC2 pour chaque ressource de calcul configurée dans une file d'attente de cluster. Pour ce faire, créez le `/opt/slurm/etc/pcluster/run_instances_overrides.json` fichier sur le nœud principal du cluster avec le contenu de l'extrait de code suivant :

- `${queue_name}` est le nom de la file d'attente à laquelle vous souhaitez appliquer des remplacements.
- `${compute_resource_name}` est la ressource de calcul à laquelle vous souhaitez appliquer des dérogations.
- `${overrides}` est un objet JSON arbitraire qui contient une liste de RunInstances remplacements à utiliser pour la combinaison spécifique de file d'attente et de type d'instance. La syntaxe des overrides doit suivre les mêmes spécifications que celles documentées dans un appel [run\\_instances](#) boto3.

```
{
 "${queue_name}": {
 "${compute_resource_name}": {
 ${overrides}
 },
 ...
 },
 ...
}
```

Par exemple, le JSON suivant configure le groupe ODCR `group_arn` à utiliser pour les `p4d.24xlarge` instances configurées dans `my-queue` et `my-compute-resource`

```
{
 "my-queue": {
 "my-compute-resource": {
 "CapacityReservationSpecification": {
 "CapacityReservationTarget": {
 "CapacityReservationResourceGroupArn": "group_arn"
 }
 }
 }
 }
}
```

```
 }
 }
}
```

Une fois ce fichier JSON généré, les AWS ParallelCluster démons responsables de la mise à l'échelle du cluster utilisent automatiquement la configuration de remplacement pour les lancements d'instances. Pour vérifier que les paramètres spécifiés sont utilisés pour le provisionnement des instances, consultez les fichiers journaux suivants :

- `/var/log/parallelcluster/clustermgtd`(pour la capacité statique)
- `/var/log/parallelcluster/slurm_resume.log`(pour la capacité dynamique)

Si les paramètres sont corrects, vous trouverez une entrée de journal contenant les éléments suivants :

```
Found RunInstances parameters override. Launching instances with: <parameters_list>
```

**OBSOLÈTE/DÉCONSEILLÉ** - Créez un cluster avec des réservations de capacité **targeted** à la demande (ODCR)

#### Warning

- À partir de AWS ParallelCluster la version 3.3.0, nous ne recommandons pas cette méthode. Cette section reste une référence pour les implémentations utilisant des versions antérieures.
- Cette méthode n'est pas compatible avec [Allocation de plusieurs types d'instances avec Slurm](#).

1. Créez un groupe de ressources pour regrouper les capacités.

```
$ aws resource-groups create-group --name EC2CRGroup \
 --configuration '{"Type":"AWS::EC2::CapacityReservationPool"}'
 '{"Type":"AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-
resource-types", "Values": ["AWS::EC2::CapacityReservation"]}]}'
```



### Note

Un groupe de ressources ne prend pas en charge les ressources partagées par d'autres comptes.

Si l'ODCR cible est partagé par un autre compte, il n'est pas nécessaire de créer un groupe de ressources. À utiliser à la `CapacityReservationId` place d'un groupe de ressources à l'étape 3.

```
#!/bin/bash
set -e

Override run_instance attributes
cat > /opt/slurm/etc/pcluster/run_instances_overrides.json << EOF
{
 "my-queue": {
 "my-compute-resource": {
 "CapacityReservationSpecification": {
 "CapacityReservationTarget": {
 "CapacityReservationId": "cr-abcdef01234567890"
 }
 }
 }
 }
}
EOF
```

Ajoutez des réservations de capacité au groupe de ressources. Chaque fois que vous créez un nouvel ODCR, ajoutez-le à la réservation de groupe. `ACCOUNT_ID` Remplacez-le par votre identifiant de compte, `PLACEHOLDER_CAPACITY_RESERVATION` par votre numéro de réservation de capacité et `REGION_ID` par votre Région AWS identifiant (par exemple, us-east-1).

```
$ aws resource-groups group-resources --region REGION_ID --group EC2CRGroup \
 --resource-arns arn:aws:ec2:REGION_ID:ACCOUNT_ID:capacity-
 reservation/PLACEHOLDER_CAPACITY_RESERVATION
```

Créez un document de politique sur votre ordinateur local. *ACCOUNT\_ID* Remplacez-le par votre identifiant de compte et *REGION\_ID* par votre Région AWS identifiant (par exemple, us-east-1).

```
cat > policy.json << EOF
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "RunInstancesInCapacityReservation",
 "Effect": "Allow",
 "Action": "ec2:RunInstances",
 "Resource": [
 "arn:aws:ec2:REGION_ID:ACCOUNT_ID:capacity-reservation/*",
 "arn:aws:resource-groups:REGION_ID:ACCOUNT_ID:group/*"
]
 }
]
}
EOF
```

2. Créez la politique IAM qui vous convient à Compte AWS l'aide du fichier json que vous avez créé.

```
$ aws iam create-policy --policy-name RunInstancesCapacityReservation --policy-document file://policy.json
```

3. Créez le script de post-installation suivant localement sur l'instance et nommez-le **postinstall.sh**.

*ACCOUNT\_ID* Remplacez-le par votre Compte AWS identifiant et *REGION\_ID* par votre Région AWS identifiant (par exemple, us-east-1).

```
#!/bin/bash
set -e

Override run_instance attributes
cat > /opt/slurm/etc/pcluster/run_instances_overrides.json << EOF
{
 "my-queue": {
 "my-compute-resource": {
 "CapacityReservationSpecification": {
```

```

 "CapacityReservationTarget": {
 "CapacityReservationResourceGroupArn": "arn:aws:resource-
groups:REGION_ID:ACCOUNT_ID:group/EC2CRGroup"
 }
 }
}
EOF

```

Téléchargez le fichier dans un compartiment Amazon S3. Remplacez *S3\_NAME\_BUCKET* par le nom de votre compartiment S3 spécifique.

```

$ aws s3 mb s3://S3_NAME_BUCKET
aws s3 cp postinstall.sh s3://S3_NAME_BUCKET/postinstall.sh

```

4. Créez la configuration du cluster local en remplaçant les espaces réservés par vos propres valeurs.

```

Region: REGION_ID
Image:
 Os: alinux2
HeadNode:
 InstanceType: c5.2xlarge
 Ssh:
 KeyName: YOUR_SSH_KEY
 Iam:
 S3Access:
 - BucketName: S3_NAME_BUCKET
 AdditionalIamPolicies:
 - Policy: arn:aws:iam::ACCOUNT_ID:policy/RunInstancesCapacityReservation
 ## This post-install script is executed after the node is configured.
 ## It is used to install scripts at boot time and specific configurations
 ## In the script below we are overriding the calls to RunInstance to force
 ## the provisioning of our my-queue partition to go through
 ## the On-Demand Capacity Reservation
 CustomActions:
 OnNodeConfigured:
 Script: s3://S3_NAME_BUCKET/postinstall.sh
 Networking:
 SubnetId: YOUR_PUBLIC_SUBNET_IN_TARGET_AZ

```

```
Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: my-queue
 ComputeResources:
 - MinCount: 0
 MaxCount: 100
 InstanceType: p4d.24xlarge
 Name: my-compute-resource
 Efa:
 Enabled: true
 Networking:
 ## PlacementGroup:
 ## Enabled: true ## Keep PG disabled if using targeted ODCR
 SubnetIds:
 - YOUR_PRIVATE_SUBNET_IN_TARGET_AZ
```

## 5. Créez le cluster.

Utilisez la commande suivante pour créer le cluster. *cluster-config.yaml* Remplacez-le par le nom de votre fichier de configuration, *cluster-dl* par le nom de votre cluster, et *REGION\_ID* par votre ID de région (par exemple, us-east-1).

```
$ pcluster create-cluster --cluster-configuration cluster-config.yaml --cluster-name cluster-dl --region REGION_ID
```

Une fois le cluster créé, le script de post-installation s'exécute dans le nœud principal. Le script crée le `run_instances_overrides.json` fichier et remplace les appels pour `RunInstances` forcer le provisionnement de la partition à passer par la réservation de capacité à la demande.

Les AWS ParallelCluster démons responsables de la mise à l'échelle du cluster utilisent automatiquement cette configuration pour les nouvelles instances lancées. Pour vérifier que les paramètres spécifiés sont utilisés pour provisionner les instances, vous pouvez consulter les fichiers journaux suivants :

- `/var/log/parallelcluster/clustermgtd`(pour la capacité statique - [MinCount](#) > 0)
- `/var/log/parallelcluster/slurm_resume.log`(pour la capacité dynamique)

Si les paramètres sont corrects, vous trouverez une entrée de journal contenant les éléments suivants.

```
Found RunInstances parameters override. Launching instances with: <parameters_list>
```

## Mettre à jour **RunInstances** les dérogations

Vous pouvez mettre à jour la configuration JSON générée à tout moment sans arrêter le parc informatique. Une fois les modifications appliquées, toutes les nouvelles instances sont lancées avec la configuration mise à jour. Si vous devez appliquer la configuration mise à jour aux nœuds en cours d'exécution, recyclez les nœuds en forçant la fermeture d'une instance et AWS ParallelCluster en attendant de les remplacer. Vous pouvez le faire en mettant fin à l'instance depuis la console Amazon EC2 AWS CLI ou en définissant Slurm les nœuds dans DOWN l'état ou. DRAIN

Utilisez la commande suivante pour définir le Slurm nœud sur DOWN ouDRAIN.

```
$ scontrol update nodename=my-queue-dy-my-compute-resource-1 state=down
reason=your_reason
scontrol update nodename=my-queue-dy-my-compute-resource-1 state=drain
reason=your_reason
```

## Lancer des instances avec des blocs de capacité (CB)

AWS ParallelCluster prend en charge les [réservations de capacité à la demande \(ODCR\)](#) et les [blocs de capacité \(CB\) pour le Machine Learning](#). Contrairement à l'ODCR, le CB peut avoir une future heure de début et est limité dans le temps. Pour plus d'informations sur le lancement avec ODCR, voir [Lancer des instances avec des réservations de capacité à la demande \(ODCR\)](#).

## Utiliser CB avec AWS ParallelCluster

Pour configurer vos clusters nouveaux ou existants afin qu'ils utilisent un CB, vous devez d'abord avoir un CB valide sur votre AWS compte. Vous pouvez utiliser le AWS Management Console SDK ou le SDK pour trouver et acheter un CB disponible en suivant la documentation officielle. AWS Command Line Interface Une fois que vous avez un CB valide, vous pouvez définir le nom de ressource Amazon (ARN) du CB et les paramètres associés dans votre fichier AWS ParallelCluster de configuration. Pour plus d'informations, voir [Rechercher et acheter des blocs de capacité \(CB\)](#)

## CB dans la configuration du cluster

Pour utiliser un CB pour une file d'attente spécifique, vous devez utiliser le `CapacityReservationId` paramètre. Configurez-le sur un CB ID existant. Vous pouvez obtenir l'ARN du CB à partir du AWS Management Console ou du SDK que vous avez utilisé pour créer le CB. AWS CLI

Vous devez définir `CapacityType = CAPACITY_BLOCK` la file d'attente dans laquelle vous souhaitez utiliser le CB. Réglez-le sur la ressource `InstanceType` de calcul (le même type d'instance Amazon Elastic Compute Cloud que le CB).

`CapacityReservationId` Le moment spécifié au niveau de la ressource de calcul `InstanceType` est facultatif car il sera automatiquement extrait de la réservation.

Lors de l'utilisation `CapacityType = CAPACITY_BLOCK`, `MaxCount` il doit être égal ou supérieur à `MinCount` 0, car toutes les instances faisant partie de la réservation CB sont gérées comme des nœuds statiques.

Au moment de la création du cluster, le nœud principal attend que tous les nœuds statiques soient prêts avant de signaler le succès de la création du cluster. Toutefois, lors de l'utilisation `CapacityType = CAPACITY_BLOCK`, les nœuds faisant partie des ressources de calcul associées à ne seront pas pris en compte pour cette vérification. Le cluster sera créé même si tous les éléments configurés ne sont pas actifs.

L'extrait de fichier de configuration suivant indique les paramètres requis à activer dans le fichier de AWS ParallelCluster configuration.

```
SlurmQueues:
- Name: string
 CapacityType: CAPACITY_BLOCK
 InstanceType: String (EC2 Instance type of the CB)
 MinCount: integer (<= total capacity of the CB)
 MaxCount: integer (equal to MinCount)
 ComputeResources:
 - Name: string
 CapacityReservationTarget:
 CapacityReservationId: String (CB id)
```

## Comment AWS ParallelCluster utilise les blocs de capacité (CB)

AWS ParallelCluster gère les nœuds statiques associés d'une manière particulière. AWS ParallelCluster crée un cluster même si le CB n'est pas encore actif, et les instances sont lancées automatiquement une fois que le CB est actif.

Les Slurm nœuds qui correspondent aux ressources de calcul, associés à, mais qui ne sont pas encore actifs, sont maintenus en maintenance jusqu'à ce qu'ils atteignent l'heure de début du CB. Slurm les nœuds restent dans un état de réservation/maintenance et sont associés à l'utilisateur administrateur de slurm. Cela signifie qu'ils peuvent accepter des emplois, mais que ceux-ci restent en vigueur `pending` jusqu'à ce que la réservation soit supprimée.

AWS ParallelCluster met automatiquement à jour les Slurm réservations et met les nœuds CB associés en maintenance (correspondant à l'état du CB). Lorsque le CB est actif, la Slurm réservation est supprimée, les nœuds démarrent et deviennent disponibles pour les tâches en attente ou pour les nouvelles soumissions de tâches.

Lorsque l'heure de fin du CB est atteinte, les nœuds repassent à l'état de réservation/maintenance. Il appartient aux utilisateurs de soumettre/de mettre en attente les tâches dans une nouvelle file d'attente/ressource de calcul lorsque le CB n'est plus actif et que les instances sont résiliées.

## Corriger les AMI et remplacer les instances Amazon EC2

Pour garantir que tous les nœuds de calcul de cluster lancés dynamiquement se comportent de manière cohérente, AWS ParallelCluster désactive les mises à jour automatiques du système d'exploitation des instances de cluster. En outre, un ensemble spécifique d' AWS ParallelCluster AMI est créé pour chaque version de AWS ParallelCluster et la CLI associée. Cet ensemble spécifique d'AMI reste inchangé et ils ne sont pris en charge que par la AWS ParallelCluster version pour laquelle ils ont été conçus. AWS ParallelCluster Les AMI des versions publiées ne sont pas mises à jour.

Toutefois, en raison de problèmes de sécurité émergents, les clients souhaiteront peut-être ajouter des correctifs à ces AMI, puis mettre à jour leurs clusters avec l'AMI corrigée. Cela s'aligne sur le [modèle de responsabilité AWS ParallelCluster partagée](#).

Pour afficher l'ensemble spécifique d' AWS ParallelCluster AMI pris en charge par la version de la AWS ParallelCluster CLI que vous utilisez actuellement, exécutez :

```
$ pcluster version
```

```
$ pcluster list-official-images
```

Le nœud AWS ParallelCluster principal est une instance statique et vous pouvez le mettre à jour manuellement. Le redémarrage et le démarrage du nœud principal sont entièrement pris en charge à partir de AWS ParallelCluster la version 3.0.0.

Si vos instances disposent de magasins d'instances éphémères, n'oubliez pas de sauvegarder les données des magasins d'instances avant les mises à jour manuelles. Pour plus d'informations, consultez la configuration du [EphemeralVolumecluster HeadNodeLocalStorage](#)//et les [types d'instances avec volumes de stockage d'instance](#) dans le guide de l'utilisateur Amazon EC2 pour les instances Linux.

Les nœuds de calcul sont des instances éphémères. Par défaut, vous ne pouvez y accéder qu'à partir du nœud principal. À partir de AWS ParallelCluster la version 3.0.0, vous pouvez mettre à jour l'AMI associée aux instances de calcul en modifiant le [CustomAmi](#) paramètre [SchedulingSlurmQueues//Image](#)/et en exécutant la [pcluster update-cluster](#) commande, après avoir arrêté le parc de calcul avec [pcluster update-compute-fleet](#) :

```
$ pcluster update-compute-fleet-status --status STOP_REQUESTED
```

Il est possible d'automatiser la création d'une AMI personnalisée mise à jour pour les nœuds de calcul en utilisant l'une des méthodes suivantes :

- Utilisez la [pcluster build-image](#) commande avec un [Build](#)/mis à jour [ParentImage](#).
- Exécutez le build avec [Build/UpdateOsPackages/Enabled>true](#).

## Mise à jour ou remplacement de l'instance du nœud principal

Dans certains cas, il peut vous être demandé de redémarrer ou de redémarrer le nœud principal. Par exemple, cela est nécessaire lorsque vous mettez à jour manuellement le système d'exploitation ou lorsqu'une mise hors service [planifiée d'une AWS instance](#) impose le redémarrage de l'instance du nœud principal.

Si votre instance ne possède pas de lecteurs éphémères, vous pouvez l'arrêter et la redémarrer à tout moment. Dans le cas d'une mise hors service planifiée, le démarrage de l'instance arrêtée permet de la faire migrer vers le nouveau matériel.



De même, vous pouvez arrêter et démarrer manuellement une instance qui ne possède pas de magasins d'instances. Pour ce cas et pour les autres cas d'instances sans volumes éphémères, passez à [Arrêter et démarrer le nœud principal d'un cluster](#)

Si votre instance possède des disques éphémères et qu'elle a été arrêtée, les données du magasin d'instances sont perdues. Vous pouvez déterminer si le type d'instance utilisé pour le nœud principal comporte des magasins d'instance à partir du tableau figurant dans la section [Volumes de stockage d'instance](#).

## Enregistrez des données à partir de disques éphémères

À partir de AWS ParallelCluster la version 3.0.0, le redémarrage et le redémarrage du nœud principal sont entièrement pris en charge pour chaque type d'instance. Toutefois, si les instances possèdent un lecteur éphémère, ses données sont perdues. Suivez les étapes suivantes pour préserver vos données avant le redémarrage ou le redémarrage d'un nœud principal.

Pour vérifier si certaines données doivent être conservées, consultez le contenu du [MountDir](#)dossier [EphemeralVolume](#)(/scratchpar défaut).

Vous pouvez transférer les données vers le volume racine ou vers les systèmes de stockage partagés connectés au cluster, tels qu'Amazon FSx, Amazon EFS ou Amazon EBS. Notez que le transfert de données vers un stockage à distance peut entraîner des coûts supplémentaires.

Après avoir enregistré les données, passez à [Arrêter et démarrer le nœud principal d'un cluster](#).

## Arrêter et démarrer le nœud principal d'un cluster

1. Vérifiez qu'aucune tâche n'est en cours d'exécution dans le cluster.

Lorsque vous utilisez un Slurm planificateur :

- Si l'`sbatch --no-requeueoption` n'est pas spécifiée, les tâches en cours d'exécution sont requises.
- Si l'`--no-requeueoption` est spécifiée, les tâches en cours d'exécution échouent.

2. Demandez l'arrêt d'un parc de calcul en cluster :

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status STOP_REQUESTED
{
 "status": "STOP_REQUESTED",
 ...
}
```

```
}
```

3. Attendez que l'état du parc informatique soit le suivant STOPPED :

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status STOP_REQUESTED
{
 "status": "STOPPED",
 ...
}
```

4. Pour les mises à jour manuelles avec redémarrage du système d'exploitation ou redémarrage d'une instance, vous pouvez utiliser le AWS Management Console ou AWS CLI. Voici un exemple d'utilisation du AWS CLI.

```
Retrieve head node instance id
$ pcluster describe-cluster --cluster-name cluster-name --status STOP_REQUESTED
{
 "headNode": {
 "instanceId": "i-1234567890abcdef0",
 ...
 },
 ...
}
stop and start the instance
$ aws ec2 stop-instances --instance-ids 1234567890abcdef0
{
 "StoppingInstances": [
 {
 "CurrentState": {
 "Name": "stopping"
 ...
 },
 "InstanceId": "i-1234567890abcdef0",
 "PreviousState": {
 "Name": "running"
 ...
 }
 }
]
}
$ aws ec2 start-instances --instance-ids 1234567890abcdef0
{
 "StartingInstances": [
```

```
{
 "CurrentState": {
 "Name": "pending"
 ...
 },
 "InstanceId": "i-1234567890abcdef0",
 "PreviousState": {
 "Name": "stopped"
 ...
 }
}
]
```

5. Démarrez le parc de calcul du cluster :

```
$ pcluster update-compute-fleet --cluster-name cluster-name --status
START_REQUESTED
{
 "status": "START_REQUESTED",
 ...
}
```

## Systèmes d'exploitation

AWS ParallelCluster prend en charge Amazon Linux 2, Amazon Linux 2023, CentOS 7, Ubuntu 22.04, Ubuntu 2004, Red Hat Enterprise Linux 8 (RHEL8), Rocky 8, Red Hat Enterprise Linux 9 (RHEL9) et Rocky 9. AWS ParallelCluster propose des AMI prédéfinies pour certains systèmes d'exploitation. Pour plus de détails sur les AMI fournies, AWS ParallelCluster reportez-vous à [Image Section](#).

### Considérations relatives au système d'exploitation

#### Ubuntu 22.04

Ubuntu 2204 nécessite des clés plus sécurisées pour SSH et ne prend pas en charge les clés RSA par défaut. Générez une clé ed25519 et utilisez-la pour créer un cluster.

Ubuntu 2204 ne peut pas être mis à jour vers le dernier noyau car il n'existe pas de client Fsx pour ce noyau.

## RHEL 8

RedHat Enterprise Linux 8.7 (rhel8) est ajouté à partir de la AWS ParallelCluster version 3.6.0. Si vous configurez votre cluster pour utiliser rhel8, le coût à la demande pour tout type d'instance est plus élevé que lorsque vous configurez votre cluster pour utiliser d'autres systèmes d'exploitation pris en charge.

Pour plus d'informations sur la tarification, consultez les [sections Tarification à la demande](#) et [Quels sont les tarifs et offres de Red Hat Enterprise Linux sur Amazon Elastic Compute Cloud ?](#) .

## Rocky 8

AWS ParallelCluster La version 3.8.0 prend en charge Rocky Linux 8, mais les AMI Rocky Linux 8 prédéfinies (pour les architectures x86 et ARM) ne sont pas disponibles. AWS ParallelCluster La version 3.8.0 prend en charge la création de clusters avec Rocky Linux 8 en utilisant des AMI personnalisées à l'aide de la [CustomAmi](#) propriété. Pour plus d'informations sur la création d'AMI personnalisées, reportez-vous à [AWS ParallelCluster Personnalisation de l'AMI](#).

Pour créer votre AMI personnalisée à partir d'une AMI Rocky Linux 8 de base, vous pouvez envisager de vous abonner aux AMI [Rocky Linux 8](#) disponibles sur le AWS [Marketplace](#). N'oubliez pas de consulter les prix et les coûts d'abonnement des AMI Rocky Linux 8 sur le AWS Marketplace. Vous pouvez également utiliser les AMI [officielles de Rocky Linux 8 comme AMI](#) de base.

## Centos 7

[Gdrcopy](#) a supprimé Centos7 de sa matrice de support du système d'exploitation. Cela signifie que gdrcopy 2.3.1 est la dernière version compatible avec ce système d'exploitation. Vous devez épingler les versions NVIDIA et gdrcopy pour Centos7, car les dernières versions du pilote open source NVIDIA (OpenRM, c'est-à-dire 535.129.03+) sont incompatibles avec cette version de gdrcopy. À partir de la ParallelCluster version 3.8.0, nos AMI Centos7 officielles seront publiées avec gdrcopy 2.3.1 et le pilote NVIDIA 535.129.03.

## Rocky 9

AWS ParallelCluster La version 3.9.0 prend en charge Rocky Linux 9, mais les AMI Rocky Linux 9 prédéfinies (pour les architectures x86 et ARM) ne sont pas disponibles. AWS ParallelCluster La version 3.9.0 prend en charge la création de clusters avec Rocky Linux 9 à l'aide d'AMI personnalisées à l'aide de la [CustomAmi](#) propriété. Pour plus d'informations sur la création d'AMI personnalisées, reportez-vous à la section [Personnalisation des AWS ParallelCluster AMI](#). Pour créer votre AMI personnalisée à partir d'une AMI Rocky Linux 9 de base, vous pouvez également utiliser

les AMI [Rocky Linux 9 officielles comme AMI](#) de base. La création personnalisée de l'AMI Rocky Linux 9 peut échouer si l'AMI de base ne possède pas le dernier noyau. Pour mettre à jour le noyau avant de créer l'AMI :

- [Lancez une instance à l'aide d'un identifiant d'AMI rocky9 à partir d'ici : https://rockylinux.org/cloud-images/](https://rockylinux.org/cloud-images/)
- connectez-vous à l'instance par ssh et exécutez la commande suivante : `sudo yum -y update`
- Créez une image à partir de l'instance à utiliser en tant que ParentImage

# Référence pour AWS ParallelCluster

## Rubriques

- [Commandes de l'interface de ligne de commande AWS ParallelCluster](#)
- [Fichiers de configuration](#)
- [AWS ParallelCluster Référence d'API](#)
- [AWS ParallelClusterAPI de bibliothèque Python](#)

## Commandes de l'interface de ligne de commande AWS ParallelCluster

`pcluster` est la commande AWS ParallelCluster CLI principale. Vous l'utilisez `pcluster` pour lancer et gérer des clusters HPC dans le AWS Cloud et pour créer et gérer des images AMI personnalisées.

`pcluster3-config-converter` est utilisé pour convertir les configurations de cluster du format de AWS ParallelCluster la version 2 au format de AWS ParallelCluster la version 3.

```
pcluster [-h] (build-image | configure |
 create-cluster | dcv-connect |
 delete-cluster | delete-cluster-instances | delete-image |
 describe-cluster | describe-cluster-instances |
 describe-compute-fleet | describe-image |
 export-cluster-logs | export-image-logs |
 get-cluster-log-events | get-cluster-stack-events |
 get-image-log-events | get-image-stack-events |
 list-cluster-log-streams | list-clusters |
 list-images | list-image-log-streams | list-official-images |
 ssh | update-cluster |
 update-compute-fleet | version) ...
pcluster3-config-converter [-h] [-t CLUSTER_TEMPLATE]
 [-c CONFIG_FILE]
 [--force-convert]
 [-o OUTPUT_FILE]
```

## Rubriques

- [pcluster](#)
- [pcluster3-config-converter](#)

# pcluster

`pcluster` est la principale commande de la AWS ParallelCluster CLI. Vous pouvez `pcluster` lancer et gérer des clusters HPC dans le AWS Cloud.

`pcluster` enregistre les journaux de vos commandes dans `pcluster.log.# des fichiers/home/user/.parallelcluster/`. Pour plus d'informations, consultez [pcluster Journaux de la CLI](#).

Pour l'utiliser `pcluster`, vous devez disposer d'un rôle IAM doté des [autorisations](#) requises pour l'exécuter.

```
pcluster [-h]
```

## Arguments

### `pcluster` *command*

Choix possibles : [build-image](#)[configure](#)[create-cluster](#)[dcv-connect](#)[delete-cluster](#)[delete-cluster-instances](#)[delete-image](#)[describe-cluster](#)[describe-cluster-instances](#)[describe-compute-fleet](#)[describe-image](#)[export-cluster-log](#)[export-image-log](#)[get-cluster-log-events](#)[get-cluster-stack-events](#)[get-image-log-events](#)[get-image-stack-events](#)[list-clusters](#)[list-cluster-log-streams](#)[list-images](#)[list-image-log-streams](#)[list-official-images](#)[ssh](#)[update-cluster](#)[update-compute-fleet](#)[version](#)

## Sous-commandes :

### Rubriques

- [pcluster build-image](#)
- [pcluster configure](#)
- [pcluster create-cluster](#)
- [pcluster dcv-connect](#)
- [pcluster delete-cluster](#)
- [pcluster delete-cluster-instances](#)
- [pcluster delete-image](#)
- [pcluster describe-cluster](#)

- [pcluster describe-cluster-instances](#)
- [pcluster describe-compute-fleet](#)
- [pcluster describe-image](#)
- [pcluster export-cluster-logs](#)
- [pcluster export-image-logs](#)
- [pcluster get-cluster-log-events](#)
- [pcluster get-cluster-stack-events](#)
- [pcluster get-image-log-events](#)
- [pcluster get-image-stack-events](#)
- [pcluster list-clusters](#)
- [pcluster list-cluster-log-streams](#)
- [pcluster list-images](#)
- [pcluster list-image-log-streams](#)
- [pcluster list-official-images](#)
- [pcluster ssh](#)
- [pcluster update-cluster](#)
- [pcluster update-compute-fleet](#)
- [pcluster version](#)

## pcluster build-image

Créez une AWS ParallelCluster image personnalisée dans la région spécifiée.

```
pcluster build-image [-h]
 --image-configuration IMAGE_CONFIGURATION
 --image-id IMAGE_ID
 [--debug]
 [--dryrun DRYRUN]
 [--query QUERY]
 [--region REGION]
 [--rollback-on-failure ROLLBACK_ON_FAILURE]
 [--suppress-validators SUPPRESS_VALIDATORS [SUPPRESS_VALIDATORS ...]]
 [--validation-failure-level {INFO,WARNING,ERROR}]
```



## Arguments nommés

### **-h, --help**

Affiche le texte d'aide pour `pcluster build-image`.

### **--image-configuration, -c *IMAGE\_CONFIGURATION***

Spécifie le fichier de configuration de l'image sous forme de document YAML.

### **--image-id, -i *IMAGE\_ID***

Spécifie l'identifiant de l'image qui sera créée.

### **--debug**

Activation de la journalisation du débogage

### **--dryrun *DRYRUN***

Lorsque `true`, la commande effectue la validation sans créer de ressources. Vous pouvez l'utiliser pour valider la configuration de l'image. (La valeur par défaut est `false`.)

### **--query *QUERY***

Requête JMESpath à exécuter en sortie.

### **--region, -r *REGION***

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide du paramètre [Region](#) dans le fichier de configuration de l'image, de la variable d'AWS\_DEFAULT\_REGION environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

### **--rollback-on-failure *ROLLBACK\_ON\_FAILURE***

Quand `true`, lance automatiquement une restauration de la pile d'images en cas d'échec. (La valeur par défaut est `false`.)

### **--suppress-validators *SUPPRESS\_VALIDATORS* [*SUPPRESS\_VALIDATORS ...*]**

Identifie un ou plusieurs validateurs de configuration à supprimer.

Format : (ALL|type:[A-Za-z0-9]+)

### **--validation-failure-level {INFO,WARNING,ERROR}**

Spécifie le niveau de validation minimal qui entraînera l'échec de la création. (La valeur par défaut est `ERROR`.)

Exemple utilisant AWS ParallelCluster la version 3.1.2 :

```
$ pcluster build-image --image-configuration image-config.yaml --image-id custom-
alinux2-image
{
 "image": {
 "imageId": "custom-alinux2-image",
 "imageBuildStatus": "BUILD_IN_PROGRESS",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
custom-alinux2-image/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "region": "us-east-1",
 "version": "3.1.2"
 }
}
```

#### Warning

`pcluster build-image` utilise le VPC par défaut. Si le VPC par défaut a été supprimé, par exemple en utilisant AWS Control Tower ou AWS Landing Zone, l'ID du sous-réseau doit être spécifié dans le fichier de configuration de l'image. Pour plus d'informations, consultez [SubnetId](#).

## pcluster configure

Lance un assistant de configuration interactif pour AWS ParallelCluster la version 3. Pour plus d'informations, consultez [Configuration et création d'un cluster à l'aide de l'interface de ligne de AWS ParallelCluster commande](#).

```
pcluster configure [-h]
 --config CONFIG
 [--debug]
 [--region REGION]
```

Arguments nommés

### **-h, --help**

Affiche le texte d'aide pour `pcluster configure`.

**--config** *CONFIG*

Chemin de sortie du fichier de configuration généré.

**--debug**

Activation de la journalisation du débogage

**--region, -r** *REGION*

Spécifie le Région AWS à utiliser. La région doit être spécifiée à l'aide du paramètre [Région](#) du fichier de configuration de l'image, de la variable d'AWS\_DEFAULT\_REGION environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

**pcluster create-cluster**

Crée un AWS ParallelCluster cluster.

```
pcluster create-cluster [-h]
 --cluster-configuration CLUSTER_CONFIGURATION
 --cluster-name CLUSTER_NAME
 [--debug]
 [--dryrun DRYRUN]
 [--query QUERY]
 [--region REGION]
 [--rollback-on-failure ROLLBACK_ON_FAILURE]
 [--suppress-validators SUPPRESS_VALIDATORS [SUPPRESS_VALIDATORS ...]]
 [--validation-failure-level {INFO,WARNING,ERROR}]
```

Arguments nommés

**-h, --help**

Affiche le texte d'aide pour `pcluster create-cluster`.

**--cluster-configuration, -c** *CLUSTER\_CONFIGURATION*

Spécifie le fichier de configuration du cluster YAML.

**--cluster-name, -n** *CLUSTER\_NAME*

Spécifie le nom du cluster à créer.

Le nom doit commencer par un caractère alphabétique. Le nom peut comporter jusqu'à 60 caractères. Si la Slurm comptabilité est activée, le nom peut comporter jusqu'à 40 caractères.

Caractères valides : a-z, A-Z, 0-9 et - (tiret).

### **--debug**

Active la journalisation du débogage.

### **--dryrun *DRYRUN***

Lorsque `true`, la commande effectue la validation sans créer de ressources. Vous pouvez l'utiliser pour valider la configuration du cluster. (La valeur par défaut est `false`.)

### **--query *QUERY***

Spécifie la requête JMESpath à exécuter sur la sortie.

### **--region, -r *REGION***

Spécifie le Région AWS à utiliser. Le Région AWS doit être spécifié à l'aide du [Region](#) paramètre du fichier de configuration du cluster, de la variable d'AWS\_DEFAULT\_REGION environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

### **--rollback-on-failure *ROLLBACK\_ON\_FAILURE***

Quand `true`, lance automatiquement une restauration de la pile du cluster en cas de défaillance. (La valeur par défaut est `true`.)

### **--suppress-validators *SUPPRESS\_VALIDATORS* [*SUPPRESS\_VALIDATORS* ...]**

Identifie un ou plusieurs validateurs de configuration à supprimer.

Format : (ALL|type :[A-Za-z0-9]+)

### **--validation-failure-level {INFO,WARNING,ERROR}**

Spécifie le niveau de validation minimal qui entraînera l'échec de la création. (La valeur par défaut est `ERROR`.)

Exemple utilisant AWS ParallelCluster la version 3.1.4 :

```
$ pcluster create-cluster -c cluster-config.yaml -n cluster-v3
{
```

```
"cluster": {
 "clusterName": "cluster-v3",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "region": "us-east-1",
 "version": "3.1.4",
 "clusterStatus": "CREATE_IN_PROGRESS"
}
```

## pcluster dcv-connect

Permet de se connecter au nœud principal via une session interactive à l'aide de NICE DCV.

```
pcluster dcv-connect [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--key-path KEY_PATH]
 [--region REGION]
 [--show-url]
```

### Arguments nommés

#### **-h, --help**

Affiche le texte d'aide pour `pcluster dcv-connect`.

#### **--cluster-name, -n *CLUSTER\_NAME***

Spécifie le nom du cluster.

#### **--debug**

Active la journalisation du débogage.

#### **--key-path *KEY\_PATH***

Spécifie le chemin de la clé SSH à utiliser pour la connexion.

#### **--region, -r *REGION***

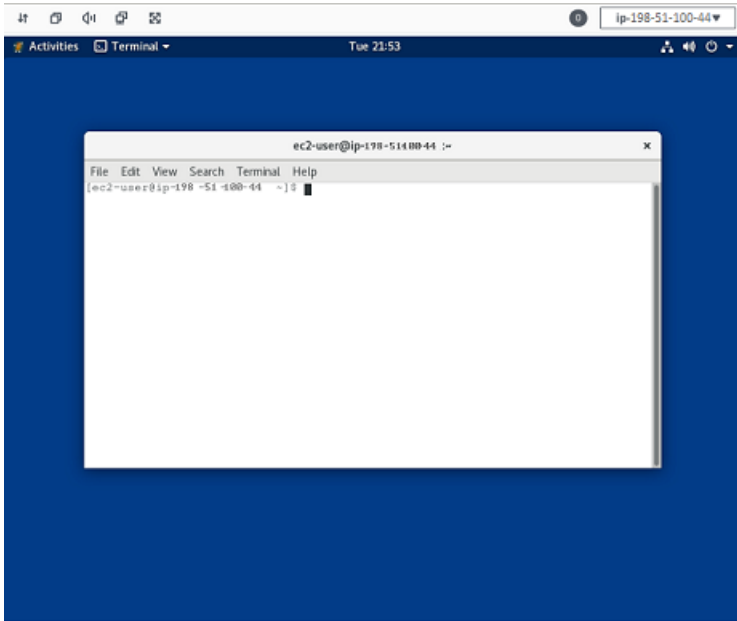
Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'`AWS_DEFAULT_REGION` environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

## --show-url

Imprime l'URL qui sera utilisée pour la connexion DCV et s'en sort.

Exemple utilisant AWS ParallelCluster la version 3.1.4 :

```
$ pcluster dcv-connect -n cluster-3Dcv -r us-east-1 --key-path /home/user/.ssh/key.pem
```



## pcluster delete-cluster

Lance la suppression d'un cluster.

```
pcluster delete-cluster [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--query QUERY]
 [--region REGION]
```

Arguments nommés

### -h, --help

Affiche le texte d'aide pour `pcluster delete-cluster`.

**--cluster-name, -n *CLUSTER\_NAME***

Spécifie le nom du cluster.

**--debug**

Active la journalisation du débogage.

**--query *QUERY***

Spécifie la requête JMESpath à exécuter sur la sortie.

**--region, -r *REGION***

Spécifie le Région AWS à utiliser. La région doit être spécifiée à l'aide de la variable d'AWS\_DEFAULT\_REGION environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

Exemple utilisant AWS ParallelCluster la version 3.1.4 :

```
$ pcluster delete-cluster -n cluster-v3
{
 "cluster": {
 "clusterName": "cluster-v3",
 "cloudformationStackStatus": "DELETE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "region": "us-east-1",
 "version": "3.1.4",
 "clusterStatus": "DELETE_IN_PROGRESS"
 }
}
```

## **pcluster delete-cluster-instances**

Initiez l'arrêt forcé de tous les nœuds de calcul du cluster. Cela ne fonctionne pas avec les AWS Batch clusters.

```
pcluster delete-cluster-instances [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--force FORCE]
 [--query QUERY]
```

```
[--region REGION]
```

## Arguments nommés

### **-h, --help**

Affiche le texte d'aide pour `pcluster delete-cluster-instances`.

### **--cluster-name, -n *CLUSTER\_NAME***

Spécifie le nom du cluster.

### **--debug**

Active la journalisation du débogage.

### **--force *FORCE***

Quand `true`, force la suppression en ignorant les erreurs de validation. (La valeur par défaut est `false`.)

### **--query *QUERY***

Spécifie la requête JMESpath à exécuter sur la sortie.

### **--region, -r *REGION***

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'`AWS_DEFAULT_REGION` environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

```
$ pcluster delete-cluster-instances -n cluster-v3
```

## **pcluster delete-image**

Lance la suppression de l' AWS ParallelCluster image personnalisée.

```
pcluster delete-image [-h]
 --image-id IMAGE_ID
 [--debug]
 [--force FORCE]
 [--query QUERY]
 [--region REGION]
```



## Arguments nommés

### **-h, --help**

Affiche le texte d'aide pour `pcluster delete-image`.

### **--image-id, -i *IMAGE\_ID***

Spécifie l'identifiant de l'image qui sera supprimée.

### **--debug**

Active la journalisation du débogage.

### **--force *FORCE***

Lorsque `true`, force la suppression si certaines instances utilisent l'AMI ou si l'AMI est partagée. (La valeur par défaut est `false`.)

### **--query *QUERY***

Spécifie la requête JMESpath à exécuter sur la sortie.

### **--region, -r *REGION***

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'`AWS_DEFAULT_REGION` environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

Exemple utilisant AWS ParallelCluster la version 3.1.4 :

```
$ pcluster delete-image --image-id custom-alinux2-image
{
 "image": {
 "imageId": "custom-alinux2-image",
 "imageBuildStatus": "DELETE_IN_PROGRESS",
 "region": "us-east-1",
 "version": "3.1.4"
 }
}
```

## **pcluster describe-cluster**

Obtenez des informations détaillées sur un cluster.

```
pcluster describe-cluster [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--query QUERY]
 [--region REGION]
```

## Arguments nommés

### **-h, --help**

Affiche le texte d'aide pour `pcluster describe-cluster`.

### **--cluster-name, -n *CLUSTER\_NAME***

Spécifie le nom du cluster.

### **--debug**

Active la journalisation du débogage.

### **--query *QUERY***

Spécifie la requête JMESpath à exécuter sur la sortie.

### **--region, -r *REGION***

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'`AWS_DEFAULT_REGION` environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

Exemples utilisant AWS ParallelCluster la version 3.1.4 :

Décrivez les détails du cluster :

```
$ pcluster describe-cluster -n cluster-v3
{
 "creationTime": "2022-07-12T17:19:16.101Z",
 "headNode": {
 "launchTime": "2022-07-12T17:22:21.000Z",
 "instanceId": "i-1234567890abcdef0",
 "publicIpAddress": "198.51.100.44",
 "instanceType": "t2.micro",
 "state": "running",
 "privateIpAddress": "192.0.2.0.196"
```

```

},
"loginNodes": {
 "status": "active",
 "address": "8af2145440569xyz.us-east-1.amazonaws.com",
 "scheme": "internet-facing|internal",
 "healthyNodes": 3,
 "unhealthyNodes": 0
},
"version": "3.1.4",
"clusterConfiguration": {
 "url": "https://parallelcluster-e5ca74255d6c3886-v1-do-not-delete..."
},
"tags": [
 {
 "value": "3.1.4",
 "key": "parallelcluster:version"
 }
],
"cloudFormationStackStatus": "CREATE_COMPLETE",
"clusterName": "cluster-v3",
"computeFleetStatus": "RUNNING",
"cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
"lastUpdatedTime": "2022-07-12T17:19:16.101Z",
"region": "us-east-1",
"clusterStatus": "CREATE_COMPLETE"
}

```

`describe-cluster` À utiliser pour récupérer la configuration du cluster :

```

$ curl -o - - $(pcluster describe-cluster -n cluster-v3 --query clusterConfiguration.url
| xargs echo)
Region: us-east-1
Image:
 Os: alinux2
HeadNode:
 InstanceType: t2.micro
 Networking:
 SubnetId: subnet-abcdef01234567890
 Ssh:
 KeyName: adpc
 Iam:
 S3Access:

```

```
- BucketName: cluster-v3-bucket
 KeyName: logs
 EnableWriteAccess: true
Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: queue1
 ComputeResources:
 - Name: t2micro
 InstanceType: t2.micro
 MinCount: 0
 MaxCount: 10
 Networking:
 SubnetIds:
 - subnet-021345abcdef6789
```

## pcluster describe-cluster-instances

Décrivez les instances d'un cluster.

```
pcluster describe-cluster-instances [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--next-token NEXT_TOKEN]
 [--node-type {HeadNode,ComputeNode,LoginNode}]
 [--query QUERY]
 [--queue-name QUEUE_NAME]
 [--region REGION]
```

Arguments nommés

### **-h, --help**

Affiche le texte d'aide pour `pcluster describe-cluster-instances`.

### **--cluster-name, -n *CLUSTER\_NAME***

Spécifie le nom du cluster.

### **--debug**

Active la journalisation du débogage.

**--next-token** *NEXT\_TOKEN*

Spécifie le jeton à utiliser pour les demandes paginées.

**--node-type** {HeadNode,ComputeNode,LoginNode}

Spécifie les types de nœuds à répertorier. Les valeurs prises en charge sont HeadNode, ComputeNode et LoginNode. Si ce paramètre n'est pas spécifié, les instances HeadNode, ComputeNode et LoginNode sont décrites.

**--query** *QUERY*

Spécifie la requête JMESpath à exécuter sur la sortie.

**--queue-name** *QUEUE\_NAME*

Spécifie le nom de la file d'attente à répertorier. Si ce paramètre n'est pas spécifié, les instances de toutes les files d'attente sont décrites.

**--region, -r** *REGION*

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'AWS\_DEFAULT\_REGION environnement, du region paramètre de la [default] section du ~/.aws/config fichier ou du --region paramètre.

Exemple utilisant AWS ParallelCluster la version 3.1.4 :

```
$ pcluster describe-cluster-instances -n cluster-v3
{
 "instances": [
 {
 "launchTime": "2022-07-12T17:22:21.000Z",
 "instanceId": "i-1234567890abcdef0",
 "publicIpAddress": "198.51.100.44",
 "instanceType": "t2.micro",
 "state": "running",
 "nodeType": "HeadNode",
 "privateIpAddress": "192.0.2.0.196"
 },
 {
 "launchTime": "2022-07-12T17:37:42.000Z",
 "instanceId": "i-021345abcdef6789",
 "queueName": "queue1",
 "publicIpAddress": "198.51.100.44",
 "instanceType": "t2.micro",
```

```
 "state": "pending",
 "nodeType": "ComputeNode",
 "privateIpAddress": "192.0.2.0.196"
 }
]
}
```

## pcluster describe-compute-fleet

Décrivez l'état du parc informatique.

```
pcluster describe-compute-fleet [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--query QUERY]
 [--region REGION]
```

Arguments nommés

### -h, --help

Affiche le texte d'aide pour `pcluster describe-compute-fleet`.

### --cluster-name, -n *CLUSTER\_NAME*

Spécifie le nom du cluster.

### --debug

Active la journalisation du débogage.

### --query *QUERY*

Spécifie la requête JMESpath à exécuter sur la sortie.

### --region, -r *REGION*

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'AWS\_DEFAULT\_REGION environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

Exemple utilisant AWS ParallelCluster la version 3.1.4 :

```
$ pcluster describe-compute-fleet -n pcluster-v3
{
```

```
"status": "RUNNING",
"lastStatusUpdatedTime": "2022-07-12T17:24:26.000Z"
}
```

## pcluster describe-image

Obtenez des informations détaillées sur une image.

```
pcluster describe-image [-h]
 --image-id IMAGE_ID
 [--debug]
 [--query QUERY]
 [--region REGION]
```

Arguments nommés

### **-h, --help**

Affiche le texte d'aide pour `pcluster describe-image`.

### **--image-id, -i *IMAGE\_ID***

Spécifie l'ID de l'image.

### **--debug**

Active la journalisation du débogage.

### **--query *QUERY***

Spécifie la requête JMESpath à exécuter sur la sortie.

### **--region, -r *REGION***

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'AWS\_DEFAULT\_REGION environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

Exemple utilisant AWS ParallelCluster la version 3.1.2 :

```
$ pcluster describe-image --image-id custom-linux2-image
{
 "imageConfiguration": {
 "url": "https://parallelcluster-1234abcd5678-v1-do-not-delete.../configs/image-
config.yaml"
```

```

},
"imageId": "custom-alinux2-image",
"creationTime": "2022-04-05T20:23:07.000Z"
"imageBuildStatus": "BUILD_COMPLETE",
"region": "us-east-1",
"ec2AmiInfo": {
 "amiName": "custom-alinux2-image 2022-04-05T19-55-22.518Z",
 "amiId": "ami-1234abcd5678efgh",
 "description": "AWS ParallelCluster AMI for alinux2,
kernel-4.14.268-205.500.amzn2.x86_64, lustre-2.10.8-5.amzn2.x86_64,
efa-1.14.2-1.amzn2.x86_64, dcv-2021.3.11591-1.el7.x86_64, slurm-21-08-6-1",
 "state": "AVAILABLE",
"tags": [
 {
 "value": "arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-custom-alinux2-image/3.1.2/1",
 "key": "Ec2ImageBuilderArn"
 },
 {
 "value": "parallelcluster-1234abcd5678efgh-v1-do-not-delete",
 "key": "parallelcluster:s3_bucket"
 },
 {
 "value": "custom-alinux2-image",
 "key": "parallelcluster:image_name"
 },
 {
 "value": "available",
 "key": "parallelcluster:build_status"
 },
 {
 "value": "s3://parallelcluster-1234abcd5678efgh-v1-do-not-delete/
parallelcluster/3.1.2/images/custom-alinux2-image-1234abcd5678efgh/configs/image-
config.yaml",
 "key": "parallelcluster:build_config"
 },
 {
 "value": "Amazon EC2 Image Builder",
 "key": "CreatedBy"
 },
 {
 "value": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/imagebuilder/
ParallelClusterImage-custom-alinux2-image",
 "key": "parallelcluster:build_log"
 }
]
}

```



```
 },
 {
 "value": "4.14.268-205.500.amzn2.x86_64",
 "key": "parallelcluster:kernel_version"
 },
 {
 "value": "arn:aws:imagebuilder:us-east-1:444455556666:image/amazon-linux-2-
x86/2022.3.16/1",
 "key": "parallelcluster:parent_image"
 },
 {
 "value": "3.1.2",
 "key": "parallelcluster:version"
 },
 {
 "value": "0.5.14",
 "key": "parallelcluster:munge_version"
 },
 {
 "value": "21-08-6-1",
 "key": "parallelcluster:slurm_version"
 },
 {
 "value": "2021.3.11591-1.el7.x86_64",
 "key": "parallelcluster:dcv_version"
 },
 {
 "value": "alinux2-image",
 "key": "parallelcluster:image_id"
 },
 {
 "value": "3.2.3",
 "key": "parallelcluster:pmix_version"
 },
 {
 "value": "parallelcluster/3.7.0/images/alinux2-image-abcd1234efgh56781234",
 "key": "parallelcluster:s3_image_dir"
 },
 {
 "value": "1.14.2-1.amzn2.x86_64",
 "key": "parallelcluster:efa_version"
 },
 {
 "value": "alinux2",
```

```

 "key": "parallelcluster:os"
 },
 {
 "value": "aws-parallelcluster-cookbook-3.1.2",
 "key": "parallelcluster:bootstrap_file"
 },
 {
 "value": "1.8.23-10.amzn2.1.x86_64",
 "key": "parallelcluster:sudo_version"
 },
 {
 "value": "2.10.8-5.amzn2.x86_64",
 "key": "parallelcluster:lustre_version"
 }
],
"architecture": "x86_64"
},
"version": "3.1.2"
}

```

## pcluster export-cluster-logs

Exportez les journaux du cluster vers une tar.gz archive locale en les faisant passer par un compartiment Amazon S3.

```

pcluster export-cluster-logs [-h]
 --bucket BUCKET_NAME
 --cluster-name CLUSTER_NAME
 [--bucket-prefix BUCKET_PREFIX]
 [--debug]
 [--end-time END_TIME]
 [--filters FILTER [FILTER ...]]
 [--keep-s3-objects KEEP_S3_OBJECTS]
 [--output-file OUTPUT_FILE]
 [--region REGION]
 [--start-time START_TIME]

```

Arguments nommés

**-h, --help**

Affiche le texte d'aide pour `pcluster export-cluster-logs`.

**--bucket** *BUCKET\_NAME*

Spécifie le nom du compartiment Amazon S3 vers lequel exporter les données des journaux du cluster. Il doit se trouver dans la même région que le cluster.

**Note**

Vous devez ajouter des autorisations à la politique du compartiment Amazon S3 pour accorder CloudWatch l'accès. Pour plus d'informations, consultez la section [Définir des autorisations sur un compartiment Amazon S3](#) dans le Guide de l'utilisateur CloudWatch des journaux.

**--cluster-name, -n** *CLUSTER\_NAME*

Spécifie le nom du cluster.

**--bucket-prefix** *BUCKET\_PREFIX*

Spécifie le chemin dans le compartiment Amazon S3 où les données des journaux exportés doivent être stockées.

Par défaut, le préfixe du bucket est :

```
cluster-name-logs-202209061743.tar.gz
```

*202209061743* est l'heure actuelle au %Y%m%d%H%M format.

**--debug**

Active la journalisation du débogage.

**--end-time** *END\_TIME*

Spécifie la fin de la plage de temps pour collecter les événements du journal, exprimée au format ISO 8601 (YYYY-MM-DDThh:mm:ssZ par exemple 2021-01-01T20:00:00Z ). Les événements dont l'horodatage est égal ou supérieur à cette heure ne sont pas inclus. Les éléments temporels (par exemple les minutes et les secondes) peuvent être omis. La valeur par défaut est l'heure actuelle.

**--filters** *FILTER [FILTER ...]*

Spécifie les filtres pour le journal. Format :Name=a,Values=1 Name=b,Values=2,3. Les filtres pris en charge sont les suivants :

`private-dns-name`

Spécifie la forme abrégée du nom DNS privé de l'instance (par exemple `ip-10-0-0-101`).

`node-type`

Spécifie le type de nœud, la seule valeur acceptée pour ce filtre est `HeadNode`.

**`--keep-s3-objects` *KEEP\_S3\_OBJECTS***

Dans `true` ce cas, les objets exportés exportés vers Amazon S3 sont conservés. (La valeur par défaut est `false`.)

**`--output-file` *OUTPUT\_FILE***

Spécifie le chemin du fichier dans lequel enregistrer l'archive du journal. Si cela est fourni, les journaux sont enregistrés localement. Dans le cas contraire, ils sont chargés sur Amazon S3 avec l'URL renvoyée dans la sortie. Par défaut, le téléchargement vers Amazon S3 est effectué.

**`--region, -r` *REGION***

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'`AWS_DEFAULT_REGION` environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

**`--start-time` *START\_TIME***

Spécifie le début de la plage horaire, exprimé au format ISO 8601 (`YYYY-MM-DDThh:mm:ssZ` par exemple `2021-01-01T20:00:00Z`). Les événements du journal dont l'horodatage est égal à cette heure ou à une date ultérieure sont inclus. Si ce n'est pas spécifié, la valeur par défaut est l'heure à laquelle le cluster a été créé.

Exemple utilisant AWS ParallelCluster la version 3.1.4 :

```
$ pcluster export-cluster-logs --bucket cluster-v3-bucket -n cluster-v3
{
 "url": "https://cluster-v3-bucket..."
}
```

## **pcluster export-image-logs**

Exportez les journaux de la pile du générateur d'images vers une `tar.gz` archive locale en les faisant passer par un compartiment Amazon S3.

```
pcluster export-image-logs [-h]
 --bucket BUCKET
 --image-id IMAGE_ID
 [--bucket-prefix BUCKET_PREFIX]
 [--debug]
 [--end-time END_TIME]
 [--keep-s3-objects KEEP_S3_OBJECTS]
 [--output-file OUTPUT_FILE]
 [--region REGION]
 [--start-time START_TIME]
```

## Arguments nommés

### **-h, --help**

Affiche le texte d'aide pour `pcluster export-image-logs`.

### **--bucket *BUCKET\_NAME***

Spécifie le nom du compartiment Amazon S3 vers lequel exporter les journaux de création d'images. Il doit se trouver dans la même région que l'image.

#### Note

Vous devez ajouter des autorisations à la politique du compartiment Amazon S3 pour accorder CloudWatch l'accès. Pour plus d'informations, consultez la section [Définir des autorisations sur un compartiment Amazon S3](#) dans le Guide de l'utilisateur CloudWatch des journaux.

### **--image-id, -i *IMAGE\_ID***

ID de l'image dont les journaux seront exportés.

### **--bucket-prefix *BUCKET\_PREFIX***

Spécifie le chemin dans le compartiment Amazon S3 où les données des journaux exportés doivent être stockées.

Par défaut, le préfixe du bucket est :

```
ami-id-logs-202209061743.tar.gz
```

**202209061743** est l'heure actuelle au `%Y%m%d%H%M` format.

### **--debug**

Active la journalisation du débogage.

### **--end-time** *END\_TIME*

Spécifie la fin de la plage de temps pour collecter les événements du journal, exprimée au format ISO 8601 (YYYY-MM-DDThh:mm:ssZ par exemple `2021-01-01T20:00:00Z` '). Les événements dont l'horodatage est égal ou supérieur à cette heure ne sont pas inclus. Les éléments temporels (par exemple les minutes et les secondes) peuvent être omis. La valeur par défaut est l'heure actuelle.

### **--keep-s3-objects** *KEEP\_S3\_OBJECTS*

Dans `true` ce cas, les objets exportés exportés vers Amazon S3 sont conservés. (La valeur par défaut est `false`.)

### **--output-file** *OUTPUT\_FILE*

Spécifie le chemin du fichier dans lequel enregistrer l'archive du journal. Si cela est fourni, les journaux sont enregistrés localement. Dans le cas contraire, ils sont chargés sur Amazon S3 avec l'URL renvoyée dans la sortie. Par défaut, le téléchargement vers Amazon S3 est effectué.

### **--region, -r** *REGION*

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'`AWS_DEFAULT_REGION` environnement, du `region` paramètre de la `[default]` section du `~/aws/config` fichier ou du `--region` paramètre.

### **--start-time** *START\_TIME*

Spécifie le début de la plage horaire, exprimé au format ISO 8601 (YYYY-MM-DDThh:mm:ssZ par exemple `2021-01-01T20:00:00Z`). Les événements du journal dont l'horodatage est égal à cette heure ou à une date ultérieure sont inclus. Si ce n'est pas spécifié, la valeur par défaut est l'heure à laquelle le cluster a été créé.

Exemple utilisant AWS ParallelCluster la version 3.1.4 :

```
$ pcluster export-image-logs --bucket image-v3-bucket --image-id ami-1234abcd5678efgh
{
 "url": "https://image-v3-bucket..."
}
```

## pcluster get-cluster-log-events

Récupérez les événements associés à un flux de log.

```
pcluster get-cluster-log-events [-h]
 --cluster-name CLUSTER_NAME
 --log-stream-name LOG_STREAM_NAME
 [--debug]
 [--end-time END_TIME]
 [--limit LIMIT]
 [--next-token NEXT_TOKEN]
 [--query QUERY]
 [--region REGION]
 [--start-from-head START_FROM_HEAD]
 [--start-time START_TIME]
```

Arguments nommés

### **-h, --help**

Affiche le texte d'aide pour `pcluster get-cluster-log-events`.

### **--cluster-name, -n *CLUSTER\_NAME***

Spécifie le nom du cluster.

### **--log-stream-name *LOG\_STREAM\_NAME***

Spécifie le nom du flux de journal. Vous pouvez utiliser la `list-cluster-log-streams` commande pour récupérer un flux de journal associé à un ou plusieurs événements.

### **--debug**

Active la journalisation du débogage.

### **--end-time *END\_TIME***

Spécifie la fin de la plage de temps, exprimée au format ISO 8601 (YYYY-MM-DDThh:mm:ssZ par exemple 2021-01-01T20:00:00Z). Les événements dont l'horodatage est égal ou supérieur à cette heure ne sont pas inclus.

### **--limit *LIMIT***

Spécifie le nombre maximum d'événements de journal renvoyés. Si aucune valeur n'est spécifiée, le maximum est le nombre d'événements de journal que peut contenir une taille de réponse de 1 Mo, jusqu'à 10 000 événements de journal.

**--next-token** *NEXT\_TOKEN*

Spécifie le jeton à utiliser pour les demandes paginées.

**--query** *QUERY*

Spécifie la requête JMESpath à exécuter sur la sortie.

**--region, -r** *REGION*

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'AWS\_DEFAULT\_REGION environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

**--start-from-head** *START\_FROM\_HEAD*

Si la valeur est égale à cette valeur `true`, les premiers événements du journal sont renvoyés en premier. Si la valeur est égale à `false`, les événements du journal les plus récents sont renvoyés en premier. (La valeur par défaut est `false`.)

**--start-time** *START\_TIME*

Spécifie le début de la plage horaire, exprimé au format ISO 8601 (YYYY-MM-DDThh:mm:ssZ par exemple `2021-01-01T20:00:00Z`). Les événements dont l'horodatage est égal à cette heure ou postérieur à cette heure sont inclus.

Exemple utilisant AWS ParallelCluster la version 3.1.4 :

```
$ pcluster get-cluster-log-events \
 -c cluster-v3 \
 -r us-east-1 \
 --log-stream-name ip-198-51-100-44.i-1234567890abcdef0.clustermgtd \
 --limit 3
{
 "nextToken": "f/36966906399261933213029082268132291405859205452101451780/s",
 "prevToken": "b/36966906399239632467830551644990755687586557090595471362/s",
 "events": [
 {
 "message": "2022-07-12 19:16:53,379 - [slurm_plugin.clustermgtd:_maintain_nodes]
- INFO - Performing node maintenance actions",
 "timestamp": "2022-07-12T19:16:53.379Z"
 },
 {
 "message": "2022-07-12 19:16:53,380 - [slurm_plugin.clustermgtd:_maintain_nodes]
- INFO - Following nodes are currently in replacement: (x0) []",
```



```
 "timestamp": "2022-07-12T19:16:53.380Z"
 },
 {
 "message": "2022-07-12 19:16:53,380 -
[slurm_plugin.clustermgtd:_terminate_orphaned_instances] - INFO - Checking for
orphaned instance",
 "timestamp": "2022-07-12T19:16:53.380Z"
 }
]
}
```

## pcluster get-cluster-stack-events

Récupérez les événements associés à la pile pour le cluster spécifié.

### Note

À partir de la version 3.6.0, AWS ParallelCluster utilise des piles imbriquées pour créer les ressources associées aux files d'attente et aux ressources de calcul. L'GetClusterStackEventsAPI et la `pcluster get-cluster-stack-events` commande renvoient uniquement les événements de la pile principale du cluster. Vous pouvez consulter les événements de la pile de clusters, y compris ceux liés aux files d'attente et aux ressources de calcul, dans la CloudFormation console.

```
pcluster get-cluster-stack-events [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--next-token NEXT_TOKEN]
 [--query QUERY]
 [--region REGION]
```

### Arguments nommés

#### **-h, --help**

Affiche le texte d'aide pour `pcluster get-cluster-stack-events`.

#### **--cluster-name, -n *CLUSTER\_NAME***

Spécifie le nom du cluster.

**--debug**

Active la journalisation du débogage.

**--next-token** *NEXT\_TOKEN*

Spécifie le jeton à utiliser pour les demandes paginées.

**--query** *QUERY*

Spécifie la requête JMESpath à exécuter sur la sortie.

**--region, -r** *REGION*

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'AWS\_DEFAULT\_REGION environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

Exemple utilisant AWS ParallelCluster la version 3.1.4 :

```
$ pcluster get-cluster-stack-events \
 -n cluster-v3 \
 -r us-east-1 \
 --query "events[0]"
{
 "eventId": "1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "physicalResourceId": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-
v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "resourceStatus": "CREATE_COMPLETE",
 "stackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-
v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "stackName": "cluster-v3",
 "logicalResourceId": "cluster-v3",
 "resourceType": "AWS::CloudFormation::Stack",
 "timestamp": "2022-07-12T18:29:12.140Z"
}
```

**pcluster get-image-log-events**

Récupérez les événements associés à la création d'une image.

```
pcluster get-image-log-events [-h]
```

```
--image-id IMAGE_ID
--log-stream-name LOG_STREAM_NAME
[--debug]
[--end-time END_TIME]
[--limit LIMIT]
[--next-token NEXT_TOKEN]
[--query QUERY]
[--region REGION]
[--start-from-head START_FROM_HEAD]
[--start-time START_TIME]
```

## Arguments nommés

### **-h, --help**

Affiche le texte d'aide pour `pcluster get-image-log-events`.

### **--image-id, -i *IMAGE\_ID***

Spécifie l'ID de l'image.

### **--log-stream-name *LOG\_STREAM\_NAME***

Spécifie le nom du flux de journal. Vous pouvez utiliser la `list-image-log-streams` commande pour récupérer un flux de journal associé à un ou plusieurs événements.

### **--debug**

Active la journalisation du débogage.

### **--end-time *END\_TIME***

Spécifie la fin de la plage de temps, exprimée au format ISO 8601 (YYYY-MM-DDThh:mm:ssZ par exemple `2021-01-01T20:00:00Z`). Les événements dont l'horodatage est égal ou supérieur à cette heure ne sont pas inclus.

### **--limit *LIMIT***

Spécifie le nombre maximum d'événements de journal renvoyés. Si aucune valeur n'est spécifiée, le maximum est le nombre d'événements de journal que peut contenir une taille de réponse de 1 Mo, jusqu'à 10 000 événements de journal.

### **--next-token *NEXT\_TOKEN***

Spécifie le jeton à utiliser pour les demandes paginées.

**--query *QUERY***

Spécifie la requête JMESpath à exécuter sur la sortie.

**--region, -r *REGION***

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'AWS\_DEFAULT\_REGION environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

**--start-from-head *START\_FROM\_HEAD***

Si la valeur est égale à cette valeur `true`, les premiers événements du journal sont renvoyés en premier. Si la valeur est égale à `false`, les événements du journal les plus récents sont renvoyés en premier. (La valeur par défaut est `false`.)

**--start-time *START\_TIME***

Spécifie le début de la plage horaire, exprimé au format ISO 8601 (YYYY-MM-DDThh:mm:ssZ par exemple `2021-01-01T20:00:00Z`). Les événements dont l'horodatage est égal ou supérieur à cette heure sont inclus.

Exemple utilisant AWS ParallelCluster la version 3.1.2 :

```
$ pcluster get-image-log-events --image-id custom-linux2-image --region us-east-1 --
log-stream-name 3.1.2/1 --limit 3
{
 "nextToken": "f/36778317771100849897800729464621464113270312017760944178/s",
 "prevToken": "b/36778317766952911290874033560295820514557716777648586800/s",
 "events": [
 {
 "message": "ExecuteBash: FINISHED EXECUTION",
 "timestamp": "2022-04-05T22:13:26.633Z"
 },
 {
 "message": "Document arn:aws:imagebuilder:us-east-1:123456789012:component/
parallelclusterimage-test-1234abcd-56ef-78gh-90ij-abcd1234efgh/3.1.2/1",
 "timestamp": "2022-04-05T22:13:26.741Z"
 },
 {
 "message": "TOE has completed execution successfully",
 "timestamp": "2022-04-05T22:13:26.819Z"
 }
]
}
```

```
]
}
```

## **pcluster get-image-stack-events**

Récupérez les événements associés à la pile pour la génération d'image spécifiée.

```
pcluster get-image-stack-events [-h]
 --image-id IMAGE_ID
 [--debug]
 [--next-token NEXT_TOKEN]
 [--query QUERY]
 [--region REGION]
```

Arguments nommés

### **-h, --help**

Affiche le texte d'aide pour `pcluster get-image-stack-events`.

### **--image-id, -i *IMAGE\_ID***

Spécifie l'ID de l'image.

### **--debug**

Active la journalisation du débogage.

### **--next-token *NEXT\_TOKEN***

Spécifie le jeton à utiliser pour les demandes paginées.

### **--query *QUERY***

Spécifie la requête JMESpath à exécuter sur la sortie.

### **--region, -r *REGION***

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'AWS\_DEFAULT\_REGION environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

Exemple utilisant AWS ParallelCluster la version 3.1.2 :

```
$ pcluster get-image-stack-events --image-id custom-alinux2-image --region us-east-1 --
query "events[0]"
{
 "eventId": "ParallelClusterImage-CREATE_IN_PROGRESS-2022-04-05T21:39:24.725Z",
 "physicalResourceId": "arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-custom-alinux2-image/3.1.2/1",
 "resourceStatus": "CREATE_IN_PROGRESS",
 "resourceStatusReason": "Resource creation Initiated",
 "resourceProperties": "{\"InfrastructureConfigurationArn\":
\\\"arn:aws:imagebuilder:us-east-1:123456789012:infrastructure-configuration/
parallelclusterimage-1234abcd-56ef-78gh-90ij-abcd1234efgh\\\",\\\"ImageRecipeArn
\\\":\\\"arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/
parallelclusterimage-custom-alinux2-image/3.1.2\\\",\\\"DistributionConfigurationArn
\\\":\\\"arn:aws:imagebuilder:us-east-1:123456789012:distribution-
configuration/parallelclusterimage-1234abcd-56ef-78gh-90ij-abcd1234efgh\\\",
\\\"EnhancedImageMetadataEnabled\\\":\\\"false\\\",\\\"Tags\\\":{\\\"parallelcluster:image_name\\\":
\\\"custom-alinux2-image\\\",\\\"parallelcluster:image_id\\\":\\\"custom-alinux2-image\\\"}}\",
 "stackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/custom-alinux2-
image/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "stackName": "custom-alinux2-image",
 "logicalResourceId": "ParallelClusterImage",
 "resourceType": "AWS::ImageBuilder::Image",
 "timestamp": "2022-04-05T21:39:24.725Z"
}
```

## pcluster list-clusters

Récupérez la liste des clusters existants.

```
pcluster list-clusters [-h]
 [--cluster-status {CREATE_IN_PROGRESS,CREATE_FAILED,CREATE_COMPLETE,
DELETE_IN_PROGRESS,DELETE_FAILED,UPDATE_IN_PROGRESS,
UPDATE_COMPLETE,UPDATE_FAILED}]
 [{CREATE_IN_PROGRESS,CREATE_FAILED,CREATE_COMPLETE,
DELETE_IN_PROGRESS,DELETE_FAILED,UPDATE_IN_PROGRESS,
UPDATE_COMPLETE,UPDATE_FAILED} ...]]
 [--debug]
 [--next-token NEXT_TOKEN]
 [--query QUERY]
 [--region REGION]
```

## Arguments nommés

### **-h, --help**

Affiche le texte d'aide pour `pcluster list-clusters`.

**--cluster-status** {CREATE\_IN\_PROGRESS, CREATE\_FAILED, CREATE\_COMPLETE, DELETE\_IN\_PROGRESS, DELETE\_FAILED, UPDATE\_IN\_PROGRESS, UPDATE\_COMPLETE, UPDATE\_FAILED} [{CREATE\_IN\_PROGRESS, CREATE\_FAILED, CREATE\_COMPLETE, DELETE\_IN\_PROGRESS, DELETE\_FAILED, UPDATE\_IN\_PROGRESS, UPDATE\_COMPLETE, UPDATE\_FAILED} ...]

Spécifie la liste des états de cluster à filtrer. (La valeur par défaut est all.)

### **--debug**

Active la journalisation du débogage.

**--next-token** *NEXT\_TOKEN*

Spécifie le jeton à utiliser pour les demandes paginées.

**--query** *QUERY*

Spécifie la requête JMESpath à exécuter sur la sortie.

**--region, -r** *REGION*

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'AWS\_DEFAULT\_REGION environnement, du `region` paramètre de la [default] section du `~/.aws/config` fichier ou du `--region` paramètre.

Exemple utilisant AWS ParallelCluster la version 3.1.4 :

```
$ pcluster list-clusters
{
 "clusters": [
 {
 "clusterName": "cluster-v3",
 "cloudformationStackStatus": "CREATE_COMPLETE",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "region": "us-east-1",
 "version": "3.1.4",
```

```

 "clusterStatus": "CREATE_COMPLETE"
 }
]
}

```

## pcluster list-cluster-log-streams

Récupérez la liste des flux de journaux associés à un cluster.

```

pcluster list-cluster-log-streams [-h]
 --cluster-name CLUSTER_NAME
 [--filters FILTERS [FILTERS ...]]
 [--next-token NEXT_TOKEN] [--debug]
 [--query QUERY]
 [--region REGION]

```

Arguments nommés

### -h, --help

Affiche le texte d'aide pour `pcluster list-cluster-log-streams`.

### --cluster-name, -n *CLUSTER\_NAME*

Spécifie le nom du cluster.

### --debug

Active la journalisation du débogage.

### --filters *FILTERS* [*FILTERS* ...]

Spécifie les filtres pour les flux de journaux. Format :Name=a,Values=1 Name=b,Values=2,3.

Les filtres pris en charge sont les suivants :

private-dns-name

Spécifie la forme abrégée du nom DNS privé de l'instance (par exemple `i-10-0-0-101`).

node-type

Spécifie le type de nœud, la seule valeur acceptée pour ce filtre est `HeadNode`.

### --next-token *NEXT\_TOKEN*

Spécifie le jeton à utiliser pour les demandes paginées.



**--query *QUERY***

Spécifie la requête JMESpath à exécuter sur la sortie.

**--region, -r *REGION***

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'AWS\_DEFAULT\_REGION environnement, du region paramètre de la [default] section du ~/.aws/config fichier ou du --region paramètre.

Exemple utilisant AWS ParallelCluster la version 3.1.4 :

```
$ pcluster list-cluster-log-streams \
 -n cluster-v3 \
 -r us-east-1 \
 --query 'logStreams[*].LogStreamName'
[
 "ip-172-31-58-205.i-1234567890abcdef0.cfn-init",
 "ip-172-31-58-205.i-1234567890abcdef0.chef-client",
 "ip-172-31-58-205.i-1234567890abcdef0.cloud-init",
 "ip-172-31-58-205.i-1234567890abcdef0.clustermgtd",
 "ip-172-31-58-205.i-1234567890abcdef0.slurmctld",
 "ip-172-31-58-205.i-1234567890abcdef0.supervisord",
 "ip-172-31-58-205.i-1234567890abcdef0.system-messages"
]
```

**pcluster list-images**

Récupérez la liste des images personnalisées existantes.

```
pcluster list-images [-h]
 --image-status {AVAILABLE,PENDING,FAILED}
 [--debug]
 [--next-token NEXT_TOKEN]
 [--query QUERY]
 [--region REGION]
```

Arguments nommés

**-h, --help**

Affiche le texte d'aide pour `pcluster list-images`.

**--image-status {AVAILABLE,PENDING,FAILED}**

Filtrez les images renvoyées en fonction de l'état indiqué.

**--debug**

Active la journalisation du débogage.

**--next-token *NEXT\_TOKEN***

Spécifie le jeton à utiliser pour les demandes paginées.

**--query *QUERY***

Spécifie la requête JMESpath à exécuter sur la sortie.

**--region, -r *REGION***

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'AWS\_DEFAULT\_REGION environnement, du region paramètre de la [default] section du ~/.aws/config fichier ou du --region paramètre.

Exemple utilisant AWS ParallelCluster la version 3.1.2 :

```
$ pcluster list-images --image-status AVAILABLE
{
 "images": [
 {
 "imageId": "custom-alinux2-image",
 "imageBuildStatus": "BUILD_COMPLETE",
 "ec2AmiInfo": {
 "amiId": "ami-1234abcd5678efgh"
 },
 "region": "us-east-1",
 "version": "3.1.2"
 }
]
}
```

**pcluster list-image-log-streams**

Récupérez la liste des flux de journaux associés à une image.

```
pcluster list-image-log-streams [-h]
```

```
--image-id IMAGE_ID
[--next-token NEXT_TOKEN] [--debug]
[--query QUERY]
[--region REGION]
```

## Arguments nommés

### **-h, --help**

Affiche le texte d'aide pour `pcluster list-image-log-streams`.

### **--image-id, -i *IMAGE\_ID***

Spécifie l'ID de l'image.

### **--debug**

Active la journalisation du débogage.

### **--next-token *NEXT\_TOKEN***

Spécifie le jeton à utiliser pour les demandes paginées.

### **--query *QUERY***

Spécifie la requête JMESpath à exécuter sur la sortie.

### **--region, -r *REGION***

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'AWS\_DEFAULT\_REGION environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

Exemple utilisant AWS ParallelCluster la version 3.1.2 :

```
$ pcluster list-image-log-streams --image-id custom-alinux2-image --region us-east-1 --
query 'logStreams[*].logStreamName'
[
 "3.0.0/1",
 "3.1.2/1"
]
```

## **pcluster list-official-images**

Décrivez les AWS ParallelCluster AMI officielles.

```
pcluster list-official-images [-h]
 [--architecture ARCHITECTURE]
 [--debug]
 [--os OS]
 [--query QUERY]
 [--region REGION]
```

## Arguments nommés

### **-h, --help**

Affiche le texte d'aide pour `pcluster list-official-images`.

### **--architecture *ARCHITECTURE***

Spécifie l'architecture à utiliser pour filtrer les résultats. Si ce paramètre n'est pas spécifié, toutes les architectures sont renvoyées.

### **--debug**

Active la journalisation du débogage.

### **--os *OS***

Spécifie le système d'exploitation à utiliser pour filtrer les résultats. Si ce paramètre n'est pas spécifié, tous les systèmes d'exploitation sont renvoyés.

### **--query *QUERY***

Spécifie la requête JMESpath à exécuter sur la sortie.

### **--region, -r *REGION***

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide du paramètre [Region](#) dans le fichier de configuration de l'image, de la variable d'AWS\_DEFAULT\_REGION environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

Exemple utilisant AWS ParallelCluster la version 3.1.2 :

```
$ pcluster list-official-images
{
 "images": [
 {
 "amiId": "ami-015cfef4e0d6306b2",
```

```
 "os": "ubuntu2004",
 "name": "aws-parallelcluster-3.1.2-ubuntu-2004-lts-hvm-x86_64-202202261505
2022-02-26T15-08-34.759Z",
 "version": "3.1.2",
 "architecture": "x86_64"
 },
 {
 "amiId": "ami-036f23237ce49d25b",
 "os": "ubuntu2204",
 "name": "aws-parallelcluster-3.1.2-ubuntu-1804-lts-hvm-x86_64-202202261505
2022-02-26T15-08-17.558Z",
 "version": "3.1.2",
 "architecture": "x86_64"
 },
 {
 "amiId": "ami-09e5327e694d89ef4",
 "os": "ubuntu2004",
 "name": "aws-parallelcluster-3.1.2-ubuntu-2004-lts-hvm-arm64-202202261505
2022-02-26T15-08-45.736Z",
 "version": "3.1.2",
 "architecture": "arm64"
 },
 {
 "amiId": "ami-0b9b0874c35f626ae",
 "os": "alinux2",
 "name": "aws-parallelcluster-3.1.2-amzn2-hvm-x86_64-202202261505
2022-02-26T15-08-31.311Z",
 "version": "3.1.2",
 "architecture": "x86_64"
 },
 {
 "amiId": "ami-0bf6d01f398f3737e",
 "os": "centos7",
 "name": "aws-parallelcluster-3.1.2-centos7-hvm-x86_64-202202261505
2022-02-26T15-08-25.001Z",
 "version": "3.1.2",
 "architecture": "x86_64"
 },
 {
 "amiId": "ami-0d0de4f95f56374bc",
 "os": "alinux2",
 "name": "aws-parallelcluster-3.1.2-amzn2-hvm-arm64-202202261505
2022-02-26T15-08-46.088Z",
 "version": "3.1.2",
```

```
 "architecture": "arm64"
 },
 {
 "amiId": "ami-0ebf7bc54b8740dc6",
 "os": "ubuntu2204",
 "name": "aws-parallelcluster-3.1.2-ubuntu-1804-lts-hvm-arm64-202202261505
2022-02-26T15-08-45.293Z",
 "version": "3.1.2",
 "architecture": "arm64"
 }
]
```

## pcluster ssh

Exécute une ssh commande avec le nom d'utilisateur et l'adresse IP du cluster préremplis. Des arguments arbitraires sont ajoutés à la fin de la ligne de ssh commande.

```
pcluster ssh [-h]
 --cluster-name CLUSTER_NAME
 [--debug]
 [--dryrun DRYRUN]
 [--region REGION]
```

### Arguments nommés

#### **-h, --help**

Affiche le texte d'aide pour `pcluster ssh`.

#### **--cluster-name, -n *CLUSTER\_NAME***

Spécifie le nom du cluster auquel se connecter.

#### **--debug**

Active la journalisation du débogage.

#### **--dryrun *DRYRUN***

Quand `true`, imprime la ligne de commande qui doit être exécutée et s'arrête. (La valeur par défaut est `false`.)

**--region, -r *REGION***

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'AWS\_DEFAULT\_REGION environnement, du region paramètre de la [default] section du ~/.aws/config fichier ou du --region paramètre.

Exemple :

```
$ pcluster ssh --cluster-name mycluster -i ~/.ssh/id_rsa
```

Exécute une ssh commande avec le nom d'utilisateur et l'adresse IP du cluster préremplis :

```
ssh ec2-user@1.1.1.1 -i ~/.ssh/id_rsa
```

**pcluster update-cluster**

Met à jour un cluster existant pour qu'il corresponde aux paramètres d'un fichier de configuration spécifié.

```
pcluster update-cluster [-h]
 --cluster-configuration CLUSTER_CONFIGURATION
 --cluster-name CLUSTER_NAME
 [--debug]
 [--dryrun DRYRUN]
 [--force-update FORCE_UPDATE]
 [--query QUERY]
 [--region REGION]
 [--suppress-validators SUPPRESS_VALIDATORS [SUPPRESS_VALIDATORS ...]]
 [--validation-failure-level {INFO,WARNING,ERROR}]
```

Arguments nommés

**-h, --help**

Affiche le texte d'aide pour `pcluster update-cluster`.

**--cluster-configuration, -c *CLUSTER\_CONFIGURATION***

Spécifie le fichier de configuration du cluster YAML.

**--cluster-name, -n *CLUSTER\_NAME***

Spécifie le nom du cluster.

**--debug**

Active la journalisation du débogage.

**--dryrun *DRYRUN***

Quand `true`, effectue la validation sans mettre à jour le cluster ni créer de ressources. Il peut être utilisé pour valider la configuration de l'image et les exigences de mise à jour. (La valeur par défaut est `false`.)

**--force-update *FORCE\_UPDATE***

Quand `true`, force la mise à jour en ignorant les erreurs de validation de mise à jour. (La valeur par défaut est `false`.)

**--query *QUERY***

Spécifie la requête JMESpath à exécuter sur la sortie.

**--region, -r *REGION***

Spécifie le Région AWS à utiliser. Le Région AWS doit être spécifié à l'aide du [Region](#) paramètre du fichier de configuration du cluster, de la variable d'AWS\_DEFAULT\_REGION environnement, du `region` paramètre de la `[default]` section du `~/.aws/config` fichier ou du `--region` paramètre.

**--suppress-validators *SUPPRESS\_VALIDATORS* [*SUPPRESS\_VALIDATORS ...*]**

Identifie un ou plusieurs validateurs de configuration à supprimer.

Format : `(ALL|type:[A-Za-z0-9]+)`

**--validation-failure-level *{INFO,WARNING,ERROR}***

Spécifie le niveau d'échecs de validation signalés pour la mise à jour.

Exemple utilisant AWS ParallelCluster la version 3.1.4 :

```
$ pcluster update-cluster -c cluster-config.yaml -n cluster-v3 -r us-east-1
{
 "cluster": {
 "clusterName": "cluster-v3",
 "cloudformationStackStatus": "UPDATE_IN_PROGRESS",
```



```

 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster-v3/1234abcd-56ef-78gh-90ij-abcd1234efgh",
 "region": "us-east-1",
 "version": "3.1.4",
 "clusterStatus": "UPDATE_IN_PROGRESS"
 },
 "changeSet": [
 {
 "parameter": "HeadNode.Iam.S3Access",
 "requestedValue": {
 "BucketName": "pc-beta-test",
 "KeyName": "output",
 "EnableWriteAccess": false
 }
 },
 {
 "parameter": "HeadNode.Iam.S3Access",
 "currentValue": {
 "BucketName": "pcluster-east-test-bucket",
 "KeyName": "logs",
 "EnableWriteAccess": true
 }
 }
]
}

```

## pcluster update-compute-fleet

Met à jour l'état du parc de calcul du cluster.

```

pcluster update-compute-fleet [-h]
 --cluster-name CLUSTER_NAME
 --status {START_REQUESTED,STOP_REQUESTED,ENABLED,DISABLED}

 [--debug]
 [--query QUERY]
 [--region REGION]

```

Arguments nommés

### -h, --help

Affiche le texte d'aide pour `pcluster update-compute-fleet`.

**--cluster-name, -n *CLUSTER\_NAME***

Spécifie le nom du cluster.

**--status {START\_REQUESTED,STOP\_REQUESTED,ENABLED,DISABLED}**

Spécifie le statut appliqué au parc de calcul du cluster. Les statuts START\_REQUESTED et C STOP\_REQUESTED correspondent au Slurm planificateur tandis que les statuts ENABLED DISABLED correspondent au planificateur. AWS Batch

**--debug**

Active la journalisation du débogage.

**--query *QUERY***

Spécifie la requête JMESpath à exécuter sur la sortie.

**--region, -r *REGION***

Spécifie le Région AWS à utiliser. Région AWS Il doit être spécifié à l'aide de la variable d'AWS\_DEFAULT\_REGIONenvironnement, du region paramètre de la [default] section du ~/.aws/config fichier ou du --region paramètre.

Exemple utilisant AWS ParallelCluster la version 3.1.4 :

```
$ pcluster update-compute-fleet -n cluster-v3 --status STOP_REQUESTED
{
 "status": "STOP_REQUESTED",
 "lastStatusUpdateTime": "2022-07-12T20:19:47.653Z"
}
```

## pcluster version

Affiche la version de AWS ParallelCluster.

```
pcluster version [-h] [--debug]
```

Arguments nommés

**-h, --help**

Affiche le texte d'aide pour `pcluster version`.

## --debug

Active la journalisation du débogage.

Exemple utilisant AWS ParallelCluster la version 3.1.4 :

```
$ pcluster version
{
 "version": "3.1.4"
}
```

## pcluster3-config-convert

Lit un fichier de configuration AWS ParallelCluster version 2 et écrit un fichier de configuration AWS ParallelCluster version 3.

```
pcluster3-config-convert [-h]
 [-t CLUSTER_TEMPLATE]
 [-c CONFIG_FILE]
 [--force-convert]
 [-o OUTPUT_FILE]
```

### Arguments nommés

#### -h, --help

Affiche le texte d'aide pour `pcluster3-config-convert`.

#### -t **CLUSTER\_TEMPLATE**, --cluster-template **CLUSTER\_TEMPLATE**

Spécifie la [\[cluster\]section](#) du fichier de configuration à convertir. S'il n'est pas spécifié, le script recherchera le paramètre [cluster-template](#) dans la [\[global\]section](#) ou recherchera. `[cluster default]`

#### -c **CONFIG\_FILE**, --config-file **CONFIG\_FILE**


Spécifie le fichier de configuration de la AWS ParallelCluster version 2 à lire.

#### --force-convert

Permet une conversion même si un ou plusieurs paramètres ne sont pas pris en charge et ne sont pas recommandés.

**-o *OUTPUT\_FILE*, --output-file *OUTPUT\_FILE***

Spécifie le fichier de configuration de la AWS ParallelCluster version 3 à écrire. Si ce paramètre n'est pas spécifié, la configuration est écrite dans stdout.

 Note

La `pcluster3-config-convert` commande a été ajoutée dans la AWS ParallelCluster version 3.0.1.

## Fichiers de configuration

AWS ParallelCluster utilise des fichiers YAML 1.1 pour les paramètres de configuration.

### Rubriques

- [Fichier de configuration du cluster](#)
- [Créez des fichiers de configuration d'image](#)

## Fichier de configuration du cluster

AWS ParallelCluster la version 3 utilise des fichiers de configuration distincts pour contrôler la définition de l'infrastructure du cluster et la définition des AMI personnalisées. Tous les fichiers de configuration utilisent des fichiers YAML 1.1. Des informations détaillées pour chacun de ces fichiers de configuration sont disponibles ci-dessous. Pour des exemples de configurations, consultez [https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/example\\_configs](https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/example_configs).

Ces objets sont utilisés pour la configuration du cluster AWS ParallelCluster version 3.

### Rubriques

- [Propriétés du fichier de configuration du cluster](#)
- [Imds Section](#)
- [Image Section](#)
- [HeadNode Section](#)
- [Scheduling Section](#)

- [SharedStorage Section](#)
- [Iam Section](#)
- [LoginNodes Section](#)
- [Monitoring Section](#)
- [Tags Section](#)
- [AdditionalPackages Section](#)
- [DirectoryService Section](#)
- [DeploymentSettings Section](#)

## Propriétés du fichier de configuration du cluster

### Region(Facultatif,String)

Spécifie le Région AWS pour le cluster. Par exemple, us-east-2.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

### CustomS3Bucket(Facultatif,String)

Spécifie le nom d'un compartiment Amazon S3 créé dans votre espace Compte AWS pour stocker les ressources utilisées par vos clusters, telles que le fichier de configuration du cluster. AWS ParallelCluster gère un compartiment Amazon S3 dans chaque compartiment dans Région AWS lequel vous créez des clusters. Par défaut, ces compartiments Amazon S3 sont nommésparallelcluster-*hash*-v1-D0-NOT-DELETE.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée. Si vous forcez la mise à jour, la nouvelle valeur sera ignorée et l'ancienne valeur sera utilisée.](#)

### AdditionalResources(Facultatif,String)

Définit un AWS CloudFormation modèle supplémentaire à lancer en même temps que le cluster. Ce modèle supplémentaire est utilisé pour créer des ressources extérieures au cluster mais qui font partie du cycle de vie du cluster.

La valeur doit être une URL HTTPS vers un modèle public, avec tous les paramètres fournis.

Il n'existe aucune valeur par défaut.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## Imds Section

(Facultatif) Spécifie la configuration du service global de métadonnées d'instance (IMDS).

Imds:

ImdsSupport: *string*

### Propriétés de **Imds**

ImdsSupport(Facultatif,String)

Spécifie les versions IMDS prises en charge dans les nœuds du cluster. Les valeurs prises en charge sont `v1.0` et `v2.0`. La valeur par défaut est `v2.0`.

Si `ImdsSupport` ce paramètre est défini sur `v1.0`, `IMDSv1` et `IMDSv2` sont tous deux pris en charge.

Si `ImdsSupport` ce paramètre est défini sur `v2.0`, seul `IMDSv2` est pris en charge.

Pour plus d'informations, consultez la section [Utiliser IMDSv2](#) dans le guide de l'utilisateur Amazon EC2 pour les instances Linux.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

#### Note

À partir de la AWS ParallelCluster version 3.7.0, la valeur `ImdsSupport` par défaut est `v2.0`. Nous vous recommandons de définir `v2.0` et `ImdsSupport` de remplacer `IMDSv1` par `IMDSv2` dans vos appels d'actions personnalisés.

Support pour [Imds/ImdsSupport](#) est ajouté avec AWS ParallelCluster la version 3.3.0.

## Image Section

(Obligatoire) Définit le système d'exploitation du cluster.

Image:

Os: *string*

CustomAmi: *string*

## Propriétés de **Image**

Os(Obligatoire,String)

Spécifie le système d'exploitation à utiliser pour le cluster. Les valeurs prises en charge sont `alinux2`,`alinux2023`,`centos7`,`ubuntu2204`,`ubuntu2004`,`rhel8`,`rocky8`,`rhel9`,`rocky9`.

### Note

RedHat Enterprise Linux 8.7 (`rhel8`) est ajouté à partir de la AWS ParallelCluster version 3.6.0.

Si vous configurez votre cluster pour l'utiliser `rhel`, le coût à la demande pour tout type d'instance est plus élevé que lorsque vous configurez votre cluster pour utiliser d'autres systèmes d'exploitation pris en charge. Pour plus d'informations sur la tarification, consultez les [sections Tarification à la demande](#) et [Quels sont les prix et offres de Red Hat Enterprise Linux sur Amazon EC2 ?](#).

RedHat Enterprise Linux 9 (`rhel9`) est ajouté à partir de la AWS ParallelCluster version 3.9.0.

À l'exception des informations spécifiques Régions AWS mentionnées dans le tableau suivant qui ne sont pas prises en charge `centos7`. Toutes les autres régions AWS commerciales prennent en charge tous les systèmes d'exploitation suivants.

| Cloison (Régions AWS)                                                 | <code>alinux2</code> | <code>centos7</code> | <code>ubuntu2204</code> et <code>ubuntu2004</code> | <code>rhel8</code> | <code>rhel9</code> | <code>alinux2023</code> |
|-----------------------------------------------------------------------|----------------------|----------------------|----------------------------------------------------|--------------------|--------------------|-------------------------|
| Commercial (tout cela Régions AWS n'est pas spécifiquement mentionné) | True                 | True                 | True                                               | True               | True               | True                    |
| AWS GovCloud (USA Est) ( <code>us-gov-east-1</code> )                 | True                 | False                | True                                               | True               | True               | True                    |

| Cloison (Régions AWS)                    | alinux2 | centos7 | ubuntu224 et ubuntu204 | rhel8 | rhel9 | alinux2023 |
|------------------------------------------|---------|---------|------------------------|-------|-------|------------|
| AWS GovCloud (US-Ouest) (us-gov-west-1 ) | True    | False   | True                   | True  | True  | True       |
| Chine (Beijing) (cn-north-1 )            | True    | False   | True                   | True  | True  | True       |
| Chine (Ningxia) (cn-northwest-1 )        | True    | False   | True                   | True  | True  | True       |

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

#### Note

AWS ParallelCluster La version 3.8.0 prend en charge Rocky Linux 8, mais les AMI Rocky Linux 8 prédéfinies (pour les architectures x86 et ARM) ne sont pas disponibles. AWS ParallelCluster La version 3.8.0 prend en charge la création de clusters avec Rocky Linux 8 à l'aide d'AMI personnalisées. Pour plus d'informations, reportez-vous à [Considérations relatives au système d'exploitation](#). AWS ParallelCluster La version 3.9.0 prend en charge Rocky Linux 9, mais les AMI Rocky Linux 9 prédéfinies (pour les architectures x86 et ARM) ne sont pas disponibles. AWS ParallelCluster La version 3.9.0 prend en charge la création de clusters avec Rocky Linux 9 à l'aide d'AMI personnalisées. Pour plus d'informations, reportez-vous à la section [Considérations relatives au système d'exploitation](#).

### CustomAmi(Facultatif,String)

Spécifie l'ID d'une AMI personnalisée à utiliser pour les nœuds de tête et de calcul au lieu de l'AMI par défaut. Pour plus d'informations, consultez [AWS ParallelCluster Personnalisation de l'AMI](#).

Si l'AMI personnalisée nécessite des autorisations supplémentaires pour son lancement, ces autorisations doivent être ajoutées aux politiques de l'utilisateur et du nœud principal.



Par exemple, si un instantané chiffré est associé à une AMI personnalisée, les politiques supplémentaires suivantes sont requises à la fois dans les politiques de l'utilisateur et du nœud principal :

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "kms:DescribeKey",
 "kms:ReEncrypt*",
 "kms:CreateGrant",
 "kms:Decrypt"
],
 "Resource": [
 "arn:aws:kms:<AWS_REGION>:<AWS_ACCOUNT_ID>:key/<AWS_KMS_KEY_ID>"
]
 }
]
}
```

Pour créer une AMI personnalisée RedHat Enterprise Linux, vous devez configurer le système d'exploitation pour installer les packages fournis par les référentiels RHUI (AWS) : `rhel-<version>-baseos-rhui-rpms` `rhel-<version>-appstream-rhui-rpms`, et `codeready-builder-for-rhel-<version>-rhui-rpms`. De plus, les référentiels de l'AMI personnalisée doivent contenir des `kernel-devel` packages de la même version que la version du noyau en cours d'exécution. `kernel`.

Limitations connues :

- Seule RHEL 8.2 et les versions ultérieures prennent en charge FSx for Lustre.
- La version 4.18.0-425.3.1.el8 du noyau RHEL 8.7 ne prend pas en charge FSx for Lustre.
- Seule RHEL 8.4 et les versions ultérieures prennent en charge l'EFA.
- AL23 ne prend pas en charge NICE DCV, car il n'inclut pas d'environnement de bureau graphique, nécessaire pour exécuter NICE DCV. Pour plus d'informations, consultez la documentation officielle de [NICE DCV](#).

Pour résoudre les problèmes liés aux avertissements de validation d'AMI personnalisés, consultez [Résolution des problèmes liés aux AMI personnalisées](#).

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

## HeadNode Section

(Obligatoire) Spécifie la configuration du nœud principal.

```
HeadNode:
 InstanceType: string
 Networking:
 SubnetId: string
 ElasticIp: string/boolean
 SecurityGroups:
 - string
 AdditionalSecurityGroups:
 - string
 Proxy:
 HttpProxyAddress: string
 DisableSimultaneousMultithreading: boolean
 Ssh:
 KeyName: string
 AllowedIps: string
 LocalStorage:
 RootVolume:
 Size: integer
 Encrypted: boolean
 VolumeType: string
 Iops: integer
 Throughput: integer
 DeleteOnTermination: boolean
 EphemeralVolume:
 MountDir: string
 SharedStorageType: string
 Dcv:
 Enabled: boolean
 Port: integer
 AllowedIps: string
 CustomActions:
 OnNodeStart:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
```

```
Args:
- string
OnNodeConfigured:
Sequence:
- Script: string
Args:
- string
Script: string
Args:
- string
OnNodeUpdated:
Sequence:
- Script: string
Args:
- string
Script: string
Args:
- string
Iam:
InstanceRole: string
InstanceProfile: string
S3Access:
- BucketName: string
EnableWriteAccess: boolean
KeyName: string
AdditionalIamPolicies:
- Policy: string
Ims:
Secured: boolean
Image:
CustomAmi: string
```

## Propriétés de **HeadNode**

### InstanceType(Obligatoire,String)

Spécifie le type d'instance pour le nœud principal.

Spécifie le type d'instance Amazon EC2 utilisé pour le nœud principal. L'architecture du type d'instance doit être identique à celle utilisée pour le Slurm [InstanceType](#) paramètre AWS Batch [InstanceType](#) or.

**Note**

AWS ParallelCluster ne prend pas en charge les types d'instances suivants pour le HeadNode paramètre.

- hpc6id

Si vous définissez un type d'instance p4d ou un autre type d'instance doté de plusieurs interfaces réseau ou d'une carte d'interface réseau, vous devez configurer `ElasticIp>true` pour fournir un accès public. AWS les adresses IP publiques ne peuvent être attribuées qu'aux instances lancées avec une seule interface réseau. Dans ce cas, nous vous recommandons d'utiliser une [passerelle NAT](#) pour fournir un accès public aux nœuds de calcul du cluster. Pour plus d'informations, consultez la section [Attribuer une adresse IPv4 publique lors du lancement de l'instance](#) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

`DisableSimultaneousMultithreading`(Facultatif, Boolean)

Si `true`, désactive l'hyperthreading sur le nœud principal. La valeur par défaut est `false`.

Tous les types d'instance ne peuvent pas désactiver l'hyper-threading. Pour obtenir la liste des types d'instances qui prennent en charge la désactivation de l'hyperthreading, consultez la section [Cœurs et threads de processeur pour chaque cœur de processeur par type d'instance](#) dans le guide de l'utilisateur Amazon EC2.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

`SharedStorageType`(Facultatif, String)

Spécifie le type de stockage utilisé pour les données partagées en interne. Les données partagées en interne incluent les données AWS ParallelCluster utilisées pour gérer le cluster et les données partagées par défaut `/home` si elles ne sont pas spécifiées dans [SharedStorage Section](#) le répertoire `Mount` pour monter un volume de système de fichiers partagé. Pour plus de détails sur les données partagées internes, reportez-vous à [AWS ParallelCluster Répertoires internes](#).

S'il s'agit du type de stockage par défaut, le nœud principal exportera des parties de son volume racine sous forme de répertoires partagés pour les nœuds de calcul et les nœuds de connexion utilisant NFS.

Si Efs, Parallelcluster créera un système de fichiers EFS à utiliser pour le partage de données internes et. /home

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

### Note

Lorsque le cluster prend de l'ampleur, le type de stockage EBS peut présenter des problèmes de performances car le nœud principal partage les données du volume racine avec les nœuds de calcul à l'aide d'exportations NFS. Grâce à EFS, vous pouvez éviter les exportations NFS à mesure que votre cluster s'agrandit et éviter les goulots d'étranglement liés aux performances qui leur sont associés. Il est recommandé de choisir EBS pour un potentiel de lecture/écriture maximal pour les petits fichiers et le processus d'installation. Choisissez EFS pour l'échelle.

## Networking

(Obligatoire) Définit la configuration réseau du nœud principal.

### Networking:

SubnetId: *string*

ElasticIp: *string/boolean*

SecurityGroups:

- *string*

AdditionalSecurityGroups:

- *string*

Proxy:

HttpProxyAddress: *string*

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

## Propriétés de **Networking**

SubnetId(Obligatoire,String)

Spécifie l'ID d'un sous-réseau existant dans lequel le nœud principal doit être provisionné.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

## ElasticIp(Facultatif,String)

Crée ou attribue une adresse IP élastique au nœud principal. Les valeurs prises en charge sont `true`/`false`, ou l'ID d'une adresse IP élastique existante. L'argument par défaut est `false`.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## SecurityGroups(Facultatif,[String])

Liste des identifiants de groupe de sécurité Amazon VPC à utiliser pour le nœud principal. Ils remplacent les groupes de sécurité AWS ParallelCluster créés si cette propriété n'est pas incluse.

Vérifiez que les groupes de sécurité sont correctement configurés pour vos [SharedStorage](#) systèmes.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## AdditionalSecurityGroups(Facultatif,[String])

Liste des identifiants de groupe de sécurité Amazon VPC supplémentaires à utiliser pour le nœud principal.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## Proxy(Facultatif)

Spécifie les paramètres du proxy pour le nœud principal.

```
Proxy:
HttpProxyAddress: string
```

## HttpProxyAddress(Facultatif,String)

Définit un serveur proxy HTTP ou HTTPS, généralement `https://x.x.x.x:8080`.

Il n'existe aucune valeur par défaut.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## Ssh

(Facultatif) Définit la configuration de l'accès SSH au nœud principal.

```
Ssh:
```

```
KeyName: string
AllowedIps: string
```

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## Propriétés de **Ssh**

KeyName(Facultatif,String)

Nomme une paire de clés Amazon EC2 existante pour permettre l'accès SSH au nœud principal.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

AllowedIps(Facultatif,String)

Spécifie la plage d'adresses IP au format CIDR ou un identifiant de liste de préfixes pour les connexions SSH au nœud principal. L'argument par défaut est `0.0.0.0/0`.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## LocalStorage

(Facultatif) Définit la configuration de stockage local pour le nœud principal.

```
LocalStorage:
 RootVolume:
 Size: integer
 Encrypted: boolean
 VolumeType: string
 Iops: integer
 Throughput: integer
 DeleteOnTermination: boolean
 EphemeralVolume:
 MountDir: string
```

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## Propriétés de **LocalStorage**

RootVolume(Obligatoire)

Spécifie le volume de stockage racine pour le nœud principal.

**RootVolume:****Size:** *integer***Encrypted:** *boolean***VolumeType:** *string***Iops:** *integer***Throughput:** *integer***DeleteOnTermination:** *boolean*

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

**Size(Facultatif,Integer)**

Spécifie la taille du volume racine du nœud principal en gibioctets (GiB). La taille par défaut provient de l'AMI. L'utilisation d'une taille différente nécessite que l'AMI soit compatible `growroot`.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

**Encrypted(Facultatif,Boolean)**

Spécifie si le volume racine est chiffré. La valeur par défaut est `true`.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

**VolumeType(Facultatif,String)**

Spécifie le [type de volume Amazon EBS](#). Les valeurs prises en charge sont `gp2`, `gp3io1`, `io2`, `sc1`, `st1`, et `standard`. La valeur par défaut est `gp3`.

Pour plus d'informations, consultez [Types de volumes Amazon EBS](#) dans le Guide de l'utilisateur Amazon EC2.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

**Iops(Facultatif,Integer)**

Définit le nombre d'IOPS pour `io1io2`, et le `gp3` type de volumes.

La valeur par défaut, les valeurs prises en charge et `volume_iops` le `volume_size` ratio de production varient selon `VolumeType` et `Size`.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

`VolumeType = io1`

Valeur par défaut Iops = 100



Valeurs prises en charge Iops = 100—64 000 †

SizeRapport Iops maximum = 50 IOPS par GiB. 5000 IOPS nécessitent au Size moins 100 GiB.

VolumeType = io2

Valeur par défaut Iops = 100

Valeurs prises en charge Iops = 100 à 64 000 (256 000 pour les volumes io2 Block Express) †

SizeRapport Iops maximum = 500 IOPS par GiB. 5000 IOPS nécessitent au Size moins 10 GiB.

VolumeType = gp3

Valeur par défaut Iops = 3000

Valeurs prises en charge Iops = 3000 à 16 000

SizeRapport Iops maximum = 500 IOPS par GiB. 5000 IOPS nécessitent au Size moins 10 GiB.

† Le maximum d'IOPS est garanti uniquement sur les [instances basées sur le système Nitro](#) et approvisionnées avec plus de 32 000 IOPS. Les autres instances garantissent jusqu'à 32 000 IOPS. Les anciens io1 volumes risquent de ne pas atteindre leurs performances optimales à moins que vous ne [modifiez le volume](#). io2 Les volumes Block Express prennent en charge Iops des valeurs allant jusqu'à 256 000 pour les types d'R5instances. Pour plus d'informations, consultez la section sur les [volumes io2 Block Express](#) dans le guide de l'utilisateur Amazon EC2.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

Throughput(Facultatif,Integer)

Définit le débit pour les types de gp3 volumes, en Mbits/s. Ce paramètre n'est valide que lorsqu'il l'VolumeTypeestgp3. La valeur par défaut est 125. Valeurs prises en charge : 125-1000 MiB/s

Le rapport de Throughput à ne Iops peut pas être supérieur à 0,25. Le débit maximal de 1 000 Mbits/s nécessite que le Iops paramètre soit d'au moins 4 000.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## DeleteOnTermination(Facultatif,Boolean)

Spécifie si le volume racine doit être supprimé lorsque le nœud principal est arrêté. La valeur par défaut est `true`.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## EphemeralVolume(Facultatif)

Spécifie les détails de tout volume de stockage d'instance. Pour plus d'informations, consultez [Volumes de stockage d'instances](#) dans le Guide de l'utilisateur Amazon EC2.

[EphemeralVolume](#):

[MountDir](#): *string*

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## MountDir(Facultatif,String)

Spécifie le répertoire de montage pour le volume de stockage de l'instance. L'argument par défaut est `/scratch`.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## Dcv

(Facultatif) Définit les paramètres de configuration pour le serveur DCV NICE exécuté sur le nœud principal.

Pour plus d'informations, consultez [Connectez-vous au nœud principal via NICE DCV](#).

[Dcv](#):

[Enabled](#): *boolean*

[Port](#): *integer*

[AllowedIps](#): *string*

### Important

Par défaut, le port DCV NICE configuré par AWS ParallelCluster est ouvert à toutes les adresses IPv4. Toutefois, vous ne pouvez vous connecter à un port NICE DCV que si vous disposez de l'URL de la session DCV NICE et si vous vous connectez à la session NICE

DCV dans les 30 secondes suivant le renvoi de l'URL. `pccluster dcv-connect` Utilisez le `AllowedIps` paramètre pour restreindre davantage l'accès au port DCV NICE avec une plage d'adresses IP au format CIDR, et utilisez-le pour `Port` définir un port non standard.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## Propriétés de **Dcv**

`Enabled`(Obligatoire, Boolean)

Spécifie si NICE DCV est activé sur le nœud principal. La valeur par défaut est `false`.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

### Note

NICE DCV génère automatiquement un certificat auto-signé qui est utilisé pour sécuriser le trafic entre le client NICE DCV et le serveur NICE DCV exécuté sur le nœud principal. Pour configurer votre propre certificat, veuillez consulter [Certificat HTTPS NICE DCV](#).

`Port`(Facultatif, Integer)

Spécifie le port pour NICE DCV. La valeur par défaut est 8443.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

`AllowedIps`(Facultatif, recommandé, String)

Spécifie la plage d'adresses IP au format CIDR pour les connexions à NICE DCV. Ce paramètre est utilisé uniquement lors de la AWS ParallelCluster création du groupe de sécurité. La valeur par défaut est `0.0.0.0/0`, ce qui permet l'accès à partir de n'importe quelle adresse Internet.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## CustomActions

(Facultatif) Spécifie les scripts personnalisés à exécuter sur le nœud principal.

[CustomActions:](#)

```
OnNodeStart:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
OnNodeConfigured:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
OnNodeUpdated:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
```

## Propriétés de **CustomActions**

### OnNodeStart(Facultatif)

Spécifie un script unique ou une séquence de scripts à exécuter sur le nœud principal avant le lancement de toute action d'amorçage du déploiement du nœud. Pour plus d'informations, consultez [Actions de bootstrap personnalisées](#).

### Sequence(Facultatif)

Liste des scripts à exécuter. AWS ParallelCluster exécute les scripts dans l'ordre indiqué dans le fichier de configuration, en commençant par le premier.

### Script(Obligatoire,String)

Spécifie le fichier à utiliser. Le chemin du fichier peut commencer par `https://` ou `s3://`.

### Args(Facultatif,[String])

Liste des arguments à transmettre au script.

### Script(Obligatoire,String)

Spécifie le fichier à utiliser pour un seul script. Le chemin du fichier peut commencer par `https://ous3://`.

### Args(Facultatif,[String])

Liste des arguments à transmettre au script unique.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

### OnNodeConfigured(Facultatif)

Spécifie un script unique ou une séquence de scripts à exécuter sur le nœud principal une fois les actions d'amorçage du nœud terminées. Pour plus d'informations, consultez [Actions de bootstrap personnalisées](#).

### Sequence(Facultatif)

Spécifie la liste des scripts à exécuter.

### Script(Obligatoire,String)

Spécifie le fichier à utiliser. Le chemin du fichier peut commencer par `https://ous3://`.

### Args(Facultatif,[String])

Liste des arguments à transmettre au script.

### Script(Obligatoire,String)

Spécifie le fichier à utiliser pour un seul script. Le chemin du fichier peut commencer par `https://ous3://`.

### Args(Facultatif,[String])

Liste des arguments à transmettre au script unique.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

### OnNodeUpdated(Facultatif)

Spécifie un script unique ou une séquence de scripts à exécuter sur le nœud principal une fois les actions de mise à jour du nœud terminées. Pour plus d'informations, consultez [Actions de bootstrap personnalisées](#).

### Sequence(Facultatif)

Spécifie la liste des scripts à exécuter.

**Script(Obligatoire,String)**

Spécifie le fichier à utiliser. Le chemin du fichier peut commencer par `https://ous3://`.

**Args(Facultatif,[String])**

Liste des arguments à transmettre au script.

**Script(Obligatoire,String)**

Spécifie le fichier à utiliser pour le script unique. Le chemin du fichier peut commencer par `https://ous3://`.

**Args(Facultatif,[String])**

Liste des arguments à transmettre au script unique.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

**Note**

`OnNodeUpdate` est ajouté à partir de la AWS ParallelCluster version 3.4.0.  
`Sequence` est ajouté à partir de AWS ParallelCluster la version 3.6.0. Lorsque vous le spécifiez `Sequence`, vous pouvez répertorier plusieurs scripts pour une action personnalisée. AWS ParallelCluster continue de prendre en charge la configuration d'une action personnalisée avec un seul script, sans inclure `Sequence`.  
AWS ParallelCluster ne permet pas d'inclure à la fois un seul script et `Sequence` pour la même action personnalisée.

**Iam**

(Facultatif) Spécifie un rôle d'instance ou un profil d'instance à utiliser sur le nœud principal pour remplacer le rôle d'instance ou le profil d'instance par défaut du cluster.

**Iam:**

```
InstanceRole: string
InstanceProfile: string
S3Access:
- BucketName: string
 EnableWriteAccess: boolean
 KeyName: string
AdditionalIamPolicies:
```

- `Policy`: *string*

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## Propriétés de **Iam**

### InstanceProfile(Facultatif,String)

Spécifie un profil d'instance pour remplacer le profil d'instance du nœud principal par défaut. Vous ne pouvez pas spécifier à la fois InstanceProfile et InstanceRole. Le format est `arn:Partition:iam::Account:instance-profile/InstanceProfileName`.

Si cela est spécifié, les AdditionalIamPolicies paramètres S3Access et ne peuvent pas être spécifiés.

Nous vous recommandons de spécifier l'un des AdditionalIamPolicies paramètres S3Access et, ou les deux, car les fonctionnalités ajoutées nécessitent AWS ParallelCluster souvent de nouvelles autorisations.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

### InstanceRole(Facultatif,String)

Spécifie un rôle d'instance pour remplacer le rôle d'instance de nœud principal par défaut. Vous ne pouvez pas spécifier à la fois InstanceProfile et InstanceRole. Le format est `arn:Partition:iam::Account:role/RoleName`.

Si cela est spécifié, les AdditionalIamPolicies paramètres S3Access et ne peuvent pas être spécifiés.

Nous vous recommandons de spécifier l'un des AdditionalIamPolicies paramètres S3Access et, ou les deux, car les fonctionnalités ajoutées nécessitent AWS ParallelCluster souvent de nouvelles autorisations.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## **S3Access**

### S3Access(Facultatif)

Spécifie un compartiment. Ceci est utilisé pour générer des politiques visant à accorder l'accès spécifié au compartiment.

Si cela est spécifié, les InstanceRole paramètres InstanceProfile et ne peuvent pas être spécifiés.

Nous vous recommandons de spécifier l'un des AdditionalIamPolicies paramètres S3Access et, ou les deux, car les fonctionnalités ajoutées nécessitent AWS ParallelCluster souvent de nouvelles autorisations.

#### S3Access:

- BucketName: *string*
- EnableWriteAccess: *boolean*
- KeyName: *string*

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

BucketName(Obligatoire,String)

Le nom du compartiment .

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

KeyName(Facultatif,String)

La clé du seau. La valeur par défaut est « \* ».

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

EnableWriteAccess(Facultatif,Boolean)

Indique si l'accès en écriture est activé pour le compartiment. La valeur par défaut est false.

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

## **AdditionalIamPolicies**

AdditionalIamPolicies(Facultatif)

Spécifie une liste des Amazon Resource Names (ARN) des politiques IAM pour Amazon EC2. Cette liste est attachée au rôle root utilisé pour le nœud principal en plus des autorisations requises par AWS ParallelCluster.

Le nom d'une politique IAM et son ARN sont différents. Les noms ne peuvent pas être utilisés.

Si cela est spécifié, les InstanceRole paramètres InstanceProfile et ne peuvent pas être spécifiés.



Nous vous recommandons d'utiliser `AdditionalIamPolicies` car `AdditionalIamPolicies` ils sont ajoutés aux autorisations AWS ParallelCluster requises et `InstanceRole` doivent inclure toutes les autorisations requises. Les autorisations requises changent souvent d'une version à l'autre au fur et à mesure que des fonctions sont ajoutées.

Il n'existe aucune valeur par défaut.

`AdditionalIamPolicies`:

- `Policy`: *string*

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

`Policy`(Facultatif,[String])

Liste des politiques IAM.

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

## **Imds**

(Facultatif) Spécifie les propriétés du service de métadonnées d'instance (IMDS). Pour plus d'informations, consultez [Comment fonctionne le service de métadonnées d'instance version 2](#) dans le guide de l'utilisateur Amazon EC2.

`Imds`:

`Secured`: *boolean*

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

Propriétés de **Imds**

`Secured`(Facultatif,Boolean)

Si `true`, restreint l'accès à l'IMDS du nœud principal (et aux informations d'identification du profil d'instance) à un sous-ensemble de superutilisateurs.

Si `false`, chaque utilisateur du nœud principal a accès à l'IMDS du nœud principal.

Les utilisateurs suivants sont autorisés à accéder à l'IMDS du nœud principal :

- utilisateur root
- utilisateur administratif du cluster (`pc-cluster-admin` par défaut)

- utilisateur par défaut spécifique au système d'exploitation (ec2-user sur Amazon Linux 2 et RedHat, ubuntu sur Ubuntu 18.04, centos sur CentOS 7)

L'argument par défaut est `true`.

Les `default` utilisateurs sont chargés de s'assurer qu'un cluster dispose des autorisations nécessaires pour interagir avec les AWS ressources. Si vous désactivez `default` l'accès IMDS de l'utilisateur, vous ne pouvez pas gérer les nœuds de calcul et cessez de fonctionner. Ne désactivez pas l'accès IMDS de `default` l'utilisateur.

Lorsqu'un utilisateur obtient l'accès à l'IMDS du nœud principal, il peut utiliser les autorisations incluses dans le [profil d'instance du nœud principal](#). Par exemple, ils peuvent utiliser ces autorisations pour lancer des instances Amazon EC2 ou pour lire le mot de passe d'un domaine AD que le cluster est configuré pour utiliser à des fins d'authentification.

Pour restreindre l'accès à l'IMDS, AWS ParallelCluster gère une chaîne de règles iptables.

Les utilisateurs du cluster disposant `sudo` d'un accès peuvent activer ou désactiver de manière sélective l'accès à l'IMDS du nœud principal pour d'autres utilisateurs individuels, y compris `default` les utilisateurs, en exécutant la commande suivante :

```
$ sudo /opt/parallelcluster/scripts/imds/imds-access.sh --allow <USERNAME>
```

Vous pouvez désactiver l'accès IMDS de l'utilisateur à l'option `--deny` de cette commande.

Si vous désactivez sans le savoir l'accès IMDS de `default` l'utilisateur, vous pouvez rétablir l'autorisation en utilisant cette option. `--allow`

#### Note

Toute personnalisation `iptables` ou `ip6tables` règle peut interférer avec le mécanisme utilisé pour restreindre l'accès à l'IMDS sur le nœud principal.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## Image

(Facultatif) Définit une image personnalisée pour le nœud principal.

Image:

CustomAmi: *string*

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

## Propriétés de **Image**

CustomAmi(Facultatif,String)

Spécifie l'ID d'une AMI personnalisée à utiliser pour le nœud principal au lieu de l'AMI par défaut. Pour plus d'informations, consultez [AWS ParallelCluster Personnalisation de l'AMI](#).

Si l'AMI personnalisée nécessite des autorisations supplémentaires pour son lancement, ces autorisations doivent être ajoutées aux politiques de l'utilisateur et du nœud principal.

Par exemple, si un instantané chiffré est associé à une AMI personnalisée, les politiques supplémentaires suivantes sont requises à la fois dans les politiques de l'utilisateur et du nœud principal :

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "kms:DescribeKey",
 "kms:ReEncrypt*",
 "kms:CreateGrant",
 "kms:Decrypt"
],
 "Resource": [
 "arn:aws:kms:<AWS_REGION>:<AWS_ACCOUNT_ID>:key/<AWS_KMS_KEY_ID>"
]
 }
]
}
```

Pour résoudre les problèmes liés aux avertissements de validation d'AMI personnalisés, consultez [Résolution des problèmes liés aux AMI personnalisées](#).

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

## Scheduling Section

(Obligatoire) Définit le planificateur de tâches utilisé dans le cluster et les instances de calcul qu'il gère. Vous pouvez utiliser le planificateur Slurm ou le AWS Batch planificateur. Chacun prend en charge un ensemble différent de paramètres et de propriétés.

### Rubriques

- [Propriétés de Scheduling](#)
- [AwsBatchQueues](#)
- [SlurmQueues](#)
- [SlurmSettings](#)

#### Scheduling:

Scheduler: `slurm`

ScalingStrategy: `string`

#### SlurmSettings:

MungeKeySecretArn: `string`

ScaledownIdleTime: `integer`

QueueUpdateStrategy: `string`

EnableMemoryBasedScheduling: `boolean`

CustomSlurmSettings: `[dict]`

CustomSlurmSettingsIncludeFile: `string`

#### Database:

Uri: `string`

UserName: `string`

PasswordSecretArn: `string`

DatabaseName: `string`

ExternalSlurmdbd: `boolean`

Host: `string`

Port: `integer`

#### Dns:

DisableManagedDns: `boolean`

HostedZoneId: `string`

UseEc2Hostnames: `boolean`

#### SlurmQueues:

- Name: `string`

#### ComputeSettings:

#### LocalStorage:

RootVolume:

Size: `integer`

```
Encrypted: boolean
VolumeType: string
Iops: integer
Throughput: integer
EphemeralVolume:
 MountDir: string
CapacityReservationTarget:
 CapacityReservationId: string
 CapacityReservationResourceGroupArn: string
CapacityType: string
AllocationStrategy: string
JobExclusiveAllocation: boolean
CustomSlurmSettings: dict
Tags:
 - Key: string
 Value: string
HealthChecks:
 Gpu:
 Enabled: boolean
Networking:
 SubnetIds:
 - string
 AssignPublicIp: boolean
 SecurityGroups:
 - string
 AdditionalSecurityGroups:
 - string
 PlacementGroup:
 Enabled: boolean
 Id: string
 Name: string
 Proxy:
 HttpProxyAddress: string
ComputeResources:
 - Name: string
 InstanceType: string
 Instances:
 - InstanceType: string
 MinCount: integer
 MaxCount: integer
 DynamicNodePriority: integer
 StaticNodePriority: integer
 SpotPrice: float
 DisableSimultaneousMultithreading: boolean
```

```
SchedulableMemory: integer
HealthChecks:
 Gpu:
 Enabled: boolean
 Efa:
 Enabled: boolean
 GdrSupport: boolean
CapacityReservationTarget:
 CapacityReservationId: string
 CapacityReservationResourceGroupArn: string
Networking:
 PlacementGroup:
 Enabled: boolean
 Name: string
CustomSlurmSettings: dict
Tags:
 - Key: string
 Value: string
CustomActions:
 OnNodeStart:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeConfigured:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
Iam:
 InstanceProfile: string
 InstanceRole: string
 S3Access:
 - BucketName: string
 EnableWriteAccess: boolean
 KeyName: string
 AdditionalIamPolicies:
 - Policy: string
```

**Image:**

**CustomAmi:** *string*

**Scheduling:**

**Scheduler:** awsbatch

**AwsBatchQueues:**

- **Name:** *string*

**CapacityType:** *string*

**Networking:**

**SubnetIds:**

- *string*

**AssignPublicIp:** *boolean*

**SecurityGroups:**

- *string*

**AdditionalSecurityGroups:**

- *string*

**ComputeResources:** # this maps to a Batch compute environment (initially we support only 1)

- **Name:** *string*

**InstanceTypes:**

- *string*

**MinvCpus:** *integer*

**DesiredvCpus:** *integer*

**MaxvCpus:** *integer*

**SpotBidPercentage:** *float*

## Propriétés de **Scheduling**

### **Scheduler**(Obligatoire,**String**)

Spécifie le type de planificateur utilisé. Les valeurs prises en charge sont slurm et awsbatch.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

#### Note

awsbatch ne prend en charge que le système alinux2 d'exploitation et x86\_64 la plateforme.

## ScalingStrategy(Facultatif,String)

Vous permet de choisir le mode d'Slurmévolution des nœuds dynamiques. Les valeurs prises en charge sont `all-or-nothing`, `greedy-all-or-nothing` et `best-effort`. La valeur par défaut est `all-or-nothing`.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

### Note

La stratégie de dimensionnement s'applique uniquement aux nœuds devant être repris par Slurm, et non aux nœuds qui sont déjà en cours d'exécution.

- `all-or-nothing` Cette stratégie suit strictement un `all-or-nothing-approach`, visant à éviter les instances inactives à la fin du processus de dimensionnement. Il fonctionne sur une `all-or-nothing` base, ce qui signifie qu'il évolue complètement ou pas du tout. Sachez que le lancement temporaire d'instances peut entraîner des coûts supplémentaires lorsque les tâches nécessitent plus de 500 nœuds ou couvrent plusieurs ressources de calcul. Cette stratégie présente le débit le plus faible parmi les trois stratégies de mise à l'échelle possibles. Le délai de mise à l'échelle dépend du nombre de tâches soumises par exécution du programme de Slurm reprise. En outre, vous ne pouvez pas dépasser de loin la limite de compte de `RunInstances` ressources par défaut par exécution, qui est de 1 000 instances par défaut. Vous trouverez plus de détails dans la documentation relative à la [régulation des API Amazon EC2](#).
- `greedy-all-or-nothing` À l'instar de la `all-or-nothing` stratégie, elle vise à éviter les instances inactives après le dimensionnement. Cette stratégie permet de surdimensionner temporairement pendant le processus de dimensionnement afin d'atteindre un débit supérieur à celui de l' `all-or-nothing` approche, mais elle est également assortie de la même limite de mise à l'échelle de 1 000 instances conformément à la limite du compte de `RunInstances` ressources.
- `best-effort` Cette stratégie donne la priorité au débit élevé, même si cela signifie que certaines instances peuvent être inactives à la fin du processus de dimensionnement. Il essaie d'allouer autant de nœuds que demandé par les tâches, mais il est possible que la demande ne soit pas traitée dans son intégralité. Contrairement aux autres stratégies, l'approche basée sur le meilleur effort permet d'accumuler un plus grand nombre d'instances que la `RunInstances` limite standard, au prix de l'inactivité des ressources lors des multiples exécutions des processus de dimensionnement.



Chaque stratégie est conçue pour répondre à différents besoins de mise à l'échelle, ce qui vous permet de sélectionner celle qui répond à vos exigences et contraintes spécifiques.

## AwsBatchQueues

(Facultatif) Les paramètres de la AWS Batch file d'attente. Une seule file d'attente est prise en charge. Si [Scheduler](#) est défini sur `awsbatch`, cette section est obligatoire. Pour plus d'informations sur le `awsbatch` planificateur, consultez les sections [Configuration réseau](#) et [AWS Batch \(awsbatch\)](#)

### AwsBatchQueues:

- Name: *string*
- CapacityType: *string*
- Networking:
  - SubnetIds:
    - *string*
  - AssignPublicIp: *boolean*
  - SecurityGroups:
    - *string*
  - AdditionalSecurityGroups:
    - *string*
- ComputeResources: # this maps to a Batch compute environment (initially we support only 1)
  - Name: *string*
  - InstanceTypes:
    - *string*
  - MinvCpus: *integer*
  - DesiredvCpus: *integer*
  - MaxvCpus: *integer*
  - SpotBidPercentage: *float*

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

## Propriétés de **AwsBatchQueues**

### **Name**(Obligatoire, **String**)

Nom de la AWS Batch file d'attente.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

## CapacityType(Facultatif,String)

Type de ressources de calcul utilisées par la AWS Batch file d'attente. Les valeurs prises en charge sont ONDEMAND, SPOT ou CAPACITY\_BLOCK. La valeur par défaut est ONDEMAND.

### Note

Si vous le définissez CapacityTypeSPOT, votre compte doit contenir un rôle AWSServiceRoleForEC2Spot lié à un service. Vous pouvez créer ce rôle à l'aide de la AWS CLI commande suivante.

```
$ aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Pour plus d'informations, consultez la section [Rôle lié au service pour les demandes d'instance Spot](#) dans le guide de l'utilisateur Amazon EC2 pour les instances Linux.

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

## Networking

(Obligatoire) Définit la configuration réseau de la AWS Batch file d'attente.

### Networking:

#### SubnetIds:

- *string*

AssignPublicIp: *boolean*

#### SecurityGroups:

- *string*

#### AdditionalSecurityGroups:

- *string*

## Propriétés de Networking

### SubnetIds(Obligatoire,[String])

Spécifie l'ID d'un sous-réseau existant dans lequel approvisionner la AWS Batch file d'attente. Actuellement, un seul sous-réseau est pris en charge.

Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.

### **AssignPublicIp**(Facultatif,**String**)

Crée ou attribue une adresse IP publique aux nœuds de la AWS Batch file d'attente. Les valeurs prises en charge sont `true` et `false`. La valeur par défaut dépend du sous-réseau que vous avez spécifié.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

### **SecurityGroups**(Facultatif,**[String]**)

Liste des groupes de sécurité utilisés par la AWS Batch file d'attente. Si vous ne spécifiez aucun groupe de sécurité, AWS ParallelCluster crée de nouveaux groupes de sécurité.

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

### **AdditionalSecurityGroups**(Facultatif,**[String]**)

Liste des groupes de sécurité utilisés par la AWS Batch file d'attente.

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

## **ComputeResources**

(Obligatoire) Définit la `ComputeResources` configuration de la AWS Batch file d'attente.

```
ComputeResources: # this maps to a Batch compute environment (initially we support only 1)
- Name: string
 InstanceTypes:
 - string
 MinvCpus: integer
 DesiredvCpus: integer
 MaxvCpus: integer
 SpotBidPercentage: float
```

### Propriétés de **ComputeResources**

#### **Name**(Obligatoire,**String**)

Nom de l'environnement de calcul de la AWS Batch file d'attente.

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

### **InstanceTypes**(Obligatoire, **[String]**)

Le tableau des types d'instances de l'environnement de AWS Batch calcul. Tous les types d'instances doivent utiliser l'x86\_64architecture.

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

### **MinvCpus**(Facultatif, **Integer**)

Nombre minimal de processeurs virtuels qu'un environnement AWS Batch informatique peut utiliser.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

### **DesiredVcpus**(Facultatif, **Integer**)

Nombre de vCPU souhaités dans l'environnement AWS Batch informatique. AWS Batch ajuste cette valeur entre `MinvCpus` et en `MaxvCpus` fonction de la demande dans la file d'attente des tâches.

[Politique de mise à jour : ce paramètre n'est pas analysé lors d'une mise à jour.](#)

### **MaxvCpus**(Facultatif, **Integer**)

Nombre maximal de processeurs virtuels pour l'environnement AWS Batch informatique. Vous ne pouvez pas définir une valeur inférieure à `DesiredVcpus`.

[Politique de mise à jour : ce paramètre ne peut pas être réduit lors d'une mise à jour.](#)

### **SpotBidPercentage**(Facultatif, **Float**)

Pourcentage maximum du prix à la demande pour le type d'instance que le prix d'une instance Amazon EC2 Spot peut atteindre avant le lancement des instances. La valeur par défaut est 100 (100 %). La plage prise en charge est de 1 -100.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## **SlurmQueues**

(Facultatif) Paramètres de la Slurm file d'attente. S'[Scheduler](#)il est défini sur `slurm`, cette section est obligatoire.

SlurmQueues:

- Name: *string*

ComputeSettings:LocalStorage:RootVolume:

Size: *integer*

Encrypted: *boolean*

VolumeType: *string*

Iops: *integer*

Throughput: *integer*

EphemeralVolume:

MountDir: *string*

CapacityReservationTarget:

CapacityReservationId: *string*

CapacityReservationResourceGroupArn: *string*

CapacityType: *string*

AllocationStrategy: *string*

JobExclusiveAllocation: *boolean*

CustomSlurmSettings: *dict*

Tags:

- Key: *string*

Value: *string*

HealthChecks:Gpu:

Enabled: *boolean*

Networking:SubnetIds:

- *string*

AssignPublicIp: *boolean*

SecurityGroups:

- *string*

AdditionalSecurityGroups:

- *string*

PlacementGroup:

Enabled: *boolean*

Id: *string*

Name: *string*

Proxy:

HttpProxyAddress: *string*

ComputeResources:

- Name: *string*

InstanceType: *string*

Instances:

```
- InstanceType: string
MinCount: integer
MaxCount: integer
DynamicNodePriority: integer
StaticNodePriority: integer
SpotPrice: float
DisableSimultaneousMultithreading: boolean
SchedulableMemory: integer
HealthChecks:
 Gpu:
 Enabled: boolean
 Efa:
 Enabled: boolean
 GdrSupport: boolean
CapacityReservationTarget:
 CapacityReservationId: string
 CapacityReservationResourceGroupArn: string
Networking:
 PlacementGroup:
 Enabled: boolean
 Name: string
CustomSlurmSettings: dict
Tags:
 - Key: string
 Value: string
CustomActions:
 OnNodeStart:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeConfigured:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
Iam:
 InstanceProfile: string
```

```
InstanceRole: string
S3Access:
- BucketName: string
 EnableWriteAccess: boolean
 KeyName: string
AdditionalIamPolicies:
- Policy: string
Image:
 CustomAmi: string
```

Politique de mise à jour : pour ce paramètre de valeurs de liste, une nouvelle valeur peut être ajoutée lors d'une mise à jour ou le parc informatique doit être arrêté lors de la suppression d'une valeur existante.

## Propriétés de **SlurmQueues**

### **Name**(Obligatoire,**String**)

Nom de la Slurm file d'attente.

#### Note

La taille du cluster peut changer lors d'une mise à jour. Pour plus d'informations, voir [Taille et mise à jour de la capacité du cluster](#)

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

## **CapacityReservationTarget**

#### Note

CapacityReservationTarget est ajouté avec la AWS ParallelCluster version 3.3.0.

```
CapacityReservationTarget:
 CapacityReservationId: string
 CapacityReservationResourceGroupArn: string
```

Spécifie la réservation de capacité à la demande pour les ressources de calcul de la file d'attente.

## CapacityReservationId(Facultatif,String)

L'ID de la réservation de capacité existante à cibler pour les ressources de calcul de la file d'attente. L'identifiant peut faire référence à un [ODCR](#) ou à un [bloc de capacité pour ML](#).

La réservation doit utiliser la même plateforme que celle utilisée par l'instance. Par exemple, si vos instances s'exécutent sur `rhel8`, votre réservation de capacité doit s'exécuter sur la plate-forme Red Hat Enterprise Linux. Pour de plus amples informations, veuillez consulter [Plateformes prises en charge](#) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

### Note

Si vous incluez [Instances](#) dans la configuration du cluster, vous devez exclure ce `CapacityReservationId` paramètre de niveau de file d'attente de la configuration.

## CapacityReservationResourceGroupArn(Facultatif,String)

Le nom de ressource Amazon (ARN) du groupe de ressources qui sert de groupe lié au service pour les réservations de capacité pour les ressources de calcul de la file d'attente. AWS ParallelCluster identifie et utilise la réservation de capacité la plus appropriée auprès du groupe de ressources en fonction des conditions suivantes :

- Si cette option `PlacementGroup` est activée dans [SlurmQueues/Networking](#) ou [SlurmQueues/ComputeResources/Networking](#), AWS ParallelCluster sélectionne un groupe de ressources qui cible le type d'instance et `PlacementGroup` pour une ressource de calcul, si la ressource de calcul existe.


Ils `PlacementGroup` doivent cibler l'un des types d'instance définis dans [ComputeResources](#).

- Si `PlacementGroup` ce n'est pas activé dans [SlurmQueues/Networking](#) ou [SlurmQueues/ComputeResources/Networking](#), AWS ParallelCluster sélectionne un groupe de ressources qui cible uniquement le type d'instance d'une ressource de calcul, si la ressource de calcul existe.

Le groupe de ressources doit disposer d'au moins un ODCR pour chaque type d'instance réservé dans une zone de disponibilité pour toutes les ressources de calcul et zones de disponibilité de la file d'attente. Pour plus d'informations, consultez [Lancez des instances avec des réservations de capacité à la demande \(ODCR\)](#).



Pour plus d'informations sur les exigences de configuration de plusieurs sous-réseaux, consultez [Networking/SubnetIds](#).


 Note

Plusieurs zones de disponibilité ont été ajoutées dans la AWS ParallelCluster version 3.4.0.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

### **CapacityType**(Facultatif, **String**)

Type de ressources de calcul utilisées par la Slurm file d'attente. Les valeurs prises en charge sont ONDEMAND, SPOT ou CAPACITY\_BLOCK. La valeur par défaut est ONDEMAND.

 Note

Si vous définissez le paramètre CapacityType sur SPOT, votre compte doit avoir un rôle AWSServiceRoleForEC2Spot lié au service. Vous pouvez créer ce rôle à l'aide de la AWS CLI commande suivante.

```
$ aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Pour plus d'informations, consultez la section [Rôle lié au service pour les demandes d'instance Spot](#) dans le guide de l'utilisateur Amazon EC2 pour les instances Linux.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

### **AllocationStrategy**(Facultatif, **String**)

Spécifiez la stratégie d'allocation pour toutes les ressources de calcul définies dans [Instances](#).

Valeurs valides : lowest-price | capacity-optimized | price-capacity-optimized

Par défaut : lowest-price

## lowest-price

- Si vous l'utilisez `CapacityType = ONDEMAND`, Amazon EC2 Fleet utilise le prix pour déterminer la commande et lance d'abord les instances les moins chères.
- Si vous en utilisez `CapacityType = SPOT`, Amazon EC2 Fleet lance des instances à partir du pool d'instances Spot le moins cher disposant de la capacité disponible. Si un pool est à court de capacité avant d'atteindre la capacité requise, Amazon EC2 Fleet répond à votre demande en lançant des instances pour vous. Amazon EC2 Fleet lance notamment des instances à partir du pool d'instances Spot le moins cher disposant de la capacité disponible. Amazon EC2 Fleet peut lancer des instances Spot à partir de plusieurs pools différents.
- Si vous définissez `CapacityType = CAPACITY_BLOCK`, il n'existe aucune stratégie d'allocation, le `AllocationStrategy` paramètre ne peut donc pas être configuré.

## capacity-optimized

- Si vous avez défini `CapacityType = ONDEMAND`, `capacity-optimized` n'est pas disponible.
- Si vous le définissez `CapacityType = SPOT`, Amazon EC2 Fleet lance des instances à partir de pools d'instances Spot avec une capacité optimale compte tenu du nombre d'instances à lancer.

## price-capacity-optimized

- Si vous avez défini `CapacityType = ONDEMAND`, `capacity-optimized` n'est pas disponible.
- Si vous le définissez `CapacityType = SPOT`, Amazon EC2 Fleet identifie les pools présentant la disponibilité de capacité la plus élevée par rapport au nombre d'instances lancées. Cela signifie que nous demanderons des instances Spot auprès des groupes qui, selon nous, présentent le moins de risques d'interruption à court terme. Amazon EC2 Fleet demande ensuite des instances Spot auprès du pool le moins cher.

[Politique de mise à jour : le parc informatique doit être arrêté ou `QueueUpdateStrategy` défini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

### Note

`AllocationStrategy` est pris en charge à partir de AWS ParallelCluster la version 3.3.0.

## JobExclusiveAllocation(Facultatif,String)

S'il est défini sur `true`, l'OverSubscribe indicateur de Slurm partition est défini sur `EXCLUSIVE`. Lorsque `OverSubscribe = EXCLUSIVE`, les tâches de la partition ont un accès exclusif à tous les nœuds alloués. Pour plus d'informations, consultez [EXCLUSIVE](#) dans la Slurm documentation.

Valeurs valides : `true` | `false`

Par défaut: `false`

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

### Note

JobExclusiveAllocation est pris en charge à partir de AWS ParallelCluster la version 3.7.0.

## CustomSlurmSettings(Facultatif,Dict)

Définit les paramètres de configuration de Slurm partition (file d'attente) personnalisés.

Spécifie un dictionnaire de paires clé-valeur de paramètres de Slurm configuration personnalisés qui s'appliquent aux files d'attente (partitions).

Chaque paire clé-valeur distincte, telle que `Param1: Value1`, est ajoutée séparément à la fin de la ligne de configuration de Slurm partition dans le format. `Param1=Value1`

Vous ne pouvez spécifier que des paramètres Slurm de configuration qui ne figurent pas dans la liste refusée. CustomSlurmSettings Pour plus d'informations sur les paramètres de Slurm configuration refusés, consultez. [Listé par DenySlurmparamètres de configuration pourCustomSlurmSettings](#)

AWS ParallelCluster vérifie uniquement si un paramètre figure dans une liste de refus. AWS ParallelCluster ne valide pas la syntaxe ou la sémantique de vos paramètres de Slurm configuration personnalisés. Vous êtes responsable de la validation de vos paramètres de Slurm configuration personnalisés. Des paramètres Slurm de configuration personnalisés non valides peuvent provoquer des défaillances de Slurm daemon susceptibles d'entraîner des échecs de création et de mise à jour de clusters.

Pour plus d'informations sur la façon de spécifier des paramètres Slurm de configuration personnalisés avec AWS ParallelCluster, consultez [Slurmpersonnalisation de la configuration](#).

Pour plus d'informations sur les paramètres Slurm de configuration, consultez [slurm.conf](#) dans la documentation. Slurm

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

**Note**

CustomSlurmSettings est pris en charge à partir de AWS ParallelCluster la version 3.6.0.

### **Tags**(Facultatif, [Chaîne])

Une liste de paires clé-valeur de balises. [ComputeResource](#) les balises remplacent les balises dupliquées spécifiées dans le [Tags Section](#) ou dans `SlurmQueues/Tags`.

#### **Key**(Facultatif, **String**)

Identification de balise.

#### **Value**(Facultatif, **String**)

Valeur de balise.

Politique de mise à jour : le parc informatique doit être arrêté ou `QueueUpdateStrategy` défini pour que ce paramètre soit modifié en vue d'une mise à jour.

### **HealthChecks**(Facultatif)

Spécifiez les contrôles de santé des nœuds de calcul sur toutes les ressources informatiques de la file d'attente.

#### **Gpu**(Facultatif)

Spécifiez les contrôles de santé du GPU sur toutes les ressources de calcul d'une file d'attente.

**Note**

AWS ParallelCluster ne prend pas en charge `HealthChecks/Gpu` dans les nœuds utilisant les systèmes d'exploitation `linux2 ARM`. Ces plateformes ne sont pas compatibles avec le [NVIDIA Data Center GPU Manager \(DCGM\)](#).

## Enabled(Facultatif, Boolean)

AWS ParallelCluster Effectue des vérifications de l'état du GPU sur les nœuds de calcul. L'argument par défaut est `false`.

### Gpucomportement en matière de bilan de santé

- Si `SiGpu/Enabled` est défini sur `true`, AWS ParallelCluster effectue des contrôles de santé du GPU sur les ressources de calcul de la file d'attente.
- Le bilan Gpu de santé effectue des contrôles de l'état du GPU sur les ressources de calcul afin d'empêcher la soumission de tâches sur les nœuds dont le GPU est dégradé.
- Si un nœud de calcul échoue à un contrôle de Gpu santé, l'état du nœud de calcul passe à DRAIN. Les nouvelles tâches ne démarrent pas sur ce nœud. Les tâches existantes s'exécutent jusqu'à leur fin. Une fois toutes les tâches en cours d'exécution terminées, le nœud de calcul se termine s'il s'agit d'un nœud dynamique, et il est remplacé s'il s'agit d'un nœud statique.
- La durée du bilan de Gpu santé dépend du type d'instance sélectionné, du nombre de GPU dans l'instance et du nombre de cibles de contrôle de Gpu santé (équivalent au nombre de cibles GPU de la tâche). Pour une instance dotée de 8 GPU, la durée typique est inférieure à 3 minutes.
- Si le bilan Gpu de santé s'exécute sur une instance qui n'est pas prise en charge, elle s'arrête et la tâche s'exécute sur le nœud de calcul. Par exemple, si une instance ne possède pas de GPU, ou si une instance possède un GPU, mais qu'il ne s'agit pas d'un GPU NVIDIA, le bilan de santé s'arrête et la tâche s'exécute sur le nœud de calcul. Seuls les GPU NVIDIA sont pris en charge.
- Le Gpu bilan de santé utilise l'`dcgmi` outil pour effectuer des contrôles de santé sur un nœud et suit les étapes suivantes :

Lorsque le Gpu bilan de santé commence dans un nœud :

1. Il détecte si les `nvidia-fabricmanager services nvidia-dcgm` et sont en cours d'exécution.
2. Si ces services ne fonctionnent pas, le bilan Gpu de santé les démarre.
3. Il détecte si le mode de persistance est activé.
4. Si le mode de persistance n'est pas activé, le Gpu bilan de santé l'active.

À la fin du bilan de santé, le bilan de Gpu santé rétablit ces services et ressources dans leur état initial.

- Si la tâche est attribuée à un ensemble spécifique de GPU de nœuds, le contrôle de Gpu santé ne s'exécute que sur cet ensemble spécifique. Dans le cas contraire, le bilan de Gpu santé s'exécute sur tous les GPU du nœud.
- Si un nœud de calcul reçoit au moins deux demandes de contrôle de Gpu santé en même temps, seul le premier contrôle de santé est exécuté et les autres sont ignorés. C'est également le cas pour les bilans de santé ciblant les GPU des nœuds. Vous pouvez consulter les fichiers journaux pour obtenir des informations supplémentaires concernant cette situation.
- Le journal de vérification de l'état d'un nœud de calcul spécifique est disponible dans le `/var/log/parallelcluster/slurm_health_check.log` fichier. Le fichier est disponible sur Amazon CloudWatch, dans le groupe de CloudWatch journaux du cluster, où vous pouvez trouver :
  - Détails sur l'action exécutée par le bilan de Gpu santé, y compris l'activation et la désactivation des services et le mode de persistance.
  - L'identifiant du GPU, l'ID de série et l'UUID.
  - Le résultat du bilan de santé.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

**Note**

HealthCheck est pris en charge à partir de AWS ParallelCluster la version 3.6.0.

## Networking

(Obligatoire) Définit la configuration réseau de la Slurm file d'attente.

### Networking:

#### SubnetIds:

- *string*

AssignPublicIp: *boolean*

#### SecurityGroups:

- *string*

#### AdditionalSecurityGroups:

- *string*

#### PlacementGroup:

Enabled: *boolean*

Id: *string*

```
Name: string
Proxy:
HttpProxyAddress: string
```

Politique de mise à jour : le parc informatique doit être arrêté ou `QueueUpdateStrategy` défini pour que ce paramètre soit modifié en vue d'une mise à jour.

## Propriétés de **Networking**

### **SubnetIds**(Obligatoire, **[String]**)

Les ID des sous-réseaux existants dans lesquels vous approvisionnez la Slurm file d'attente.

Si vous configurez les types d'instances dans [SlurmQueues/ComputeResources/InstanceType](#), vous ne pouvez définir qu'un seul sous-réseau.

Si vous configurez les types d'instances dans [SlurmQueuesComputeResources//Instances](#), vous pouvez définir un ou plusieurs sous-réseaux.

Si vous utilisez plusieurs sous-réseaux, tous les sous-réseaux définis pour une file d'attente doivent se trouver dans le même VPC, chaque sous-réseau se trouvant dans une zone de disponibilité (AZ) distincte.

Supposons, par exemple, que vous définissiez le sous-réseau 1 et le sous-réseau 2 pour votre file d'attente.

subnet-1 et je ne subnet-2 ne peuvent pas être les deux dans l'AZ-1.

subnet-1 peut être dans AZ-1 et subnet-2 peut être dans AZ-2.

Si vous configurez un seul type d'instance et que vous souhaitez utiliser plusieurs sous-réseaux, définissez votre type d'instance dans `Instances` plutôt que `InstanceType` dans.

Par exemple, définissez `ComputeResources/Instances/InstanceType=instance.type` au lieu de `ComputeResources/InstanceType=instance.type`.

#### Note

Elastic Fabric Adapter (EFA) n'est pas pris en charge dans différentes zones de disponibilité.

L'utilisation de plusieurs zones de disponibilité peut entraîner une augmentation de la latence du réseau de stockage et des coûts supplémentaires de transfert de données inter-AZ. Cela peut se produire, par exemple, lorsqu'une instance accède à un stockage de fichiers situé dans une autre zone de disponibilité. Pour plus d'informations, consultez la section [Transfert de données au sein d'un même appareil Région AWS](#).

Mises à jour du cluster pour passer de l'utilisation d'un seul sous-réseau à plusieurs sous-réseaux :

- Supposons que la définition de sous-réseau d'un cluster soit définie avec un seul sous-réseau et un système de fichiers FSx for Lustre AWS ParallelCluster géré. Dans ce cas, vous ne pouvez pas mettre à jour directement ce cluster avec une définition d'ID de sous-réseau mise à jour. Pour effectuer la mise à jour du cluster, vous devez d'abord remplacer le système de fichiers géré par un système de fichiers externe. Pour plus d'informations, consultez [Convertir le stockage AWS ParallelCluster géré en stockage externe](#).
- Supposons que la définition de sous-réseau d'un cluster soit définie avec un seul sous-réseau et un système de fichiers Amazon EFS externe si les cibles de montage EFS n'existent pas pour toutes les AZ des multiples sous-réseaux définis pour être ajoutés. Dans ce cas, vous ne pouvez pas mettre à jour directement ce cluster avec une définition d'ID de sous-réseau mise à jour. Pour mettre à jour le cluster ou pour créer un cluster, vous devez d'abord créer toutes les cibles de montage pour toutes les AZ des multiples sous-réseaux définis.

Les zones de disponibilité et les réservations de capacité du cluster sont définies dans [CapacityReservationResourceGroupArn](#):

- Vous ne pouvez pas créer de cluster s'il n'y a aucun chevauchement entre l'ensemble de types d'instances et de zones de disponibilité couverts par le groupe de ressources de réservation de capacité défini et l'ensemble de types d'instances et de zones de disponibilité définis pour la file d'attente.
- Vous pouvez créer un cluster s'il existe un chevauchement partiel entre l'ensemble de types d'instances et de zones de disponibilité couverts par le groupe de ressources de réservation de capacité défini et l'ensemble de types d'instances et de zones de disponibilité définis pour la file d'attente. AWS ParallelCluster envoie un message d'avertissement concernant le chevauchement partiel dans ce cas.
- Pour plus d'informations, consultez [Lancez des instances avec des réservations de capacité à la demande \(ODCR\)](#).



**Note**

Plusieurs zones de disponibilité ont été ajoutées dans la AWS ParallelCluster version 3.4.0.

**Warning**

Cet avertissement s'applique à toutes les AWS ParallelCluster versions 3.x.y antérieures à la version 3.3.1. AWS ParallelCluster la version 3.3.1 n'est pas affectée si ce paramètre est modifié.

Pour les AWS ParallelCluster 3 versions antérieures à la version 3.3.1 :

La modification de ce paramètre et la mise à jour d'un cluster créent un nouveau système de fichiers géré FSx for Lustre et suppriment le système de fichiers géré FSx for Lustre existant sans conserver les données existantes. Cela entraîne une perte de données.

Avant de continuer, assurez-vous de sauvegarder les données du système de fichiers FSx for Lustre existant si vous souhaitez conserver les données. Pour plus d'informations, consultez la section [Utilisation des sauvegardes](#) dans le guide de l'utilisateur de FSx for Lustre.

Si une nouvelle valeur de sous-réseau est ajoutée, [Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

Si une valeur de sous-réseau est supprimée, [Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

**AssignPublicIp(Facultatif,String)**

Crée ou attribue une adresse IP publique aux nœuds de la Slurm file d'attente. Les valeurs prises en charge sont `true` et `false`. Le sous-réseau que vous spécifiez détermine la valeur par défaut. Un sous-réseau doté d'adresses IP publiques attribue par défaut des adresses IP publiques.

Si vous définissez un type d'`hpc6idinstance p4d` ou un autre type d'instance doté de plusieurs interfaces réseau ou d'une carte d'interface réseau, vous devez définir [HeadNode/Networking](#) sur [ElasticIp](#) `true` pour fournir un accès public. AWS les adresses IP publiques ne peuvent être attribuées qu'aux instances lancées avec une seule interface réseau. Dans ce cas, nous vous recommandons d'utiliser une [passerelle NAT](#) pour fournir un accès public aux nœuds de

calcul du cluster. Dans ce cas, réglez `AssignPublicIp` sur `false`. Pour plus d'informations sur les adresses IP, consultez la section [Attribuer une adresse IPv4 publique lors du lancement de l'instance](#) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

### **SecurityGroups**(Facultatif, **[String]**)

Liste des groupes de sécurité à utiliser pour la Slurm file d'attente. Si aucun groupe de sécurité n'est spécifié, AWS ParallelCluster crée des groupes de sécurité pour vous.

Vérifiez que les groupes de sécurité sont correctement configurés pour vos [SharedStorage](#) systèmes.

#### Warning

Cet avertissement s'applique aux 3. x. Y AWS ParallelCluster versions antérieures à la version 3.3.0. AWS ParallelCluster la version 3.3.0 n'est pas affectée si ce paramètre est modifié.

Pour les AWS ParallelCluster 3 versions antérieures à la version 3.3.0 :

La modification de ce paramètre et la mise à jour d'un cluster créent un nouveau système de fichiers géré FSx for Lustre et suppriment le système de fichiers géré FSx for Lustre existant sans conserver les données existantes. Cela entraîne une perte de données.

Assurez-vous de sauvegarder les données du système de fichiers FSx for Lustre existant si vous souhaitez conserver les données. Pour plus d'informations, consultez la section [Utilisation des sauvegardes](#) dans le guide de l'utilisateur de FSx for Lustre.

#### Warning

Si vous activez [Efa](#) pour vos instances de calcul, assurez-vous que celles-ci sont membres d'un groupe de sécurité qui autorise tout le trafic entrant et sortant à se diriger vers lui-même.

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

### **AdditionalSecurityGroups**(Facultatif, **[String]**)

Liste des groupes de sécurité supplémentaires à utiliser pour la Slurm file d'attente.

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

## PlacementGroup(Facultatif)

Spécifie les paramètres du groupe de placement pour la Slurm file d'attente.

### PlacementGroup:

Enabled: *boolean*

Id: *string*

Name: *string*

Politique de mise à jour : tous les nœuds de calcul doivent être arrêtés pour la suppression d'un groupe de placement géré. Le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.

## Enabled(Facultatif,Boolean)

Indique si un groupe de placement est utilisé pour la Slurm file d'attente. L'argument par défaut est false.

Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.

## Id(Facultatif,String)

Nom du groupe de placement d'un groupe de placement de clusters existant utilisé par la Slurm file d'attente. Assurez-vous de fournir le nom du groupe de placement et non l'ID.

Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.

## Name(Facultatif,String)

Nom du groupe de placement d'un groupe de placement de clusters existant utilisé par la Slurm file d'attente. Assurez-vous de fournir le nom du groupe de placement et non l'ID.

Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.

### Note

- SiPlacementGroup/Enabledest défini surtrue, sans Name ou Id défini, chaque ressource de calcul se voit attribuer son propre groupe de placement géré, sauf si

[ComputeResourcesNetworking//PlacementGroup](#) est défini pour remplacer ce paramètre.

- À partir de AWS ParallelCluster la version 3.3.0, [SlurmQueuesNetworking/PlacementGroup//Name](#) a été ajouté comme alternative préférée à [SlurmQueues//NetworkingPlacementGroup/Id](#).

[PlacementGroup/Id](#) et [PlacementGroup/Name](#) sont équivalents. Vous pouvez utiliser l'un ou l'autre.

Si vous incluez à la fois [PlacementGroup/Id](#) et [PlacementGroup/Name](#), cela AWS ParallelCluster échoue. Vous ne pouvez choisir que l'un ou l'autre.

Vous n'avez pas besoin de mettre à jour votre cluster pour utiliser [PlacementGroup/Name](#).

## Proxy(Facultatif)

Spécifie les paramètres du proxy pour la Slurm file d'attente.

```
Proxy:
HttpProxyAddress: string
```

Politique de mise à jour : le parc informatique doit être arrêté ou [QueueUpdateStrategy](#) défini pour que ce paramètre soit modifié en vue d'une mise à jour.

### **HttpProxyAddress**(Facultatif,**String**)

Définit un serveur proxy HTTP ou HTTPS pour la Slurm file d'attente. Généralement, c'est le `cashttps://x.x.x.x:8080`.

Il n'existe aucune valeur par défaut.

Politique de mise à jour : le parc informatique doit être arrêté ou [QueueUpdateStrategy](#) défini pour que ce paramètre soit modifié en vue d'une mise à jour.

## Image

(Facultatif) Spécifie l'image à utiliser pour la Slurm file d'attente. Pour utiliser la même AMI pour tous les nœuds, utilisez le [CustomAmi](#) paramètre de la [Image](#) section.

Image:

CustomAmi: *string*

Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.

## ImagePropriétés

### CustomAmi (Facultatif, String)

L'AMI à utiliser pour la Slurm file d'attente au lieu des AMI par défaut. Vous pouvez utiliser la commande pcluster CLI pour afficher la liste des AMI par défaut.

#### Note

L'AMI doit être basée sur le même système d'exploitation que celui utilisé par le nœud principal.

```
pcluster list-official-images
```

Si l'AMI personnalisée nécessite des autorisations supplémentaires pour son lancement, vous devez ajouter ces autorisations à la politique du nœud principal.

Par exemple, si un instantané chiffré est associé à une AMI personnalisée, les politiques supplémentaires suivantes sont requises dans les politiques du nœud principal.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "kms:DescribeKey",
 "kms:ReEncrypt*",
 "kms:CreateGrant",
 "kms:Decrypt"
],
 "Resource": [
 "arn:aws:kms:<AWS_REGION>:<AWS_ACCOUNT_ID>:key/<AWS_KMS_KEY_ID>"
]
 }
]
}
```

```

]
 }
]
}

```

Pour résoudre les problèmes liés aux avertissements de validation d'AMI personnalisés, consultez [Résolution des problèmes liés aux AMI personnalisées](#).

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

## ComputeResources

(Obligatoire) Définit la ComputeResources configuration de la Slurm file d'attente.

### Note

La taille du cluster peut changer lors d'une mise à jour. Pour plus d'informations, voir [Taille et mise à jour de la capacité du cluster](#)

### ComputeResources:

- Name: *string*
- InstanceType: *string*
- Instances:
  - InstanceType: *string*
- MinCount: *integer*
- MaxCount: *integer*
- DynamicNodePriority: *integer*
- StaticNodePriority: *integer*
- SpotPrice: *float*
- DisableSimultaneousMultithreading: *boolean*
- SchedulableMemory: *integer*
- HealthChecks:
  - Gpu:
    - Enabled: *boolean*
- Efa:
  - Enabled: *boolean*
  - GdrSupport: *boolean*
- CapacityReservationTarget:
  - CapacityReservationId: *string*

```
CapacityReservationResourceGroupArn: string
Networking:
 PlacementGroup:
 Enabled: boolean
 Name: string
CustomSlurmSettings: dict
Tags:
 - Key: string
 Value: string
```

Politique de mise à jour : pour ce paramètre de valeurs de liste, une nouvelle valeur peut être ajoutée lors d'une mise à jour ou le parc informatique doit être arrêté lors de la suppression d'une valeur existante.

## Propriétés de **ComputeResources**

### **Name**(Obligatoire,**String**)

Nom de l'environnement de calcul de la Slurm file d'attente. Le nom peut comporter jusqu'à 25 caractères.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

### **InstanceType**(Obligatoire,**String**)

Type d'instance utilisé dans cette ressource Slurm de calcul. Tous les types d'instances d'un cluster doivent utiliser la même architecture de processeur. Les instances peuvent utiliser l'arm64architecture x86\_64 ou.

La configuration du cluster doit définir soit les instances, [InstanceTypes](#) soit [les instances](#). Si les deux sont définis, AWS ParallelCluster échoue.

Lorsque vous définissez `InstanceType`, vous ne pouvez pas définir plusieurs sous-réseaux. Si vous configurez un seul type d'instance et que vous souhaitez utiliser plusieurs sous-réseaux, définissez votre type d'instance dans `Instances` plutôt que dans `InstanceType`. Pour plus d'informations, consultez [Networking/SubnetIds](#).

Si vous définissez un type d'instance `hpc6idinstance p4d` ou un autre type d'instance doté de plusieurs interfaces réseau ou d'une carte d'interface réseau, vous devez lancer les instances de calcul dans un sous-réseau privé comme décrit dans [AWS ParallelCluster en utilisant deux sous-réseaux](#). AWS les adresses IP publiques ne peuvent être attribuées qu'aux instances lancées avec une seule interface réseau. Pour plus d'informations, consultez la section [Attribuer une](#)

[adresse IPv4 publique lors du lancement de l'instance](#) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

## Instances(Obligatoire)

Spécifie la liste des types d'instances pour une ressource de calcul. Pour spécifier la stratégie d'allocation pour la liste des types d'instances, consultez [AllocationStrategy](#).

La configuration du cluster doit définir l'un [InstanceType](#) ou l'autre [Instances](#). Si les deux sont définis, AWS ParallelCluster échoue.

Pour plus d'informations, consultez [Allocation de plusieurs types d'instances avec Slurm](#).

### [Instances](#):

- [InstanceType](#): *string*

### Note

À partir de AWS ParallelCluster la version 3.7.0, `EnableMemoryBasedScheduling` peut être activée si vous configurez plusieurs types d'[instances dans Instances](#).

Pour AWS ParallelCluster les versions 3.2.0 à 3.6. *x*, ne

`EnableMemoryBasedScheduling` peut pas être activé si vous configurez plusieurs types d'[instances dans Instances](#).

[Politique de mise à jour : pour ce paramètre de valeurs de liste, une nouvelle valeur peut être ajoutée lors d'une mise à jour ou le parc informatique doit être arrêté lors de la suppression d'une valeur existante.](#)

## InstanceType(Obligatoire,String)

Type d'instance à utiliser dans cette ressource Slurm de calcul. Tous les types d'instances d'un cluster doivent utiliser la même architecture de processeur, x86\_64 soit arm64.

Les types d'instances répertoriés dans [Instances](#) doivent comporter les éléments suivants :

- Le même nombre de vCPU ou, s'il [DisableSimultaneousMultithreading](#) est défini sur `true`, le même nombre de cœurs.



- Le même nombre d'accélérateurs des mêmes fabricants.
- EFA est pris en charge, si `Efa/Enabled` est défini sur `true`.

Les types d'instances répertoriés dans [Instances](#) peuvent être les suivants :

- Quantité de mémoire différente.

Dans ce cas, la mémoire minimale doit être définie comme Slurm ressource consommable.

**Note**

À partir de AWS ParallelCluster la version 3.7.0, elle `EnableMemoryBasedScheduling` peut être activée si vous configurez plusieurs types d'[instances dans Instances](#).

Pour AWS ParallelCluster les versions 3.2.0 à 3.6. x, ne `EnableMemoryBasedScheduling` peut pas être activé si vous configurez plusieurs types d'[instances dans Instances](#).

- Différentes cartes réseau.

Dans ce cas, le nombre d'interfaces réseau configurées pour la ressource de calcul est défini par le type d'instance possédant le plus petit nombre de cartes réseau.

- Bande passante réseau différente.
- Taille de magasin d'instance différente.

Si vous définissez un type d'`hpc6idinstance p4d` ou un autre type d'instance doté de plusieurs interfaces réseau ou d'une carte d'interface réseau, vous devez lancer les instances de calcul dans un sous-réseau privé comme décrit dans [AWS ParallelCluster en utilisant deux sous-réseaux](#). AWS les adresses IP publiques ne peuvent être attribuées qu'aux instances lancées avec une seule interface réseau. Pour plus d'informations, consultez la section [Attribuer une adresse IPv4 publique lors du lancement de l'instance](#) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

**Note**

`Instances` est pris en charge à partir de AWS ParallelCluster la version 3.3.0.

## **MinCount**(Facultatif,**Integer**)

Le nombre minimal d'instances utilisées par la ressource de Slurm calcul. La valeur par défaut est 0.

### Note

La taille du cluster peut changer lors d'une mise à jour. Pour plus d'informations, voir [Taille et mise à jour de la capacité du cluster](#)

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

## **MaxCount**(Facultatif,**Integer**)

Nombre maximal d'instances utilisées par la ressource de Slurm calcul. La valeur par défaut est 10.

Lors de l'utilisation `CapacityType = CAPACITY_BLOCK`, `MaxCount` il doit être égal `MinCount` ou supérieur à 0, car toutes les instances faisant partie de la réservation du bloc de capacité sont gérées en tant que nœuds statiques.

Au moment de la création du cluster, le nœud principal attend que tous les nœuds statiques soient prêts avant de signaler le succès de la création du cluster. Toutefois, lors de l'utilisation `CapacityType = CAPACITY_BLOCK`, les nœuds faisant partie des ressources de calcul associées aux blocs de capacité ne seront pas pris en compte pour cette vérification. Le cluster sera créé même si tous les blocs de capacité configurés ne sont pas actifs.


### Note

La taille du cluster peut changer lors d'une mise à jour. Pour plus d'informations, voir [Taille et mise à jour de la capacité du cluster](#)


## **DynamicNodePriority**(Facultatif,**Integer**)

Priorité des nœuds dynamiques dans une ressource de calcul de file d'attente. La priorité correspond au paramètre de [Weight](#) configuration des Slurm nœuds pour les nœuds dynamiques des ressources informatiques. La valeur par défaut est 1000.

Slurmdonne la priorité aux nœuds ayant les `Weight` valeurs les plus faibles en premier.

 Warning

L'utilisation de nombreuses `Weight` valeurs différentes dans une Slurm partition (file d'attente) peut ralentir le rythme de planification des tâches dans la file d'attente. Dans AWS ParallelCluster les versions antérieures à la version 3.7.0, les nœuds statiques et dynamiques se voyaient attribuer le même poids par défaut de 1. Dans ce cas, Slurm peut donner la priorité aux nœuds dynamiques inactifs par rapport aux nœuds statiques inactifs en raison du schéma de dénomination des nœuds statiques et dynamiques. Lorsque toutes les choses sont égales par ailleurs, Slurm planifie les nœuds par ordre alphabétique de leur nom.

 Note


`DynamicNodePriority` est ajouté dans la AWS ParallelCluster version 3.7.0.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

**`StaticNodePriority`**(Facultatif, **Integer**)

Priorité des nœuds statiques dans une ressource de calcul de file d'attente. La priorité correspond au paramètre de [Weight](#) configuration des Slurm nœuds pour les nœuds statiques des ressources de calcul. La valeur par défaut est 1.

Slurmdonne la priorité aux nœuds ayant les `Weight` valeurs les plus faibles en premier.

 Warning

L'utilisation de nombreuses `Weight` valeurs différentes dans une Slurm partition (file d'attente) peut ralentir le rythme de planification des tâches dans la file d'attente.

 Note

`StaticNodePriority` est ajouté dans la AWS ParallelCluster version 3.7.0.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

### **SpotPrice**(Facultatif,**Float**)

Le prix maximum payé pour une instance Amazon EC2 Spot avant le lancement de toute instance. La valeur par défaut est le prix à la demande.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

### **DisableSimultaneousMultithreading**(Facultatif,**Boolean**)

Si `true` le multithreading sur les nœuds de la Slurm file d'attente est désactivé. La valeur par défaut est `false`.

Tous les types d'instances ne peuvent pas désactiver le multithreading. Pour obtenir la liste des types d'instances qui prennent en charge la désactivation du multithreading, consultez la section relative aux [cœurs de processeur et aux threads de chaque cœur de processeur par type d'instance](#) dans le guide de l'utilisateur Amazon EC2.

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

### **SchedulableMemory**(Facultatif,**Integer**)

Quantité de mémoire en MiB configurée dans le Slurm paramètre `RealMemory` pour les nœuds de calcul d'une ressource de calcul. Cette valeur est la limite supérieure de la mémoire de nœuds disponible pour les tâches lorsque [SlurmSettings/EnableMemoryBasedScheduling](#) est activé. La valeur par défaut est de 95 % de la mémoire répertoriée dans les [types d'instances Amazon EC2](#) et renvoyée par l'API Amazon EC2. [DescribeInstanceTypes](#) Assurez-vous de convertir les valeurs données en GiB en MiB.

Valeurs prises en charge : 1-`EC2Memory`

`EC2Memory` est la mémoire (en MiB) répertoriée dans les types d'[instances Amazon EC2](#) et renvoyée par l'API Amazon EC2. [DescribeInstanceTypes](#) Assurez-vous de convertir les valeurs données en GiB en MiB.

Cette option est particulièrement pertinente lorsque [SlurmSettings/EnableMemoryBasedScheduling](#) est activé. Pour plus d'informations, consultez [Slurmplanification basée sur la mémoire](#).

**Note**

`SchedulableMemory` est pris en charge à partir de AWS ParallelCluster la version 3.2.0. À partir de la version 3.2.0, 95 % AWS ParallelCluster de la `RealMemory` mémoire renvoyée par l'API Amazon EC2 est configurée par défaut pour les nœuds de Slurm calcul. `DescribeInstanceTypes` Cette configuration est indépendante de la valeur de `EnableMemoryBasedScheduling`.

[Politique de mise à jour : le parc informatique doit être arrêté ou `QueueUpdateStrategy` défini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

**HealthChecks**(Facultatif)

Spécifiez les contrôles de santé d'une ressource de calcul.

**Gpu**(Facultatif)

Spécifiez les contrôles de santé du GPU sur une ressource de calcul.

**Enabled**(Facultatif, **Boolean**)

AWS ParallelCluster Effectue des contrôles de santé du GPU lors du calcul d'une ressource dans une file d'attente. L'argument par défaut est `false`.

**Note**

AWS ParallelCluster ne prend pas en charge `HealthChecks/Gpu` dans les nœuds utilisant les systèmes d'exploitation `alinux2 ARM`. Ces plateformes ne sont pas compatibles avec le [NVIDIA Data Center GPU Manager \(DCGM\)](#).

**Gpu**comportement en matière de bilan de santé

- Si `Gpu/Enabled` est défini sur `true`, AWS ParallelCluster effectue des contrôles de santé du GPU sur une ressource de calcul.
- Le `Gpu` bilan de santé effectue des contrôles de santé sur une ressource de calcul afin d'empêcher la soumission de tâches sur des nœuds dont le GPU est dégradé.
- Si un nœud de calcul échoue à un contrôle de `Gpu` santé, l'état du nœud de calcul passe à `DRAIN`. Les nouvelles tâches ne démarrent pas sur ce nœud. Les tâches existantes s'exécutent jusqu'à leur fin. Une fois toutes les tâches en cours d'exécution terminées, le

nœud de calcul se termine s'il s'agit d'un nœud dynamique, et il est remplacé s'il s'agit d'un nœud statique.

- La durée du bilan de Gpu santé dépend du type d'instance sélectionné, du nombre de GPU dans l'instance et du nombre de cibles de contrôle de Gpu santé (équivalent au nombre de cibles GPU de la tâche). Pour une instance dotée de 8 GPU, la durée typique est inférieure à 3 minutes.
- Si le bilan Gpu de santé s'exécute sur une instance qui n'est pas prise en charge, elle s'arrête et la tâche s'exécute sur le nœud de calcul. Par exemple, si une instance ne possède pas de GPU, ou si une instance possède un GPU, mais qu'il ne s'agit pas d'un GPU NVIDIA, le bilan de santé s'arrête et la tâche s'exécute sur le nœud de calcul. Seuls les GPU NVIDIA sont pris en charge.
- Le Gpu bilan de santé utilise l'`dcgmi`outil pour effectuer des contrôles de santé sur un nœud et suit les étapes suivantes :

Lorsque le Gpu bilan de santé commence dans un nœud :

1. Il détecte si les `nvidia-fabricmanager` services `nvidia-dcgm` et sont en cours d'exécution.
2. Si ces services ne fonctionnent pas, le bilan Gpu de santé les démarre.
3. Il détecte si le mode de persistance est activé.
4. Si le mode de persistance n'est pas activé, le Gpu bilan de santé l'active.

À la fin du bilan de santé, le bilan de Gpu santé rétablit ces services et ressources dans leur état initial.

- Si la tâche est attribuée à un ensemble spécifique de GPU de nœuds, le contrôle de Gpu santé ne s'exécute que sur cet ensemble spécifique. Dans le cas contraire, le bilan de Gpu santé s'exécute sur tous les GPU du nœud.
- Si un nœud de calcul reçoit au moins deux demandes de contrôle de Gpu santé en même temps, seul le premier contrôle de santé est exécuté et les autres sont ignorés. C'est également le cas pour les bilans de santé ciblant les GPU des nœuds. Vous pouvez consulter les fichiers journaux pour obtenir des informations supplémentaires concernant cette situation.
- Le journal de vérification de l'état d'un nœud de calcul spécifique est disponible dans le `/var/log/parallelcluster/slurm_health_check.log` fichier. Ce fichier est disponible sur Amazon CloudWatch, dans le groupe de CloudWatch journaux du cluster, où vous pouvez trouver :

- Détails sur l'action exécutée par le bilan de Gpu santé, y compris l'activation et la désactivation des services et le mode de persistance.
- L'identifiant du GPU, l'ID de série et l'UUID.
- Le résultat du bilan de santé.

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

**Note**

HealthCheck est pris en charge à partir de AWS ParallelCluster la version 3.6.0.

## Efa (Facultatif)

Spécifie les paramètres Elastic Fabric Adapter (EFA) pour les nœuds de la file d'attente. Slurm

Efa:

Enabled: *boolean*

GdrSupport: *boolean*

Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.

### Enabled (Facultatif, Boolean)

Spécifie qu'Elastic Fabric Adapter (EFA) est activé. Pour consulter la liste des instances Amazon EC2 qui prennent en charge l'EFA, consultez la section [Types d'instances pris en charge](#) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux. Pour plus d'informations, consultez [Elastic Fabric Adapter](#). Nous vous recommandons d'utiliser un cluster [SlurmQueues/Networking/PlacementGroup](#) pour minimiser les latences entre les instances.

La valeur par défaut est `false`.

**Note**

Elastic Fabric Adapter (EFA) n'est pas pris en charge dans différentes zones de disponibilité. Pour plus d'informations, consultez [SubnetIds](#).

**⚠ Warning**

Si vous définissez un groupe de sécurité personnalisé dans [SecurityGroups](#), assurez-vous que vos instances compatibles avec EFA sont membres d'un groupe de sécurité qui autorise l'intégralité du trafic entrant et sortant à destination de lui-même.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini](#) pour que ce paramètre soit modifié en vue d'une mise à jour.

**GdrSupport**(Facultatif,**Boolean**)

(Facultatif) À partir de AWS ParallelCluster la version 3.0.2, ce paramètre n'a aucun effet. La prise en charge de GPUDirect RDMA (Remote Direct Memory Access) par Elastic Fabric Adapter (EFA) est toujours activée si elle est prise en charge par le type d'instance de la ressource de calcul et Slurm du système d'exploitation.

**i Note**

AWS ParallelCluster versions 3.0.0 à 3.0.1 : le support de GPUDirect RDMA est activé pour les ressources de calcul. Slurm Support pour GPUDirect RDMA est pris en charge par des types d'instances spécifiques (p4d.24xlarge) sur des systèmes d'exploitation spécifiques ([Osis](#) a linux2 centos7ubuntu1804, ou). ubuntu2004  
La valeur par défaut est false.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini](#) pour que ce paramètre soit modifié en vue d'une mise à jour.

**CapacityReservationTarget**

[CapacityReservationTarget](#):

[CapacityReservationId](#): *string*

[CapacityReservationResourceGroupArn](#): *string*

Spécifie la réservation de capacité à la demande à utiliser pour la ressource de calcul.

**CapacityReservationId**(Facultatif,**String**)

L'ID de la réservation de capacité existante à cibler pour les ressources de calcul de la file d'attente. L'identifiant peut faire référence à un [ODCR](#) ou à un [bloc de capacité pour ML](#).



Lorsque ce paramètre est spécifié au niveau de la ressource de calcul, s'il InstanceType est facultatif, il sera automatiquement extrait de la réservation.

### CapacityReservationResourceGroupArn(Facultatif,String)

Indique le nom de ressource Amazon (ARN) du groupe de ressources qui sert de groupe de réservations de capacité lié au service pour la ressource de calcul. AWS ParallelCluster identifie et utilise la réservation de capacité la plus appropriée auprès du groupe. Le groupe de ressources doit disposer d'au moins un ODCR pour chaque type d'instance répertorié pour la ressource de calcul. Pour plus d'informations, consultez [Lancez des instances avec des réservations de capacité à la demande \(ODCR\)](#).

- Si cette option PlacementGroup est activée dans [SlurmQueuesSlurmQueues/Networking](#) ou [ComputeResources//Networking](#), AWS ParallelCluster sélectionne un groupe de ressources qui cible le type PlacementGroup d'instance et une ressource de calcul si elle existe.

Ils PlacementGroup doivent cibler l'un des types d'instances définis dans [ComputeResources](#).

- S'il PlacementGroup n'est pas activé dans [SlurmQueues/Networking](#) ou [SlurmQueues/ComputeResources/Networking](#), AWS ParallelCluster sélectionne un groupe de ressources qui cible uniquement le type d'instance d'une ressource de calcul, s'il existe.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

#### Note

CapacityReservationTargetest ajouté avec la AWS ParallelCluster version 3.3.0.

## Networking

[Networking](#):

[PlacementGroup](#):

[Enabled](#): *boolean*

[Name](#): *string*

Politique de mise à jour : tous les nœuds de calcul doivent être arrêtés pour la suppression d'un groupe de placement géré. Le parc informatique doit être arrêté ou `QueueUpdateStrategy` défini pour que ce paramètre soit modifié en vue d'une mise à jour.

### **PlacementGroup**(Facultatif)

Spécifie les paramètres du groupe de placement pour la ressource de calcul.

#### **Enabled**(Facultatif, **Boolean**)

Indique si un groupe de placement est utilisé pour la ressource de calcul.

- Si elle est définie sur `true`, sans définition, cette ressource de calcul se voit attribuer son propre groupe de placement géré, quel que soit le [PlacementGroup](#) paramètre `SlurmQueuesNetworking`.
- Si elle est définie sur `true`, avec une valeur `Name` définie, cette ressource de calcul se voit attribuer le groupe de placement nommé, quels que soient `PlacementGroup` les paramètres `SlurmQueuesNetworking`.

Politique de mise à jour : le parc informatique doit être arrêté ou `QueueUpdateStrategy` défini pour que ce paramètre soit modifié en vue d'une mise à jour.

#### **Name**(Facultatif, **String**)

Le nom du groupe de placement d'un groupe de placement de clusters existant utilisé pour la ressource de calcul.

Politique de mise à jour : le parc informatique doit être arrêté ou `QueueUpdateStrategy` défini pour que ce paramètre soit modifié en vue d'une mise à jour.

#### Note

- Si les paramètres `PlacementGroup/Enabled` et `Name` non sont définis, leurs valeurs respectives sont par défaut les [PlacementGroup](#) paramètres `SlurmQueuesNetworking`.
- `ComputeResources/Networking/PlacementGroup` est ajouté avec AWS ParallelCluster la version 3.3.0.

## CustomSlurmSettings(Facultatif,Dict)

(Facultatif) Définit les paramètres de configuration personnalisés du Slurm nœud (ressource de calcul).

Spécifie un dictionnaire de paires clé-valeur de paramètres de Slurm configuration personnalisés qui s'appliquent aux Slurm nœuds (ressources de calcul).

Chaque paire clé-valeur distincte, telle que `Param1: Value1`, est ajoutée séparément à la fin de la ligne de configuration du Slurm nœud dans le format. `Param1=Value1`

Vous ne pouvez spécifier que des paramètres Slurm de configuration qui ne figurent pas dans la liste refusée. CustomSlurmSettings Pour plus d'informations sur les paramètres de Slurm configuration refusés, consultez. [Listé par DenySlurmparamètres de configuration pourCustomSlurmSettings](#)

AWS ParallelCluster vérifie uniquement si un paramètre figure dans une liste de refus. AWS ParallelCluster ne valide pas la syntaxe ou la sémantique de vos paramètres de Slurm configuration personnalisés. Vous êtes responsable de la validation de vos paramètres de Slurm configuration personnalisés. Des paramètres Slurm de configuration personnalisés non valides peuvent provoquer des défaillances de Slurm daemon susceptibles d'entraîner des échecs de création et de mise à jour de clusters.

Pour plus d'informations sur la façon de spécifier des paramètres Slurm de configuration personnalisés avec AWS ParallelCluster, consultez [Slurmpersonnalisation de la configuration](#).

Pour plus d'informations sur les paramètres Slurm de configuration, consultez [slurm.conf](#) dans la documentation. Slurm

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

### Note

CustomSlurmSettings est pris en charge à partir de AWS ParallelCluster la version 3.6.0.

## Tags(Facultatif, [Chaîne])

Une liste de paires clé-valeur de balises. ComputeResourceles balises remplacent les balises dupliquées spécifiées dans le [Tags Section](#) ou [SlurmQueues/Tags](#).

**Key**(Facultatif,**String**)

Identification de balise.

**Value**(Facultatif,**String**)

Valeur de balise.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

**ComputeSettings**

(Obligatoire) Définit la ComputeSettings configuration de la Slurm file d'attente.

Propriétés de **ComputeSettings**

Spécifie les propriétés ComputeSettings des nœuds de la Slurm file d'attente.

```
ComputeSettings:
 LocalStorage:
 RootVolume:
 Size: integer
 Encrypted: boolean
 VolumeType: string
 Iops: integer
 Throughput: integer
 EphemeralVolume:
 MountDir: string
```

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

**LocalStorage**(Facultatif)

Spécifie les propriétés LocalStorage des nœuds de la Slurm file d'attente.

```
LocalStorage:
 RootVolume:
 Size: integer
 Encrypted: boolean
```

```
VolumeType: string
Iops: integer
Throughput: integer
EphemeralVolume:
MountDir: string
```

Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.

### **RootVolume**(Facultatif)

Spécifie les détails du volume racine des nœuds de la Slurm file d'attente.

```
RootVolume:
Size: integer
Encrypted: boolean
VolumeType: string
Iops: integer
Throughput: integer
```

Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.

### **Size**(Facultatif, **Integer**)

Spécifie la taille du volume racine en gibioctets (GiB) pour les nœuds de la file d'attente. Slurm La taille par défaut provient de l'AMI. L'utilisation d'une taille différente nécessite que l'AMI soit compatible `growroot`.

Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.

### **Encrypted**(Facultatif, **Boolean**)

Si `true`, le volume racine des nœuds de la Slurm file d'attente est chiffré. La valeur par défaut est `false`.

Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.

## **VolumeType**(Facultatif,**String**)

Spécifie le [type de volume Amazon EBS](#) des nœuds de la Slurm file d'attente. Les valeurs prises en charge sont gp2 gp3io1,io2,sc1,st1, etstandard. La valeur par défaut est gp3.

Pour plus d'informations, consultez [Types de volumes Amazon EBS](#) dans le Guide de l'utilisateur Amazon EC2.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

## **Iops**(Facultatif,**Boolean**)

Définit le nombre d'IOPS pour io1io2, et le gp3 type de volumes.

La valeur par défaut, les valeurs prises en charge et volume\_iops le volume\_size ratio de production varient selon VolumeType etSize.

### **VolumeType = io1**

Valeur par défaut Iops = 100

Valeurs prises en charge Iops = 100—64 000 †

volume\_sizeRapport volume\_iops maximum = 50 IOPS par GiB. 5000 IOPS nécessitent au volume\_size moins 100 GiB.

### **VolumeType = io2**

Valeur par défaut Iops = 100

Valeurs prises en charge Iops = 100 à 64 000 (256 000 pour les volumes io2 Block Express) †

SizeRapport Iops maximum = 500 IOPS par GiB. 5000 IOPS nécessitent au Size moins 10 GiB.

### **VolumeType = gp3**

Valeur par défaut Iops = 3000

Valeurs prises en charge Iops = 3000 à 16 000 †

SizeRapport maximum Iops = 500 IOPS par GiB pour les volumes dont le nombre d'IOPS est supérieur à 3 000.

† Le maximum d'IOPS est garanti uniquement sur les [instances créées sur le système Nitro](#) qui sont également provisionnées avec plus de 32 000 IOPS. Les autres instances peuvent avoir jusqu'à 32 000 IOPS. `io1` Les volumes antérieurs risquent de ne pas atteindre leurs performances optimales à moins que vous ne [modifiez le volume](#). `io2` Les volumes Block Express prennent en charge `volume_iops` des valeurs allant jusqu'à 256 000 pour les types d'`R5b` instances. Pour plus d'informations, consultez la section sur les [volumes `io2` Block Express](#) dans le guide de l'utilisateur Amazon EC2.

[Politique de mise à jour : le parc informatique doit être arrêté ou `QueueUpdateStrategy` défini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

### **Throughput**(Facultatif,**Integer**)

Définit le débit pour les types de `gp3` volumes, en Mbits/s. Ce paramètre n'est valide que lorsqu'il l'`VolumeType` est `gp3`. La valeur par défaut est 125. Valeurs prises en charge : 125-1000 MiB/s

Le rapport de Throughput à ne Iops peut pas être supérieur à 0,25. Le débit maximal de 1 000 Mbits/s nécessite que le Iops paramètre soit d'au moins 4 000.

[Politique de mise à jour : le parc informatique doit être arrêté ou `QueueUpdateStrategy` défini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

### **EphemeralVolume**(Facultatif,**Boolean**)

Spécifie les paramètres du volume éphémère. Le volume éphémère est créé en combinant tous les volumes de stockage d'instance en un seul volume logique formaté avec le `ext4` système de fichiers. L'argument par défaut est `/scratch`. Si le type d'instance ne possède aucun volume de stockage d'instance, aucun volume éphémère n'est créé. Pour plus d'informations, consultez [Volumes de stockage d'instances](#) dans le Guide de l'utilisateur Amazon EC2.

[EphemeralVolume](#):  
[MountDir](#): *string*

Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.

### **MountDir**(Facultatif, **String**)

Le répertoire de montage du volume éphémère pour chaque nœud de la Slurm file d'attente.

Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.

### **CustomActions**

(Facultatif) Spécifie les scripts personnalisés à exécuter sur les nœuds de la Slurm file d'attente.

```
CustomActions:
 OnNodeStart:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
 OnNodeConfigured:
 Sequence:
 - Script: string
 Args:
 - string
 Script: string
 Args:
 - string
```

Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.



## CustomActionsPropriétés

### OnNodeStart(Facultatif,String)

Spécifie une séquence de scripts ou un script unique à exécuter sur les nœuds de la Slurm file d'attente avant le lancement de toute action d'amorçage du déploiement des nœuds. AWS ParallelCluster ne permet pas d'inclure à la fois un seul script et Sequence pour la même action personnalisée. Pour plus d'informations, consultez [Actions de bootstrap personnalisées](#).

### Sequence(Facultatif)

Liste des scripts à exécuter.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

### Script(Obligatoire,String)

Le fichier à utiliser. Le chemin du fichier peut commencer par `https://ous3://`.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

### Args(Facultatif,[String])

La liste des arguments à transmettre au script.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

### Script(Obligatoire,String)

Le fichier à utiliser pour un seul script. Le chemin du fichier peut commencer par `https://ous3://`.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

### Args(Facultatif,[String])

La liste des arguments à transmettre au script unique.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

### **OnNodeConfigured**(Facultatif,**String**)

Spécifie une séquence de scripts ou un script unique à exécuter sur les nœuds de la Slurm file d'attente une fois toutes les actions d'amorçage des nœuds terminées. AWS ParallelCluster ne permet pas d'inclure à la fois un seul script et Sequence pour la même action personnalisée. Pour plus d'informations, consultez [Actions de bootstrap personnalisées](#).

#### **Sequence**(Facultatif)

Liste des scripts à exécuter.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

#### **Script**(Obligatoire,**String**)

Le fichier à utiliser. Le chemin du fichier peut commencer par `https://` ou `s3://`.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

#### **Args**(Facultatif,**[String]**)

La liste des arguments à transmettre au script.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

#### **Script**(Obligatoire,**String**)

Le fichier à utiliser pour un seul script. Le chemin du fichier peut commencer par `https://` ou `s3://`.

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

## Args(Facultatif, [String])

Liste d'arguments à transmettre au script unique.

[Politique de mise à jour : le parc informatique doit être arrêté ou](#)

[QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategydéfini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

### Note

Sequenceest ajouté à partir de AWS ParallelCluster la version 3.6.0. Lorsque vous le spécifiezSequence, vous pouvez répertorier plusieurs scripts pour une action personnalisée. AWS ParallelCluster continue de prendre en charge la configuration d'une action personnalisée avec un seul script, sans inclureSequence.

AWS ParallelCluster ne permet pas d'inclure à la fois un seul script et Sequence pour la même action personnalisée.

## Iam

(Facultatif) Définit les paramètres IAM facultatifs pour la Slurm file d'attente.

### Iam:

#### S3Access:

- BucketName: *string*
- EnableWriteAccess: *boolean*
- KeyName: *string*

#### AdditionalIamPolicies:

- Policy: *string*

#### InstanceProfile: *string*

#### InstanceRole: *string*

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## IamPropriétés

## InstanceProfile(Facultatif,String)

Spécifie un profil d'instance pour remplacer le rôle d'instance ou le profil d'instance par défaut pour la Slurm file d'attente. Vous ne pouvez pas spécifier à la fois InstanceProfile

etInstanceRole. Le format est `arn:${Partition}:iam:${Account}:instance-profile/${InstanceProfileName}`.

Si cela est spécifié, les AdditionalIamPolicies paramètres S3Access et ne peuvent pas être spécifiés.

Nous vous recommandons de spécifier l'un des AdditionalIamPolicies paramètres S3Access et, ou les deux, car les fonctionnalités ajoutées nécessitent AWS ParallelCluster souvent de nouvelles autorisations.

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

### **InstanceRole**(Facultatif,**String**)

Spécifie un rôle d'instance pour remplacer le rôle d'instance ou le profil d'instance par défaut pour la Slurm file d'attente. Vous ne pouvez pas spécifier à la fois InstanceProfile etInstanceRole. Le format est `arn:${Partition}:iam:${Account}:role/${RoleName}`.

Si cela est spécifié, les AdditionalIamPolicies paramètres S3Access et ne peuvent pas être spécifiés.

Nous vous recommandons de spécifier l'un des AdditionalIamPolicies paramètres S3Access et, ou les deux, car les fonctionnalités ajoutées nécessitent AWS ParallelCluster souvent de nouvelles autorisations.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

### **S3Access**(Facultatif)

Spécifie un compartiment pour la Slurm file d'attente. Ceci est utilisé pour générer des politiques visant à accorder l'accès spécifié au compartiment dans la Slurm file d'attente.

Si cela est spécifié, les InstanceRole paramètres InstanceProfile et ne peuvent pas être spécifiés.

Nous vous recommandons de spécifier l'un des AdditionalIamPolicies paramètres S3Access et, ou les deux, car les fonctionnalités ajoutées nécessitent AWS ParallelCluster souvent de nouvelles autorisations.

[S3Access](#):

```
- BucketName: string
 EnableWriteAccess: boolean
 KeyName: string
```

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

### **BucketName**(Obligatoire,**String**)

Le nom du compartiment .

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

### **KeyName**(Facultatif,**String**)

La clé du seau. La valeur par défaut est \*.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

### **EnableWriteAccess**(Facultatif,**Boolean**)

Indique si l'accès en écriture est activé pour le compartiment.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

### **AdditionalIamPolicies**(Facultatif)

Spécifie une liste des Amazon Resource Names (ARN) des politiques IAM pour Amazon EC2. Cette liste est attachée au rôle root utilisé pour la Slurm file d'attente en plus des autorisations requises par AWS ParallelCluster.

Le nom d'une politique IAM et son ARN sont différents. Les noms ne peuvent pas être utilisés.

Si cela est spécifié, les InstanceRole paramètres InstanceProfile et ne peuvent pas être spécifiés.

Nous vous recommandons d'utiliser AdditionalIamPolicies car AdditionalIamPolicies ils sont ajoutés aux autorisations AWS ParallelCluster requises et InstanceRole doivent inclure toutes les autorisations requises. Les autorisations requises changent souvent d'une version à l'autre au fur et à mesure que des fonctions sont ajoutées.

Il n'existe aucune valeur par défaut.

[AdditionalIamPolicies](#):

- `Policy`: *string*

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

### **Policy**(Obligatoire,**[String]**)

Liste des politiques IAM.

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

## **SlurmSettings**

(Facultatif) Définit les paramètres Slurm qui s'appliquent à l'ensemble du cluster.

```
SlurmSettings:
 ScaledownIdleTime: integer
 QueueUpdateStrategy: string
 EnableMemoryBasedScheduling: boolean
 CustomSlurmSettings: [dict]
 CustomSlurmSettingsIncludeFile: string
 Database:
 Uri: string
 UserName: string
 PasswordSecretArn: string
 ExternalSlurmdbd:
 Host: string
 Port: integer
 Dns:
 DisableManagedDns: boolean
 HostedZoneId: string
 UseEc2Hostnames: boolean
```

## **SlurmSettingsPropriétés**

### **ScaledownIdleTime**(Facultatif,**Integer**)

Définit la durée (en minutes) pendant laquelle il n'y a aucune tâche et pendant laquelle le Slurm nœud se termine.

La valeur par défaut est 10.

Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.

## MungeKeySecretArn(Facultatif,String)

Nom de ressource Amazon (ARN) du secret AWS Secrets Manager en texte clair qui contient la clé munge codée en base64 à utiliser dans le cluster. Slurm Cette clé munge sera utilisée pour authentifier les appels RPC entre les commandes du Slurm client et les Slurm démons agissant en tant que serveurs distants. S'il n' MungeKeySecretArn est pas fourni, AWS ParallelCluster il générera une clé munge aléatoire pour le cluster.

### Note

MungeKeySecretArn est pris en charge à partir de AWS ParallelCluster la version 3.8.0.

### Warning

Si le MungeKeySecretArn est récemment ajouté à un cluster existant, la clé munge précédente ne ParallelCluster sera pas restaurée en cas de restauration ou lors de la suppression ultérieure de la. MungeKeySecretArn Au lieu de cela, une nouvelle clé de munge aléatoire sera générée.

Si l' AWS ParallelCluster utilisateur est autorisé à accéder [DescribeSecret](#) à cette ressource secrète spécifique, elle MungeKeySecretArn est validée. MungeKeySecretArn est valide si :

- Le secret spécifié existe, et
- Le secret est en texte brut et contient une chaîne codée en base64 valide, et
- La clé de munge binaire décodée a une taille comprise entre 256 et 8192 bits.

Si la politique IAM de l'utilisateur pcluster n'inclut pas DescribeSecret, n' MungeKeySecretArn est pas validée et un message d'avertissement s'affiche. Pour plus d'informations, consultez [Politique AWS ParallelCluster pcluster utilisateur de base](#).

Lorsque vous effectuez une mise à jour MungeKeySecretArn, le parc informatique et tous les nœuds de connexion doivent être arrêtés.

Si la valeur secrète de l'ARN secret est modifiée alors que l'ARN reste le même, le cluster ne sera pas automatiquement mis à jour avec la nouvelle clé munge. Pour utiliser la nouvelle clé munge de l'ARN secret, vous devez arrêter le parc informatique et connecter les nœuds, puis exécuter la commande suivante depuis le nœud principal.

```
sudo /opt/parallelcluster/scripts/slurm/update_munge_key.sh
```

Après avoir exécuté la commande, vous pouvez reprendre à la fois le parc de calcul et les nœuds de connexion : les nœuds de calcul et de connexion nouvellement provisionnés commenceront automatiquement à utiliser la nouvelle clé munge.

Pour générer une clé munge personnalisée codée en base64, vous pouvez utiliser l'[utilitaire mungekey distribué avec le logiciel munge](#), puis l'encoder à l'aide de l'utilitaire base64 généralement disponible sur votre système d'exploitation. Vous pouvez également utiliser bash (veuillez définir le paramètre bs entre 32 et 1024)

```
dd if=/dev/random bs=128 count=1 2>/dev/null | base64 -w 0
```

ou Python comme suit :

```
import random
import os
import base64

key length in bytes
key_length=128

base64.b64encode(os.urandom(key_length)).decode("utf-8")
```

Politique de mise à jour : NOUVELLE POLITIQUE DE MISE À JOUR AVEC ARRÊT DE LA FLOTTE DE CALCUL ET DES NŒUDS DE CONNEXION (non ajoutée par erreur dans la version 3.7.0).

### **QueueUpdateStrategy**(Facultatif,**String**)

Spécifie la stratégie de remplacement pour les paramètres de [SlurmQueues](#) section soumis à la politique de mise à jour suivante :

[Politique de mise à jour : le parc informatique doit être arrêté ou QueueUpdateStrategy défini pour que ce paramètre soit modifié en vue d'une mise à jour.](#)

La QueueUpdateStrategy valeur est utilisée uniquement lorsqu'un processus de mise à jour du cluster démarre.

Valeurs valides : COMPUTE\_FLEET\_STOP | DRAIN | TERMINATE

Valeur par défaut : COMPUTE\_FLEET\_STOP



## DRAIN

Les nœuds des files d'attente dont les valeurs de paramètres ont été modifiées sont définis sur `DRAINING`. Les nœuds dans cet état n'acceptent pas de nouvelles tâches et les tâches en cours d'exécution se poursuivent jusqu'à leur fin.

Une fois qu'un nœud devient `idle` (`DRAINED`), un nœud est remplacé s'il est statique, et le nœud est terminé s'il est dynamique. Les autres nœuds d'autres files d'attente dont les valeurs de paramètres n'ont pas été modifiées ne sont pas affectés.

Le temps nécessaire à cette stratégie pour remplacer tous les nœuds de file d'attente par des valeurs de paramètres modifiées dépend de la charge de travail en cours.

## COMPUTE\_FLEET\_STOP

La valeur par défaut du `QueueUpdateStrategy` paramètre. Avec ce paramètre, la mise à jour des paramètres dans la [SlurmQueues](#) section vous oblige à [arrêter le parc de calcul](#) avant d'effectuer une mise à jour du cluster :

```
$ pcluster update-compute-fleet --status STOP_REQUESTED
```

## TERMINATE

Dans les files d'attente dont les valeurs de paramètres ont été modifiées, les tâches en cours sont interrompues et les nœuds sont immédiatement mis hors tension.

Les nœuds statiques sont remplacés et les nœuds dynamiques sont interrompus.

Les autres nœuds d'autres files d'attente dont les valeurs de paramètres n'ont pas été modifiées ne sont pas affectés.

[Politique de mise à jour : ce paramètre n'est pas analysé lors d'une mise à jour.](#)

### Note

`QueueUpdateStrategy` est pris en charge à partir de AWS ParallelCluster la version 3.2.0.

## EnableMemoryBasedScheduling(Facultatif,**Boolean**)

Si `true`, la planification basée sur la mémoire est activée dans Slurm. Pour plus d'informations, consultez [SlurmQueues/ComputeResources/SchedulableMemory](#).

La valeur par défaut est `false`.

### Warning

L'activation de la planification basée sur la mémoire a un impact sur la façon dont le Slurm planificateur gère les tâches et l'allocation des nœuds.

Pour plus d'informations, consultez [Slurmpianification basée sur la mémoire](#).

### Note

`EnableMemoryBasedScheduling` est pris en charge à partir de AWS ParallelCluster la version 3.2.0.

### Note

À partir de AWS ParallelCluster la version 3.7.0, elle `EnableMemoryBasedScheduling` peut être activée si vous configurez plusieurs types d'[instances dans Instances](#).

Pour AWS ParallelCluster les versions 3.2.0 à 3.6. `x`, ne

`EnableMemoryBasedScheduling` peut pas être activé si vous configurez plusieurs types d'[instances dans Instances](#).

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

## CustomSlurmSettings(Facultatif,**[Dict]**)

Définit les Slurm paramètres personnalisés qui s'appliquent à l'ensemble du cluster.

Spécifie une liste de dictionnaires de Slurm configuration de paires clé-valeur à ajouter à la fin du `slurm.conf` fichier généré. AWS ParallelCluster

Chaque dictionnaire de la liste apparaît sous la forme d'une ligne distincte ajoutée au fichier Slurm de configuration. Vous pouvez définir des paramètres simples ou complexes.

Les paramètres simples se composent d'une seule paire de clés, comme le montrent les exemples suivants :

```
- Param1: 100
- Param2: "SubParam1,SubParam2=SubValue2"
```

Exemple rendu dans la Slurm configuration :

```
Param1=100
Param2=SubParam1,SubParam2=SubValue2
```

Les paramètres Slurm de configuration complexes se composent de plusieurs paires clé-valeur séparées par des espaces, comme indiqué dans les exemples suivants :

```
- nodeName: test-nodes[1-10]
 CPUs: 4
 RealMemory: 4196
 ... # other node settings
- NodeSet: test-nodeset
 Nodes: test-nodes[1-10]
 ... # other nodeset settings
- PartitionName: test-partition
 Nodes: test-nodeset
 ... # other partition settings
```

Exemple, rendu dans la Slurm configuration :

```
nodeName=test-nodes[1-10] CPUs=4 RealMemory=4196 ... # other node settings
NodeSet=test-nodeset Nodes=test-nodes[1-10] ... # other nodeset settings
PartitionName=test-partition Nodes=test-nodeset ... # other partition settings
```

#### Note

SlurmLes nœuds personnalisés ne doivent pas contenir les `-dy-` modèles `-st-` or dans leurs noms. Ces modèles sont réservés aux nœuds gérés par AWS ParallelCluster.

Si vous spécifiez des paramètres Slurm de configuration personnalisés dans `CustomSlurmSettings`, vous ne devez pas spécifier de paramètres Slurm de configuration personnalisés pour `CustomSlurmSettingsIncludeFile`.


Vous ne pouvez spécifier que des paramètres Slurm de configuration qui ne figurent pas dans la liste refusée. `CustomSlurmSettings` Pour plus d'informations sur les paramètres de Slurm configuration refusés, consultez [Listé par DenySlurmparamètres de configuration pourCustomSlurmSettings](#)

AWS ParallelCluster vérifie uniquement si un paramètre figure dans une liste de refus. AWS ParallelCluster ne valide pas la syntaxe ou la sémantique de vos paramètres de Slurm configuration personnalisés. Vous êtes responsable de la validation de vos paramètres de Slurm configuration personnalisés. Des paramètres Slurm de configuration personnalisés non valides peuvent provoquer des défaillances de Slurm daemon susceptibles d'entraîner des échecs de création et de mise à jour de clusters.

Pour plus d'informations sur la façon de spécifier des paramètres Slurm de configuration personnalisés avec AWS ParallelCluster, consultez [Slurmpersonnalisation de la configuration](#).

Pour plus d'informations sur les paramètres Slurm de configuration, consultez [slurm.conf](#) dans la documentation. Slurm

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

 Note

`CustomSlurmSettings` est pris en charge à partir de AWS ParallelCluster la version 3.6.0.


### **CustomSlurmSettingsIncludeFile**(Facultatif,**String**)

Définit les Slurm paramètres personnalisés qui s'appliquent à l'ensemble du cluster.

Spécifie le Slurm fichier personnalisé composé de paramètres de Slurm configuration personnalisés à ajouter à la fin du `slurm.conf` fichier AWS ParallelCluster généré.

Vous devez inclure le chemin du fichier. Le chemin peut commencer par `https://ous3://`.

Si vous spécifiez des paramètres Slurm de configuration personnalisés pour `CustomSlurmSettingsIncludeFile`, vous ne devez pas spécifier de paramètres Slurm de configuration personnalisés pour `CustomSlurmSettings`.

 Note

Slurm Les nœuds personnalisés ne doivent pas contenir les `-dy-` modèles `-st-` ou dans leurs noms. Ces modèles sont réservés aux nœuds gérés par AWS ParallelCluster.


Vous ne pouvez spécifier que des paramètres Slurm de configuration qui ne figurent pas dans la liste refusée. `CustomSlurmSettingsIncludeFile` Pour plus d'informations sur les paramètres de Slurm configuration refusés, consultez [Listé par DenySlurmparamètres de configuration pourCustomSlurmSettings](#)

AWS ParallelCluster vérifie uniquement si un paramètre figure dans une liste de refus. AWS ParallelCluster ne valide pas la syntaxe ou la sémantique de vos paramètres de Slurm configuration personnalisés. Vous êtes responsable de la validation de vos paramètres de Slurm configuration personnalisés. Des paramètres Slurm de configuration personnalisés non valides peuvent provoquer des défaillances de Slurm daemon susceptibles d'entraîner des échecs de création et de mise à jour de clusters.

Pour plus d'informations sur la façon de spécifier des paramètres Slurm de configuration personnalisés avec AWS ParallelCluster, consultez [Slurmpersonnalisation de la configuration](#).

Pour plus d'informations sur les paramètres Slurm de configuration, consultez [slurm.conf](#) dans la documentation. Slurm

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

 Note

`CustomSlurmSettings` est pris en charge à partir de AWS ParallelCluster la version 3.6.0.

## Database

(Facultatif) Définit les paramètres permettant d'activer la Slurm comptabilité sur le cluster. Pour plus d'informations, consultez [Slurmcomptabilité avec AWS ParallelCluster](#).

**Database:****Uri:** *string***UserName:** *string***PasswordSecretArn:** *string*

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

**Propriétés de Database****Uri(Obligatoire,String)**

Adresse du serveur de base de données utilisé comme serveur principal pour la Slurm comptabilité. Cet URI doit être formaté `host:port` et ne doit pas contenir de schéma, tel que `mysql://`. L'hôte peut être une adresse IP ou un nom DNS pouvant être résolu par le nœud principal. Si aucun port n'est fourni, AWS ParallelCluster utilise le port MySQL par défaut 3306.

AWS ParallelCluster connecte la base de données de Slurm comptabilité au cluster et doit accéder à la base de données.

La base de données doit être accessible avant que les événements suivants ne se produisent :

- Un cluster est créé.
- Slurmla comptabilité est activée avec une mise à jour du cluster.

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

**UserName(Obligatoire,String)**

Identité Slurm utilisée pour se connecter à la base de données, écrire des journaux de comptabilité et exécuter des requêtes. L'utilisateur doit disposer d'autorisations de lecture et d'écriture sur la base de données.

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

**PasswordSecretArn(Obligatoire,String)**

L'Amazon Resource Name (ARN) du AWS Secrets Manager secret qui contient le mot de passe en `UserName` texte clair. Ce mot de passe est utilisé conjointement avec la Slurm gestion `UserName` des comptes pour s'authentifier sur le serveur de base de données.

**Note**

Lorsque vous créez un secret à l'aide de la AWS Secrets Manager console, assurez-vous de sélectionner « Autre type de secret », de sélectionner du texte brut et d'inclure uniquement le texte du mot de passe dans le secret.

Pour plus d'informations sur la façon de AWS Secrets Manager créer un secret, reportez-vous à la section [Créer un AWS Secrets Manager secret](#)

Si l'utilisateur est autorisé à le faire [DescribeSecret](#), `PasswordSecretArn` est validé. `PasswordSecretArn` est valide si le secret spécifié existe. Si la politique IAM de l'utilisateur n'inclut pas `DescribeSecret`, `PasswordSecretArn` n'est pas validée et un message d'avertissement s'affiche. Pour plus d'informations, consultez [Politique AWS ParallelCluster pcluster utilisateur de base](#).

Lorsque vous effectuez une mise à jour `PasswordSecretArn`, le parc informatique doit être arrêté. Si la valeur du secret change et que l'ARN secret ne change pas, le cluster n'est pas automatiquement mis à jour avec le nouveau mot de passe de base de données. Pour mettre à jour le cluster en fonction de la nouvelle valeur secrète, vous devez exécuter la commande suivante depuis le nœud principal une fois le parc informatique arrêté.

```
$ sudo /opt/parallelcluster/scripts/slurm/update_slurm_database_password.sh
```

**Warning**

Nous vous recommandons de ne modifier le mot de passe de la base de données que lorsque le parc informatique est arrêté afin d'éviter toute perte de données comptables.

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

**DatabaseName**(Facultatif,**String**)

Nom de la base de données sur le serveur de base de données (défini par le paramètre `Uri`) à utiliser pour la Slurm comptabilité.

Le nom de la base de données peut contenir des lettres minuscules, des chiffres et des traits de soulignement. Le nom ne doit pas comporter plus de 64 caractères.

Ce paramètre correspond au StorageLoc paramètre de [slurmdbd.conf](#).

Si DatabaseName ce n'est pas le cas, il ParallelCluster utilisera le nom du cluster pour définir une valeur pourStorageLoc.

La mise à jour du DatabaseName est autorisée, sous réserve des considérations suivantes :

- Si une base de données portant un nom DatabaseName n'existe pas encore sur le serveur de base de données, slurmdbd la créera. Il vous incombera de reconfigurer la nouvelle base de données selon les besoins (par exemple en ajoutant les entités comptables : clusters, comptes, utilisateurs, associations, QoS, etc.).
- Si une base de données portant un nom existe DatabaseName déjà sur le serveur de base de données, slurmdbd l'utilisera pour la Slurm fonctionnalité de comptabilité.

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

#### Note

Databaseest ajouté à partir de la version 3.3.0.

## ExternalSlurmdbd

(Facultatif) Définit les paramètres permettant d'activer la Slurm comptabilité avec un serveur slurmdbd externe. Pour plus d'informations, consultez la section [SlurmComptabilité avec AWS ParallelCluster](#).

ExternalSlurmdbd:

Host: *string*

Port: *integer*

## Propriétés de **ExternalSlurmdbd**

### **Host**(Obligatoire,**String**)

Adresse du serveur slurmdbd externe pour la comptabilité. Slurm L'hôte peut être une adresse IP ou un nom DNS pouvant être résolu par le nœud principal.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)



## Port(Facultatif,Integer)

Port écouté par le service slurmdbd. La valeur par défaut est 6819.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## Dns

(Facultatif) Définit les paramètres Slurm qui s'appliquent à l'ensemble du cluster.

### Dns:

[DisableManagedDns](#): *boolean*

[HostedZoneId](#): *string*

[UseEc2Hostnames](#): *boolean*

## Propriétés de Dns

### DisableManagedDns(Facultatif,Boolean)

Dans `true` ce cas, les entrées DNS du cluster ne sont pas créées et les noms de Slurm nœuds ne peuvent pas être résolus.

Par défaut, AWS ParallelCluster crée une zone hébergée Route 53 dans laquelle les nœuds sont enregistrés lors du lancement. La valeur par défaut est `false`. S'il `DisableManagedDns` est défini sur `true`, la zone hébergée n'est pas créée par AWS ParallelCluster.

Pour savoir comment utiliser ce paramètre pour déployer des clusters dans des sous-réseaux sans accès à Internet, consultez [AWS ParallelCluster dans un seul sous-réseau sans accès à Internet.](#)

### Warning

Un système de résolution de noms est nécessaire au bon fonctionnement du cluster. Si `DisableManagedDns` ce paramètre est défini sur `true`, vous devez fournir un système de résolution de noms. Pour utiliser le DNS par défaut d'Amazon EC2, définissez sur `UseEc2Hostnames`. `true` Vous pouvez également configurer votre propre résolveur DNS et vous assurer que les noms de nœuds sont enregistrés lors du lancement des instances. Par exemple, vous pouvez le faire en configurant [CustomActions/OnNodeStart.](#)

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

### **HostedZoneId**(Facultatif,**String**)

Définit un ID de zone hébergée Route 53 personnalisé à utiliser pour la résolution des noms DNS du cluster. Lorsqu'il est fourni, il AWS ParallelCluster enregistre les nœuds du cluster dans la zone hébergée spécifiée et ne crée pas de zone hébergée gérée.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

### **UseEc2Hostnames**(Facultatif,**Boolean**)

Si `true`, les nœuds de calcul du cluster sont configurés avec le nom d'hôte EC2 par défaut. `SlurmNodeHostName` est également mis à jour avec ces informations. L'argument par défaut est `false`.

Pour savoir comment utiliser ce paramètre pour déployer des clusters dans des sous-réseaux sans accès à Internet, consultez [AWS ParallelCluster dans un seul sous-réseau sans accès à Internet](#).

#### Note

Cette note n'est pas pertinente à partir de AWS ParallelCluster la version 3.3.0. Pour les versions AWS ParallelCluster prises en charge avant la version 3.3.0 : Lorsqu'il `UseEc2Hostnames` est défini sur `true`, le fichier de configuration de Slurm est défini avec les scripts AWS ParallelCluster `prolog` et `epilog` :

- `prolog`s'exécute pour ajouter des informations `/etc/hosts` sur les nœuds aux nœuds de calcul lorsque chaque tâche est allouée.
- `epilog`s'exécute pour nettoyer le contenu écrit par `prolog`.

Pour ajouter des `epilog` scripts personnalisés `prolog` ou personnalisés, ajoutez-les respectivement aux `/opt/slurm/etc/pcluster/epilog.d/` dossiers `/opt/slurm/etc/pcluster/prolog.d/` or.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## **SharedStorage** Section


(Facultatif) Les paramètres de stockage partagé pour le cluster.

AWS ParallelCluster [prend en charge l'utilisation d'Amazon EBS, FSx for ONTAP et FSx pour les volumes de stockage partagés OpenZFS, d'Amazon EFS et FSx pour les systèmes de fichiers de stockage partagé Lustre ou de caches de fichiers.](#)

Dans SharedStorage cette section, vous pouvez définir le stockage externe ou géré :

- Le stockage externe fait référence à un volume ou à un système de fichiers existant que vous gérez. AWS ParallelCluster ne le crée ni ne le supprime.
- AWS ParallelCluster le stockage géré fait référence à un volume ou à un système de fichiers AWS ParallelCluster créé et pouvant être supprimé.

Pour les [quotas de stockage partagé](#) et plus d'informations sur la configuration de votre stockage partagé, consultez [Stockage partagé](#) la section Utilisation AWS ParallelCluster.

 Note

S'il AWS Batch est utilisé comme planificateur, FSx for Lustre n'est disponible que sur le nœud principal du cluster.

SharedStorage:

- MountDir: *string*  
Name: *string*  
StorageType: Ebs  
EbsSettings:
  - VolumeType: *string*
  - Iops: *integer*
  - Size: *integer*
  - Encrypted: *boolean*
  - KmsKeyId: *string*
  - SnapshotId: *string*
  - Throughput: *integer*
  - VolumeId: *string*
  - DeletionPolicy: *string*
- MountDir: *string*  
Name: *string*  
StorageType: Efs

EfsSettings:Encrypted: *boolean*KmsKeyId: *string*EncryptionInTransit: *boolean*IamAuthorization: *boolean*PerformanceMode: *string*ThroughputMode: *string*ProvisionedThroughput: *integer*FileSystemId: *string*DeletionPolicy: *string*- MountDir: *string*Name: *string*StorageType: FsxLustreFsxLustreSettings:StorageCapacity: *integer*DeploymentType: *string*ImportedFileChunkSize: *integer*DataCompressionType: *string*ExportPath: *string*ImportPath: *string*WeeklyMaintenanceStartTime: *string*AutomaticBackupRetentionDays: *integer*CopyTagsToBackups: *boolean*DailyAutomaticBackupStartTime: *string*PerUnitStorageThroughput: *integer*BackupId: *string*KmsKeyId: *string*FileSystemId: *string*AutoImportPolicy: *string*DriveCacheType: *string*StorageType: *string*DeletionPolicy: *string*DataRepositoryAssociations:- Name: *string*BatchImportMetaDataOnCreate: *boolean*DataRepositoryPath: *string*FileSystemPath: *string*ImportedFileChunkSize: *integer*AutoExportPolicy: *string*AutoImportPolicy: *string*- MountDir: *string*Name: *string*StorageType: FsxOntapFsxOntapSettings:

```

 VolumeId: string
- MountDir: string
 Name: string
 StorageType: FsxOpenZfs
 FsxOpenZfsSettings:
 VolumeId: string
- MountDir: string
 Name: string
 StorageType: FileCache
 FileCacheSettings:
 FileCacheId: string

```

## SharedStorage politiques de mise à jour

- Pour les EBS gérés/externes, les EFS gérés et les FSx Lustre gérés, la politique de mise à jour est [Politique de mise à jour : pour ce paramètre de valeurs de liste, le parc de calcul QueueUpdateStrategy doit être arrêté ou configuré pour ajouter une nouvelle valeur ; le parc de calcul doit être arrêté lors de la suppression d'une valeur existante.](#)
- Pour les systèmes EFS, FSx Lustre, FSx ONTAP, FSx OpenZfs et File Cache externes, la politique de mise à jour est la suivante : [Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## Propriétés de SharedStorage

MountDir(Obligatoire,String)

Le chemin où le stockage partagé est monté.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

Name(Obligatoire,String)

Nom du stockage partagé. Vous utilisez ce nom lorsque vous mettez à jour les paramètres.

### Warning

Si vous spécifiez le stockage partagé AWS ParallelCluster géré et que vous modifiez la valeur pourName, le stockage partagé géré et les données existants sont supprimés et un nouveau stockage partagé géré est créé. La modification de la valeur pour lors Name d'une mise à jour du cluster revient à remplacer le stockage partagé géré existant par un

nouveau. Assurez-vous de sauvegarder vos données avant de les modifier Name si vous devez conserver les données du stockage partagé existant.

Politique de mise à jour : pour ce paramètre de valeurs de liste, le parc de calcul QueueUpdateStrategy doit être arrêté ou configuré pour ajouter une nouvelle valeur ; le parc de calcul doit être arrêté lors de la suppression d'une valeur existante.

StorageType(Obligatoire,String)

Type de stockage partagé. Les valeurs prises en charge sont Ebs EfsFsxLustre,FsxOntap, etFsxOpenZfs.

Pour plus d'informations, consultez [FsxLustreSettings](#), [FsxOntapSettings](#) et [FsxOpenZfsSettings](#).

**Note**

Si vous l'utilisez AWS Batch comme planificateur, FSx for Lustre n'est disponible que sur le nœud principal du cluster.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

## EbsSettings

(Facultatif) Les paramètres d'un volume Amazon EBS.

```
EbsSettings:
 VolumeType: string
 Iops: integer
 Size: integer
 Encrypted: boolean
 KmsKeyId: string
 SnapshotId: string
 VolumeId: string
 Throughput: integer
 DeletionPolicy: string
 Raid:
 Type: string
```

`NumberOfVolumes`: *integer*

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

## Propriétés de **EbsSettings**

Lorsque le [DeletionPolicy](#) paramètre est défini sur `Delete`, un volume géré, avec ses données, est supprimé si le cluster est supprimé ou si le volume est supprimé lors d'une mise à jour du cluster.

Pour plus d'informations, voir [Stockage partagé](#) la section Utilisation AWS ParallelCluster.

### VolumeType(Facultatif,String)

Spécifie le [type de volume Amazon EBS](#). Les valeurs prises en charge sont `gp2`, `gp3io1`, `io2`, `sc1`, `st1`, et `standard`. La valeur par défaut est `gp3`.

Pour plus d'informations, consultez [Types de volumes Amazon EBS](#) dans le Guide de l'utilisateur Amazon EC2.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

### Iops(Facultatif,Integer)

Définit le nombre d'IOPS pour `io1io2`, et le `gp3` type de volumes.

La valeur par défaut, les valeurs prises en charge et `volume_iops` le `volume_size` ratio de production varient selon `VolumeType` et `Size`.

`VolumeType = io1`

Valeur par défaut `Iops = 100`

Valeurs prises en charge `Iops = 100—64 000 †`

`volume_size`Rapport `volume_iops` maximum = 50 IOPS pour chaque GiB. 5000 IOPS nécessitent au `volume_size` moins 100 GiB.

`VolumeType = io2`

Valeur par défaut `Iops = 100`

Valeurs prises en charge `Iops = 100 à 64 000 (256 000 pour les volumes io2 Block Express)`  
†

SizeRapport Iops maximum = 500 IOPS pour chaque GiB. 5000 IOPS nécessitent au Size moins 10 GiB.

VolumeType = gp3

Valeur par défaut Iops = 3000

Valeurs prises en charge Iops = 3000 à 16 000

SizeRapport Iops maximum = 500 IOPS pour chaque GiB. 5000 IOPS nécessitent au Size moins 10 GiB.

† Le maximum d'IOPS est garanti uniquement sur les [instances basées sur le système Nitro](#) et approvisionnées avec plus de 32 000 IOPS. Les autres instances garantissent jusqu'à 32 000 IOPS. À moins que vous ne [modifiez le volume](#), io1 les volumes antérieurs risquent de ne pas atteindre leurs performances optimales. io2 Les volumes Block Express prennent en charge volume\_iops des valeurs allant jusqu'à 256 000 pour les types d'R5instances. Pour plus d'informations, consultez la section sur les [volumes io2 Block Express](#) dans le guide de l'utilisateur Amazon EC2.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

Size(Facultatif,Integer)

Spécifie la taille du volume en gibioctets (GiB). La valeur par défaut est 35.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

Encrypted(Facultatif,Boolean)

Spécifie si le volume est chiffré. La valeur par défaut est true.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

KmsKeyId(Facultatif,String)

Spécifie une AWS KMS clé personnalisée à utiliser pour le chiffrement. Ce paramètre nécessite qu'il soit défini sur true. Encrypted

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

SnapshotId(Facultatif,String)

Spécifie l'ID du snapshot Amazon EBS si vous utilisez un instantané comme source pour le volume.




Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

VolumeId(Facultatif,String)

Spécifie l'ID du volume Amazon EBS. Lorsque cela est spécifié pour une EbsSettings instance, seul le MountDir paramètre peut également être spécifié.

Le volume doit être créé dans la même zone de disponibilité que leHeadNode.

 Note

Plusieurs zones de disponibilité ont été ajoutées dans la AWS ParallelCluster version 3.4.0.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

Throughput(Facultatif,Integer)

Débit, en Mbits/s à provisionner pour un volume, avec un maximum de 1 000 Mbits/s.

Ce paramètre n'est valide que lorsqu'il l'VolumeTypeestgp3. La plage prise en charge est comprise entre 125 et 1 000, avec une valeur par défaut de 125.

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.


DeletionPolicy(Facultatif,String)

Spécifie si le volume doit être conservé, supprimé ou capturé en instantané lorsque le cluster est supprimé ou que le volume est supprimé. Les valeurs prises en charge sont DeleteRetain, etSnapshot. La valeur par défaut est Delete.

Lorsque ce paramètre est [DeletionPolicy](#) défini sur Delete, un volume géré, avec ses données, est supprimé si le cluster est supprimé ou si le volume est supprimé lors d'une mise à jour du cluster.

Pour plus d'informations, consultez [Stockage partagé](#).

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

 Note

DeletionPolicyest pris en charge à partir de AWS ParallelCluster la version 3.2.0.

## Raid

(Facultatif) Définit la configuration d'un volume RAID.

### Raid:

Type: *string*

NumberOfVolumes: *integer*

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

### Propriétés de **Raid**

Type(Obligatoire,String)

Définit le type de matrice RAID. Les valeurs prises en charge sont « 0 » (rayé) et « 1 » (en miroir).

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

NumberOfVolumes(Facultatif,Integer)

Définit le nombre de volumes Amazon EBS à utiliser pour créer la matrice RAID. La plage de valeurs prise en charge est comprise entre 2 et 5. La valeur par défaut (lorsque le Raid paramètre est défini) est 2.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

## EfsSettings

(Facultatif) Les paramètres d'un système de fichiers Amazon EFS.

### EfsSettings:

Encrypted: *boolean*

KmsKeyId: *string*

EncryptionInTransit: *boolean*

IamAuthorization: *boolean*

PerformanceMode: *string*

ThroughputMode: *string*

ProvisionedThroughput: *integer*

FileSystemId: *string*

DeletionPolicy: *string*

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

## Propriétés de `EfsSettings`

Lorsque ce paramètre est [DeletionPolicy](#) défini sur `Delete`, un système de fichiers géré, avec ses données, est supprimé si le cluster est supprimé ou si le système de fichiers est supprimé lors d'une mise à jour du cluster.

Pour plus d'informations, voir [Stockage partagé](#) la section Utilisation AWS ParallelCluster.

`Encrypted`(Facultatif, Boolean)

Spécifie si le système de fichiers Amazon EFS est chiffré. La valeur par défaut est `false`.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

`KmsKeyId`(Facultatif, String)

Spécifie une AWS KMS clé personnalisée à utiliser pour le chiffrement. Ce paramètre nécessite qu'il soit défini sur `true`. `Encrypted`

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

`EncryptionInTransit`(Facultatif, Boolean)

S'il est défini sur `true`, les systèmes de fichiers Amazon EFS sont montés à l'aide du protocole TLS (Transport Layer Security). Par défaut, ce paramètre est défini sur `false`.

### Note

S'il AWS Batch est utilisé comme planificateur, il `EncryptionInTransit` n'est pas pris en charge.

### Note

`EncryptionInTransit` est ajouté à partir de AWS ParallelCluster la version 3.4.0.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

`IamAuthorization`(Facultatif, Boolean)

Si ce paramètre est défini sur `true`, Amazon EFS est authentifié à l'aide de l'identité IAM du système. Par défaut, ce paramètre est défini sur `false`.

**Note**

Si `IamAuthorization` est défini sur `true`, `EncryptionInTransit` doit également être défini sur `true`.

**Note**

S'il AWS Batch est utilisé comme planificateur, il `IamAuthorization` n'est pas pris en charge.

**Note**

`IamAuthorization` est ajouté à partir de AWS ParallelCluster la version 3.4.0.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

`PerformanceMode(Facultatif,String)`

Spécifie le mode de performance du système de fichiers Amazon EFS. Les valeurs prises en charge sont `generalPurpose` et `maxIO`. La valeur par défaut est `generalPurpose`. Pour plus d'informations, consultez [Modes de performances](#) dans le guide de l'utilisateur Amazon Elastic File System.

Nous recommandons le mode de performance `generalPurpose` pour la plupart des systèmes de fichiers.

Les systèmes de fichiers qui utilisent le mode de performance `maxIO` peuvent évoluer vers des niveaux plus élevés de débit cumulé et d'opérations par seconde. Cependant, il existe un compromis entre des latences légèrement plus élevées pour la plupart des opérations sur les fichiers.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

`ThroughputMode(Facultatif,String)`

Spécifie le mode de débit du système de fichiers Amazon EFS. Les valeurs prises en charge sont `bursting` et `provisioned`. La valeur par défaut est `bursting`. Quand il `provisioned` est utilisé, `ProvisionedThroughput` doit être spécifié.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

ProvisionedThroughput(Obligatoire quand ThroughputMode c'est le cas provisioned,Integer)

Définit le débit provisionné (en Mbits/s) du système de fichiers Amazon EFS, mesuré en Mbits/s. Cela correspond au [ProvisionedThroughputInMibps](#) paramètre indiqué dans la référence d'API Amazon EFS.

Si vous utilisez ce paramètre, vous devez définir ThroughputMode sur provisioned.

La plage prise en charge est de 1 -1024. Pour demander une augmentation de limite, contactez AWS Support.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

FileSystemId(Facultatif,String)

Définit l'ID du système de fichiers Amazon EFS pour un système de fichiers existant.

Si le cluster est configuré pour couvrir plusieurs zones de disponibilité, vous devez définir une cible de montage du système de fichiers dans chaque zone de disponibilité utilisée par le cluster.

Lorsque cela est spécifié, seul MountDir peut être spécifié. Aucune autre option EfsSettings ne peut être spécifiée.

Si vous définissez cette option, les conditions suivantes doivent être vraies pour les systèmes de fichiers que vous définissez :

- Les systèmes de fichiers ont une cible de montage existante dans chacune des zones de disponibilité du cluster, le trafic NFS entrant et sortant étant autorisé depuis et. HeadNode ComputeNodes Plusieurs zones de disponibilité sont configurées dans [Scheduling/SlurmQueues/Networking/SubnetIds](#).

Pour vous assurer que le trafic est autorisé entre le cluster et le système de fichiers, vous pouvez effectuer l'une des opérations suivantes :

- Configurez les groupes de sécurité de la cible de montage pour autoriser le trafic à destination et en provenance du CIDR ou de la liste de préfixes des sous-réseaux du cluster.

**Note**

AWS ParallelCluster valide que les ports sont ouverts et que le CIDR ou la liste de préfixes est configuré. AWS ParallelCluster ne valide pas le contenu du bloc ou de la liste de préfixes CIDR.

- Définissez des groupes de sécurité personnalisés pour les nœuds de cluster en utilisant [SlurmQueuesNetworking//SecurityGroupset](#) [HeadNode/Networking/SecurityGroups](#). Les groupes de sécurité personnalisés doivent être configurés pour autoriser le trafic entre le cluster et le système de fichiers.

**Note**

Si tous les nœuds du cluster utilisent des groupes de sécurité personnalisés, cela confirme AWS ParallelCluster uniquement que les ports sont ouverts. AWS ParallelCluster ne confirme pas que la source et la destination sont correctement configurées.

**Warning**

OneZone L'EFS n'est pris en charge que si tous les nœuds de calcul et le nœud principal se trouvent dans la même zone de disponibilité. L'EFS ne OneZone peut avoir qu'une seule cible de montage.

**Note**

Plusieurs zones de disponibilité ont été ajoutées dans la AWS ParallelCluster version 3.4.0.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)


**DeletionPolicy(Facultatif,String)**

Spécifie si le système de fichiers doit être conservé ou supprimé lorsque le système de fichiers est supprimé du cluster ou lorsque le cluster est supprimé. Les valeurs prises en charge sont Delete et Retain. La valeur par défaut est Delete.

Lorsque le [DeletionPolicy](#) paramètre est défini sur `Delete`, un système de fichiers géré, avec ses données, est supprimé si le cluster est supprimé ou si le système de fichiers est supprimé lors d'une mise à jour du cluster.


Pour plus d'informations, consultez [Stockage partagé](#).

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

 Note

`DeletionPolicy` est pris en charge à partir de AWS ParallelCluster la version 3.3.0.

## FsxLustreSettings

 Note

Vous devez définir `FsxLustreSettings` s'il `FsxLustre` est spécifié pour [StorageType](#).

(Facultatif) Les paramètres d'un système de fichiers FSx for Lustre.

### FsxLustreSettings:

```
StorageCapacity: integer
DeploymentType: string
ImportedFileChunkSize: integer
DataCompressionType: string
ExportPath: string
ImportPath: string
WeeklyMaintenanceStartTime: string
AutomaticBackupRetentionDays: integer
CopyTagsToBackups: boolean
DailyAutomaticBackupStartTime: string
PerUnitStorageThroughput: integer
BackupId: string # BackupId cannot coexist with some of the fields
KmsKeyId: string
FileSystemId: string # FileSystemId cannot coexist with other fields
AutoImportPolicy: string
DriveCacheType: string
StorageType: string
DeletionPolicy: string
```

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

**Note**

S'il AWS Batch est utilisé comme planificateur, FSx for Lustre n'est disponible que sur le nœud principal du cluster.

## Propriétés de **FsxLustreSettings**

Lorsque le [DeletionPolicy](#) paramètre est défini sur `Delete`, un système de fichiers géré, avec ses données, est supprimé si le cluster est supprimé ou si le système de fichiers est supprimé lors d'une mise à jour du cluster.

Pour plus d'informations, consultez [Stockage partagé](#).

### StorageCapacity(Obligatoire,Integer)

Définit la capacité de stockage du système de fichiers FSx for Lustre, en GiB.

`StorageCapacity` est obligatoire si vous créez un nouveau système de fichiers. N'incluez pas `StorageCapacity` si `BackupId` ou `FileSystemId` est spécifié.

- Pour `SCRATCH_2` les `PERSISTENT_1` types de `PERSISTENT_2` déploiement et les types de déploiement, les valeurs valides sont 1200 GiB, 2400 GiB et des incréments de 2400 GiB.
- Pour le type de déploiement `SCRATCH_1`, les valeurs valides sont 1 200 Gio, 2 400 Gio et des incréments de 3 600 Gio.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

### DeploymentType(Facultatif,String)

Spécifie le type de déploiement du système de fichiers FSx for Lustre. Les valeurs prises en charge sont `SCRATCH_1`, `SCRATCH_2`, `PERSISTENT_1` et `PERSISTENT_2`. La valeur par défaut est `SCRATCH_2`.

Choisissez `SCRATCH_1` les types de `SCRATCH_2` déploiement lorsque vous avez besoin d'un stockage temporaire et d'un traitement des données à court terme. Le type de `SCRATCH_2` déploiement fournit un chiffrement des données en transit et une capacité de débit en rafale supérieure à celle de `SCRATCH_1`.

Choisissez `PERSISTENT_1` le type de déploiement pour le stockage à long terme et pour les charges de travail axées sur le débit qui ne sont pas sensibles à la latence. `PERSISTENT_1` prend




en charge le chiffrement des données en transit. Il est disponible partout Régions AWS où FSx for Lustre est disponible.

Choisissez PERSISTENT\_2 le type de déploiement pour le stockage à long terme et pour les charges de travail sensibles à la latence qui nécessitent les plus hauts niveaux d'IOPS et de débit. PERSISTENT\_2 prend en charge le stockage SSD et offre une valeur supérieure PerUnitStorageThroughput (jusqu'à 1000 Mo/s/TiB). PERSISTENT\_2 est disponible en nombre limité de Régions AWS. Pour plus d'informations sur les types de déploiement et la liste des Régions AWS emplacements PERSISTENT\_2 disponibles, consultez les [options de déploiement du système de fichiers pour FSx for Lustre](#) dans le guide de l'utilisateur d'Amazon FSx for Lustre.

Le chiffrement des données en transit est automatiquement activé lorsque vous accédez à SCRATCH\_2 des systèmes de fichiers ou que vous PERSISTENT\_2 déployez à partir d'instances Amazon EC2 compatibles avec [cette](#) fonctionnalité. PERSISTENT\_1

Le chiffrement des données en transit pour SCRATCH\_2 et les types de PERSISTENT\_2 déploiement est pris en charge lorsque l'accès à partir des types d'instances pris en charge est pris en charge Régions AWS. PERSISTENT\_1 Pour plus d'informations, consultez la section [Chiffrement des données en transit dans](#) le guide de l'utilisateur d'Amazon FSx for Lustre.

 Note

Support pour le type de PERSISTENT\_2 déploiement a été ajouté avec AWS ParallelCluster la version 3.2.0.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

### ImportedFileChunkSize(Facultatif,Integer)

Pour les fichiers importés depuis un référentiel de données, cette valeur détermine le nombre de bandes et la quantité maximale de données pour chaque fichier (en MiB) stocké sur un seul disque physique. Le nombre maximal de disques sur lesquels un fichier unique peut être agrégé par bandes est limité au nombre total de disques qui composent le système de fichiers.

La taille de bloc par défaut est 1 024 Mio (1 Gio) et peut atteindre 512 000 Mio (500 Gio). Les objets Amazon S3 ont une taille maximale de 5 To.

**Note**

Ce paramètre n'est pas pris en charge pour les systèmes de fichiers utilisant le type de PERSISTENT\_2 déploiement. Pour savoir comment configurer les associations de référentiels de données, consultez la section [Lier votre système de fichiers à un compartiment S3](#) dans le guide de l'utilisateur d'Amazon FSx for Lustre.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

**DataCompressionType(Facultatif,String)**

Définit la configuration de compression des données pour le système de fichiers FSx for Lustre. La valeur prise en charge est LZ4. LZ4 indique que la compression des données est activée avec l'algorithme LZ4. Lorsque DataCompressionType ce n'est pas spécifié, la compression des données est désactivée lors de la création du système de fichiers.

Pour plus d'informations, consultez la section [Compression de données Lustre](#).

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

**ExportPath(Facultatif,String)**

Le chemin dans Amazon S3 où la racine de votre système de fichiers FSx for Lustre est exportée. Ce paramètre n'est pris en charge que ImportPath lorsqu'il est spécifié. Le chemin doit utiliser le même compartiment Amazon S3 que celui spécifié dans ImportPath. Vous pouvez fournir un préfixe facultatif vers lequel les données nouvelles et modifiées doivent être exportées depuis votre système de fichiers FSx for Lustre. Si aucune ExportPath valeur n'est fournie, FSx for Lustre définit un chemin `s3://import-bucket/FSxLustre[creation-timestamp]` d'exportation par défaut. L'horodatage est au format UTC, par exemple `s3://import-bucket/FSxLustre20181105T222312Z`.

Le compartiment d'exportation Amazon S3 doit être le même que le compartiment d'importation spécifié par ImportPath. Si vous spécifiez uniquement un nom de compartiment, par exemples `s3://import-bucket`, vous obtenez un mappage 1:1 des objets du système de fichiers vers les objets du compartiment Amazon S3. Ce mappage signifie que les données d'entrée dans Amazon S3 sont remplacées lors de l'exportation. Si vous fournissez un préfixe personnalisé dans le chemin d'exportation, par exemples `s3://import-bucket/[custom-optional-prefix]`, FSx for Lustre exporte le contenu de votre système de fichiers vers ce préfixe d'exportation dans le compartiment Amazon S3.

**Note**

Ce paramètre n'est pas pris en charge pour les systèmes de fichiers utilisant le type de PERSISTENT\_2 déploiement. Configurez les associations de référentiels de données comme décrit dans la section [Liaison de votre système de fichiers à un compartiment S3](#) dans le guide de l'utilisateur Amazon FSx for Lustre.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

**ImportPath(Facultatif,String)**

Le chemin d'accès au compartiment Amazon S3 (y compris le préfixe facultatif) que vous utilisez comme référentiel de données pour votre système de fichiers FSx for Lustre. La racine de votre système de fichiers FSx for Lustre sera mappée à la racine du compartiment Amazon S3 que vous sélectionnez. Par exemple : `s3://import-bucket/optional-prefix`. Si vous spécifiez un préfixe après le nom du compartiment Amazon S3, seules les clés d'objet comportant ce préfixe sont chargées dans le système de fichiers.

**Note**

Ce paramètre n'est pas pris en charge pour les systèmes de fichiers utilisant le type de PERSISTENT\_2 déploiement. Configurez les associations de référentiels de données comme décrit dans la section [Liaison de votre système de fichiers à un compartiment S3](#) dans le guide de l'utilisateur Amazon FSx for Lustre.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

**WeeklyMaintenanceStartTime(Facultatif,String)**

Heure de début préférée pour effectuer la maintenance hebdomadaire. Il est au "d:HH:MM" format du fuseau horaire UTC+0. Pour ce format, d il s'agit du numéro du jour de la semaine compris entre 1 et 7, commençant par le lundi et se terminant par le dimanche. Les guillemets sont obligatoires pour ce champ.

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.

## AutomaticBackupRetentionDays(Facultatif,Integer)

Nombre de jours de conservation des sauvegardes automatiques. La définition de cette valeur sur 0 désactive les sauvegardes automatiques. La plage prise en charge est comprise entre 0 et 90. La valeur par défaut est 0. Ce paramètre est uniquement valide pour une utilisation avec PERSISTENT\_1 et pour les types de PERSISTENT\_2 déploiement. Pour plus d'informations, consultez [Utilisation des sauvegardes](#) dans le Manuel de l'utilisateur Amazon FSx pour Lustre.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## CopyTagsToBackups(Facultatif,Boolean)

Si true, copiez les balises du système de fichiers FSx for Lustre dans les sauvegardes. La valeur par défaut est false. Si elle est définie sur true, toutes les balises du système de fichiers sont copiées dans toutes les sauvegardes automatiques et initiées par l'utilisateur lorsque l'utilisateur ne spécifie pas de balises. Si cette valeur est true et que vous spécifiez une ou plusieurs balises, seules les balises spécifiées sont copiées vers les sauvegardes. Si vous spécifiez une ou plusieurs balises lors de la création d'une sauvegarde initiée par l'utilisateur, aucune étiquette n'est copiée à partir du système de fichiers, quelle que soit cette valeur. Ce paramètre est uniquement valide pour une utilisation avec PERSISTENT\_1 et pour les types de PERSISTENT\_2 déploiement.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## DailyAutomaticBackupStartTime(Facultatif,String)

Une heure quotidienne récurrente, au HH:MM format. HH est l'heure de la journée remplie de zéro (00-23). MM est la minute de l'heure remplie de zéro (00-59). Par exemple, 05:00 indique 5 h du matin tous les jours. Ce paramètre est uniquement valide pour une utilisation avec PERSISTENT\_1 et pour les types de PERSISTENT\_2 déploiement.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## PerUnitStorageThroughput(Obligatoire pour les types PERSISTENT\_1 de PERSISTENT\_2 déploiement et les types de déploiement,Integer)

Décrit le débit de lecture et d'écriture pour chaque tebioctet de stockage, en Mo/s/TiB. La capacité de débit du système de fichiers est calculée en multipliant la capacité de stockage du système de fichiers (To) par le PerUnitStorageThroughput (Mo/S/To). Pour un système de fichiers de 2,4 To, l'allocation de 50 Mo/s/To de PerUnitStorageThroughput génère 120 Mo/s de débit de système de fichiers. Vous payez le débit que vous allouez. Cela correspond à la [PerUnitStorageThroughput](#) propriété.

Valeurs valides :

Stockage SSD PERSISTENT\_1 : 50, 100, 200 Mo/s/TiB.

PERSISTENT\_1 Stockage sur disque dur : 12, 40 Mo/s/TiB.

Stockage SSD PERSISTENT\_2 : 125, 250, 500, 1 000 Mo/s/TiB.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

BackupId(Facultatif,String)

Spécifie l'ID de la sauvegarde à utiliser pour restaurer le système de fichiers FSx for Lustre à partir d'une sauvegarde existante. Lorsque le BackupId paramètre est spécifié, les PerUnitStorageThroughput paramètres AutoImportPolicyDeploymentType,ExportPath,KmsKeyId,ImportPath,ImportedFileChunkSize et ne doivent pas être spécifiés. Ces paramètres sont lus à partir de la sauvegarde. En outreAutoImportPolicy, les ImportedFileChunkSize paramètres ExportPathImportPath,, et ne doivent pas être spécifiés. Cela correspond à la [BackupId](#)propriété.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

KmsKeyId(Facultatif,String)

ID de l'identifiant de clé AWS Key Management Service (AWS KMS) utilisé pour chiffrer les données du système de fichiers FSx for Lustre pour les systèmes de fichiers FSx for Lustre persistants au repos. Si elle n'est pas spécifiée, la clé gérée FSx for Lustre est utilisée. Les systèmes de fichiers SCRATCH\_1 et SCRATCH\_2 FSx for Lustre sont toujours chiffrés au repos à l'aide des clés gérées par FSx for Lustre. Pour plus d'informations, consultez la section [Chiffrer](#) dans le guide de référence de l'AWS Key Management Service API.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

FileSystemId(Facultatif,String)

Spécifie l'ID d'un système de fichiers FSx for Lustre existant.

Si cette option est spécifiée, seuls les FileSystemId paramètres MountDir et du FsxLustreSettings sont utilisés. Tous les autres paramètres du FsxLustreSettings sont ignorés.

**Note**

Si le AWS Batch planificateur est utilisé, FSx for Lustre n'est disponible que sur le nœud principal.

**Note**

Le système de fichiers doit être associé à un groupe de sécurité qui autorise le trafic TCP entrant et sortant via les ports 988, 1021, 1022 et 1023.

Assurez-vous que le trafic est autorisé entre le cluster et le système de fichiers en effectuant l'une des opérations suivantes :

- Configurez les groupes de sécurité du système de fichiers pour autoriser le trafic à destination et en provenance du CIDR ou de la liste de préfixes des sous-réseaux du cluster.

**Note**

AWS ParallelCluster valide que les ports sont ouverts et que le CIDR ou la liste de préfixes est configuré. AWS ParallelCluster ne valide pas le contenu du bloc ou de la liste de préfixes CIDR.

- Définissez des groupes de sécurité personnalisés pour les nœuds de cluster en utilisant [SlurmQueuesNetworking//SecurityGroupset](#) et [HeadNode/Networking/SecurityGroups](#). Les groupes de sécurité personnalisés doivent être configurés pour autoriser le trafic entre le cluster et le système de fichiers.

**Note**

Si tous les nœuds du cluster utilisent des groupes de sécurité personnalisés, cela confirme AWS ParallelCluster uniquement que les ports sont ouverts. AWS ParallelCluster ne confirme pas que la source et la destination sont correctement configurées.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## AutoImportPolicy(Facultatif,String)

Lorsque vous créez votre système de fichiers FSx for Lustre, vos objets Amazon S3 existants apparaissent sous forme de listes de fichiers et de répertoires. Utilisez cette propriété pour choisir comment FSx for Lustre met à jour vos listes de fichiers et de répertoires lorsque vous ajoutez ou modifiez des objets dans votre compartiment Amazon S3 associé. `AutoImportPolicy` peut avoir les valeurs suivantes :

- `NEW`- L'importation automatique est activée. FSx for Lustre importe automatiquement les listes de répertoires de tous les nouveaux objets ajoutés au compartiment Amazon S3 lié qui n'existent pas actuellement dans le système de fichiers FSx for Lustre.
- `NEW_CHANGED`- L'importation automatique est activée. FSx for Lustre importe automatiquement les listes de fichiers et de répertoires de tous les nouveaux objets ajoutés au compartiment Amazon S3 et de tous les objets existants modifiés dans le compartiment Amazon S3 une fois que vous avez choisi cette option.
- `NEW_CHANGED_DELETED`- L'importation automatique est activée. FSx for Lustre importe automatiquement les listes de fichiers et de répertoires de tous les nouveaux objets ajoutés au compartiment Amazon S3, de tous les objets existants modifiés dans le compartiment Amazon S3 et de tous les objets supprimés du compartiment Amazon S3 une fois que vous avez choisi cette option.

### Note

Support pour `NEW_CHANGED_DELETED` a été ajouté dans la AWS ParallelCluster version 3.1.1.

Si `AutoImportPolicy` ce n'est pas spécifié, l'importation automatique est désactivée. FSx for Lustre met uniquement à jour les listes de fichiers et de répertoires à partir du compartiment Amazon S3 lié lors de la création du système de fichiers. FSx for Lustre ne met pas à jour les listes de fichiers et de répertoires pour les objets nouveaux ou modifiés après avoir sélectionné cette option.

Pour plus d'informations, consultez [Importer automatiquement les mises à jour depuis votre compartiment S3](#) dans le guide de l'utilisateur d'Amazon FSx for Lustre.

**Note**

Ce paramètre n'est pas pris en charge pour les systèmes de fichiers utilisant le type de PERSISTENT\_2 déploiement. Pour savoir comment configurer les associations de référentiels de données, consultez la section [Lier votre système de fichiers à un compartiment S3](#) dans le guide de l'utilisateur d'Amazon FSx for Lustre.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

**DriveCacheType(Facultatif,String)**

Spécifie que le système de fichiers possède un cache de disque SSD. Cela ne peut être défini que si le `StorageType` paramètre est défini sur `HDD`, et le `DeploymentType` paramètre est défini sur `PERSISTENT_1`. Cela correspond à la `DriveCacheType` propriété. Pour plus d'informations, consultez les options de [déploiement de FSx for Lustre](#) dans le guide de l'utilisateur d'Amazon FSx for Lustre.

La seule valeur valide est `READ`. Pour désactiver le cache du disque SSD, ne spécifiez pas le `DriveCacheType` paramètre.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

**StorageType(Facultatif,String)**

Définit le type de stockage pour le système de fichiers FSx for Lustre que vous créez. Les valeurs valides sont `SSD` et `HDD`.

- Définissez ce paramètre sur `SSD` pour utiliser un stockage SSD.
- Configurez `HDD` pour utiliser le stockage sur disque dur. `HDD` est pris en charge sur les types de `PERSISTENT` déploiement.

La valeur par défaut est `SSD`. Pour plus d'informations, consultez [Options de type de stockage](#) dans le Guide de l'utilisateur Amazon FSx for Windows et [Options de stockage multiples](#) dans le Guide de l'utilisateur Amazon FSX for Lustre.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

**DeletionPolicy(Facultatif,String)**


Spécifie si le système de fichiers doit être conservé ou supprimé lorsque le système de fichiers est supprimé du cluster ou lorsque le cluster est supprimé. Les valeurs prises en charge sont `Delete` et `Retain`. La valeur par défaut est `Delete`.



Lorsque le [DeletionPolicy](#) paramètre est défini sur `Delete`, un système de fichiers géré, avec ses données, est supprimé si le cluster est supprimé ou si le système de fichiers est supprimé lors d'une mise à jour du cluster.

Pour plus d'informations, consultez [Stockage partagé](#).

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

 Note

`DeletionPolicy` est pris en charge à partir de AWS ParallelCluster la version 3.3.0.

`DataRepositoryAssociations`(Facultatif,String)

Liste des DRA (jusqu'à 8 par système de fichiers)

Chaque association de référentiel de données doit disposer d'un répertoire unique de système de fichiers Amazon FSx, et d'un préfixe ou compartiment S3 unique qui lui est associé.

Vous ne pouvez pas utiliser [ExportPath](#) et [ImportPath](#) en même temps que les DRA.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

`Name`(Obligatoire,String)

Le nom du DRA. Vous utilisez ce nom lorsque vous mettez à jour les paramètres.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

`BatchImportMetaDataOnCreate`(Facultatif,Boolean)

Un indicateur booléen déterminant si une tâche de référentiel de données d'importation permettant d'importer des métadonnées doit être exécutée après la création de l'association de référentiels de données. La tâche s'exécute si cet indicateur est défini sur `true`.

Valeur par défaut : `false`

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

`DataRepositoryPath`(Obligatoire,String)

Chemin d'accès au référentiel de données Amazon S3 qui sera lié au système de fichiers.

Le chemin d'accès peut être un compartiment S3 ou un préfixe au format `s3://myBucket/`

myPrefix/. Ce chemin d'accès indique l'emplacement d'importation ou d'exportation des fichiers dans le référentiel de données S3.

Ne peut pas se chevaucher avec d'autres DRA

Modèle : `^[^\u0000\u0085\u2028\u2029\r\n]{3,4357}$`

Minimum : 3

Maximum : 4357


[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

FileSystemPath(Obligatoire,String)

Un chemin d'accès sur le système de fichiers Amazon FSx pour Lustre qui pointe vers un répertoire de haut niveau (tel que /ns1/) ou un sous-répertoire (tel que /ns1/subdir/) qui sera mappé 1 à 1 avec DataRepositoryPath. Une barre oblique est requise au début du nom. Deux associations de référentiels de données ne peuvent pas avoir des chemins d'accès de système de fichiers qui se chevauchent. Par exemple, si un référentiel de données est associé au chemin d'accès du système de fichiers /ns1/, vous ne pouvez pas lier un autre référentiel de données au chemin d'accès du système de fichiers /ns1/ns2.

Ce chemin d'accès indique où, dans votre système de fichiers, les fichiers seront exportés ou importés. Ce répertoire de système de fichiers ne peut être lié qu'à un seul compartiment Amazon S3, et aucun autre compartiment S3 ne peut être lié à ce répertoire.

Ne peut pas se chevaucher avec d'autres DRA

 Note

Si vous spécifiez uniquement une barre oblique (/) comme chemin d'accès du système de fichiers, vous ne pouvez lier qu'un seul référentiel de données au système de fichiers. Vous ne pouvez spécifier « / » que comme chemin du système de fichiers pour le premier référentiel de données associé à un système de fichiers.

Modèle : `^[^\u0000\u0085\u2028\u2029\r\n]{1,4096}$`

Minimum : 1

Maximum : 4096

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

`ImportedFileChunkSize(Facultatif,Integer)`

Pour les fichiers importés à partir d'un référentiel de données, cette valeur détermine le nombre de bandes et la quantité maximale de données par fichier (en Mio) stockées sur un seul disque physique. Le nombre maximal de disques sur lesquels un fichier unique peut être agrégé par bandes est limité au nombre total de disques qui composent le système de fichiers ou le cache.

La taille de bloc par défaut est 1 024 Mio (1 Gio) et peut atteindre 512 000 Mio (500 Gio). Les objets Amazon S3 ont une taille maximale de 5 To.

Minimum : 1

Maximum : 4096

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

`AutoExportPolicy(Facultatif,Array of strings)`

La liste peut contenir une ou plusieurs des valeurs suivantes :

- **NEW** : les nouveaux fichiers et répertoires sont automatiquement exportés vers le référentiel de données à mesure qu'ils sont ajoutés au système de fichiers.
- **CHANGED** : les modifications apportées aux fichiers et aux répertoires du système de fichiers sont automatiquement exportées vers le référentiel de données.
- **DELETED** : les fichiers et les répertoires sont automatiquement supprimés du référentiel de données lorsqu'ils sont supprimés du système de fichiers.

Vous pouvez définir n'importe quelle combinaison de types d'événements pour votre `AutoExportPolicy`.

Maximum : 3

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

`AutoImportPolicy(Facultatif,Array of strings)`

La liste peut contenir une ou plusieurs des valeurs suivantes :

- **NEW** : Amazon FSx importe automatiquement les métadonnées des fichiers ajoutés au compartiment S3 lié qui n'existent pas actuellement dans le système de fichiers FSx.

- **CHANGED** : Amazon FSx met automatiquement à jour les métadonnées des fichiers et invalide le contenu des fichiers existants sur le système de fichiers à mesure que les fichiers changent dans le référentiel de données.
- **DELETED** : Amazon FSx supprime automatiquement les fichiers du système de fichiers à mesure que les fichiers correspondants sont supprimés dans le référentiel de données.

Vous pouvez définir n'importe quelle combinaison de types d'événements pour votre `AutoImportPolicy`.

Maximum : 3

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## FsxOntapSettings

### Note

Vous devez définir `FsxOntapSettings` s'il `FsxOntap` est spécifié pour [StorageType](#).

(Facultatif) Les paramètres d'un système de fichiers FSx for ONTAP.

```
FsxOntapSettings:
VolumeId: string
```

## Propriétés de FsxOntapSettings

`VolumeId`(Obligatoire,String)

Spécifie l'ID de volume du système FSx for ONTAP existant.


### Note

- Si un AWS Batch planificateur est utilisé, FSx for ONTAP n'est disponible que sur le nœud principal.
- Si le type de déploiement FSx for ONTAP est `Multi-AZ`, assurez-vous que la table de routage du sous-réseau du nœud principal est correctement configurée.
- Support de FSx pour ONTAP a été ajouté dans AWS ParallelCluster la version 3.2.0.

- Le système de fichiers doit être associé à un groupe de sécurité qui autorise le trafic TCP et UDP entrant et sortant via les ports 111, 635, 2049 et 4046.


Assurez-vous que le trafic est autorisé entre le cluster et le système de fichiers en effectuant l'une des actions suivantes :

- Configurez les groupes de sécurité du système de fichiers pour autoriser le trafic à destination et en provenance du CIDR ou de la liste de préfixes des sous-réseaux du cluster.

 Note

AWS ParallelCluster valide que les ports sont ouverts et que le CIDR ou la liste de préfixes est configuré. AWS ParallelCluster ne valide pas le contenu du bloc ou de la liste de préfixes CIDR.


- Définissez des groupes de sécurité personnalisés pour les nœuds de cluster en utilisant [SlurmQueuesNetworking//SecurityGroups](#) et [HeadNode/Networking/SecurityGroups](#). Les groupes de sécurité personnalisés doivent être configurés pour autoriser le trafic entre le cluster et le système de fichiers.

 Note

Si tous les nœuds du cluster utilisent des groupes de sécurité personnalisés, cela confirme AWS ParallelCluster uniquement que les ports sont ouverts. AWS ParallelCluster ne confirme pas que la source et la destination sont correctement configurées.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

## FsxOpenZfsSettings

 Note

Vous devez définir FsxOpenZfsSettings s'il FsxOpenZfs est spécifié pour [StorageType](#).

(Facultatif) Les paramètres d'un système de fichiers FSx pour OpenZFS.

FsxOpenZfsSettings:

VolumeId: *string*

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

## Propriétés de **FsxOpenZfsSettings**

VolumeId(Obligatoire,String)

Spécifie l'ID de volume du système FSx pour OpenZFS existant.

### Note

- Si un AWS Batch planificateur est utilisé, FSx pour OpenZFS n'est disponible que sur le nœud principal.
- Support de FSx pour OpenZFS a été ajouté dans la version 3.2.0. AWS ParallelCluster
- Le système de fichiers doit être associé à un groupe de sécurité qui autorise le trafic TCP et UDP entrant et sortant via les ports 111, 2049, 20001, 20002 et 20003.

Assurez-vous que le trafic est autorisé entre le cluster et le système de fichiers en effectuant l'une des opérations suivantes :

- Configurez les groupes de sécurité du système de fichiers pour autoriser le trafic à destination et en provenance du CIDR ou de la liste de préfixes des sous-réseaux du cluster.

### Note

AWS ParallelCluster valide que les ports sont ouverts et que le CIDR ou la liste de préfixes est configuré. AWS ParallelCluster ne valide pas le contenu du bloc ou de la liste de préfixes CIDR.

- Définissez des groupes de sécurité personnalisés pour les nœuds de cluster en utilisant [SlurmQueuesNetworking//SecurityGroupset](#) [HeadNode/Networking/SecurityGroups](#). Les groupes de sécurité personnalisés doivent être configurés pour autoriser le trafic entre le cluster et le système de fichiers.

**Note**

Si tous les nœuds du cluster utilisent des groupes de sécurité personnalisés, cela confirme AWS ParallelCluster uniquement que les ports sont ouverts. AWS ParallelCluster ne confirme pas que la source et la destination sont correctement configurées.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

**FileCacheSettings****Note**

Vous devez définir FileCacheSettings s'il FileCache est spécifié pour [StorageType](#).

(Facultatif) Les paramètres d'un cache de fichiers.

[FileCacheSettings](#):

[FileCacheId](#): *string*

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

**Propriétés de FileCacheSettings**

FileCacheId(Obligatoire,String)


Spécifie l'ID de cache de fichiers d'un cache de fichiers existant.

**Note**

- Le cache de fichiers ne prend pas en charge AWS Batch les planificateurs.
- Support pour le cache de fichiers a été ajouté dans AWS ParallelCluster la version 3.7.0.
- Le système de fichiers doit être associé à un groupe de sécurité qui autorise le trafic TCP entrant et sortant via le port 988.


Assurez-vous que le trafic est autorisé entre le cluster et le système de fichiers en effectuant l'une des opérations suivantes :

- Configurez les groupes de sécurité du cache de fichiers pour autoriser le trafic à destination et en provenance du CIDR ou de la liste de préfixes des sous-réseaux du cluster.

 Note

AWS ParallelCluster valide que les ports sont ouverts et que le CIDR ou la liste de préfixes est configuré. AWS ParallelCluster ne valide pas le contenu du bloc ou de la liste de préfixes CIDR.

- Définissez des groupes de sécurité personnalisés pour les nœuds de cluster en utilisant [SlurmQueuesNetworking//SecurityGroups](#) et [HeadNode/Networking/SecurityGroups](#). Les groupes de sécurité personnalisés doivent être configurés pour autoriser le trafic entre le cluster et le système de fichiers.

 Note

Si tous les nœuds du cluster utilisent des groupes de sécurité personnalisés, cela confirme AWS ParallelCluster uniquement que les ports sont ouverts. AWS ParallelCluster ne confirme pas que la source et la destination sont correctement configurées.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

## Iam Section

(Facultatif) Spécifie les propriétés IAM du cluster.

Iam:

Roles:

LambdaFunctionsRole: *string*

PermissionsBoundary: *string*

ResourcePrefix: *string*

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.



## Propriétés de **Iam**

### PermissionsBoundary(Facultatif,String)

L'ARN de la politique IAM à utiliser comme limite d'autorisations pour tous les rôles créés par AWS ParallelCluster. Pour plus d'informations, consultez [Limites d'autorisations pour les entités IAM](#) dans le Guide de l'utilisateur IAM. Le format est `arn:${Partition}:iam::${Account}:policy/${PolicyName}`.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

### Roles(Facultatif)

Spécifie les paramètres des rôles IAM utilisés par le cluster.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

### LambdaFunctionsRole(Facultatif,String)

L'ARN du rôle IAM à utiliser pour AWS Lambda. Cela remplace le rôle par défaut attaché à toutes les fonctions Lambda AWS CloudFormation soutenant des ressources personnalisées. Lambda doit être configuré en tant que principal autorisé à assumer le rôle. Cela ne remplacera pas le rôle des fonctions Lambda utilisées pour AWS Batch. Le format est `arn:${Partition}:iam::${Account}:role/${RoleName}`.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

### ResourcePrefix(Facultatif)

Spécifie un chemin ou un préfixe de nom pour les ressources IAM créées par AWS ParallelCluster

Le préfixe de ressource doit suivre les [règles de dénomination spécifiées par IAM](#) :

- Un nom peut contenir jusqu'à 30 caractères.
- Un nom ne peut être qu'une chaîne sans barre oblique (/).
- Un chemin peut comporter jusqu'à 512 caractères.
- Un chemin doit commencer et se terminer par une barre oblique (/). Il peut contenir plusieurs barres obliques (/) entre les barres obliques de début et de fin (). /
- Vous pouvez combiner le chemin et le nom/path/name.

Spécifiez un nom.

```
Iam:
 ResourcePrefix: my-prefix
```

Spécifiez un chemin.

```
Iam:
 ResourcePrefix: /org/dept/team/project/user/
```

Spécifiez un chemin et un nom.

```
Iam:
 ResourcePrefix: /org/dept/team/project/user/my-prefix
```

Si vous le spécifiez/*my-prefix*, une erreur est renvoyée.

```
Iam:
 ResourcePrefix: /my-prefix
```

Une erreur de configuration est renvoyée. Un chemin doit comporter deux / s. Un préfixe en lui-même ne peut pas avoir de / s.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## LoginNodes Section

### Note

Support pour LoginNodes est ajouté dans la AWS ParallelCluster version 3.7.0.

(Facultatif) Spécifie la configuration du pool de nœuds de connexion.

```
LoginNodes:
 Pools:
 - Name: string
 Count: integer
 InstanceType: string
 GracetimePeriod: integer
```

```
Image:
 CustomAmi: string
Ssh:
 KeyName: string
Networking:
 SubnetIds:
 - string
 SecurityGroups:
 - string
 AdditionalSecurityGroups:
 - string
Iam:
 InstanceRole: string
 InstanceProfile: string
 AdditionalIamPolicies:
 - Policy: string
```

Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.

## Propriétés de **LoginNodes**

## Propriétés de **Pools**

Définit des groupes de nœuds de connexion ayant la même configuration de ressources. Un seul pool peut être spécifié.

```
Pools:
 - Name: string
 Count: integer
 InstanceType: string
 GracetimePeriod: integer
 Image:
 CustomAmi: string
 Ssh:
 KeyName: string
 Networking:
 SubnetIds:
 - string
 SecurityGroups:
 - string
 AdditionalSecurityGroups:
 - string
```

```
Iam:
 InstanceRole: string
 InstanceProfile: string
 AdditionalIamPolicies:
 - Policy: string
```

### Name(ObligatoireString)

Spécifie le nom du LoginNodes pool. Ceci est utilisé pour étiqueter les LoginNodes ressources.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

### Count(ObligatoireInteger)

Spécifie le nombre de nœuds de connexion à maintenir actifs.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

### InstanceType(ObligatoireString)

Spécifie le type d'instance Amazon EC2 utilisé pour le nœud de connexion. L'architecture du type d'instance doit être identique à celle utilisée pour le Slurm InstanceType réglage.

[Politique de mise à jour](#) : ce paramètre peut être modifié si le pool de nœuds de connexion est arrêté.

### GracetimePeriod(FacultatifInteger)

Spécifie le délai minimal en minutes qui s'écoule entre la notification à l'utilisateur connecté indiquant qu'un nœud de connexion doit être mis hors service et l'événement d'arrêt réel. Les valeurs valides pour GracetimePeriod sont comprises entre 3 et 120 minutes. La valeur par défaut de est de 60 minutes.

#### Note

L'événement déclencheur implique des interactions entre plusieurs AWS services. Parfois, la latence du réseau et la propagation des informations peuvent prendre un certain temps, de sorte que le délai de grâce peut être plus long que prévu en raison de retards internes dans les AWS services.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## Image(Facultatif)

Définit la configuration de l'image pour les nœuds de connexion.

```
Image:
CustomAmi: String
```

### CustomAmi(FacultatifString)

Spécifie l'AMI personnalisée utilisée pour approvisionner les nœuds de connexion. Si elle n'est pas spécifiée, la valeur par défaut est celle spécifiée dans le [HeadNode Section](#).

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

## Ssh(Facultatif)

Définit la ssh configuration des nœuds de connexion.

```
Ssh:
KeyName: string
```

### KeyName(FacultatifString)

Spécifie la ssh clé utilisée pour se connecter aux nœuds de connexion. Si elle n'est pas spécifiée, la valeur par défaut est celle spécifiée dans le [HeadNode Section](#).

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

## Networking(Obligatoire)

```
Networking:
SubnetIds:
- string
SecurityGroups:
- string
AdditionalSecurityGroups:
- string
```

### SubnetIds(Obligatoire[String])

ID du sous-réseau existant dans lequel vous approvisionnez le pool de nœuds de connexion. Vous ne pouvez définir qu'un seul sous-réseau.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

### SecurityGroups(Facultatif[String])

Liste des groupes de sécurité à utiliser pour le pool de nœuds de connexion. Si aucun groupe de sécurité n'est spécifié, AWS ParallelCluster crée des groupes de sécurité pour vous.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

### AdditionalSecurityGroups(Facultatif[String])

Liste des groupes de sécurité supplémentaires à utiliser pour le pool de nœuds de connexion.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

### Iam(Facultatif)

Spécifie un rôle d'instance ou un profil d'instance à utiliser sur les nœuds de connexion pour remplacer le rôle d'instance ou le profil d'instance par défaut du cluster.

#### Iam:

```
InstanceRole: string
InstanceProfile: string
AdditionalIamPolicies:
- Policy: string
```

### InstanceProfile(FacultatifString)

Spécifie un profil d'instance pour remplacer le profil d'instance du nœud de connexion par défaut. Vous ne pouvez pas spécifier à la fois InstanceProfile et InstanceRole. Le format est `arn:Partition:iam::Account:instance-profile/InstanceProfileName`. Si cela est spécifié, les AdditionalIamPolicies paramètres InstanceRole et ne peuvent pas être spécifiés.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

### InstanceRole(FacultatifString)

Spécifie un rôle d'instance pour remplacer le rôle d'instance du nœud de connexion par défaut. Vous ne pouvez pas spécifier à la fois InstanceProfile et InstanceRole. Le format est `arn:Partition:iam::Account:role/RoleName`. Si cela est spécifié, les AdditionalIamPolicies paramètres S3Access et ne peuvent pas être spécifiés. Si cela

est spécifié, les `AdditionalIamPolicies` paramètres `InstanceProfile` et ne peuvent pas être spécifiés.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

`AdditionalIamPolicies`(Facultatif)

[AdditionalIamPolicies](#):

- [Policy](#): *string*

Une politique IAM Amazon Resource Name (ARN).

Spécifie une liste des Amazon Resource Names (ARN) des politiques IAM pour Amazon EC2. Cette liste est attachée au rôle root utilisé pour le nœud de connexion, en plus des autorisations requises par AWS ParallelCluster.

Le nom d'une politique IAM et son ARN sont différents. Les noms ne peuvent pas être utilisés.

Si cela est spécifié, les `InstanceRole` paramètres `InstanceProfile` et ne peuvent pas être spécifiés. Nous vous recommandons d'utiliser `AdditionalIamPolicies` car `AdditionalIamPolicies` ils sont ajoutés aux autorisations AWS ParallelCluster requises et `InstanceRole` doivent inclure toutes les autorisations requises. Les autorisations requises changent souvent d'une version à l'autre au fur et à mesure que des fonctionnalités sont ajoutées.

Il n'existe aucune valeur par défaut.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

`Policy`(Obligatoire[`String`])

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## Monitoring Section

(Facultatif) Spécifie les paramètres de surveillance du cluster.

Monitoring:

[Logs](#):

[CloudWatch](#):

[Enabled](#): *boolean*

RetentionInDays: *integer*

DeletionPolicy: *string*

Rotation:

Enabled: *boolean*

Dashboards:

CloudWatch:

Enabled: *boolean*

DetailedMonitoring: *boolean*

Alarms:

Enabled: *boolean*

Politique de mise à jour : ce paramètre n'est pas analysé lors d'une mise à jour.

## Propriétés de **Monitoring**

### Logs(Facultatif)

Les paramètres du journal pour le cluster.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

### CloudWatch(Facultatif)

Les paramètres CloudWatch des journaux pour le cluster.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

Enabled(Obligatoire,Boolean)

Si `true`, les journaux du cluster sont diffusés vers CloudWatch Logs. La valeur par défaut est `true`.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

RetentionInDays(Facultatif,Integer)

Le nombre de jours pendant lesquels les événements du journal sont conservés dans CloudWatch les journaux. La valeur par défaut est 180. Les valeurs prises en charge sont 0, 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827 et 3653. Une valeur de 0 utilisera le paramètre de conservation du CloudWatch journal par défaut, c'est-à-dire qu'il n'expirera jamais.

Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.



## DeletionPolicy(Facultatif,String)

Indique s'il faut supprimer les événements du journal dans les CloudWatch journaux lorsque le cluster est supprimé. Les valeurs possibles sont `Delete` et `Retain`. La valeur par défaut est `Retain`.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## Rotation(Facultatif)

Les paramètres de rotation des journaux pour le cluster.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## Enabled(Obligatoire,Boolean)

Si `true`, la rotation des journaux est activée. L'argument par défaut est `true`. Lorsqu'un fichier journal AWS ParallelCluster configuré atteint une certaine taille, il fait l'objet d'une rotation et une seule sauvegarde est conservée. Pour plus d'informations, consultez [AWS ParallelCluster rotation des journaux configurée](#).

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## Dashboards(Facultatif)

Les paramètres du tableau de bord pour le cluster.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## CloudWatch(Facultatif)

Les paramètres du CloudWatch tableau de bord pour le cluster.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## Enabled(Obligatoire,Boolean)

Si `true`, le CloudWatch tableau de bord est activé. La valeur par défaut est `true`.


[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## DetailedMonitoring(Facultatif,Boolean)

Si ce paramètre est défini sur `true`, la surveillance détaillée est activée pour le parc informatique des instances Amazon EC2. Lorsqu'elle est activée, la console Amazon EC2 affiche des graphiques permettant de surveiller les instances à intervalles d'une minute. L'activation de cette fonctionnalité entraîne des coûts supplémentaires. L'argument par défaut est `false`.

Pour plus d'informations, consultez [Activer ou désactiver la surveillance détaillée pour vos instances](#) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

 Note

DetailedMonitoringest ajouté à partir de AWS ParallelCluster la version 3.6.0.

## Alarms(Facultatif)


CloudWatch Alarmes pour le cluster.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

## Enabled(Facultatif)

Dans true ce cas, les CloudWatch alarmes du cluster seront créées. La valeur par défaut est true.

[Politique de mise à jour : ce paramètre peut être modifié lors d'une mise à jour.](#)

 Note

À partir de la AWS ParallelCluster version 3.8.0, les alarmes suivantes sont créées pour le nœud principal : Amazon EC2 Health Check, utilisation du CPU/de la mémoire/du disque et une alarme composite incluant toutes les autres.

## Tags Section

(Facultatif), Array définit les balises utilisées AWS CloudFormation et propagées à toutes les ressources du cluster. Pour plus d'informations, voir le [tag de AWS CloudFormation ressource](#) dans le guide de AWS CloudFormation l'utilisateur.

[Tags:](#)

- [Key:](#) *string*
- [Value:](#) *string*

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## Propriétés de **Tags**

Key(Obligatoire,String)

Définit le nom de la balise.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

Value(Obligatoire,String)

Définit la valeur de la balise.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## **AdditionalPackages** Section

(Facultatif) Utilisé pour identifier les packages supplémentaires à installer.

```
AdditionalPackages:
 IntelSoftware:
 IntelHpcPlatform: boolean
```

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## **IntelSoftware**

(Facultatif) Définit la configuration des solutions Intel Select.

```
IntelSoftware:
 IntelHpcPlatform: boolean
```

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## Propriétés de **IntelSoftware**

IntelHpcPlatform(Facultatif,Boolean)

Si true, indique que le [contrat de licence utilisateur final](#) pour Intel Parallel Studio est accepté. Cela entraîne l'installation d'Intel Parallel Studio sur le nœud principal et le partage avec les

nœuds de calcul. Cela ajoute plusieurs minutes au temps nécessaire au nœud principal pour démarrer. Le `IntelHpcPlatform` paramètre n'est pris en charge que sur CentOS 7.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

## DirectoryService Section

### Note

Support pour `DirectoryService` a été ajouté dans la AWS ParallelCluster version 3.1.1.

(Facultatif) Les paramètres du service d'annuaire pour un cluster qui prend en charge l'accès de plusieurs utilisateurs.

AWS ParallelCluster gère les autorisations qui prennent en charge l'accès de plusieurs utilisateurs aux clusters avec un Active Directory (AD) via le protocole LDAP (Lightweight Directory Access Protocol) pris en charge par le [démon des services de sécurité du système \(SSSD\)](#). Pour plus d'informations, consultez [Qu'est-ce qu' AWS Directory Service ?](#) dans le Guide de l'utilisateur AWS Directory Service .

Nous vous recommandons d'utiliser le protocole LDAP sur TLS/SSL (LDAPS en abrégé) pour garantir que toutes les informations potentiellement sensibles sont transmises via des canaux cryptés.

```
DirectoryService:
DomainName: string
DomainAddr: string
PasswordSecretArn: string
DomainReadOnlyUser: string
LdapTlsCaCert: string
LdapTlsReqCert: string
LdapAccessFilter: string
GenerateSshKeysForUsers: boolean
AdditionalSssdConfigs: dict
```

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

## Propriétés de **DirectoryService**

### Note

Si vous prévoyez de l'utiliser AWS ParallelCluster dans un seul sous-réseau sans accès à Internet, consultez [AWS ParallelCluster dans un seul sous-réseau sans accès à Internet](#) les exigences supplémentaires.

### DomainName(Obligatoire,String)

Le domaine Active Directory (AD) que vous utilisez pour les informations d'identité.

DomainName accepte à la fois les formats de nom de domaine complet (FQDN) et de nom distinctif LDAP (DN).

- Exemple de FQDN : corp.*example*.com
- Exemple de DN LDAP : DC=*corp*,DC=*example*,DC=*com*

Cette propriété correspond au paramètre sssd-ldap appelé. ldap\_search\_base

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

### DomainAddr(Obligatoire,String)

L'URI ou les URI qui pointent vers le contrôleur de domaine AD utilisé comme serveur LDAP. L'URI correspond au paramètre SSSD-LDAP appelé. ldap\_uri La valeur peut être une chaîne d'URI séparée par des virgules. Pour utiliser LDAP, vous devez ajouter des ldap:// éléments au début de chaque URI.

Exemples de valeur :

```
ldap://192.0.2.0,ldap://203.0.113.0 # LDAP
ldaps://192.0.2.0,ldaps://203.0.113.0 # LDAPS without support for certificate
verification
ldaps://abcdef01234567890.corp.example.com # LDAPS with support for certificate
verification
192.0.2.0,203.0.113.0 # AWS ParallelCluster uses LDAPS by
default
```

Si vous utilisez LDAPS avec vérification des certificats, les URI doivent être des noms d'hôte.


Si vous utilisez LDAPS sans vérification de certificat ni LDAP, les URI peuvent être des noms d'hôtes ou des adresses IP.

Utilisez le protocole LDAP sur TLS/SSL (LDAPS) pour éviter la transmission de mots de passe et d'autres informations sensibles sur des canaux non chiffrés. S'il AWS ParallelCluster ne trouve aucun protocole, il s'ajoute `ldaps://` au début de chaque URI ou nom d'hôte.

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

`PasswordSecretArn(Obligatoire,String)`

L'Amazon Resource Name (ARN) du AWS Secrets Manager secret qui contient le mot de passe en `DomainReadOnlyUser` texte clair. Le contenu du secret correspond au paramètre `SSSD-LDAP` appelé. `ldap_default_authtok`

 Note

Lorsque vous créez un secret à l'aide de la AWS Secrets Manager console, assurez-vous de sélectionner « Autre type de secret », de sélectionner du texte brut et d'inclure uniquement le texte du mot de passe dans le secret.

Pour plus d'informations sur la façon de AWS Secrets Manager créer un secret, reportez-vous à la section [Créer un AWS Secrets Manager secret](#)

Le client LDAP utilise le mot de passe pour s'authentifier auprès du domaine AD `DomainReadOnlyUser` lorsqu'il demande des informations d'identité.

Si l'utilisateur est autorisé à le faire [DescribeSecret](#), `PasswordSecretArn` est validé. `PasswordSecretArn` est valide si le secret spécifié existe. Si la politique IAM de l'utilisateur n'inclut pas `DescribeSecret`, `PasswordSecretArn` n'est pas validée et un message d'avertissement s'affiche. Pour plus d'informations, consultez [Politique AWS ParallelCluster pcluster utilisateur de base](#).

Lorsque la valeur du secret change, le cluster n'est pas automatiquement mis à jour. Pour mettre à jour le cluster en fonction de la nouvelle valeur secrète, vous devez arrêter le parc de calcul à l'aide de la [the section called "pcluster update-compute-fleet"](#) commande, puis exécuter la commande suivante depuis le nœud principal.

```
$ sudo /opt/parallelcluster/scripts/directory_service/
update_directory_service_password.sh
```

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

DomainReadOnlyUser(Obligatoire,String)

Identité utilisée pour interroger le domaine AD pour obtenir des informations d'identité lors de l'authentification des connexions des utilisateurs du cluster. Il correspond au paramètre SSSD-LDAP appelé. `ldap_default_bind_dn` Utilisez vos informations d'identité AD pour cette valeur.

Spécifiez l'identité sous la forme requise par le client LDAP spécifique qui se trouve sur le nœud :

- Microsoft AD :

```
cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
```

- Prothèse simple :

```
cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
```

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

LdapTlsCaCert(Facultatif,String)

Le chemin absolu vers un ensemble de certificats contenant les certificats de chaque autorité de certification de la chaîne de certification qui a émis un certificat pour les contrôleurs de domaine. Il correspond au paramètre SSSD-LDAP appelé. `ldap_tls_cacert`

Un bundle de certificats est un fichier composé de la concaténation de certificats distincts au format PEM, également connu sous le nom de format DER Base64 sous Windows. Il est utilisé pour vérifier l'identité du contrôleur de domaine AD qui agit en tant que serveur LDAP.

AWS ParallelCluster n'est pas responsable du placement initial des certificats sur les nœuds. En tant qu'administrateur du cluster, vous pouvez configurer le certificat dans le nœud principal manuellement une fois le cluster créé ou vous pouvez utiliser un [script bootstrap](#). Vous pouvez également utiliser une Amazon Machine Image (AMI) qui inclut le certificat configuré sur le nœud principal.

[Simple AD](#) ne fournit pas de support LDAPS. Pour savoir comment intégrer un annuaire Simple AD AWS ParallelCluster, consultez [Comment configurer un point de terminaison LDAPS pour Simple AD](#) dans le blog sur la AWS sécurité.

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

#### LdapTlsReqCert(Facultatif,String)

Spécifie les contrôles à effectuer sur les certificats de serveur dans une session TLS. Il correspond au paramètre SSSD-LDAP appelé. `ldap_tls_reqcert`

Valeurs valides : `never`, `allow`, `try`, `demand` et `hard`.

`neverallow`, et `try` autorisez les connexions à poursuivre même si des problèmes liés aux certificats sont détectés.

`demand` et `hard` autorisez la poursuite de la communication si aucun problème lié aux certificats n'est détecté.

Si l'administrateur du cluster utilise une valeur qui ne nécessite pas la validation du certificat pour réussir, un message d'avertissement lui est renvoyé. Pour des raisons de sécurité, nous vous recommandons de ne pas désactiver la vérification des certificats.

La valeur par défaut est `hard`.

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

#### LdapAccessFilter(Facultatif,String)

Spécifie un filtre pour limiter l'accès au répertoire à un sous-ensemble d'utilisateurs. Cette propriété correspond au paramètre SSSD-LDAP appelé. `ldap_access_filter` Vous pouvez l'utiliser pour limiter les requêtes à un AD qui prend en charge un grand nombre d'utilisateurs.

Ce filtre peut bloquer l'accès des utilisateurs au cluster. Cependant, cela n'a aucune incidence sur la détectabilité des utilisateurs bloqués.

Si cette propriété est définie, le paramètre SSSD `access_provider` est défini en ldap interne par AWS ParallelCluster et ne doit pas être modifié par [DirectoryService/AdditionalSssdConfigssettings](#).



Si cette propriété est omise et que l'accès utilisateur personnalisé n'est pas spécifié dans [DirectoryService/AdditionalSssdConfigs](#), tous les utilisateurs de l'annuaire peuvent accéder au cluster.

Exemples :

```
"!(cn=SomeUser*)" # denies access to every user with alias starting with "SomeUser"
"(cn=SomeUser*)" # allows access to every user with alias starting with "SomeUser"
"memberOf=cn=TeamOne,ou=Users,ou=CORP,dc=corp,dc=example,dc=com" # allows access
only to users in group "TeamOne".
```

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

GenerateSshKeysForUsers(Facultatif,Boolean)

Définit si AWS ParallelCluster une clé SSH est générée pour les utilisateurs du cluster immédiatement après leur authentification initiale sur le nœud principal.

Si ce paramètre est défini sur `true`, une clé SSH est générée et enregistrée `USER_HOME_DIRECTORY/.ssh/id_rsa`, si elle n'existe pas, pour chaque utilisateur après sa première authentification sur le nœud principal.

Pour un utilisateur qui n'a pas encore été authentifié sur le nœud principal, la première authentification peut avoir lieu dans les cas suivants :

- L'utilisateur se connecte au nœud principal pour la première fois avec son propre mot de passe.
- Dans le nœud principal, un `sudoer` passe pour la première fois à l'utilisateur : `su USERNAME`
- Dans le nœud principal, un `sudoer` exécute une commande en tant qu'utilisateur pour la première fois : `su -u USERNAME COMMAND`

Les utilisateurs peuvent utiliser la clé SSH pour les connexions suivantes au nœud principal du cluster et aux nœuds de calcul. Avec AWS ParallelCluster, les connexions par mot de passe aux nœuds de calcul du cluster sont désactivées par conception. Si un utilisateur ne s'est pas connecté au nœud principal, les clés SSH ne sont pas générées et l'utilisateur ne pourra pas se connecter aux nœuds de calcul.

L'argument par défaut est `true`.

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

## AdditionalSssdConfigs(Facultatif,Dict)

Un dict de paires clé-valeur contenant des paramètres SSSD et des valeurs à écrire dans le fichier de configuration SSSD sur les instances de cluster. Pour une description complète du fichier de configuration SSSD, consultez les pages de manuel sur instance SSSD et les fichiers de configuration associés.

Les paramètres et valeurs SSSD doivent être compatibles avec AWS ParallelCluster la configuration SSSD décrite dans la liste suivante.

- `id_provider` est défini sur `ldap` interne par AWS ParallelCluster et ne doit pas être modifié.
- `access_provider` est défini en `ldap` interne AWS ParallelCluster lorsque [DirectoryService/LdapAccessFilter](#) est spécifié, et ce paramètre ne doit pas être modifié.

Si [DirectoryService/LdapAccessFilter](#) est omis, sa `access_provider` spécification est également omise. Par exemple, si vous définissez sur `access_provider` `simple` in [AdditionalSssdConfigs](#), alors [DirectoryService](#)/ne [LdapAccessFilter](#) doit pas être spécifié.

Les extraits de configuration suivants sont des exemples de configurations valides pour. `AdditionalSssdConfigs`

Cet exemple active le niveau de débogage pour les journaux SSSD, restreint la base de recherche à une unité organisationnelle spécifique et désactive la mise en cache des informations d'identification.

```
DirectoryService:
...
AdditionalSssdConfigs:
 debug_level: "0xFFF0"
 ldap_search_base: OU=Users,OU=CORP,DC=corp,DC=example,DC=com
 cache_credentials: False
```

Cet exemple indique la configuration d'un SSSD. `simple` `access_provider` Les utilisateurs du `EngineeringTeam` ont accès à l'annuaire. [DirectoryService](#)/ne [LdapAccessFilter](#) doit pas être défini dans ce cas.

```
DirectoryService:
...
```

```
AdditionalSssdConfigs:
 access_provider: simple
 simple_allow_groups: EngineeringTeam
```

[Politique de mise à jour : le parc informatique doit être arrêté pour que ce paramètre soit modifié pour une mise à jour.](#)

## DeploymentSettings Section

### Note

DeploymentSettings est ajouté à partir de AWS ParallelCluster la version 3.4.0.

(Facultatif) Spécifie la configuration des paramètres de déploiement.

```
DeploymentSettings:
 LambdaFunctionsVpcConfig:
 SecurityGroupIds
 - string
 SubnetIds
 - string
 DisableSudoAccessForDefaultUser: Boolean
 DefaultUserHome: string # 'Shared' or 'Local'
```

## Propriétés de **DeploymentSettings**

### **LambdaFunctionsVpcConfig**

(Facultatif) Spécifie les AWS Lambda configurations VPC des fonctions. Pour plus d'informations, consultez [AWS Lambda Configuration VPC dans AWS ParallelCluster.](#)

```
LambdaFunctionsVpcConfig:
 SecurityGroupIds
 - string
 SubnetIds
 - string
```

## LambdaFunctionsVpcConfig properties

### SecurityGroupIds(Obligatoire,[String])

Liste des identifiants de groupes de sécurité Amazon VPC associés aux fonctions Lambda.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

### SubnetIds(Obligatoire,[String])

Liste des ID de sous-réseaux attachés aux fonctions Lambda.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

#### Note

Les sous-réseaux et les groupes de sécurité doivent se trouver dans le même VPC.

### DisableSudoAccessForDefaultUser propriété

#### Note

Cette option de configuration n'est prise en charge qu'avec Slurm les clusters.

(Facultatif) Dans ce cas `True`, les privilèges sudo de l'utilisateur par défaut seront désactivés. Cela s'applique à tous les nœuds du cluster.

```
Main DeploymentSettings section in config yaml(applyes to HN, CF and LN)
DeploymentSettings:
 DisableSudoAccessForDefaultUser: True
```

Pour mettre à jour la valeur de `DisableSudoAccessForDefaultUser`, vous devez arrêter le parc informatique et tous les nœuds de connexion.

Politique de mise à jour : le parc informatique et les nœuds de connexion doivent être arrêtés pour que ce paramètre soit modifié pour une mise à jour.

## propriété DefaultUserHome

Lorsqu'il est défini sur `Shared`, le cluster utilisera la configuration par défaut et partagera le répertoire des utilisateurs par défaut sur le cluster en `/home/<default user>`.

Lorsqu'il est défini sur `Local`, le nœud principal, les nœuds de connexion et les nœuds de calcul disposeront chacun d'un répertoire utilisateur local par défaut distinct stocké dans `local/home/<default user>`.

## Créez des fichiers de configuration d'image

AWS ParallelCluster la version 3 utilise les fichiers YAML 1.1 pour les paramètres de configuration de l'image de construction. Vérifiez que l'indentation est correcte afin de réduire les erreurs de configuration. Pour plus d'informations, consultez la spécification YAML 1.1 à l'adresse. <https://yaml.org/spec/1.1/>

Ces fichiers de configuration sont utilisés pour définir la manière dont vos AWS ParallelCluster AMI personnalisées sont créées à l'aide d'EC2 Image Builder. Les processus de création d'AMI personnalisés sont déclenchés à l'aide de la `pcluster build-image` commande. Pour des exemples de fichiers de configuration, consultez [https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/schemas/test\\_imagebuilder\\_schema/test\\_imagebuilder\\_schema](https://github.com/aws/aws-parallelcluster/tree/release-3.0/cli/tests/pcluster/schemas/test_imagebuilder_schema/test_imagebuilder_schema).

### Rubriques

- [Propriétés du fichier de configuration d'image de construction](#)
- [Build Section](#)
- [Image Section](#)
- [DeploymentSettings Section](#)

## Propriétés du fichier de configuration d'image de construction

`Region(Facultatif,String)`

Spécifie le Région AWS pour l'`build-image` opération. Par exemple, `us-east-2`.

## **Build Section**

(Obligatoire) Spécifie la configuration dans laquelle l'image sera créée.

**Build:****Iam:****IamSupport:** *string***InstanceType:** *string***SubnetId:** *string***ParentImage:** *string***Iam:****InstanceRole:** *string***InstanceProfile:** *string***CleanupLambdaRole:** *string***AdditionalIamPolicies:**- **Policy:** *string***PermissionsBoundary:** *string***Components:**- **Type:** *string***Value:** *string***Tags:**- **Key:** *string***Value:** *string***SecurityGroupIds:**- *string***UpdateOsPackages:****Enabled:** *boolean***Propriétés de Build****InstanceType**(Obligatoire,String)

Spécifie le type d'instance pour l'instance utilisée pour créer l'image.

**SubnetId**(Facultatif,String)

Spécifie l'ID d'un sous-réseau existant dans lequel fournir l'instance pour créer l'image. Le sous-réseau fourni nécessite un accès Internet.

**⚠ Warning**

`pcluster build-image` utilise le VPC par défaut. Si le VPC par défaut a été supprimé, par exemple en utilisant AWS Control Tower ou AWS Landing Zone, l'ID du sous-réseau doit être spécifié.

## ParentImage(Obligatoire,String)

Spécifie l'image de base. L'image parent peut être une AWS ParallelCluster AMI non-AMI ou une AWS ParallelCluster AMI officielle pour la même version. Vous ne pouvez pas utiliser une AMI AWS ParallelCluster officielle ou personnalisée provenant d'une autre version de AWS ParallelCluster. Le format doit être l'ARN d'une image `arn:Partition:imagebuilder:Region:Account:image/ImageName/ImageVersion` ou un ID d'AMIami-12345678.

## SecurityGroupIds(Facultatif,[String])

Spécifie la liste des identifiants de groupe de sécurité pour l'image.

## Imds

### Propriétés de **Imds**

(Facultatif) Spécifie les paramètres du service de métadonnées d'instance de ImageBuilder génération et de test (IMDS) Amazon EC2.

Imds:

ImdsSupport: *string*

## ImdsSupport(Facultatif,String)

Spécifie les versions IMDS prises en charge dans les instances de build et de ImageBuilder test Amazon EC2. Les valeurs prises en charge sont `v2.0` et `v1.0`. La valeur par défaut est `v2.0`.

Si `ImdsSupport` ce paramètre est défini sur `v1.0`, `IMDSv1` et `IMDSv2` sont tous deux pris en charge.

Si `ImdsSupport` ce paramètre est défini sur `v2.0`, seul `IMDSv2` est pris en charge.

Pour plus d'informations, consultez la section [Utiliser IMDSv2](#) dans le guide de l'utilisateur Amazon EC2 pour les instances Linux.

[Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.](#)

**Note**

À partir de AWS ParallelCluster la version 3.7.0, la valeur `ImdsSupport` par défaut est `v2.0`. Nous vous recommandons de définir `v2.0` et `ImdsSupport` de remplacer `IMDSv1` par `IMDSv2` dans vos appels d'actions personnalisés.

Support pour [Imds/ImdsSupport](#) est ajouté avec AWS ParallelCluster la version 3.3.0.

**Iam**Propriétés de **Iam**

(Facultatif) Spécifie les ressources IAM pour la création de l'image.

Iam:

```

InstanceRole: string
InstanceProfile: string
CleanupLambdaRole: string
AdditionalIamPolicies:
 - Policy: string
PermissionsBoundary: string

```

## InstanceProfile(Facultatif,String)

Spécifie un profil d'instance pour remplacer le profil d'instance par défaut pour l'instance EC2 Image Builder. `InstanceProfile` et `InstanceRole` et `AdditionalIamPolicies` ne peuvent pas être spécifiés ensemble. Le format est `arn:Partition:iam::Account:instance-profile/InstanceProfileName`.

## InstanceRole(Facultatif,String)

Spécifie un rôle d'instance pour remplacer le rôle d'instance par défaut pour l'instance EC2 Image Builder. `InstanceProfile` et `InstanceRole` et `AdditionalIamPolicies` ne peuvent pas être spécifiés ensemble. Le format est `arn:Partition:iam::Account:role/RoleName`.

## CleanupLambdaRole(Facultatif,String)

L'ARN du rôle IAM à utiliser pour la AWS Lambda fonction de sauvegarde de la ressource AWS CloudFormation personnalisée qui supprime les artefacts de génération à la fin de la génération. Lambda doit être configuré en tant que principal autorisé à assumer le rôle. Le format est `arn:Partition:iam::Account:role/RoleName`.



## AdditionalIamPolicies(Facultatif)

Spécifie des politiques IAM supplémentaires à associer à l'instance EC2 Image Builder utilisée pour produire l'AMI personnalisée.

### AdditionalIamPolicies:

- Policy: *string*

## Policy(Facultatif,[String])

Liste des politiques IAM. Le format est

arn:*Partition*:iam::*Account*:policy/*PolicyName*.

## PermissionsBoundary(Facultatif,String)

L'ARN de la politique IAM à utiliser comme limite d'autorisations pour tous les rôles créés par AWS ParallelCluster. Pour plus d'informations sur les limites des autorisations IAM, reportez-vous à la section [Limites des autorisations pour les entités IAM](#) dans le guide de l'utilisateur IAM. Le format est arn:*Partition*:iam::*Account*:policy/*PolicyName*.

## Components

### Propriétés de **Components**

(Facultatif) Spécifie les ImageBuilder composants Amazon EC2 à utiliser pendant le processus de création de l'AMI en plus de ceux fournis par défaut par AWS ParallelCluster. Ces composants peuvent être utilisés pour personnaliser le processus de création de l'AMI. Pour plus d'informations, consultez [AWS ParallelCluster Personnalisation de l'AMI](#).

### Components:

- Type: *string*
- Value: *string*

## Type(Facultatif,String)

Spécifie le type de la paire type/valeur du composant. Le type peut être arn ou script.

## Value(Facultatif,String)

Spécifie la valeur de la paire type/valeur du composant. Lorsque le type est arn, il s'agit de l'ARN d'un composant EC2 Image Builder. Lorsque le type est script, il s'agit du lien https ou s3 pointant vers le script à utiliser lors de la création du composant EC2 Image Builder.

## Tags

### Propriétés de **Tags**

(Facultatif) Spécifie la liste des balises à définir dans les ressources utilisées pour créer l'AMI.

#### Tags:

- Key: *string*
- Value: *string*

#### Key(Facultatif,String)

Définit le nom de la balise.

#### Value(Facultatif,String)

Définit la valeur de la balise.

## UpdateOsPackages

### Propriétés de **UpdateOsPackages**

(Facultatif) Spécifie si le système d'exploitation est mis à jour avant l'installation de la pile AWS ParallelCluster logicielle.

#### UpdateOsPackages:

- Enabled: *boolean*

#### Enabled(Facultatif,Boolean)

Si `true`, le système d'exploitation est mis à jour et redémarré avant d'installer le AWS ParallelCluster logiciel. L'argument par défaut est `false`.

#### Note

Lorsque cette option `UpdateOsPackages` est activée, tous les packages de système d'exploitation disponibles sont mis à jour, y compris le noyau. En tant que client, il vous incombe de vérifier que la mise à jour est compatible avec les dépendances de l'AMI qui ne sont pas incluses dans la mise à jour.

Supposons, par exemple, que vous créez une AMI pour la AWS ParallelCluster version X.0 fournie avec la version du noyau Y.0 et la version Z.0 de certains composants.

Supposons que la mise à jour disponible inclut la version Y.1 du noyau mise à jour sans mise à jour du composant Z.0. Avant de l'activerUpdateOsPackages, il est de votre responsabilité de vérifier que le composant Z.0 supporte le noyau Y.1.

## Image Section

(Facultatif) Définit les propriétés de l'image pour la création de l'image.

```
Image:
 Name: string
 RootVolume:
 Size: integer
 Encrypted: boolean
 KmsKeyId: string
 Tags:
 - Key: string
 Value: string
```

### Propriétés de **Image**

Name(Facultatif,String)

Spécifie le nom de l'AMI. S'il n'est pas spécifié, le nom utilisé lors de l'appel de la [pcluster build-image](#) commande est utilisé.

### Tags

Propriétés de **Tags**

(Facultatif) Spécifie les paires clé-valeur pour l'image.

```
Tags:
 - Key: string
 Value: string
```

Key(Facultatif,String)

Définit le nom de la balise.

Value(Facultatif,String)

Définit la valeur de la balise.

## RootVolume

Propriétés de **RootVolume**

(Facultatif) Spécifie les propriétés du volume racine de l'image.

RootVolume:

Size: *integer*

Encrypted: *boolean*

KmsKeyId: *string*

Size(Facultatif,Integer)

Spécifie la taille du volume racine de l'image, en GiB. La taille par défaut est la taille du [ParentImage](#) plus 27 GiB.

Encrypted(Facultatif,Boolean)

Spécifie si le volume est chiffré. La valeur par défaut est *false*.

KmsKeyId(Facultatif,String)

Spécifie l'ARN de la AWS KMS clé utilisée pour chiffrer le volume. Le format est "arn:*Partition*:kms:*Region*:*Account*:key/*KeyId*."

## DeploymentSettings Section

(Facultatif) Spécifie la configuration des paramètres de déploiement.

DeploymentSettings:

LambdaFunctionsVpcConfig:

SecurityGroupIds

- *string*

SubnetIds

- *string*

## Propriétés de **DeploymentSettings**

### **LambdaFunctionsVpcConfig**

(Facultatif) Spécifie les AWS Lambda configurations VPC des fonctions. Pour plus d'informations, consultez [AWS Lambda Configuration VPC dans AWS ParallelCluster](#).

LambdaFunctionsVpcConfig:

SecurityGroupIds

- *string*

SubnetIds

- *string*

### **LambdaFunctionsVpcConfig properties**

SecurityGroupIds(Obligatoire,[String])

Liste des identifiants de groupes de sécurité Amazon VPC associés aux fonctions Lambda.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

SubnetIds(Obligatoire,[String])

Liste des ID de sous-réseaux attachés aux fonctions Lambda.

Politique de mise à jour : si ce paramètre est modifié, la mise à jour n'est pas autorisée.

#### Note

Les sous-réseaux et les groupes de sécurité doivent se trouver dans le même VPC.

#### Note

DeploymentSettings est ajouté à partir de AWS ParallelCluster la version 3.4.0.

## AWS ParallelCluster Référence d'API

Cette section fournit des descriptions, des syntaxes et des exemples d'utilisation pour chacune des actions de l' AWS ParallelCluster API.

## Rubriques

- [Construire une image](#)
- [createCluster](#)
- [Supprimer le cluster](#)
- [deleteClusterInstances](#)
- [Supprimer l'image](#)
- [Décrire le cluster](#)
- [describeClusterInstances](#)
- [describeComputeFleet](#)
- [Décrire l'image](#)
- [getClusterLogÉvènements](#)
- [getClusterStackÉvènements](#)
- [getImageLogÉvènements](#)
- [getImageStackÉvènements](#)
- [Clusters de listes](#)
- [listClusterLogStreams](#)
- [listImageLogStreams](#)
- [Images de la liste](#)
- [listOfficialImages](#)
- [Mettre à jour le cluster](#)
- [updateComputeFleet](#)

## Construire une image

Créez une AWS ParallelCluster image personnalisée dans un Région AWS.

### Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)
- [Corps de la réponse](#)
- [Exemple](#)

## Syntaxe de demande

```
POST /v3/images/custom
{
 "imageConfiguration": "string",
 "imageId": "string",
 "dryrun": boolean,
 "region": "string",
 "rollbackOnFailure": boolean,
 "supressValidators": ["string"],
 "validationFailureLevel": "string"
}
```

## Corps de la demande

### Configuration de l'image

La configuration de l'image sous forme de document YAML.

Type : chaîne

Obligatoire : oui

### imageId

L'ID de l'image à créer.

Type : chaîne

Obligatoire : oui

### run à sec

Si ce paramètre est défini sur `true`, effectuez uniquement la validation des demandes sans créer de ressource. Utilisez ce paramètre pour valider la configuration de l'image. L'argument par défaut est `false`.

Type : valeur booléenne

Obligatoire : non

### region

Le dans Région AWS lequel vous exécutez la commande pour créer l'image.

Type : chaîne

Obligatoire : non

### rollbackOnFailure

Si cette valeur est définie sur `true`, la restauration de la pile d'images se produit si l'image ne parvient pas à être créée. L'argument par défaut est `false`.

Type : valeur booléenne

Obligatoire : non

### Supprimer les validateurs

Identifiez un ou plusieurs validateurs de configuration à supprimer.

Type : liste de chaînes

Format : (ALL | type:[A-Za-z0-9]+)

Obligatoire : non

### validationFailureLevel

Le niveau de validation minimal qui entraîne l'échec de la création de l'image. L'argument par défaut est `ERROR`.

Type : chaîne

Valeurs valides : INFO | WARNING | ERROR

Obligatoire : non

## Syntaxe de réponse

```
{
 "image": {
 "imageId": "string",
 "ec2AmiInfo": {
 "amiId": "string"
 },
 "region": "string",
```



```
"version": "string",
"cloudformationStackArn": "string",
"imageBuildStatus": "BUILD_IN_PROGRESS",
"cloudformationStackStatus": "CREATE_IN_PROGRESS"
},
"validationMessages": [
 {
 "id": "string",
 "type": "string",
 "level": "INFO",
 "message": "string"
 }
]
}
```

## Corps de la réponse

### image

#### imageId

ID de l'image.

Type : chaîne

#### cloudformationStackArn

Le nom de ressource Amazon (ARN) de la CloudFormation pile principale.

Type : chaîne

#### cloudformationStackStatus

L'état de la CloudFormation pile.

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

## ec2 AmiInfo

### ami\_id

L'ID de l'AMI Amazon EC2.

Type : chaîne

### imageBuildStatus

État de création de l'image.

Type : chaîne

Valeurs valides : BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE | DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

### region

Le dans Région AWS le quel l'image est intégrée.

Type : chaîne

### version

AWS ParallelCluster Version utilisée pour créer l'image.

Type : chaîne

## Messages de validation

Liste des messages dont le niveau de validation est inférieur à `validationFailureLevel`. La liste des messages est collectée lors de la validation de la configuration.

### id

L'ID du validateur.

Type : chaîne

### level

Le niveau de validation.

Type : chaîne

Valeurs valides : INFO | WARNING | ERROR

## message

Un message de validation.

Type : chaîne

## type

Type de validateur.

Type : chaîne

## Exemple

### Python

#### Demande

```
$ build_image(custom-image-id, custom-image-config.yaml)
```

#### Réponse 200

```
{
 'image': {
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/custom-image-id/711b76b0-af81-11ec-a29f-0ee549109f1f',
 'cloudformation_stack_status': 'CREATE_IN_PROGRESS',
 'image_build_status': 'BUILD_IN_PROGRESS',
 'image_id': 'custom-image-id',
 'region': 'us-east-1',
 'version': '3.2.1'
 }
}
```

## createCluster

Créez un cluster géré dans un Région AWS.

### Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)

- [Syntaxe de réponse](#)
- [Corps de la réponse](#)
- [Exemple](#)

## Syntaxe de demande

```
POST /v3/clusters
{
 "clusterName": "string",
 "clusterConfiguration": "string",
 "dryrun": boolean,
 "region": "string",
 "rollbackOnFailure": boolean,
 "suppressValidators": ["string"],
 "validationFailureLevel": "string"
}
```

## Corps de la demande

### Configuration du cluster

La configuration du cluster sous forme de document YAML.

Type : chaîne

Obligatoire : oui

### Nom du cluster

Nom du cluster à créer.

Le nom doit commencer par un caractère alphabétique. Le nom peut comporter jusqu'à 60 caractères. Si la Slurm comptabilité est activée, le nom peut comporter jusqu'à 40 caractères.

Type : chaîne

Obligatoire : oui

### run à sec

Si ce paramètre est défini sur `true`, effectuez uniquement la validation des demandes sans créer de ressource. Utilisez ce paramètre pour valider la configuration du cluster. L'argument par défaut est `false`.

Type : valeur booléenne

Obligatoire : non

region

Le dans Région AWS lequel se trouve le cluster.

Type : chaîne

Obligatoire : non

rollbackOnFailure

Si cette valeur est définie sur `true`, la restauration de la pile du cluster se produit si le cluster ne parvient pas à être créé. L'argument par défaut est `true`.

Type : valeur booléenne

Obligatoire : non

Supprimer les validateurs

Identifiez un ou plusieurs validateurs de configuration à supprimer.

Type : liste de chaînes

Format : (ALL | type : [A-Za-z0-9]+)

Obligatoire : non

validationFailureLevel

Le niveau de validation minimal qui entraîne l'échec de la création du cluster. L'argument par défaut est `ERROR`.

Type : chaîne

Valeurs valides : `INFO` | `WARNING` | `ERROR`

Obligatoire : non

## Syntaxe de réponse

```
{
```

```
"cluster": {
 "clusterName": "string",
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "clusterStatus": "CREATE_IN_PROGRESS",
 "scheduler": {
 "type": "string",
 "metadata": {
 "name": "string",
 "version": "string"
 }
 }
},
"validationMessages": [
 {
 "id": "string",
 "type": "string",
 "level": "INFO",
 "message": "string"
 }
]
}
```

## Corps de la réponse

### Nom du cluster

Nom du cluster.

Type : chaîne

### cloudformationStackArn

Le nom de ressource Amazon (ARN) de la CloudFormation pile principale.

Type : chaîne

### cloudformationStackStatus

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE

| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS |  
UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS | UPDATE\_ROLLBACK\_FAILED |  
UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS | UPDATE\_ROLLBACK\_COMPLETE

État du cluster

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

region

Le dans Région AWS lequel le cluster est créé.

Type : chaîne

planificateur

métadonnées

Les métadonnées du planificateur

name

Nom du planificateur.

Type : chaîne

version

Version du planificateur.

Type : chaîne

type

Type de planificateur.

Type : chaîne

version

AWS ParallelCluster Version utilisée pour créer le cluster.

Type : chaîne

## messages\_de validation

Liste des messages dont le niveau de validation est inférieur à `validationFailureLevel`. La liste des messages est collectée lors de la validation de la configuration.

id

L'ID du validateur.

Type : chaîne

level

Type : chaîne

Valeurs valides : INFO | WARNING | ERROR

message

Un message de validation.

Type : chaîne

type

Type du validateur.

Type : chaîne

## Exemple

Python

Demande

```
$ create_cluster(cluster_name_3x, cluster-config.yaml)
```

Réponse 200

```
{
 'cluster': {
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/cluster-3x/e0462730-50b5-11ed-99a3-0a5ddc4a34c7',
```



```
'cloudformation_stack_status': 'CREATE_IN_PROGRESS',
'cluster_name': 'cluster-3x',
'cluster_status': 'CREATE_IN_PROGRESS',
'region': 'us-east-1',
'scheduler': {
 'type': 'slurm'
},
'version': '3.2.1'
}
}
```

## Supprimer le cluster

Lancez la suppression d'un cluster.

### Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)
- [Corps de la réponse](#)
- [Exemple](#)

### Syntaxe de demande

```
DELETE /v3/clusters/{clusterName}
{
 "region": "string"
}
```

### Corps de la demande

#### Nom du cluster

Nom du cluster.

Type : chaîne

Obligatoire : oui

## region

Le dans Région AWS lequel le cluster est supprimé.

Type : chaîne

Obligatoire : non

## Syntaxe de réponse

```
{
 "cluster": {
 "clusterName": "string",
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "cloudformationStackStatus": "DELETE_IN_PROGRESS",
 "clusterStatus": "DELETE_IN_PROGRESS",
 "scheduler": {
 "type": "string",
 "metadata": {
 "name": "string",
 "version": "string"
 }
 }
 }
}
```

## Corps de la réponse

### cluster

Liste des instances de cluster

Nom du cluster

Nom d'un cluster.

Type : chaîne

cloudformationStackArn

Le nom de ressource Amazon (ARN) de la CloudFormation pile principale.

Type : chaîne

cloudformationStackStatus

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

État du cluster

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

region

Le dans Région AWS lequel le cluster est créé.

Type : chaîne

planificateur

métadonnées

Les métadonnées du planificateur.

name

Nom du planificateur.

Type : chaîne

version

La version du planificateur

Type : chaîne

## type

Type de planificateur.

Type : chaîne

## version

AWS ParallelCluster Version utilisée pour créer le cluster.

Type : chaîne

## Exemple

### Python

#### Demande

```
$ delete_cluster(cluster_name_3x)
```

#### Réponse 200

```
{
 'cluster': {
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/cluster_name_3x/16b49540-ae5-11ec-8e18-0ac1d712b241',
 'cloudformation_stack_status': 'DELETE_IN_PROGRESS',
 'cluster_name': 'cluster_name_3x',
 'cluster_status': 'DELETE_IN_PROGRESS',
 'region': 'us-east-1',
 'version': '3.2.1'
 }
}
```

## deleteClusterInstances

Initiez l'arrêt forcé de tous les nœuds de calcul du cluster. Cette action ne prend pas en charge AWS Batch les clusters.

### Rubriques

- [Syntaxe de demande](#)

- [Corps de la demande](#)
- [Corps de la réponse](#)
- [Exemple](#)

## Syntaxe de demande

```
DELETE /v3/clusters/{clusterName}/instances
{
 "force": boolean,
 "region": "string"
}
```

## Corps de la demande

### Nom du cluster

Nom du cluster.

Type : chaîne

Obligatoire : oui

### force

Si ce paramètre est défini sur `true`, forcez la suppression lorsque le cluster portant le nom donné est introuvable. L'argument par défaut est `false`.

Type : valeur booléenne

Obligatoire : non

### region

Le dans Région AWS lequel se trouve le cluster.

Type : chaîne

Obligatoire : non

## Corps de la réponse

Aucun

## Exemple

### Python

Demande

```
$ delete_cluster_instances(cluster_name_3x)
```

Réponse 200

Aucun

## Supprimer l'image

Commencez à supprimer l' AWS ParallelCluster image personnalisée.

### Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)
- [Corps de la réponse](#)
- [Exemple](#)

### Syntaxe de demande

```
DELETE /v3/images/custom/{imageId}
{
 "force": boolean,
 "region": "string"
}
```

### Corps de la demande

#### imageId

ID de l'image.

Type : chaîne

Obligatoire : oui

force

Si ce paramètre est défini sur `true`, forcez la suppression de l'AMI. Utilisez ce paramètre si certaines instances utilisent l'AMI ou si l'AMI est partagée. L'argument par défaut est `false`.

Type : valeur booléenne

Obligatoire : non

region

Le dans Région AWS lequel l'image a été créée.

Type : chaîne

Obligatoire : non

## Syntaxe de réponse

```
{
 "image": {
 "imageId": "string",
 "ec2AmiInfo": {
 "amiId": "string"
 },
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "imageBuildStatus": "DELETE_IN_PROGRESS",
 "cloudformationStackStatus": "DELETE_IN_PROGRESS"
 }
}
```

## Corps de la réponse

image

cloudformationStackArn

Le nom de ressource Amazon (ARN) de la CloudFormation pile principale.

Type : chaîne

## cloudformationStackStatus

L'état de la CloudFormation pile.

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

## ec2 AmiInfo

AMID

L'ID de l'AMI Amazon EC2.

Type : chaîne

## imageBuildStatus

État de création de l'image.

Type : chaîne

Valeurs valides : BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

## imageId

ID de l'image.

Type : chaîne

## region

Le dans Région AWS lequel l'image est créée.

Type : chaîne

## version

AWS ParallelCluster Version utilisée pour créer l'image.



Type : chaîne

## Exemple

### Python

Demande

```
$ delete_image(custom-image-id)
```

Réponse 200

```
{
 'image': {
 'image_build_status': 'DELETE_IN_PROGRESS',
 'image_id': 'custom-image-id',
 'region': 'us-east-1',
 'version': '3.2.1'
 }
}
```

## Décrire le cluster

Obtenez des informations détaillées sur un cluster existant.

### Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)
- [Corps de la réponse](#)
- [Exemple](#)

### Syntaxe de demande

```
GET /v3/clusters/{clusterName}
{
```

```
"region": "string"
}
```

## Corps de la demande

### Nom du cluster

Nom du cluster.

Type : chaîne

Obligatoire : oui

### region

Le dans Région AWS lequel se trouve le cluster.

Type : chaîne

Obligatoire : non

## Syntaxe de réponse

### Note

`failureReason` est passé à la version 3.5.0 à `failures` partir de AWS ParallelCluster la version 3.5.0.

```
{
 "clusterName": "string",
 "region": "string",
 "version": "string",
 "cloudFormationStackStatus": "CREATE_IN_PROGRESS",
 "clusterStatus": "CREATE_IN_PROGRESS",
 "scheduler": {
 "type": "string",
 "metadata": {
 "name": "string",
 "version": "string"
 }
 }
}
```

```
},
"cloudformationStackArn": "string",
"creationTime": "2019-08-24T14:15:22Z",
"lastUpdatedTime": "2019-08-24T14:15:22Z",
"clusterConfiguration": {
 "url": "string"
},
"computeFleetStatus": "START_REQUESTED",
"tags": [
 {
 "key": "string",
 "value": "string"
 }
],
"headNode": {
 "instanceId": "string",
 "instanceType": "string",
 "launchTime": "2019-08-24T14:15:22Z",
 "privateIpAddress": "string",
 "publicIpAddress": "string",
 "state": "pending"
},
"failures": [
 {
 "failureCode": "string",
 "failureReason": "string"
 }
]
"loginNodes": {
 "status": "string",
 "address": "string",
 "scheme": "string",
 "healthyNodes": integer,
 "unhealthyNodes": integer
}
}
```

## Corps de la réponse

### Nom du cluster

Nom du cluster.

Type : chaîne

## cloudformationStackArn

Le nom de ressource Amazon (ARN) de la CloudFormation pile principale.

Type : chaîne

## cloudformationStackStatus

L'état de la CloudFormation pile.

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS |  
UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS | UPDATE\_ROLLBACK\_FAILED |  
UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS | UPDATE\_ROLLBACK\_COMPLETE

## Configuration du cluster

url

URL du fichier de configuration du cluster.

Type : chaîne

## État du cluster

État du cluster.

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

## computeFleetStatus

État du parc informatique.

Type : chaîne

Valeurs valides : START\_REQUESTED | STARTING | RUNNING | PROTECTED |  
STOP\_REQUESTED | STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

## creationTime

Horodatage de la création du cluster.

Type : datetime

## lastUpdatedTime

Horodatage de la dernière mise à jour du cluster.

Type : datetime

## region

Le dans Région AWS lequel le cluster est créé.

Type : chaîne

## balises

Liste des balises associées au cluster.

### clé

Nom du tag.

Type : chaîne

### balise

Valeur du tag.

Type : chaîne

## version

AWS ParallelCluster Version utilisée pour créer le cluster.

Type : chaîne

## échecs

Liste des défaillances lorsque la pile de clusters est en CREATE\_FAILED état.

### Code d'échec

Le code d'échec lorsque la pile de clusters est en CREATE\_FAILED état.

Type : chaîne

Raison de l'échec

La raison de l'échec lorsque la pile de clusters est en CREATE\_FAILED état.

Type : chaîne

head\_node

Le nœud principal du cluster.

instanceld

L'ID de l'instance Amazon EC2.

Type : chaîne

instanceType

Type d'instance Amazon EC2.

Type : chaîne

launchTime

Heure à laquelle l'instance Amazon EC2 a été lancée.

Type : datetime

privateIpAddress

Adresse IP privée du cluster.

Type : chaîne

publicIpAddress

Adresse IP publique du cluster.

Type : chaîne

state

État de l'instance du nœud principal.

Type : chaîne

Valeurs valides : `pending` | `running` | `shutting-down` | `terminated` | `stopping`  
| `stopped`

## planificateur

### métadonnées

Les métadonnées du planificateur.

#### name

Nom du planificateur.

Type : chaîne

#### version

Version du planificateur.

Type : chaîne

## Nodes de connexion

### status

État du nœud de connexion.

Type : chaîne

Valeurs valides : `PENDING` | `FAILED` | `ACTIVE`

### adresse

Adresse du nœud de connexion.

Type : chaîne

### scheme

Le schéma du nœud de connexion.

Type : chaîne

### scheme

Nombre de nœuds sains.

Type : entier

## scheme

Le nombre de nœuds défectueux.

Type : entier

## type

Type de planificateur.

Type : chaîne

## Exemple

### Python

Demande

```
$ describe_cluster(cluster_name_3x)
```

Réponse 200

```
{
 'cloud_formation_stack_status': 'CREATE_COMPLETE',
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster_name_3x/16b49540-ae5-11ec-8e18-0ac1d712b241',
 'cluster_configuration': {
 'url': 'https://parallelcluster-....'
 },
 'cluster_name': 'cluster_name_3x',
 'cluster_status': 'CREATE_COMPLETE',
 'compute_fleet_status': 'RUNNING',
 'creation_time': datetime.datetime(2022, 3, 28, 22, 19, 9, 661000,
tzinfo=tzlocal()),
 'head_node': {
 'instance_id': 'i-abcdef01234567890',
 'instance_type': 't2.micro',
 'launch_time': datetime.datetime(2022, 3, 28, 22, 21, 56, tzinfo=tzlocal()),
 'private_ip_address': '172.31.56.3',
 'public_ip_address': '107.23.100.164',
 'state': 'running'
 },
}
```



```
'last_updated_time': datetime.datetime(2022, 3, 28, 22, 19, 9, 661000,
tzinfo=tzlocal()),
'region': 'us-east-1',
'tags': [
 {
 'key': 'parallelcluster:version', 'value': '3.2.1'
 }
],
'version': '3.2.1'
}
```

## describeClusterInstances

Décrivez les instances qui appartiennent à un cluster.

### Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)
- [Corps de la réponse](#)
- [Exemple](#)

### Syntaxe de demande

```
GET /v3/clusters/{clusterName}/instances
{
 "nextToken": "string",
 "nodeType": "string",
 "queueName": "string",
 "region": "string"
}
```

### Corps de la demande

#### Nom du cluster

Nom du cluster.

Type : chaîne

Obligatoire : oui

#### nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

Obligatoire : non

#### Type de nœud

Filtrez les instances par type de nœud.

Type : chaîne

Valeurs valides: HeadNode, ComputeNode, LoginNode

Obligatoire : non

#### Nom de la file d'attente

Filtrez les instances par nom de file d'attente.

Type : chaîne

Obligatoire : non

#### region

Le dans Région AWS lequel se trouve le cluster.

Type : chaîne

Obligatoire : non

## Syntaxe de réponse

```
{
 "nextToken": "string",
 "instances": [
 {
 "instanceId": "string",
 "instanceType": "string",
```

```
 "launchTime": "2019-08-24T14:15:22Z",
 "privateIpAddress": "string",
 "publicIpAddress": "string",
 "state": "pending",
 "nodeType": "HeadNode",
 "queueName": "string"
 }
]
```

## Corps de la réponse

### Instances

La liste des instances de cluster.

`instanceId`

L'ID de l'instance Amazon EC2.

Type : chaîne

`instanceType`

Type d'instance Amazon EC2.

Type : chaîne

`launchTime`

Heure à laquelle l'instance Amazon EC2 a été lancée.

Type : datetime

Type de nœud

Type de nœud.

Type : chaîne

Valeurs valides: HeadNode, ComputeNode, LoginNode

`publicIpAddress`

Adresse IP publique du cluster.

Type : chaîne

Nom de la file d'attente

Nom de la file d'attente dans laquelle l'instance Amazon EC2 sauvegarde un nœud.

Type : chaîne

state

État de l'instance Amazon EC2 du nœud.

Type : chaîne

Valeurs valides : pending | running | shutting-down | terminated | stopping  
| stopped

nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

## Exemple

Python

Demande

```
$ describe_cluster_instances(cluster_name_3x)
```

Réponse 200

```
{
 'instances': [
 {
 'instance_id': 'i-abcdef01234567890',
 'instance_type': 't2.micro',
 'launch_time': datetime.datetime(2022, 3, 30, 14, 2, 7, tzinfo=tzlocal()),
 'node_type': 'HeadNode',
 'private_ip_address': '192.0.2.5',
 'public_ip_address': '198.51.100.180',
 'state': 'running'
 }
]
}
```

```
 }
]
}
```

## describeComputeFleet

Décrivez l'état du parc informatique.

### Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)
- [Corps de la réponse](#)
- [Exemple](#)

### Syntaxe de demande

```
GET /v3/clusters/{clusterName}/computefleet
{
 "region": "string"
}
```

### Corps de la demande

#### Nom du cluster

Nom du cluster.

Type : chaîne

Obligatoire : oui

#### region

Le dans Région AWS lequel se trouve le cluster.

Type : chaîne

Obligatoire : non

## Syntaxe de réponse

```
{
 "status": "START_REQUESTED",
 "lastStatusUpdateTime": "2019-08-24T14:15:22Z"
}
```

## Corps de la réponse

### status

Type : chaîne

Valeurs valides : START\_REQUESTED | STARTING | RUNNING | PROTECTED | STOP\_REQUESTED | STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

### lastStatusUpdatedHeure

L'horodatage représentant l'heure de la dernière mise à jour du statut.

Type : datetime

## Exemple

### Python

#### Demande

```
$ describe_compute_fleet(cluster_name_3x)
```

#### Réponse 200

```
{
 'last_status_updated_time': datetime.datetime(2022, 3, 28, 22, 27, 14,
 tzinfo=tzlocal()),
 'status': 'RUNNING'
}
```

## Décrire l'image

Obtenez des informations détaillées sur une image existante.

## Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)
- [Corps de la réponse](#)
- [Exemple](#)

## Syntaxe de demande

```
GET /v3/images/custom/{imageId}
{
 "region": "string"
}
```

## Corps de la demande

### imageId

ID de l'image.

Type : chaîne

Obligatoire : oui

### region

Le dans Région AWS lequel l'image a été créée.

Type : chaîne

Obligatoire : non

## Syntaxe de réponse

```
{
 "imageId": "string",
 "region": "string",
 "version": "string",
```

```
"imageBuildStatus": "BUILD_IN_PROGRESS",
"imageBuildLogsArn": "string",
"cloudformationStackStatus": "CREATE_IN_PROGRESS",
"cloudformationStackStatusReason": "string",
"cloudformationStackArn": "string",
"creationTime": "2019-08-24T14:15:22Z",
"cloudformationStackCreationTime": "2019-08-24T14:15:22Z",
"cloudformationStackTags": [
 {
 "key": "string",
 "value": "string"
 }
],
"imageConfiguration": {
 "url": "string"
},
"imagebuilderImageStatus": "PENDING",
"imagebuilderImageStatusReason": "string",
"ec2AmiInfo": {
 "amiId": "string",
 "tags": [
 {
 "key": "string",
 "value": "string"
 }
],
 "amiName": "string",
 "architecture": "string",
 "state": "PENDING",
 "description": "string"
}
}
```

## Corps de la réponse

### imageId

ID de l'image pour laquelle vous souhaitez récupérer des informations détaillées.

Type : chaîne

### imageBuildStatus

État de création de l'image.



Type : chaîne

Valeurs valides : BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

Configuration de l'image

url

URL du fichier de configuration de l'image.

Type : chaîne

region

Le dans Région AWS lequel l'image est créée.

Type : chaîne

version

AWS ParallelCluster Version utilisée pour créer l'image.

Type : chaîne

cloudformationStackArn

Le nom de ressource Amazon (ARN) de la CloudFormation pile principale.

Type : chaîne

cloudformationStackCreationHeure

L'horodatage de la création de la CloudFormation pile.

Type : datetime

cloudformationStackStatus

L'état de la CloudFormation pile.

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS |

UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS | UPDATE\_ROLLBACK\_FAILED |  
UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS | UPDATE\_ROLLBACK\_COMPLETE

cloudformationStackStatusMotif

La raison de l'état de la CloudFormation pile.

Type : chaîne

cloudformationStackTags

La liste des balises de la CloudFormation pile.

clé

Le nom du tag.

Type : chaîne

value

Valeur de balise.

Type : chaîne

creationTime

Horodatage indiquant le moment où l'image a été créée.

Type : datetime

ec2 AmiInfo

AMID

L'ID de l'AMI Amazon EC2.

Type : chaîne

Nom de mon nom

Nom de l'AMI Amazon EC2.

Type : chaîne

application

L'architecture de l'AMI Amazon EC2.

Type : chaîne

state

État de l'AMI Amazon EC2.

Type : chaîne

Valeurs valides : PENDING | AVAILABLE | INVALID | DEREGISTERED | TRANSIENT  
| FAILED | ERROR

balises

Liste des balises AMI Amazon EC2.

clé

Nom du tag.

Type : chaîne

value

Valeur du tag.

Type : chaîne

imagebuilderImageStatus

Le ImageBuilder statut.

Type : chaîne

Valeurs valides : PENDING | CREATING | BUILDING | TESTING | DISTRIBUTING |  
INTEGRATING | AVAILABLE | CANCELLED | FAILED | DEPRECATED | DELETED

imagebuilderImageStatusMotif

Raison du statut de ImageBuilder l'image.

Type : chaîne

imageBuildLogsArn

Le nom de ressource Amazon (ARN) des journaux pour le processus de création de l'image.

Type : chaîne

## Exemple

### Python

#### Demande

```
$ describe_image(custom-image-id)
```

#### Réponse 200

```
{
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
custom-image-id/6accc570-b080-11ec-845e-0e2dc6386985',
 'cloudformation_stack_creation_time': datetime.datetime(2022, 3, 30, 23, 23, 33,
731000, tzinfo=tzlocal()),
 'cloudformation_stack_status': 'CREATE_IN_PROGRESS',
 'cloudformation_stack_tags': [
 {
 'key': 'parallelcluster:version', 'value': '3.2.1'
 },
 {
 'key': 'parallelcluster:image_name',
 'value': 'custom-image-id'
 },
 {
 'key': 'parallelcluster:custom-image-id',
 'value': 'custom-image-id'
 },
 {
 'key': 'parallelcluster:s3_bucket',
 'value': 'parallelcluster-abcdef01234567890-v1-do-not-delete'
 },
 {
 'key': 'parallelcluster:s3_image_dir',
 'value': 'parallelcluster/3.2.1/images/custom-image-id-1234567890abcdef0'
 },
 {
 'key': 'parallelcluster:build_log',
 'value': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/imagebuilder/
ParallelClusterImage-custom-image-id'
 },
 {
 'key': 'parallelcluster:build_config',
```

```
 'value': 's3://parallelcluster-abcdef01234567890-v1-do-not-delete/parallelcluster/3.2.1/images/custom-image-id-1234567890abcdef0/configs/image-config.yaml'
 }
],
'image_build_logs_arn': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/imagebuilder/ParallelClusterImage-alinux2-image',
'image_build_status': 'BUILD_IN_PROGRESS',
'image_configuration': {
 'url': 'https://parallelcluster-abcdef01234567890-v1-do-not-delete.s3.amazonaws.com/parallelcluster/3.2.1/images/custom-image-id-1234567890abcdef0/configs/image-config.yaml?...'
},
'image_id': 'custom-image-id',
'imagebuilder_image_status': 'PENDING',
'region': 'us-east-1',
'version': '3.2.1'
}
```

## getClusterLogÉvènements

Récupérez les événements associés à un flux de log.

### Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)
- [Corps de la réponse](#)
- [Exemple](#)

### Syntaxe de demande

```
GET /v3/clusters/{clusterName}/logstreams/{logStreamName}
{
 "endTime": datetime,
 "limit": float,
 "nextToken": "string",
 "region": "string",
 "startFromHead": boolean,
```

```
"startTime": datetime
}
```

## Corps de la demande

### Nom du cluster

Nom du cluster.

Type : chaîne

Obligatoire : oui

### logStreamName

Nom du flux de journaux.

Type : chaîne

Obligatoire : oui

### endTime

Fin de la plage de temps, exprimée au format ISO 8601. Les événements dont l'horodatage est égal ou supérieur à cette heure ne sont pas inclus.

Type : datetime

Format : 2021-01-01T20:00:00Z

Obligatoire : non

### limite

Le nombre maximum d'événements de journal renvoyés. Si vous ne spécifiez aucune valeur, le maximum est le nombre d'événements de journal que peut contenir une taille de réponse de 1 Mo, jusqu'à 10 000 événements de journal.

Type : décimal

Obligatoire : non

### nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

Obligatoire : non

region

Le dans Région AWS lequel se trouve le cluster.

Type : chaîne

Obligatoire : non

startFromHead

Si ce paramètre est défini sur `true`, les premiers événements du journal sont renvoyés en premier. Si la valeur est égale à cette valeur `false`, les derniers événements du journal sont renvoyés en premier. L'argument par défaut est `false`.

Type : valeur booléenne

Obligatoire : non

startTime

Début de la plage horaire, exprimé au format ISO 8601. Les événements dont l'horodatage est égal à cette heure ou postérieur à cette heure sont inclus.

Type : datetime

Format : `2021-01-01T20:00:00Z`

Obligatoire : non

## Syntaxe de réponse

```
{
 "nextToken": "string",
 "prevToken": "string",
 "events": [
 {
 "timestamp": "2019-08-24T14:15:22Z",
 "message": "string"
 }
]
}
```

```
}
```

## Corps de la réponse

### événements

Liste des événements filtrés.

message

Le message de l'événement.

Type : chaîne

timestamp

Horodatage de l'événement.

Type : datetime

### nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

### PrevToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

## Exemple

### Python

Demande

```
$ get_cluster_log_events(cluster_name_3x, log_stream_name=ip-192-0-2-26.i-
abcdef01234567890.cfn-init)
```

Réponse 200

```
"events": [
 {
```



```
"message": "2022-09-22 16:40:15,127 [DEBUG] CloudFormation client initialized
with endpoint https://cloudformation.us-east-1.amazonaws.com",
 "timestamp": "2022-09-22T16:40:15.127Z"
},
{
 "message": "2022-09-22 16:40:15,127 [DEBUG] Describing resource
HeadNodeLaunchTemplate in stack cluster_name_3x",
 "timestamp": "2022-09-22T16:40:15.127Z"
},
...
]
```

## getClusterStackÉvènements

Récupérez les événements associés à la pile d'un cluster.

### Note

À partir de la version 3.6.0, AWS ParallelCluster utilise des piles imbriquées pour créer les ressources associées aux files d'attente et aux ressources de calcul. L'GetClusterStackEventsAPI et la `pcluster get-cluster-stack-events` commande renvoient uniquement les événements de la pile principale du cluster. Vous pouvez consulter les événements de la pile de clusters, y compris ceux liés aux files d'attente et aux ressources de calcul, dans la CloudFormation console.

### Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)
- [Corps de la réponse](#)
- [Exemple](#)

### Syntaxe de demande

```
GET /v3/clusters/{clusterName}/stackevents
{
```

```
"nextToken": "string",
"region": "string"
}
```

## Corps de la demande

### Nom du cluster

Nom du cluster.

Type : chaîne

Obligatoire : oui

### nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

Obligatoire : non

### region

Le dans Région AWS lequel se trouve le cluster.

Type : chaîne

Obligatoire : non

## Syntaxe de réponse

```
{
 "nextToken": "string",
 "events": [
 {
 "stackId": "string",
 "eventId": "string",
 "stackName": "string",
 "logicalResourceId": "string",
 "physicalResourceId": "string",
 "resourceType": "string",
 "timestamp": "2019-08-24T14:15:22Z",
 "resourceStatus": "CREATE_IN_PROGRESS",
```

```
 "resourceStatusReason": "string",
 "resourceProperties": "string",
 "clientRequestToken": "string"
 }
]
}
```

## Corps de la réponse

### événements

Liste des événements filtrés.

`clientRequestToken`

Le jeton transmis à l'action qui a généré cet événement.

Type : chaîne

`eventId`

L'identifiant unique de cet événement.

Type : chaîne

`logicalResourceid`

Nom logique de la ressource spécifiée dans le modèle.

Type : chaîne

`physicalResourceid`

Le nom ou l'identifiant unique associé à l'instance physique de la ressource.

Type : chaîne

Propriétés des ressources

Blob des propriétés utilisées pour créer la ressource.

Type : chaîne

État des ressources

Statut de la ressource.

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE | DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE | DELETE\_SKIPPED | UPDATE\_IN\_PROGRESS | UPDATE\_FAILED | UPDATE\_COMPLETE | IMPORT\_FAILED | IMPORT\_COMPLETE | IMPORT\_IN\_PROGRESS | IMPORT\_ROLLBACK\_IN\_PROGRESS | IMPORT\_ROLLBACK\_FAILED | IMPORT\_ROLLBACK\_COMPLETE

resourceStatusReason

Message de réussite ou d'échec associé à la ressource.

Type : chaîne

resourceType

Type de ressource.

Type : chaîne

ID de pile

Le nom d'identification unique de l'instance de la pile.

Type : chaîne

Nom de la pile

Le nom associé à une pile.

Type : chaîne

timestamp

Heure à laquelle le statut a été mis à jour.

Type : datetime

nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

## Exemple

Python

Demande

```
$ get_cluster_stack_events(cluster_name_3x)
```

Réponse 200

```
{
 'events': [
 {
 'event_id': '590b3820-b081-11ec-985e-0a7af5751497',
 'logical_resource_id': 'cluster_name_3x',
 'physical_resource_id': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster_name_3x/11a59710-b080-11ec-b8bd-129def1380e9',
 'resource_status': 'CREATE_COMPLETE',
 'resource_type': 'AWS::CloudFormation::Stack',
 'stack_id': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
cluster_name_3x/11a59710-b080-11ec-b8bd-129def1380e9',
 'stack_name': 'cluster_name_3x',
 'timestamp': datetime.datetime(2022, 3, 30, 23, 30, 13, 268000,
tzinfo=tzlocal())
 },
 ...
]
}
```

## getImageLogÉvènements

Récupérez les événements associés à une création d'image.

Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)
- [Corps de la réponse](#)
- [Exemple](#)

### Syntaxe de demande

```
GET /v3/images/custom/{imageId}/logstreams/{logStreamName}
{
```

```
"endTime": datetime,
"limit": float,
"nextToken": "string",
"region": "string",
"startFromHead": boolean,
"startTime": datetime
}
```

## Corps de la demande

### imageId

ID de l'image.

Type : chaîne

Obligatoire : oui

### logStreamName

Le nom du flux de log.

Type : chaîne

Obligatoire : oui

### endTime

Fin de la plage de temps, exprimée au format ISO 8601. Les événements dont l'horodatage est égal ou supérieur à cette heure ne sont pas inclus.

Type : datetime

Format : 2021-01-01T20:00:00Z

Obligatoire : non

### limite

Le nombre maximum d'événements de journal renvoyés. Si vous ne spécifiez aucune valeur, le maximum est le nombre d'événements de journal que peut contenir une taille de réponse de 1 Mo, jusqu'à 10 000 événements de journal.

Type : décimal

Obligatoire : non

## nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

Obligatoire : non

## region

C'est dans Région AWS cela que se trouve l'image.

Type : chaîne

Obligatoire : non

## startFromHead

Si ce paramètre est défini sur `true`, renvoie d'abord les premiers événements du journal. Si ce paramètre est défini sur `false`, renvoyez d'abord les derniers événements du journal. L'argument par défaut est `false`.

Type : valeur booléenne

Obligatoire : non

## startTime

Début de la plage horaire, exprimé au format ISO 8601. Les événements dont l'horodatage est égal à cette heure ou postérieur à cette heure sont inclus.

Type : datetime

Format : `2021-01-01T20:00:00Z`

Obligatoire : non

## Syntaxe de réponse

```
{
 "nextToken": "string",
 "prevToken": "string",
 "events": [
 {
```

```
 "timestamp": "2019-08-24T14:15:22Z",
 "message": "string"
 }
]
}
```

## Corps de la réponse

### événements

Liste des événements filtrés.

message

Le message de l'événement.

Type : chaîne

timestamp

Horodatage de l'événement.

Type : datetime

nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

PrevToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

## Exemple

### Python

Demande

```
$ get_image_log_events(image_id, log_stream_name=3.2.1/1)
```

Réponse 200



```
"events": [
 {
 "message": "ExecuteBash: STARTED EXECUTION",
 "timestamp": "2022-04-05T15:51:20.228Z"
 },
 {
 "message": "ExecuteBash: Created temporary directory: /tmp/1234567890abcdef0",
 "timestamp": "2022-04-05T15:51:20.228Z"
 },
 ...
]
```

## getImageStackÉvènements

Récupérez les événements associés à la pile pour la création d'une image.

### Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)
- [Corps de la réponse](#)
- [Exemple](#)

### Syntaxe de demande

```
GET /v3/images/custom/{imageId}/stackevents
{
 "nextToken": "string",
 "region": "string"
}
```

### Corps de la demande

#### imageId

ID de l'image.

Type : chaîne

Obligatoire : oui

nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

Obligatoire : non

region

Le dans Région AWS lequel se trouve l'image.

Type : chaîne

Obligatoire : non

## Syntaxe de réponse

```
{
 "nextToken": "string",
 "events": [
 {
 "stackId": "string",
 "eventId": "string",
 "stackName": "string",
 "logicalResourceId": "string",
 "physicalResourceId": "string",
 "resourceType": "string",
 "timestamp": "2019-08-24T14:15:22Z",
 "resourceStatus": "CREATE_IN_PROGRESS",
 "resourceStatusReason": "string",
 "resourceProperties": "string",
 "clientRequestToken": "string"
 }
]
}
```

## Corps de la réponse

événements

Liste des événements filtrés.

## clientRequestToken

Le jeton transmis à l'action qui a généré cet événement.

Type : chaîne

## eventId

L'identifiant unique de cet événement.

Type : chaîne

## logicalResourceId

Nom logique de la ressource spécifiée dans le modèle.

Type : chaîne

## physicalResourceId

Le nom ou l'identifiant unique associé à l'instance physique de la ressource.

Type : chaîne

## Propriétés des ressources

Blob des propriétés utilisées pour créer la ressource.

Type : chaîne

## État des ressources

Statut de la ressource.

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE | DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE | DELETE\_SKIPPED | UPDATE\_IN\_PROGRESS | UPDATE\_FAILED | UPDATE\_COMPLETE | IMPORT\_FAILED | IMPORT\_COMPLETE | IMPORT\_IN\_PROGRESS | IMPORT\_ROLLBACK\_IN\_PROGRESS | IMPORT\_ROLLBACK\_FAILED | IMPORT\_ROLLBACK\_COMPLETE

## resourceStatusReason

Message de réussite ou d'échec associé à la ressource.

Type : chaîne

## resourceType

Type de ressource.

Type : chaîne

## ID de pile

Le nom d'identification unique de l'instance de la pile.

Type : chaîne

## Nom de la pile

Le nom associé à une pile.

Type : chaîne

## timestamp

Heure à laquelle le statut a été mis à jour.

Type : datetime

## nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

## Exemple

### Python

#### Demande

```
$ get_image_stack_events(image_id)
```

#### Réponse 200

```
{
 'events': [
 {
```

```

 'event_id': 'ParallelClusterImage-
CREATE_IN_PROGRESS-2022-03-30T23:26:33.499Z',
 'logical_resource_id': 'ParallelClusterImage',
 'physical_resource_id': 'arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-alinux2-image/3.2.1/1',
 'resource_properties': {
 "InfrastructureConfigurationArn": "arn:aws:imagebuilder:us-
east-1:123456789012:infrastructure-configuration/parallelclusterimage-6accc570-
b080-11ec-845e-0e2dc6386985",
 "ImageRecipeArn": "arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/
parallelclusterimage-alinux2-image/3.2.1",
 "DistributionConfigurationArn": "arn:aws:imagebuilder:us-
east-1:123456789012:distribution-configuration/parallelclusterimage-6accc570-
b080-11ec-845e-0e2dc6386985",
 "EnhancedImageMetadataEnabled": "false",
 "Tags": {
 "parallelcluster:image_name": "alinux2-
image", "parallelcluster:image_id": "alinux2-image"
 }
 },
 'resource_status': 'CREATE_IN_PROGRESS',
 'resource_status_reason': 'Resource creation Initiated',
 'resource_type': 'AWS::ImageBuilder::Image',
 'stack_id': 'arn:aws:cloudformation:us-east-1:123456789012:stack/alinux2-
image/6accc570-b080-11ec-845e-0e2dc6386985',
 'stack_name': 'alinux2-image',
 'timestamp': datetime.datetime(2022, 3, 30, 23, 26, 33, 499000,
tzinfo=tzlocal())
 },
 ...
]
}

```

## Clusters de listes

Récupérez la liste des clusters existants.

### Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)

- [Corps de la réponse](#)
- [Exemple](#)

## Syntaxe de demande

```
GET /v3/clusters
{
 "clusterStatus": "string",
 "nextToken": "string",
 "region": "string"
}
```

## Corps de la demande

### État du cluster

Filtrez par statut du cluster. La valeur par défaut est tous les clusters.

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | UPDATE\_IN\_PROGRESS |  
UPDATE\_COMPLETE | UPDATE\_FAILED

Obligatoire : non

### nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

Obligatoire : non

### region

Le Région AWS des clusters.

Type : chaîne

Obligatoire : non

## Syntaxe de réponse

```
{
 "nextToken": "string",
 "clusters": [
 {
 "clusterName": "string",
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "clusterStatus": "CREATE_IN_PROGRESS",
 "scheduler": {
 "type": "string",
 "metadata": {
 "name": "string",
 "version": "string"
 }
 }
 }
]
}
```

## Corps de la réponse

### clusters

#### cloudformationStackArn

Le nom de ressource Amazon (ARN) de la CloudFormation pile principale.

Type : chaîne

#### cloudformationStackStatus

L'état de la CloudFormation pile.

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS

| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

Nom du cluster

Nom du cluster.

Type : chaîne

État du cluster

État du cluster.

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

planificateur

métadonnées

Les métadonnées du planificateur.

name

Nom du planificateur.

Type : chaîne

version

Version du planificateur.

Type : chaîne

type

Type de planificateur.

Type : chaîne

region

Le dans Région AWS lequel le cluster est créé.

Type : chaîne



## version

AWS ParallelCluster Version utilisée pour créer le cluster.

Type : chaîne

## nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

## Exemple

### Python

#### Demande

```
$ list_clusters()
```

#### Réponse 200

```
{
 'clusters':
 [
 {
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/cluster_name_3x/16b49540-ae5-11ec-8e18-0ac1d712b241',
 'cloudformation_stack_status': 'CREATE_COMPLETE',
 'cluster_name': 'cluster_name_3x',
 'cluster_status': 'CREATE_COMPLETE',
 'region': 'us-east-1',
 'version': '3.2.1'
 },
 ...
]
}
```

## listClusterLogStreams

Récupérez la liste des flux de journaux associés à un cluster.

## Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)
- [Corps de la réponse](#)
- [Exemple](#)

## Syntaxe de demande

```
GET /v3/clusters/{clusterName}/logstreams
{
 "filters": ["string"],
 "nextToken": "string",
 "region": "string"
}
```

## Corps de la demande

### Nom du cluster

Nom du cluster.

Type : chaîne

Obligatoire : oui

### filtres

Filtrez les flux de journaux.

Les filtres acceptés sont les suivants :

- private-dns-name: forme abrégée du nom DNS privé de l'instance (par exemple ip-10-0-0-101).
- type de nœud : valeur valide :. HeadNode

Type : Tableau de chaînes unique

Format : Name=a,Values=1 Name=b,Values=2,3

Obligatoire : non

## nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

Obligatoire : non

## region

Le dans Région AWS lequel se trouve le cluster.

Type : chaîne

Obligatoire : non

## Syntaxe de réponse

```
{
 "nextToken": "string",
 "logStreams": [
 {
 "logStreamName": "string",
 "creationTime": "2019-08-24T14:15:22Z",
 "firstEventTimestamp": "2019-08-24T14:15:22Z",
 "lastEventTimestamp": "2019-08-24T14:15:22Z",
 "lastIngestionTime": "2019-08-24T14:15:22Z",
 "uploadSequenceToken": "string",
 "logStreamArn": "string"
 }
]
}
```

## Corps de la réponse

### LogStreams

Liste des flux de journaux.

#### creationTime

Heure à laquelle le flux a été créé.

Type : datetime

firstEventTimestamp

Heure du premier événement du stream.

Type : datetime

lastEventTimestamp

Heure du dernier événement du stream. La lastEventTime valeur est mise à jour sur une base de cohérence éventuelle. Il est généralement mis à jour moins d'une heure après l'ingestion, mais dans de rares cas, cela peut prendre plus de temps.

Type : datetime

lastIngestionTime

Heure de la dernière ingestion.

Type : datetime

logStreamArn

Le nom de ressource Amazon (ARN) du flux de log.

Type : chaîne

logStreamName

Nom du flux de log.

Type : chaîne

uploadSequenceToken

Le jeton de séquence.

Type : chaîne

nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

## Exemple

### Python

#### Demande

```
$ list_cluster_log_streams(cluster_name_3x)
```

#### Réponse 200

```
{
 'log_streams': [
 {
 'creation_time': datetime.datetime(2022, 3, 30, 14, 7, 34, 354000,
tzinfo=tzlocal()),
 'first_event_timestamp': datetime.datetime(2022, 3, 30, 14, 6, 41, 444000,
tzinfo=tzlocal()),
 'last_event_timestamp': datetime.datetime(2022, 3, 30, 14, 25, 55, 462000,
tzinfo=tzlocal()),
 'last_ingestion_time': datetime.datetime(2022, 3, 30, 14, 49, 50, 62000,
tzinfo=tzlocal()),
 'log_stream_arn': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/
parallelcluster/cluster_name_3x:log-stream:ip-192-0-2-26.i-abcdef01234567890.cfn-
init',
 'log_stream_name': 'ip-192-0-2-26.i-abcdef01234567890.cfn-init',
 ...
 'upload_sequence_token': '####'
 },
 ...
]
}
```

## listImageLogStreams

Récupérez la liste des flux de journaux associés à une image.

### Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)

- [Corps de la réponse](#)
- [Exemple](#)

## Syntaxe de demande

```
GET /v3/images/custom/{imageId}/logstreams
{
 "nextToken": "string",
 "region": "string"
}
```

## Corps de la demande

### imageId

ID de l'image.

Type : chaîne

Obligatoire : oui

### nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

Obligatoire : non

### region

Le dans Région AWS lequel se trouve l'image.

Type : chaîne

Obligatoire : non

## Syntaxe de réponse

```
{
 "nextToken": "string",
```

```
"logStreams": [
 {
 "logStreamName": "string",
 "creationTime": "2019-08-24T14:15:22Z",
 "firstEventTimestamp": "2019-08-24T14:15:22Z",
 "lastEventTimestamp": "2019-08-24T14:15:22Z",
 "lastIngestionTime": "2019-08-24T14:15:22Z",
 "uploadSequenceToken": "string",
 "logStreamArn": "string"
 }
]
}
```

## Corps de la réponse

### LogStreams

Liste des flux de journaux.

#### creationTime

Heure à laquelle le flux a été créé.

Type : datetime

#### firstEventTimestamp

Heure du premier événement dans le stream.

Type : datetime

#### lastEventTimestamp

Heure du dernier événement du stream. La lastEventTime valeur est mise à jour sur une base de cohérence éventuelle. Il est généralement mis à jour moins d'une heure après l'ingestion, mais dans de rares cas, cela peut prendre plus de temps.

Type : datetime

#### lastIngestionTime

Heure de la dernière ingestion.

Type : datetime

## logStreamArn

Le nom de ressource Amazon (ARN) du flux de log.

Type : chaîne

## logStreamName

Nom du flux de journaux.

Type : chaîne

## uploadSequenceToken

Le jeton de séquence.

Type : chaîne

## jeton suivant

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

## Exemple

### Python

#### Demande

```
$ list_image_log_streams(custom-image-id)
```

#### Réponse 200

```
{
 'log_streams': [
 {
 'creation_time': datetime.datetime(2022, 3, 29, 20, 29, 24, 875000,
tzinfo=tzlocal()),
 'first_event_timestamp': datetime.datetime(2022, 3, 29, 20, 29, 24, 775000,
tzinfo=tzlocal()),
 'last_event_timestamp': datetime.datetime(2022, 3, 29, 20, 38, 23, 944000,
tzinfo=tzlocal()),
```



```
 'last_ingestion_time': datetime.datetime(2022, 3, 29, 20, 51, 56, 26000,
 tzinfo=tzlocal()),
 'log_stream_arn': 'arn:aws:logs:us-east-1:123456789012:log-group:/aws/
imagebuilder/ParallelClusterImage-alinux2-image:log-stream:3.2.1/1',
 'log_stream_name': '3.2.1/1',
 'upload_sequence_token': '####'
 },
 ...
]
}
```

## Images de la liste

Récupérez la liste des images personnalisées existantes.

### Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)
- [Corps de la réponse](#)
- [Exemple](#)

## Syntaxe de demande

```
GET /images/custom
{
 "imageStatus": "string",
 "nextToken": "string",
 "region": "string"
}
```

## Corps de la demande

### État de l'image

Filtrez les images en fonction du statut indiqué.

Type : chaîne

Valeurs valides : AVAILABLE | PENDING | FAILED

Obligatoire : oui

nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

Obligatoire : non

region

Ces Région AWS images sont dedans.

Type : chaîne

Obligatoire : non

## Syntaxe de réponse

```
{
 "nextToken": "string",
 "images": [
 {
 "imageId": "string",
 "ec2AmiInfo": {
 "amiId": "string"
 },
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "imageBuildStatus": "BUILD_IN_PROGRESS",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS"
 }
]
}
```

## Corps de la réponse

images

Une liste d'images.

## cloudformationStackArn

Le nom de ressource Amazon (ARN) de la CloudFormation pile principale.

Type : chaîne

## cloudformationStackStatus

L'état de la CloudFormation pile.

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

## ec2 AmilInfo

### ami\_id

L'ID de l'AMI Amazon EC2.

Type : chaîne

## imageBuildStatus

État de création de l'image.

Valeurs valides : BUILD\_IN\_PROGRESS | BUILD\_FAILED | BUILD\_COMPLETE |  
DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE

Type : chaîne

## imageId

ID de l'image.

Type : chaîne

## region

Le dans Région AWS lequel l'image est créée.

Type : chaîne

version

AWS ParallelCluster Version utilisée pour créer l'image.

Type : chaîne

nextToken

Un jeton utilisé pour les requêtes paginées.

Type : chaîne

## Exemple

Python

Demande

```
$ list_images("AVAILABLE")
```

Réponse 200

```
{
 'images': [
 {
 'ec2_ami_info': {
 'ami_id': 'ami-abcdef01234567890'
 },
 'image_build_status': 'BUILD_COMPLETE',
 'image_id': 'custom-image',
 'region': 'us-east-1',
 'version': '3.2.1'
 }
]
}
```

## listOfficialImages

Récupérez la liste des images AWS ParallelCluster officielles.

## Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)
- [Corps de la réponse](#)
- [Exemple](#)

## Syntaxe de demande

```
GET /v3/images/official
{
 "architecture": "string",
 "os": "string",
 "region": "string"
}
```

## Corps de la demande

### application

Filtrer par architecture. Par défaut, il n'y a aucun filtrage.

Type : chaîne

Valeurs valides : x86\_64 | arm64

Obligatoire : non

### os

Filtrer par distribution du système d'exploitation. Par défaut, il n'y a aucun filtrage.

Type : chaîne

Valeurs valides : alinux2 | centos7 | ubuntu2204 | ubuntu2004 | rhel8

Obligatoire : non

### region

Les Région AWS images officielles sont répertoriées dans.

Type : chaîne

Obligatoire : non

## Syntaxe de réponse

```
{
 "images": [
 {
 "architecture": "string",
 "amiId": "string",
 "name": "string",
 "os": "string",
 "version": "string"
 }
]
}
```

## Corps de la réponse

### images

#### AMID

ID de l'AMI.

Type : chaîne

#### application

L'architecture de l'AMI.

Type : chaîne

#### name

Le nom de l'AMI.

Type : chaîne

#### os

Le système d'exploitation AMI.

Type : chaîne

## version

La AWS ParallelCluster version.

Type : chaîne

## Exemple

### Python

#### Demande

```
$ list_official_images()
```

#### Réponse 200

```
{
 'images': [
 {
 'ami_id': 'ami-015cfefb4e0d6306b2',
 'architecture': 'x86_64',
 'name': 'aws-parallelcluster-3.2.1-ubuntu-2004-lts-hvm-x86_64-202202261505 '
 '2022-02-26T15-08-34.759Z',
 'os': 'ubuntu2004',
 'version': '3.2.1'
 },
 ...
]
}
```

## Mettre à jour le cluster

Mettez à jour le cluster.

### Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)
- [Corps de la réponse](#)

- [Exemple](#)

## Syntaxe de demande

```
PUT /v3/clusters/{clusterName}
{
 "clusterConfiguration": "string",
 "dryrun": boolean,
 "forceUpdate": boolean,
 "region": "string",
 "suppressValidators": "string",
 "validationFailureLevel": "string"
}
```

## Corps de la demande

### Configuration du cluster

La configuration du cluster sous forme de document YAML.

Obligatoire : oui

### Nom du cluster

Nom du cluster.

Type : chaîne

Obligatoire : oui

### run à sec

Si ce paramètre est défini sur `true`, effectuez uniquement la validation des demandes sans créer de ressource. Utilisez ce paramètre pour valider la configuration du cluster et les exigences de mise à jour. L'argument par défaut est `false`.

Type : valeur booléenne

Obligatoire : non

### ForceUpdate

Si ce paramètre est défini sur `true`, ignorez les erreurs de validation de mise à jour et forcez la mise à jour. L'argument par défaut est `false`.



Type : valeur booléenne

Obligatoire : non

region

Le dans Région AWS lequel se trouve le cluster.

Type : chaîne

Obligatoire : non

Supprimer les validateurs

Identifie un ou plusieurs validateurs de configuration à supprimer.

Type : chaîne

Format : (ALL | type:[A-Za-z0-9]+)

Obligatoire : non

Exemples de valeurs valides :currentValue,requestedValue, message

validationFailureLevel

Le niveau de validation minimal pour entraîner l'échec de la mise à jour.

Type : chaîne

Valeurs valides : INFO | WARNING | ERROR

Obligatoire : non

## Syntaxe de réponse

```
{
 "cluster": {
 "clusterName": "string",
 "region": "string",
 "version": "string",
 "cloudformationStackArn": "string",
 "cloudformationStackStatus": "UPDATE_IN_PROGRESS",
 "clusterStatus": "UPDATE_IN_PROGRESS",
 "scheduler": {
```

```
 "type": "string",
 "metadata": {
 "name": "string",
 "version": "string"
 }
 },
 "validationMessages": [
 {
 "id": "string",
 "type": "string",
 "level": "INFO",
 "message": "string"
 }
],
 "changeSet": [
 {
 "parameter": "string",
 "currentValue": "string",
 "requestedValue": "string"
 }
]
}
```

## Corps de la réponse

### changeSet

Modification définie pour la mise à jour du cluster.

#### Valeur actuelle

La valeur actuelle du paramètre à mettre à jour.

Type : chaîne

#### paramètre

Paramètre à mettre à jour.

Type : chaîne

#### Valeur demandée

La valeur demandée pour le paramètre à mettre à jour.

Type : chaîne

cluster

cloudformationStackArn

Le nom de ressource Amazon (ARN) de la CloudFormation pile principale.

Type : chaîne

cloudformationStackStatus

L'état de la CloudFormation pile.

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| ROLLBACK\_IN\_PROGRESS | ROLLBACK\_FAILED | ROLLBACK\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_COMPLETE | UPDATE\_ROLLBACK\_IN\_PROGRESS |  
UPDATE\_ROLLBACK\_FAILED | UPDATE\_ROLLBACK\_COMPLETE\_CLEANUP\_IN\_PROGRESS  
| UPDATE\_ROLLBACK\_COMPLETE

Nom du cluster

Nom du cluster.

Type : chaîne

État du cluster

État du cluster.

Type : chaîne

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE  
| DELETE\_IN\_PROGRESS | DELETE\_FAILED | DELETE\_COMPLETE |  
UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

region

Le dans Région AWS lequel le cluster est créé.

Type : chaîne

## planificateur

### métadonnées

Les métadonnées du planificateur.

#### name

Nom du planificateur.

Type : chaîne

#### version

Version du planificateur.

Type : chaîne

#### type

Type de planificateur.

Type : chaîne

#### version

AWS ParallelCluster version utilisée pour créer le cluster.

Type : chaîne

## Messages de validation

Liste des messages dont le niveau de validation est inférieur à `validationFailureLevel`. La liste des messages est collectée lors de la validation de la configuration.

#### id

L'ID du validateur.

Type : chaîne

#### level

Le niveau de validation.

Type : chaîne

Valeurs valides : INFO | WARNING | ERROR

## message

Le message de validation.

Type : chaîne

## type

Type du validateur.

Type : chaîne

## Exemple

### Python

Demande

```
$ update_cluster(cluster_name_3x, path/config-file.yaml)
```

Réponse 200

```
{
 'change_set': [
 {
 'current_value': '10',
 'parameter':
'Scheduling.SlurmQueues[queue1].ComputeResources[t2micro].MaxCount',
 'requested_value': '15'
 }
],
 'cluster': {
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-
east-1:123456789012:stack/test-api-cluster/e0462730-50b5-11ed-99a3-0a5ddc4a34c7',
 'cloudformation_stack_status': 'UPDATE_IN_PROGRESS',
 'cluster_name': 'cluster-3x',
 'cluster_status': 'UPDATE_IN_PROGRESS',
 'region': 'us-east-1',
 'scheduler': {
 'type': 'slurm'
 },
 'version': '3.2.1'
 }
}
```

```
}
```

## updateComputeFleet

Mettez à jour l'état du parc informatique du cluster.

### Rubriques

- [Syntaxe de demande](#)
- [Corps de la demande](#)
- [Syntaxe de réponse](#)
- [Corps de la réponse](#)
- [Exemple](#)

### Syntaxe de demande

```
PATCH /v3/clusters/{clusterName}/computefleet
{
 "status": "string",
 "region": "string"
}
```

### Corps de la demande

#### Nom du cluster

Nom du cluster.

Type : chaîne

Obligatoire : oui

#### status

État du parc informatique.

Type : chaîne

Valeurs valides : START\_REQUESTED | STOP\_REQUESTED | ENABLED | DISABLED

Obligatoire : oui

region

Le dans Région AWS lequel se trouve le cluster.

Type : chaîne

Obligatoire : non

## Syntaxe de réponse

```
{
 "status": "START_REQUESTED",
 "lastStatusUpdateTime": "2019-08-24T14:15:22Z"
}
```

## Corps de la réponse

status

État du parc informatique.

Type : chaîne

Valeurs valides : START\_REQUESTED | STARTING | RUNNING | PROTECTED | STOP\_REQUESTED | STOPPING | STOPPED | UNKNOWN | ENABLED | DISABLED

lastStatusUpdatedHeure

L'horodatage qui représente l'heure de la dernière mise à jour du statut.

Type : datetime

## Exemple

Python

Demande

```
$ update_compute_fleet(cluster_name_3x, "START_REQUESTED")
```

## Réponse 200

```
{
 'last_status_updated_time': datetime.datetime(2022, 3, 28, 22, 27, 14,
tzinfo=tzlocal()),
 'status': 'START_REQUESTED'
}
```

## AWS ParallelClusterAPI de bibliothèque Python

Commençant parAWS ParallelClusterversion 3.5.0, vous pouvez accéderAWS ParallelClusteravec leAWS ParallelClusterbibliothèque Python. Vous pouvez accéder auAWS ParallelClusterbibliothèque dans votrepclusterenvironnement ou depuis l'intérieur d'unAWS Lambdatemps d'exécution. Découvrez comment accéder auAWS ParallelClusterAPI à l'aide duAWS ParallelClusterbibliothèque Python. LeAWS ParallelClusterLa bibliothèque Python offre les mêmes fonctionnalités queAWS ParallelClusterL'API tient ses promesses.

LeAWS ParallelClusterLes opérations et paramètres de la bibliothèque Python reflètent ceux des paramètres de l'API lorsqu'ils sont convertis ensnake\_casesans majuscules.

### Rubriques

- [AWS ParallelClusterAutorisation de bibliothèque Python](#)
- [Installez leAWS ParallelClusterbibliothèque Python](#)
- [Opérations de l'API du cluster](#)
- [Opérations de l'API du parc de calcul](#)
- [Opérations de journalisation des clusters et des piles](#)
- [Opérations de l'API Image](#)
- [Opérations de journalisation des images et des piles](#)
- [Exemple](#)
- [AWS Lambdapour leAWS ParallelClusterbibliothèque Python](#)

## AWS ParallelClusterAutorisation de bibliothèque Python

Spécifiez les informations d'identification en utilisant l'une des méthodes standard valides pour boto3. Pour plus d'informations, consultez le[documentation boto3](#).



## Installez leAWS ParallelClusterbibliothèque Python

1. Installer `pcclusterCLI` version 3.5.0 ou ultérieure en suivant les instructions données dans [Con AWS ParallelCluster figuration](#).
2. Importez le `pccluster` module et commencez à utiliser la bibliothèque, comme indiqué dans l'exemple suivant :

```
import pcluster.lib as pc
pc.create_cluster(cluster_name="mycluster", cluster_configuration="config.yaml")
```

## Opérations de l'API du cluster

### Rubriques

- [list\\_clusters](#)
- [create\\_cluster](#)
- [delete\\_cluster](#)
- [describe\\_cluster](#)
- [update\\_cluster](#)

## list\_clusters

```
list_clusters(region, next_token, cluster_status)
```

Récupérez la liste des clusters existants.

Paramètres :

### **region**

Répertorie les clusters déployés sur un site donné Région AWS.

### **next\_token**

Le jeton à utiliser pour les requêtes paginées.

### **cluster\_status**

Filtre en fonction de l'état du cluster. Par défaut, tous les clusters sont répertoriés.

Valeurs valides : CREATE\_IN\_PROGRESS | CREATE\_FAILED | CREATE\_COMPLETE | DELETE\_IN\_PROGRESS | DELETE\_FAILED | UPDATE\_IN\_PROGRESS | UPDATE\_COMPLETE | UPDATE\_FAILED

## create\_cluster

```
create_cluster(cluster_name, cluster_configuration, region, suppress_validators, validation_failure_level, dry_run, rollback_on_failure, wait)
```

Créez un cluster dans une région donnée.

Paramètres :

### **cluster\_name** (obligatoire)

Le nom du cluster.

### **cluster\_configuration** (obligatoire)

La configuration du cluster en tant que type de données Python.

### **region**

Le clusterRégion AWS.

### **suppress\_validators**

Identifie un ou plusieurs validateurs de configuration de cluster à supprimer.

Format : (ALL | type:[A-Za-z0-9]+)

### **validation\_failure\_level**

Le niveau de validation minimal qui entraîne l'échec de la création du cluster. La valeur par défaut est ERROR.

Valeurs valides : INFO | WARNING | ERROR.

### **dry\_run**

Effectue la validation de la demande sans créer de ressources. Vous pouvez l'utiliser pour valider la configuration du cluster. La valeur par défaut est False.

## **rollback\_on\_failure**

Si défini sur `True`, AWS ParallelCluster lance automatiquement une restauration de la pile du cluster en cas de défaillance. La valeur par défaut est `True`.

## **wait**

Si défini sur `True`, AWS ParallelCluster attend la fin de l'opération. La valeur par défaut est `False`.

## **delete\_cluster**

```
delete_cluster(cluster_name, region, wait)
```

Supprime un cluster dans une région donnée.

Paramètres :

### **cluster\_name** (obligatoire)

Le nom du cluster.

### **region**

Le cluster Région AWS.

### **wait**

Si défini sur `True`, attend la fin de l'opération. La valeur par défaut est `False`.

## **describe\_cluster**

```
describe_cluster(cluster_name, region)
```

Obtenez des informations détaillées sur un cluster existant.

Paramètres :

### **cluster\_name** (obligatoire)

Le nom du cluster.

### **region**

Le cluster Région AWS.

## update\_cluster

```
update_cluster(cluster_name, cluster_configuration, suppress_validators,
validation_failure_level, region, force_update, dry_run, wait)
```

Mettez à jour un cluster dans une région donnée.

Paramètres :

### **cluster\_name** (obligatoire)

Le nom du cluster.

### **cluster\_configuration** (obligatoire)

La configuration du cluster en tant que type de données Python.

### **suppress\_validators**

Identifie un ou plusieurs validateurs de configuration de cluster à supprimer.

Format : (ALL | type:[A-Za-z0-9]+)

### **validation\_failure\_level**

Le niveau de validation minimal qui entraîne l'échec de la mise à jour du cluster. La valeur par défaut est ERROR.

Valeurs valides : INFO | WARNING | ERROR

### **region**

Le clusterRégion AWS.

### **dry\_run**

Effectue la validation de la demande sans créer ni mettre à jour de ressources. Vous pouvez l'utiliser pour valider la configuration du cluster. La valeur par défaut est False.

### **force\_update**

Si défini sur True, force la mise à jour en ignorant les erreurs de validation des mises à jour. La valeur par défaut est False.

### **wait**

Si défini sur True, attend la fin de l'opération. La valeur par défaut est False.

# Opérations de l'API du parc de calcul

## Rubriques

- [describe\\_compute\\_fleet](#)
- [update\\_compute\\_fleet](#)
- [delete\\_cluster\\_instances](#)
- [describe\\_cluster\\_instances](#)

## describe\_compute\_fleet

```
describe_compute_fleet(cluster_name, region)
```

Décrivez l'état d'un parc informatique de cluster pour un cluster donné.

Paramètres :

**cluster\_name** (obligatoire)

Le nom du cluster.

**region**

Décrit l'état du parc informatique d'un cluster déployé sur un point donné Région AWS.

## update\_compute\_fleet

```
update_compute_fleet(cluster_name, status, region)
```

Mettez à jour l'état du parc informatique du cluster.

Paramètres :

**cluster\_name** (obligatoire)

Le nom du cluster.

**status** (obligatoire)

État vers lequel effectuer la mise à jour.

Valeurs valides: START\_REQUESTED | STOP\_REQUESTED | ENABLED | DISABLED

## **region**

Le clusterRégion AWS.

## **delete\_cluster\_instances**

```
delete_cluster_instances(cluster_name, region, force)
```

Supprime un cluster dans une région donnée.

Paramètres :

### **cluster\_name** (obligatoire)

Le nom du cluster.

## **region**

Le clusterRégion AWS.

## **force**

Si défini sur `True`, force la suppression lorsque le cluster avec le paramètre donné `cluster_name` n'est pas trouvé. La valeur par défaut est `False`.

## **describe\_cluster\_instances**

```
describe_cluster_instances(cluster_name, region, next_token, node_type, queue_name)
```

Décrivez les instances d'un cluster.

Paramètres :

### **cluster\_name** (obligatoire)

Le nom du cluster.

## **region**

Le clusterRégion AWS.

**next\_token**

Le jeton à utiliser pour les requêtes paginées.

**node\_type**

Filtre les instances par `node_type`.

Valeurs valides : `HeadNode` | `ComputeNode`

**queue\_name**

Filtre les instances par nom de file d'attente.

## Opérations de journalisation des clusters et des piles

Rubriques

- [list\\_cluster\\_log\\_streams](#)
- [get\\_cluster\\_log\\_events](#)
- [get\\_cluster\\_stack\\_events](#)

### **list\_cluster\_log\_streams**

```
list_cluster_log_streams(cluster_name, region, filters, next_token)
```

Répertoriez les flux de journaux pour un cluster donné.

Paramètres :

**cluster\_name** (obligatoire)

Le nom du cluster.

**region**

Le cluster Région AWS.

**filters**

Filtre les flux de journaux du cluster.

Format : `'Name=a,Values=1 Name=b,Values=2,3'`

Filtres acceptés :

**code-dns-name**

Forme abrégée du nom DNS privé de l'instance ; par exemple, `ip-10-0-0-101`.

**type de nœud**

Type de nœud.

Valeurs valides : `HeadNode`

**next\_token**

Le jeton à utiliser pour les requêtes paginées.

## **get\_cluster\_log\_events**

```
get_cluster_log_events(cluster_name, log_stream_name, region, next_token,
start_from_head, limit, start_time, end_time)
```

Obtenez les événements du journal pour un cluster et un flux de journaux donnés.

Paramètres :

**cluster\_name** (obligatoire)

Le nom du cluster.

**log\_stream\_name** (obligatoire)

Le nom du flux du journal.

**region**

Le clusterRégion AWS.

**next\_token**

Le jeton à utiliser pour les requêtes paginées.

**start\_from\_head**

Si défini sur `True`, AWS ParallelCluster renvoie les premiers événements du journal en premier. Si défini sur `False`, il renvoie d'abord les derniers événements du journal. La valeur par défaut est `False`.



## **limit**

Le nombre maximum d'événements de journal renvoyés. Si vous ne spécifiez aucune valeur, le maximum est le nombre de journaux pouvant correspondre à une taille de réponse de 1 Mo, jusqu'à 10 000 événements de journal.

## **start\_time**

Début de la plage horaire pour les événements du journal, exprimée au format ISO 8601 ; par exemple, '2021-01-01T20:00:00Z'. Les événements dont l'horodatage est égal ou supérieur à cette heure sont inclus.

## **end\_time**

Fin de la plage de temps pour les événements du journal, exprimée au format ISO 8601 ; par exemple, '2021-01-01T20:00:00Z'. Les événements dont l'horodatage est égal ou supérieur à cette heure ne sont pas inclus.

## **get\_cluster\_stack\_events**

```
get_cluster_stack_events(cluster_name, region, next_token)
```

Obtenez des événements de pile pour un cluster donné.

Paramètres :

**cluster\_name** (obligatoire)

Le nom du cluster.

**region**

Le clusterRégion AWS.

**next\_token**

Le jeton à utiliser pour les requêtes paginées.

## Opérations de l'API Image

Rubriques

- [list\\_images](#)
- [build\\_image](#)
- [delete\\_image](#)
- [describe\\_image](#)

## list\_images

```
list_images(image_status, region, next_token)
```

Récupérez la liste des images existantes.

Paramètres :

**image\_status** (obligatoire)

Filtre en fonction du statut de l'image.

Valeurs valides : AVAILABLE | PENDING | FAILED

**region**

Répertorie les images créées dans une donnée Région AWS.

**next\_token**

Jeton à utiliser pour les demandes paginées.

## build\_image

```
build_image(image_configuration, image_id, suppress_validators,
validation_failure_level, dry_run, rollback_on_failure, region)
```

Créez une personnalisation AWS ParallelCluster image dans une région donnée.

Paramètres :

**image\_configuration** (obligatoire)

La configuration de l'image sous forme de données Python.

**image\_id** (obligatoire)

L'identifiant de l'image.

**suppress\_validators**

Identifie un ou plusieurs validateurs de configuration d'image à supprimer.

Format : (ALL | type:[A-Za-z0-9]+)

**validation\_failure\_level**

Le niveau de validation minimal qui entraîne l'échec de la création de l'image. La valeur par défaut est ERROR.

Valeurs valides : INFO | WARNING | ERROR

**dry\_run**

Si défini sur `True`, AWS ParallelCluster effectue la validation de la demande sans créer de ressources. Vous pouvez l'utiliser pour valider la configuration de l'image. La valeur par défaut est `False`.

**rollback\_on\_failure**

Si défini sur `True`, AWS ParallelCluster lance automatiquement une restauration de la pile d'images en cas d'échec. La valeur par défaut est `False`.

**region**

L'image Région AWS.

**delete\_image**

```
delete_image(image_id, region, force)
```

Supprime une image dans une région donnée.

Paramètres :

**image\_id** (obligatoire)

L'identifiant de l'image.

## region

L'imageRégion AWS.

## force

Si défini sur `True`, AWS ParallelCluster force la suppression si les instances utilisent l'AMI ou si l'AMI est partagée. La valeur par défaut est `False`.

## describe\_image

```
describe_image(image_id, region)
```

Obtenez des informations détaillées sur une image existante.

Paramètres :

### image\_id (obligatoire)

L'identifiant de l'image.

### region

L'imageRégion AWS.

## Opérations de journalisation des images et des piles

Rubriques

- [list\\_image\\_log\\_streams](#)
- [get\\_image\\_log\\_events](#)
- [get\\_image\\_stack\\_events](#)
- [list\\_official\\_images](#)

## list\_image\_log\_streams

```
list_image_log_streams(image_id, region, next_token)
```

Répertoriez les flux de log d'une image.

Paramètres :

**image\_id** (obligatoire)

L'identifiant de l'image.

**region**

L'imageRégion AWS.

**next\_token**

Le jeton à utiliser pour les requêtes paginées.

## get\_image\_log\_events

```
get_image_log_events(image_id, log_stream_name, region, next_token, start_from_head,
 limit, start_time, end_time)
```

Obtenez les événements du journal pour une image et un flux de journal donnés.

Paramètres :

**image\_id** (obligatoire)

L'identifiant de l'image.

**log\_stream\_name** (obligatoire)

Le nom du flux du journal.

**region**

L'imageRégion AWS.

**next\_token**

Le jeton à utiliser pour les requêtes paginées.

**start\_from\_head**

Si défini sur `True`, AWS ParallelCluster renvoie les premiers événements du journal en premier. Si défini sur `False`, il renvoie d'abord les derniers événements du journal. La valeur par défaut est `False`.

## limit

Le nombre maximum d'événements de journal renvoyés. Si vous ne spécifiez aucune valeur, le maximum est le nombre de journaux pouvant correspondre à une taille de réponse de 1 Mo, jusqu'à 10 000 événements de journal.

## start\_time

Début de la plage horaire pour les événements du journal, exprimée au format ISO 8601 ; par exemple, '2021-01-01T20:00:00Z'. Les événements dont l'horodatage est égal ou supérieur à cette heure sont inclus.

## end\_time

Fin de la plage de temps pour les événements du journal, exprimée au format ISO 8601 ; par exemple, '2021-01-01T20:00:00Z'. Les événements dont l'horodatage est égal ou supérieur à cette heure ne sont pas inclus.

## get\_image\_stack\_events

```
get_image_stack_events(image_id, region, next_token)
```

Obtenez des événements de pile pour une image donnée.

Paramètres :

### image\_id (obligatoire)

L'identifiant de l'image.

### region

L'imageRégion AWS.

### next\_token

Le jeton à utiliser pour les requêtes paginées.

## list\_official\_images

```
list_official_images(region,os, architecture)
```

Récupérez la liste des officielsAWS ParallelClusterimages.

Paramètres :

### **region**

L'imageRégion AWS.

### **os**

Filtres par distribution du système d'exploitation. Par défaut, il n'y a aucun filtrage.

### **architecture**

Filtres par architecture. Par défaut, il n'y a aucun filtrage.

## Exemple

Rubriques

- [Créer un cluster](#)

### Créer un cluster

Lorsque vous exécutez l'exemple de script suivant, avec les entrées données stockées dans votre environnement, vous créez un cluster. La configuration du cluster est créée en tant que type de données Python basé sur[documentation de configuration du cluster](#).

```
import os
import pprint
import pcluster.lib as pc
pp = pprint.PrettyPrinter()

HEAD_NODE_SUBNET = os.environ["HEAD_NODE_SUBNET"]
COMPUTE_NODE_SUBNET = os.environ["HEAD_NODE_SUBNET"]
KEY_NAME = os.environ["KEY_NAME"]
CONFIG = {'Image': {'Os': 'alinux2'},
 'HeadNode': {'InstanceType': 't2.large',
 'Networking': {'SubnetId': HEAD_NODE_SUBNET},
 'Ssh': {'KeyName': KEY_NAME}},
 'Scheduling': {'Scheduler': 'slurm',
 'SlurmQueues':
```

```
[{'Name': 'queue0',
 'ComputeResources':
 [{'Name': 'queue0-i0', 'InstanceType': 't2.micro',
 'MinCount': 0, 'MaxCount': 10}],
 'Networking': {'SubnetIds': [COMPUTE_NODE_SUBNET]}]}
```

```
pp.pprint(pc.create_cluster(cluster_name="mycluster", cluster_configuration=CONFIG))
```

Sortie :

```
{'cluster': {'cloudformationStackArn': 'arn:aws:cloudformation:us-
east-2:123456789012:stack/mycluster/00000000-aaaa-1111-999-000000000000',
 'cloudformationStackStatus': 'CREATE_IN_PROGRESS',
 'clusterName': 'mycluster',
 'clusterStatus': 'CREATE_IN_PROGRESS',
 'region': 'us-east-2',
 'scheduler': {'type': 'slurm'},
 'version': '3.7.0'}}
```

## AWS Lambda pour le AWS ParallelCluster bibliothèque Python

Vous pouvez déployer une couche Lambda et un environnement d'exécution pour accéder au AWS ParallelCluster bibliothèque Python. Nous hébergeons AWS ParallelCluster fichiers zip que vous pouvez utiliser en saisissant le lien vers le fichier zip comme décrit dans les étapes suivantes. Lambda utilise les fichiers zip pour préparer l'environnement d'exécution afin de prendre en charge l'accès à la bibliothèque Python. Le AWS ParallelCluster La bibliothèque Python est ajoutée avec AWS ParallelCluster version 3.5.0. Vous ne pouvez utiliser la bibliothèque que pour les versions 3.5.0 et ultérieures.

L'URL du fichier zip hébergé est au format suivant : `s3://aws-region-id-aws-parallelcluster/parallelcluster/3.7.0/layers/aws-parallelcluster/lambda-layer.zip`

Commencez à accéder au AWS ParallelCluster bibliothèque Python avec AWS Lambda

Création d'une couche Lambda

1. Connectez-vous au AWS Management Console et naviguez jusqu'au AWS Lambda console.
2. Dans le volet de navigation, sélectionnez Couches, puis Créer une couche.



3. Entrez un nom pour votre couche et sélectionnez Charger un fichier depuis Amazon S3.
4. Entrez l'URL du fichier zip : `s3://aws-region-id-aws-parallelcluster/parallelcluster/3.7.0/layers/aws-parallelcluster/lambda-layer.zip`.
5. Pour Architectures compatibles, choisissez `lex86_64` architecture.
6. Pour Runtimes compatibles, choisissez le Python 3.9 temps d'exécution.
7. Sélectionnez Create (Créer).

### Utilisez votre couche Lambda

1. Dans le volet de navigation de la console Lambda, sélectionnez Fonctions, puis Créer une fonction.
2. Entrez un nom pour votre fonction.
3. Pour Durée d'exécution, choisissez le Python 3.9 temps d'exécution.
4. Pour Architecture, choisissez `lex86_64` architecture.
5. Sélectionnez Create function (Créer une fonction).
6. Une fois la fonction créée, choisissez Couches et sélectionnez Ajouter une couche.
7. Sélectionnez Couches personnalisées et choisissez la couche que vous avez créée lors des étapes précédentes.
8. Choisissez la version de la couche.
9. Choisissez Add (Ajouter).
10. Votre Lambda a besoin d'autorisations pour gérer les clusters créés avec AWS ParallelCluster. Créez un rôle Lambda avec les autorisations répertoriées dans [Politique AWS ParallelCluster pcluster utilisateur de base](#).

Vous pouvez désormais accéder AWS ParallelCluster à partir de la bibliothèque Python, comme décrit dans [AWS ParallelCluster API de bibliothèque Python](#).

# Fonctionnement d'AWS ParallelCluster

AWS ParallelCluster a été conçu non seulement comme un moyen de gérer les clusters, mais aussi comme une référence sur la manière d'utiliser les AWS services pour créer votre environnement HPC.

## Rubriques

- [AWS ParallelCluster processus](#)
- [AWS services utilisés par AWS ParallelCluster](#)
- [AWS ParallelCluster Répertoires internes](#)

## AWS ParallelCluster processus

Cette section s'applique aux clusters déployés avec Slurm. Lorsqu'il est utilisé avec ce planificateur, il AWS ParallelCluster gère le provisionnement et la suppression des nœuds de calcul en interagissant avec le planificateur de tâches sous-jacent.

Pour les clusters HPC basés sur AWS Batch, AWS ParallelCluster s'appuie sur les fonctionnalités fournies par le AWS Batch pour la gestion des nœuds de calcul.

## **clustermgtd**

Les tâches suivantes sont effectuées par le démon de gestion du cluster.

- Nettoyage de partitions inactives
- Gestion des Slurm réservations et des nœuds associés aux blocs de capacité (voir section suivante)
- Gestion de la capacité statique : assurez-vous que la capacité statique est toujours active et saine
- Synchronisez le planificateur avec Amazon EC2.
- Nettoyage d'une instance orpheline
- Restaurez l'état du nœud du planificateur lors de la résiliation d'Amazon EC2 survenant en dehors du flux de travail de suspension
- Gestion des instances Amazon EC2 défectueuse (échec des contrôles de santé Amazon EC2)
- Gestion des événements de maintenance planifiés
- Gestion des nœuds du planificateur défaillante (échec des contrôles de santé du planificateur)

## Gestion des Slurm réservations et des nœuds associés aux Capacity Blocks

ParallelCluster prend en charge les réservations de capacité à la demande (ODCR) et les blocs de capacité pour le Machine Learning (CB). Contrairement à l'ODCR, le CB peut avoir une future heure de début et est limité dans le temps.

Clustermgtd recherche en boucle les nœuds défectueux et met fin à toutes les instances Amazon EC2 inactives, en les remplaçant par de nouvelles instances s'il s'agit de nœuds statiques.

ParallelCluster gère différemment les nœuds statiques associés aux blocs de capacité. AWS ParallelCluster crée un cluster même si le CB n'est pas encore actif, et les instances sont lancées automatiquement une fois que le CB est actif.

Les Slurm nœuds correspondant aux ressources de calcul associées aux CB qui ne sont pas encore actifs sont maintenus en maintenance jusqu'à ce que l'heure de début des CB soit atteinte. Slurm les nœuds resteront dans un état de réservation/maintenance associé à l'utilisateur Slurm administrateur, ce qui signifie qu'ils peuvent accepter des tâches, mais les tâches resteront en attente jusqu'à ce que la Slurm réservation soit supprimée.

Clustermgtd créera ou supprimera automatiquement des Slurm réservations, mettant les nœuds CB associés en maintenance en fonction de l'état du CB. Lorsque CB sera actif, la Slurm réservation sera supprimée, les nœuds démarreront et seront disponibles pour les tâches en attente ou pour les soumissions de nouvelles offres d'emploi.

Lorsque l'heure de fin du CB est atteinte, les nœuds repassent à l'état de réservation/maintenance. Il appartient aux utilisateurs de soumettre/de mettre en attente les tâches dans une nouvelle file d'attente/ressource de calcul lorsque le CB n'est plus actif et que les instances sont résiliées.

### **clusterstatusmgtd**

Le démon de gestion de l'état du cluster gère la mise à jour de l'état du parc informatique. Chaque minute, il récupère l'état du parc enregistré dans une table DynamoDB et gère toute demande STOP/START.

### **computemgtd**

Les processus Compute Management Daemon (`computemgtd`) s'exécutent sur chacun des nœuds de calcul du cluster. Toutes les cinq (5) minutes, le démon de gestion du calcul confirme que le nœud principal est joignable et qu'il est sain. Si cinq (5) minutes s'écoulent pendant lesquelles le nœud principal ne peut pas être atteint ou n'est pas sain, le nœud de calcul est arrêté.

# AWS services utilisés par AWS ParallelCluster

Les services Amazon Web Services (AWS) suivants sont utilisés par AWS ParallelCluster.

## Rubriques

- [Amazon API Gateway](#)
- [AWS Batch](#)
- [AWS CloudFormation](#)
- [Amazon CloudWatch](#)
- [CloudWatch Événements Amazon](#)
- [Amazon CloudWatch Logs](#)
- [AWS CodeBuild](#)
- [Amazon DynamoDB](#)
- [Amazon Elastic Block Store](#)
- [Amazon Elastic Compute Cloud](#)
- [Amazon Elastic Container Registry](#)
- [Amazon EFS](#)
- [Amazon FSx pour Lustre](#)
- [Amazon FSx pour ONTAP NetApp](#)
- [Amazon FSx pour OpenZFS](#)
- [AWS Identity and Access Management](#)
- [AWS Lambda](#)
- [Amazon RDS](#)
- [Amazon Route 53](#)
- [Amazon Simple Notification Service](#)
- [Amazon Simple Storage Service](#)
- [Amazon VPC](#)
- [Elastic Fabric Adapter](#)
- [EC2 Image Builder](#)
- [NICE DCV](#)

## Amazon API Gateway

Amazon API Gateway est un AWS service de création, de publication, de maintenance, de surveillance et de sécurisation des API REST, HTTP et des WebSocket API à n'importe quelle échelle

AWS ParallelCluster utilise API Gateway pour héberger l' AWS ParallelCluster API.

Pour plus d'informations sur AWS Batch, consultez <https://aws.amazon.com/api-gateway/> et <https://docs.aws.amazon.com/apigateway/>.

## AWS Batch

AWS Batch est un service de planification de tâches AWS géré. Il fournit de manière dynamique la quantité et le type optimaux de ressources de calcul (par exemple, des instances optimisées pour le processeur ou la mémoire) dans AWS Batch les clusters. Ces ressources sont provisionnées en fonction des exigences spécifiques de vos tâches par lots, y compris les exigences en matière de volume. Grâce à AWS Batch cela, vous n'avez pas besoin d'installer ou de gérer des logiciels de traitement par lots ou des clusters de serveurs supplémentaires pour exécuter vos tâches efficacement.

AWS Batch est utilisé uniquement avec les AWS Batch clusters.

Pour plus d'informations sur AWS Batch, consultez <https://aws.amazon.com/batch/> et <https://docs.aws.amazon.com/batch/>.

## AWS CloudFormation

AWS CloudFormation est un infrastructure-as-code service qui fournit un langage commun pour modéliser AWS et fournir des ressources d'applications tierces dans votre environnement cloud. C'est le principal service utilisé par AWS ParallelCluster. Chaque cluster AWS ParallelCluster est représenté sous la forme d'une pile, et toutes les ressources requises par chaque cluster sont définies dans le AWS ParallelCluster AWS CloudFormation modèle. Dans la plupart des cas, les commandes AWS ParallelCluster CLI correspondent directement aux commandes de AWS CloudFormation pile, telles que les commandes de création, de mise à jour et de suppression. Les instances lancées au sein d'un cluster envoient des appels HTTPS au AWS CloudFormation point de terminaison dans Région AWS lequel le cluster est lancé.

Pour plus d'informations sur AWS CloudFormation, consultez <https://aws.amazon.com/cloudformation/> et <https://docs.aws.amazon.com/cloudformation/>.

## Amazon CloudWatch

Amazon CloudWatch (CloudWatch) est un service de surveillance et d'observabilité qui vous fournit des données et des informations exploitables. Ces informations peuvent être utilisées pour surveiller vos applications, répondre aux changements de performances et aux exceptions de service, et optimiser l'utilisation des ressources. In AWS ParallelCluster, CloudWatch est utilisé comme tableau de bord, pour surveiller et enregistrer les étapes de création de l'image Docker et le résultat des AWS Batch tâches.

Avant AWS ParallelCluster la version 2.10.0, CloudWatch il n'était utilisé qu'avec des AWS Batch clusters.

Pour plus d'informations sur CloudWatch, consultez <https://aws.amazon.com/cloudwatch/> et <https://docs.aws.amazon.com/cloudwatch/>.

## CloudWatch Événements Amazon

Amazon CloudWatch Events (CloudWatch Events) fournit un flux quasi en temps réel d'événements système décrivant les modifications apportées aux ressources Amazon Web Services (AWS). A l'aide de règles simples et rapidement configurées, vous pouvez faire correspondre des événements et les acheminer vers un ou plusieurs flux ou une ou plusieurs fonctions cibles. Dans AWS ParallelCluster, CloudWatch Events est utilisé pour les AWS Batch tâches.

Pour plus d'informations sur CloudWatch les événements, consultez <https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-cwe-now-eb>.

## Amazon CloudWatch Logs

Amazon CloudWatch Logs (CloudWatch Logs) est l'une des fonctionnalités principales d'Amazon CloudWatch. Vous pouvez l'utiliser pour surveiller, stocker, afficher et rechercher dans les fichiers journaux de nombreux composants utilisés par AWS ParallelCluster.

Avant AWS ParallelCluster la version 2.6.0, CloudWatch Logs n'était utilisé qu'avec des AWS Batch clusters.

Pour plus d'informations, consultez [Intégration avec Amazon CloudWatch Logs avec Amazon Logs](#).

## AWS CodeBuild

AWS CodeBuild (CodeBuild) est un service AWS géré d'intégration continue qui respecte le code source, exécute des tests et produit des progiciels prêts à être déployés. In AWS ParallelCluster,

CodeBuild est utilisé pour créer automatiquement et de manière transparente des images Docker lors de la création de clusters.

CodeBuild est utilisé uniquement avec les AWS Batch clusters.

Pour plus d'informations sur CodeBuild, consultez <https://aws.amazon.com/codebuild/> et <https://docs.aws.amazon.com/codebuild/>.

## Amazon DynamoDB

Amazon DynamoDB (DynamoDB) est un service de base de données NoSQL rapide et flexible. Il est utilisé pour stocker les informations d'état minimales du cluster. Le nœud principal suit les instances provisionnées dans une table DynamoDB.

DynamoDB n'est pas utilisé avec les clusters. AWS Batch

Pour plus d'informations sur DynamoDB, consultez <https://aws.amazon.com/dynamodb/> et <https://docs.aws.amazon.com/dynamodb/>.

## Amazon Elastic Block Store

Amazon Elastic Block Store (Amazon EBS) est un service de stockage par blocs à hautes performances qui fournit un stockage persistant pour les volumes partagés. Tous les paramètres Amazon EBS peuvent être transmis via la configuration. Les volumes Amazon EBS peuvent être initialisés vides ou à partir d'un instantané Amazon EBS existant.

Pour plus d'informations sur Amazon EBS, consultez <https://aws.amazon.com/ebs/> et <https://docs.aws.amazon.com/ebs/>.

## Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud (Amazon EC2) fournit la capacité de calcul pour. AWS ParallelCluster La tête et les nœuds de calcul sont des instances Amazon EC2. N'importe quel type d'instance qui prend en charge HVM peut être sélectionné. Les nœuds de tête et de calcul peuvent être de types d'instances différents. De plus, si plusieurs files d'attente sont utilisées, certains ou tous les nœuds de calcul peuvent également être lancés en tant qu'instance ponctuelle. Les volumes de stockage d'instance disponibles sur les instances sont montés sous la forme d'un volume LVM agrégé par bandes.

Pour plus d'informations sur Amazon EC2, consultez <https://aws.amazon.com/ec2/> et <https://docs.aws.amazon.com/ec2/>.

## Amazon Elastic Container Registry

Amazon Elastic Container Registry (Amazon ECR) est un registre de conteneurs Docker entièrement géré qui facilite le stockage, la gestion et le déploiement d'images de conteneurs Docker. Dans AWS ParallelCluster, Amazon ECR stocke les images Docker créées lors de la création de clusters. Les images Docker sont ensuite utilisées AWS Batch pour exécuter les conteneurs pour les tâches soumises.

Amazon ECR est utilisé uniquement avec des AWS Batch clusters.

Pour plus d'informations, consultez <https://aws.amazon.com/ecr/> et <https://docs.aws.amazon.com/ecr/>.

## Amazon EFS

Amazon Elastic File System (Amazon EFS) fournit un système de fichiers NFS élastique simple, évolutif et entièrement géré à utiliser avec les AWS Cloud services et les ressources sur site. Amazon EFS est utilisé lorsqu'ils [EfsSettings](#) sont spécifiés. Support pour Amazon EFS a été ajouté dans la AWS ParallelCluster version 2.1.0.

Pour plus d'informations sur Amazon EFS, consultez <https://aws.amazon.com/efs/> et <https://docs.aws.amazon.com/efs/>.

## Amazon FSx pour Lustre

FSx for Lustre fournit un système de fichiers performant qui utilise le système de fichiers open source Lustre. FSx for Lustre est utilisé lorsqu'ils sont [Propriétés de FsxLustreSettings](#) spécifiés. Support pour FSx for Lustre a été ajouté AWS ParallelCluster dans la version 2.2.1.

Pour plus d'informations sur FSx for Lustre, [consultez](https://aws.amazon.com/fsx/lustre/) <https://aws.amazon.com/fsx/lustre/> [et](https://docs.aws.amazon.com/fsx/) <https://docs.aws.amazon.com/fsx/>.

## Amazon FSx pour ONTAP NetApp

FSx for ONTAP fournit un système de stockage partagé entièrement géré basé sur le célèbre système NetApp de fichiers ONTAP. FSx pour ONTAP est utilisé lorsqu'ils [Propriétés de FsxOntapSettings](#) sont spécifiés. Support de FSx pour ONTAP a été ajouté dans AWS ParallelCluster la version 3.2.0.

Pour plus d'informations sur FSx for ONTAP, consultez <https://aws.amazon.com/fsx/netapp-ontap/> et <https://docs.aws.amazon.com/fsx/>.



## Amazon FSx pour OpenZFS

FSx pour OpenZFS fournit un système de stockage partagé entièrement géré basé sur le célèbre système de fichiers OpenZFS. FSx pour OpenZFS est utilisé lorsqu'ils sont spécifiés. [Propriétés de FsxOpenZfsSettings](#) Support de FSx pour OpenZFS a été ajouté dans la version 3.2.0. AWS ParallelCluster

[Pour plus d'informations sur FSx pour OpenZFS, consultez https://aws.amazon.com/fsx/opensfs/](https://aws.amazon.com/fsx/opensfs/) et <https://docs.aws.amazon.com/fsx/>.

## AWS Identity and Access Management

AWS Identity and Access Management (IAM) est utilisé dans le cadre AWS ParallelCluster pour fournir le rôle IAM le moins privilégié à Amazon EC2 pour l'instance spécifique à chaque cluster individuel. AWS ParallelCluster les instances n'ont accès qu'aux appels d'API spécifiques nécessaires au déploiement et à la gestion du cluster.

Avec les AWS Batch clusters, des rôles IAM sont également créés pour les composants impliqués dans le processus de création d'images Docker lors de la création de clusters. Ces composants incluent les fonctions Lambda qui sont autorisées à ajouter et à supprimer des images Docker dans et depuis le référentiel Amazon ECR. Elles incluent également les fonctions permettant de supprimer le compartiment Amazon S3 créé pour le cluster et le CodeBuild projet. Il existe également des rôles pour les AWS Batch ressources, les instances et les tâches.

Pour plus d'informations sur IAM, consultez <https://aws.amazon.com/iam/> et <https://docs.aws.amazon.com/iam/>.

## AWS Lambda

AWS Lambda (Lambda) exécute les fonctions qui orchestrent la création d'images Docker. Lambda gère également le nettoyage des ressources de cluster personnalisées, telles que les images Docker stockées dans le référentiel Amazon ECR et sur Amazon S3.

[Pour plus d'informations sur Lambda, consultez https://aws.amazon.com/lambda/](https://aws.amazon.com/lambda/) et <https://docs.aws.amazon.com/lambda/>.

## Amazon RDS

Amazon Relational Database Service (Amazon RDS) est un service Web qui facilite la configuration, l'exploitation et le dimensionnement d'une base de données relationnelle dans le AWS cloud.

AWS ParallelCluster utilise Amazon RDS pour AWS Batch et. Slurm

Pour plus d'informations sur Amazon RDS, consultez <https://aws.amazon.com/rds/> et <https://docs.aws.amazon.com/rds/>.

## Amazon Route 53

Amazon Route 53 (Route 53) est utilisé pour créer des zones hébergées avec des noms d'hôtes et des noms de domaine complets pour chacun des nœuds de calcul.

Pour plus d'informations sur la Route 53, consultez <https://aws.amazon.com/route53/> et <https://docs.aws.amazon.com/route53/>.

## Amazon Simple Notification Service

(Amazon SNS) est un service géré qui fournit des messages des éditeurs aux abonnés (également appelés producteurs et consommateurs).

AWS ParallelCluster utilise Amazon SNS pour l'hébergement d'API.

[Pour plus d'informations sur Amazon SNS, consultez https://aws.amazon.com/sns/](https://aws.amazon.com/sns/) et <https://docs.aws.amazon.com/sns/>.

## Amazon Simple Storage Service

Amazon Simple Storage Service (Amazon S3) AWS ParallelCluster stocke les modèles situés dans chacun d'entre eux. Région AWS AWS ParallelCluster peut être configuré pour permettre aux outils CLI/SDK d'utiliser Amazon S3.

AWS ParallelCluster crée également un compartiment Amazon S3 dans votre répertoire Compte AWS pour stocker les ressources utilisées par vos clusters, telles que le fichier de configuration du cluster. AWS ParallelCluster gère un compartiment Amazon S3 dans chaque compartiment dans Région AWS lequel vous créez des clusters.

Lorsque vous utilisez un AWS Batch cluster, un compartiment Amazon S3 de votre compte est utilisé pour stocker les données associées. Par exemple, le bucket stocke les artefacts créés lorsqu'une image Docker et des scripts sont créés à partir de tâches soumises.

Pour plus d'informations, consultez <https://aws.amazon.com/s3/> et <https://docs.aws.amazon.com/s3/>.

## Amazon VPC

Un Amazon VPC définit un réseau utilisé par les nœuds de votre cluster.

Pour plus d'informations sur Amazon VPC, consultez <https://aws.amazon.com/vpc/> et <https://docs.aws.amazon.com/vpc/>.

## Elastic Fabric Adapter

Elastic Fabric Adapter (EFA) est une interface réseau destinée aux instances que les clients peuvent utiliser pour exécuter des applications nécessitant des niveaux élevés de communications inter-nœuds à grande échelle. AWS

Pour plus d'informations sur EC2 Image Builder, [consultez](https://aws.amazon.com/hpc/efa/) <https://aws.amazon.com/hpc/efa/>.

## EC2 Image Builder

EC2 Image Builder est un service entièrement AWS géré qui vous aide à automatiser la création, la gestion et le déploiement d'images personnalisées, sécurisées up-to-date et de serveur.

AWS ParallelCluster utilise Image Builder pour créer et gérer AWS ParallelCluster des images.

Pour plus d'informations sur EC2 Image Builder, [consultez](https://aws.amazon.com/image-builder/) <https://aws.amazon.com/image-builder/> [et https://docs.aws.amazon.com/imagebuilder/](https://docs.aws.amazon.com/imagebuilder/).

## NICE DCV

NICE DCV est un protocole d'affichage à distance à hautes performances qui fournit un moyen sécurisé de diffuser des postes de travail distants et des applications en streaming sur n'importe quel appareil, quelles que soient les conditions du réseau. NICE DCV est utilisé lorsque les [Dcv](#) paramètres [HeadNode Section](#)/sont spécifiés. Support pour NICE DCV a été ajouté dans la AWS ParallelCluster version 2.5.0.

Pour plus d'informations sur NICE DCV, consultez <https://aws.amazon.com/hpc/dcv/> et <https://docs.aws.amazon.com/dcv/>.

## AWS ParallelCluster Répertoires internes

Plusieurs annuaires internes sont AWS ParallelCluster utilisés pour partager des données au sein du cluster. Les répertoires suivants sont partagés entre le nœud principal, les nœuds de calcul et les nœuds de connexion :

`/opt/slurm`

`/opt/intel`

`/opt/parallelcluster/shared` (only with compute nodes)

`/opt/parallelcluster/shared_login_nodes` (only with login nodes)

`/home` (unless specified in `SharedStorage`)

#### Note

Par défaut, ces répertoires sont créés sur le volume EBS du nœud principal et partagés sous forme d'exportations NFS vers les nœuds de calcul et de connexion. À partir de la version AWS ParallelCluster 3.8, vous pouvez AWS ParallelCluster activer la création et la gestion d'un système de fichiers Amazon EFS pour héberger et partager ces répertoires en définissant le paramètre sur [SharedStorageTypeefs](#).

Lorsque le cluster prend de l'ampleur, les exportations NFS via le volume EBS peuvent entraîner des problèmes de performances. Grâce à EFS, vous pouvez éviter les exportations NFS à mesure que votre cluster s'agrandit et éviter les blocages de performances qui y sont associés.

# Didacticiels

Les didacticiels suivants vous montrent comment démarrer avec AWS ParallelCluster la version 3 et fournissent des conseils sur les meilleures pratiques pour certaines tâches courantes.

Lorsque vous utilisez l'interface de ligne de commande (CLI) ou l'API, vous ne payez que pour les AWS ressources créées lorsque vous créez ou mettez à jour AWS ParallelCluster des images et des clusters. Pour plus d'informations, consultez [AWS services utilisés par AWS ParallelCluster](#).

L' AWS ParallelCluster interface utilisateur repose sur une architecture sans serveur et vous pouvez l'utiliser dans la catégorie AWS Free Tier dans la plupart des cas. Pour plus d'informations, voir [AWS ParallelCluster Coûts d'interface utilisateur](#).

## Rubriques

- [Exécution de votre premier job sur AWS ParallelCluster](#)
- [Création d'une AWS ParallelCluster AMI personnalisée](#)
- [Intégration d'Active Directory](#)
- [Configuration du chiffrement du stockage partagé à l'aide d'une AWS KMS clé](#)
- [Exécution de tâches dans un cluster en mode de files d'attente multiples](#)
- [Utilisation de l' AWS ParallelCluster API](#)
- [Création d'un cluster avec Slurm comptabilité](#)
- [Création d'un cluster avec une Slurmdbd comptabilité externe](#)
- [Revenir à une version précédente d'un document AWS Systems Manager](#)
- [Création d'un cluster avec AWS CloudFormation](#)
- [Déploiement ParallelCluster d'une API avec Terraform](#)
- [Création d'un cluster avec Terraform](#)
- [Création d'une AMI personnalisée avec Terraform](#)
- [AWS ParallelCluster Intégration de l'interface utilisateur à Identity Center](#)

## Exécution de votre premier job sur AWS ParallelCluster

Ce didacticiel vous explique comment exécuter votre première tâche Hello World sur AWS ParallelCluster

Lorsque vous utilisez l'interface de ligne de commande (CLI) ou l'API, vous ne payez que pour les AWS ressources créées lorsque vous créez ou mettez à jour AWS ParallelCluster des images et des clusters. Pour plus d'informations, consultez [AWS services utilisés par AWS ParallelCluster](#).

L' AWS ParallelCluster interface utilisateur repose sur une architecture sans serveur et vous pouvez l'utiliser dans la catégorie AWS Free Tier dans la plupart des cas. Pour plus d'informations, consultez [AWS ParallelCluster Coûts d'interface utilisateur](#).

## Prérequis

- AWS ParallelCluster [est installé](#).
- Le AWS CLI [est installé et configuré](#).
- Vous possédez une paire de [clés Amazon EC2](#).
- Vous disposez d'un rôle IAM doté des [autorisations](#) requises pour exécuter la [pcluster](#) CLI.

## Vérification de votre installation

Tout d'abord, nous vérifions que AWS ParallelCluster c'est correctement installé et configuré, y compris la dépendance Node.js.

```
$ node --version
v16.8.0
$ pcluster version
{
 "version": "3.7.0"
}
```

Cela renvoie la version en cours d'exécution de AWS ParallelCluster.

## Création de votre premier cluster

Il est temps de créer votre premier cluster. Comme la charge de travail pour ce didacticiel n'est pas exigeante, nous pouvons utiliser la taille d'instance par défaut, à savoir `t2.micro`. (Pour les charges de travail en production, vous choisissez une taille d'instance qui répond le mieux à vos besoins.) Appelons votre cluster `hello-world`.

```
$ pcluster create-cluster \
 --cluster-name hello-world \
```

```
--cluster-configuration hello-world.yaml
```

### Note

Le Région AWS à utiliser doit être spécifié pour la plupart des `pcluster` commandes. S'il n'est pas spécifié dans la variable d'`AWS_DEFAULT_REGION` environnement ou dans le `region` paramètre dans la `[default]` section du `~/.aws/config` fichier, le `--region` paramètre doit être fourni sur la ligne de `pcluster` commande.

Si le résultat affiche un message concernant la configuration, vous devez exécuter ce qui suit pour configurer AWS ParallelCluster :

```
$ pcluster configure --config hello-world.yaml
```

Si la [pcluster create-cluster](#) commande aboutit, vous obtenez un résultat similaire à ce qui suit :

```
{
 "cluster": {
 "clusterName": "hello-world",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:xxx:stack/xxx",
 "region": "...",
 "version": "...",
 "clusterStatus": "CREATE_IN_PROGRESS"
 }
}
```

Vous surveillez la création du cluster à l'aide de :

```
$ pcluster describe-cluster --cluster-name hello-world
```

Les `clusterStatus` rapports « `CREATE_IN_PROGRESS` » lors de la création du cluster. Il `clusterStatus` passe à « `CREATE_COMPLETE` » lorsque le cluster est créé avec succès. La sortie nous fournit également la `publicIpAddress` fin `privateIpAddress` de notre nœud principal.

## Connexion à votre nœud principal

Utilisez votre fichier `pem` OpenSSH pour vous connecter à votre nœud principal.

```
$ pcluster ssh --cluster-name hello-world -i /path/to/keyfile.pem
```

Une fois connecté, exécutez la commande `sinfo` pour vérifier que vos nœuds de calcul sont créés et configurés.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
queue1* up infinite 10 idle~ queue1-dy-queue1t2micro-[1-10]
```

Le résultat indique que nous avons une file d'attente dans notre cluster, avec jusqu'à dix nœuds.

## Exécution de votre première tâche avec Slurm

Ensuite, nous créons une tâche qui dort pendant un certain temps, puis génère son propre nom d'hôte. Créez un fichier nommé `hellojob.sh` avec le contenu suivant :

```
#!/bin/bash
sleep 30
echo "Hello World from $(hostname)"
```

Ensuite, soumettez la tâche avec `sbatch`, et vérifiez qu'elle s'exécute.

```
$ sbatch hellojob.sh
Submitted batch job 2
```

Maintenant, vous pouvez afficher votre file d'attente et vérifier le statut de la tâche. Le provisionnement d'une nouvelle instance Amazon EC2 démarre en arrière-plan. Vous pouvez surveiller l'état des instances de cluster à l'aide de la `sinfo` commande.

```
$ squeue
 JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
 2 queue1 hellojob ec2-user CF 3:30 1 queue1-dy-
queue1t2micro-1
```

La sortie indique que le travail a été soumis à `queue1`. Attendez 30 secondes que la tâche se termine et exécutez `squeue` à nouveau.

```
$ squeue
```



| JOBID | PARTITION | NAME | USER | ST | TIME | NODES | NODELIST(REASON) |
|-------|-----------|------|------|----|------|-------|------------------|
|-------|-----------|------|------|----|------|-------|------------------|

Maintenant qu'il n'y a pas de tâche dans la file d'attente, nous pouvons rechercher la sortie dans notre répertoire actuel.

```
$ ls -l
total 8
-rw-rw-r-- 1 ec2-user ec2-user 57 Sep 1 14:25 hellojob.sh
-rw-rw-r-- 1 ec2-user ec2-user 43 Sep 1 14:30 slurm-2.out
```

Dans le résultat, nous voyons un fichier out « ». Nous pouvons voir les résultats de notre travail :

```
$ cat slurm-2.out
Hello World from queue1-dy-queue1t2micro-1
```

La sortie indique également que notre tâche a été exécutée avec succès sur l'instance queue1-dy-queue1t2micro-1.

Dans le cluster que vous venez de créer, seul le répertoire de base est partagé entre tous les nœuds du cluster.

Pour en savoir plus sur la création et l'utilisation de clusters, consultez [Bonnes pratiques](#).

Si votre application nécessite le partage de logiciels, de bibliothèques ou de données, envisagez les options suivantes :

- Créez une AMI personnalisée AWS ParallelCluster activée qui inclut votre logiciel, comme décrit dans [Création d'une AWS ParallelCluster AMI personnalisée](#).
- Utilisez l'[StorageSettings](#) option du fichier AWS ParallelCluster de configuration pour spécifier un système de fichiers partagé et stocker le logiciel installé dans l'emplacement de montage indiqué.
- [Actions de bootstrap personnalisées](#) À utiliser pour automatiser la procédure d'amorçage de chaque nœud de votre cluster.

## Création d'une AWS ParallelCluster AMI personnalisée

Lorsque vous utilisez l'interface de ligne de commande (CLI) ou l'API, vous ne payez que pour les AWS ressources créées lorsque vous créez ou mettez à jour AWS ParallelCluster des images et des clusters. Pour plus d'informations, consultez [AWS services utilisés par AWS ParallelCluster](#).

L' AWS ParallelCluster interface utilisateur repose sur une architecture sans serveur et vous pouvez l'utiliser dans la catégorie AWS Free Tier dans la plupart des cas. Pour plus d'informations, consultez [AWS ParallelCluster Coûts d'interface utilisateur](#).

### Important

Si vous créez une AMI personnalisée, vous devez répéter les étapes que vous avez utilisées pour créer votre AMI personnalisée à chaque nouvelle AWS ParallelCluster version.

Avant de poursuivre votre lecture, nous vous recommandons de consulter d'abord [Actions de bootstrap personnalisées](#) cette section. Déterminez si les modifications que vous souhaitez apporter peuvent être scriptées et prises en charge dans les AWS ParallelCluster versions futures.

Même si la création d'une AMI personnalisée n'est généralement pas idéale, il existe des scénarios spécifiques dans lesquels il AWS ParallelCluster est nécessaire de créer une AMI personnalisée pour. Ce didacticiel explique comment créer une AMI personnalisée pour ces scénarios.

### Prérequis

- AWS ParallelCluster [est installé](#).
- Le AWS CLI [est installé et configuré](#).
- Vous possédez une paire de [clés Amazon EC2](#).
- Vous disposez d'un rôle IAM avec les [autorisations](#) requises pour exécuter la [pcluster](#) CLI et créer des images.

## Comment personnaliser l' AWS ParallelCluster AMI

Il existe deux manières de créer une AWS ParallelCluster AMI personnalisée. L'une de ces deux méthodes consiste à créer une nouvelle AMI à l'aide de la AWS ParallelCluster CLI. Une autre méthode nécessite que vous apportiez des modifications manuelles pour créer une nouvelle AMI disponible sous votre Compte AWS.

## Créez une AWS ParallelCluster AMI personnalisée

Si vous disposez d'une AMI et d'un logiciel personnalisés, vous pouvez appliquer les modifications nécessaires par AWS ParallelCluster dessus. AWS ParallelCluster s'appuie sur le service EC2

Image Builder pour créer des AMI personnalisées. Pour plus d'informations, consultez le [guide de l'utilisateur d'Image Builder](#).

Points clés :

- Le processus dure environ 1 heure. Cette durée peut varier s'il y a un [Build](#)/supplémentaire [Components](#) à installer au moment de la construction.
- L'AMI est étiquetée avec les versions des principaux composants. Il s'agit notamment du noyau, du planificateur et du pilote [EFA](#). Un sous-ensemble des versions des composants est également indiqué dans la description de l'AMI.
- À partir de la AWS ParallelCluster version 3.0.0, un nouvel ensemble de commandes CLI peut être utilisé pour gérer le cycle de vie des images. Cela inclut [build-image](#), [list-images](#), [describe-image](#) et [delete-image](#).
- Cette méthode est reproductible. Vous pouvez le réexécuter pour maintenir les AMI à jour (par exemple, les mises à jour du système d'exploitation), puis les utiliser lorsque vous mettez à jour un cluster existant.

#### Note

Si vous utilisez cette méthode dans la partition AWS chinoise, vous risquez de rencontrer des erreurs réseau. Par exemple, ces erreurs peuvent apparaître dans la `pc1uster build-image` commande lorsqu'elle télécharge des packages depuis GitHub ou depuis un référentiel du système d'exploitation. Dans ce cas, nous vous recommandons d'utiliser l'une des méthodes alternatives suivantes :

1. Suivez l'[Modifier une AWS ParallelCluster AMI](#) approche qui contourne cette commande.
2. Créez l'image dans une autre partition ou région, par exemple, puis stockez-la et restaurez-la pour la déplacer vers la région de Chine. `us-east-1` Pour plus d'informations, consultez [Stocker et restaurer une AMI à l'aide de S3](#) dans le guide de l'utilisateur Amazon EC2.

Étapes :

1. Configurez vos Compte AWS informations d'identification afin que le AWS ParallelCluster client puisse appeler les opérations d' AWS API en votre nom. Pour obtenir la liste des

autorisations requises, consultez [AWS Identity and Access Management autorisations dans AWS ParallelCluster](#).

2. Créez un fichier de configuration d'image de construction de base. Pour ce faire, spécifiez le [InstanceType](#) à utiliser pour créer l'image et le [ParentImage](#). Ils sont utilisés comme point de départ pour créer l'AMI. Pour plus d'informations sur les paramètres de construction facultatifs, consultez la section [Configuration de l'image](#).

```
Build:
 InstanceType: <BUILD_INSTANCE_TYPE>
 ParentImage: <BASE_AMI_ID>
```

3. Utilisez la commande CLI [pcluster build-image](#) pour créer une AWS ParallelCluster AMI à partir de l'AMI que vous fournissez comme base.

```
$ pcluster build-image --image-id IMAGE_ID --image-configuration IMAGE_CONFIG.yaml --
region REGION
{
 "image": {
 "imageId": "IMAGE_ID",
 "imageBuildStatus": "BUILD_IN_PROGRESS",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
IMAGE_ID/abcd1234-ef56-gh78-ij90-1234abcd5678",
 "region": "us-east-1",
 "version": "3.7.0"
 }
}
```

#### Warning

`pcluster build-image` utilise le VPC par défaut. Si vous supprimez le VPC par défaut en utilisant AWS Control Tower ou AWS Landing Zone, l'ID de sous-réseau doit être spécifié dans le fichier de configuration de l'image. Pour plus d'informations, consultez [SubnetId](#).

Pour obtenir la liste des autres paramètres, consultez la page de référence des [pcluster build-image](#) commandes. Les résultats de la commande précédente sont les suivants :

- Une CloudFormation pile est créée en fonction de la configuration de l'image. La pile inclut toutes les ressources EC2 Image Builder requises pour le build.
  - Les ressources créées incluent les AWS ParallelCluster composants officiels d'Image Builder auxquels des composants Image Builder personnalisés peuvent être ajoutés. Pour savoir comment créer des composants personnalisés, consultez les [exemples d'AMI personnalisés présentés](#) dans l'atelier HPC pour les clients du secteur public.
  - EC2 Image Builder lance une instance de build, applique AWS ParallelCluster le livre de recettes, installe AWS ParallelCluster la pile logicielle et effectue les tâches de configuration nécessaires. Le AWS ParallelCluster livre de recettes est utilisé pour construire et démarrer. AWS ParallelCluster
  - L'instance est arrêtée et une nouvelle AMI est créée à partir de celle-ci.
  - Une autre instance est lancée à partir de l'AMI nouvellement créée. Pendant la phase de test, EC2 Image Builder exécute des tests définis dans les composants Image Builder.
  - Si la construction est réussie, la pile est supprimée. Si la construction échoue, la pile est conservée et disponible pour inspection.
4. Vous pouvez surveiller l'état du processus de génération en exécutant la commande suivante. Une fois la compilation terminée, vous pouvez l'exécuter pour récupérer l'ID d'AMI indiqué dans la réponse.

```
$ pcluster describe-image --image-id IMAGE_ID --region REGION

BEFORE COMPLETE
{
 "imageConfiguration": {
 "url": "https://parallelcluster-1234abcd5678efgh-v1-do-not-
delete.s3.amazonaws.com/parallelcluster/3.7.0/images/IMAGE_ID-abcd1234efgh5678/
configs/image-config.yaml?...",
 },
 "imageId": "IMAGE_ID",
 "imagebuilderImageStatus": "BUILDING",
 "imageBuildStatus": "BUILD_IN_PROGRESS",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/
IMAGE_ID/abcd1234-ef56-gh78-ij90-1234abcd5678",
 "region": "us-east-1",
 "version": "3.7.0",
 "cloudformationStackTags": [
 {
 "value": "3.7.0",
 "key": "parallelcluster:version"
```

```
 },
 {
 "value": "IMAGE_ID",
 "key": "parallelcluster:image_name"
 },
 ...
],
 "imageBuildLogsArn": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/
imagebuilder/ParallelClusterImage-IMAGE_ID",
 "cloudformationStackCreationTime": "2022-04-05T21:36:26.176Z"
}

AFTER COMPLETE
{
 "imageConfiguration": {
 "url": "https://parallelcluster-1234abcd5678efgh-v1-do-not-delete.s3.us-
east-1.amazonaws.com/parallelcluster/3.7.0/images/IMAGE_ID-abcd1234efgh5678/configs/
image-config.yaml?Signature=..."
 },
 "imageId": "IMAGE_ID",
 "imageBuildStatus": "BUILD_COMPLETE",
 "region": "us-east-1",
 "ec2AmiInfo": {
 "amiName": "IMAGE_ID 2022-04-05T21-39-24.020Z",
 "amiId": "ami-1234stuv5678wxyz",
 "description": "AWS ParallelCluster AMI for alinux2,
kernel-4.14.238-182.422.amzn2.x86_64, lustre-2.10.8-5.amzn2.x86_64,
efa-1.13.0-1.amzn2.x86_64, dcv-2021.1.10598-1.el7.x86_64, slurm-20-11-8-1",
 "state": "AVAILABLE",
 "tags": [
 {
 "value": "2021.3.11591-1.el7.x86_64",
 "key": "parallelcluster:dcv_version"
 },
 ...
],
 "architecture": "x86_64"
 },
 "version": "3.7.0"
}
```

5. Pour créer votre cluster, entrez l'ID de l'AMI dans le [CustomAmi](#) champ de configuration de votre cluster.

## Résolution des problèmes et surveillance du processus de création d'AMI

La création de l'image se termine en une heure environ. Vous pouvez surveiller le processus en exécutant la [pcluster describe-image](#) commande ou les commandes de récupération du journal.

```
$ pcluster describe-image --image-id IMAGE_ID --region REGION
```

La [build-image](#) commande crée une CloudFormation pile contenant toutes les ressources Amazon EC2 nécessaires à la création de l'image et lance le processus EC2 Image Builder.

Après avoir exécuté la [build-image](#) commande, il est possible de récupérer les événements de la CloudFormation pile en utilisant [pcluster get-image-stack-events](#). Vous pouvez filtrer les résultats à l'aide du `--query` paramètre pour voir les derniers événements. Pour plus d'informations, consultez la section [Filtrage AWS CLI de la sortie](#) dans le guide de AWS Command Line Interface l'utilisateur.

```
$ pcluster get-image-stack-events --image-id IMAGE_ID --region REGION --query
"events[0]"
{
 "eventId": "ParallelClusterImage-CREATE_IN_PROGRESS-2022-04-05T21:39:24.725Z",
 "physicalResourceId": "arn:aws:imagebuilder:us-east-1:123456789012:image/
parallelclusterimage-IMAGE_ID/3.7.0/1",
 "resourceStatus": "CREATE_IN_PROGRESS",
 "resourceStatusReason": "Resource creation Initiated",
 "resourceProperties": "{\"InfrastructureConfigurationArn\":
\\\"arn:aws:imagebuilder:us-east-1:123456789012:infrastructure-configuration/
parallelclusterimage-abcd1234-ef56-gh78-ij90-1234abcd5678\\\", \\\"ImageRecipeArn\\\":
\\\"arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/parallelclusterimage-
IMAGE_ID/3.7.0\\\", \\\"DistributionConfigurationArn\\\": \\\"arn:aws:imagebuilder:us-
east-1:123456789012:distribution-configuration/parallelclusterimage-abcd1234-ef56-
gh78-ij90-1234abcd5678\\\", \\\"Tags\\\": {\\\"parallelcluster:image_name\\\": \\\"IMAGE_ID\\\",
\\\"parallelcluster:image_id\\\": \\\"IMAGE_ID\\\"}}\",
 "stackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/IMAGE_ID/abcd1234-
ef56-gh78-ij90-1234abcd5678",
 "stackName": "IMAGE_ID",
 "logicalResourceId": "ParallelClusterImage",
 "resourceType": "AWS::ImageBuilder::Image",
 "timestamp": "2022-04-05T21:39:24.725Z"
}
```

Après environ 15 minutes, les événements de la pile apparaissent dans l'entrée du journal des événements liés à la création d'Image Builder. Vous pouvez désormais répertorier les flux de journaux d'images et suivre les étapes de l'Image Builder à l'aide `pcluster list-image-log-streams` des `pcluster get-image-log-events` commandes and.

```
$ pcluster list-image-log-streams --image-id IMAGE_ID --region REGION \
 --query 'logStreams[*].logStreamName'

"3.7.0/1"
]

$ pcluster get-image-log-events --image-id IMAGE_ID --region REGION \
 --log-stream-name 3.7.0/1 --limit 3
{
 "nextToken": "f/36295977202298886557255241372854078762600452615936671762",
 "prevToken": "b/36295977196879805474012299949460899222346900769983430672",
 "events": [
 {
 "message": "ExecuteBash: FINISHED EXECUTION",
 "timestamp": "2022-04-05T22:13:26.633Z"
 },
 {
 "message": "Document arn:aws:imagebuilder:us-east-1:123456789012:component/
parallelclusterimage-test-abcd1234-ef56-gh78-ij90-1234abcd5678/3.7.0/1",
 "timestamp": "2022-04-05T22:13:26.741Z"
 },
 {
 "message": "TOE has completed execution successfully",
 "timestamp": "2022-04-05T22:13:26.819Z"
 }
]
}
```

Continuez à vérifier la `describe-image` commande jusqu'à ce que vous voyiez le `BUILD_COMPLETE` statut.

```
$ pcluster describe-image --image-id IMAGE_ID --region REGION
{
 "imageConfiguration": {
 "url": "https://parallelcluster-1234abcd5678efgh-v1-do-not-delete.s3.us-
east-1.amazonaws.com/parallelcluster/3.7.0/images/IMAGE_ID-abcd1234efgh5678/configs/
image-config.yaml?Signature=..."
 }
}
```



```

},
"imageId": "IMAGE_ID",
"imageBuildStatus": "BUILD_COMPLETE",
"region": "us-east-1",
"ec2AmiInfo": {
 "amiName": "IMAGE_ID 2022-04-05T21-39-24.020Z",
 "amiId": "ami-1234stuv5678wxyz",
 "description": "AWS ParallelCluster AMI for alinux2,
kernel-4.14.238-182.422.amzn2.x86_64, lustre-2.10.8-5.amzn2.x86_64,
efa-1.13.0-1.amzn2.x86_64, dcv-2021.1.10598-1.el7.x86_64, slurm-20-11-8-1",
 "state": "AVAILABLE",
 "tags": [
 {
 "value": "2021.3.11591-1.el7.x86_64",
 "key": "parallelcluster:dcv_version"
 },
 ...
],
 "architecture": "x86_64"
},
"version": "3.7.0"
}

```

Si vous devez résoudre un problème de création d'AMI personnalisée, créez une archive des journaux d'images comme décrit dans les étapes suivantes.

Il est possible d'archiver les journaux dans un compartiment Amazon S3 ou dans un fichier local, selon le `--output` paramètre.

```

$ pcluster export-image-logs --image-id IMAGE_ID --region REGION \
--bucket BUCKET_NAME --bucket-prefix BUCKET_FOLDER
{
 "url": "https://BUCKET_NAME.s3.us-east-1.amazonaws.com/BUCKET-FOLDER/IMAGE_ID-
logs-202209071136.tar.gz?AWSAccessKeyId=..."
}

$ pcluster export-image-logs --image-id IMAGE_ID \
--region REGION --bucket BUCKET_NAME --bucket-prefix BUCKET_FOLDER --output-file /tmp/
archive.tar.gz
{
 "path": "/tmp/archive.tar.gz"
}

```

L'archive contient les flux de CloudWatch journaux liés au processus Image Builder et aux événements de la AWS CloudFormation pile. L'exécution de la commande peut prendre plusieurs minutes.

## Gestion des AMI personnalisées

À partir de la AWS ParallelCluster version 3.0.0, un nouvel ensemble de commandes a été ajouté à la CLI pour créer, surveiller et gérer le cycle de vie des images. Pour plus d'informations sur les commandes, consultez la section [commandes pcluster](#).

## Modifier une AWS ParallelCluster AMI

Cette méthode consiste à modifier une AWS ParallelCluster AMI officielle en y ajoutant une personnalisation. Les AWS ParallelCluster AMI de base sont mises à jour avec les nouvelles versions. Ces AMI possèdent tous les composants nécessaires AWS ParallelCluster pour fonctionner lorsqu'elles sont installées et configurées. Vous pouvez commencer par l'un d'entre eux comme base.

Points clés :

- Cette méthode est plus rapide que la [build-image](#) commande. Cependant, il s'agit d'un processus manuel qui ne se répète pas automatiquement.
- Avec cette méthode, vous n'avez pas accès aux commandes de récupération des journaux et de gestion du cycle de vie des images disponibles via la CLI.

Étapes :

New Amazon EC2 console

1. Trouvez l'AMI qui correspond à l'AMI spécifique Région AWS que vous utilisez. Pour le trouver, utilisez la [pcluster list-official-images](#) commande avec le `--region` paramètre pour sélectionner les `--architecture` paramètres spécifiques Région AWS `--os` et à filtrer en fonction de l'AMI souhaitée avec le système d'exploitation et l'architecture que vous souhaitez utiliser. À partir de la sortie, récupérez l'identifiant de l'image Amazon EC2.
2. [Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/ec2/.](https://console.aws.amazon.com/ec2/)
3. Dans le volet de navigation, choisissez Images, puis AMIs. Recherchez l'ID d'image EC2 récupéré, sélectionnez l'AMI, puis choisissez Launch instance from AMI.
4. Faites défiler la page vers le bas et choisissez votre type d'instance.

5. Choisissez votre paire de clés et lancez votre instance.
6. Connectez-vous à votre instance à l'aide de l'utilisateur du système d'exploitation et de votre SSH clé.
7. Personnalisez manuellement votre instance pour répondre à vos besoins.
8. Exécutez la commande suivante pour préparer votre instance pour la création d'AMI.

```
sudo /usr/local/sbin/ami_cleanup.sh
```

9. Dans la console, choisissez Instance state et Stop instance.

Accédez à Instances, choisissez la nouvelle instance, sélectionnez État de l'instance et Arrêtez l'instance.

10. [Créez une nouvelle AMI à partir de l'instance à l'aide de la console Amazon EC2 ou AWS CLI de la commande create-image.](#)

Depuis la console Amazon EC2

- a. Choisissez Instances dans le volet de navigation.
  - b. Choisissez l'instance que vous avez créée et modifiée.
  - c. Dans Actions, choisissez Image, puis Créer une image.
  - d. Choisissez Créer une image.
- 11 Entrez le nouvel ID AMI dans le [CustomAmi](#) champ de configuration de votre cluster et créez un cluster.

## Old Amazon EC2 console

1. Trouvez l' AWS ParallelCluster AMI qui correspond à l'AMI spécifique Région AWS que vous utilisez. Pour le trouver, vous pouvez utiliser la [pcluster list-official-images](#) commande avec le `--region` paramètre pour sélectionner les `--architecture` paramètres spécifiques Région AWS `--os` et à filtrer pour l'AMI souhaitée avec le système d'exploitation et l'architecture que vous souhaitez utiliser. À partir de la sortie, vous pouvez récupérer l'identifiant de l'image Amazon EC2.
2. [Connectez-vous à la console Amazon EC2 AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/ec2/.](https://console.aws.amazon.com/ec2/)
3. Dans le volet de navigation, choisissez Images, puis AMIs. Définissez le filtre pour les images publiques et recherchez l'ID d'image EC2 récupéré, sélectionnez l'AMI, puis choisissez Launch.

4. Choisissez votre type d'instance et sélectionnez Suivant : Configurer les détails de l'instance ou Vérifier et lancer pour lancer votre instance.
5. Choisissez Launch, sélectionnez votre paire de clés et lancez les instances.
6. Connectez-vous à votre instance à l'aide de l'utilisateur du système d'exploitation et de votre clé SSH. Pour plus d'informations, accédez à Instances, sélectionnez la nouvelle instance et Connect.
7. Personnalisez manuellement votre instance pour répondre à vos besoins.
8. Exécutez la commande suivante pour préparer votre instance pour la création d'une AMI :

```
sudo /usr/local/sbin/ami_cleanup.sh
```

9. Depuis la console Amazon EC2, choisissez Instances dans le volet de navigation, sélectionnez votre nouvelle instance et choisissez Actions, Instance State et Stop.
10. [Créez une nouvelle AMI à partir de l'instance à l'aide de la console Amazon EC2 ou AWS CLI de la commande create-image.](#)

Depuis la console Amazon EC2

- a. Choisissez Instances dans le volet de navigation.
  - b. Choisissez l'instance que vous avez créée et modifiée.
  - c. Dans Actions, choisissez Image, puis Créer une image.
  - d. Choisissez Créer une image.
- 11 Entrez le nouvel ID AMI dans le [CustomAmi](#) champ de configuration de votre cluster et créez un cluster.

## Intégration d'Active Directory

Dans ce didacticiel, vous allez créer un environnement multi-utilisateurs. Cet environnement inclut un AWS ParallelCluster qui est intégré à un AWS Managed Microsoft AD (Active Directory) à `corp.example.com`. Vous configurez un Admin utilisateur pour gérer le répertoire, un ReadOnly utilisateur pour lire le répertoire et un `user000` utilisateur pour se connecter au cluster. Vous pouvez utiliser le chemin automatique ou le chemin manuel pour créer les ressources réseau, un Active Directory (AD) et l'instance Amazon EC2 que vous utilisez pour configurer l'AD. Quel que soit le chemin, l'infrastructure que vous créez est préconfigurée pour s'intégrer à AWS ParallelCluster l'aide de l'une des méthodes suivantes :

- LDAPS avec vérification des certificats (recommandé comme option la plus sûre)
- LDAPS sans vérification du certificat
- LDAP

Le protocole LDAP en lui-même ne fournit pas de chiffrement. Pour garantir la transmission sécurisée d'informations potentiellement sensibles, nous vous recommandons vivement d'utiliser LDAPS (LDAP over TLS/SSL) pour les clusters intégrés à ADs. Pour plus d'informations, voir [Activer le protocole LDAPS côté serveur à l'aide](#) du AWS Directory Service guide d' AWS Managed Microsoft AD administration.

Après avoir créé ces ressources, configurez et créez votre cluster intégré à votre Active Directory (AD). Une fois le cluster créé, connectez-vous en tant qu'utilisateur que vous avez créé. Pour plus d'informations sur la configuration que vous créez dans ce didacticiel, consultez [Accès de plusieurs utilisateurs aux clusters](#) la section relative à la [DirectoryService](#) configuration.

Ce didacticiel explique comment créer un environnement prenant en charge l'accès de plusieurs utilisateurs aux clusters. Ce didacticiel n'explique pas comment créer et utiliser un AWS Directory Service AD. Les étapes que vous suivez pour configurer un AWS Managed Microsoft AD dans ce didacticiel sont fournies à des fins de test uniquement. Ils ne sont pas fournis pour remplacer la documentation officielle et les meilleures pratiques que vous pouvez trouver sur [AWS Managed Microsoft AD Simple AD](#) dans le Guide d'AWS Directory Service administration.

#### Note

Les mots de passe des utilisateurs de l'annuaire expirent conformément aux définitions des propriétés de la politique de mot de passe d'annuaire. Pour plus d'informations, consultez la section [Paramètres de stratégie pris en charge](#). Pour réinitialiser les mots de passe des annuaires avec AWS ParallelCluster, voir [Comment réinitialiser le mot de passe d'un utilisateur et les mots de passe expirés](#).

#### Note

Les adresses IP du contrôleur de domaine de l'annuaire peuvent changer en raison des modifications du contrôleur de domaine et de la maintenance de l'annuaire. Si vous avez choisi la méthode de création rapide automatisée pour créer l'infrastructure d'annuaire, vous devez aligner manuellement l'équilibreur de charge devant les contrôleurs d'annuaire lorsque

les adresses IP des répertoires changent. Si vous utilisez la méthode de création rapide, les adresses IP du répertoire ne sont pas automatiquement alignées avec les équilibres de charge.

Lorsque vous utilisez l'interface de ligne de commande (CLI) ou l'API, vous ne payez que pour les AWS ressources créées lorsque vous créez ou mettez à jour AWS ParallelCluster des images et des clusters. Pour plus d'informations, consultez [AWS services utilisés par AWS ParallelCluster](#).

L' AWS ParallelCluster interface utilisateur repose sur une architecture sans serveur et vous pouvez l'utiliser dans la catégorie AWS Free Tier dans la plupart des cas. Pour plus d'informations, consultez [AWS ParallelClusterCoûts d'interface utilisateur](#).

## Prérequis

- AWS ParallelCluster [est installé](#).
- Le AWS CLI [est installé et configuré](#).
- Vous possédez une paire de [clés Amazon EC2](#).
- Vous disposez d'un rôle IAM doté des [autorisations](#) requises pour exécuter la [pcluster](#) CLI.

Au fur et à mesure que vous parcourez le didacticiel *inputs highlighted in red*, remplacez *d-abcdef01234567890*, tels que *region-id* et, par vos propres noms et identifiants. Remplacez *0123456789012* par votre Compte AWS numéro.

## Étape 1 : Création de l'infrastructure AD

Choisissez l'onglet Automatisé pour créer l'infrastructure Active Directory (AD) à l'aide d'un modèle de création AWS CloudFormation rapide.

Choisissez l'onglet Manuel pour créer manuellement l'infrastructure AD.

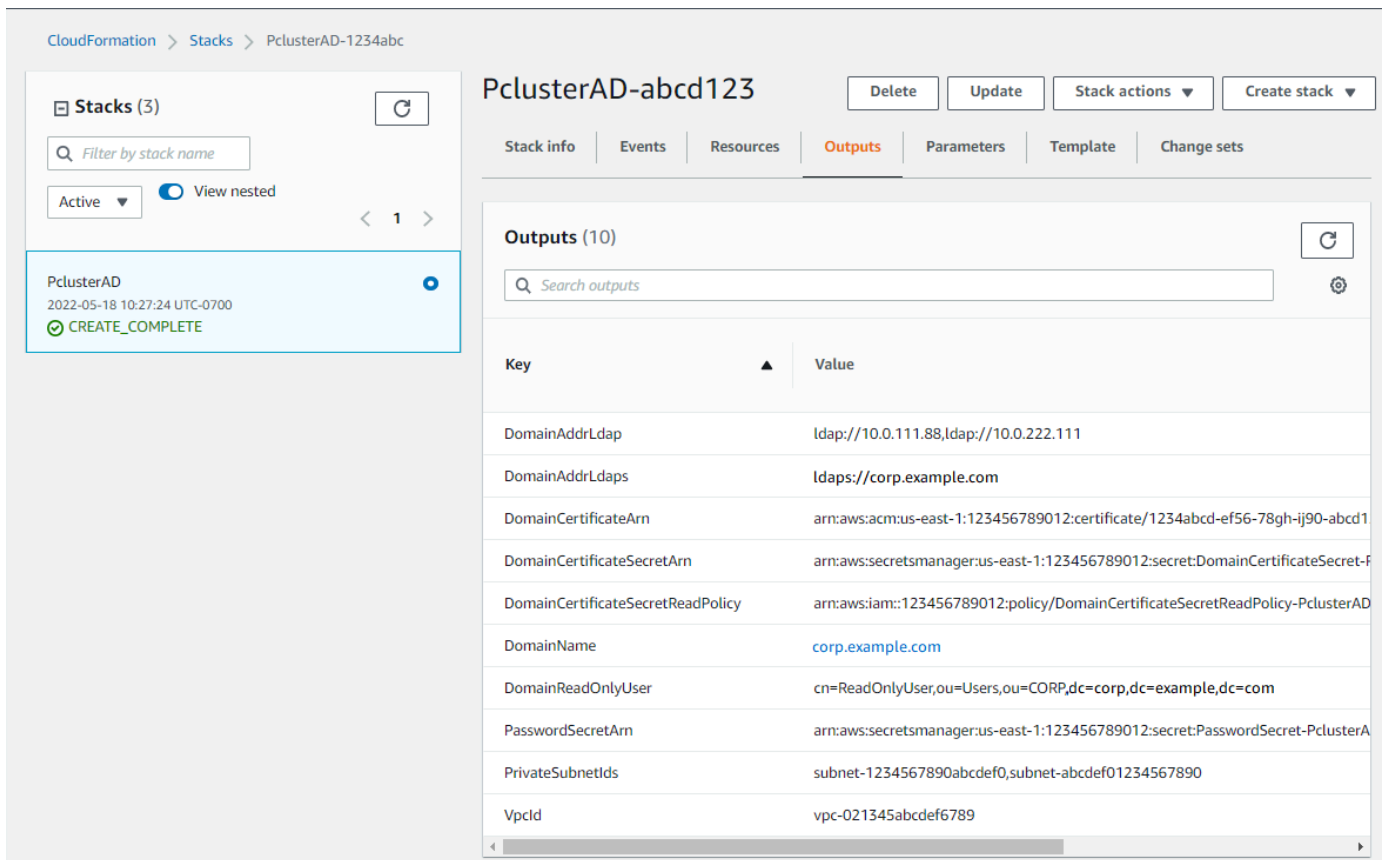
### Automatisé

1. Connectez-vous au AWS Management Console.
2. Ouvrez [CloudFormation Quick Create \(région us-east-1\)](#) pour créer les ressources suivantes dans la console : CloudFormation

- Un VPC avec deux sous-réseaux et un routage pour un accès public, si aucun VPC n'est spécifié.
  - Un AWS Managed Microsoft AD.
  - Une instance Amazon EC2 jointe à l'AD que vous pouvez utiliser pour gérer l'annuaire.
3. Dans la section Paramètres de la page de création rapide d'une pile, entrez des mots de passe pour les paramètres suivants :
- AdminPassword
  - ReadOnlyPassword
  - UserPassword

Prenez note des mots de passe. Vous les utiliserez ultérieurement dans ce didacticiel.

4. Pour DomainName, entrez **corp.example.com**.
5. Pour Keypair, entrez le nom d'une paire de clés Amazon EC2.
6. Cochez les cases correspondant à chacune des fonctionnalités d'accès au bas de la page.
7. Sélectionnez Créer la pile.
8. Une fois que la CloudFormation pile a atteint son CREATE\_COMPLETE état, choisissez l'onglet Sorties de la pile. Notez les noms et les identifiants des ressources de sortie, car vous devrez les utiliser ultérieurement. Les sorties fournissent les informations nécessaires à la création du cluster.



CloudFormation > Stacks > PclusterAD-1234abc

Stacks (3)    View nested < 1 >

PclusterAD  
2022-05-18 10:27:24 UTC-0700  
CREATE\_COMPLETE

PclusterAD-abcd123

Stack info | Events | Resources | **Outputs** | Parameters | Template | Change sets

Outputs (10)

| Key                               | Value                                                                          |
|-----------------------------------|--------------------------------------------------------------------------------|
| DomainAddrLdap                    | ldap://10.0.111.88,ldap://10.0.222.111                                         |
| DomainAddrLdaps                   | ldaps://corp.example.com                                                       |
| DomainCertificateArn              | arn:aws:acm:us-east-1:123456789012:certificate/1234abcd-ef56-78gh-ij90-abcd1   |
| DomainCertificateSecretArn        | arn:aws:secretsmanager:us-east-1:123456789012:secret:DomainCertificateSecret-F |
| DomainCertificateSecretReadPolicy | arn:aws:iam::123456789012:policy/DomainCertificateSecretReadPolicy-PclusterAD  |
| DomainName                        | corp.example.com                                                               |
| DomainReadOnlyUser                | cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com                     |
| PasswordSecretArn                 | arn:aws:secretsmanager:us-east-1:123456789012:secret>PasswordSecret-PclusterA  |
| PrivateSubnetIds                  | subnet-1234567890abcdef0,subnet-abcdef01234567890                              |
| VpcId                             | vpc-021345abcdef6789                                                           |

9. Pour terminer les exercices ([Facultatif](#)) [Étape 2 : Gérer les utilisateurs et les groupes AD](#), vous avez besoin de l'ID du répertoire. Choisissez Ressources et faites défiler l'écran vers le bas pour prendre note de l'ID du répertoire.
10. Continuez sur [\(Facultatif\) Étape 2 : Gérer les utilisateurs et les groupes AD](#) ou [Étape 3 : Création du cluster](#).

## Manuelle

Créez un VPC pour le service d'annuaire avec deux sous-réseaux dans différentes zones de disponibilité et un. AWS Managed Microsoft AD

## Créez l'AD

### Note

- Le répertoire et le nom de domaine sont `corp.example.com`. Le nom abrégé est `CORP`.
- Modifiez le Admin mot de passe dans le script.



- La création de l'Active Directory (AD) prend au moins 15 minutes.

Utilisez le script Python suivant pour créer le VPC, les sous-réseaux et les ressources AD dans votre environnement local. Région AWS Enregistrez ce fichier sous `ad.py` et exécutez-le.

```
import boto3
import time
from pprint import pprint

vpc_name = "PclusterVPC"
ad_domain = "corp.example.com"
admin_password = "asdfASDF1234"

Amazon EC2 = boto3.client("ec2")
ds = boto3.client("ds")
region = boto3.Session().region_name

Create the VPC, Subnets, IGW, Routes
vpc = ec2.create_vpc(CidrBlock="10.0.0.0/16")["Vpc"]
vpc_id = vpc["VpcId"]
time.sleep(30)
ec2.create_tags(Resources=[vpc_id], Tags=[{"Key": "Name", "Value": vpc_name}])
subnet1 = ec2.create_subnet(VpcId=vpc_id, CidrBlock="10.0.0.0/17",
 AvailabilityZone=f"{region}a")["Subnet"]
subnet1_id = subnet1["SubnetId"]
time.sleep(30)
ec2.create_tags(Resources=[subnet1_id], Tags=[{"Key": "Name", "Value": f"{vpc_name}/
subnet1"}])
ec2.modify_subnet_attribute(SubnetId=subnet1_id, MapPublicIpOnLaunch={"Value": True})
subnet2 = ec2.create_subnet(VpcId=vpc_id, CidrBlock="10.0.128.0/17",
 AvailabilityZone=f"{region}b")["Subnet"]
subnet2_id = subnet2["SubnetId"]
time.sleep(30)
ec2.create_tags(Resources=[subnet2_id], Tags=[{"Key": "Name", "Value": f"{vpc_name}/
subnet2"}])
ec2.modify_subnet_attribute(SubnetId=subnet2_id, MapPublicIpOnLaunch={"Value": True})
igw = ec2.create_internet_gateway()["InternetGateway"]
ec2.attach_internet_gateway(InternetGatewayId=igw["InternetGatewayId"], VpcId=vpc_id)
route_table = ec2.describe_route_tables(Filters=[{"Name": "vpc-id", "Values":
 [vpc_id]}])["RouteTables"][0]
ec2.create_route(RouteTableId=route_table["RouteTableId"],
 DestinationCidrBlock="0.0.0.0/0", GatewayId=igw["InternetGatewayId"])
```

```
ec2.modify_vpc_attribute(VpcId=vpc_id, EnableDnsSupport={"Value": True})
ec2.modify_vpc_attribute(VpcId=vpc_id, EnableDnsHostnames={"Value": True})

Create the Active Directory
ad = ds.create_microsoft_ad(
 Name=ad_domain,
 Password=admin_password,
 Description="ParallelCluster AD",
 VpcSettings={"VpcId": vpc_id, "SubnetIds": [subnet1_id, subnet2_id]},
 Edition="Standard",
)
directory_id = ad["DirectoryId"]

Wait for completion
print("Waiting for the directory to be created...")
directories = ds.describe_directories(DirectoryIds=[directory_id])
["DirectoryDescriptions"]
directory = directories[0]
while directory["Stage"] in {"Requested", "Creating"}:
 time.sleep(3)
 directories = ds.describe_directories(DirectoryIds=[directory_id])
["DirectoryDescriptions"]
 directory = directories[0]

dns_ip_addrs = directory["DnsIpAddrs"]

pprint({"directory_id": directory_id,
 "vpc_id": vpc_id,
 "subnet1_id": subnet1_id,
 "subnet2_id": subnet2_id,
 "dns_ip_addrs": dns_ip_addrs})
```

Voici un exemple de sortie du script Python.

```
{
 "directory_id": "d-abcdef01234567890",
 "dns_ip_addrs": ["192.0.2.254", "203.0.113.237"],
 "subnet1_id": "subnet-021345abcdef6789",
 "subnet2_id": "subnet-1234567890abcdef0",
 "vpc_id": "vpc-021345abcdef6789"
}
```

Notez les noms et les identifiants des ressources de sortie. Vous les utiliserez ultérieurement.

Une fois le script terminé, passez à l'étape suivante.

## Création d'une instance Amazon EC2

### New Amazon EC2 console

1. Connectez-vous au AWS Management Console.
2. Si aucun rôle n'est associé aux politiques répertoriées à l'étape 4 ci-jointe, ouvrez la console IAM à l'[adresse https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/). Sinon, passez à l'étape 5.
3. Créez la ResetUserPassword politique en remplaçant le contenu surligné en rouge par votre Région AWS identifiant, votre identifiant de compte et l'identifiant du répertoire à partir de la sortie du script que vous avez exécuté pour créer l'AD.

### ResetUserPassword

```
{
 "Statement": [
 {
 "Action": [
 "ds:ResetUserPassword"
],
 "Resource": "arn:aws:ds:region-id:123456789012:directory/d-abcdef01234567890",
 "Effect": "Allow"
 }
]
}
```

4. Créez un rôle IAM avec les politiques suivantes associées.
  - AWS politique gérée | [AmazonSSM ManagedInstanceCore](#)
  - AWS politique gérée | [AmazonSSM DirectoryServiceAccess](#)
  - ResetUserPassword politique
5. Ouvrez la console Amazon EC2 à l'adresse <https://console.aws.amazon.com/ec2/>.
6. Dans le tableau de bord Amazon EC2, choisissez Launch Instance.
7. Dans Images d'applications et de systèmes d'exploitation, sélectionnez une AMI Amazon Linux 2 récente.
8. Pour Type d'instance, choisissez t2.micro.
9. Pour Key pair, choisissez une paire de clés.

10. Sous Paramètres réseau, choisissez Modifier.
11. Pour VPC, sélectionnez le répertoire VPC.
12. Faites défiler l'écran vers le bas et sélectionnez Détails avancés.
13. Dans Détails avancés, Répertoire des jointures de domaines, sélectionnez **corp.example.com**.
14. Pour le profil d'instance IAM, choisissez le rôle que vous avez créé à l'étape 1 ou un rôle dont les politiques sont répertoriées à l'étape 4 ci-jointes.
15. Dans Résumé, choisissez Launch instance.
16. Notez l'ID de l'instance (par exemple, i-1234567890abcdef0) et attendez que le lancement de l'instance soit terminé.
17. Une fois l'instance lancée, passez à l'étape suivante.

### Old Amazon EC2 console

1. Connectez-vous au AWS Management Console.
2. Si aucun rôle n'est associé aux politiques répertoriées à l'étape 4 ci-jointe, ouvrez la console IAM à l'[adresse https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/). Sinon, passez à l'étape 5.
3. Créez la ResetUserPassword politique. Remplacez le contenu surligné en rouge par votre Région AWS Compte AWS identifiant, votre identifiant et l'identifiant du répertoire figurant dans le résultat du script que vous avez exécuté pour créer l'Active Directory (AD).

### ResetUserPassword

```
{
 "Statement": [
 {
 "Action": [
 "ds:ResetUserPassword"
],
 "Resource": "arn:aws:ds:region-id:123456789012:directory/d-
abcdef01234567890",
 "Effect": "Allow"
 }
]
}
```

4. Créez un rôle IAM avec les politiques suivantes associées.

- AWS politique gérée | [AmazonSSM ManagedInstanceCore](#)
  - AWS politique gérée | [AmazonSSM DirectoryServiceAccess](#)
  - Stratégie ResetUserPassword
5. Ouvrez la console Amazon EC2 à l'adresse <https://console.aws.amazon.com/ec2/>.
  6. Dans le tableau de bord Amazon EC2, choisissez Launch Instance.
  7. Dans Images d'applications et de systèmes d'exploitation, sélectionnez une AMI Amazon Linux 2 récente.
  8. Concernant l'option Instance type (Type d'instance), choisissez t2.micro.
  9. Pour Key pair, choisissez une paire de clés.
  10. Dans Paramètres réseau, choisissez Modifier.
  11. Dans Paramètres réseau, VPC, sélectionnez le répertoire VPC.
  12. Faites défiler l'écran vers le bas et sélectionnez Détails avancés.
  13. Dans Détails avancés, Répertoire des jointures de domaines, sélectionnez **corp.example.com**.
  14. Dans Détails avancés, Profil d'instance, choisissez le rôle que vous avez créé à l'étape 1 ou un rôle auquel sont jointes les politiques répertoriées à l'étape 4.
  15. Dans Résumé, choisissez Launch instance.
  16. Notez l'ID de l'instance (par exemple, `i-1234567890abcdef0`) et attendez que le lancement de l'instance soit terminé.
  17. Une fois l'instance lancée, passez à l'étape suivante.

## Joignez votre instance à l'AD

1. Connectez-vous à votre instance et rejoignez le domaine AD en tant que **admin**.

Exécutez les commandes suivantes pour vous connecter à l'instance.

```
$ INSTANCE_ID="i-1234567890abcdef0"
```

```
$ PUBLIC_IP=$(aws ec2 describe-instances \
--instance-ids $INSTANCE_ID \
--query "Reservations[0].Instances[0].PublicIpAddress" \
--output text)
```

```
$ ssh -i ~/.ssh/keys/keypair.pem ec2-user@$PUBLIC_IP
```

2. Installez le logiciel nécessaire et rejoignez le royaume.

```
$ sudo yum -y install sssd realmd oddjob oddjob-mkhomedir adcli samba-common samba-common-tools krb5-workstation openldap-clients policycoreutils-python
```

3. Remplacez le mot de passe administrateur par votre **admin** mot de passe.

```
$ ADMIN_PW="asdfASDF1234"
```

```
$ echo $ADMIN_PW | sudo realm join -U Admin corp.example.com
Password for Admin:
```

Si la procédure précédente a réussi, vous êtes rattaché au royaume et pouvez passer à l'étape suivante.

## Ajouter des utilisateurs à l'AD

1. Créez le ReadOnlyUser et un utilisateur supplémentaire.

Dans cette étape, vous utiliserez les outils [adcli](#) et [openldap-clients](#) que vous avez installés lors d'une étape précédente.

```
$ echo $ADMIN_PW | adcli create-user -x -U Admin --domain=corp.example.com --
display-name=ReadOnlyUser ReadOnlyUser
```

```
$ echo $ADMIN_PW | adcli create-user -x -U Admin --domain=corp.example.com --
display-name=user000 user000
```

2. Vérifiez que les utilisateurs sont créés :

Les adresses IP DNS du répertoire sont des sorties du script Python.

```
$ DIRECTORY_IP="192.0.2.254"
```

```
$ ldapsearch -x -h $DIRECTORY_IP -D Admin -w $ADMIN_PW -b
"cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com"
```

```
$ ldapsearch -x -h $DIRECTORY_IP -D Admin -w $ADMIN_PW -b
"cn=user000,ou=Users,ou=CORP,dc=corp,dc=example,dc=com"
```

Par défaut, lorsque vous créez un utilisateur avec `lead-cli`, l'utilisateur est désactivé.

3. Réinitialisez et activez les mots de passe utilisateur depuis votre ordinateur local :

Déconnectez-vous de votre instance Amazon EC2.

#### Note

- `ro-p@ssw0rd` est le mot de passe de `ReadOnlyUser`, extrait de AWS Secrets Manager.
- `user-p@ssw0rd` est le mot de passe d'un utilisateur du cluster fourni lorsque vous vous connectez (ssh) au cluster.

`directory-id` s'agit d'une sortie du script Python.

```
$ DIRECTORY_ID="d-abcdef01234567890"
```

```
$ aws ds reset-user-password \
--directory-id $DIRECTORY_ID \
--user-name "ReadOnlyUser" \
--new-password "ro-p@ssw0rd" \
--region "region-id"
```

```
$ aws ds reset-user-password \
--directory-id $DIRECTORY_ID \
--user-name "user000" \
--new-password "user-p@ssw0rd" \
--region "region-id"
```

4. Ajoutez le mot de passe à un secret du Gestionnaire de Secrets.

Maintenant que vous avez créé `ReadOnlyUser` et défini le mot de passe, stockez-le dans un secret qui AWS ParallelCluster servira à valider les connexions.

Utilisez Secrets Manager pour créer un nouveau secret contenant le mot de passe correspondant `ReadOnlyUser` à la valeur. Le format de la valeur secrète doit être uniquement en texte brut (pas au format JSON). Notez l'ARN secret pour les étapes futures.

```
$ aws secretsmanager create-secret --name "ADSecretPassword" \
--region region_id \
--secret-string "ro-p@ssw0rd" \
--query ARN \
--output text
arn:aws:secretsmanager:region-id:123456789012:secret:ADSecretPassword-1234
```

Configuration de LDAPS avec vérification des certificats (recommandée)

Notez les identifiants des ressources. Vous les utiliserez par étapes ultérieures.

1. Générez un certificat de domaine, localement.

```
$ PRIVATE_KEY="corp-example-com.key"
CERTIFICATE="corp-example-com.crt"
printf ".\n.\n.\n.\n.\n.\ncorp.example.com\n.\n\n" | openssl req -x509 -sha256 -nodes -
newkey rsa:2048 -keyout $PRIVATE_KEY -days 365 -out $CERTIFICATE
```

2. Stockez le certificat dans Secrets Manager pour qu'il puisse être récupéré ultérieurement dans le cluster.

```
$ aws secretsmanager create-secret --name example-cert \
--secret-string file://$CERTIFICATE \
--region region-id
{
 "ARN": "arn:aws:secretsmanager:region-id:123456789012:secret:example-
cert-123abc",
 "Name": "example-cert",
 "VersionId": "14866070-092a-4d5a-bcdd-9219d0566b9c"
}
```

3. Ajoutez la politique suivante au rôle IAM que vous avez créé pour joindre l'instance Amazon EC2 au domaine AD.



## PutDomainCertificateSecrets

```
{
 "Statement": [
 {
 "Action": [
 "secretsmanager:PutSecretValue"
],
 "Resource": [
 "arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-123abc",
],
 "Effect": "Allow"
 }
]
}
```

4. Importez le certificat dans AWS Certificate Manager (ACM).

```
$ aws acm import-certificate --certificate fileb://$CERTIFICATE \
 --private-key fileb://$PRIVATE_KEY \
 --region region-id
{
 "CertificateArn": "arn:aws:acm:region-
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72"
}
```

5. Créez et l'équilibreur de charge placé devant les points de terminaison Active Directory.

```
$ aws elbv2 create-load-balancer --name CorpExampleCom-NLB \
 --type network \
 --scheme internal \
 --subnets subnet-1234567890abcdef0 subnet-021345abcdef6789 \
 --region region-id
{
 "LoadBalancers": [
 {
 "LoadBalancerArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:loadbalancer/net/CorpExampleCom-NLB/3afe296bf4ba80d4",
 "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-id.amazonaws.com",
 "CanonicalHostedZoneId": "Z2IF0LAFXWL04F",
 "CreatedTime": "2022-05-05T12:56:55.988000+00:00",
 }
]
}
```

```

 "LoadBalancerName": "CorpExampleCom-NLB",
 "Scheme": "internal",
 "VpcId": "vpc-021345abcdef6789",
 "State": {
 "Code": "provisioning"
 },
 "Type": "network",
 "AvailabilityZones": [
 {
 "ZoneName": "region-idb",
 "SubnetId": "subnet-021345abcdef6789",
 "LoadBalancerAddresses": []
 },
 {
 "ZoneName": "region-ida",
 "SubnetId": "subnet-1234567890abcdef0",
 "LoadBalancerAddresses": []
 }
],
 "IpAddressType": "ipv4"
 }
]
}

```

6. Créez le groupe cible qui cible les points de terminaison Active Directory.

```

$ aws elbv2 create-target-group --name CorpExampleCom-Targets --protocol TCP \
 --port 389 \
 --target-type ip \
 --vpc-id vpc-021345abcdef6789 \
 --region region-id
{
 "TargetGroups": [
 {
 "TargetGroupArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81",
 "TargetGroupName": "CorpExampleCom-Targets",
 "Protocol": "TCP",
 "Port": 389,
 "VpcId": "vpc-021345abcdef6789",
 "HealthCheckProtocol": "TCP",
 "HealthCheckPort": "traffic-port",
 "HealthCheckEnabled": true,
 "HealthCheckIntervalSeconds": 30,

```

```

 "HealthCheckTimeoutSeconds": 10,
 "HealthyThresholdCount": 3,
 "UnhealthyThresholdCount": 3,
 "TargetType": "ip",
 "IpAddressType": "ipv4"
 }
]
}

```

7. Enregistrez les points de terminaison Active Directory (AD) dans le groupe cible.

```

$ aws elbv2 register-targets --target-group-arn
arn:aws:elasticloadbalancing:region-id:123456789012:targetgroup/CorpExampleCom-
Targets/44577c583b695e81 \
--targets Id=192.0.2.254,Port=389 Id=203.0.113.237,Port=389 \
--region region-id

```

8. Créez l'écouteur LB avec le certificat.

```

$ aws elbv2 create-listener --load-balancer-arn
arn:aws:elasticloadbalancing:region-id:123456789012:loadbalancer/net/
CorpExampleCom-NLB/3afe296bf4ba80d4 \
--protocol TLS \
--port 636 \
--default-actions
Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81 \
--ssl-policy ELBSecurityPolicy-TLS-1-2-2017-01 \
--certificates CertificateArn=arn:aws:acm:region-
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72 \
--region region-id
"Listeners": [
{
 "ListenerArn": "arn:aws:elasticloadbalancing:region-id:123456789012:listener/
net/CorpExampleCom-NLB/3afe296bf4ba80d4/a8f9d97318743d4b",
 "LoadBalancerArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:loadbalancer/net/CorpExampleCom-NLB/3afe296bf4ba80d4",
 "Port": 636,
 "Protocol": "TLS",
 "Certificates": [
 {
 "CertificateArn": "arn:aws:acm:region-
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72"
 }
]
}
]

```

```

 }
],
 "SslPolicy": "ELBSecurityPolicy-TLS-1-2-2017-01",
 "DefaultActions": [
 {
 "Type": "forward",
 "TargetGroupArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81",
 "ForwardConfig": {
 "TargetGroups": [
 {
 "TargetGroupArn": "arn:aws:elasticloadbalancing:region-
id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81"
 }
]
 }
 }
]
}

```

9. Créez la zone hébergée pour rendre le domaine détectable dans le VPC du cluster.

```

$ aws route53 create-hosted-zone --name corp.example.com \
 --vpc VPCRegion=region-id,VPCId=vpc-021345abcdef6789 \
 --caller-reference "ParallelCluster AD Tutorial"
{
 "Location": "https://route53.amazonaws.com/2013-04-01/hostedzone/
Z09020002B5MZQNXMSJUB",
 "HostedZone": {
 "Id": "/hostedzone/Z09020002B5MZQNXMSJUB",
 "Name": "corp.example.com.",
 "CallerReference": "ParallelCluster AD Tutorial",
 "Config": {
 "PrivateZone": true
 },
 "ResourceRecordSetCount": 2
 },
 "ChangeInfo": {
 "Id": "/change/C05533343BF3IKSORW1TQ",
 "Status": "PENDING",
 "SubmittedAt": "2022-05-05T13:21:53.863000+00:00"
 },
}

```

```
"VPC": {
 "VPCRegion": "region-id",
 "VPCId": "vpc-021345abcdef6789"
}
}
```

10. Créez un fichier nommé **recordset-change.json** avec le contenu suivant. **HostedZoneId** est l'ID de zone hébergée canonique de l'équilibreur de charge.

```
{
 "Changes": [
 {
 "Action": "CREATE",
 "ResourceRecordSet": {
 "Name": "corp.example.com",
 "Type": "A",
 "Region": "region-id",
 "SetIdentifier": "example-active-directory",
 "AliasTarget": {
 "HostedZoneId": "Z2IF0LAFXWL04F",
 "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-
id.amazonaws.com",
 "EvaluateTargetHealth": true
 }
 }
 }
]
}
```

11. Soumettez la modification du jeu d'enregistrements à la zone hébergée, cette fois en utilisant l'ID de zone hébergée.

```
$ aws route53 change-resource-record-sets --hosted-zone-id Z09020002B5MZQNMSJUB \
--change-batch file://recordset-change.json
{
 "ChangeInfo": {
 "Id": "/change/C0137926I56R3GC7XW2Y",
 "Status": "PENDING",
 "SubmittedAt": "2022-05-05T13:40:36.553000+00:00"
 }
}
```

12. Créez un document de politique **policy.json** avec le contenu suivant.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
 "secretsmanager:GetSecretValue"
],
 "Resource": [
 "arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-abc123"
],
 "Effect": "Allow"
 }
]
}
```

13. Créez un document de politique nommé **policy.json** avec le contenu suivant.

```
$ aws iam create-policy --policy-name ReadCertExample \
 --policy-document file://policy.json
{
 "Policy": {
 "PolicyName": "ReadCertExample",
 "PolicyId": "ANPAUXUVBC42VZSI4LDY",
 "Arn": "arn:aws:iam::123456789012:policy/ReadCertExample-efg456",
 "Path": "/",
 "DefaultVersionId": "v1",
 "AttachmentCount": 0,
 "PermissionsBoundaryUsageCount": 0,
 "IsAttachable": true,
 "CreateDate": "2022-05-05T13:42:18+00:00",
 "UpdateDate": "2022-05-05T13:42:18+00:00"
 }
}
```

14. Continuez à suivre les étapes indiquées dans [\(Facultatif\) Étape 2 : Gérer les utilisateurs et les groupes AD](#) ou [Étape 3 : Création du cluster](#).

## (Facultatif) Étape 2 : Gérer les utilisateurs et les groupes AD

Au cours de cette étape, vous gérez les utilisateurs et les groupes à partir d'une instance Amazon EC2 Amazon Linux 2 jointe au domaine Active Directory (AD).

Si vous avez suivi le chemin automatique, redémarrez et connectez-vous à l'instance jointe à AD créée dans le cadre de l'automatisation.

Si vous avez suivi le chemin manuel, redémarrez et connectez-vous à l'instance que vous avez créée et jointe à AD lors des étapes précédentes.

Au cours de ces étapes, vous utilisez les outils [adcli](#) et [openldap-clients](#) qui ont été installés dans l'instance dans le cadre d'une étape précédente.

Connectez-vous à une instance Amazon EC2 jointe au domaine AD

1. Depuis la console Amazon EC2, sélectionnez l'instance Amazon EC2 sans titre créée lors des étapes précédentes. L'état de l'instance est peut-être Arrêté.
2. Si l'état de l'instance est Arrêté, choisissez État de l'instance, puis Démarrer l'instance.
3. Une fois les vérifications d'état terminées, sélectionnez l'instance, puis choisissez Connect and SSH in to the instance.

Gérez les utilisateurs et les groupes lorsque vous êtes connecté à une instance Amazon EC2 Amazon Linux 2 qui a rejoint l'AD

Lorsque vous exécutez les `adcli` commandes avec l'option `-U "Admin"`, vous êtes invité à saisir le mot de passe AD. Vous incluez le mot de passe AD dans les `ldapsearch` commandes.

1. Créez un utilisateur.

```
$ adcli create-user "clusteruser" --domain "corp.example.com" -U "Admin"
```

- Définissez un mot de passe utilisateur.

```
$ aws --region "region-id" ds reset-user-password --directory-id "d-
abcdef01234567890" --user-name "clusteruser" --new-password "new-p@ssw0rd"
```

- Créez un groupe.

```
$ adcli create-group "clusterteam" --domain "corp.example.com" -U "Admin"
```

- Ajoutez un utilisateur à un groupe.

```
$ adcli add-member "clusterteam" "clusteruser" --domain "corp.example.com" -U
"Admin"
```

- Décrivez les utilisateurs et les groupes.

Décrivez tous les utilisateurs.

```
$ ldapsearch "(&(objectClass=user))" -x -h "192.0.2.254" -b
"DC=corp,DC=example,DC=com" -D
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

Décrivez un utilisateur spécifique.

```
$ ldapsearch "(&(objectClass=user)(cn=clusteruser))"
-x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

Décrivez tous les utilisateurs avec un modèle de nom.

```
$ ldapsearch "(&(objectClass=user)(cn=user*))" -x -h "192.0.2.254" -b
"DC=corp,DC=example,DC=com" -D
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

Décrivez tous les utilisateurs qui font partie d'un groupe spécifique.

```
$ ldapsearch "(&(objectClass=user)
(memberOf=CN=clusterteam,OU=Users,OU=CORP,DC=corp,DC=example,DC=com))"
-x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D
"CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```



## Décrire tous les groupes

```
$ ldapsearch "objectClass=group" -x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D "CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

## Décrire un groupe spécifique

```
$ ldapsearch "(&(objectClass=group)(cn=clusterteam))" -x -h "192.0.2.254" -b "DC=corp,DC=example,DC=com" -D "CN=Admin,OU=Users,OU=CORP,DC=corp,DC=example,DC=com" -w "p@ssw0rd"
```

## 6. Supprimer un utilisateur d'un groupe.

```
$ adcli remove-member "clusterteam" "clusteruser" --domain "corp.example.com" -U "Admin"
```

## 7. Supprime un utilisateur.

```
$ adcli delete-user "clusteruser" --domain "corp.example.com" -U "Admin"
```

## 8. Supprimez un groupe.

```
$ adcli delete-group "clusterteam" --domain "corp.example.com" -U "Admin"
```

## Étape 3 : Création du cluster

Si vous n'avez pas quitté l'instance Amazon EC2, faites-le maintenant.

L'environnement est configuré pour créer un cluster capable d'authentifier les utilisateurs auprès d'Active Directory (AD).

Créez une configuration de cluster simple et fournissez les paramètres nécessaires à la connexion à l'AD. Pour plus d'informations, consultez la section [DirectoryService](#).

Choisissez l'une des configurations de cluster suivantes et copiez-la dans un fichier nommé `ldaps_config.yaml`, `ldaps_nocert_config.yaml`, ou `ldap_config.yaml`.

Nous vous recommandons de choisir la configuration LDAPS avec vérification des certificats. Si vous choisissez cette configuration, vous devez également copier le script bootstrap dans un

fichier nommé `active-directory.head.post.sh`. Vous devez également le stocker dans un compartiment Amazon S3 comme indiqué dans le fichier de configuration.

LDAPS avec configuration de vérification des certificats (recommandé)

### Note

Les composants suivants doivent être modifiés.

- `KeyName`: l'une de vos paires de clés Amazon EC2.
- `SubnetId` / `SubnetIds`: l'un des identifiants de sous-réseau fournis dans la sortie de la pile de création CloudFormation rapide (didacticiel automatisé) ou du script python (didacticiel manuel).
- `Region`: Région dans laquelle vous avez créé l'infrastructure AD.
- `DomainAddr`: Cette adresse IP est l'une des adresses DNS de votre service AD.
- `PasswordSecretArn`: le nom de ressource Amazon (ARN) du secret qui contient le mot de passe du `DomainReadOnlyUser`.
- `BucketName`: nom du compartiment contenant le script bootstrap.
- `AdditionalPolicies/Policy`: Le nom de ressource Amazon (ARN) de la politique de certification du domaine lu `ReadCertExample`.
- `CustomActions/OnNodeConfigured/Args`: Le nom de ressource Amazon (ARN) du secret contenant la politique de certification du domaine.

Pour une meilleure sécurité, nous vous suggérons d'utiliser la `AllowedIps` configuration `HeadNodeSsh` pour limiter l'accès SSH au nœud principal.

```
Region: region-id
Image:
 Os: alinux2
HeadNode:
 InstanceType: t2.micro
 Networking:
 SubnetId: subnet-abcdef01234567890
 Ssh:
 KeyName: keypair
 Iam:
```

```

AdditionalIamPolicies:
 - Policy: arn:aws:iam::123456789012:policy/ReadCertExample
S3Access:
 - BucketName: my-bucket
 EnableWriteAccess: false
 KeyName: bootstrap/active-directory/active-directory.head.post.sh
CustomActions:
 OnNodeConfigured:
 Script: s3://my-bucket/bootstrap/active-directory/active-directory.head.post.sh
 Args:
 - arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-123abc
 - /opt/parallelcluster/shared/directory_service/domain-certificate.crt
Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: queue0
 ComputeResources:
 - Name: queue0-t2-micro
 InstanceType: t2.micro
 MinCount: 1
 MaxCount: 10
 Networking:
 SubnetIds:
 - subnet-abcdef01234567890
DirectoryService:
 DomainName: corp.example.com
 DomainAddr: ldaps://corp.example.com
 PasswordSecretArn: arn:aws:secretsmanager:region-id:123456789012:secret:ADSecretPassword-1234
 DomainReadOnlyUser: cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
 LdapTlsCaCert: /opt/parallelcluster/shared/directory_service/domain-certificate.crt
 LdapTlsReqCert: hard

```

## Script Bootstrap

Après avoir créé le fichier bootstrap et avant de le télécharger dans votre compartiment S3, exécutez-le `chmod +x active-directory.head.post.sh` pour donner l'autorisation d' AWS ParallelCluster exécution.

```

#!/bin/bash
set -e

CERTIFICATE_SECRET_ARN="$1"

```

```
CERTIFICATE_PATH="$2"

[[-z $CERTIFICATE_SECRET_ARN]] && echo "[ERROR] Missing CERTIFICATE_SECRET_ARN" &&
 exit 1
[[-z $CERTIFICATE_PATH]] && echo "[ERROR] Missing CERTIFICATE_PATH" && exit 1

source /etc/parallelcluster/cfnconfig
REGION="${cfn_region:?}"

mkdir -p $(dirname $CERTIFICATE_PATH)
aws secretsmanager get-secret-value --region $REGION --secret-id
 $CERTIFICATE_SECRET_ARN --query SecretString --output text > $CERTIFICATE_PATH
```

## LDAPS sans configuration de vérification de certificat

### Note

Les composants suivants doivent être modifiés.

- **KeyName:** l'une de vos paires de clés Amazon EC2.
- **SubnetId / SubnetIds:** l'un des identifiants de sous-réseau figurant dans la sortie de la pile de création CloudFormation rapide (didacticiel automatisé) ou du script python (didacticiel manuel).
- **Region:** Région dans laquelle vous avez créé l'infrastructure AD.
- **DomainAddr:** Cette adresse IP est l'une des adresses DNS de votre service AD.
- **PasswordSecretArn:** le nom de ressource Amazon (ARN) du secret qui contient le mot de passe du `DomainReadOnlyUser`.

Pour une meilleure sécurité, nous vous suggérons d'utiliser la `AllowedIps` configuration `HeadNode /Ssh/` pour limiter l'accès SSH au nœud principal.

```
Region: region-id
Image:
 Os: alinux2
HeadNode:
 InstanceType: t2.micro
Networking:
```

```
SubnetId: subnet-abcdef01234567890
Ssh:
 KeyName: keypair
Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: queue0
 ComputeResources:
 - Name: queue0-t2-micro
 InstanceType: t2.micro
 MinCount: 1
 MaxCount: 10
 Networking:
 SubnetIds:
 - subnet-abcdef01234567890
DirectoryService:
 DomainName: corp.example.com
 DomainAddr: ldaps://corp.example.com
 PasswordSecretArn: arn:aws:secretsmanager:region-id:123456789012:secret:ADSecretPassword-1234
 DomainReadOnlyUser: cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
 LdapTlsReqCert: never
```

## Configuration LDAP

### Note

Les composants suivants doivent être modifiés.

- `KeyName`: l'une de vos paires de clés Amazon EC2.
- `SubnetId` / `SubnetIds`: l'un des identifiants de sous-réseau fournis dans la sortie de la pile de création CloudFormation rapide (didacticiel automatisé) ou du script python (didacticiel manuel).
- `Region`: Région dans laquelle vous avez créé l'infrastructure AD.
- `DomainAddr`: Cette adresse IP est l'une des adresses DNS de votre service AD.
- `PasswordSecretArn`: le nom de ressource Amazon (ARN) du secret qui contient le mot de passe du `DomainReadOnlyUser`.

Pour une meilleure sécurité, nous vous suggérons d'utiliser la AllowedIps configuration HeadNode /Ssh/ pour limiter l'accès SSH au nœud principal.

```

Region: region-id
Image:
 Os: alinux2
HeadNode:
 InstanceType: t2.micro
 Networking:
 SubnetId: subnet-abcdef01234567890
 Ssh:
 KeyName: keypair
Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: queue0
 ComputeResources:
 - Name: queue0-t2-micro
 InstanceType: t2.micro
 MinCount: 1
 MaxCount: 10
 Networking:
 SubnetIds:
 - subnet-abcdef01234567890
DirectoryService:
 DomainName: dc=corp,dc=example,dc=com
 DomainAddr: ldap://192.0.2.254,ldap://203.0.113.237
 PasswordSecretArn: arn:aws:secretsmanager:region-id:123456789012:secret:ADSecretPassword-1234
 DomainReadOnlyUser: cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
 AdditionalSssdConfigs:
 ldap_auth_disable_tls_never_use_in_production: True

```

Créez votre cluster à l'aide de la commande suivante.

```

$ pcluster create-cluster --cluster-name "ad-cluster" --cluster-configuration "./ldaps_config.yaml"
{
 "cluster": {
 "clusterName": "pcluster",

```

```
"cloudformationStackStatus": "CREATE_IN_PROGRESS",
"cloudformationStackArn": "arn:aws:cloudformation:region-id:123456789012:stack/ad-cluster/1234567-abcd-0123-def0-abcdef0123456",
"region": "region-id",
"version": 3.7.0,
"clusterStatus": "CREATE_IN_PROGRESS"
}
}
```

## Étape 4 : Se connecter au cluster en tant qu'utilisateur

Vous pouvez déterminer l'état du cluster à l'aide des commandes suivantes.

```
$ pcluster describe-cluster -n ad-cluster --region "region-id" --query "clusterStatus"
```

La sortie est la suivante.

```
"CREATE_IN_PROGRESS" / "CREATE_COMPLETE"
```

Lorsque le statut atteint "CREATE\_COMPLETE", connectez-vous avec le nom d'utilisateur et le mot de passe créés.

```
$ HEAD_NODE_IP=$(pcluster describe-cluster -n "ad-cluster" --region "region-id" --query headNode.publicIpAddress | xargs echo)
```

```
$ ssh user000@$HEAD_NODE_IP
```

Vous pouvez vous connecter sans mot de passe en fournissant la SSH clé créée pour le nouvel utilisateur à l'adresse/home/user000@HEAD\_NODE\_IP/.ssh/id\_rsa.

Si la ssh commande a réussi, vous vous êtes connecté avec succès au cluster en tant qu'utilisateur authentifié pour utiliser Active Directory (AD).

## Étape 5 : nettoyer

1. Depuis votre ordinateur local, supprimez le cluster.

```
$ pcluster delete-cluster --cluster-name "ad-cluster" --region "region-id"
```

```
{
 "cluster": {
 "clusterName": "ad-cluster",
 "cloudformationStackStatus": "DELETE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:region-id:123456789012:stack/ad-cluster/1234567-abcd-0123-def0-abcdef0123456",
 "region": "region-id",
 "version": "3.7.0",
 "clusterStatus": "DELETE_IN_PROGRESS"
 }
}
```

2. Vérifiez la progression du cluster en cours de suppression.

```
$ pcluster describe-cluster --cluster-name "ad-cluster" --region "region-id" --
query "clusterStatus"
"DELETE_IN_PROGRESS"
```

Une fois le cluster supprimé avec succès, passez à l'étape suivante.

## Automatisé

### Supprimer les ressources Active Directory

1. Tiré de <https://console.aws.amazon.com/cloudformation/>.
2. Dans le volet de navigation, choisissez Stack (Piles).
3. Dans la liste des piles, choisissez la pile AD (par exemple, `pcluster-ad`).
4. Sélectionnez Delete (Supprimer).

## Manuelle

1. Supprimez l'instance Amazon EC2.
  - a. Sur <https://console.aws.amazon.com/ec2/>, sélectionnez Instances dans le volet de navigation.
  - b. Dans la liste des instances, choisissez l'instance que vous avez créée pour ajouter des utilisateurs à l'annuaire.
  - c. Choisissez État de l'instance, puis Terminate instance.



## 2. Supprimez la zone hébergée.

- a. Créez un `recordset-delete.json` avec le contenu suivant. Dans cet exemple, `HostedZoneId` il s'agit de l'ID de zone hébergée canonique de l'équilibreur de charge.

```
{
 "Changes": [
 {
 "Action": "DELETE",
 "ResourceRecordSet": {
 "Name": "corp.example.com",
 "Type": "A",
 "Region": "region-id",
 "SetIdentifier": "pcluster-active-directory",
 "AliasTarget": {
 "HostedZoneId": "Z2IFOLAFXWL04F",
 "DNSName": "CorpExampleCom-NLB-3afe296bf4ba80d4.elb.region-id.amazonaws.com",
 "EvaluateTargetHealth": true
 }
 }
 }
]
}
```

- b. Soumettez la modification du jeu d'enregistrements à la zone hébergée à l'aide de l'ID de zone hébergée.

```
$ aws route53 change-resource-record-sets --hosted-zone-id Z09020002B5MZQNXSJUB \
 --change-batch file://recordset-delete.json
{
 "ChangeInfo": {
 "Id": "/change/C04853642A0TH2TJ5NLNI",
 "Status": "PENDING",
 "SubmittedAt": "2022-05-05T14:25:51.046000+00:00"
 }
}
```

- c. Supprimez la zone hébergée.

```
$ aws route53 delete-hosted-zone --id Z09020002B5MZQNXSJUB
{
```

```
"ChangeInfo": {
 "Id": "/change/C0468051QFABTVHMDEG9",
 "Status": "PENDING",
 "SubmittedAt": "2022-05-05T14:26:13.814000+00:00"
}
```

3. Supprimez l'écouteur LB.

```
$ aws elbv2 delete-listener \
 --listener-arn arn:aws:elasticloadbalancing:region-id:123456789012:listener/net/
 CorpExampleCom-NLB/3afe296bf4ba80d4/a8f9d97318743d4b --region region-id
```

4. Supprimez le groupe cible.

```
$ aws elbv2 delete-target-group \
 --target-group-arn arn:aws:elasticloadbalancing:region-
 id:123456789012:targetgroup/CorpExampleCom-Targets/44577c583b695e81 --
 region region-id
```

5. Supprimez l'équilibreur de charge.

```
$ aws elbv2 delete-load-balancer \
 --load-balancer-arn arn:aws:elasticloadbalancing:region-
 id:123456789012:loadbalancer/net/CorpExampleCom-NLB/3afe296bf4ba80d4 --
 region region-id
```

6. Supprimez la politique utilisée par le cluster pour lire le certificat depuis Secrets Manager.

```
$ aws iam delete-policy --policy-arn arn:aws:iam::123456789012:policy/
 ReadCertExample
```

7. Supprimez le secret qui contient le certificat de domaine.

```
$ aws secretsmanager delete-secret \
 --secret-id arn:aws:secretsmanager:region-id:123456789012:secret:example-
 cert-123abc \
 --region region-id
{
 "ARN": "arn:aws:secretsmanager:region-id:123456789012:secret:example-cert-123abc",
 "Name": "example-cert",
 "DeletionDate": "2022-06-04T16:27:36.183000+02:00"
```

```
}
```

8. Supprimez le certificat d'ACM.

```
$ aws acm delete-certificate \
 --certificate-arn arn:aws:acm:region-
id:123456789012:certificate/343db133-490f-4077-b8d4-3da5bfd89e72 --region region-id
```

9. Supprimez les ressources Active Directory (AD).

a. Obtenez les identifiants de ressources suivants à partir de la sortie du script python `ad.py` :

- IDENTIFIANT DE L'ANNONCE
- ID de sous-réseau AD
- AJOUTER UN IDENTIFIANT VPC

b. Supprimez le répertoire en exécutant la commande suivante.

```
$ aws ds delete-directory --directory-id d-abcdef0123456789 --region region-id
{
 "DirectoryId": "d-abcdef0123456789"
}
```

c. Répertoriez les groupes de sécurité du VPC.

```
$ aws ec2 describe-security-groups --filters '[{"Name":"vpc-id","Values":
["vpc-07614ade95ebad1bc"]}]' --region region-id
```

d. Supprimez le groupe de sécurité personnalisé.

```
$ aws ec2 delete-security-group --group-id sg-021345abcdef6789 --region region-
id
```

e. Supprimer les sous-réseaux.

```
$ aws ec2 delete-subnet --subnet-id subnet-1234567890abcdef --region region-id
```

```
$ aws ec2 delete-subnet --subnet-id subnet-021345abcdef6789 --region region-id
```

f. Décrivez la passerelle Internet.

```
$ aws ec2 describe-internet-gateways \

```

```

--filters Name=attachment.vpc-id,Values=vpc-021345abcdef6789 \
--region region-id
{
 "InternetGateways": [
 {
 "Attachments": [
 {
 "State": "available",
 "VpcId": "vpc-021345abcdef6789"
 }
],
 "InternetGatewayId": "igw-1234567890abcdef",
 "OwnerId": "123456789012",
 "Tags": []
 }
]
}

```

- g. Détachez la passerelle Internet.

```

$ aws ec2 detach-internet-gateway \
 --internet-gateway-id igw-1234567890abcdef \
 --vpc-id vpc-021345abcdef6789 \
 --region region-id

```

- h. Supprimez la passerelle Internet.

```

$ aws ec2 delete-internet-gateway \
 --internet-gateway-id igw-1234567890abcdef \
 --region region-id

```

- i. Supprimer le VPC.

```

$ aws ec2 delete-vpc \
 --vpc-id vpc-021345abcdef6789 \
 --region region-id

```

- j. Supprimez le secret qui contient le ReadOnlyUser mot de passe.

```

$ aws secretsmanager delete-secret \
 --secret-id arn:aws:secretsmanager:region-
id:123456789012:secret:ADSecretPassword-1234" \

```

```
--region region-id
```

## Configuration du chiffrement du stockage partagé à l'aide d'une AWS KMS clé

Découvrez comment configurer une AWS KMS clé gérée par le client pour chiffrer et protéger vos données dans les systèmes de stockage de fichiers en cluster configurés pour AWS ParallelCluster.

Lorsque vous utilisez l'interface de ligne de commande (CLI) ou l'API, vous ne payez que pour les AWS ressources créées lorsque vous créez ou mettez à jour AWS ParallelCluster des images et des clusters. Pour plus d'informations, consultez [AWS services utilisés par AWS ParallelCluster](#).

L'AWS ParallelCluster interface utilisateur repose sur une architecture sans serveur et vous pouvez l'utiliser dans la catégorie AWS Free Tier dans la plupart des cas. Pour plus d'informations, consultez [AWS ParallelCluster Coûts d'interface utilisateur](#).

AWS ParallelCluster prend en charge les options de configuration de stockage partagé suivantes :

- [SharedStorage](#) / [EbsSettings](#) / [KmsKeyId](#)
- [SharedStorage](#) / [EfsSettings](#) / [KmsKeyId](#)
- [SharedStorage](#) / [FsxLustreSettings](#) / [KmsKeyId](#)

Vous pouvez utiliser ces options pour fournir une AWS KMS clé gérée par le client pour le chiffrement des systèmes de stockage partagé Amazon EBS, Amazon EFS et FSx for Lustre. Pour les utiliser, vous devez créer et configurer une politique IAM pour les éléments suivants :

- [HeadNode](#) / [Iam](#) / [AdditionalIamPolicies](#) / [Policy](#)
- [Scheduler](#) / [SlurmQueues](#) / [Iam](#) / [AdditionalIamPolicies](#) / [Policy](#)

### Prérequis

- AWS ParallelCluster [est installé](#).
- Le AWS CLI [est installé et configuré](#).
- Vous possédez une paire de [clés Amazon EC2](#).
- Vous disposez d'un rôle IAM doté des [autorisations](#) requises pour exécuter la [pcluster](#) CLI.

## Rubriques

- [Création de la stratégie](#)
- [Configuration et création du cluster](#)

## Création de la stratégie

Créez une stratégie.

1. Accédez à la console IAM : <https://console.aws.amazon.com/iam/home>.
2. Choisissez Politiques (Politiques).
3. Choisissez Créer une politique.
4. Choisissez l'onglet JSON et collez la politique suivante. Assurez-vous de remplacer toutes les occurrences de **123456789012** par votre Compte AWS identifiant et la clé Amazon Resource Name (ARN) et Région AWS par votre propre identifiant.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "kms:DescribeKey",
 "kms:ReEncrypt*",
 "kms:CreateGrant",
 "kms:Decrypt"
],
 "Resource": [
 "arn:aws:kms:region-id:123456789012:key/abcd1234-ef56-gh78-ij90-
 abcd1234efgh5678"
]
 }
]
}
```

5. Pour ce didacticiel, nommez la politique `ParallelClusterKmsPolicy`, puis choisissez Create Policy.
6. Prenez note de l'ARN de la politique. Vous en avez besoin pour configurer votre cluster.

## Configuration et création du cluster

Voici un exemple de configuration de cluster qui inclut un système de fichiers partagé Amazon Elastic Block Store avec chiffrement.

```
Region: eu-west-1
Image:
 Os: alinux2
HeadNode:
 InstanceType: t2.micro
 Networking:
 SubnetId: subnet-abcdef01234567890
 Ssh:
 KeyName: my-ssh-key
 Iam:
 AdditionalIamPolicies:
 - Policy: arn:aws:iam::123456789012:policy/ParallelClusterKmsPolicy
Scheduling:
 Scheduler: slurm
 SlurmQueues:
 - Name: q1
 ComputeResources:
 - Name: t2micro
 InstanceType: t2.micro
 MinCount: 0
 MaxCount: 10
 Networking:
 SubnetIds:
 - subnet-abcdef01234567890
 Iam:
 AdditionalIamPolicies:
 - Policy: arn:aws:iam::123456789012:policy/ParallelClusterKmsPolicy
SharedStorage:
 - MountDir: /shared/ebs1
 Name: shared-ebs1
 StorageType: Ebs
 EbsSettings:
 Encrypted: True
 KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

Remplacez les éléments en rouge par vos propres valeurs. Créez ensuite un cluster qui utilise votre AWS KMS clé pour chiffrer vos données dans Amazon EBS.

La configuration est similaire pour les systèmes de fichiers Amazon EFS et FSx for Lustre.

La SharedStorage configuration d'Amazon EFS est la suivante.

```
...
SharedStorage:
 - MountDir: /shared/efs1
 Name: shared-efs1
 StorageType: Efs
 EfsSettings:
 Encrypted: True
 KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

La configuration de FSx for SharedStorage Lustre est la suivante.

```
...
SharedStorage:
 - MountDir: /shared/fsx1
 Name: shared-fsx1
 StorageType: FsxLustre
 FsxLustreSettings:
 StorageCapacity: 1200
 DeploymentType: PERSISTENT_1
 PerUnitStorageThroughput: 200
 KmsKeyId: abcd1234-ef56-gh78-ij90-abcd1234efgh5678
```

## Exécution de tâches dans un cluster en mode de files d'attente multiples

Ce didacticiel explique comment exécuter votre premier « Hello World » job en AWS ParallelCluster [mode file d'attente multiple](#).

Lorsque vous utilisez l'interface de ligne de commande (CLI) ou l'API, vous ne payez que pour les AWS ressources créées lorsque vous créez ou mettez à jour AWS ParallelCluster des images et des clusters. Pour plus d'informations, consultez [AWS services utilisés par AWS ParallelCluster](#).

L' AWS ParallelCluster interface utilisateur repose sur une architecture sans serveur et vous pouvez l'utiliser dans la catégorie AWS Free Tier dans la plupart des cas. Pour plus d'informations, consultez [AWS ParallelClusterCoûts d'interface utilisateur](#).



## Prérequis

- AWS ParallelCluster [est installé](#).
- Le AWS CLI [est installé et configuré](#).
- Vous possédez une paire de [clés Amazon EC2](#).
- Vous disposez d'un rôle IAM doté des [autorisations](#) requises pour exécuter la [pcluster](#) CLI.

## Configurez votre cluster

Tout d'abord, vérifiez qu' AWS ParallelCluster il est correctement installé en exécutant la commande suivante.

```
$ pcluster version
```

Pour plus d'informations sur `pcluster version`, consultez [pcluster version](#).

Cette commande renvoie la version en cours d'exécution de AWS ParallelCluster.

Exécutez ensuite `pcluster configure` pour générer un fichier de configuration de base. Suivez toutes les instructions qui suivent cette commande.

```
$ pcluster configure --config multi-queue-mode.yaml
```

Pour plus d'informations sur la commande `pcluster configure`, consultez [pcluster configure](#).

Une fois cette étape terminée, un fichier de configuration de base nommé `multi-queue-mode.yaml` apparaît. Ce fichier contient une configuration de base du cluster.

À l'étape suivante, vous modifiez votre nouveau fichier de configuration et lancez un cluster avec plusieurs files d'attente.

### Note

Certaines instances utilisées dans ce didacticiel ne sont pas éligibles au niveau gratuit.

Pour ce didacticiel, modifiez votre fichier de configuration pour qu'il corresponde à la configuration suivante. Les éléments surlignés en rouge représentent les valeurs de votre fichier de configuration. Gardez vos propres valeurs.

```
Region: region-id
Image:
 Os: alinux2
HeadNode:
 InstanceType: c5.xlarge
Networking:
 SubnetId: subnet-abcdef01234567890
Ssh:
 KeyName: yourkeypair
Scheduling:
 Scheduler: slurm
SlurmQueues:
- Name: spot
 ComputeResources:
- Name: c5xlarge
 InstanceType: c5.xlarge
 MinCount: 1
 MaxCount: 10
- Name: t2micro
 InstanceType: t2.micro
 MinCount: 1
 MaxCount: 10
Networking:
 SubnetIds:
- subnet-abcdef01234567890
- Name: ondemand
 ComputeResources:
- Name: c52xlarge
 InstanceType: c5.2xlarge
 MinCount: 0
 MaxCount: 10
Networking:
 SubnetIds:
- subnet-021345abcdef6789
```

## Créer votre cluster

Créez un cluster nommé `multi-queue-cluster` fonction de votre fichier de configuration.

```
$ pcluster create-cluster --cluster-name multi-queue-cluster --cluster-configuration
multi-queue-mode.yaml
{
 "cluster": {
 "clusterName": "multi-queue-cluster",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/
multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
 "region": "eu-west-1",
 "version": "3.7.0",
 "clusterStatus": "CREATE_IN_PROGRESS"
 }
}
```

Pour plus d'informations sur la commande `pcluster create-cluster`, consultez [pcluster create-cluster](#).

Pour vérifier l'état du cluster, exécutez la commande suivante.

```
$ pcluster list-clusters
{
 "cluster": {
 "clusterName": "multi-queue-cluster",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/
multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
 "region": "eu-west-1",
 "version": "3.7.0",
 "clusterStatus": "CREATE_IN_PROGRESS"
 }
}
```

Lorsque le cluster est créé, le `clusterStatus` champ s'affiche `CREATE_COMPLETE`.

## Connectez-vous au nœud principal

Utilisez votre fichier de clé SSH privée pour vous connecter au nœud principal.

```
$ pcluster ssh --cluster-name multi-queue-cluster -i ~/path/to/yourkeyfile.pem
```

Pour plus d'informations sur `pcluster ssh`, consultez [pcluster ssh](#).

Une fois connecté, exécutez la `sinfo` commande pour vérifier que les files d'attente de votre planificateur sont installées et configurées.

Pour plus d'informationssinfo, consultez [sinfo](#) dans la Slurmdocumentation.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
spot* up infinite 18 idle~ spot-dy-c5xlarge-[1-9],spot-dy-t2micro-[1-9]
spot* up infinite 2 idle spot-st-c5xlarge-1,spot-st-t2micro-1
ondemand up infinite 10 idle~ ondemand-dy-c52xlarge-[1-10]
```

La sortie indique que vous disposez d'un `t2.micro` et d'un nœud de `c5.xlarge` calcul dans l'`idle` état qui sont disponibles dans votre cluster.

Les autres nœuds sont tous en état d'économie d'énergie, indiqué par le `~` suffixe dans l'état du nœud, et aucune instance Amazon EC2 ne les prend en charge. La file d'attente par défaut est indiquée par un `*` suffixe après son nom. `spotest` votre file d'attente de tâches par défaut.

## Exécuter le job en mode file d'attente multiple

Ensuite, essayez d'exécuter une tâche pour dormir pendant un moment. La tâche génère ensuite son propre nom d'hôte. Assurez-vous que ce script peut être exécuté par l'utilisateur actuel.

```
$ tee <<EOF hellojob.sh
#!/bin/bash
sleep 30
echo "Hello World from \$(hostname)"
EOF

$ chmod +x hellojob.sh
$ ls -l hellojob.sh
-rwxrwxr-x 1 ec2-user ec2-user 57 Sep 23 21:57 hellojob.sh
```

Soumettez le travail à l'aide de la `sbatch` commande. Demandez deux nœuds pour cette tâche avec l'`-N 2` option, et vérifiez que la tâche est correctement soumise. Pour plus d'informationssbatch, consultez la [sbatch](#) documentation de Slurm.

```
$ sbatch -N 2 --wrap "srun hellojob.sh"
Submitted batch job 1
```

Vous pouvez consulter votre file d'attente et vérifier l'état de la tâche à l'aide de la `squeue` commande. Comme vous n'avez pas spécifié de file d'attente spécifique, la file d'attente par défaut (spot) est utilisée. Pour plus d'informations `squeue`, consultez [squeue](#) la Slurmdocumentation.

```
$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
 1 spot wrap ec2-user R 0:10 2 spot-st-c5xlarge-1,spot-st-t2micro-1
```

La sortie indique que la tâche est actuellement en cours d'exécution. Attendez que le travail soit terminé. Cela prend environ 30 secondes. Ensuite, courez `squeue` à nouveau.

```
$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
```

Maintenant que les tâches de la file d'attente sont toutes terminées, recherchez le fichier de sortie nommé `slurm-1.out` dans votre répertoire actuel.

```
$ cat slurm-1.out
Hello World from spot-st-t2micro-1
Hello World from spot-st-c5xlarge-1
```

Le résultat indique que la tâche s'est exécutée avec succès sur les `spot-st-c5xlarge-1` nœuds `spot-st-t2micro-1` et.

Soumettez maintenant le même travail en spécifiant des contraintes pour des instances spécifiques à l'aide des commandes suivantes.

```
$ sbatch -N 3 -p spot -C "[c5.xlarge*1&t2.micro*2]" --wrap "srun hellojob.sh"
Submitted batch job 2
```

Vous avez utilisé ces paramètres pour `sbatch` :

- `-N 3`— demande trois nœuds.
- `-p spot`— soumet la tâche à la `spot` file d'attente. Vous pouvez également soumettre une tâche à la `ondemand` file d'attente en spécifiant `-p ondemand`.
- `-C "[c5.xlarge*1&t2.micro*2]"`— spécifie les contraintes de nœud spécifiques pour cette tâche. Cela demande qu'un `c5.xlarge` nœud et deux `t2.micro` nœuds soient utilisés pour cette tâche.

Exécutez la `sinfo` commande pour afficher les nœuds et les files d'attente. Les files d'attente AWS ParallelCluster entrantes sont appelées partitions entrantes. Slurm

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
spot* up infinite 1 alloc# spot-dy-t2micro-1
spot* up infinite 17 idle~ spot-dy-c5xlarge-[2-10],spot-dy-t2micro-[2-9]
spot* up infinite 1 mix spot-st-c5xlarge-1
spot* up infinite 1 alloc spot-st-t2micro-1
ondemand up infinite 10 idle~ ondemand-dy-c52xlarge-[1-10]
```

Les nœuds s'allument. Cela est indiqué par le `#` suffixe sur l'état du nœud. Exécutez la `squeue` commande pour afficher les informations relatives aux tâches du cluster.

```
$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
 2 spot wrap ec2-user CF 0:04 3 spot-dy-c5xlarge-1,spot-dy-
t2micro-1,spot-st-t2micro-1
```

Votre tâche est à l'état CF (CONFIGURING) et attend que les instances s'agrandissent et rejoignent le cluster.

Au bout de trois minutes environ, les nœuds sont disponibles et la tâche passe à l'état R (RUNNING).

```
$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
 2 spot wrap ec2-user R 0:07 3 spot-dy-t2micro-1,spot-st-
c5xlarge-1,spot-st-t2micro-1
```

La tâche est terminée et les trois nœuds sont dans l'`idle` état.

```
$ squeue
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
spot* up infinite 17 idle~ spot-dy-c5xlarge-[1-9],spot-dy-t2micro-[2-9]
spot* up infinite 3 idle spot-dy-t2micro-1,spot-st-c5xlarge-1,spot-st-
t2micro-1
ondemand up infinite 10 idle~ ondemand-dy-c52xlarge-[1-10]
```

Ensuite, une fois qu'aucune tâche ne reste dans la file d'attente, `slurm-2.out` recherchez-la dans votre répertoire local.

```
$ cat slurm-2.out
Hello World from spot-st-t2micro-1
Hello World from spot-dy-t2micro-1
Hello World from spot-st-c5xlarge-1
```

Il s'agit de l'état final du cluster.

```
$ sinfo
PARTITION AVAIL TIMELIMIT NODES STATE NODELIST
spot* up infinite 17 idle~ spot-dy-c5xlarge-[1-9],spot-dy-t2micro-[2-9]
spot* up infinite 3 idle spot-dy-t2micro-1,spot-st-c5xlarge-1,spot-st-
t2micro-1
ondemand up infinite 10 idle~ ondemand-dy-c52xlarge-[1-10]
```

Après vous être déconnecté du cluster, vous pouvez le nettoyer en exécutant `pcluster delete-cluster`. Pour plus d'informations, consultez [pcluster list-clusters](#) et [pcluster delete-cluster](#).

```
$ pcluster list-clusters
{
 "clusters": [
 {
 "clusterName": "multi-queue-cluster",
 "cloudformationStackStatus": "CREATE_COMPLETE",
 "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
 "region": "eu-west-1",
 "version": "3.1.4",
 "clusterStatus": "CREATE_COMPLETE"
 }
]
}
$ pcluster delete-cluster -n multi-queue-cluster
{
 "cluster": {
 "clusterName": "multi-queue-cluster",
 "cloudformationStackStatus": "DELETE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:123456789012:stack/multi-queue-cluster/1234567-abcd-0123-def0-abcdef0123456",
```

```
"region": "eu-west-1",
"version": "3.1.4",
"clusterStatus": "DELETE_IN_PROGRESS"
}
}
```

## Utilisation de l' AWS ParallelCluster API

Dans ce didacticiel, vous allez créer et tester l'API avec [Amazon API Gateway](#) et un AWS ParallelCluster CloudFormation modèle. Ensuite, vous utilisez l'exemple de client disponible sur GitHub pour utiliser l'API. Pour en savoir plus sur l'utilisation de l'API, consultez le document [AWS ParallelCluster API](#).

Ce didacticiel est extrait de l'atelier [HPC pour les clients du secteur public](#).

Lorsque vous utilisez l'interface de ligne de commande (CLI) ou l'API, vous ne payez que pour les AWS ressources créées lorsque vous créez ou mettez à jour AWS ParallelCluster des images et des clusters. Pour plus d'informations, consultez [AWS services utilisés par AWS ParallelCluster](#).

L' AWS ParallelCluster interface utilisateur repose sur une architecture sans serveur et vous pouvez l'utiliser dans la catégorie AWS Free Tier dans la plupart des cas. Pour plus d'informations, consultez [AWS ParallelCluster Coûts d'interface utilisateur](#).

### Prérequis

- AWS CLI II est [installé](#) et configuré dans votre environnement informatique.
- AWS ParallelCluster est installé dans un environnement virtuel. Pour plus d'informations, consultez la section [Installation AWS ParallelCluster dans un environnement virtuel](#).
- Vous possédez une paire de [clés Amazon EC2](#).
- Vous disposez d'un rôle IAM doté des [autorisations](#) requises pour exécuter la [pcluster](#) CLI.

## Étape 1 : créer l'API avec Amazon API Gateway

Restez dans votre répertoire d'utilisateurs personnel et activez votre environnement virtuel :

1. Installez un processeur de ligne de commande JSON utile.



```
$ sudo yum groupinstall -y "Development Tools"
sudo yum install -y jq python3-devel
```

2. Exécutez la commande suivante pour obtenir votre AWS ParallelCluster version et l'affecter à une variable d'environnement.

```
$ PCLUSTER_VERSION=$(pcluster version | jq -r '.version')
echo "export PCLUSTER_VERSION=${PCLUSTER_VERSION}" | tee -a ~/.bashrc
```

3. Créez une variable d'environnement et attribuez-lui votre identifiant de région.

```
$ export AWS_DEFAULT_REGION="us-east-1"
echo "export AWS_DEFAULT_REGION=${AWS_DEFAULT_REGION}" | tee -a ~/.bashrc
```

4. Exécutez les commandes suivantes pour déployer l'API.

```
API_STACK_NAME="pc-api-stack"
echo "export API_STACK_NAME=${API_STACK_NAME}" | tee -a ~/.bashrc
```

```
aws cloudformation create-stack \
 --region ${AWS_DEFAULT_REGION} \
 --stack-name ${API_STACK_NAME} \
 --template-url https://${AWS_DEFAULT_REGION}-aws-parallelcluster.s3.
 ${AWS_DEFAULT_REGION}.amazonaws.com/parallelcluster/${PCLUSTER_VERSION}/api/
 parallelcluster-api.yaml \
 --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND \
 --parameters ParameterKey=EnableIamAdminAccess,ParameterValue=true

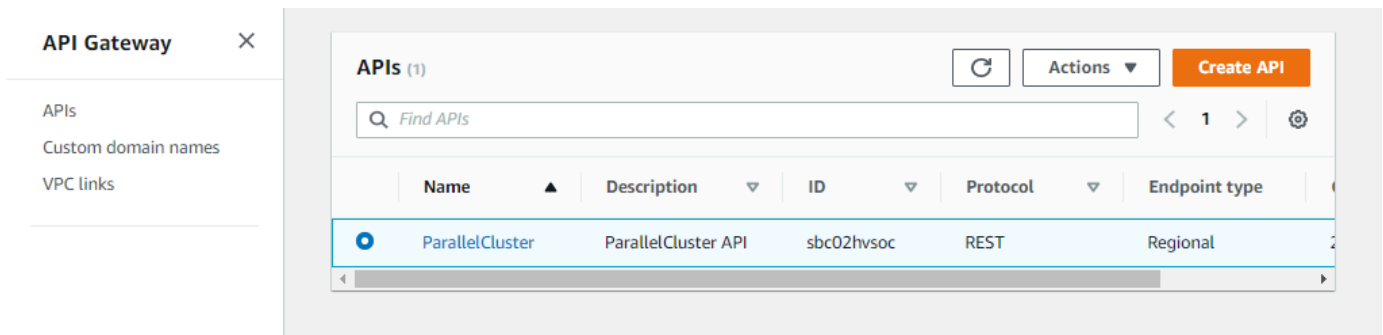
 {
 "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/my-api-
 stack/abcd1234-ef56-gh78-ei90-1234abcd5678"
 }
```

Une fois le processus terminé, passez à l'étape suivante.

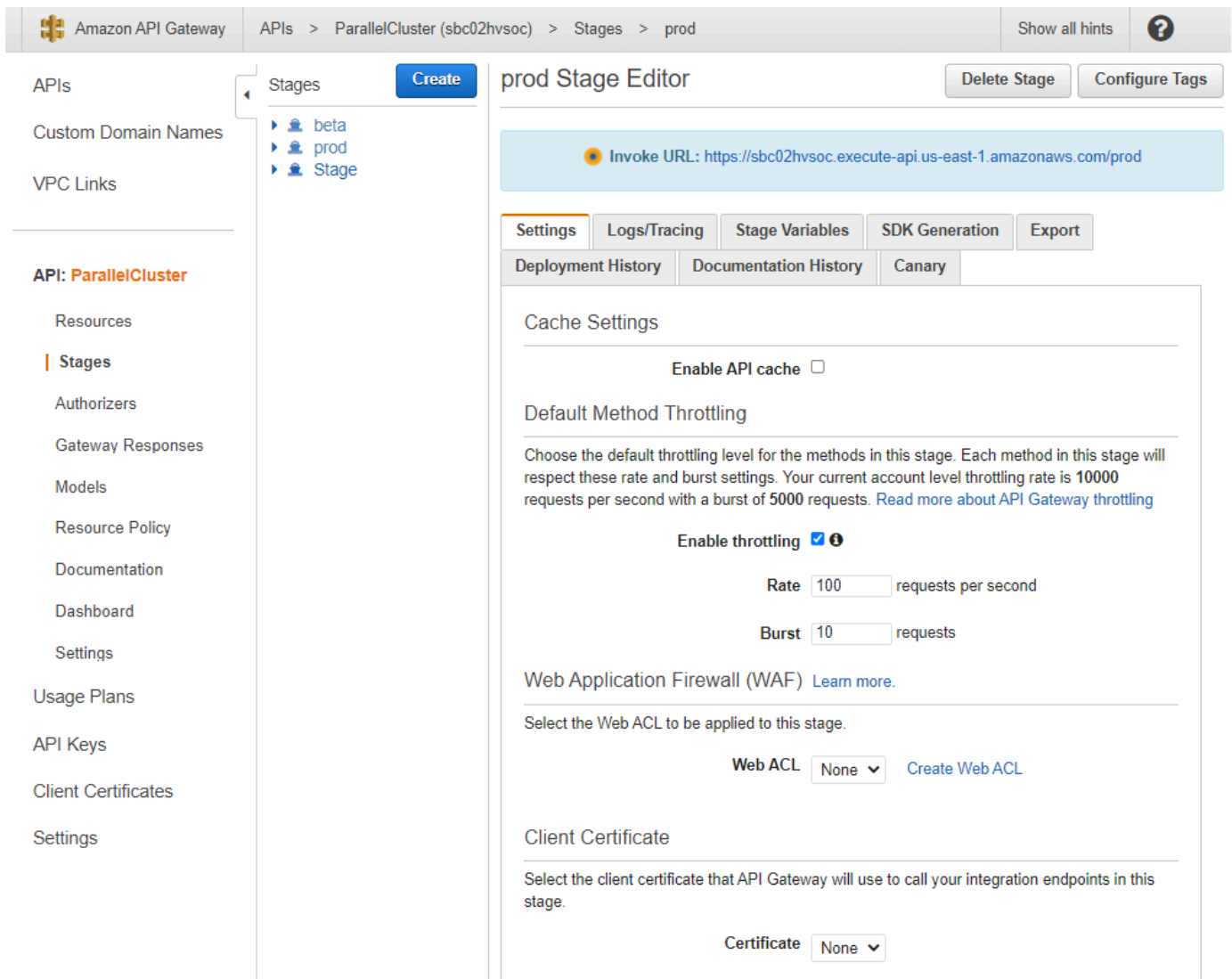
## Étape 2 : tester l'API dans la console Amazon API Gateway

1. Connectez-vous au AWS Management Console.
2. Accédez à la console [Amazon API Gateway](#).

### 3. Choisissez le déploiement de votre API.



### 4. Choisissez Stages, puis sélectionnez un stage.



5. Notez l'URL fournie par API Gateway pour accéder à votre API ou pour l'appeler. Il est surligné en bleu.

6. Choisissez Ressources, puis sélectionnez **GET** sous **/clusters**.

7. Choisissez l'icône TEST, puis faites défiler l'écran vers le bas et choisissez l'icône TEST.

The screenshot shows the AWS ParallelCluster console interface for testing an API endpoint. At the top, the breadcrumb trail reads: APIs > ParallelCluster (sbc02hvsoc) > Resources > /v3/clusters (ulfkw2) > GET. Below this, the main area is titled "/v3/clusters - GET - Method Execution". On the left, there is a "Resources" sidebar with a tree view showing the API structure, where "/v3/clusters/GET" is selected. In the center, a vertical box labeled "Client" contains a "TEST" button with a lightning bolt icon. To the right of the client, there are two panels: "Method Request" and "Method Response". The "Method Request" panel shows the following details: Auth: AWS IAM, ARN: arn:aws:execute-api:us-east-1:123456789012:sbc02hvsoc/\*/GET/v3/clusters, and Query Strings: region, nextToken, clusterStatus. The "Method Response" panel contains the text "Select an integration response." Arrows indicate the flow of data from the client to the request and back to the response.

La réponse à votre message /clusters GET apparaît.

APIs > ParallelCluster (sbc02hvsoc) > Resources > /v3/clusters (ulfkw2) > GET

Show all hints ?

Resources Actions

← Method Execution /v3/clusters - GET - Method Test

Make a test call to your method. When you make a test call, API Gateway skips authorization and directly invokes your method

**Path**  
No path parameters exist for this resource. You can define path parameters by using the syntax `{myPathParam}` in a resource path.

**Request:** /v3/clusters  
**Status:** 200  
**Latency:** 3203 ms  
**Response Body**

**Query Strings**  
{clusters}

**Headers**  
{clusters}  
Use a colon (:) to separate header name and value, and new lines to declare multiple headers. eg. `Accept:application/json`.

**Stage Variables**  
No stage variables exist for this method.

**Client Certificate**  
No client certificates have been generated.

```
{
 "clusters": [
 {
 "cloudformationStackArn": "arn:aws:cloudformation:us-east-1:123456789012:stack/test-cluster/4450d850-b684-11ec-84a7-0a047567c9f3",
 "cloudformationStackStatus": "CREATE_COMPLETE",
 "clusterName": "test-cluster",
 "clusterStatus": "CREATE_COMPLETE",
 "region": "us-east-1",
 "version": "3.1.2"
 }
]
}
```

**Response Headers**

```
{"Content-Length": "360", "X-Amzn-Trace-Id": "Root=1-62686455-c1cf243417b2721e33822ac5;Sampled=1", "Content-Type": "application/json"}
```

Logs

### Étape 3 : préparer et tester un exemple de client pour appeler l'API

Clonez le code AWS ParallelCluster source `api` dans le répertoire et installez les bibliothèques clientes Python. `cd`

1. 

```
$ git clone -b v${PCLUSTER_VERSION} https://github.com/aws/aws-parallelcluster aws-parallelcluster-v${PCLUSTER_VERSION}
cd aws-parallelcluster-v${PCLUSTER_VERSION}/api
```

```
$ pip3 install client/src
```

2. Retournez à votre répertoire d'utilisateurs personnel.
3. Exportez l'URL de base d'API Gateway que le client utilise lors de l'exécution.

```
$ export PCLUSTER_API_URL=$(aws cloudformation describe-stacks
 --stack-name ${API_STACK_NAME} --query 'Stacks[0].Outputs[?
OutputKey==`ParallelClusterApiInvokeUrl`].OutputValue' --output text)
echo "export PCLUSTER_API_URL=${PCLUSTER_API_URL}" |tee -a ~/.bashrc
```

4. Exportez un nom de cluster que le client utilise pour créer un cluster.

```
$ export CLUSTER_NAME="test-api-cluster"
echo "export CLUSTER_NAME=${CLUSTER_NAME}" |tee -a ~/.bashrc
```

5. Exécutez les commandes suivantes pour stocker les informations d'identification utilisées par le client d'exemple pour accéder à l'API.

```
$ export PCLUSTER_API_USER_ROLE=$(aws cloudformation describe-
stacks --stack-name ${API_STACK_NAME} --query 'Stacks[0].Outputs[?
OutputKey==`ParallelClusterApiUserRole`].OutputValue' --output text)
echo "export PCLUSTER_API_USER_ROLE=${PCLUSTER_API_USER_ROLE}" |tee -a ~/.bashrc
```

## Étape 4 : Copier le script de code client et exécuter des tests de cluster

1. Copiez l'exemple de code client suivant `test_pcluster_client.py` dans votre répertoire utilisateur personnel. Le code client demande les opérations suivantes :
  - Créez le cluster .
  - Décrivez le cluster.
  - Répertoriez les clusters.
  - Décrivez le parc informatique.
  - Décrivez les instances de cluster.

```
Copyright 2021 Amazon.com, Inc. or its affiliates. All Rights Reserved.
SPDX-License-Identifier: MIT-0
#
Permission is hereby granted, free of charge, to any person obtaining a copy of
this
```

```
software and associated documentation files (the "Software"), to deal in the
Software
without restriction, including without limitation the rights to use, copy,
modify,
merge, publish, distribute, sublicense, and/or sell copies of the Software, and
to
permit persons to whom the Software is furnished to do so.
#
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
#
Author: Evan F. Bollig (Github: bollig)

import time, datetime
import os
import pcluster_client
from pprint import pprint
from pcluster_client.api import (
 cluster_compute_fleet_api,
 cluster_instances_api,
 cluster_operations_api
)
from pcluster_client.model.create_cluster_request_content import
 CreateClusterRequestContent
from pcluster_client.model.cluster_status import ClusterStatus
region=os.environ.get("AWS_DEFAULT_REGION")

Defining the host is optional and defaults to http://localhost
See configuration.py for a list of all supported configuration parameters.
configuration = pcluster_client.Configuration(
 host = os.environ.get("PCLUSTER_API_URL")
)
cluster_name=os.environ.get("CLUSTER_NAME")

Enter a context with an instance of the API client
with pcluster_client.ApiClient(configuration) as api_client:
 cluster_ops = cluster_operations_api.ClusterOperationsApi(api_client)
 fleet_ops = cluster_compute_fleet_api.ClusterComputeFleetApi(api_client)
```

```
instance_ops = cluster_instances_api.ClusterInstancesApi(api_client)

Create cluster
build_done = False
try:
 with open('cluster-config.yaml', encoding="utf-8") as f:
 body = CreateClusterRequestContent(cluster_name=cluster_name,
cluster_configuration=f.read())
 api_response = cluster_ops.create_cluster(body, region=region)
except pcluster_client.ApiException as e:
 print("Exception when calling create_cluster: %s\n" % e)
 build_done = True
time.sleep(60)

Confirm cluster status with describe_cluster
while not build_done:
 try:
 api_response = cluster_ops.describe_cluster(cluster_name,
region=region)
 pprint(api_response)
 if api_response.cluster_status == ClusterStatus('CREATE_IN_PROGRESS'):
 print('. . . working . . .', end='', flush=True)
 time.sleep(60)
 elif api_response.cluster_status == ClusterStatus('CREATE_COMPLETE'):
 print('READY!')
 build_done = True
 else:
 print('ERROR!!!!')
 build_done = True
 except pcluster_client.ApiException as e:
 print("Exception when calling describe_cluster: %s\n" % e)

List clusters
try:
 api_response = cluster_ops.list_clusters(region=region)
 pprint(api_response)
except pcluster_client.ApiException as e:
 print("Exception when calling list_clusters: %s\n" % e)

DescribeComputeFleet
try:
 api_response = fleet_ops.describe_compute_fleet(cluster_name,
region=region)
 pprint(api_response)
```

```

except pcluster_client.ApiException as e:
 print("Exception when calling compute fleet: %s\n" % e)

DescribeClusterInstances
try:
 api_response = instance_ops.describe_cluster_instances(cluster_name,
region=region)
 pprint(api_response)
except pcluster_client.ApiException as e:
 print("Exception when calling describe_cluster_instances: %s\n" % e)

```

## 2. Créez une configuration de cluster.

```
$ pcluster configure --config cluster-config.yaml
```

- La bibliothèque du client API détecte automatiquement les détails de configuration à partir de vos variables d'environnement (par exemple `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, ou `AWS_SESSION_TOKEN`) ou `$HOME/.aws`. La commande suivante fait passer votre rôle IAM actuel au rôle désigné `ParallelClusterApiUserRole`.

```
$ eval $(aws sts assume-role --role-arn ${PCLUSTER_API_USER_ROLE} --role-session-name ApiTestSession | jq -r '.Credentials | "export AWS_ACCESS_KEY_ID=\(.AccessKeyId)\nexport AWS_SECRET_ACCESS_KEY=\(.SecretAccessKey)\nexport AWS_SESSION_TOKEN=\(.SessionToken)\n"')
```

Erreur à surveiller :

Si vous voyez un message d'erreur similaire à ce qui suit, c'est que vous avez déjà supposé que le `ParallelClusterApiUserRole` votre `AWS_SESSION_TOKEN` a expiré.

```

An error occurred (AccessDenied) when calling the AssumeRole operation:
User: arn:aws:sts::XXXXXXXXXXXX:assumed-role/ParallelClusterApiUserRole-XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX/ApiTestSession
is not authorized to perform: sts:AssumeRole on resource:
arn:aws:iam::XXXXXXXXXXXX:role/ParallelClusterApiUserRole-XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX

```

Supprimez le rôle, puis réexécutez la `aws sts assume-role` commande pour utiliser le `ParallelClusterApiUserRole`.

```
$ unset AWS_SESSION_TOKEN
```



```
unset AWS_SECRET_ACCESS_KEY
unset AWS_ACCESS_KEY_ID
```

Pour fournir à vos utilisateurs actuels les autorisations d'accès à l'API, vous devez [développer la politique de ressources](#).

4. Exécutez la commande suivante pour tester l'exemple de client.

```
$ python3 test_pcluster_client.py
{'cluster_configuration': 'Region: us-east-1\n'
 'Image:\n'
 ' Os: alinux2\n'
 'HeadNode:\n'
 ' InstanceType: t2.micro\n'
 ' Networking . . . :\n'
 ' SubnetId: subnet-1234567890abcdef0\n'
 ' Ssh:\n'
 ' KeyName: adpc\n'
 'Scheduling:\n'
 ' Scheduler: slurm\n'
 ' SlurmQueues:\n'
 ' - Name: queue1\n'
 ' ComputeResources:\n'
 ' - Name: t2micro\n'
 ' InstanceType: t2.micro\n'
 ' MinCount: 0\n'
 ' MaxCount: 10\n'
 ' Networking . . . :\n'
 ' SubnetIds:\n'
 ' - subnet-1234567890abcdef0\n',
 'cluster_name': 'test-api-cluster'}
{'cloud_formation_stack_status': 'CREATE_IN_PROGRESS',
 'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
 'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-not-delete...'},
 'cluster_name': 'test-api-cluster',
 'cluster_status': 'CREATE_IN_PROGRESS',
 'compute_fleet_status': 'UNKNOWN',
 'creation_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000, tzinfo=tzlocal()),
 'last_updated_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000, tzinfo=tzlocal()),
```

```

'region': 'us-east-1',
'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
'version': '3.1.3'
.
.
.
. . . working . . . {'cloud_formation_stack_status': 'CREATE_COMPLETE',
'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-
not-delete...'},
'cluster_name': 'test-api-cluster',
'cluster_status': 'CREATE_COMPLETE',
'compute_fleet_status': 'RUNNING',
'creation_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000,
tzinfo=tzlocal()),
'head_node': {'instance_id': 'i-abcdef01234567890',
'instance_type': 't2.micro',
'launch_time': datetime.datetime(2022, 4, 28, 16, 21, 46,
tzinfo=tzlocal()),
'private_ip_address': '172.31.27.153',
'public_ip_address': '52.90.156.51',
'state': 'running'},
'last_updated_time': datetime.datetime(2022, 4, 28, 16, 18, 47, 972000,
tzinfo=tzlocal()),
'region': 'us-east-1',
'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
'version': '3.1.3'
READY!

```

## Étape 5 : Copier le script de code client et supprimer le cluster

1. Copiez l'exemple de code client suivant dans `delete_cluster_client.py`. Le code client demande la suppression du cluster.

```

Copyright 2021 Amazon.com, Inc. or its affiliates. All Rights Reserved.
SPDX-License-Identifier: MIT-0
#
Permission is hereby granted, free of charge, to any person obtaining a copy of
this
software and associated documentation files (the "Software"), to deal in the
Software

```

```
without restriction, including without limitation the rights to use, copy,
modify,
merge, publish, distribute, sublicense, and/or sell copies of the Software, and
to
permit persons to whom the Software is furnished to do so.
#
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED,
INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
#
Author: Evan F. Bollig (Github: bollig)

import time, datetime
import os
import pcluster_client
from pprint import pprint
from pcluster_client.api import (
 cluster_compute_fleet_api,
 cluster_instances_api,
 cluster_operations_api
)
from pcluster_client.model.create_cluster_request_content import
 CreateClusterRequestContent
from pcluster_client.model.cluster_status import ClusterStatus
region=os.environ.get("AWS_DEFAULT_REGION")

Defining the host is optional and defaults to http://localhost
See configuration.py for a list of all supported configuration parameters.
configuration = pcluster_client.Configuration(
 host = os.environ.get("PCLUSTER_API_URL")
)
cluster_name=os.environ.get("CLUSTER_NAME")

Enter a context with an instance of the API client
with pcluster_client.ApiClient(configuration) as api_client:
 cluster_ops = cluster_operations_api.ClusterOperationsApi(api_client)

 # Delete the cluster
 gone = False
```

```

try:
 api_response = cluster_ops.delete_cluster(cluster_name, region=region)
except pcluster_client.ApiException as e:
 print("Exception when calling delete_cluster: %s\n" % e)
time.sleep(60)

Confirm cluster status with describe_cluster
while not gone:
 try:
 api_response = cluster_ops.describe_cluster(cluster_name,
region=region)
 pprint(api_response)
 if api_response.cluster_status == ClusterStatus('DELETE_IN_PROGRESS'):
 print('. . . working . . .', end='', flush=True)
 time.sleep(60)
 except pcluster_client.ApiException as e:
 gone = True
 print("DELETE COMPLETE or Exception when calling describe_cluster: %s
\n" % e)

```

2. Exécutez la commande suivante pour supprimer le cluster.

```

$ python3 delete_cluster_client.py
{'cloud_formation_stack_status': 'DELETE_IN_PROGRESS',
'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-
not-delete...'},
'cluster_name': 'test-api-cluster',
'cluster_status': 'DELETE_IN_PROGRESS',
'compute_fleet_status': 'UNKNOWN',
'creation_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'head_node': {'instance_id': 'i-abcdef01234567890',
'instance_type': 't2.micro',
'launch_time': datetime.datetime(2022, 4, 28, 16, 53, 48,
tzinfo=tzlocal()),
'private_ip_address': '172.31.17.132',
'public_ip_address': '34.201.100.37',
'state': 'running'},
'last_updated_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'region': 'us-east-1',
'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],

```

```
'version': '3.1.3'}
.
.
.
. . . working . . . {'cloud_formation_stack_status': 'DELETE_IN_PROGRESS',
'cloudformation_stack_arn': 'arn:aws:cloudformation:us-east-1:123456789012:stack/
test-api-cluster/abcd1234-ef56-gh78-ij90-1234abcd5678',
'cluster_configuration': {'url': 'https://parallelcluster-021345abcdef6789-v1-do-
not-delete...'},
'cluster_name': 'test-api-cluster',
'cluster_status': 'DELETE_IN_PROGRESS',
'compute_fleet_status': 'UNKNOWN',
'creation_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'last_updated_time': datetime.datetime(2022, 4, 28, 16, 50, 47, 943000,
tzinfo=tzlocal()),
'region': 'us-east-1',
'tags': [{'key': 'parallelcluster:version', 'value': '3.1.3'}],
'version': '3.1.3'}
. . . working . . . DELETE COMPLETE or Exception when calling describe_cluster:
(404)
Reason: Not Found
.
.
.
HTTP response body: {"message": "Cluster 'test-api-cluster' does not exist or
belongs to an incompatible ParallelCluster major version."}
```

3. Une fois les tests terminés, désactivez les variables d'environnement.

```
$ unset AWS_SESSION_TOKEN
unset AWS_SECRET_ACCESS_KEY
unset AWS_ACCESS_KEY_ID
```

## Étape 6 : Nettoyer

Vous pouvez utiliser le AWS Management Console ou AWS CLI pour supprimer votre API.

1. Dans la AWS CloudFormation console, choisissez la pile d'API, puis choisissez Supprimer.
2. Exécutez la commande suivante si vous utilisez le AWS CLI.

En utilisant AWS CloudFormation.

```
$ aws cloudformation delete-stack --stack-name ${API_STACK_NAME}
```

## Création d'un cluster avec Slurm comptabilité

Découvrez comment configurer et créer un cluster avec la Slurm gestion des comptes. Pour plus d'informations, consultez [Slurmcomptabilité avec AWS ParallelCluster](#).

Lorsque vous utilisez l'interface de ligne de commande (CLI) ou l'API, vous ne payez que pour les AWS ressources créées lorsque vous créez ou mettez à jour AWS ParallelCluster des images et des clusters. Pour plus d'informations, consultez [AWS services utilisés par AWS ParallelCluster](#).

L' AWS ParallelCluster interface utilisateur repose sur une architecture sans serveur et vous pouvez l'utiliser dans la catégorie AWS Free Tier dans la plupart des cas. Pour plus d'informations, consultez [AWS ParallelCluster Coûts d'interface utilisateur](#).

Dans ce didacticiel, vous allez utiliser un [modèle de création CloudFormation rapide \(us-east-1\) pour créer une base de données sans serveur pour MySQL. Amazon Aurora](#). Le modèle indique de CloudFormation créer tous les composants nécessaires pour déployer une base de données Amazon Aurora sans serveur sur le même VPC que le cluster. Le modèle crée également une configuration réseau et de sécurité de base pour la connexion entre le cluster et la base de données.

### Note

À partir de la version 3.3.0, AWS ParallelCluster prend en charge la Slurm comptabilité avec le paramètre de configuration du cluster [SlurmSettings/Database](#).

### Note

Le modèle de création rapide sert d'exemple. Ce modèle ne couvre pas tous les cas d'utilisation possibles d'un serveur de base de données de Slurm comptabilité. Il est de votre responsabilité de créer un serveur de base de données dont la configuration et la capacité sont adaptées à vos charges de travail de production.

## Prérequis :

- AWS ParallelCluster [est installé](#).
- Le AWS CLI [est installé et configuré](#).
- Vous possédez une paire de [clés Amazon EC2](#).
- Vous disposez d'un rôle IAM doté des [autorisations](#) requises pour exécuter la [pcluster](#) CLI.
- La région dans laquelle vous déployez le modèle de création rapide prend en charge Amazon Aurora MySQL serverless v2. Pour plus d'informations, consultez [Aurora Serverless v2 avec Aurora MySQL](#).

## Étape 1 : créer le VPC et les sous-réseaux pour AWS ParallelCluster

Pour utiliser le CloudFormation modèle fourni pour la base de données de Slurm comptabilité, le VPC du cluster doit être prêt. Vous pouvez le faire manuellement ou dans le cadre de la [Configuration et création d'un cluster à l'aide de l'interface de ligne de AWS ParallelCluster commande](#) procédure. Si vous l'avez déjà utilisé AWS ParallelCluster, vous disposez peut-être d'un VPC prêt pour le déploiement du cluster et du serveur de base de données.

## Étape 2 : Création de la pile de base de données

Utilisez le [modèle de création CloudFormation rapide \(us-east-1\) pour créer](#) une pile de bases de données pour la comptabilité. Slurm Le modèle nécessite les entrées suivantes :

- Informations d'identification du serveur de base de données, en particulier le nom d'utilisateur et le mot de passe de l'administrateur.
- Dimensionnement du cluster Amazon Aurora sans serveur. Cela dépend du chargement attendu du cluster.
- Les paramètres réseau, en particulier le VPC cible et les sous-réseaux ou blocs CIDR pour la création des sous-réseaux.

Sélectionnez les informations d'identification et la taille appropriées pour votre serveur de base de données. Pour les options de mise en réseau, vous devez utiliser le même VPC sur lequel le AWS ParallelCluster cluster est déployé. Vous pouvez créer les sous-réseaux de la base de données et les transmettre en entrée au modèle. Vous pouvez également fournir deux blocs d'adresse CIDR disjoints pour les deux sous-réseaux et laisser le CloudFormation modèle créer les deux sous-réseaux pour les blocs d'adresse CIDR. Assurez-vous que les blocs CIDR ne se chevauchent pas

avec les sous-réseaux existants. Si les blocs CIDR se chevauchent avec des sous-réseaux existants, la pile ne peut pas être créée.

La création du serveur de base de données prend plusieurs minutes.

## Étape 3 : créer un cluster avec la Slurm comptabilité activée

Le CloudFormation modèle fourni génère une CloudFormation pile avec certaines sorties définies. À partir du AWS Management Console, vous pouvez afficher les sorties dans l'onglet Sorties de la vue en CloudFormation pile. Pour activer la Slurm comptabilité, certaines des sorties suivantes doivent être utilisées dans le fichier de configuration du AWS ParallelCluster cluster :

- `DatabaseHost`: utilisé pour le paramètre de configuration [Uri](#) du cluster [SlurmSettingsDatabase](#).
- `DatabaseAdminUser`: utilisé pour la valeur du paramètre de configuration [UserName](#) du cluster [SlurmSettingsDatabase](#).
- `DatabaseSecretArn`: utilisé pour le paramètre de configuration [PasswordSecretArn](#) du cluster [SlurmSettingsDatabase](#).
- `DatabaseClientSecurityGroup`: il s'agit du groupe de sécurité attaché au nœud principal du cluster défini dans le paramètre de [SecurityGroups](#) configuration [HeadNodeNetworking](#).

Mettez à jour les Database paramètres du fichier de configuration de votre cluster avec les valeurs de sortie. Utilisez la [pcluster](#) CLI pour créer le cluster.

```
$ pcluster create-cluster -n cluster-3.x -c path/to/cluster-config.yaml
```

Une fois le cluster créé, vous pouvez commencer à utiliser des commandes de Slurm comptabilité telles que `sacctmgr` ou `sacct`.

## Création d'un cluster avec une Slurmdbd comptabilité externe

Découvrez comment configurer et créer un cluster avec une comptabilité externe Slurmdbd. Pour plus d'informations, consultez la section [SlurmComptabilité avec AWS ParallelCluster](#).

Lorsque vous utilisez l'interface de ligne de AWS ParallelCluster commande (CLI) ou l'API, vous ne payez que pour les AWS ressources créées lorsque vous créez ou mettez à jour AWS ParallelCluster



des images et des clusters. Pour plus d'informations, consultez la section [AWS Services utilisés par AWS ParallelCluster](#).

L'AWS ParallelCluster interface utilisateur repose sur une architecture sans serveur et vous pouvez l'utiliser dans AWS Free Tier cette catégorie dans la plupart des cas. Pour plus d'informations, consultez la section [Coûts de l'AWS ParallelCluster interface utilisateur](#).

Dans ce didacticiel, vous allez utiliser un modèle de AWS CloudFormation création rapide pour créer les composants nécessaires au déploiement d'une instance Slurmdbd sur le même VPC que le cluster. Le modèle crée une configuration réseau et de sécurité de base pour la connexion entre le cluster et la base de données.

#### Note

En commençant par version 3.10.0, AWS ParallelCluster prend en charge le Slurmdbd externe avec le paramètre de configuration du cluster. `SlurmSettings / ExternalSlurmdbd`

#### Note

Le modèle de création rapide sert d'exemple. Ce modèle ne couvre pas tous les cas d'utilisation possibles. Il est de votre responsabilité de créer un Slurmdbd externe avec la configuration et la capacité adaptées à vos charges de travail de production.

Prérequis :

- AWS ParallelCluster [est installé](#).
- Le AWS CLI [est installé et configuré](#).
- Vous possédez une [paire de clés Amazon Elastic Compute Cloud](#).
- Vous avez un AWS Identity and Access Management rôle doté des [autorisations](#) requises pour exécuter la `pcluster` CLI.
- Vous disposez d'une base de données Slurm comptable. Pour suivre un didacticiel sur la création d'une base de données de Slurm comptabilité, suivez les étapes 1 et 2 de la section [Création de la pile de base de données de comptabilité Slurm](#).

## Étape 1 : Création de la pile Slurmdbd

Dans ce didacticiel, utilisez un [modèle de CloudFormation création rapide \(us-east-1\) pour créer](#) une pile Slurmdbd. Le modèle nécessite les entrées suivantes :

### Réseaux

- VpcId : ID VPC permettant de lancer l'instance Slurmdbd.
- SubnetId: ID de sous-réseau pour lancer l'instance Slurmdbd.
- PrivatePrefix: le préfixe CIDR du VPC.
- PrivateIp: adresse IP privée secondaire à attribuer à l'instance de Slurmdbd.

### Connexion de la base de données

- dbmsClientSg : groupe de sécurité à associer à l'instance de Slurmdbd. Ce groupe de sécurité doit autoriser les connexions entre le serveur de base de données et l'instance de Slurmdbd.
- DBMS DatabaseName : nom de la base de données.
- DBMSUserName : nom d'utilisateur de la base de données.
- DBMS PasswordSecretArn : secret contenant le mot de passe de la base de données.
- DBMSUri : URI du serveur de base de données.

### Paramètres des instances

- InstanceType: type d'instance à utiliser pour l'instance de slurmdbd.
- KeyName: une paire de clés Amazon EC2 à utiliser pour l'instance slurmdbd.

### Paramètres de Slurmdbd

- AMIID : AMI de l'instance de Slurmdbd. L'AMI doit être une ParallelCluster AMI. La version de l' ParallelCluster AMI détermine la version de Slurmdbd.
- MungeKeySecretArn: Le secret contenant la clé munge à utiliser pour authentifier les communications entre Slurmdbd et les clusters.
- SlurmdbdPort: numéro de port utilisé par le slurmdbd.
- EnableSlurmdbdSystemService: active slurmdbd en tant que service système et le fait fonctionner au lancement d'une instance.

**⚠ Warning**

Si la base de données a été créée par une autre version de SlurmDB, ne l'utilisez pas Slurmdbd en tant que service système.

Si la base de données contient un grand nombre d'entrées, la mise à jour de la base de données Slurm Database Daemon (SlurmDBD) peut prendre des dizaines de minutes et ne pas répondre pendant cet intervalle de temps.

Avant de procéder à SlurmDB la mise à niveau, effectuez une sauvegarde de la base de données. Pour en savoir plus, consultez la [documentation Slurm](#).

## Étape 2 : Création d'un cluster avec l'Slurmdbdoption externe activée

Le AWS CloudFormation modèle fourni génère une AWS CloudFormation pile avec certaines sorties définies.

À partir du AWS Management Console, consultez l'onglet Sorties de la AWS CloudFormation pile pour passer en revue les entités créées. Pour permettre la Slurm comptabilisation, certaines de ces sorties doivent être utilisées dans le fichier de AWS ParallelCluster configuration :

- SlurmdbdPrivatelp: Utilisé pour le paramètre de configuration du [cluster SlurmSettingsExternalSlurmdbd//Host](#).
- SlurmdbdPort: Utilisé pour la valeur du paramètre de configuration du cluster [SlurmSettingsExternalSlurmdbd//Port](#).
- AccountingClientSecurityGroup: il s'agit du groupe de sécurité attaché au nœud principal du cluster défini dans le paramètre de configuration [HeadNode/Networking/AdditionalSecurityGroups](#).

De plus, depuis l'onglet Paramètres de la vue de la AWS CloudFormation pile :

- MungeKeySecretArn: Utilisé pour la valeur du paramètre de configuration [SlurmSettings/MungeKeySecretArn](#)cluster.

Mettez à jour les paramètres de la base de données du fichier de configuration de votre cluster avec les valeurs de sortie. Utilisez le pcluster AWS CLI pour créer le cluster.

```
$ pcluster create-cluster -n cluster-3.x -c path/to/cluster-config.yaml
```

Une fois le cluster créé, vous pouvez commencer à utiliser des commandes de Slurm comptabilité telles que `sacctmgr` ou `sacct`.

#### Warning

Le trafic entre `ParallelCluster` et l'extérieur n'est pas crypté. Il est recommandé d'exécuter le cluster et l'interface externe SlurmDB sur un réseau sécurisé.

## Revenir à une version précédente d'un document AWS Systems Manager

Découvrez comment revenir à une version précédente d'un document AWS Systems Manager. Pour plus d'informations sur les documents SSM, consultez les documents de [AWS Systems Manager](#) dans le Guide de l'utilisateur de AWS Systems Manager.

Lorsque vous utilisez l'interface de ligne de commande (CLI) ou l'API, vous ne payez que pour les AWS ressources créées lorsque vous créez ou mettez à jour AWS ParallelCluster des images et des clusters. Pour plus d'informations, veuillez consulter [AWS services utilisés par AWS ParallelCluster](#).

L'interface utilisateur AWS ParallelCluster repose sur une architecture sans serveur et vous pouvez l'utiliser dans la catégorie AWS Free Tier dans la plupart des cas. Pour plus d'informations, veuillez consulter [AWS ParallelCluster Coûts d'interface utilisateur](#).

Prérequis :

- Et Compte AWS avec des autorisations pour gérer les documents SSM.
- Le AWS CLI [est installé et configuré](#).

## Revenir à une version précédente du document SSM

1. Dans votre terminal, exécutez la commande suivante pour obtenir la liste des documents SSM existants que vous possédez.

```
$ aws ssm list-documents --document-filter "key=Owner,value=Self"
```

2. Rétablir la version précédente d'un document SSM. Dans cet exemple, nous revenons à une version précédente du `SessionManagerRunShell` document. Vous pouvez utiliser le `SessionManagerRunShell` document SSM pour personnaliser chaque session de shell SSM que vous lancez.
  - a. Trouvez le `DocumentVersion` paramètre pour `SessionManagerRunShell` en exécutant la commande suivante :

```
$ aws ssm describe-document --name "SSM-SessionManagerRunShell"
{
 "Document": {
 "Hash": "...",
 "HashType": "Sha256",
 "Name": "SSM-SessionManagerRunShell",
 "Owner": "123456789012",
 "CreateDate": "2023-02-20T19:04:32.390000+00:00",
 "Status": "Active",
 "DocumentVersion": "1",
 "Parameters": [
 {
 "Name": "linuxcmd",
 "Type": "String",
 "Description": "The command to run on connection...",
 "DefaultValue": "if [-d '/opt/parallelcluster']; then
source /opt/parallelcluster/cfnconfig; sudo su - $cfn_cluster_user; fi; /bin/
bash"
 }
],
 "PlatformTypes": [
 "Windows",
 "Linux",
 "MacOS"
],
 "DocumentType": "Session",
 "SchemaVersion": "1.0",
 "LatestVersion": "2",
 "DefaultVersion": "1",
 "DocumentFormat": "JSON",
 "Tags": []
 }
}
```

La dernière version est 2.

- b. Revenez à la version précédente en exécutant la commande suivante :

```
$ aws ssm delete-document --name "SSM-SessionManagerRunShell" --document-version 2
```

3. Vérifiez que la version du document a été rétablie en exécutant à nouveau la describe-document commande :

```
$ aws ssm describe-document --name "SSM-SessionManagerRunShell"
{
 "Document": {
 "Hash": "...",
 "HashType": "Sha256",
 "Name": "SSM-SessionManagerRunShell",
 "Owner": "123456789012",
 "CreateDate": "2023-02-20T19:04:32.390000+00:00",
 "Status": "Active",
 "DocumentVersion": "1",
 "Parameters": [
 {
 "Name": "linuxcmd",
 "Type": "String",
 "Description": "The command to run on connection...",
 "DefaultValue": "if [-d '/opt/parallelcluster']; then source /opt/parallelcluster/cfnconfig; sudo su - $cfn_cluster_user; fi; /bin/bash"
 }
],
 "PlatformTypes": [
 "Windows",
 "Linux",
 "MacOS"
],
 "DocumentType": "Session",
 "SchemaVersion": "1.0",
 "LatestVersion": "1",
 "DefaultVersion": "1",
 "DocumentFormat": "JSON",
 "Tags": []
 }
}
```

La dernière version est 1.

## Création d'un cluster avec AWS CloudFormation

Découvrez comment créer un cluster avec une ressource AWS ParallelCluster CloudFormation personnalisée. Pour plus d'informations, consultez [AWS CloudFormation ressource personnalisée](#).

Lors de l'utilisation AWS ParallelCluster, vous ne payez que pour les AWS ressources créées lorsque vous créez ou mettez à jour AWS ParallelCluster des images et des clusters. Pour plus d'informations, consultez [AWS services utilisés par AWS ParallelCluster](#).

Prérequis :

- Le AWS CLI [est installé et configuré](#).
- Une [paire de clés Amazon EC2](#).
- Rôle IAM doté des [autorisations](#) requises pour exécuter la [pcluster](#) CLI.

## Création de clusters à l'aide d'une pile CloudFormation à création rapide

Dans ce didacticiel, vous allez utiliser une pile de création rapide pour déployer un CloudFormation modèle qui crée un cluster et les ressources suivantes : AWS

- Une CloudFormation pile racine créée à l'aide d'une pile de CloudFormation création rapide.
- CloudFormation Des piles imbriquées qui incluent des politiques par défaut, une configuration VPC par défaut et un fournisseur de ressources personnalisé.
- Exemple de pile de AWS ParallelCluster clusters et de cluster auxquels vous pouvez vous connecter et exécuter des tâches.

Créez un cluster avec AWS CloudFormation

1. Connectez-vous au AWS Management Console.
2. Ouvrez le [lien de CloudFormation création rapide](#) pour créer les ressources suivantes dans la CloudFormation console :
  - Une CloudFormation pile imbriquée avec un VPC doté d'un sous-réseau public et privé pour exécuter le nœud principal du cluster et les nœuds de calcul, respectivement.

- Une CloudFormation pile imbriquée avec une ressource AWS ParallelCluster personnalisée pour gérer le cluster.
- Une CloudFormation pile imbriquée avec les politiques par défaut pour gérer le cluster.
- Une CloudFormation pile racine pour les piles imbriquées.
- Un AWS ParallelCluster cluster avec le Slurm planificateur et un nombre défini de nœuds de calcul.



CloudFormation > Stacks > Create stack

## Quick create stack

### Template

Template URL  
https://pcluster-cfn-us-east-2.s3.amazonaws.com/parallelcluster/3.5.0/templates/custom\_resource/cluster-1-click.yaml

Stack description  
AWS ParallelCluster CloudFormation Cluster

### Stack name

Stack name  
cluster-0

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

### Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

AvailabilityZone  
Availability zone where instances will be launched  
us-east-2a

KeyName  
KeyPair to login to the head node  
Select AWS::EC2::KeyPair::KeyName

### Capabilities

**The following resource(s) require capabilities: [AWS::CloudFormation::Stack]**

This template contains Identity and Access Management (IAM) resources. Check that you want to create each of these resources and that they have the minimum required permissions. In addition, they have custom names. Check that the custom names are unique within your AWS account. [Learn more](#)

For this template, AWS CloudFormation might require an unrecognized capability: {0}. Check the capabilities of these resources. [Learn more](#)

I acknowledge that AWS CloudFormation might create IAM resources with custom names.

I acknowledge that AWS CloudFormation might require the following capability: CAPABILITY\_AUTO\_EXPAND

Cancel Create change set Create stack

3. Dans la section Paramètres de création rapide de la pile, entrez les valeurs des paramètres suivants :
  - a. Pour KeyName, entrez le nom de votre paire de clés Amazon EC2.

- b. Pour AvailabilityZone, choisissez un AZ pour les nœuds de votre cluster, par exemple, `us-east-1a`.
4. Cochez les cases correspondant à chacune des fonctionnalités d'accès au bas de la page.
5. Sélectionnez Créer la pile.
6. Attendez que la CloudFormation pile atteigne son CREATE\_COMPLETE état.

## Création de clusters avec l'interface de ligne de AWS CloudFormation commande (CLI)

Dans ce didacticiel, vous allez utiliser l'interface de ligne de AWS commande (CLI) CloudFormation pour déployer un CloudFormation modèle qui crée un cluster.

Créez les AWS ressources suivantes :

- Une CloudFormation pile racine créée à l'aide d'une pile de CloudFormation création rapide.
- CloudFormation Des piles imbriquées qui incluent des politiques par défaut, une configuration VPC par défaut et un fournisseur de ressources personnalisé.
- Exemple de pile de AWS ParallelCluster clusters et de cluster auxquels vous pouvez vous connecter et exécuter des tâches.

Remplacez *les entrées surlignées en rouge*, telles que *keypair*, par vos propres valeurs.

Créez un cluster avec AWS CloudFormation

1. Créez un CloudFormation modèle nommé `cluster_template.yaml` avec le contenu suivant :

```
AWSTemplateFormatVersion: '2010-09-09'
Description: > AWS ParallelCluster CloudFormation Template

Parameters:
 KeyName:
 Description: KeyPair to login to the head node
 Type: AWS::EC2::KeyPair::KeyName

 AvailabilityZone:
 Description: Availability zone where instances will be launched
 Type: AWS::EC2::AvailabilityZone::Name
 Default: us-east-2a
```

```
Mappings:
 ParallelCluster:
 Constants:
 Version: 3.7.0

Resources:
 PclusterClusterProvider:
 Type: AWS::CloudFormation::Stack
 Properties:
 TemplateURL: !Sub
 - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.
 ${AWS::URLSuffix}/parallelcluster/${Version}/templates/custom_resource/cluster.yaml
 - { Version: !FindInMap [ParallelCluster, Constants, Version] }

 PclusterVpc:
 Type: AWS::CloudFormation::Stack
 Properties:
 Parameters:
 PublicCIDR: 10.0.0.0/24
 PrivateCIDR: 10.0.16.0/20
 AvailabilityZone: !Ref AvailabilityZone
 TemplateURL: !Sub
 - https://${AWS::Region}-aws-parallelcluster.s3.${AWS::Region}.
 ${AWS::URLSuffix}/parallelcluster/${Version}/templates/networking/public-private-
 ${Version}.cfm.json
 - { Version: !FindInMap [ParallelCluster, Constants, Version] }

 PclusterCluster:
 Type: Custom::PclusterCluster
 Properties:
 ServiceToken: !GetAtt [PclusterClusterProvider , Outputs.ServiceToken]
 ClusterName: !Sub 'c-${AWS::StackName}'
 ClusterConfiguration:
 Image:
 Os: alinux2
 HeadNode:
 InstanceType: t2.medium
 Networking:
 SubnetId: !GetAtt [PclusterVpc , Outputs.PublicSubnetId]
 Ssh:
 KeyName: !Ref KeyName
 Scheduling:
 Scheduler: slurm
```

```

SlurmQueues:
- Name: queue0
 ComputeResources:
 - Name: queue0-cr0
 InstanceType: t2.micro
 Networking:
 SubnetIds:
 - !GetAtt [PclusterVpc , Outputs.PrivateSubnetId]

Outputs:
 HeadNodeIp:
 Description: The Public IP address of the HeadNode
 Value: !GetAtt [PclusterCluster, headNode.publicIpAddress]

```

2. Exécutez la commande AWS CLI suivante pour déployer la CloudFormation pile à des fins de création et de gestion de clusters.

```

$ aws cloudformation deploy --template-file ./cluster_template.yaml \
 --stack-name mycluster \
 --parameter-overrides KeyName=keypair \
 AvailabilityZone=us-east-2b \
 --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND

```

## Afficher la sortie CloudFormation du cluster

Consultez la sortie du CloudFormation cluster pour obtenir des informations utiles sur le cluster. La `ValidationMessages` propriété ajoutée permet d'accéder aux messages de validation issus des opérations de création et de mise à jour du cluster.

1. Accédez à la [CloudFormation console](#) et sélectionnez la pile qui inclut votre ressource AWS ParallelCluster personnalisée.
2. Choisissez Stack details, puis sélectionnez l'onglet Outputs.

| Key                | Value                                                                                                                                                                                                                            | Description                                            |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| HeadNodeIp         | 1.2.3.4                                                                                                                                                                                                                          | The Public IP address of the HeadNode                  |
| ValidationMessages | <pre>[{"level": "WARNING", "type": "KeyPairValidator", "message": "If you do not specify a key pair, you can't connect to the instance unless you choose an AMI that is configured to allow users another way to log in"}]</pre> | Any warnings from cluster create or update operations. |

Les messages de validation peuvent être tronqués. Pour plus d'informations sur la façon de récupérer les journaux, consultez [AWS ParallelCluster résolution des problèmes](#).

## Accédez à votre cluster

Accédez au cluster.

**ssh** dans le nœud principal du cluster

1. Une fois le déploiement de la CloudFormation pile terminé, obtenez l'adresse IP du nœud principal à l'aide de la commande suivante :

```
$ HEAD_NODE_IP=$(aws cloudformation describe-stacks --stack-name=mycluster --query "Stacks|[0].Outputs[?OutputKey=='HeadNodeIp']|[0].OutputValue" --output=text)
```

Vous pouvez également récupérer l'adresse IP du nœud principal à partir du HeadNodeIp paramètre de l'onglet Cluster Stack Outputs de la CloudFormation console.

Vous pouvez trouver l'adresse IP du nœud principal ici car elle a été ajoutée dans la Outputs section du CloudFormation modèle de cluster, spécifiquement pour cet exemple de cluster.

2. Connectez-vous au nœud principal du cluster en exécutant la commande suivante :

```
$ ssh -i keyname.pem ec2-user@$HEAD_NODE_IP
```

## Nettoyage

Supprimez le cluster.

1. Exécutez la commande AWS CLI suivante pour supprimer la CloudFormation pile et le cluster.

```
$ aws cloudformation delete-stack --stack-name=mycluster
```

2. Vérifiez l'état de suppression de la pile en exécutant la commande suivante.

```
$ aws cloudformation describe-stacks --stack-name=mycluster
```

# Déploiement ParallelCluster d'une API avec Terraform

Dans ce didacticiel, vous allez définir un projet Terraform simple pour déployer une ParallelCluster API.

## Prérequis

- Terraform v1.5.7+ est installé.
- Rôle IAM autorisé à déployer l' ParallelCluster API. veuillez consulter [the section called "Autorisations nécessaires"](#).

## Définir un projet Terraform

1. Créez un répertoire appelé `pcluster-api`.

Tous les fichiers que vous créez se trouveront dans ce répertoire.

2. Créez le fichier `provider.tf` pour configurer le AWS fournisseur.

```
provider "aws" {
 region = var.region
 profile = var.profile
}
```

3. Créez le fichier `main.tf` pour définir les ressources à l'aide du ParallelCluster module.

```
module "parallelcluster_pcluster_api" {
 source = "aws-tf/parallelcluster/aws//modules/pcluster_api"
 version = "1.0.0"

 region = var.region
 api_stack_name = var.api_stack_name
 api_version = var.api_version
 deploy_pcluster_api = true
 parameters = {
 EnableIamAdminAccess = "true"
 }
}
```

4. Créez le fichier `variables.tf` pour définir les variables qui peuvent être injectées pour ce projet.

```
variable "region" {
 description = "The region the ParallelCluster API is deployed in."
 type = string
 default = "us-east-1"
}

variable "profile" {
 type = string
 description = "The AWS profile used to deploy the clusters."
 default = null
}

variable "api_stack_name" {
 type = string
 description = "The name of the CloudFormation stack used to deploy the
ParallelCluster API."
 default = "ParallelCluster"
}

variable "api_version" {
 type = string
 description = "The version of the ParallelCluster API."
}
```

5. Créez le fichier `terraform.tfvars` pour définir des valeurs arbitraires pour les variables.

Le fichier ci-dessous déploie une ParallelCluster API 3.10.0 en `us-east-1` utilisant le nom de la pile. `MyParallelClusterAPI-310` Vous pourrez référencer ce déploiement d' ParallelCluster API à l'aide de son nom de pile.

```
region = "us-east-1"
api_stack_name = "MyParallelClusterAPI-310"
api_version = "3.10.0"
```

6. Créez le fichier `outputs.tf` pour définir les sorties renvoyées par ce projet.

```
output "pcluster_api_stack_outputs" {
 value = module.parallelcluster_pcluster_api.stack_outputs
}
```

Le répertoire du projet est le suivant :

```
my-pcluster-api
main.tf - Terraform entrypoint to define the resources using the
ParallelCluster module.
outputs.tf - Defines the outputs returned by Terraform.
providers.tf - Configures the AWS provider.
terraform.tfvars - Set the arbitrary values for the variables, i.e. region,
PCAPI version, PCAPI stack name
variables.tf - Defines the variables, e.g. region, PCAPI version, PCAPI stack
name.
```

## Déploiement de l'API

Pour déployer l'API, exécutez les commandes Terraform standard dans l'ordre.

1. Construisez le projet :

```
terraform init
```

2. Définissez le plan de déploiement :

```
terraform plan -out tfplan
```

3. Déployez le plan :

```
terraform apply tfplan
```

## Autorisations nécessaires

Vous avez besoin des autorisations suivantes pour déployer l' ParallelCluster API avec Terraform :

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
 "cloudformation:DescribeStacks",
 "cloudformation:GetTemplate"
],
 "Resource": "arn:PARTITION:cloudformation:REGION:ACCOUNT:stack/*",
```



```

 "Effect": "Allow",
 "Sid": "CloudFormationRead"
 },
 {
 "Action": [
 "cloudformation:CreateStack",
 "cloudformation>DeleteStack",
 "cloudformation:CreateChangeSet"
],
 "Resource": "arn:PARTITION:cloudformation:REGION:ACCOUNT:stack/
MyParallelClusterAPI*",
 "Effect": "Allow",
 "Sid": "CloudFormationWrite"
 },
 {
 "Action": [
 "cloudformation:CreateChangeSet"
],
 "Resource": [
 "arn:PARTITION:cloudformation:REGION:aws:transform/Include",
 "arn:PARTITION:cloudformation:REGION:aws:transform/
Serverless-2016-10-31"
],
 "Effect": "Allow",
 "Sid": "CloudFormationTransformWrite"
 },
 {
 "Action": [
 "s3:GetObject"
],
 "Resource": [
 "arn:PARTITION:s3::*-aws-parallelcluster/parallelcluster/*/api/
ParallelCluster.openapi.yaml",
 "arn:PARTITION:s3::*-aws-parallelcluster/parallelcluster/*/layers/aws-
parallelcluster/lambda-layer.zip"
],
 "Effect": "Allow",
 "Sid": "S3ParallelClusterArtifacts"
 },
 {
 "Action": [
 "iam:CreateRole",
 "iam>DeleteRole",
 "iam:GetRole",

```

```

 "iam:CreatePolicy",
 "iam>DeletePolicy",
 "iam:GetPolicy",
 "iam:GetRolePolicy",
 "iam:AttachRolePolicy",
 "iam:DetachRolePolicy",
 "iam:PutRolePolicy",
 "iam>DeleteRolePolicy",
 "iam:ListPolicyVersions"
],
 "Resource": [
 "arn:PARTITION:iam::ACCOUNT:role/*",
 "arn:PARTITION:iam::ACCOUNT:policy/*"
],
 "Effect": "Allow",
 "Sid": "IAM"
},
{
 "Action": [
 "iam:PassRole"
],
 "Resource": [
 "arn:PARTITION:iam::ACCOUNT:role/ParallelClusterLambdaRole-*",
 "arn:PARTITION:iam::ACCOUNT:role/APIGatewayExecutionRole-*"
],
 "Effect": "Allow",
 "Sid": "IAMPassRole"
},
{
 "Action": [
 "lambda:CreateFunction",
 "lambda>DeleteFunction",
 "lambda:GetFunction",
 "lambda:PublishLayerVersion",
 "lambda>DeleteLayerVersion",
 "lambda:GetLayerVersion",
 "lambda:TagResource",
 "lambda:UntagResource"
],
 "Resource": [
 "arn:PARTITION:lambda:REGION:ACCOUNT:layer:PCLayer-*",
 "arn:PARTITION:lambda:REGION:ACCOUNT:function:*-
ParallelClusterFunction-*"
],

```

```

 "Effect": "Allow",
 "Sid": "Lambda"
 },
 {
 "Action": [
 "logs:CreateLogGroup",
 "logs>DeleteLogGroup",
 "logs:DescribeLogGroups",
 "logs:PutRetentionPolicy",
 "logs:TagLogGroup",
 "logs:UntagLogGroup"
],
 "Resource": [
 "arn:PARTITION:logs:REGION:ACCOUNT:log-group:/aws/lambda/*-
ParallelClusterFunction-*"
],
 "Effect": "Allow",
 "Sid": "Logs"
 },
 {
 "Action": [
 "apigateway:DELETE",
 "apigateway:GET",
 "apigateway:PATCH",
 "apigateway:POST",
 "apigateway:PUT",
 "apigateway:UpdateRestApiPolicy"
],
 "Resource": [
 "arn:PARTITION:apigateway:REGION::/restapis",
 "arn:PARTITION:apigateway:REGION::/restapis/*",
 "arn:PARTITION:apigateway:REGION::/tags/*"
],
 "Effect": "Allow",
 "Sid": "APIGateway"
 }
]
}

```

# Création d'un cluster avec Terraform

Lors de l'utilisation AWS ParallelCluster, vous ne payez que pour les AWS ressources créées lorsque vous créez ou mettez à jour AWS ParallelCluster des images et des clusters. Pour plus d'informations, consultez [the section called "AWS services utilisés par AWS ParallelCluster"](#).

## Prérequis

- Terraform v1.5.7+ est installé.
- [the section called "AWS ParallelCluster API"](#) La version 3.8.0+ est déployée sur votre compte. veuillez consulter [the section called "Déploiement ParallelCluster d'une API avec Terraform"](#).
- Rôle IAM autorisé à invoquer l' ParallelCluster API. Voir [Autorisations requises]

## Définir un projet Terraform

Dans ce didacticiel, vous allez définir un projet Terraform simple pour déployer un cluster.

1. Créez un répertoire appelé `clusters`.

Tous les fichiers que vous créez se trouveront dans ce répertoire.

2. Créez le fichier `terraform.tf` pour importer le ParallelCluster fournisseur.

```
terraform {
 required_version = ">= 1.5.7"
 required_providers {
 aws-parallelcluster = {
 source = "aws-tf/aws-parallelcluster"
 version = "1.0.0"
 }
 }
}
```

3. Créez le fichier `providers.tf` pour configurer les AWS fournisseurs ParallelCluster et.

```
provider "aws" {
 region = var.region
 profile = var.profile
}

provider "aws-parallelcluster" {
```

```

region = var.region
profile = var.profile
api_stack_name = var.api_stack_name
use_user_role = true
}

```

4. Créez le fichier `main.tf` pour définir les ressources à l'aide du `ParallelCluster` module.

```

module "pcluster" {
 source = "aws-tf/parallelcluster/aws"
 version = "1.0.0"

 region = var.region
 api_stack_name = var.api_stack_name
 api_version = var.api_version
 deploy_pcluster_api = false

 template_vars = local.config_vars
 cluster_configs = local.cluster_configs
 config_path = "config/clusters.yaml"
}

```

5. Créez le fichier `clusters.tf` pour définir plusieurs clusters en tant que variables locales Terraform.

#### Note

Vous pouvez définir plusieurs clusters au sein de l'`cluster_config` élément. Pour chaque cluster, vous pouvez définir explicitement les propriétés du cluster dans les variables locales (voir `DemoCluster01`) ou référencer un fichier externe (voir `DemoCluster02`).

Pour consulter les propriétés du cluster que vous pouvez définir dans l'élément de configuration, reportez-vous à [the section called "Fichier de configuration du cluster"](#).

Pour passer en revue les options que vous pouvez définir pour la création de clusters, consultez [the section called "pcluster create-cluster"](#).

```

locals {
 cluster_configs = {

```

```
DemoCluster01 : {
 region : local.config_vars.region
 rollbackOnFailure : false
 validationFailureLevel : "WARNING"
 suppressValidators : [
 "type:KeyPairValidator"
]
 configuration : {
 Region : local.config_vars.region
 Image : {
 Os : "alinux2"
 }
 HeadNode : {
 InstanceType : "t3.small"
 Networking : {
 SubnetId : local.config_vars.subnet
 }
 Iam : {
 AdditionalIamPolicies : [
 { Policy : "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore" }
]
 }
 }
 }
 Scheduling : {
 Scheduler : "slurm"
 SlurmQueues : [{
 Name : "queue1"
 CapacityType : "ONDEMAND"
 Networking : {
 SubnetIds : [local.config_vars.subnet]
 }
 Iam : {
 AdditionalIamPolicies : [
 { Policy : "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore" }
]
 }
 ComputeResources : [{
 Name : "compute"
 InstanceType : "t3.small"
 MinCount : "1"
 MaxCount : "4"
 }]
 }]
 }
 SlurmSettings : {
```

```

 QueueUpdateStrategy : "TERMINATE"
 }
 }
 }
}
DemoCluster02 : {
 configuration : "config/cluster_config.yaml"
}
}
}

```

6. Créez le fichier `config/clusters.yaml` pour définir plusieurs clusters en tant que configuration YAML.

```

DemoCluster03:
 region: ${region}
 rollbackOnFailure: true
 validationFailureLevel: WARNING
 suppressValidators:
 - type:KeyPairValidator
 configuration: config/cluster_config.yaml
DemoCluster04:
 region: ${region}
 rollbackOnFailure: false
 configuration: config/cluster_config.yaml

```

7. Créez le fichier `config/cluster_config.yaml`, qui est un fichier de ParallelCluster configuration standard dans lequel les variables Terraform peuvent être injectées.

Pour consulter les propriétés du cluster que vous pouvez définir dans l'élément de configuration, reportez-vous à [the section called "Fichier de configuration du cluster"](#).

```

Region: ${region}
Image:
 Os: alinux2
HeadNode:
 InstanceType: t3.small
 Networking:
 SubnetId: ${subnet}
 Iam:
 AdditionalIamPolicies:
 - Policy: arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
Scheduling:

```

```
Scheduler: slurm
SlurmQueues:
 - Name: queue1
 CapacityType: ONDEMAND
 Networking:
 SubnetIds:
 - ${subnet}
 Iam:
 AdditionalIamPolicies:
 - Policy: arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
 ComputeResources:
 - Name: compute
 InstanceType: t3.small
 MinCount: 1
 MaxCount: 5
SlurmSettings:
 QueueUpdateStrategy: TERMINATE
```

8. Créez le fichier `clusters_vars.tf` pour définir les variables qui peuvent être injectées dans les configurations de cluster.

Ce fichier vous permet de définir des valeurs dynamiques qui peuvent être utilisées dans les configurations de cluster, telles que la région et le sous-réseau.

Cet exemple extrait des valeurs directement à partir des variables du projet, mais vous devrez peut-être utiliser une logique personnalisée pour les déterminer.

```
locals {
 config_vars = {
 subnet = var.subnet_id
 region = var.cluster_region
 }
}
```

9. Créez le fichier `variables.tf` pour définir les variables qui peuvent être injectées pour ce projet.

```
variable "region" {
 description = "The region the ParallelCluster API is deployed in."
 type = string
 default = "us-east-1"
}
```



```
variable "cluster_region" {
 description = "The region the clusters will be deployed in."
 type = string
 default = "us-east-1"
}

variable "profile" {
 type = string
 description = "The AWS profile used to deploy the clusters."
 default = null
}

variable "subnet_id" {
 type = string
 description = "The id of the subnet to be used for the ParallelCluster
instances."
}

variable "api_stack_name" {
 type = string
 description = "The name of the CloudFormation stack used to deploy the
ParallelCluster API."
 default = "ParallelCluster"
}

variable "api_version" {
 type = string
 description = "The version of the ParallelCluster API."
}
```

10. Créez le fichier `terraform.tfvars` pour définir des valeurs arbitraires pour les variables.

Le fichier ci-dessous déploie les clusters `eu-west-1` au sein du sous-réseau `subnet-123456789`, en utilisant l' `ParallelCluster API 3.10.0` existante, qui est déjà déployée avec le nom de la pile. `us-east-1 MyParallelClusterAPI-310`

```
region = "us-east-1"
api_stack_name = "MyParallelClusterAPI-310"
api_version = "3.10.0"

cluster_region = "eu-west-1"
subnet_id = "subnet-123456789"
```

## 11. Créez le fichier `outputs.tf` pour définir les sorties renvoyées par ce projet.

```
output "clusters" {
 value = module.pcluster.clusters
}
```

Le répertoire du projet est le suivant :

```
my-clusters
config
cluster_config.yaml - Cluster configuration, where terraform variables can
 be injected..
clusters.yaml - File listing all the clusters to deploy.
clusters.tf - Clusters defined as Terraform local variables.
clusters_vars.tf - Variables that can be injected into cluster configurations.
main.tf - Terraform entrypoint where the ParallelCluster module is configured.
outputs.tf - Defines the cluster as a Terraform output.
providers.tf - Configures the providers: ParallelCluster and AWS.
terraform.tf - Import the ParallelCluster provider.
terraform.tfvars - Defines values for variables, e.g. region, PCAPI stack name.
variables.tf - Defines the variables, e.g. region, PCAPI stack name.
```

## Déploiement du cluster

Pour déployer le cluster, exécutez les commandes Terraform standard dans l'ordre.

### Note

Cet exemple suppose que vous avez déjà déployé l' ParallelCluster API dans votre compte.

### 1. Construisez le projet :

```
terraform init
```

### 2. Définissez le plan de déploiement :

```
terraform plan -out tfplan
```

### 3. Déployez le plan :

```
terraform apply tfplan
```

## Déployez l' ParallelCluster API avec des clusters

Si vous n'avez pas déployé l' ParallelCluster API et que vous souhaitez la déployer avec les clusters, modifiez les fichiers suivants :

- `main.tf`

```
module "pcluster" {
 source = "aws-tf/aws/parallelcluster"
 version = "1.0.0"

 region = var.region
 api_stack_name = var.api_stack_name
 api_version = var.api_version
 deploy_pcluster_api = true

 template_vars = local.config_vars
 cluster_configs = local.cluster_configs
 config_path = "config/clusters.yaml"
}
```

- `providers.tf`

```
provider "aws-parallelcluster" {
 region = var.region
 profile = var.profile
 endpoint = module.pcluster.pcluster_api_stack_outputs.ParallelClusterApiInvokeUrl
 role_arn = module.pcluster.pcluster_api_stack_outputs.ParallelClusterApiUserRole
}
```

## Autorisations nécessaires

Vous avez besoin des autorisations suivantes pour déployer un cluster avec Terraform :

- assumer le rôle d' ParallelCluster API, qui est chargé d'interagir avec l' ParallelCluster API

- décrire la AWS CloudFormation pile de l' ParallelCluster API pour vérifier son existence et récupérer ses paramètres et ses sorties

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": "sts:AssumeRole",
 "Resource": "arn:PARTITION:iam::ACCOUNT:role/PCAPIUserRole-*",
 "Effect": "Allow",
 "Sid": "AssumePCAPIUserRole"
 },
 {
 "Action": [
 "cloudformation:DescribeStacks"
],
 "Resource": "arn:PARTITION:cloudformation:REGION:ACCOUNT:stack/*",
 "Effect": "Allow",
 "Sid": "CloudFormation"
 }
]
}
```

## Création d'une AMI personnalisée avec Terraform

Lors de l'utilisation AWS ParallelCluster, vous ne payez que pour les AWS ressources créées lorsque vous créez ou mettez à jour AWS ParallelCluster des images et des clusters. Pour plus d'informations, consultez [the section called "AWS services utilisés par AWS ParallelCluster"](#).

### Prérequis

- Terraform v1.5.7+ est installé.
- [the section called "AWS ParallelCluster API"](#) La version 3.8.0+ est déployée sur votre compte. veuillez consulter [the section called "Création d'un cluster avec Terraform"](#).
- Rôle IAM autorisé à invoquer l' ParallelCluster API. veuillez consulter [the section called "Autorisations nécessaires"](#).

## Définir un projet Terraform

Dans ce didacticiel, vous allez définir un projet Terraform simple pour déployer une ParallelCluster AMI personnalisée.

1. Créez un répertoire appelé `amis`.

Tous les fichiers que vous créez se trouveront dans ce répertoire.

2. Créez le fichier `terraform.tf` pour importer le ParallelCluster fournisseur.

```
terraform {
 required_version = ">= 1.5.7"
 required_providers {
 aws-parallelcluster = {
 source = "aws-tf/aws-parallelcluster"
 version = "1.0.0"
 }
 }
}
```

3. Créez le fichier `providers.tf` pour configurer les AWS fournisseurs ParallelCluster et.

```
provider "aws" {
 region = var.region
 profile = var.profile
}

provider "aws-parallelcluster" {
 region = var.region
 profile = var.profile
 api_stack_name = var.api_stack_name
 use_user_role = true
}
```

4. Créez le fichier `main.tf` pour définir les ressources à l'aide du ParallelCluster module.

Pour consulter les propriétés de l'image que vous pouvez définir au sein de l'image\_configuration élément, voir [the section called "Créez des fichiers de configuration d'image"](#).

Pour consulter les options que vous pouvez définir pour la création d'images, par exemple `rollback_on_failure`, `image_id` et voir [the section called "pcluster build-image"](#).

```
data "aws-parallelcluster_list_official_images" "parent_image" {
 region = var.region
 os = var.os
 architecture = var.architecture
}

resource "aws-parallelcluster_image" "demo01" {
 image_id = "demo01"
 image_configuration = yamlencode({
 "Build":{
 "InstanceType": "c5.2xlarge",
 "ParentImage": data.aws-
parallelcluster_list_official_images.parent_image.official_images[0].amiId,
 "UpdateOsPackages": {"Enabled": false}
 }
 })
 rollback_on_failure = false
}
```

5. Créez le fichier `variables.tf` pour définir les variables qui peuvent être injectées pour ce projet.

```
variable "region" {
 description = "The region the ParallelCluster API is deployed in."
 type = string
 default = "us-east-1"
}

variable "profile" {
 type = string
 description = "The AWS profile used to deploy the clusters."
 default = null
}

variable "api_stack_name" {
 type = string
 description = "The name of the CloudFormation stack used to deploy the
ParallelCluster API."
}
```

```
default = "ParallelCluster"
}

variable "api_version" {
 type = string
 description = "The version of the ParallelCluster API."
}

variable "os" {
 type = string
 description = "The OS of the ParallelCluster image."
}

variable "architecture" {
 type = string
 description = "The architecture of the ParallelCluster image."
}
```

6. Créez le fichier `terraform.tfvars` pour définir des valeurs arbitraires pour les variables.

Le fichier ci-dessous déploie l'AMI personnalisée `us-east-1` basée sur l'architecture Amazon Linux 2 pour `x86_64`, en utilisant l' `ParallelCluster API 3.10.0` existante qui est déjà déployée avec le nom de la pile. `us-east-1 MyParallelClusterAPI-310`

```
region = "us-east-1"
api_stack_name = "MyParallelClusterAPI-310"
api_version = "3.10.0"

os = "alinux2"
architecture = "x86_64"
```

7. Créez le fichier `outputs.tf` pour définir les sorties renvoyées par ce projet.

```
output "parent_image" {
 value = data.aws-
parallelcluster_list_official_images.parent_image.official_images[0]
}

output "custom_image" {
 value = aws-parallelcluster_image.demo01
}
```

Le répertoire du projet est le suivant :

```
my-amis
main.tf - Terraform entrypoint where the ParallelCluster module is configured.
outputs.tf - Defines the cluster as a Terraform output.
providers.tf - Configures the providers: ParallelCluster and AWS.
terraform.tf - Import the ParallelCluster provider.
terraform.tfvars - Defines values for variables, e.g. region, PCAPI stack name.
variables.tf - Defines the variables, e.g. region, PCAPI stack name.
```

## Déployer l'AMI

Pour déployer l'AMI, exécutez les commandes Terraform standard dans l'ordre.

1. Construisez le projet :

```
terraform init
```

2. Définissez le plan de déploiement :

```
terraform plan -out tfplan
```

3. Déployez le plan :

```
terraform apply tfplan
```

## Autorisations nécessaires

Vous avez besoin des autorisations suivantes pour déployer une AMI personnalisée avec Terraform :

- assumer le rôle d' ParallelCluster API, qui est chargé d'interagir avec l' ParallelCluster API
- décrire la AWS CloudFormation pile de l' ParallelCluster API, pour vérifier son existence et récupérer ses paramètres et ses sorties

```
{
 "Version": "2012-10-17",
 "Statement": [
```



```
{
 "Action": "sts:AssumeRole",
 "Resource": "arn:PARTITION:iam::ACCOUNT:role/PCAPIUserRole-*",
 "Effect": "Allow",
 "Sid": "AssumePCAPIUserRole"
},
{
 "Action": [
 "cloudformation:DescribeStacks"
],
 "Resource": "arn:PARTITION:cloudformation:REGION:ACCOUNT:stack/*",
 "Effect": "Allow",
 "Sid": "CloudFormation"
}
]
```

## AWS ParallelClusterIntégration de l'interface utilisateur à Identity Center

L'objectif de ce didacticiel est de montrer comment intégrer l'AWS ParallelClusterinterface utilisateur à IAM Identity Center pour créer une solution d'authentification unique qui unifie les utilisateurs dans Active Directory et qui peut être partagée avec des clusters. AWS ParallelCluster

Lors de l'utilisationAWS ParallelCluster, vous ne payez que pour les AWS ressources créées lorsque vous créez ou mettez à jour AWS ParallelCluster des images et des clusters. Pour en savoir plus, consultez [AWS services utilisés par AWS ParallelCluster](#).

Prérequis :

- Une AWS ParallelCluster interface utilisateur existante qui peut être installée en suivant les instructions [ici](#).
- Un Active Directory géré existant, de préférence un annuaire que vous utiliserez également pour [l'intégration AWS ParallelCluster](#).

## Activer IAM Identity Center

Si vous avez déjà un centre d'identité connecté à votre AWS Managed Microsoft AD (Active Directory), il peut être utilisé et vous pouvez passer à la section Ajouter votre application à IAM Identity Center.

Si aucun centre d'identité n'est déjà connecté à un AWS Managed Microsoft AD, suivez les étapes ci-dessous pour le configurer.

### Activation du centre d'identité

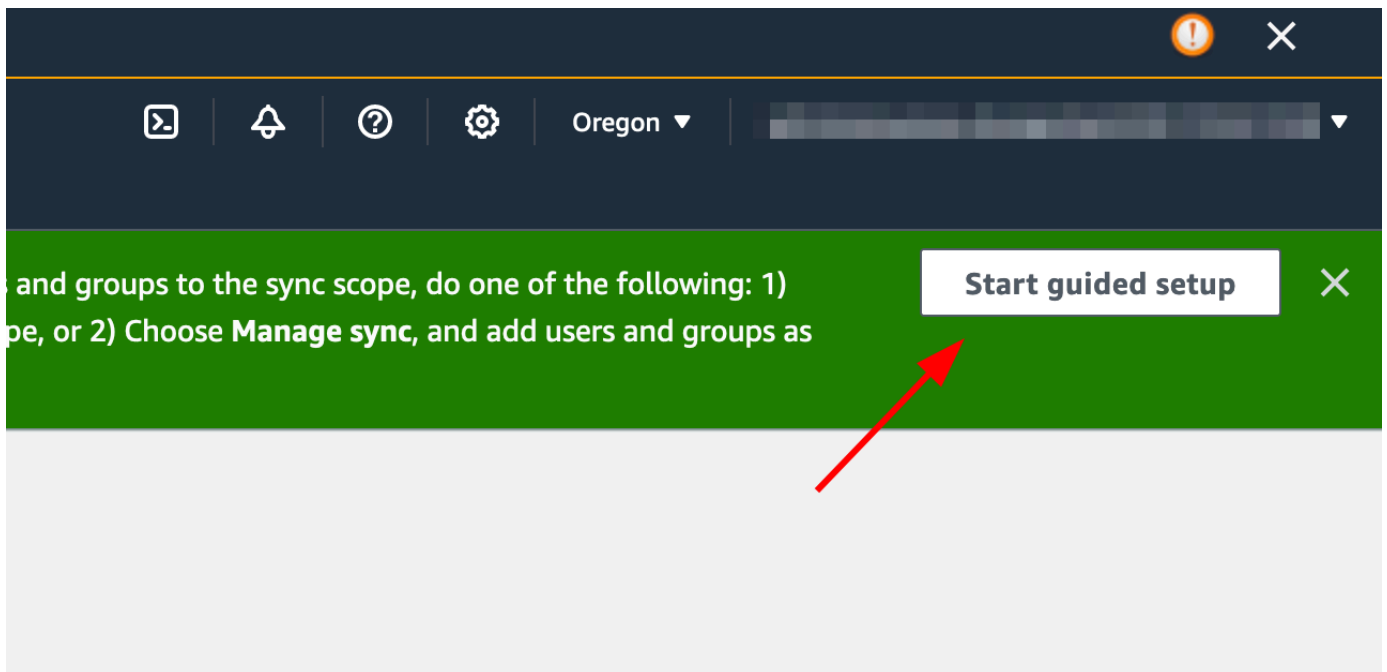
1. Dans la console, accédez à IAM Identity Center. (Assurez-vous que vous vous trouvez dans la région dans laquelle vous avez votre AWS Managed Microsoft AD.)
2. Cliquez sur le bouton Activer. Cela peut vous demander si vous souhaitez activer les organisations. C'est une obligation pour que vous puissiez choisir de l'activer. Remarque : Cela enverra à l'administrateur de votre compte un e-mail de confirmation vous demandant de suivre le lien pour confirmer.

### Connecter Identity Center à Managed AD

1. Sur la page suivante, après avoir activé le centre d'identité, vous devriez voir les étapes de configuration recommandées. À l'étape 1, sélectionnez Choisissez votre source d'identité.
2. Dans la section Source d'identité, cliquez sur le menu déroulant Actions (en haut à droite), puis sélectionnez Modifier la source d'identité.
3. Sélectionnez Active Directory.
4. Sous Répertoires existants, choisissez votre répertoire.
5. Cliquez sur Next (Suivant).
6. Passez en revue vos modifications, faites défiler l'écran vers le bas, tapez ACCEPT dans la zone de texte pour confirmer, puis cliquez sur Modifier la source d'identité.
7. Attendez que les modifications soient terminées, puis vous devriez voir une bannière verte en haut.

### Synchronisation des utilisateurs et des groupes avec Identity Center

1. Dans le bandeau vert, cliquez sur Démarrer la configuration guidée (bouton en haut à droite)



2. Dans la section Configurer les mappages d'attributs, cliquez sur Suivant
3. Dans la section Configurer l'étendue de synchronisation, saisissez le nom des utilisateurs que vous souhaitez synchroniser avec le centre d'identité, puis cliquez sur Ajouter
4. Une fois que vous avez terminé d'ajouter des utilisateurs et des groupes, cliquez sur Suivant

**Users** | **Groups**

---

**User**

corp.pcluster.com ▼    🔍 Enter exact match user to add to sync scope    **Add**

---

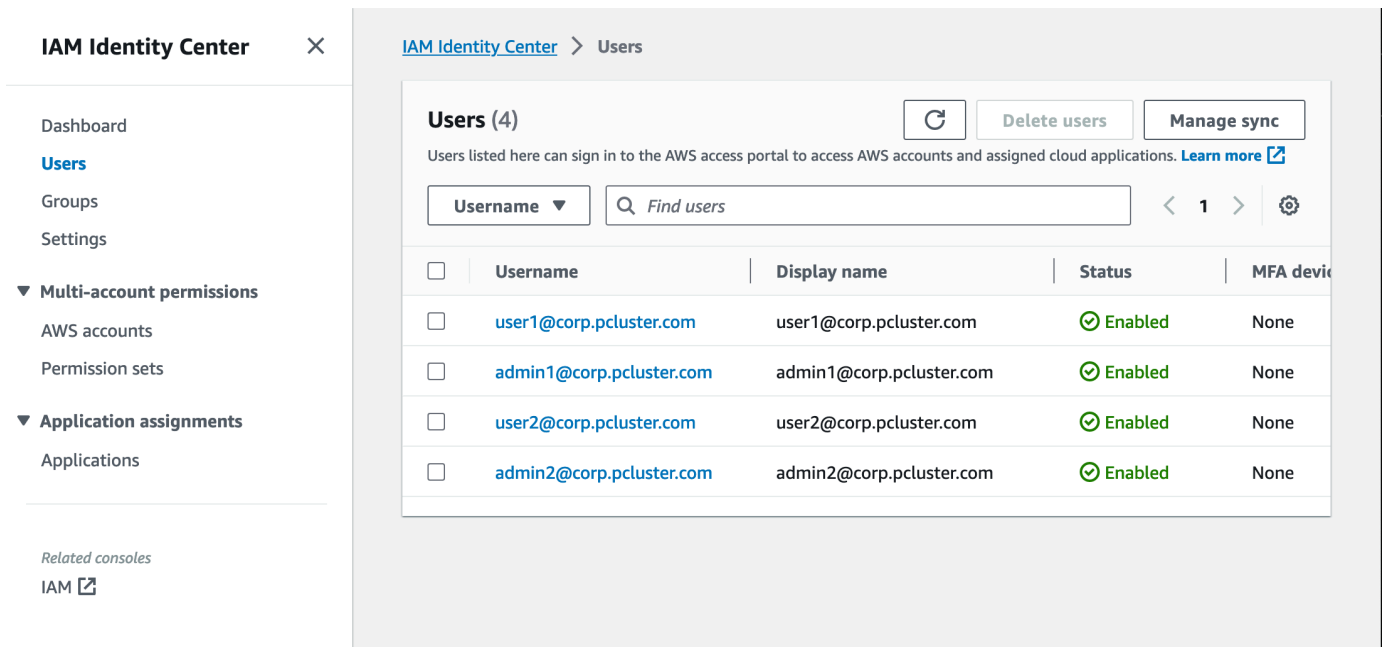
**Added users and groups (4)** **Remove**

| <input type="checkbox"/> | Username / Group name | Type | Domain            |
|--------------------------|-----------------------|------|-------------------|
| <input type="checkbox"/> | user1                 | User | corp.pcluster.com |
| <input type="checkbox"/> | user2                 | User | corp.pcluster.com |
| <input type="checkbox"/> | admin1                | User | corp.pcluster.com |
| <input type="checkbox"/> | admin2                | User | corp.pcluster.com |

**Cancel**    **Previous**    **Next**

5. Vérifiez vos modifications, puis cliquez sur Enregistrer la configuration
6. Si l'écran suivant affiche un avertissement indiquant que les utilisateurs ne sont pas synchronisés, sélectionnez le bouton Reprendre la synchronisation en haut à droite.
7. Ensuite, pour activer les utilisateurs, dans l'onglet Utilisateurs sur la gauche, sélectionnez un utilisateur, puis cliquez sur Activer l'accès utilisateur > Activer l'accès utilisateur

Remarque : vous devrez peut-être sélectionner Reprendre la synchronisation si une bannière d'avertissement apparaît en haut, puis attendre que les utilisateurs se synchronisent (essayez le bouton d'actualisation pour voir s'ils sont déjà synchronisés).



**IAM Identity Center** ×

[IAM Identity Center](#) > [Users](#)

**Users (4)** Refresh Delete users Manage sync

Users listed here can sign in to the AWS access portal to access AWS accounts and assigned cloud applications. [Learn more](#)

< 1 > Settings

| <input type="checkbox"/> | Username                                 | Display name             | Status  | MFA device |
|--------------------------|------------------------------------------|--------------------------|---------|------------|
| <input type="checkbox"/> | <a href="#">user1@corp.pcluster.com</a>  | user1@corp.pcluster.com  | Enabled | None       |
| <input type="checkbox"/> | <a href="#">admin1@corp.pcluster.com</a> | admin1@corp.pcluster.com | Enabled | None       |
| <input type="checkbox"/> | <a href="#">user2@corp.pcluster.com</a>  | user2@corp.pcluster.com  | Enabled | None       |
| <input type="checkbox"/> | <a href="#">admin2@corp.pcluster.com</a> | admin2@corp.pcluster.com | Enabled | None       |

*Related consoles*  
[IAM](#)

## Ajouter votre application à IAM Identity Center

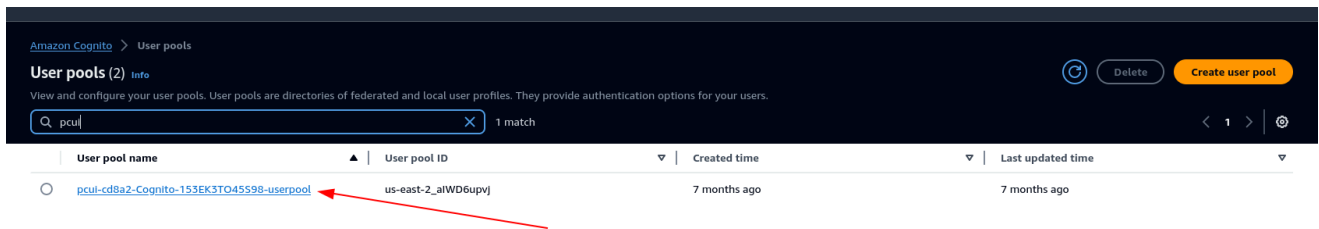
Une fois que vous avez synchronisé vos utilisateurs avec IAM Identity Center, vous devez ajouter une nouvelle application. Cela permet de configurer les applications compatibles SSO qui seront disponibles sur votre portail IAM Identity Center. Dans ce cas, nous ajouterons l'AWS ParallelClusterinterface utilisateur en tant qu'application tandis que IAM Identity Center sera le fournisseur d'identité.

L'étape suivante consiste à ajouter l'AWS ParallelClusterinterface utilisateur en tant qu'application dans IAM Identity Center. AWS ParallelCluster L'interface utilisateur est un portail Web qui aide l'utilisateur à gérer ses clusters. Pour plus d'informations, voir [AWS ParallelClusterUI](#).

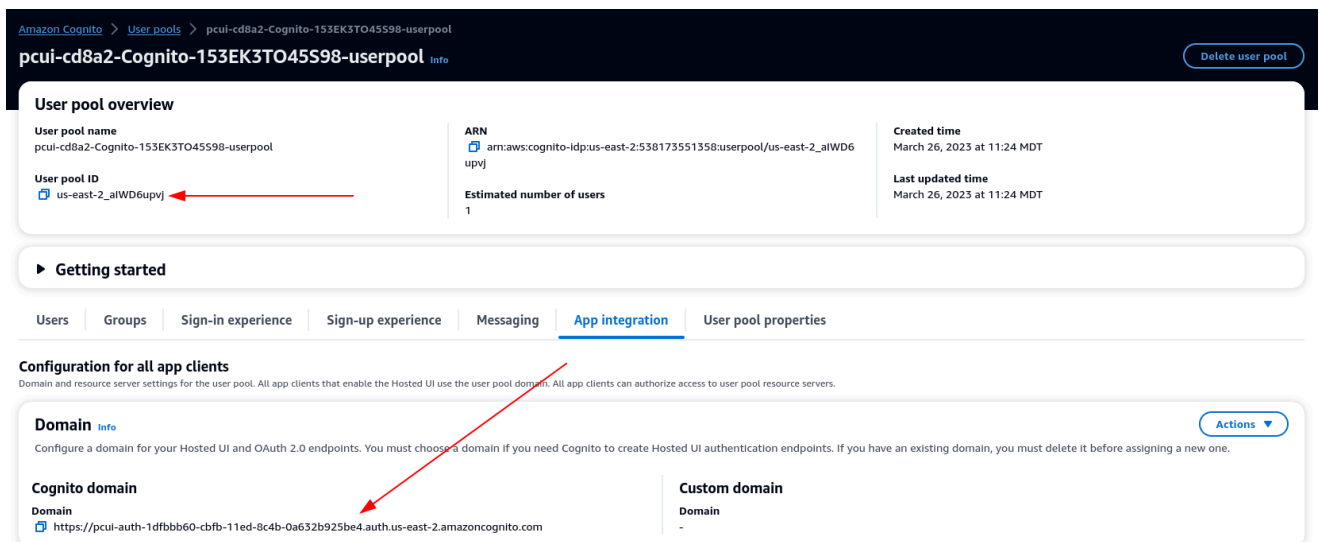
### Configuration de l'application dans Identity Center

1. Sous IAM Identity Center > Applications (dans la barre de menu de gauche, cliquez sur Applications)
2. Cliquez sur Ajouter une application
3. Sélectionnez Ajouter une application SAML 2.0 personnalisée
4. Cliquez sur Suivant
5. Sélectionnez le nom d'affichage et la description que vous souhaitez utiliser (par exemple, PCUI et AWS ParallelCluster interface utilisateur)

6. Sous métadonnées IAM Identity Center, copiez le lien vers le fichier de métadonnées SAML d'IAM Identity Center et enregistrez-le pour plus tard. Il sera utilisé lors de la configuration de l'authentification unique sur l'application Web
7. Sous Propriétés de l'application, dans l'URL de démarrage de l'application, saisissez votre adresse PCUI. Vous pouvez le trouver en accédant à la CloudFormation console, en sélectionnant la pile qui correspond à PCUI (par exemple parallelcluster-ui) et en accédant à l'onglet Outputs pour trouver UIURL ParallelCluster  
par exemple <https://m2iwazsi1j.execute-api.us-east-1.amazonaws.com>
8. Sous Métadonnées de l'application, choisissez Tapez manuellement vos valeurs de métadonnées. Fournissez ensuite les valeurs suivantes.
  - a. Important : assurez-vous de remplacer les valeurs du préfixe de domaine, de la région et de l'identifiant du pool d'utilisateurs par des informations spécifiques à votre environnement.
  - b. Le préfixe de domaine, la région et l'identifiant du pool d'utilisateurs peuvent être obtenus en ouvrant la console Amazon Cognito > Groupes d'utilisateurs



- c. Sélectionnez le groupe d'utilisateurs correspondant à PCUI (qui aura un nom de groupe d'utilisateurs tel que PCUI-CD8A2-Cognito-153EK3to45S98-UserPool)
- d. Accédez à l'intégration des applications



- URL du service ACS (Application Assertion Consumer Service) : `https://<domain-prefix>.auth.<region>.amazoncognito.com/saml2/idpresponse`

Audience SAML de l'application : `urn:amazon:cognito:sp : <userpool-id>`

- Sélectionnez Envoyer. Accédez ensuite à la page Détails de l'application que vous avez ajoutée.
- Sélectionnez la liste déroulante Actions et choisissez Modifier les mappages d'attributs. Fournissez ensuite les attributs suivants.
  - Attribut utilisateur dans l'application : sujet (Remarque : le sujet est prérempli.) → Correspond à cette valeur de chaîne ou à cet attribut utilisateur dans IAM Identity Center : `{user:email}`, Format : EmailAddress
  - Attribut utilisateur dans l'application : email → Correspond à cette valeur de chaîne ou à cet attribut utilisateur dans IAM Identity Center : `{user:email}`, Format : non spécifié

#### Attribute mappings for PCUI

Attributes you map here become part of the SAML assertion that is sent to the application. You can choose which user attributes in your application map to corresponding user attributes in your connected directory. [Learn more](#)

| User attribute in the application | Maps to this string value or user attribute in IAM Identity Center | Format       |        |
|-----------------------------------|--------------------------------------------------------------------|--------------|--------|
| Subject                           | <code>{user:email}</code>                                          | emailAddress |        |
| email                             | <code>{user:email}</code>                                          | unspecified  | Remove |
| Add new attribute mapping         |                                                                    |              |        |

Cancel Save changes

- Enregistrez vos modifications.
- Cliquez sur le bouton Attribuer des utilisateurs, puis assignez votre utilisateur à l'application. Il s'agit des utilisateurs de votre Active Directory qui auront accès à l'interface PCUI.

IAM Identity Center > Applications > PCUI

### PCUI

**Details** Actions

|  |                      |                                       |
|--|----------------------|---------------------------------------|
|  | Display name<br>PCUI | Description<br>AWS ParallelCluster UI |
|--|----------------------|---------------------------------------|

**Assigned users (1)** Remove access Assign Users

The following users and groups from your connected directory can access this application. [Learn more](#)

Search for an assigned user or group

| <input type="checkbox"/> | User/Group name          | Type |
|--------------------------|--------------------------|------|
| <input type="checkbox"/> | admin1@corp.pcluster.com | User |

## Configurer IAM Identity Center en tant qu'IdP SAML dans votre groupe d'utilisateurs

1. Dans les paramètres de votre groupe d'utilisateurs, sélectionnez **Expérience de connexion > Ajouter un fournisseur d'identité**

The screenshot shows the AWS IAM console interface for a user pool. The breadcrumb navigation is **Amazon Cognito > User pools > parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool**. The user pool name is **parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool** and the ID is **us-east-1\_Bgyu7LLz6**. The ARN is **arn:aws:cognito-idp:us-east-1:538173551358:userpool/us-east-1\_Bgyu7LLz6**. The estimated number of users is **1**. The user pool was created on **November 5, 2023 at 12:32 MST** and last updated on **November 5, 2023 at 12:32 MST**. The **Getting started** section includes tabs for **Users**, **Groups**, **Sign-in experience** (selected), **Sign-up experience**, **Messaging**, **App integration**, and **User pool properties**. The **Cognito user pool sign-in** section describes sign-in options (Email) and provides an **Add identity provider** button. The **Federated identity provider sign-in (0)** section includes a search bar and a table with columns for **Identity provider**, **Identity provider type**, **Created time**, and **Last updated time**. A red arrow points to the **Add identity provider** button in this section.

2. Choisissez un IdP SAML
3. Pour le nom du fournisseur, indiquez IdentityCenter
4. Sous **Source** du document de métadonnées, choisissez **Entrer l'URL du point de terminaison du document de métadonnées** et fournissez l'URL copiée lors de la configuration de l'application d'Identity Center
5. Sous **Attributs**, pour e-mail, choisissez e-mail



**SAML**  
Configure a SAML 2.0 identity provider for your user pool.

**Register your app with your SAML provider**  
To connect a SAML provider to Cognito, add your user pool as a relying party or application with your SAML 2.0 identity provider, and upload a metadata document to Cognito.

**Set up SAML federation with this user pool**

**Provider name** [Info](#)  
Enter a friendly name for your SAML 2.0 identity provider.  
IdentityCenter

**Identifiers - optional** [Info](#)  
Enter identifiers for this provider. Identifiers can be used to redirect users to the correct IdP in multitenant apps.  
Enter identifiers

Separate each identifier by a comma

**Sign-out flow** [Info](#)  
 Add sign-out flow  
Enable simultaneous sign-out from the SAML provider and Cognito.

**Metadata document source** [Info](#)  
Provide a SAML metadata document. This document is issued by your SAML provider. It includes the issuer's name, expiration information, and keys that can be used to validate the response from the identity provider.  
 Upload metadata document  
 Enter metadata document endpoint URL

**Enter metadata document endpoint URL** [Info](#)  
https://portal.sso.us-east-1.amazonaws.com/saml/metadata/NTM4MTczNTUxMzU4X2lucy0zNmZkNmQ3Y2NmMjU5ODM3

**Map attributes between your SAML provider and your user pool** [Info](#)  
Your required attributes are mapped to the equivalent SAML attributes. Each attribute you add must be mapped to a SAML attribute.

| User pool attribute | SAML attribute |
|---------------------|----------------|
| email               | email          |

[Add another attribute](#)

[Cancel](#) [Add identity provider](#)

## 6. Sélectionnez Ajouter un fournisseur d'identité.

### Intégrer l'IdP au client de l'application du pool d'utilisateurs

1. Ensuite, dans la section Intégration des applications de votre groupe d'utilisateurs, choisissez le client répertorié dans la liste des clients des applications

Amazon Cognito > User pools > parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool

## parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool Info

[Delete user pool](#)

### User pool overview

|                                                                           |                                                                                      |                                                           |
|---------------------------------------------------------------------------|--------------------------------------------------------------------------------------|-----------------------------------------------------------|
| <b>User pool name</b><br>parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool | <b>ARN</b><br>arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-east-1_Bgyu7Lz6 | <b>Created time</b><br>November 5, 2023 at 12:32 MST      |
| <b>User pool ID</b><br>us-east-1_Bgyu7Lz6                                 | <b>Estimated number of users</b><br>1                                                | <b>Last updated time</b><br>November 5, 2023 at 12:32 MST |

### Getting started

Users | Groups | Sign-in experience | Sign-up experience | Messaging | **App integration** | User pool properties

### Configuration for all app clients

Domain and resource server settings for the user pool. All app clients that enable the Hosted UI use the user pool domain. All app clients can authorize access to user pool resource servers.

#### Domain Info

Configure a domain for your Hosted UI and OAuth 2.0 endpoints. You must choose a domain if you need Cognito to create Hosted UI authentication endpoints. If you have an existing domain, you must delete it before assigning a new one.

[Actions](#)

|                                                                                                          |                      |
|----------------------------------------------------------------------------------------------------------|----------------------|
| <b>Cognito domain</b>                                                                                    | <b>Custom domain</b> |
| <b>Domain</b><br>https://pcul-auth-06f29200-7c12-11ee-b755-0e11297ecb0d.auth.us-east-1.amazoncognito.com | <b>Domain</b><br>-   |

#### Resource servers (0) Info

Configure resource servers. A resource server is a remote server that authorizes access based on OAuth 2.0 scopes in an access token.

[Edit](#) [Delete](#) [Create resource server](#)

| Resource server name | Resource server identifier | Custom scopes |
|----------------------|----------------------------|---------------|
| No resource servers  |                            |               |

[Create resource server](#)

#### App client defaults

Hosted UI customization and advanced security settings for the user pool. You can customize the Hosted UI and advanced security in app clients to override the defaults.

#### Hosted UI customization Info

Customize the hosted sign-up and sign-in pages to match your app's style and branding by uploading your own logo and customized CSS.

[Edit](#)

|                  |                        |
|------------------|------------------------|
| <b>Logo</b><br>- | <b>Custom CSS</b><br>- |
|------------------|------------------------|

#### Advanced security Info

Configure advanced security features, including Cognito's automatic responses to suspicious user activity. Advanced security adds cost to your bill. [See pricing](#)

[Enable](#)

**Status**  
 Disabled

#### App client list

The app clients that integrate your apps with your user pool. Configure client overrides to user pool default configurations, and configure Amazon Pinpoint analytics.

#### App clients and analytics (1) Info

Configure an app client. App clients are the user pool authentication resources attached to your app. Select an app client to configure the permitted authentication actions for an app.

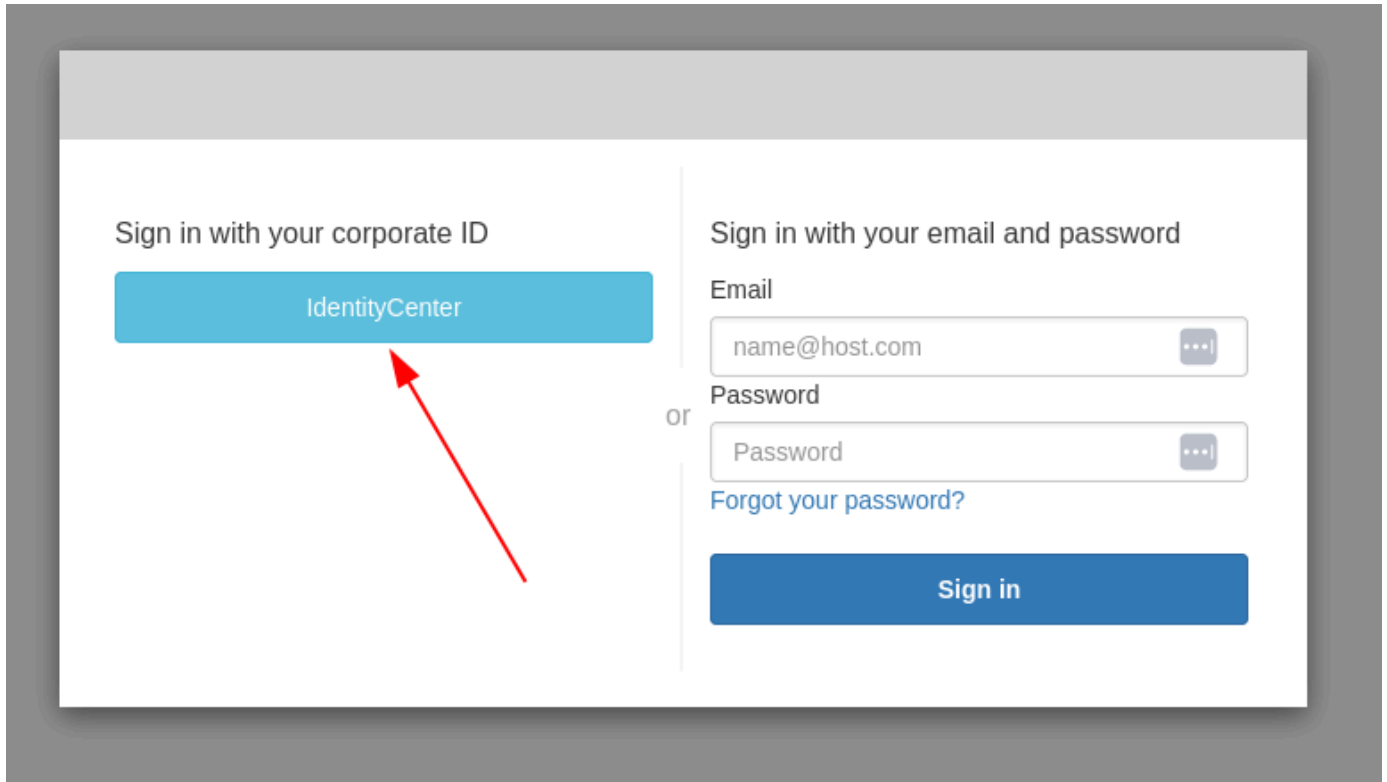
[Refresh](#) [Delete](#) [Create app client](#)

| App client name                                                     | Client ID |
|---------------------------------------------------------------------|-----------|
| <input type="radio"/> <a href="#">CognitoAppClient-ETqXbqI5wRVs</a> | ...       |

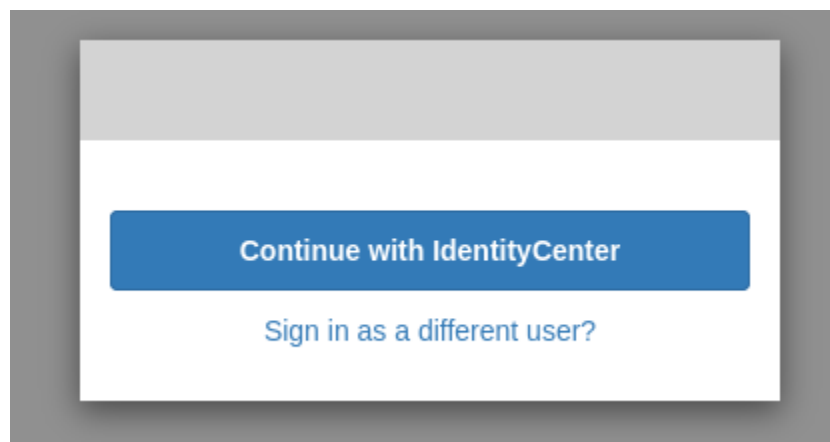
2. Sous Interface utilisateur hébergée, choisissez Modifier
3. Dans la section Identity, IdentityCenterles fournisseurs choisissent également.
4. Choisissez Enregistrer les modifications

Validez votre configuration

1. Ensuite, nous allons valider la configuration que nous venons de créer en nous connectant à PCUI. Connectez-vous à votre portail PCUI et vous devriez maintenant voir une option vous permettant de vous connecter avec votre identifiant d'entreprise :



2. Cliquez sur le IdentityCenter bouton pour accéder à la connexion IdP du IAM Identity Center, suivie d'une page contenant vos applications, y compris le PCUI. Ouvrez cette application.
3. Une fois que vous aurez atteint l'écran suivant, votre utilisateur aura été ajouté au groupe d'utilisateurs de Cognito.



Faites de votre utilisateur un administrateur

1. Accédez à présent à la console Amazon Cognito > Groupes d'utilisateurs et sélectionnez le nouvel utilisateur qui doit avoir le préfixe identitycenter

The screenshot shows the Amazon Cognito console interface for a user pool named 'parallelcluster-ui-Cognito-ZQRNCGGD3MHU-userpool'. The 'Users (2)' section is active, displaying a table of users. A red arrow points to the user 'identitycenter\_admin1@corp.pcluster.c...'.

| User name                                | Email address            | Email verified | Confirmation status | Status  |
|------------------------------------------|--------------------------|----------------|---------------------|---------|
| c8f7eccc-e647-4227-afa2-080274ebfefe     | user@amazon.com          | Yes            | Confirmed           | Enabled |
| identitycenter_admin1@corp.pcluster.c... | admin1@corp.pcluster.com | No             | External provider   | Enabled |

2. Sous Appartenances à des groupes, sélectionnez Ajouter un utilisateur au groupe, choisissez administrateur et cliquez sur Ajouter.
3. Maintenant, lorsque vous cliquez sur Continuer, IdentityCenter vous serez dirigé vers la page de l'AWS ParallelClusterinterface utilisateur.

# AWS ParallelCluster résolution des problèmes

La AWS ParallelCluster communauté gère une page Wiki qui fournit de nombreux conseils de résolution des problèmes sur le [AWS ParallelCluster GitHub Wiki](#). Pour obtenir la liste des problèmes connus, consultez la section [Problèmes connus](#).

## Rubriques

- [Essayer de créer un cluster](#)
- [Essayer d'exécuter une tâche](#)
- [Essayer de mettre à jour un cluster](#)
- [Essayer d'accéder au stockage](#)
- [Essayer de supprimer un cluster](#)
- [Essayer de mettre à niveau la pile AWS ParallelCluster d'API](#)
- [Observation des erreurs lors de l'initialisation des nœuds de calcul](#)
- [Résolution des problèmes liés aux indicateurs de santé du](#)
- [Résolution des problèmes de déploiement de clusters](#)
- [Résolution des problèmes de déploiement de clusters à l'aide de Terraform](#)
- [Résolution des problèmes de dimensionnement](#)
- [Problèmes liés aux groupes de placement et au lancement d'instances](#)
- [Répertoires qui ne peuvent pas être remplacés](#)
- [Résolution des problèmes dans NICE DCV](#)
- [Résolution des problèmes dans les clusters avec AWS Batch intégration](#)
- [Résolution des problèmes d'intégration multi-utilisateurs avec Active Directory](#)
- [Résolution des problèmes liés aux AMI personnalisées](#)
- [Résolution d'un délai d'expiration de mise à jour du cluster en cas cfn-hup d'inexécution](#)
- [Dépannage du réseau](#)
- [La mise à jour du cluster a échoué lors d'onNodeUpdatedune action personnalisée](#)
- [Voir les erreurs liées à la Slurm configuration personnalisée](#)
- [Alarmes de cluster](#)
- [Support supplémentaire](#)

## Essayer de créer un cluster

Lorsque vous utilisez AWS ParallelCluster la version 3.5.0 ou ultérieure pour créer un cluster et que la création d'un cluster a échoué avec `--rollback-on-failure` set to `false`, utilisez la commande [`pcluster describe-cluster`](#) CLI pour obtenir des informations sur l'état et les défaillances. Dans ce cas, le `pcluster describe-cluster` résultat attendu `clusterStatus` est `CREATE_FAILED`. Consultez la `failures` section de la sortie pour trouver le `failureCode` et `failureReason`. Ensuite, dans la section suivante, recherchez la solution correspondante `failureCode` pour obtenir une aide supplémentaire en matière de dépannage. Pour plus d'informations, consultez [`pcluster describe-cluster`](#).

Dans les sections suivantes, nous vous recommandons de consulter les journaux du nœud principal, tels que les `/var/log/chef-client.log` fichiers `/var/log/cfn-init.log` et. Pour plus d'informations sur AWS ParallelCluster les journaux et sur la façon de les consulter, consultez [Journaux clés pour le débogage](#) et [Récupération et conservation des journaux](#).

Si vous n'en avez pas `failureCode`, accédez à la AWS CloudFormation console pour afficher la pile du cluster. Vérifiez les `Status Reason` défaillances `HeadNodeWaitCondition` ou sur d'autres ressources pour obtenir des informations supplémentaires sur les défaillances. Pour plus d'informations, consultez [Afficher AWS CloudFormation les événements sur CREATE\\_FAILED](#). Vérifiez les `/var/log/chef-client.log` fichiers `/var/log/cfn-init.log` et sur le nœud principal.

### **failureCode est OnNodeConfiguredExecutionFailure**

- Pourquoi a-t-il échoué ?

Vous avez fourni un script personnalisé dans la section `OnNodeConfigured` du nœud principal de la configuration pour créer un cluster. Cependant, le script personnalisé n'a pas pu être exécuté.

- Comment résoudre le problème ?

Consultez le `/var/log/cfn-init.log` fichier pour en savoir plus sur l'échec et sur la manière de résoudre le problème dans votre script personnalisé. Vers la fin de ce journal, des informations relatives à l'exécution du `OnNodeConfigured` script peuvent s'afficher après le `Running command runpostinstall` message.

## failureCode est OnNodeConfiguredDownloadFailure

- Pourquoi a-t-il échoué ?

Vous avez fourni un script personnalisé dans la section `OnNodeConfigured` du nœud principal de la configuration pour créer un cluster. Cependant, le script personnalisé n'a pas pu être téléchargé.

- Comment résoudre le problème ?

Assurez-vous que l'URL est valide et que l'accès est correctement configuré. Pour plus d'informations sur la configuration des scripts bootstrap personnalisés, consultez [Actions de bootstrap personnalisées](#).

Vérifiez le `/var/log/cfn-init.log` fichier. Vers la fin de ce journal, des informations relatives à l'exécution du `OnNodeConfigured` script, notamment au téléchargement, peuvent s'afficher après le `Running command runpostinstall` message.

## failureCode est OnNodeConfiguredFailure

- Pourquoi a-t-il échoué ?

Vous avez fourni un script personnalisé dans la section `OnNodeConfigured` du nœud principal de la configuration pour créer un cluster. Cependant, l'utilisation du script personnalisé a échoué lors du déploiement du cluster. Aucune cause immédiate ne peut être déterminée et une enquête supplémentaire est nécessaire.

- Comment résoudre le problème ?

Vérifiez le `/var/log/cfn-init.log` fichier. Vers la fin de ce journal, vous pouvez voir des informations d'exécution relatives au traitement du `OnNodeConfigured` script après le `Running command runpostinstall` message.

## failureCode est OnNodeStartExecutionFailure

- Pourquoi a-t-il échoué ?

Vous avez fourni un script personnalisé dans la section `OnNodeStart` du nœud principal de la configuration pour créer un cluster. Cependant, le script personnalisé n'a pas pu être exécuté.

- Comment résoudre le problème ?

Consultez le `/var/log/cfn-init.log` fichier pour en savoir plus sur l'échec et sur la manière de résoudre le problème dans votre script personnalisé. Vers la fin de ce journal, des informations relatives à l'exécution du `OnNodeStart` script peuvent s'afficher après le `Running command runpreinstall` message.

## failureCode est OnNodeStartDownloadFailure

- Pourquoi a-t-il échoué ?

Vous avez fourni un script personnalisé dans la section `OnNodeStart` du nœud principal de la configuration pour créer un cluster. Cependant, le script personnalisé n'a pas pu être téléchargé.

- Comment résoudre le problème ?

Assurez-vous que l'URL est valide et que l'accès est correctement configuré. Pour plus d'informations sur la configuration des scripts bootstrap personnalisés, consultez [Actions de bootstrap personnalisées](#).

Vérifiez le `/var/log/cfn-init.log` fichier. Vers la fin de ce journal, des informations relatives à l'exécution du `OnNodeStart` script, notamment au téléchargement, peuvent s'afficher après le `Running command runpreinstall` message.

## failureCode est OnNodeStartFailure

- Pourquoi a-t-il échoué ?

Vous avez fourni un script personnalisé dans la section `OnNodeStart` du nœud principal de la configuration pour créer un cluster. Cependant, l'utilisation du script personnalisé a échoué lors du déploiement du cluster. Aucune cause immédiate ne peut être déterminée et une enquête supplémentaire est nécessaire.

- Comment résoudre le problème ?

Vérifiez le `/var/log/cfn-init.log` fichier. Vers la fin de ce journal, vous pouvez voir des informations d'exécution relatives au traitement du `OnNodeStart` script après le `Running command runpreinstall` message.



## failureCode est EbsMountFailure

- Pourquoi a-t-il échoué ?

Le volume EBS défini dans la configuration du cluster n'a pas pu être monté.

- Comment résoudre le problème ?

Consultez le `/var/log/chef-client.log` fichier pour obtenir des informations détaillées sur l'échec.

## failureCode est EfsMountFailure

- Pourquoi a-t-il échoué ?

Le volume Amazon EFS défini dans la configuration du cluster n'a pas pu être monté.

- Comment résoudre le problème ?

Si vous avez défini un système de fichiers Amazon EFS existant, assurez-vous que le trafic est autorisé entre le cluster et le système de fichiers. Pour plus d'informations, consultez [SharedStorage/EfsSettings/FileSystemId](#).

Consultez le `/var/log/chef-client.log` fichier pour obtenir des informations détaillées sur l'échec.

## failureCode est FsxMountFailure

- Pourquoi a-t-il échoué ?

Le système de fichiers Amazon FSx défini dans la configuration du cluster n'a pas pu être monté.

- Comment résoudre le problème ?

Si vous avez défini un système de fichiers Amazon FSx existant, assurez-vous que le trafic est autorisé entre le cluster et le système de fichiers. Pour plus d'informations, consultez [SharedStorage/FsxLustreSettings/FileSystemId](#).

Consultez le `/var/log/chef-client.log` fichier pour obtenir des informations détaillées sur l'échec.

## failureCode est RaidMountFailure

- Pourquoi a-t-il échoué ?

Les volumes RAID définis dans la configuration du cluster n'ont pas pu être montés.

- Comment résoudre le problème ?

Consultez le `/var/log/chef-client.log` fichier pour obtenir des informations détaillées sur l'échec.

## failureCode est AmiVersionMismatch

- Pourquoi a-t-il échoué ?

La AWS ParallelCluster version utilisée pour créer l'AMI personnalisée est différente de AWS ParallelCluster celle utilisée pour configurer le cluster. Dans la CloudFormation console, consultez les détails de la CloudFormation pile de clusters et cochez la case « Status Reason for HeadNodeWaitCondition the » pour obtenir des informations supplémentaires sur les AWS ParallelCluster versions et l'AMI. Pour plus d'informations, consultez [Afficher AWS CloudFormation les événements sur CREATE\\_FAILED](#).

- Comment résoudre le problème ?

Assurez-vous que la AWS ParallelCluster version utilisée pour créer l'AMI personnalisée est la même que celle AWS ParallelCluster utilisée pour configurer le cluster. Vous pouvez modifier la version personnalisée de l'AMI ou la version de la `pcluster` CLI pour les rendre identiques.

## failureCode est InvalidAmi

- Pourquoi a-t-il échoué ?

L'AMI personnalisée n'est pas valide car elle n'a pas été créée à l'aide de AWS ParallelCluster.

- Comment résoudre le problème ?

Utilisez la `pcluster build-image` commande pour créer une AMI en faisant de votre AMI l'image parent. Pour plus d'informations, consultez [pcluster build-image](#).

## **failureCode** porte **HeadNodeBootstrapFailure** la mention « **failureReason** Impossible de configurer le nœud principal ».

- Pourquoi a-t-il échoué ?

Aucune cause immédiate ne peut être déterminée et une enquête supplémentaire est nécessaire. Par exemple, il se peut que le cluster soit protégé, ce qui peut être dû à un échec du provisionnement du parc informatique statique.

- Comment résoudre le problème ?

Consultez le `/var/log/chef-client.log` fichier pour obtenir des informations détaillées sur l'échec.

### Note

Si vous voyez `RuntimeError` une `exceptionCluster state has been set to PROTECTED mode due to failures detected in static node provisioning`, le cluster est protégé. Pour plus d'informations, consultez [Comment déboguer le mode protégé](#).

## **failureCode** est que **HeadNodeBootstrapFailure** le délai de création **failureReason** du cluster est expiré.

- Pourquoi a-t-il échoué ?

Par défaut, la création du cluster est limitée à 30 minutes. Si la création du cluster n'est pas terminée dans ce délai, la création du cluster échoue avec une erreur de temporisation. La création du cluster peut être interrompue pour différentes raisons. Par exemple, les délais d'expiration peuvent être dus à un échec de création d'un nœud principal, à un problème réseau, à l'exécution de scripts personnalisés trop longs dans le nœud principal, à une erreur dans un script personnalisé exécuté dans les nœuds de calcul ou à de longs délais d'attente pour le provisionnement du nœud de calcul. Aucune cause immédiate ne peut être déterminée et une enquête supplémentaire est nécessaire.

- Comment résoudre le problème ?

Consultez les `/var/log/chef-client.log` fichiers `/var/log/cfn-init.log` et pour obtenir des informations détaillées sur les défaillances. Pour plus d'informations sur AWS ParallelCluster les journaux et sur la façon de les obtenir, consultez [Journaux clés pour le débogage](#) et [Récupération et conservation des journaux](#).

Vous découvrirez peut-être ce qui suit dans ces journaux.

- Vu **Waiting for static fleet capacity provisioning** vers la fin du **chef-client.log**

Cela indique que le délai de création du cluster a expiré lors de l'attente du démarrage des nœuds statiques. Pour plus d'informations, consultez [Observation des erreurs lors de l'initialisation des nœuds de calcul](#).

- Le script Seeing **OnNodeConfigured** ou **OnNodeStart** Head Node n'est pas terminé à la fin du **cfn-init.log**

Cela indique que l'exécution du script `OnNodeConfigured` ou du script `OnNodeStart` personnalisé a pris du temps et a provoqué une erreur de temporisation. Vérifiez que votre script personnalisé ne présente aucun problème susceptible de provoquer son exécution prolongée. Si l'exécution de votre script personnalisé prend du temps, pensez à modifier le délai d'expiration en ajoutant une `DevSettings` section au fichier de configuration de votre cluster, comme illustré dans l'exemple suivant :

```
DevSettings:
 Timeouts:
 HeadNodeBootstrapTimeout: 1800 # default setting: 1800 seconds
```

- Impossible de trouver les journaux ou le nœud principal n'a pas été créé correctement

Il est possible que le nœud principal n'ait pas été créé correctement et que les journaux soient introuvables. Dans la CloudFormation console, consultez les détails de la pile de clusters pour vérifier s'il existe des informations supplémentaires sur les défaillances.

## **failureCodeest HeadNodeBootstrapFailure** associé à **failureReason Failed to bootstrap the head node.**

- Pourquoi a-t-il échoué ?

Aucune cause immédiate ne peut être déterminée et une enquête supplémentaire est nécessaire.

- Comment résoudre le problème ?

Vérifiez les `/var/log/chef-client.log` fichiers `/var/log/cfn-init.log` et.

## **failureCode est ResourceCreationFailure**

- Pourquoi a-t-il échoué ?

La création de certaines ressources a échoué lors du processus de création du cluster. La panne peut survenir pour diverses raisons. Par exemple, les échecs de création de ressources peuvent être dus à des problèmes de capacité ou à une politique IAM mal configurée.

- Comment résoudre le problème ?

Dans la CloudFormation console, consultez la pile de clusters pour vérifier les détails supplémentaires relatifs à l'échec de création de ressources.

## **failureCode est ClusterCreationFailure**

- Pourquoi a-t-il échoué ?

Aucune cause immédiate ne peut être déterminée et une enquête supplémentaire est nécessaire.

- Comment résoudre le problème ?

Dans la CloudFormation console, consultez la pile du cluster et vérifiez la présence de Status Reason `HeadNodeWaitCondition` pour trouver des informations supplémentaires sur les défaillances.

Vérifiez les `/var/log/chef-client.log` fichiers `/var/log/cfn-init.log` et.

## **Voir WaitCondition timed out... dans la CloudFormation pile**

Pour plus d'informations, consultez [failureCode est que HeadNodeBootstrapFailure le délai de création failureReason du cluster est expiré..](#)

## Voir **Resource creation cancelled** dans la CloudFormation pile

Pour plus d'informations, consultez [failureCode est ResourceCreationFailure](#).

## Erreurs **Failed to run cfn-init...** visibles ou autres dans la AWS CloudFormation pile

Consultez le `/var/log/cfn-init.log` et `/var/log/chef-client.log` pour obtenir des informations supplémentaires sur les défaillances.

## Voir **chef-client.log** se termine par **INFO: Waiting for static fleet capacity provisioning**

Cela est lié au délai de création du cluster lorsque vous attendez que les nœuds statiques s'allument. Pour plus d'informations, consultez [Observation des erreurs lors de l'initialisation des nœuds de calcul](#).

## Voyant **Failed to run preinstall or postinstall in cfn-init.log**

Vous avez un `OnNodeStart` script `OnNodeConfigured` or dans la `HeadNode` section de configuration du cluster. Le script ne fonctionne pas correctement. Consultez le `/var/log/cfn-init.log` fichier pour obtenir des informations détaillées sur les erreurs de script personnalisées.

## Voir **This AMI was created with xxx, but is trying to be used with xxx...** dans la CloudFormation pile

Pour plus d'informations, consultez [failureCode est AmiVersionMismatch](#).

## Voir **This AMI was not baked by AWS ParallelCluster...** dans la CloudFormation pile

Pour plus d'informations, consultez [failureCode est InvalidAmi](#).

## Voir que `pcluster create-cluster` la commande ne s'exécute pas localement

Consultez le fichier `~/.parallelcluster/pcluster-cli.log` de votre système de fichiers local pour obtenir des informations sur les défaillances.

### Support supplémentaire

Suivez les instructions de dépannage dans [Résolution des problèmes de déploiement de clusters](#).

Vérifiez si votre scénario est couvert dans la section [Problèmes GitHub connus](#) AWS ParallelCluster sur le site On GitHub.

Pour obtenir une assistance supplémentaire, consultez [Support supplémentaire](#).

## Essayer d'exécuter une tâche

**`srun` la tâche interactive échoue avec une erreur `srun: error: fwd_tree_thread: can't find address for <host>, check slurm.conf`**

- Pourquoi a-t-il échoué ?

Vous avez exécuté la `srun` commande pour soumettre une tâche, puis vous avez augmenté la taille d'une file d'attente en utilisant la `pcluster update-cluster` commande sans redémarrer les Slurm démons une fois la mise à jour terminée.

Slurm organise Slurm les démons dans une arborescence afin d'optimiser la communication. Cette hiérarchie n'est mise à jour que lorsque les démons démarrent.

Supposons que vous `srun` lanciez une tâche, puis que vous exécutiez la `pcluster update-cluster` commande pour augmenter la taille de la file d'attente. De nouveaux nœuds de calcul sont lancés dans le cadre de la mise à jour. Ensuite, placez votre tâche en Slurm file d'attente sur l'un des nouveaux nœuds de calcul. Dans ce cas, les Slurm daemons et les `srun` ne détectent pas les nouveaux nœuds de calcul. `srun` renvoie une erreur car il ne détecte pas les nouveaux nœuds.

- Comment résoudre le problème ?

Redémarrez les Slurm démons sur tous les nœuds de calcul, puis utilisez-les `srun` pour soumettre votre tâche. Vous pouvez planifier le redémarrage Slurm des démons en exécutant la `scontrol reboot` commande qui redémarre les nœuds de calcul. Pour plus d'informations, consultez [scontrol reboot](#) dans la Slurm documentation. Vous pouvez également redémarrer manuellement les Slurm démons sur les nœuds de calcul en demandant le redémarrage des services correspondant `systemd`.

## Job bloqué dans son **CF** état avec **squeue** la commande

Cela peut être dû au démarrage des nœuds dynamiques. Pour plus d'informations, consultez [Observation des erreurs lors de l'initialisation des nœuds de calcul](#).

## Exécuter des travaux à grande échelle et voir **nfsd: too many open connections, consider increasing the number of threads in /var/log/messages**

Avec un système de fichiers en réseau, lorsque les limites du réseau sont atteintes, le temps d'attente des E/S augmente également. Cela peut entraîner des blocages logiciels, car le réseau est utilisé pour écrire des données à la fois pour le réseau et pour les métriques d'E/S.

Avec les instances de 5e génération, nous utilisons le pilote ENA pour exposer les compteurs de paquets. Ces compteurs comptent les paquets façonnés AWS lorsque le réseau atteint les limites de bande passante de l'instance. Vous pouvez vérifier ces compteurs pour voir s'ils sont supérieurs à 0. Si tel est le cas, cela signifie que vous avez dépassé vos limites de bande passante. Vous pouvez consulter ces compteurs en courant `ethtool -S eth0 | grep exceeded`.

Le dépassement des limites du réseau est souvent dû à la prise en charge d'un trop grand nombre de connexions NFS. C'est l'une des premières choses à vérifier lorsque vous atteignez ou dépassez les limites du réseau.

Par exemple, le résultat suivant montre les packages abandonnés :

```
$ ethtool -S eth0 | grep exceeded
bw_in_allowance_exceeded: 38750610
bw_out_allowance_exceeded: 1165693
pps_allowance_exceeded: 103
contrack_allowance_exceeded: 0
```



```
linklocal_allowance_exceeded: 0
```

Pour éviter de recevoir ce message, envisagez de remplacer le type d'instance du nœud principal par un type d'instance plus performant. Envisagez de déplacer votre stockage de données vers des systèmes de fichiers de stockage partagés qui ne sont pas exportés sous forme de partage NFS, tels qu'Amazon EFS ou Amazon FSx. Pour plus d'informations, consultez [Stockage partagé](#) les [meilleures pratiques](#) sur le AWS ParallelCluster Wiki sur GitHub.

## Exécution d'une tâche MPI

### Activation du mode de débogage

Pour activer le mode de débogage d'OpenMPI, [voir Quelles sont les commandes d'Open MPI qui facilitent le débogage ?](#)

Pour activer le mode de débogage IntelMPI, consultez la section [Autres](#) variables d'environnement.

Voir **MPI\_ERRORS\_ARE\_FATAL** et **OPAL\_ERROR** intégrer le résultat du travail

Ces codes d'erreur proviennent de la couche MPI de votre application. Pour savoir comment obtenir les journaux de débogage MPI à partir de votre application, consultez. [Activation du mode de débogage](#)

Cette erreur peut être due au fait que votre application a été compilée pour une implémentation MPI spécifique, telle qu'OpenMPI, et que vous essayez de l'exécuter avec une autre implémentation MPI, telle qu'IntelMPI. Assurez-vous de compiler et d'exécuter votre application avec la même implémentation MPI.

### Utilisation **mpirun** avec DNS géré désactivé

Pour les clusters créés avec [SlurmSettings/Dns/DisableManagedDnsset UseEc2Hostnames définis sur true](#), le nom du Slurm nœud n'est pas résolu par le DNS. Slurm peut démarrer des processus MPI lorsqu'ils n'ont pas de nœuds et si la tâche MPI est exécutée dans un contexte. Slurm Nous vous recommandons de suivre les instructions du [guide de l'utilisateur Slurm MPI](#) pour exécuter des tâches MPI avec Slurm.

## Essayer de mettre à jour un cluster

### **pcluster update-cluster** la commande ne s'exécute pas localement

Consultez le fichier `~/.parallelcluster/pcluster-cli.log` de votre système de fichiers local pour obtenir des informations sur les défaillances.

### Voir **clusterStatus** c'est **UPDATE\_FAILED** avec **pcluster describe-cluster** commande

Si la mise à jour de la pile de clusters est annulée, consultez le `/var/log/chef-client.logs` fichier pour obtenir des informations sur l'erreur.

Vérifiez si votre problème est mentionné dans la section [Problèmes GitHub connus](#) à l'adresse AWS ParallelCluster On GitHub.

### Le délai de mise à jour du cluster a expiré

Il peut s'agir d'un problème lié à l'`cf-n-hup` exécution. Si le `cf-n-hup` démon est arrêté pour une cause externe, il n'est pas redémarré automatiquement. S'il `cf-n-hup` n'est pas en cours d'exécution, lors d'une mise à jour du cluster, la CloudFormation pile lance le processus de mise à jour comme prévu, mais la procédure de mise à jour n'est pas activée sur le nœud principal et le déploiement de la pile finit par expirer. Pour plus d'informations, consultez [Résolution d'un délai d'expiration de mise à jour du cluster en cas cf-n-hup d'inexécution](#) la section pour résoudre le problème et résoudre le problème.

## Essayer d'accéder au stockage

### Utilisation d'un système de fichiers externe Amazon FSx for Lustre

Assurez-vous que le trafic est autorisé entre le cluster et le système de fichiers. Le système de fichiers doit être associé à un groupe de sécurité qui autorise le trafic TCP entrant et sortant via les ports 988, 1021, 1022 et 1023. Pour plus d'informations sur la configuration des groupes de sécurité, consultez [FileSystemId](#).

### Utilisation d'un système de fichiers Amazon Elastic File System externe

Assurez-vous que le trafic est autorisé entre le cluster et le système de fichiers. Le système de fichiers doit être associé à un groupe de sécurité qui autorise le trafic TCP entrant et sortant via les

ports 988, 1021, 1022 et 1023. Pour plus d'informations sur la configuration des groupes de sécurité, consultez [FileSystemId](#).

## Essayer de supprimer un cluster

La **pcluster delete-cluster** commande ne s'exécute pas localement

Vérifiez le `~/ .parallelcluster/pcluster-cli.log` fichier dans votre système de fichiers local.

La pile de clusters ne parvient pas à être supprimée

Si la pile de clusters ne parvient pas à être supprimée, consultez le message relatif aux événements de la CloudFormation pile.

Vérifiez si votre problème est mentionné dans la section [Problèmes GitHub connus](#) AWS ParallelCluster sur Activé GitHub.

## Essayer de mettre à niveau la pile AWS ParallelCluster d'API

Vérifiez si votre problème est mentionné dans la section [Problèmes GitHub connus](#) AWS ParallelCluster sur Activé GitHub.

## Observation des erreurs lors de l'initialisation des nœuds de calcul

Voir **Node bootstrap error** dans **clustermgtd.log**

Le problème est lié à l'échec du démarrage des nœuds de calcul. Pour plus d'informations sur le débogage d'un problème lié au mode protégé par cluster, consultez [Comment déboguer le mode protégé](#).

J'ai configuré des réservations de capacité à la demande (ODCR) ou des instances réservées zonales

ODCR qui incluent des instances dotées de plusieurs interfaces réseau, telles que P4d, P4de et Trainium (Trn) AWS

Dans le fichier de configuration du cluster, vérifiez que le HeadNode se trouve dans un sous-réseau public et que les nœuds de calcul se trouvent dans un sous-réseau privé.

## Les ODCR sont des ODCR ciblés

**Unable to read file '/opt/slurm/etc/pcluster/run\_instances\_overrides.json'**. Même si je l'ai déjà mis **/opt/slurm/etc/pcluster/run\_instances\_overrides.json** en place, en suivant les instructions données dans [Lancez des instances avec des réservations de capacité à la demande \(ODCR\)](#)

Si vous utilisez les AWS ParallelCluster versions 3.1.1 à 3.2.1 avec des ODCR ciblés et que vous utilisez également le fichier JSON de remplacement des [instances d'exécution, il est possible que le fichier JSON ne soit pas correctement formaté](#). Une erreur peut s'afficher `clustermgtd.log`, telle que la suivante :

```
Unable to read file '/opt/slurm/etc/pcluster/run_instances_overrides.json'.
Using default: {} in /var/log/parallelcluster/clustermgtd.
```

Vérifiez que le format de fichier JSON est correct en exécutant ce qui suit :

```
$ echo /opt/slurm/etc/pcluster/run_instances_overrides.json | jq
```

Voir **Found RunInstances parameters override** en `clustermgtd.log` cas d'échec de la création du cluster ou en `slurm_resume.log` cas d'échec de la tâche d'exécution

Si vous utilisez le [fichier Run Instances Override JSON](#), vérifiez que vous avez correctement défini le nom de la file d'attente et le nom des ressources de calcul dans le `/opt/slurm/etc/pcluster/run_instances_overrides.json` fichier.

Voir **An error occurred (InsufficientInstanceCapacity)slurm\_resume.log** quand je ne parviens pas à exécuter une tâche, ou `clustermgtd.log` quand je ne parviens pas à créer un cluster

### Utilisation du PG-ODCR (groupe de placement ODCR)

Lorsque vous créez un ODCR avec un groupe de placement associé, le même nom de groupe de placement doit être utilisé dans le fichier de configuration. Définissez le [nom du groupe de placement](#) correspondant dans la configuration du cluster.

### Utilisation d'instances réservées zonales

Si vous utilisez des instances réservées zonales avec `PlacementGroup/Enabledto true` dans la configuration du cluster, une erreur peut s'afficher, telle que la suivante :

We currently do not have sufficient `trn1.32xlarge` capacity in the Availability Zone you requested (`us-east-1d`). Our system will be working on provisioning additional capacity.

You can currently get `trn1.32xlarge` capacity by not specifying an Availability Zone in your request or choosing `us-east-1a`, `us-east-1b`, `us-east-1c`, `us-east-1e`, `us-east-1f`.

Cela peut être dû au fait que les instances réservées zonales ne sont pas placées dans le même UC (ou épine dorsale), ce qui peut entraîner des erreurs de capacité insuffisantes (ICE) lors de l'utilisation de groupes de placement. Vous pouvez vérifier ce cas en désactivant le paramètre `PlacementGroup` Groupe dans la configuration du cluster afin de déterminer si le cluster peut allouer les instances.

Voir **An error occurred (VcpuLimitExceeded)slurm\_resume.log** quand je ne parviens pas à exécuter une tâche, ou quand je ne parviens pas à créer un cluster **clustermgtd.log**

Vérifiez les limites de vCPU de votre compte pour le type d'instance Amazon EC2 spécifique que vous utilisez. Si vous ne voyez aucun vCPU ou moins que ce que vous demandez, demandez une augmentation de vos limites. Pour plus d'informations sur la façon de consulter les limites actuelles et de demander de nouvelles limites, consultez les [quotas de service Amazon EC2](#) dans le guide de l'utilisateur Amazon EC2.

Voir **An error occurred (InsufficientInstanceCapacity)slurm\_resume.log** quand je ne parviens pas à exécuter une tâche, ou quand je ne parviens pas à créer un cluster **clustermgtd.log**

Vous rencontrez un problème de capacité insuffisante. Suivez <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors/> pour résoudre le problème.

Voir que les nœuds sont en **DOWN** état avec **Reason (Code:InsufficientInstanceCapacity)...**

Vous rencontrez un problème de capacité insuffisante. Suivez <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors/> pour résoudre le problème. Pour

plus d'informations sur AWS ParallelCluster le mode de basculement rapide en cas de capacité insuffisante, consultez. [Slurmbasculement rapide d'une capacité insuffisante du cluster](#)

## Voir `cannot change locale (en_US.utf-8) because it has an invalid name` dans `slurm_resume.log`

Cela peut se produire en cas d'échec du processus yum d'installation qui a laissé les paramètres régionaux dans un état incohérent. Cela peut se produire, par exemple, lorsqu'un utilisateur met fin au processus d'installation.

Pour en vérifier la cause, effectuez les actions suivantes :

- Exécutez `su - pcluster-admin`.

Le shell affiche une erreur, telle que `cannot change locale...no such file or directory`.

- Exécutez `localedef --list`.

Renvoie une liste vide ou ne contient pas les paramètres régionaux par défaut.

- Vérifiez la dernière yum commande avec `yum history` et `yum history info #ID`. Est-ce que la dernière pièce d'identité existe `Return-Code: Success` ?

Si le dernier ID n'en a pas `Return-Code: Success`, les scripts de post-installation ne se sont peut-être pas exécutés correctement.

Pour résoudre le problème, essayez de reconstruire les paramètres régionaux avec `yum reinstall glibc-all-langpacks`. Après la reconstruction, aucun message d'erreur ou d'avertissement `su - pcluster-admin` ne s'affiche si le problème est résolu.

## Aucun des scénarios précédents ne s'applique à ma situation

Pour résoudre les problèmes d'initialisation des nœuds de calcul, consultez. [Résolution des problèmes d'initialisation des nœuds](#)

Vérifiez si votre scénario est couvert dans la section [Problèmes GitHub connus](#) AWS ParallelCluster sur le site On GitHub.

Pour obtenir une assistance supplémentaire, consultez [Support supplémentaire](#).

# Résolution des problèmes liés aux indicateurs de santé du

Les métriques de santé du cluster sont ajoutées au tableau de CloudWatch bord AWS ParallelCluster Amazon à partir de AWS ParallelCluster la version 3.6.0. Dans les sections suivantes, vous découvrirez les indicateurs de santé du tableau de bord et les mesures que vous pouvez prendre pour résoudre les problèmes.

## Rubriques

- [Voir le graphique des erreurs de provisionnement des instances](#)
- [Affichage du graphique des erreurs d'instance non conformes](#)
- [Voir le graphique des temps d'inactivité de la flotte de calcul](#)

## Voir le graphique des erreurs de provisionnement des instances

Si vous voyez une valeur différente de zéro dans le Instance Provisioning Errors graphique, cela signifie que l'instance Amazon EC2 de sauvegarde des nœuds slurm n'a pas pu être lancée sur l'API or. CreateFleet RunInstance

### Voyant **IAMPolicyErrors**

- Que s'est-il passé ?

Un certain nombre d'instances n'ont pas pu être lancées, en raison d'autorisations insuffisantes accompagnées d'un code d'erreurUnauthorizedOperation.

- Comment résoudre le problème ?

Si vous avez configuré un [InstanceRole](#) ou personnalisé [InstanceProfile](#), vérifiez vos politiques IAM et vérifiez que vous utilisez les informations d'identification correctes.

Consultez le `clustermgtd` fichier pour obtenir des informations détaillées sur les erreurs du nœud statique. Consultez le `slurm_resume.log` fichier pour obtenir des informations détaillées sur les erreurs de nœud dynamique. Utilisez les informations pour en savoir plus sur les autorisations manquantes qui doivent être ajoutées.

### Voyant **VcpuLimitErrors**

- Que s'est-il passé ?

AWS ParallelCluster n'a pas réussi à lancer les instances car la limite de vCPU que vous avez fixée Compte AWS pour un type d'instance Amazon EC2 spécifique que vous avez configuré pour les nœuds de calcul en cluster a été atteint.

- Comment résoudre le problème ?

Vérifiez l'`VcpuLimitExceeded` erreur dans le `clustermgtd` fichier pour les nœuds statiques et dans le `slurm_resume.log` fichier pour les nœuds dynamiques pour obtenir des informations supplémentaires. Pour résoudre ce problème, vous pouvez demander une augmentation des limites de vos vCPU. Pour plus d'informations sur la façon de consulter les limites actuelles et de demander de nouvelles limites, consultez les [quotas de service Amazon Elastic Compute Cloud](#) dans le guide de l'utilisateur Amazon Elastic Compute Cloud pour les instances Linux.

## Voyant **VolumeLimitErrors**

- Que s'est-il passé ?

Vous avez atteint la limite de volume Amazon EBS sur votre Compte AWS, et AWS ParallelCluster vous ne parvenez pas à lancer des instances avec un code d'erreur `InsufficientVolumeCapacity` ou `VolumeLimitExceeded`.

- Comment résoudre le problème ?

Vérifiez le `clustermgtd` fichier pour les nœuds statiques et pour les `slurm_resume.log` nœuds dynamiques pour obtenir des informations supplémentaires sur les limites de volume. Pour résoudre ce problème, vous pouvez utiliser un autre volume Région AWS, nettoyer les volumes existants ou contacter le AWS Support Center pour soumettre une demande d'augmentation de votre limite de volume Amazon EBS.

## Voyant **InsufficientCapacityErrors**

- Que s'est-il passé ?

AWS ParallelCluster ne dispose pas d'une capacité suffisante pour lancer des instances Amazon EC2 sur des nœuds principaux.

- Comment résoudre le problème ?



Vérifiez le `clustermgtd` fichier pour les nœuds statiques et pour les nœuds dynamiques afin d'obtenir des informations détaillées sur les erreurs de capacité insuffisante. `slurm_resume.log`

Pour résoudre le problème, suivez les instructions à l'adresse <https://aws.amazon.com/premiumsupport/knowledge-center/ec2-insufficient-capacity-errors/>.

## OtherInstanceLaunchFailures

- Que s'est-il passé ?

L'instance Amazon EC2 de sauvegarde des nœuds de calcul n'a pas pu être lancée avec l'API `CreateFleet` or `RunInstance`.

- Comment résoudre le problème ?

Vérifiez le `clustermgtd` fichier pour les nœuds statiques et pour les `slurm_resume.log` nœuds dynamiques pour obtenir des informations sur les erreurs.

## Affichage du graphique des erreurs d'instance non conformes

- Que s'est-il passé ?

Un certain nombre d'instances de calcul ont été lancées mais ont par la suite été interrompues pour cause de défaillance.

- Comment résoudre le problème ?

Pour plus d'informations sur le dépannage des nœuds défectueux, consultez [Résolution des problèmes de remplacement et de terminaison inattendus de nœuds](#).

## Voyant **InstanceBootstrapTimeoutError**

- Que s'est-il passé ?

Une instance ne peut pas rejoindre le cluster au sein du `resume_timeout` (pour les nœuds dynamiques) ou `node_replacement_timeout` (pour les nœuds statiques). Cela peut se produire si le réseau n'est pas configuré correctement pour les nœuds de calcul, ou si les scripts personnalisés exécutés sur le nœud de calcul mettent trop de temps à se terminer.

- Comment résoudre le problème ?

Pour les nœuds dynamiques, vérifiez dans le `clustermgtd` journal (`/var/log/parallelcluster/clustermgtd`) l'adresse IP du nœud de calcul et les erreurs telles que les suivantes :

```
Node bootstrap error: Resume timeout expires for node
```

Pour les nœuds statiques, vérifiez dans le `clustermgtd` journal (`/var/log/parallelcluster/clustermgtd`) l'adresse IP du nœud de calcul et les erreurs telles que les suivantes :

```
Node bootstrap error: Replacement timeout expires for node ... in replacement.
```

Pour plus de détails, vérifiez que le `/var/log/cloud-init-output.log` fichier ne contient pas d'erreurs. Vous pouvez récupérer les adresses IP des nœuds de calcul problématiques dans les fichiers `slurm_resume` journaux `clustermgtd` et.

## Voyant **EC2HealthCheckErrors**

- Que s'est-il passé ?

Le bilan de santé d'une instance a échoué sur Amazon EC2.

- Comment résoudre le problème ?

Pour plus d'informations sur la façon de résoudre ce problème, consultez [Résoudre les problèmes des instances dont les vérifications d'état ont échoué](#).

## Voyant **ScheduledEventHealthCheckErrors**

- Que s'est-il passé ?

Une instance a échoué lors d'une vérification de l'état d'un événement planifié par Amazon EC2, et elle ne fonctionne pas correctement.

- Comment résoudre le problème ?

Pour plus d'informations sur la manière de résoudre ce problème, consultez la section [Événements planifiés pour vos instances](#).

## Voyant **NoCorrespondingInstanceErrors**

- Que s'est-il passé ?

AWS ParallelCluster Impossible de trouver les instances qui soutiennent les nœuds. Les nœuds se sont probablement terminés automatiquement lors des opérations d'amorçage. [SlurmQueues](#) Des erreurs de [OnNodeConfiguredscript](#) [CustomActions//OnNodeStart](#) ou de réseau peuvent se produire **NoCorrespondingInstanceErrors**.

- Comment résoudre le problème ?

Pour plus de détails, consultez `/var/log/cloud-init-output.log` le nœud de calcul.

## Voir le graphique des temps d'inactivité de la flotte de calcul

Observer un **MaxDynamicNodeIdleTime** délai nettement supérieur au seuil de réduction du temps d'inactivité

- Que s'est-il passé ?

Votre instance ne s'arrête pas correctement. `MaxDynamicNodeIdleTime` indique la durée maximale en secondes pendant laquelle un nœud dynamique, soutenu par une instance Amazon EC2, est inactif. Le seuil de réduction du temps d'inactivité est dérivé du paramètre de configuration [ScaledownIdleTime](#) du cluster. Lorsqu'un nœud de calcul est inactif pendant plus de quelques secondes, `Scaledown` met le nœud hors Slurm tension et AWS ParallelCluster met fin à l'instance de sauvegarde. Dans ce cas, quelque chose empêche la fermeture de l'instance.

- Comment résoudre le problème ?

Pour plus d'informations sur ce problème, voir [Remplacement, arrêt ou mise hors tension des instances et des nœuds problématiques](#) dans [Résolution des problèmes de dimensionnement](#).

## Résolution des problèmes de déploiement de clusters

Si votre cluster ne parvient pas à être créé et annule la création de la pile, vous pouvez consulter les fichiers journaux pour diagnostiquer le problème. Le message d'échec ressemble probablement au résultat suivant :

```
$ pcluster create-cluster --cluster-name mycluster --region eu-west-1 \
```

```
--cluster-configuration cluster-config.yaml
{
 "cluster": {
 "clusterName": "mycluster",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f01-11ec-a3b9-024fcc6f3387",
 "region": "eu-west-1",
 "version": "3.7.0",
 "clusterStatus": "CREATE_IN_PROGRESS"
 }
}

$ pcluster describe-cluster --cluster-name mycluster --region eu-west-1
{
 "creationTime": "2021-09-06T11:03:47.696Z",
 ...
 "cloudFormationStackStatus": "ROLLBACK_IN_PROGRESS",
 "clusterName": "mycluster",
 "computeFleetStatus": "UNKNOWN",
 "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f01-11ec-a3b9-024fcc6f3387",
 "lastUpdatedTime": "2021-09-06T11:03:47.696Z",
 "region": "eu-west-1",
 "clusterStatus": "CREATE_FAILED"
}
```

## Rubriques

- [Afficher AWS CloudFormation les événements sur CREATE\\_FAILED](#)
- [Utiliser la CLI pour afficher les flux de journaux](#)
- [Recréez le cluster défaillant avec rollback-on-failure](#)

## Afficher AWS CloudFormation les événements sur **CREATE\_FAILED**

Vous pouvez utiliser la console ou la AWS ParallelCluster CLI pour afficher les CloudFormation événements relatifs aux CREATE\_FAILED erreurs afin d'en trouver la cause première.

## Rubriques

- [Afficher les événements dans la CloudFormation console](#)
- [Utilisez la CLI pour afficher et filtrer CloudFormation les événements sur CREATE\\_FAILED](#)

## Afficher les événements dans la CloudFormation console

Pour obtenir plus d'informations sur la cause de ce "CREATE\_FAILED" statut, vous pouvez utiliser la CloudFormation console.

Afficher les messages CloudFormation d'erreur depuis la console.

1. Connectez-vous au AWS Management Console et accédez à <https://console.aws.amazon.com/cloudformation>.
2. Sélectionnez la pile nommée *cluster\_name*.
3. Choisissez l'onglet Événements.
4. Vérifiez l'état de la ressource dont la création a échoué en faisant défiler la liste des événements de ressource par ID logique. Si la création d'une sous-tâche a échoué, revenez en arrière pour trouver l'événement de ressource ayant échoué.
5. Par exemple, si le message d'état suivant s'affiche, vous devez utiliser des types d'instances qui ne dépasseront pas votre limite de vCPU actuelle ou qui ne demanderont pas une capacité de vCPU accrue.

```
2022-02-04 16:09:44 UTC-0800 HeadNode CREATE_FAILED You have requested more vCPU
capacity than your current vCPU limit of 0 allows
 for the instance bucket that the specified instance type belongs to. Please
visit http://aws.amazon.com/contact-us/ec2-request to request an adjustment to
this limit.
 (Service: AmazonEC2; Status Code: 400; Error Code: VcpuLimitExceeded; Request
ID: a9876543-b321-c765-d432-dcba98766789; Proxy: null).
```

## Utilisez la CLI pour afficher et filtrer CloudFormation les événements sur CREATE\_FAILED

Pour diagnostiquer le problème de création de clusters, vous pouvez utiliser la [pcluster get-cluster-stack-events](#) commande en filtrant par CREATE\_FAILED état. Pour plus d'informations, consultez la section [Filtrage AWS CLI de la sortie](#) dans le guide de AWS Command Line Interface l'utilisateur.

```
$ pcluster get-cluster-stack-events --cluster-name mycluster --region eu-west-1 \
 --query 'events[?resourceStatus==`CREATE_FAILED`]'
[
 {
```

```

 "eventId": "3ccdedd0-0f03-11ec-8c06-02c352fe2ef9",
 "physicalResourceId": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f02-11ec-a3b9-024fcc6f3387",
 "resourceStatus": "CREATE_FAILED",
 "resourceStatusReason": "The following resource(s) failed to create: [HeadNode].
",
 "stackId": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f02-11ec-a3b9-024fcc6f3387",
 "stackName": "mycluster",
 "logicalResourceId": "mycluster",
 "resourceType": "AWS::CloudFormation::Stack",
 "timestamp": "2021-09-06T11:11:51.780Z"
 },
 {
 "eventId": "HeadNode-CREATE_FAILED-2021-09-06T11:11:50.127Z",
 "physicalResourceId": "i-04e91cc1f4ea796fe",
 "resourceStatus": "CREATE_FAILED",
 "resourceStatusReason": "Received FAILURE signal with UniqueId
i-04e91cc1f4ea796fe",
 "resourceProperties": "{\"LaunchTemplate\":{\"Version\":\"1\"},\"LaunchTemplateId
\":\"lt-057d2b1e687f05a62\"}}",
 "stackId": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f02-11ec-a3b9-024fcc6f3387",
 "stackName": "mycluster",
 "logicalResourceId": "HeadNode",
 "resourceType": "AWS::EC2::Instance",
 "timestamp": "2021-09-06T11:11:50.127Z"
 }
]

```

Dans l'exemple précédent, l'échec était lié à la configuration du nœud principal.

## Utiliser la CLI pour afficher les flux de journaux

Pour résoudre ce type de problème, vous pouvez répertorier les flux de journaux disponibles depuis le nœud principal [pcluster list-cluster-log-streams](#) en filtrant `node-type` puis en analysant le contenu des flux de journaux.

```

$ pcluster list-cluster-log-streams --cluster-name mycluster --region eu-west-1 \
--filters 'Name=node-type,Values=HeadNode'
{
 "logStreams": [

```

```

{
 "logStreamArn": "arn:aws:logs:eu-west-1:xxx:log-group:/aws/parallelcluster/
mycluster-202109061103:log-stream:ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init",
 "logStreamName": "ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init",
 ...
},
{
 "logStreamArn": "arn:aws:logs:eu-west-1:xxx:log-group:/aws/parallelcluster/
mycluster-202109061103:log-stream:ip-10-0-0-13.i-04e91cc1f4ea796fe.chef-client",
 "logStreamName": "ip-10-0-0-13.i-04e91cc1f4ea796fe.chef-client",
 ...
},
{
 "logStreamArn": "arn:aws:logs:eu-west-1:xxx:log-group:/aws/parallelcluster/
mycluster-202109061103:log-stream:ip-10-0-0-13.i-04e91cc1f4ea796fe.cloud-init",
 "logStreamName": "ip-10-0-0-13.i-04e91cc1f4ea796fe.cloud-init",
 ...
},
...
]
}

```

Les deux principaux flux de log que vous pouvez utiliser pour détecter les erreurs d'initialisation sont les suivants :

- `cfn-init` est le journal du `cfn-init` script. Vérifiez d'abord ce flux de journal. Vous verrez probablement l'`Command chef failed` erreur dans ce journal. Regardez les lignes juste avant cette ligne pour plus de détails liés au message d'erreur. Pour plus d'informations, consultez [cfn-init](#).
- `cloud-init` est le journal de [cloud-init](#). Si rien ne s'y trouve avec `cfn-init`, essayez ensuite de consulter ce journal.

Vous pouvez récupérer le contenu du flux de log en utilisant [pcluster get-cluster-log-events](#) (notez l'`--limit 5` option permettant de limiter le nombre d'événements récupérés) :

```

$ pcluster get-cluster-log-events --cluster-name mycluster \
 --region eu-west-1 --log-stream-name ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init \
 --limit 5
{
 "nextToken": "f/36370880979637159565202782352491087067973952362220945409/s",
 "prevToken": "b/36370880752972385367337528725601470541902663176996585497/s",

```

```
"events": [
 {
 "message": "2021-09-06 11:11:39,049 [ERROR] Unhandled exception during build:
Command runpostinstall failed",
 "timestamp": "2021-09-06T11:11:39.049Z"
 },
 {
 "message": "Traceback (most recent call last):\n File \"/opt/aws/bin/
cfn-init\"", line 176, in <module>\n worklog.build(metadata, configSets)\n File \"/usr/lib/python3.7/site-packages/cfnbootstrap/construction.py\"", line
135, in build\n Contractor(metadata).build(configSets, self)\n File \"/
usr/lib/python3.7/site-packages/cfnbootstrap/construction.py\"", line 561, in
build\n self.run_config(config, worklog)\n File \"/usr/lib/python3.7/
site-packages/cfnbootstrap/construction.py\"", line 573, in run_config\n CloudFormationCarpenter(config, self._auth_config).build(worklog)\n File \"/usr/
lib/python3.7/site-packages/cfnbootstrap/construction.py\"", line 273, in build\n self._config.commands)\n File \"/usr/lib/python3.7/site-packages/cfnbootstrap/
command_tool.py\"", line 127, in apply\n raise ToolError(u"Command %s failed" %
name)",
 "timestamp": "2021-09-06T11:11:39.049Z"
 },
 {
 "message": "cfnbootstrap.construction_errors.ToolError: Command runpostinstall
failed",
 "timestamp": "2021-09-06T11:11:39.049Z"
 },
 {
 "message": "2021-09-06 11:11:49,212 [DEBUG] CloudFormation client initialized
with endpoint https://cloudformation.eu-west-1.amazonaws.com",
 "timestamp": "2021-09-06T11:11:49.212Z"
 },
 {
 "message": "2021-09-06 11:11:49,213 [DEBUG] Signaling resource HeadNode in stack
mycluster with unique ID i-04e91cc1f4ea796fe and status FAILURE",
 "timestamp": "2021-09-06T11:11:49.213Z"
 }
]
}
```

Dans l'exemple précédent, l'échec est dû à un `runpostinstall` échec. Il est donc strictement lié au contenu du script bootstrap personnalisé utilisé dans le paramètre de `OnNodeConfigured` configuration du [CustomActions](#).



## Recréez le cluster défaillant avec **rollback-on-failure**

AWS ParallelCluster crée des flux de CloudWatch journaux de cluster dans des groupes de journaux. Vous pouvez consulter ces journaux dans les tableaux de bord personnalisés ou les groupes de journaux de la CloudWatch console. Pour plus d'informations, consultez [Intégration avec Amazon CloudWatch Logs avec Amazon Logs](#) et [Tableau de CloudWatch bord Amazon](#). Si aucun flux de journal n'est disponible, l'échec peut être dû au script de démarrage [CustomActions](#) personnalisé ou à un problème lié à l'AMI. Pour diagnostiquer le problème de création dans ce cas, créez à nouveau le cluster en utilisant [pcluster create-cluster](#), y compris le `--rollback-on-failure` paramètre défini sur `false`. Utilisez ensuite SSH pour afficher le cluster, comme indiqué ci-dessous :

```
$ pcluster create-cluster --cluster-name mycluster --region eu-west-1 \
 --cluster-configuration cluster-config.yaml --rollback-on-failure false
{
 "cluster": {
 "clusterName": "mycluster",
 "cloudformationStackStatus": "CREATE_IN_PROGRESS",
 "cloudformationStackArn": "arn:aws:cloudformation:eu-west-1:xxx:stack/
mycluster/1bf6e7c0-0f01-11ec-a3b9-024fcc6f3387",
 "region": "eu-west-1",
 "version": "3.7.0",
 "clusterStatus": "CREATE_IN_PROGRESS"
 }
}
$ pcluster ssh --cluster-name mycluster
```

Une fois connecté au nœud principal, vous devriez trouver trois fichiers journaux principaux que vous pouvez utiliser pour trouver l'erreur.

- `/var/log/cfn-init.log` est le journal du `cfn-init` script. Vérifiez d'abord ce journal. Il est probable que vous rencontriez une erreur comme `Command chef failed` dans ce journal. Regardez les lignes juste avant cette ligne pour plus de détails liés au message d'erreur. Pour plus d'informations, consultez [cfn-init](#).
- `/var/log/cloud-init.log` est le journal de [cloud-init](#). Si rien ne s'y trouve dans `cfn-init.log`, essayez ensuite de consulter ce journal.

- `/var/log/cloud-init-output.log` est le résultat des commandes exécutées par [cloud-init](#). Cela inclut la sortie `deconf-init`. Dans la plupart des cas, il n'est pas nécessaire de consulter ce journal pour résoudre ce type de problème.

## Résolution des problèmes de déploiement de clusters à l'aide de Terraform

Cette section concerne les clusters déployés à l'aide de Terraform.

### ParallelCluster API introuvable

La planification peut échouer car l' API ParallelCluster est introuvable. Dans ce cas, l'erreur renvoyée serait quelque chose comme :

```
Planning failed. Terraform encountered an error while generating this plan.

#
Error: Unable to retrieve ParallelCluster API cloudformation stack.
#
with provider["registry.terraform.io/aws-tf/aws-parallelcluster"],
on providers.tf line 6, in provider "aws-parallelcluster":
6: provider "aws-parallelcluster" {
#
operation error CloudFormation: DescribeStacks, https response error StatusCode: 400,
RequestID: REQUEST_ID, api error ValidationError: Stack with id PCAPI_STACK_NAME does
not exist
```

Pour résoudre cette erreur, déployez l' API ParallelCluster dans le compte sur lequel les clusters vont être créés. veuillez consulter [the section called "Création d'un cluster avec Terraform"](#).

### L'utilisateur n'est pas autorisé à appeler ParallelCluster l'API

La planification peut échouer car le rôle/utilisateur IAM que vous avez supposé pour déployer votre projet Terraform n'est pas autorisé à interagir avec l'API. ParallelCluster Dans ce cas, l'erreur renvoyée serait quelque chose comme :

```
Planning failed. Terraform encountered an error while generating this plan.

Error: 403 Forbidden
```

```
#
with
module.parallelcluster_clusters.module.clusters[0].pcluster_cluster.managed_configs["DemoCluster"]
on .terraform/modules/parallelcluster_clusters/modules/clusters/main.tf line 35, in
resource "pcluster_cluster" "managed_configs":
35: resource "pcluster_cluster" "managed_configs" {
#
{"Message":"User: USER_ARN is not authorized to perform: execute-api:Invoke on
resource: PC_API_REST_RESOURCE with an explicit deny"}
}
```

Pour résoudre cette erreur, configurez le ParallelCluster fournisseur de manière à ce qu'il utilise le rôle d' ParallelCluster API pour interagir avec l'API.

```
provider "aws-parallelcluster" {
 region = var.region
 profile = var.profile
 api_stack_name = var.api_stack_name
 use_user_role **= true**
}
```

## Résolution des problèmes de dimensionnement

Cette section concerne les clusters installés à l'aide de la AWS ParallelCluster version 3.0.0 ou ultérieure avec le planificateur de Slurm tâches. Pour plus d'informations sur la configuration de plusieurs files d'attente, consultez [Configuration de plusieurs files d'attente](#).

Si l'un de vos clusters en cours d'exécution rencontre des problèmes, mettez-le dans un STOPPED état en exécutant la commande suivante avant de commencer le dépannage. Cela évite d'encourir des coûts imprévus.

```
$ pcluster update-compute-fleet --cluster-name mycluster \
 --status STOP_REQUESTED
```

Vous pouvez répertorier les flux de journaux disponibles à partir des nœuds du cluster à l'aide de la [pcluster list-cluster-log-streams](#) commande et en filtrant à l'private-dns-nameaide de l'un des nœuds défailants ou du nœud principal :

```
$ pcluster list-cluster-log-streams --cluster-name mycluster --region eu-west-1 \
 --filters 'Name=private-dns-name,Values=ip-10-0-0-101'
```

Vous pouvez ensuite récupérer le contenu du flux de journaux pour l'analyser en utilisant la [`pcluster get-cluster-log-events`](#) commande et en transmettant le `--log-stream-name` correspondant à l'un des journaux clés mentionnés dans la section suivante :

```
$ pcluster get-cluster-log-events --cluster-name mycluster \
--region eu-west-1 --log-stream-name ip-10-0-0-13.i-04e91cc1f4ea796fe.cfn-init
```

AWS ParallelCluster crée des flux de CloudWatch journaux de cluster dans des groupes de journaux. Vous pouvez consulter ces journaux dans les tableaux de bord personnalisés ou les groupes de journaux de la CloudWatch console. Pour plus d'informations, consultez [Intégration avec Amazon CloudWatch Logs avec Amazon Logs](#) et [Tableau de CloudWatch bord Amazon](#).

## Rubriques

- [Journaux clés pour le débogage](#)
- [Affichage InsufficientInstanceCapacity d'une erreur slurm\\_resume.log lorsque je ne parviens pas à exécuter une tâche ou clustermgtd.log lorsque je ne parviens pas à créer un cluster](#)
- [Résolution des problèmes d'initialisation des nœuds](#)
- [Résolution des problèmes de remplacement et de terminaison inattendus de nœuds](#)
- [Remplacement, arrêt ou mise hors tension des instances et des nœuds problématiques](#)
- [InactiveÉtat de la file d'attente \(partition\)](#)
- [Résolution d'autres problèmes connus liés aux nœuds et aux tâches](#)

## Journaux clés pour le débogage

Le tableau suivant fournit une vue d'ensemble des journaux clés du nœud principal :

- `/var/log/cfn-init.log`- Voici le journal AWS CloudFormation d'initialisation. Il contient toutes les commandes qui ont été exécutées lors de la configuration d'une instance. Utilisez-le pour résoudre les problèmes d'initialisation.
- `/var/log/chef-client.log`- Ceci est le journal du client Chef. Il contient toutes les commandes qui ont été exécutées via CHEF/CINC. Utilisez-le pour résoudre les problèmes d'initialisation.
- `/var/log/parallelcluster/slurm_resume.log`- C'est un ResumeProgram journal. Il lance des instances pour les nœuds dynamiques. Utilisez-le pour résoudre les problèmes de lancement des nœuds dynamiques.

- `/var/log/parallelcluster/slurm_suspend.log`- C'est le SuspendProgram journal. Il est appelé lorsque des instances sont résiliées pour des nœuds dynamiques. Utilisez-le pour résoudre les problèmes de terminaison des nœuds dynamiques. Lorsque vous consultez ce journal, vous devez également le `clustermgtd` consulter.
- `/var/log/parallelcluster/clustermgtd`- C'est le `clustermgtd` journal. Il s'exécute en tant que démon centralisé qui gère la plupart des actions des opérations de cluster. Utilisez-le pour résoudre les problèmes de lancement, de terminaison ou de fonctionnement du cluster.
- `/var/log/slurmctld.log`- Il s'agit du journal du démon de Slurm contrôle. AWS ParallelCluster ne prend pas de décisions de mise à l'échelle. Il tente plutôt de lancer des ressources uniquement pour répondre aux Slurm exigences. Il est utile pour les problèmes de dimensionnement et d'allocation, les problèmes liés au travail et les problèmes de lancement et de résiliation liés au planificateur.
- `/var/log/parallelcluster/compute_console_output`- Ce journal enregistre la sortie de console d'un exemple de sous-ensemble de nœuds de calcul statiques qui se sont arrêtés de façon inattendue. Utilisez ce journal si les nœuds de calcul statiques se terminent et que les journaux des nœuds de calcul ne sont pas disponibles dans CloudWatch. Le `compute_console_output log` contenu que vous recevez est le même lorsque vous utilisez la console Amazon EC2 ou AWS CLI pour récupérer la sortie de la console d'instance.

Voici les principaux journaux des nœuds de calcul :

- `/var/log/cloud-init-output.log`- Il s'agit du journal [cloud-init](#). Il contient toutes les commandes qui ont été exécutées lors de la configuration d'une instance. Utilisez-le pour résoudre les problèmes d'initialisation.
- `/var/log/parallelcluster/computemgtd`- C'est le `computemgtd` journal. Il s'exécute sur chaque nœud de calcul pour surveiller le nœud dans le cas peu fréquent où le `clustermgtd` démon du nœud principal est hors ligne. Utilisez-le pour résoudre les problèmes de résiliation inattendus.
- `/var/log/slurmd.log`- Il s'agit du journal du démon de Slurm calcul. Utilisez-le pour résoudre les problèmes d'initialisation et de défaillance du calcul.

## Affichage `InsufficientInstanceCapacity` d'une erreur `slurm_resume.log` lorsque je ne parviens pas à exécuter une tâche ou `clustermgtd.log` lorsque je ne parviens pas à créer un cluster

Si le cluster utilise un Slurm planificateur, vous rencontrez un problème de capacité insuffisante. S'il n'y a pas suffisamment d'instances disponibles lorsqu'une demande de lancement d'instance est faite, une `InsufficientInstanceCapacity` erreur est renvoyée.

Pour la capacité d'instance statique, vous pouvez trouver l'erreur dans le `clustermgtd` journal à l'adresse `/var/log/parallelcluster/clustermgtd`.

Pour ce qui est de la capacité dynamique de l'instance, vous pouvez trouver l'erreur dans le `ResumeProgram` journal à l'adresse `/var/log/parallelcluster/slurm_resume.log`.

Le message ressemble à l'exemple suivant :

```
An error occurred (InsufficientInstanceCapacity) when calling the RunInstances/
CreateFleet operation...
```

En fonction de votre cas d'utilisation, envisagez d'utiliser l'une des méthodes suivantes pour éviter de recevoir ce type de message d'erreur :

- Désactivez le groupe de placement s'il est activé. Pour plus d'informations, consultez [Problèmes liés aux groupes de placement et au lancement d'instances](#).
- Réservez de la capacité pour les instances et lancez-les avec ODCR (On-Demand Capacity Reservations). Pour plus d'informations, consultez [Lancez des instances avec des réservations de capacité à la demande \(ODCR\)](#).
- Configurez plusieurs ressources de calcul avec différents types d'instances. Si votre charge de travail ne nécessite aucun type d'instance spécifique, vous pouvez tirer parti du basculement rapide en cas de capacité insuffisante avec plusieurs ressources de calcul. Pour plus d'informations, consultez [Slurmbasculement rapide d'une capacité insuffisante du cluster](#).
- Configurez plusieurs types d'instances dans la même ressource de calcul et tirez parti de l'allocation de plusieurs types d'instances. Pour plus d'informations sur la configuration de plusieurs instances, consultez [Allocation de plusieurs types d'instances avec Slurm](#) et [Scheduling/SlurmQueues/ComputeResources/Instances](#).
- Déplacez la file d'attente vers une autre zone de disponibilité en modifiant l'ID de sous-réseau dans la configuration du cluster [Scheduling/SlurmQueues/Networking/SubnetIds](#).

- Si votre charge de travail n'est pas étroitement liée, répartissez la file d'attente sur différentes zones de disponibilité. Pour plus d'informations sur la configuration de plusieurs sous-réseaux, consultez [Scheduling//SlurmQueuesNetworking/SubnetIds](#).

## Résolution des problèmes d'initialisation des nœuds

Cette section explique comment résoudre les problèmes d'initialisation des nœuds. Cela inclut les problèmes où le nœud ne parvient pas à démarrer, à démarrer ou à rejoindre un cluster.

### Rubriques

- [Nœud principal](#)
- [Nœuds de calcul](#)

### Nœud principal

Journaux applicables :

- `/var/log/cfn-init.log`
- `/var/log/chef-client.log`
- `/var/log/parallelcluster/clustermgtd`
- `/var/log/parallelcluster/slurm_resume.log`
- `/var/log/slurmctld.log`

Consultez les `/var/log/chef-client.log` journaux `/var/log/cfn-init.log` et ou les flux de journaux correspondants. Ces journaux contiennent toutes les actions exécutées lors de la configuration du nœud principal. La plupart des erreurs survenant lors de l'installation doivent contenir des messages d'erreur dans le `/var/log/chef-client.log` journal. Si `OnNodeStart` des `OnNodeConfigured` scripts sont spécifiés dans la configuration du cluster, vérifiez que le script s'exécute correctement via les messages du journal.

Lorsqu'un cluster est créé, le nœud principal doit attendre que les nœuds de calcul rejoignent le cluster avant de pouvoir le rejoindre. De ce fait, si les nœuds de calcul ne parviennent pas à rejoindre le cluster, le nœud principal échoue également. Vous pouvez suivre l'un de ces ensembles de procédures, en fonction du type de notes de calcul que vous utilisez, pour résoudre ce type de problème :

## Nœuds de calcul

- Journaux applicables :
  - `/var/log/cloud-init-output.log`
  - `/var/log/slurmd.log`
- Si un nœud de calcul est lancé, vérifiez d'abord `/var/log/cloud-init-output.log`, qui doit contenir les journaux de configuration similaires à ceux du nœud principal. `/var/log/chef-client.log` La plupart des erreurs survenant lors de l'installation doivent contenir des messages d'erreur dans le `/var/log/cloud-init-output.log` journal. Si des scripts de pré-installation ou de post-installation sont spécifiés dans la configuration du cluster, vérifiez qu'ils ont été exécutés correctement.
- Si vous utilisez une AMI personnalisée avec modification de la Slurm configuration, il se peut qu'une erreur Slurm liée empêche le nœud de calcul de rejoindre le cluster. Pour les erreurs liées au planificateur, consultez le `/var/log/slurmd.log` journal.

### Nœuds de calcul dynamiques :

- Recherchez dans le ResumeProgram journal (`/var/log/parallelcluster/slurm_resume.log`) le nom de votre nœud de calcul pour voir s'il ResumeProgram a déjà été appelé avec le nœud. (Si vous ResumeProgram n'avez jamais été appelé, vous pouvez consulter le `slurmctld` journal (`/var/log/slurmctld.log`) pour déterminer si vous avez Slurm déjà essayé ResumeProgram d'appeler le nœud).
- Notez que des autorisations incorrectes pour ResumeProgram peuvent ResumeProgram entraîner un échec silencieux. Si vous utilisez une AMI personnalisée dont la ResumeProgram configuration a été modifiée, vérifiez qu'elle ResumeProgram appartient à `slurmutilisateur` et qu'elle dispose de l'autorisation `744 (rwxr--r--)`.
- En cas ResumeProgram d'appel, vérifiez si une instance est lancée pour le nœud. Si aucune instance n'a été lancée, un message d'erreur décrivant l'échec du lancement peut s'afficher.
- Si l'instance est lancée, un problème est peut-être survenu lors du processus de configuration. Vous devriez voir l'adresse IP privée et l'ID d'instance correspondants dans le ResumeProgram journal. De plus, vous pouvez consulter les journaux de configuration correspondants pour l'instance en question. Pour plus d'informations sur le dépannage d'une erreur de configuration avec un nœud de calcul, consultez la section suivante.

### Nœuds de calcul statiques :



- Consultez le journal `clustermgtd (/var/log/parallelcluster/clustermgtd)` pour voir si des instances ont été lancées pour le nœud. S'ils n'ont pas été lancés, un message d'erreur clair devrait s'afficher détaillant l'échec du lancement.
- Si l'instance est lancée, il y a un problème pendant le processus de configuration. Vous devriez voir l'adresse IP privée et l'ID d'instance correspondants dans le `ResumeProgram` journal. De plus, vous pouvez consulter les journaux de configuration correspondants pour l'instance en question.

Nœuds de calcul soutenus par des instances Spot :

- Si c'est la première fois que vous utilisez des instances Spot et que la tâche est toujours au format PDF (état en attente), vérifiez le `/var/log/parallelcluster/slurm_resume.log` fichier. Vous trouverez probablement une erreur comme celle-ci :

```
2022-05-20 13:06:24,796 - [slurm_plugin.common:add_instances_for_nodes] - ERROR -
Encountered exception when launching instances for nodes (x1) ['spot-dy-t2micro-2']:
An error occurred (AuthFailure.ServiceLinkedRoleCreationNotPermitted) when calling
the RunInstances operation: The provided credentials do not have permission to
create the service-linked role for Amazon EC2 Spot Instances.
```

Lorsque vous utilisez des instances Spot, un rôle `AWSServiceRoleForEC2Spot` lié au service doit exister dans votre compte. Pour créer ce rôle dans votre compte à l'aide de AWS CLI, exécutez la commande suivante :

```
$ aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Pour plus d'informations, consultez le guide [Utilisation de instances Spot](#) de l' AWS ParallelCluster utilisateur et le [rôle lié au service pour les demandes d'instance Spot](#) dans le guide de l'utilisateur Amazon EC2.

## Résolution des problèmes de remplacement et de terminaison inattendus de nœuds

Cette section continue à explorer comment résoudre les problèmes liés aux nœuds, en particulier lorsqu'un nœud est remplacé ou arrêté de manière inattendue.

- Journaux applicables :
  - `/var/log/parallelcluster/clustermgtd`(nœud principal)

- `/var/log/slurmctld.log`(nœud principal)
- `/var/log/parallelcluster/computemgtd`(nœud de calcul)

### Nœuds remplacés ou interrompus de façon inattendue

- Consultez le `clustermgtd` journal (`/var/log/parallelcluster/clustermgtd`) pour voir si un nœud `clustermgtd` a été remplacé ou résilié. Notez que cela `clustermgtd` gère toutes les actions de maintenance normales du nœud.
- En cas de `clustermgtd` remplacement ou de résiliation du nœud, un message doit s'afficher expliquant pourquoi cette action a été entreprise sur le nœud. Si la raison est liée au planificateur (par exemple, parce que le nœud est `dedansDOWN`), consultez le `slurmctld` journal pour plus d'informations. Si la raison est liée à Amazon EC2, il doit y avoir un message informatif détaillant le problème lié à Amazon EC2 qui a nécessité le remplacement.
- Si cela `clustermgtd` n'a pas mis fin au nœud, vérifiez d'abord s'il s'agit d'une résiliation prévue par Amazon EC2, plus précisément d'une résiliation ponctuelle. `computemgtd`, exécuté sur un nœud de calcul, peut également mettre fin à un nœud s'il `clustermgtd` est déterminé qu'il ne fonctionne pas correctement. Vérifiez `computemgtd log` (`/var/log/parallelcluster/computemgtd`) pour voir si le `computemgtd` nœud a été arrêté.

### Les nœuds ont échoué

- Consultez `slurmctld log` (`/var/log/slurmctld.log`) pour savoir pourquoi une tâche ou un nœud a échoué. Notez que les tâches sont automatiquement mises en file d'attente en cas de défaillance d'un nœud.
- Si vous `slurm_resume` signalez que le nœud est lancé et qu'il `clustermgtd` indique au bout de quelques minutes qu'il n'existe aucune instance correspondante dans Amazon EC2 pour ce nœud, le nœud risque de tomber en panne lors de la configuration. Pour récupérer le journal à partir d'un `compute` (`/var/log/cloud-init-output.log`), procédez comme suit :
  - Soumettez une offre d'emploi Slurm pour créer un nouveau nœud.
  - Attendez que le nœud de calcul démarre.
  - Modifiez le comportement d'arrêt initié par l'instance afin qu'un nœud de calcul défaillant soit arrêté plutôt que résilié.

```
$ aws ec2 modify-instance-attribute \
 --instance-id i-1234567890abcdef0 \
 --stop-action=terminate
```

```
--instance-initiated-shutdown-behavior "{\"Value\": \"stop\"}"
```

- Activer la protection de la résiliation.

```
$ aws ec2 modify-instance-attribute \
 --instance-id i-1234567890abcdef0 \
 --disable-api-termination
```

- Marquez le nœud pour qu'il soit facilement identifiable.

```
$ aws ec2 create-tags \
 --resources i-1234567890abcdef0 \
 --tags Key=Name,Value=QUARANTINED-Compute
```

- Détachez le nœud du cluster en modifiant le `parallelcluster:cluster-name` tag.

```
$ aws ec2 create-tags \
 --resources i-1234567890abcdef0 \
 --tags Key=parallelcluster:clustername,Value=QUARANTINED-ClusterName
```

- Récupérez la sortie de console depuis le nœud à l'aide de cette commande.

```
$ aws ec2 get-console-output --instance-id i-1234567890abcdef0 --output text
```

## Remplacement, arrêt ou mise hors tension des instances et des nœuds problématiques

- Journaux applicables :
  - `/var/log/parallelcluster/clustermgtd`(nœud principal)
  - `/var/log/parallelcluster/slurm_suspend.log`(nœud principal)
- Dans la plupart des cas, `clustermgtd` gère toutes les actions de fermeture d'instance attendues. Consultez le `clustermgtd` journal pour savoir pourquoi il n'a pas réussi à remplacer ou à mettre fin à un nœud.
- En cas d'échec d'un nœud dynamique [SlurmSettingsPropriétés](#), consultez le `SuspendProgram` journal pour voir si `SuspendProgram` a été appelé `slurmctld` avec le nœud spécifique comme argument. Notez qu'`SuspendProgram` aucune action n'est réellement exécutée. Au contraire, il ne se connecte que lorsqu'il est appelé. La résiliation et la `NodeAdd` réinitialisation de toutes les

instances sont effectuées par `clusterimgtd`. `Slurmremot SuspendTimeout` automatiquement les nœuds dans un `POWER_SAVING` état ultérieur.

- Si les nœuds de calcul échouent continuellement en raison d'échecs d'amorçage, vérifiez s'ils sont lancés avec [Slurm mode protégé par cluster](#) cette option activée. Si le mode protégé n'est pas activé, modifiez les paramètres du mode protégé pour activer le mode protégé. Résolvez et corrigez le script bootstrap.

## Inactive État de la file d'attente (partition)

Si vous exécutez `sinfo` et que le résultat affiche des files d'attente dont le `AVAIL` statut est égal à `inact`, il est possible que votre cluster soit [Slurm mode protégé par cluster](#) activé et que la file d'attente ait été définie sur `INACTIVE` cet état pendant une période prédéfinie.

## Résolution d'autres problèmes connus liés aux nœuds et aux tâches

Un autre type de problème connu est celui qui AWS ParallelCluster peut ne pas permettre d'allouer des tâches ou de prendre des décisions de dimensionnement. Dans ce type de problème, AWS ParallelCluster seules les ressources sont lancées, résiliées ou maintenues conformément aux Slurm instructions. Pour ces problèmes, consultez le `slurmctld` journal pour les résoudre.

## Problèmes liés aux groupes de placement et au lancement d'instances

Pour obtenir le plus faible temps de latence entre nœuds, utilisez un groupe de placement. Un groupe de placement garantit que vos instances se trouvent sur le même backbone réseau. S'il n'y a pas suffisamment d'instances disponibles lorsqu'une demande est faite, une `InsufficientInstanceCapacity` erreur est renvoyée. Pour réduire le risque de recevoir cette erreur lors de l'utilisation de groupes de placement de clusters, définissez le `Enabled` paramètre [SlurmQueuesNetworkingPlacementGroup](#) sur `false`.

Pour un contrôle accru de l'accès aux capacités, envisagez de [lancer des instances avec ODCR \(On-Demand Capacity Reservations\)](#).

Pour plus d'informations, consultez les sections [Résolution des problèmes de lancement d'instance](#) et [Rôles et limites des groupes de placement](#) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

## Répertoires qui ne peuvent pas être remplacés

Les répertoires suivants sont partagés entre les nœuds et ne peuvent pas être remplacés.

- `/home`- Cela inclut le dossier personnel par défaut de l'utilisateur (`/home/ec2_users` sur Amazon Linux RedHat, `/home/centos` sur CentOS et `/home/ubuntu` sur Ubuntu).
- `/opt/intel`- Cela inclut Intel MPI, Intel Parallel Studio et les fichiers associés.
- `/opt/slurm`- Cela inclut le gestionnaire Slurm de charge de travail et les fichiers associés. (Conditionnel, seulement si `Scheduler: slurm`.)

## Résolution des problèmes dans NICE DCV

Rubriques

- [Logos pour NICE DCV](#)
- [Problèmes liés à Ubuntu NICE DCV](#)

### Logos pour NICE DCV

Les journaux de NICE DCV sont écrits dans des fichiers du `/var/log/dcv/` répertoire. L'examen de ces journaux peut aider à résoudre les problèmes.

Le type d'instance doit disposer d'au moins 1,7 Gibioctet (GiB) de RAM pour exécuter NICE DCV. Les types d'instances nano et micro ne disposent pas de suffisamment de mémoire pour exécuter NICE DCV.

AWS ParallelCluster crée des flux de journaux NICE DCV dans des groupes de journaux. Vous pouvez consulter ces journaux dans les tableaux de bord personnalisés ou les groupes de journaux de la CloudWatch console. Pour plus d'informations, consultez [Intégration avec Amazon CloudWatch Logs avec Amazon Logs](#) et [Tableau de CloudWatch bord Amazon](#).

### Problèmes liés à Ubuntu NICE DCV

Lorsque vous exécutez le terminal Gnome sur une session NICE DCV sur Ubuntu, il se peut que vous n'ayez pas automatiquement accès à l'environnement AWS ParallelCluster utilisateur mis à disposition via le shell de connexion. L'environnement utilisateur fournit des modules d'environnement tels que `openmpi` ou `intelmpi`, ainsi que d'autres paramètres utilisateur.

Les paramètres par défaut de Gnome Terminal empêchent le shell de démarrer en tant que shell de connexion. Cela signifie que les profils shell ne sont pas générés automatiquement et que l'environnement AWS ParallelCluster utilisateur n'est pas chargé.

Pour obtenir correctement le profil shell et accéder à l'environnement AWS ParallelCluster utilisateur, effectuez l'une des opérations suivantes :

- Modifiez les paramètres par défaut du terminal :
  1. Choisissez le menu Edition dans le terminal Gnome.
  2. Sélectionnez Préférences, puis Profils.
  3. Choisissez Command et sélectionnez Exécuter la commande en tant que shell de connexion.
  4. Ouvrez un nouveau terminal.
- Utilisez la ligne de commande pour rechercher les profils disponibles :

```
$ source /etc/profile && source $HOME/.bashrc
```

## Résolution des problèmes dans les clusters avec AWS Batch intégration

Cette section concerne les clusters avec intégration d'un AWS Batch planificateur.

### Rubriques

- [Problèmes liés au nœud principal](#)
- [Problèmes de calcul](#)
- [Échecs au travail](#)
- [Erreur d'expiration du délai de connexion en cas d'URL du point de terminaison](#)

### Problèmes liés au nœud principal

Vous pouvez résoudre les problèmes de configuration du nœud principal de la même manière qu'un Slurm cluster (sauf pour les journaux Slurm spécifiques). Pour de plus amples informations sur ces problèmes, veuillez consulter [Nœud principal](#).

## Problèmes de calcul

AWS Batch gère les aspects de dimensionnement et de calcul de vos services. Si vous rencontrez des problèmes liés au calcul, consultez la documentation de AWS Batch [dépannage](#) pour obtenir de l'aide.

### Échecs au travail

Si une tâche échoue, vous pouvez exécuter la [awsbout](#) commande pour récupérer le résultat de la tâche. Vous pouvez également exécuter la [awsbstat](#) commande pour obtenir un lien vers les journaux des tâches stockés par Amazon CloudWatch.

### Erreur d'expiration du délai de connexion en cas d'URL du point de terminaison

Si les tâches parallèles sur plusieurs nœuds échouent avec une erreur : `Connect timeout on endpoint URL`

- Dans le journal `awsbout` de sortie, vérifiez que la tâche est parallèle à plusieurs nœuds par rapport à la sortie : `Detected 3/3 compute nodes. Waiting for all compute nodes to start.`
- Vérifiez si le sous-réseau des nœuds de calcul est public.

Les tâches parallèles à nœuds multiples ne prennent pas en charge l'utilisation de sous-réseaux publics lors de leur utilisation AWS Batch dans AWS ParallelCluster. Utilisez un sous-réseau privé pour vos nœuds de calcul et vos tâches. Pour plus d'informations, consultez la section [Considérations relatives à l'environnement informatique](#) dans le Guide de AWS Batch l'utilisateur. Pour configurer un sous-réseau privé pour vos nœuds de calcul, consultez [AWS ParallelCluster avec AWS Batch planificateur](#).

## Résolution des problèmes d'intégration multi-utilisateurs avec Active Directory

Cette section concerne les clusters intégrés à un Active Directory.

Si la fonctionnalité d'intégration d'Active Directory ne fonctionne pas comme prévu, les journaux SSSD peuvent fournir des informations de diagnostic utiles. Ces journaux se trouvent `/var/`

log/sssdd sur les nœuds du cluster. Par défaut, ils sont également stockés dans le groupe de CloudWatch journaux Amazon d'un cluster.

## Rubriques

- [Résolution des problèmes spécifiques à Active Directory](#)
- [Activer le mode de débogage](#)
- [Comment passer de LDAPS à LDAP](#)
- [Comment désactiver la vérification du certificat du serveur LDAPS](#)
- [Comment se connecter avec une clé SSH plutôt qu'un mot de passe](#)
- [Comment réinitialiser le mot de passe d'un utilisateur et les mots de passe expirés](#)
- [Comment vérifier le domaine joint](#)
- [Comment résoudre les problèmes liés aux certificats](#)
- [Comment vérifier que l'intégration avec Active Directory fonctionne](#)
- [Comment résoudre les problèmes de connexion aux nœuds de calcul](#)
- [Problèmes connus liés aux tâches SimCenter StarCCM+ dans un environnement multi-utilisateurs](#)
- [Problèmes connus liés à la résolution des noms d'utilisateur](#)
- [Comment résoudre les problèmes de création du répertoire de base](#)

## Résolution des problèmes spécifiques à Active Directory

Cette section concerne le dépannage spécifique à un type d'Active Directory.

### Simple AD

- La `DomainReadOnlyUser` valeur doit correspondre à la recherche de base d'annuaires Simple AD pour les utilisateurs :

```
cn=ReadOnlyUser,cn=Users,dc=corp,dc=example,dc=com
```

Remarque `cn` pour `Users`.

- L'utilisateur administrateur par défaut est `Administrator`.
- `Ldapsearch` nécessite le nom NetBIOS avant le nom d'utilisateur.

`Ldapsearch` la syntaxe doit être la suivante :



```
$ ldapsearch -x -D "corp\\Administrator" -w "Password" -H ldap://192.0.2.103 \
-b "cn=Users,dc=corp,dc=example,dc=com"
```

## AWS Managed Microsoft AD

- La `DomainReadOnlyUser` valeur doit correspondre à la recherche dans la base de AWS Managed Microsoft AD répertoires pour les utilisateurs :

```
cn=ReadOnlyUser,ou=Users,ou=CORP,dc=corp,dc=example,dc=com
```

- L'utilisateur administrateur par défaut est `Admin`.
- `Ldapsearch` la syntaxe doit être la suivante :

```
$ ldapsearch -x -D "Admin" -w "Password" -H ldap://192.0.2.103 \
-b "ou=Users,ou=CORP,dc=corp,dc=example,dc=com"
```

## Activer le mode de débogage

Les journaux de débogage de SSSD peuvent être utiles pour résoudre les problèmes. Pour activer le mode de débogage, vous devez mettre à jour le cluster avec les modifications suivantes apportées à la configuration du cluster :

```
DirectoryService:
 AdditionalSssdConfigs:
 debug_level: "0x1ff"
```

## Comment passer de LDAPS à LDAP

Il est déconseillé de passer du protocole LDAPS (LDAP avec TLS/SSL) au protocole LDAP, car le protocole LDAP à lui seul ne fournit aucun chiffrement. Néanmoins, cela peut être utile à des fins de test et de débogage.

Vous pouvez rétablir la configuration précédente du cluster en le mettant à jour avec la définition de configuration précédente.

Pour passer de LDAPS à LDAP, vous devez mettre à jour le cluster en apportant les modifications suivantes à la configuration du cluster :

```
DirectoryService:
 LdapTlsReqCert: never
 AdditionalSssdConfigs:
 ldap_auth_disable_tls_never_use_in_production: True
```

## Comment désactiver la vérification du certificat du serveur LDAPS

Il peut être utile de désactiver temporairement la vérification du certificat du serveur LDAPS sur le nœud principal, à des fins de test ou de dépannage.

Vous pouvez rétablir la configuration précédente du cluster en le mettant à jour avec la définition de configuration précédente.

Pour désactiver la vérification du certificat du serveur LDAPS, vous devez mettre à jour le cluster en apportant les modifications suivantes à la configuration du cluster :

```
DirectoryService:
 LdapTlsReqCert: never
```

## Comment se connecter avec une clé SSH plutôt qu'un mot de passe

La clé SSH est créée lorsque vous vous connectez pour la première fois avec un mot de passe. /home/\$user/.ssh/id\_rsa Pour vous connecter avec la clé SSH, vous devez vous connecter avec votre mot de passe, copier la clé SSH localement, puis l'utiliser pour utiliser le protocole SSH sans mot de passe, comme d'habitude :

```
$ ssh -i $LOCAL_PATH_TO_SSH_KEY $username@$head_node_ip
```

## Comment réinitialiser le mot de passe d'un utilisateur et les mots de passe expirés

Si un utilisateur perd l'accès à un cluster, son [AWS Managed Microsoft AD mot de passe a peut-être expiré](#).

Pour réinitialiser le mot de passe, exécutez la commande suivante avec un utilisateur et un rôle autorisés à écrire sur le répertoire :

```
$ aws ds reset-user-password \
```

```
--directory-id "d-abcdef01234567890" \
--user-name "USER_NAME" \
--new-password "NEW_PASSWORD" \
--region "region-id"
```

Si vous réinitialisez le mot de passe du [DirectoryService/DomainReadOnlyUser](#):

1. Assurez-vous de mettre à jour le [DirectoryService/PasswordSecretArn](#) secret avec le nouveau mot de passe.
2. Mettez à jour le cluster pour la nouvelle valeur secrète :
  - a. Arrêtez le parc informatique à l'aide de la `pcluster update-compute-fleet` commande.
  - b. Exécutez la commande suivante depuis le nœud principal du cluster.

```
$ sudo /opt/parallelcluster/scripts/directory_service/
update_directory_service_password.sh
```

Après la réinitialisation du mot de passe et la mise à jour du cluster, l'accès au cluster de l'utilisateur doit être rétabli.

Pour plus d'informations, voir [Réinitialiser un mot de passe utilisateur](#) dans le Guide d'AWS Directory Service administration.

## Comment vérifier le domaine joint

La commande suivante doit être exécutée à partir d'une instance jointe au domaine, et non du nœud principal.

```
$ realm list corp.example.com \
type: kerberos \
realm-name: CORP.EXAMPLE.COM \
domain-name: corp.example.com \
configured: kerberos-member \
server-software: active-directory \
client-software: sssd \
required-package: oddjob \
required-package: oddjob-mkhomedir \
required-package: sssd \
required-package: adcli \
required-package: samba-common-tools \

```

```
login-formats: %U \
login-policy: allow-realm-logins
```

## Comment résoudre les problèmes liés aux certificats

Lorsque la communication LDAPS ne fonctionne pas, cela peut être dû à des erreurs dans la communication TLS, qui à leur tour peuvent être dues à des problèmes liés aux certificats.

Remarques concernant les certificats :

- Le certificat spécifié dans la configuration du cluster `LdapTlsCaCert` doit être un ensemble de certificats PEM contenant les certificats de l'ensemble de la chaîne de certificats d'autorité (CA) qui a émis les certificats pour les contrôleurs de domaine.
- Un bundle de certificats PEM est un fichier constitué de la concaténation de certificats PEM.
- Un certificat au format PEM (généralement utilisé sous Linux) est équivalent à un certificat au format Base64 DER (généralement exporté par Windows).
- Si le certificat pour les contrôleurs de domaine est émis par une autorité de certification subordonnée, le bundle de certificats doit contenir le certificat de l'autorité de certification subordonnée et celui de l'autorité de certification racine.

Étapes de vérification du dépannage :

Les étapes de vérification suivantes supposent que les commandes sont exécutées depuis le nœud principal du cluster et que le contrôleur de domaine est accessible à l'adresse `SERVER:PORT`.

Pour résoudre un problème lié aux certificats, suivez les étapes de vérification suivantes :

Étapes de vérification :

1. Vérifiez la connexion aux contrôleurs de domaine Active Directory :

Vérifiez que vous pouvez vous connecter à un contrôleur de domaine. Si cette étape réussit, la connexion SSL au contrôleur de domaine aboutit et le certificat est vérifié. Votre problème n'est pas lié aux certificats.

Si cette étape échoue, passez à la prochaine vérification.

```
$ openssl s_client -connect SERVER:PORT -CAfile PATH_TO_CA_BUNDLE_CERTIFICATE
```

## 2. Vérifiez la vérification du certificat :

Vérifiez que le bundle de certificats CA local peut valider le certificat fourni par le contrôleur de domaine. Si cette étape aboutit, votre problème n'est pas lié aux certificats, mais à d'autres problèmes de réseau.

Si cette étape échoue, passez à la prochaine vérification.

```
$ openssl verify -verbose -
CAfile PATH_TO_CA_BUNDLE_CERTIFICATE PATH_TO_A_SERVER_CERTIFICATE
```

## 3. Vérifiez le certificat fourni par les contrôleurs de domaine Active Directory :

Vérifiez que le contenu du certificat fourni par les contrôleurs de domaine est conforme aux attentes. Si cette étape aboutit, vous rencontrez probablement des problèmes avec le certificat CA utilisé pour vérifier les contrôleurs. Passez à l'étape de résolution des problèmes suivante.

Si cette étape échoue, vous devez corriger le certificat émis pour les contrôleurs de domaine et réexécuter les étapes de dépannage.

```
$ openssl s_client -connect SERVER:PORT -showcerts
```

## 4. Vérifiez le contenu d'un certificat :

Vérifiez que le contenu du certificat fourni par les contrôleurs de domaine est conforme aux attentes. Si cette étape aboutit, vous avez probablement des problèmes avec le certificat CA utilisé pour vérifier le contrôleur. Passez à l'étape de résolution des problèmes suivante.

Si cette étape échoue, vous devez corriger le certificat émis pour les contrôleurs de domaine et réexécuter les étapes de résolution des problèmes.

```
$ openssl s_client -connect SERVER:PORT -showcerts
```

## 5. Vérifiez le contenu du bundle de certificats CA local :

Vérifiez que le contenu du bundle de certificats CA local utilisé pour valider le certificat des contrôleurs de domaine est conforme aux attentes. Si cette étape aboutit, vous rencontrez probablement des problèmes avec le certificat fourni par les contrôleurs de domaine.

Si cette étape échoue, vous devez corriger le bundle de certificats CA émis pour les contrôleurs de domaine et réexécuter les étapes de dépannage.

```
$ openssl x509 -in PATH_TO_A_CERTIFICATE -text
```

## Comment vérifier que l'intégration avec Active Directory fonctionne

Si les deux vérifications suivantes aboutissent, l'intégration avec Active Directory fonctionne.

Contrôles :

1. Vous pouvez découvrir les utilisateurs définis dans l'annuaire :

Depuis le nœud principal du cluster, en tant que `ec2-user` :

```
$ getent passwd $ANY_AD_USER
```

2. Vous pouvez vous connecter par SSH au nœud principal en fournissant le mot de passe utilisateur :

```
$ ssh $ANY_AD_USER@$HEAD_NODE_IP
```

Si le premier contrôle échoue, nous nous attendons à ce que le contrôle deux échoue également.

Contrôles de résolution des problèmes supplémentaires :

- Vérifiez que l'utilisateur existe dans l'annuaire.
- Activez la [journalisation du débogage](#).
- Envisagez de désactiver temporairement le chiffrement en [passant du protocole LDAPS au protocole LDAP afin d'éliminer les problèmes liés au protocole LDAPS](#).

## Comment résoudre les problèmes de connexion aux nœuds de calcul

Cette section concerne la connexion aux nœuds de calcul des clusters intégrés à Active Directory.

Avec AWS ParallelCluster, les connexions par mot de passe aux nœuds de calcul du cluster sont désactivées par conception.

Tous les utilisateurs doivent utiliser leur propre clé SSH pour se connecter aux nœuds de calcul.

Les utilisateurs peuvent récupérer leur clé SSH dans le nœud principal après la première authentification (par exemple, connexion), si cette option [GenerateSshKeysForUsers](#) est activée dans la configuration du cluster.

Lorsque les utilisateurs s'authentifient sur le nœud principal pour la première fois, ils peuvent récupérer les clés SSH qui sont automatiquement générées pour eux en tant qu'utilisateurs de l'annuaire. Des répertoires personnels pour l'utilisateur sont également créés. Cela peut également se produire la première fois qu'un sudo-utilisateur passe à un utilisateur du nœud principal.

Si un utilisateur ne s'est pas connecté au nœud principal, les clés SSH ne sont pas générées et l'utilisateur ne pourra pas se connecter aux nœuds de calcul.

## Problèmes connus liés aux tâches SimCenter StarCCM+ dans un environnement multi-utilisateurs

Cette section concerne les tâches lancées dans un environnement multi-utilisateurs par le logiciel de dynamique des fluides Simcenter StarCCM+ de Siemens.

Si vous exécutez des tâches StarCCM+ v16 configurées pour utiliser l'IntelMPI intégré, les processus MPI sont initialisés par défaut via SSH.

En raison d'un [Slurmbogue](#) connu qui entraîne une mauvaise résolution du nom d'utilisateur, les tâches peuvent échouer avec une erreur telle que `error setting up the bootstrap proxies`. Ce bogue n'affecte que AWS ParallelCluster les versions 3.1.1 et 3.1.2.

Pour éviter que cela ne se produise, forcez IntelMPI à l'utiliser Slurm comme méthode d'amorçage MPI. Exportez la variable d'environnement `I_MPI_HYDRA_BOOTSTRAP=slurm` dans le script de tâche qui lance StarCCM+, comme décrit dans la documentation officielle d'[IntelMPI](#).

## Problèmes connus liés à la résolution des noms d'utilisateur

Cette section est pertinente pour récupérer les noms d'utilisateur dans les tâches.

En raison d'un [bugue connu dans Slurm](#), le nom d'utilisateur récupéré dans un processus de travail peut être le nobody cas si vous exécutez un travail sans `srun`. Ce bogue n'affecte que AWS ParallelCluster les versions 3.1.1 et 3.1.2.

Par exemple, si vous exécutez la commande `sbatch --wrap 'srun id'` en tant qu'utilisateur de l'annuaire, le nom d'utilisateur correct est renvoyé. Toutefois, si vous l'exécutez `sbatch --wrap 'id'` en tant qu'utilisateur d'annuaire, nobody il peut être renvoyé sous forme de nom d'utilisateur.

Vous pouvez utiliser les solutions de contournement suivantes.

1. Lancez votre travail avec 'srun' au lieu de 'sbatch', si possible.
2. Activez l'énumération SSSD en définissant la [AdditionalSssdConfigs](#) configuration du cluster comme suit.

```
AdditionalSssdConfigs:
 enumerate: true
```

## Comment résoudre les problèmes de création du répertoire de base

Cette section concerne les problèmes liés à la création du répertoire de base.

Si vous voyez des erreurs comme celle illustrée dans l'exemple suivant, aucun répertoire personnel n'a été créé pour vous lorsque vous vous êtes connecté pour la première fois au nœud principal. Ou bien, aucun répertoire personnel n'a été créé pour vous lorsque vous êtes passé pour la première fois d'un sudoer à un utilisateur Active Directory dans le nœud principal.

```
$ ssh AD_USER@$HEAD_NODE_IP
/opt/parallelcluster/scripts/generate_ssh_key.sh failed: exit code 1

 __| __|_)
 _| (/ Amazon Linux 2 AMI
 __|___|___|

https://aws.amazon.com/amazon-linux-2/
Could not chdir to home directory /home/PclusterUser85: No such file or directory
```

L'échec de création du répertoire de base peut être dû aux oddjob-mkhomedir packages oddjob et installés dans le nœud principal du cluster.

Sans répertoire de base et clé SSH, l'utilisateur ne peut pas envoyer de tâches ou de SSH aux nœuds du cluster.

Si vous avez besoin des oddjob packages dans votre système, vérifiez que le oddjobd service est en cours d'exécution et actualisez les fichiers de configuration PAM pour vous assurer que le répertoire de base a été créé. Pour ce faire, exécutez les commandes dans le nœud principal comme indiqué dans l'exemple suivant.

```
sudo systemctl start oddjobd
```



```
sudo authconfig --enablemkhomedir --updateall
```

Si vous n'avez pas besoin des `oddjob` packages dans votre système, désinstallez-les et actualisez les fichiers de configuration PAM pour vous assurer que le répertoire de base a été créé. Pour ce faire, exécutez les commandes dans le nœud principal comme indiqué dans l'exemple suivant.

```
sudo yum remove -y oddjob oddjob-mkhomedir
sudo authconfig --enablemkhomedir --updateall
```

## Résolution des problèmes liés aux AMI personnalisées

Lorsque vous utilisez une AMI personnalisée, les avertissements suivants peuvent s'afficher :

```
"validationMessages": [
 {
 "level": "WARNING",
 "type": "CustomAmiTagValidator",
 "message": "The custom AMI may not have been created by pcluster. You can ignore
this warning if the AMI is shared or copied from another pcluster AMI. If the
AMI is indeed not created by pcluster, cluster creation will fail. If the cluster
creation fails, please go to https://docs.aws.amazon.com/parallelcluster/latest/ug/troubleshooting.html#troubleshooting-stack-creation-failures for troubleshooting."
 },
 {
 "level": "WARNING",
 "type": "AmiOsCompatibleValidator",
 "message": "Could not check node AMI ami-0000012345 OS and cluster OS alinux2
compatibility, please make sure they are compatible before cluster creation and update
operations."
 }
]
```

Si vous êtes sûr que l'AMI utilisée est correcte, vous pouvez ignorer ces avertissements.

Si vous ne souhaitez pas voir ces avertissements à l'avenir, balisez l'AMI personnalisée avec les balises suivantes, où se *my-ostrouve* l'une des balises `alinux2`, `ubuntu2204`, `ubuntu2004centos7`, ou `rhel8` et « *3.7.0* » correspond à la `pcluster` version utilisée :

```
$ aws ec2 create-tags \
 --resources ami-yourcustomAmi \
```

```
--tags Key="parallelcluster:version",Value="3.7.0"
Key="parallelcluster:os",Value="my-os"
```

## Résolution d'un délai d'expiration de mise à jour du cluster en cas **cf-n-hup** d'inexécution

L'**cf-n-hup** assistant est un démon qui détecte les modifications apportées aux métadonnées des ressources et exécute les actions spécifiées par l'utilisateur lorsqu'une modification est détectée. C'est ainsi que vous pouvez effectuer des mises à jour de configuration sur vos instances Amazon EC2 en cours d'exécution par le biais de l'action UpdateStack API.

Actuellement, le **cf-n-hup** daemon est lancé par **supervisord**. Mais après le lancement, le **cf-n-hup** processus est déconnecté du **supervisord** contrôle. Si le **cf-n-hup** démon est tué par un acteur externe, il ne redémarre pas automatiquement. S'il **cf-n-hup** n'est pas en cours d'exécution, lors d'une mise à jour du cluster, la CloudFormation pile lance le processus de mise à jour comme prévu, mais la procédure de mise à jour n'est pas activée sur le nœud principal et la pile finit par atteindre un délai d'expiration. Dans les journaux du cluster `/var/log/chef-client`, vous pouvez voir que la recette de mise à jour n'est jamais invoquée.

Vérifiez et redémarrez **cf-n-hup** en cas de panne

1. Sur le nœud principal, vérifiez s'il **cf-n-hup** est en cours d'exécution :

```
$ ps aux | grep cf-n-hup
```

2. Vérifiez **cf-n-hup** le journal `/var/log/cf-n-hup.log` et `/var/log/supervisord.log` le nœud principal.
3. S'il **cf-n-hup** n'est pas en cours d'exécution, essayez de le redémarrer en exécutant :

```
$ sudo /opt/parallelcluster/pyenv/versions/cookbook_virtualenv/bin/supervisorctl
start cf-n-hup
```

## Dépannage du réseau

### Problèmes liés au cluster dans un seul sous-réseau public

Vérifiez le `cloud-init-output.log` depuis l'un des nœuds de calcul. Si vous trouvez quelque chose comme ce qui suit qui indique que le nœud est bloqué lors de l'initialisation Slurm, cela est probablement dû à l'absence d'un point de terminaison DynamoDB VPC. Ajoutez le point de terminaison DynamoDB. Pour plus d'informations, consultez [AWS ParallelCluster dans un seul sous-réseau sans accès à Internet](#).

```
ruby_block[retrieve compute node info] action run[2022-03-11T17:47:11+00:00] INFO:
 Processing ruby_block[retrieve compute node info] action run (aws-parallelcluster-
 slurm::init line 31)
```

### La mise à jour du cluster a échoué lors d'**onNodeUpdated** d'une action personnalisée

Lorsqu'un [OnNodeUpdated](#) script [HeadNode/CustomActions](#) échoue, la mise à jour échoue et le script n'est pas exécuté au moment de la restauration. Il est de votre responsabilité d'effectuer manuellement les nettoyages nécessaires une fois la restauration terminée. Par exemple, si le `OnNodeUpdated` script modifie le statut d'un champ dans un fichier de configuration (par exemple, de `true` à `false`) puis échoue, vous devez restaurer manuellement la valeur de ce champ à son état antérieur à la mise à jour (par exemple, `false` à `true`). Pour plus d'informations, consultez [Actions de bootstrap personnalisées](#).

### Voir les erreurs liées à la Slurm configuration personnalisée

À partir de AWS ParallelCluster la version 3.6.0, vous ne pouvez plus cibler un `prolog` ou plusieurs `epilog` scripts en les incluant dans une Slurm configuration personnalisée. Dans les AWS ParallelCluster versions 3.6.0 et ultérieures, vous devez localiser les scripts personnalisés `prolog` et les `epilog` scripts dans les `Epilog` dossiers `Prolog` et correspondants. Ces dossiers sont configurés par défaut pour pointer vers :

- `Prolog` pointe vers `/opt/slurm/etc/scripts/prolog.d/`.
- `Epilog` pointe vers `/opt/slurm/etc/scripts/epilog.d/`.

Nous vous recommandons de conserver le script `90_plcluster_health_check_manager` prolog et le script `90_pcluster_noop` epilog en place.

Slurm exécute les scripts dans l'ordre alphabétique inverse. Le dossier Prolog et doit contenir au moins un fichier. Pour plus d'informations, consultez [Slurm prolog et epilog](#) et [Slurm personnalisation de la configuration](#).

## Alarmes de cluster

La surveillance de l'état du cluster est essentielle pour garantir des performances optimales. AWS ParallelCluster vous permet de surveiller plusieurs alarmes CloudWatch basées sur le nœud principal du cluster.

Cette section fournit des détails sur chaque type d'alarmes du cluster de nœuds principaux, y compris ses conventions de dénomination, les conditions spécifiques qui déclenchent les alarmes et les étapes de dépannage suggérées.

La convention de dénomination pour les alarmes de cluster est `CLUSTER_NAME-COMPONENT-METRIC`, par `mycluster-HeadNode-Cpu` ex.

- `CLUSTER_NAME-HeadNode`: indique l'état général du nœud principal. Il est rouge si au moins l'une des alarmes ci-dessous l'est.
- `CLUSTER_NAME-HeadNode-Health`: rouge s'il y a au moins un échec d'Amazon EC2 Health Check. En cas d'alarme, nous vous suggérons de consulter la section [Résoudre les problèmes des instances dont les vérifications d'état ont échoué](#).
- `CLUSTER_NAME-HeadNode-Cpu`: rouge si l'utilisation du processeur est supérieure à 90 %. En cas d'alarme, vérifiez les processus qui consomment le plus le processeur 

```
ps -aux --sort=-%cpu | head -n 10
```

.
- `CLUSTER_NAME-HeadNode-Mem`: rouge si l'utilisation de la mémoire est supérieure à 90 %. En cas d'alarme, vérifiez les processus qui consomment le plus de mémoire 

```
ps -aux --sort=-%mem | head -n 10
```

.
- `CLUSTER_NAME-HeadNode-Disk`: rouge si l'espace disque occupé est supérieur à 90 % sur le chemin `/`. En cas d'alarme, vérifiez les dossiers qui occupent la plus grande partie de l'espace 

```
du -h --max-depth=2 / 2> /dev/null | sort -hr
```

.

## Support supplémentaire

Pour une liste des problèmes connus, consultez la page principale du [GitHub Wiki](#) ou la page [des problèmes](#).

Pour les problèmes plus urgents, contactez AWS Support ou ouvrez un [nouveau GitHub numéro](#).

# AWS ParallelCluster politique de support

AWS ParallelCluster prend en charge plusieurs versions en même temps. Chaque AWS ParallelCluster version a une date de fin de vie planifiée (EOSL). Après la date EOSL, aucun support ou maintenance supplémentaire n'est fourni pour cette version.

AWS ParallelCluster utilise un schéma de `major.minor.patch` version. Les nouvelles fonctionnalités, les améliorations des performances, les mises à jour de sécurité et les corrections de bogues sont incluses dans les nouvelles versions mineures de la dernière version majeure. Les versions mineures sont rétrocompatibles au sein d'une version majeure. Pour les problèmes critiques, AWS fournit des correctifs par le biais de mises à jour, mais uniquement pour les dernières versions mineures des versions qui n'ont pas atteint l'EOSL. Si vous souhaitez utiliser les mises à jour d'une nouvelle version, vous devez effectuer une mise à niveau vers la nouvelle version mineure ou corrective.

| AWS ParallelCluster versions | Date de fin de vie prise en charge (EOSL) |
|------------------------------|-------------------------------------------|
| 3,0. <i>x</i>                | 31 mars 2023                              |
| 3.1. <i>x</i>                | 31 août 2023                              |
| 3,2. <i>x</i>                | 31/01/2024                                |
| 3,3. <i>x</i>                | 31/05/2024                                |
| 3.4. <i>x</i>                | 28/06/2024                                |
| 3,5. <i>x</i>                | 31/08/2024                                |
| 3,6. <i>x</i>                | 30/11/2024                                |
| 3,7. <i>x</i>                | 28/02/2025                                |
| 3,8. <i>x</i>                | 30/06/2025                                |
| 3.9. <i>x</i>                | 09/05/2025                                |
| 3,10. <i>x</i>               | 27/12/2025                                |

# Sécurité dans AWS ParallelCluster

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité 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 ParallelCluster, voir [AWS Services concernés par programme de conformitéAWS](#) .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le ou les AWS services spécifiques que vous utilisez. Vous êtes également responsable de plusieurs autres facteurs connexes, notamment la sensibilité de vos données, les exigences de votre entreprise et les lois et réglementations applicables.

Cette documentation décrit comment appliquer le modèle de responsabilité partagée lors de son utilisation AWS ParallelCluster. Les rubriques suivantes expliquent comment procéder à la configuration AWS ParallelCluster pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à les utiliser AWS ParallelCluster de manière à surveiller et à sécuriser vos AWS ressources.

## Rubriques

- [Informations de sécurité pour les services utilisés par AWS ParallelCluster](#)
- [Protection des données dans AWS ParallelCluster](#)
- [Identity and Access Management \(Gestion des identités et des accès\) pour AWS ParallelCluster](#)
- [Validation de la conformité pour AWS ParallelCluster](#)
- [Application d'une version minimale de TLS 1.2](#)

# Informations de sécurité pour les services utilisés par AWS ParallelCluster

- [Sécurité dans Amazon EC2](#)
- [Sécurité dans Amazon API Gateway](#)
- [Sécurité dans AWS Batch](#)
- [Sécurité dans AWS CloudFormation](#)
- [Sécurité sur Amazon CloudWatch](#)
- [Sécurité dans AWS CodeBuild](#)
- [Sécurité dans Amazon DynamoDB](#)
- [Sécurité dans Amazon ECR](#)
- [Sécurité dans Amazon ECS](#)
- [Sécurité dans Amazon EFS](#)
- [La sécurité dans FSx for Lustre](#)
- [Sécurité dans AWS Identity and Access Management \(IAM\)](#)
- [Sécurité dans EC2 Image Builder](#)
- [Sécurité dans AWS Lambda](#)
- [Sécurité dans Amazon Route 53](#)
- [Sécurité sur Amazon SNS](#)
- [Sécurité dans Amazon SQS \(pour la AWS ParallelCluster version 2.x.\)](#)
- [Sécurité dans Amazon S3](#)
- [Sécurité dans Amazon VPC](#)

## Protection des données dans AWS ParallelCluster

Le [modèle de responsabilité AWS partagée](#) de s'applique à la protection des données dans AWS ParallelCluster. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des services AWS que vous utilisez. Pour plus



d'informations sur la confidentialité des données, consultez la section [Confidentialité des données FAQ](#). Pour plus d'informations sur la protection des données en Europe, consultez le [modèle de responsabilité AWS partagée](#) et le billet de GDPR blog sur le blog sur la AWS sécurité.

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) pour chaque compte.
- Utilisez SSL/TLS pour communiquer avec les AWS ressources. Nous avons besoin de la TLS version 1.2 et recommandons la TLS version 1.3.
- Configuration API et journalisation de l'activité des utilisateurs avec AWS CloudTrail.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de FIPS 140 à 3 modules cryptographiques validés pour accéder AWS via une interface de ligne de commande ou un API, utilisez un point de terminaison. FIPS Pour plus d'informations sur les FIPS points de terminaison disponibles, voir [Federal Information Processing Standard \(FIPS\) 140-3](#).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Name (Nom). Cela inclut lorsque vous travaillez avec AWS ParallelCluster ou d'autres services AWS utilisateurs de la console API, AWS CLI, ou AWS SDKs. Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez un URL à un serveur externe, nous vous recommandons vivement de ne pas inclure d'informations d'identification dans le URL afin de valider votre demande auprès de ce serveur.

## Chiffrement des données

Une caractéristique clé de tout service sécurisé est que les informations sont chiffrées lorsqu'elles ne sont pas utilisées activement.

## Chiffrement au repos

AWS ParallelCluster ne stocke elle-même aucune donnée client autre que les informations d'identification dont elle a besoin pour interagir avec les AWS services au nom de l'utilisateur.

Pour les données relatives aux nœuds du cluster, les données peuvent être chiffrées au repos.

Pour les EBS volumes Amazon, le chiffrement est configuré à l'aide des KmsKeyId paramètres [EbsSettingsEbsSettings/Encrypted](#) et de la [EbsSettings](#) section. Pour plus d'informations, consultez [Amazon EBS Encryption](#) dans le guide de EC2 l'utilisateur Amazon.

Pour les EFS volumes Amazon, le chiffrement est configuré à l'aide des KmsKeyId paramètres [EfsSettingsEfsSettings/Encrypted](#) et de la [EfsSettings](#) section. Pour plus d'informations, consultez [Comment fonctionne le chiffrement au repos](#) dans le guide de l'utilisateur d'Amazon Elastic File System.

FSxPour les systèmes de fichiers Lustre, le chiffrement des données au repos est automatiquement activé lors de la création d'un système de FSx fichiers Amazon. Pour plus d'informations, consultez la section [Chiffrer les données au repos](#) dans le guide de l'utilisateur d'Amazon FSx for Lustre.

Pour les types d'instance dotés de NVMe volumes, les données des volumes de stockage d'NVMeinstance sont chiffrées à l'aide d'un chiffrement XTS - AES -256 implémenté sur un module matériel de l'instance. Les clés de chiffrement sont générées à l'aide du module matériel et sont uniques à chaque périphérique de stockage d'NVMeinstance. Toutes les clés de chiffrement sont détruites lorsque l'instance est arrêtée ou résiliée et ne peuvent pas être récupérées. Vous ne pouvez pas désactiver le chiffrement et vous ne pouvez pas fournir votre propre clé de chiffrement. Pour plus d'informations, consultez la section [Encryption at rest](#) dans le guide de EC2 l'utilisateur Amazon.

Si vous invoquez un AWS service qui transmet les données des clients à votre ordinateur local à des fins de stockage, reportez-vous au chapitre Sécurité et conformité du guide de l'utilisateur de ce service pour obtenir des informations sur la manière dont ces données sont stockées, protégées et cryptées. AWS ParallelCluster

## Chiffrement en transit

Par défaut, toutes les données transmises depuis l'ordinateur client en cours d'exécution AWS ParallelCluster et les points de terminaison du AWS service sont cryptées en envoyant le tout via une TLS connexion HTTPS /. Le trafic entre les nœuds du cluster peut être automatiquement chiffré, en fonction des types d'instances sélectionnés. Pour plus d'informations, consultez la section [Chiffrement en transit](#) dans le guide de EC2 l'utilisateur Amazon.

## Consultez aussi

- [Protection des données sur Amazon EC2](#)
- [Protection des données dans EC2 Image Builder](#)
- [Protection des données dans AWS CloudFormation](#)
- [Protection des données sur Amazon EFS](#)
- [Protection des données dans Amazon S3](#)
- [Protection des données FSx pour Lustre](#)

## Identity and Access Management (Gestion des identités et des accès) pour AWS ParallelCluster

AWS ParallelCluster utilise des rôles pour accéder à vos AWS ressources et à leurs services. Les politiques d'instance et d'utilisateur AWS ParallelCluster utilisées pour accorder des autorisations sont documentées à l'adresse [AWS Identity and Access Management autorisations dans AWS ParallelCluster](#).

La seule différence majeure réside dans la façon dont vous vous authentifiez lorsque vous utilisez un utilisateur standard et des informations d'identification à long terme. Bien qu'un utilisateur ait besoin d'un mot de passe pour accéder à la console d'un AWS service, ce même utilisateur a besoin d'une paire de clés d'accès pour effectuer les mêmes opérations à l'aide de AWS ParallelCluster. Toutes les autres informations d'identification à court terme sont utilisées de la même manière qu'avec la console.

Les informations d'identification utilisées par AWS ParallelCluster sont stockées dans des fichiers en texte clair et ne sont pas cryptées.

- Le fichier `$HOME/.aws/credentials` stocke les informations d'identification à long terme requises pour accéder à vos ressources AWS. Vos ID de clé d'accès et clé d'accès secrète sont inclus.
- Les informations d'identification à court terme, telles que celles pour les rôles que vous assumez ou pour les services AWS IAM Identity Center, sont également stockées dans les dossiers `$HOME/.aws/cli/cache` et `$HOME/.aws/sso/cache`, respectivement.

### Atténuation des risques

- Nous vous recommandons fortement de configurer vos autorisations de système de fichiers sur le dossier `$HOME/.aws` et ses dossiers et fichiers enfants, afin de restreindre l'accès aux seuls utilisateurs autorisés.
- Utilisez des rôles avec des informations d'identification temporaires dans la mesure du possible pour réduire les risques de dommages si les informations d'identification sont compromises. Utilisez les informations d'identification à long terme uniquement pour demander et actualiser les informations d'identification de rôle à court terme.

## Validation de la conformité pour AWS ParallelCluster

Les auditeurs tiers évaluent la sécurité et la conformité des services AWS dans le cadre de plusieurs programmes de conformité AWS. L'utilisation d'un service AWS ParallelCluster pour accéder à un service ne modifie pas la conformité de ce service.

Pour obtenir la liste des services AWS relevant de programmes de conformité spécifiques, [consultez Services AWS relevant de programmes de conformité](#) . Pour obtenir des renseignements généraux, consultez [Programmes de conformité AWS](#) .

Vous pouvez télécharger les rapports de l'audit externe avec AWS Artifact. Pour plus d'informations, consultez [Téléchargement de rapports sur AWS Artifact](#).

Votre responsabilité de conformité lors de l'utilisation de AWS ParallelCluster est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise, ainsi que par la législation et la réglementation applicables. AWS fournit les ressources suivantes pour faciliter le respect de la conformité :

- [Guides de démarrage rapide de la sécurité et de la conformité](#) – Ces guides de déploiement traitent de considérations architecturales et indiquent comment déployer des environnements de référence axés sur la sécurité et la conformité sur AWS.
- [AWS Livre blanc sur l'architecture pour la sécurité et la conformité HIPAA sur Amazon Web Services](#) : ce livre blanc décrit comment les entreprises peuvent utiliser AWS pour créer des applications conformes à la loi HIPAA.
- [Ressources de conformité AWS](#) : cet ensemble de manuels et de guides peut s'appliquer à votre secteur et à votre emplacement.
- [Évaluation des ressources à l'aide de règles](#) dans le Guide du développeur AWS Config : le service évalue dans quelle mesure vos configurations de ressources sont conformes aux pratiques internes, aux directives sectorielles et aux réglementations.

- [AWS Security Hub](#) – Ce service AWS fournit une vue complète de votre état de sécurité au sein d'AWS qui vous permet de vérifier votre conformité aux normes du secteur et aux bonnes pratiques de sécurité.

## Application d'une version minimale de TLS 1.2

Pour renforcer la sécurité lors de la communication avec AWS les services, vous devez configurer votre AWS ParallelCluster système pour utiliser TLS 1.2 ou version ultérieure. Lorsque vous utilisez AWS ParallelCluster, Python est utilisé pour définir la version TLS.

Pour vous assurer qu'aucune version de TLS antérieure à TLS 1.2 n'est utilisée, vous devrez peut-être recompiler OpenSSL pour appliquer ce minimum, puis recompiler Python pour utiliser la nouvelle version d'OpenSSL.

### Déterminez vos protocoles actuellement pris en charge

Tout d'abord, créez un certificat auto-signé à utiliser pour le serveur de test et le SDK Python à l'aide d'OpenSSL.

```
$ openssl req -subj '/CN=localhost' -x509 -newkey rsa:4096 -nodes -keyout key.pem -out cert.pem -days 365
```

Ensuite, faites tourner un serveur de test à l'aide d'OpenSSL.

```
$ openssl s_server -key key.pem -cert cert.pem -www
```

Dans une nouvelle fenêtre de terminal, créez un environnement virtuel et installez le SDK Python.

```
$ python3 -m venv test-env
source test-env/bin/activate
pip install botocore
```

Créez un nouveau script Python nommé `check.py` qui utilise la bibliothèque HTTP sous-jacente du SDK.

```
$ import urllib3
URL = 'https://localhost:4433/'
```

```
http = urllib3.PoolManager(
 ca_certs='cert.pem',
 cert_reqs='CERT_REQUIRED',
)
r = http.request('GET', URL)
print(r.data.decode('utf-8'))
```

Exécutez votre nouveau script.

```
$ python check.py
```

Des détails sur la connexion effectuée s'affichent. Recherchez « Protocole : » dans le résultat. Si le résultat est « TLSv1.2 » ou version ultérieure, le SDK est par défaut TLS v1.2 ou version ultérieure. S'il s'agit d'une version antérieure, vous devez recompiler OpenSSL et recompiler Python.

Cependant, même si votre installation de Python est par défaut TLS v1.2 ou version ultérieure, il est toujours possible pour Python de renégocier vers une version antérieure à TLS v1.2 si le serveur ne prend pas en charge TLS v1.2 ou une version ultérieure. Pour vérifier que Python ne renégocie pas automatiquement des versions antérieures, redémarrez le serveur de test avec ce qui suit.

```
$ openssl s_server -key key.pem -cert cert.pem -no_tls1_3 -no_tls1_2 -www
```

Si vous utilisez une version antérieure d'OpenSSL, vous n'avez peut-être pas l'indicateur `-no_tls_3` disponible. Si c'est le cas, supprimez l'indicateur car la version d'OpenSSL que vous utilisez ne prend pas en charge TLS v1.3. Exécutez à nouveau le script Python.

```
$ python check.py
```

Si votre installation de Python ne renégocie pas correctement des versions antérieures à TLS 1.2, vous devriez recevoir une erreur SSL.

```
$ urllib3.exceptions.MaxRetryError: HTTPSConnectionPool(host='localhost',
port=4433): Max retries exceeded with url: / (Caused by SSLError(SSLError(1, '[SSL:
UNSUPPORTED_PROTOCOL] unsupported protocol (_ssl.c:1108)')))
```

Si vous êtes en mesure d'établir une connexion, vous devez recompiler OpenSSL et Python pour désactiver la négociation des protocoles antérieurs à TLS v1.2.

## Compiler OpenSSL et Python

Pour vous assurer que AWS ParallelCluster cela ne négocie pas pour une version antérieure à TLS 1.2, vous devez recompiler OpenSSL et Python. Pour ce faire, copiez le contenu suivant pour créer un script et exécutez-le.

```
#!/usr/bin/env bash
set -e

OPENSSL_VERSION="1.1.1d"
OPENSSL_PREFIX="/opt/openssl-with-min-tls1_2"
PYTHON_VERSION="3.8.1"
PYTHON_PREFIX="/opt/python-with-min-tls1_2"

curl -O "https://www.openssl.org/source/openssl-$OPENSSL_VERSION.tar.gz"
tar -xzf "openssl-$OPENSSL_VERSION.tar.gz"
cd openssl-$OPENSSL_VERSION
./config --prefix=$OPENSSL_PREFIX no-ssl3 no-tls1 no-tls1_1 no-shared
make > /dev/null
sudo make install_sw > /dev/null

cd /tmp
curl -O "https://www.python.org/ftp/python/$PYTHON_VERSION/Python-$PYTHON_VERSION.tgz"
tar -xzf "Python-$PYTHON_VERSION.tgz"
cd Python-$PYTHON_VERSION
./configure --prefix=$PYTHON_PREFIX --with-openssl=$OPENSSL_PREFIX --disable-shared > /dev/null
make > /dev/null
sudo make install > /dev/null
```

On compile ainsi une version de Python qui a un OpenSSL lié statiquement qui ne négocie pas automatiquement quoi que ce soit d'antérieur à TLS 1.2. On installe également OpenSSL dans le répertoire `/opt/openssl-with-min-tls1_2` et Python dans le répertoire `/opt/python-with-min-tls1_2`. Après avoir exécuté ce script, confirmez l'installation de la nouvelle version de Python.

```
$ /opt/python-with-min-tls1_2/bin/python3 --version
```

Ce qui suit devrait s'imprimer.

Python 3.8.1

Pour confirmer que cette nouvelle version de Python ne négocie pas une version antérieure à TLS 1.2, exécutez à nouveau les étapes à partir de [Déterminez vos protocoles actuellement pris en charge](#) à l'aide de la version Python nouvellement installée (c'est-à-dire `/opt/python-with-min-tls1_2/bin/python3`).



# Notes de mise à jour et historique du document

Le tableau suivant décrit les principales mises à jour et les nouvelles fonctions pour le Guide de l'utilisateur AWS ParallelCluster . Nous mettons aussi la documentation à jour régulièrement pour prendre en compte les commentaires qui nous sont envoyés.

| Modification                                                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Date            |
|-----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <a href="#">AWS ParallelCluster Sortie de la version 2024.07.1 de l'interface utilisateur</a> | <p>Nous sommes ravis d'annoncer la sortie de la version AWS ParallelCluster 2024.07.1 de l'interface utilisateur.</p> <p>Changements :</p> <ul style="list-style-type: none"><li>• Ajout du support pour AWS ParallelCluster 3.10.1.</li></ul> <p>Correctifs de bogue :</p> <ul style="list-style-type: none"><li>• Correction d'un bogue qui perturbait le rendu des informations de comptabilité des tâches.</li><li>• Correction d'un bogue dans le mécanisme de signalisation des fonctionnalités qui désactivait toutes les fonctionnalités de PC 3.2.0+ sur PC 3.10.0+.</li></ul> <p>Sécurité:</p> <ul style="list-style-type: none"><li>• Mettez à niveau la cryptographie de 42.0.2 à 42.0.4 et spécifiez des algorithmes</li></ul> | 24 juillet 2024 |

de JWT décodage pour remédier à la vulnérabilité et. <https://cwe.mitre.org/data/definitions/476.html>  
<https://cwe.mitre.org/data/definitions/327.html>

- Mettez à niveau Node.js de 16.20.2 à 18.20.3 et Next.js de 13.5.1 à 14.1.1 pour corriger la vulnérabilité. <https://cwe.mitre.org/data/definitions/918.html>
- Contrainte de la version 8.17.1+ pour corriger la vulnérabilité. <https://cwe.mitre.org/data/definitions/476.html>

Consultez le journal des [modifications complet](#).

### [AWS ParallelCluster version 3.10.1 publiée](#)

Nous sommes ravis de vous annoncer la sortie de la version AWS ParallelCluster 3.10.1.

8 juillet 2024

Correction d'un bogue :

- Correction d'un échec de création d'image dans les régions chinoises.

[AWS ParallelCluster Sortie de la version 2024.07.0 de l'interface utilisateur](#)

Nous sommes ravis d'annoncer la sortie de la version AWS ParallelCluster 2024.07.0 de l'interface utilisateur. 2 juillet 2024

Fonctionnalités :

- Ajout du support pour la AWS ParallelCluster version 3.10.0.

## [AWS ParallelCluster version 3.10.0 publiée](#)

Nous sommes ravis de vous annoncer la sortie de la version AWS ParallelCluster 3.10.0

27 juin 2024

Pour effectuer une mise à niveau, tapez `sudo pip install --upgrade aws-parallelcluster` .

Améliorations :

- Ajoutez une nouvelle section de configuration `Scheduling/SlurmSettings/ExternalSlurmdbd` pour connecter le cluster à un Slurmdbd externe.
- Autoriser l'exécution de build-image dans un réseau isolé.
- Ajoutez le support pour Amazon Linux 2023.
- Ajoutez le support pour `price-capacity-optimized` en tant que `AllocationStrategy` .
- Ajoutez un validateur pour empêcher l'utilisation de groupes de placement avec des blocs de capacité.

Changements :

- CentOS 7 n'est plus pris en charge.
- Mettez à niveau le client Cinc vers la version 18.4.12 à partir de la version 18.2.7.
- Mettez à jour munge vers la version 0.5.16 (à partir de 0.5.15).
- Mettez à jour Pmix vers la version 5.0.2 (à partir de la version 4.2.9).
- Mettez à niveau les dépendances des livres de recettes tiers :
  - apt-7.5.22 (à partir de apt-7.5.14)
  - openssh-2.11.12 (depuis openssh-2.11.3)
- Supprimer le livre de recettes tiers : selinux-6.1.12.
- Mettez à niveau le EFA programme d'installation vers 1.32.0.
  - Pilote EFA : efa-2.8.0-1
  - Configuration EFA : efa-config-1.16-1
  - Profil EFA : efa-profile-1.7-1
  - LibFabric-AWS : libfabric-aws-1.21.0-1

- Noyau RDMA : `rdma-core-50.0-1`
- Ouvert MPI : `openmpi40-aws-4.1.6-3` et `openmpi50-aws-5.0.2-12`
- Mettez à niveau NVIDIA le pilote vers la version 535.183.01 (à partir de 535.154.05).
- Mettez à jour Python vers la version 3.9.19 (à partir de la version 3.9.17).
- Mettez à niveau la MPI bibliothèque Intel vers 2021.12.1.8 (à partir de 2021.9.0.43482).

#### Correctifs de bogue :

- Corrigez la configuration des associations de référentiels de données à créer `AutoExportPolicy` et `AutoImportPolicy` facultative.
- Correction d'un problème lors de la suppression du cluster qui permet désormais de terminer le nettoyage du parc informatique lorsque les instances sont soit en état d'arrêt, soit en état d'arrêt. Cela permet d'éviter les échecs

de suppression de clusters pour les types d'instances dont les cycles de terminais on sont plus longs.

- Autorisez l'activation du tableau de bord Cloudwatch et la désactivation des alarmes dans la Monitoring section de configuration du cluster.
- Autoriser ParallelCluster Custom Resource à supprimer les validateurs utilisant `Pcluster/SuppressValidators`.
- Supprimé `/etc/profile.d/pcluster.sh` afin qu'il ne soit pas exécuté à chaque connexion utilisateur et qu'il ne soit `cfn_bootstrap_virtualenv` pas ajouté dans la variable d'`PATH` environnement.
- Corrigez la ParallelCluster API spécification en remplaçant le champ `failureReason` par `failures` en `DescribeCluster` réponse.
- Corrigez les ParallelCluster API spécifications en ajoutant le statut de CloudFormation pile

manquant :IMPORT\_\*,REVIEW\_IN  
\_PROGRESS ,  
etUPDATE\_FAILED .

- Correction d'un problème qui empêchait les mises à jour des clusters d'inclure des EFS systèmes de fichiers chiffrés en transit.
- Correction d'un problème qui empêchait les services slurmctld et slurmdbd de redémarrer lors du redémarrage du nœud principal lorsqu'ils étaient utilisés pour des données internes partagées. EFS
- Sur les systèmes Ubuntu, supprimez la configuration logrotate par défaut pour les fichiers journaux cloud-init qui entraient en conflit avec la configuration provenant de Parallelcluster.
- Corrigez l'échec de génération de l'image avec la version RHEL 8.10 ou une version ultérieure.



## [Terraform Provider pour AWS ParallelCluster 1.0.0 est sorti](#)

Nous sommes ravis d'annoncer la sortie de Terraform Provider pour AWS ParallelCluster la version 1.0.0.

Fonctionnalités :

- [Journal des modifications complet](#)

## [Le module Terraform pour AWS ParallelCluster 1.0.0 est sorti](#)

Nous sommes ravis d'annoncer la sortie du module Terraform pour AWS ParallelCluster la version 1.0.0.

Fonctionnalités :

- [Journal des modifications complet](#)

## [AWS ParallelCluster version 3.9.3 publiée](#)

Nous sommes ravis de vous annoncer la sortie de la version AWS ParallelCluster 3.9.3

19 juin 2024

Pour effectuer une mise à niveau, tapez `sudo pip install --upgrade aws-parallelcluster`

Fonctionnalités :

- Ajout de la prise en charge FSx Lustre en tant que type de stockage partagé dans `us-east-1` .

Correctifs de bogue :

- `cloud_dns` Supprimez `-le` de `Slurmctl` Parameters la configuration du Slurm pour éviter les problèmes de ventilation du Slurm.

Cela n'est pas obligatoire, car nous avons défini les adresses IP lors du lancement de l'instance.

[AWS ParallelCluster version 3.9.2 publiée](#)

Nous sommes ravis de vous annoncer la sortie de la version AWS ParallelCluster 3.9.2

28 mai 2024

Fonctionnalités :

- Passez Slurm à la version 23.11.7 (à partir de la version 23.11.4).
- Pour plus de détails, consultez CHANGELOG [3.9.2 GitHub](#)

[AWS ParallelCluster Sortie de la version 2024.05.0 de l'interface utilisateur](#)

AWS ParallelCluster La version 2024.05.0 de l'interface utilisateur a été publiée.

14 mai 2024

Correctifs de bogue :

- Correction d'un bug dans le frontend qui bloquait l'interface utilisateur lorsque l'utilisateur ouvrait le panneau Job Status.
- [Changelog complet](#)

[AWS ParallelCluster Sortie de la version 2024.04.0 de l'interface utilisateur](#)

AWS ParallelCluster La version 2024.04.0 de l'interface utilisateur a été publiée.

17 avril 2024

Fonctionnalités :

- Ajout du support pour la AWS ParallelCluster version 3.9.1
- [Changelog complet](#)

[AWS ParallelCluster version 3.9.1 publiée](#)

Nous sommes ravis de vous annoncer la sortie de la version AWS ParallelCluster 3.9.1

11 avril 2024

Pour effectuer la mise à niveau, entrez les informations suivantes : `sudo pip install --upgrade aws-parallelcluster`

Correctifs de bogues

- Supprimez la suppression récursive du répertoire de stockage partagé `mountdir` lors du démontage de systèmes de fichiers dans le cadre d'une opération de mise à jour du cluster.

[AWS ParallelCluster version  
3.9.1 publiée](#)

Nous sommes ravis de vous annoncer la sortie de la version AWS ParallelCluster 3.9.1

11 avril 2024

Pour effectuer la mise à niveau, entrez les informations suivantes : `sudo pip install --upgrade aws-parallelcluster`

Correctifs de bogues

- Supprimez la suppression récursive du répertoire de stockage partagé `mountdir` lors du démontage de systèmes de fichiers dans le cadre d'une opération de mise à jour du cluster.

## [AWS ParallelCluster Sortie de la version 2024.03.0 de l'interface utilisateur](#)

AWS ParallelCluster La version 2024.03.0 de l'interface utilisateur a été publiée.

12 mars 2024

### Fonctionnalités :

- Ajout du support pour la AWS ParallelCluster version 3.9.0
- Ajout du support pour Ubuntu 22.04 et Red Hat Enterprise Linux 9
- Ubuntu 18.04 obsolète

### Corrections de bugs

- Correction d'un problème qui empêchait certains clusters d'apparaître en cas d'utilisation de plusieurs clusters

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers du [aws-parallelcluster-ui](#) package sur GitHub.

## [AWS ParallelCluster version 3.9.0 publiée](#)

5 mars 2024

Nous sommes ravis de vous annoncer la sortie de la version AWS ParallelCluster 3.9.0

Pour effectuer la mise à niveau, entrez les informations suivantes : `sudo pip install --upgrade aws-parallelcluster`

Améliorations :

- Ajoutez le paramètre de configuration `DeploymentSettings/DefaultUserHome` pour permettre aux utilisateurs de déplacer le répertoire personnel de l'utilisateur par défaut au `/local/home` lieu de `/home` (par défaut).
- Permet de mettre à jour `MinCount Queue` et `ComputeResource` de configurer les paramètres sans avoir à arrêter le parc informatique. `MaxCount` Il est désormais possible de les mettre à jour en les réglant `Scheduling/SlurmSettings/QueueUpdateStrategy` sur `TERMINATE`. AWS ParallelCluster mettra fin uniquement aux nœuds supprimés lors d'un

redimensionnement de la capacité du cluster effectué par le biais d'une mise à jour du cluster.

- Permet de mettre à jour le stockage partagé externe de type Efs FsxLustre, FsxOntap,, FsxOpenZfs FileCache sans remplacer le parc de calcul et de connexion.
- Ajoutez le support pour RHEL9.
- Ajout du support pour Rocky Linux 9 tel que CustomAmi créé par le biais build-image du processus . Aucun AWS ParallelCluster Rocky9 Linux officiel n'AM est disponible pour le moment.
- Supprimer CommunicationParameters de la liste de refus des Slurm paramètres personnalisés.
- Ajouter un DeploymentSettings/DisableSudoAccessForDefaultUser paramètre pour désactiver l'accès sudo de l'utilisateur par défaut est pris en charge OSes.
- Modifications apportées FSx à quatre systèmes de fichiers Lustre créés par



ParallelCluster : Changez la version du serveur Lustre en 2.15.

- Ajoutez la possibilité de choisir entre des pilotes Nvidia open source et à code fermé lors de la création d'unAMI, via l'attribut ['cluster']['nvidia']['kernel\_open'] cookbook node.
- \* Ajoutez une option de configuration clustermgtd ec2\_instance\_missing\_max\_count pour autoriser un nombre configurable de tentatives afin d'assurer la cohérence entre les instances Amazon EC2 Descrivez et les instances exécutées.

### Modifications

- Passez Slurm à la version 23.11.4 (au lieu de 23.02.7).
- Mettez à jour NVIDIA le pilote vers la version 535.154.05.
- Ajout du support pour Python 3.11, 3.12 dans CLI pcluster et. aws-parallelcluster-batch-cli

- Créez des interfaces réseau à l'aide de l'index des cartes réseau à partir de la `NetworkCardIndex` liste des `EC2 DescribeInstances` réponses d'Amazon, au lieu de parcourir la `MaximumNetworkCards` plage en boucle.
- Échec de la création du cluster lors de l'utilisation des types d'instance P3, G3, P2 et G2 car leur GPU architecture n'est pas compatible avec les pilotes Nvidia Open Source (OpenRM) introduits dans la version 3.8.0.
- Mettre à niveau les dépendances des livres de recettes tiers : `nfs-5.1.2` (à partir de `nfs-5.0.0`)
- Mettez le EFA programme d'installation à niveau `1.30.0`.
  - Pilote EFA : `efa-2.6.0-1`
  - Configuration EFA : `efa-config-1.15-1`
  - Profil EFA : `efa-profile-1.6-1`
  - LibFabric-AWS : `libfabric-aws-1.19.0`

- Noyau RDMA : `rdma-core-46.0-1`
- Ouvert MPI : `openmpi40-aws-4.1.6-2` et `openmpi50-aws-5.0.0-11`
- Mise à niveau NICE DCV vers la version `2023.1-16388`.
  - serveur : `2023.1.16388-1`
  - xdcv : `2023.1.565-1`
  - GL : `2023.1.1047-1`
  - visualiseur Web : `2023.1.16388-1`

#### Correctifs de bogues

- Correction d'un problème qui entraînait l'échec d'une tâche lorsqu'elle était soumise en tant qu'utilisateur Active Directory à partir de nœuds de connexion. Le problème était dû à une configuration incomplète de l'intégration avec l'Active Directory externe sur le nœud principal.
- Refactorisez IAM les politiques définies dans le CloudFormation modèle `parallelcluster-policies.yaml` pour éviter les échecs de

déploiement provoqués par des politiques dépassant les limites. ParallelCluster API IAM

- Correction d'un problème empêchant le démarrage des nœuds de connexion lorsque le nœud principal met plus de temps que prévu à écrire les clés.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers du [aws-parallelcluster-ui](#) package sur GitHub.

### [AWS ParallelCluster Sortie de la version 2024.02.0 de l'interface utilisateur](#)

AWS ParallelCluster Sortie de la version 2024.02.0 de l'interface utilisateur

8 février 2024

Changements :

- Mise à jour de l'environnement d'exécution Lambda vers Python v3.9

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers du [aws-parallelcluster-ui](#) package sur GitHub.

[AWS ParallelCluster Sortie de la version 2023.12.0 de l'interface utilisateur](#)

AWS ParallelCluster La version 2023.12.0 de l'interface utilisateur a été publiée.

21 décembre 2023

Fonctionnalités :

- Ajout de la prise en charge PCUI du déploiement avec un réseau privé.
- Possibilité supplémentaire d'appliquer éventuellement une limite d'autorisations à chaque IAM rôle créé par les PCAPI infrastructures PCUI et
- Possibilité supplémentaire d'appliquer éventuellement un préfixe à chaque IAM rôle et à chaque politique créés par l'PCAPI infrastructure PCUI and.
- Ajout du support pour ParallelCluster la version 3.8.0, sans parité des fonctionnalités dans l'assistant.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers du [aws-parallelcluster-ui](#) package sur GitHub.

[AWS ParallelCluster version  
3.8.0 publiée](#)

AWS ParallelCluster version  
3.8.0 publiée.

19 décembre 2023

Améliorations :

- Ajoutez la prise en charge d'Amazon EC2 Capacity Blocks for ML.
- Ajout du support pour Rocky Linux 8 tel que CustomAmi créé par le biais build-image du processus . Aucun AWS ParallelCluster Rocky8 Linux officiel n'AM est disponible pour le moment.
- Ajoutez un Scheduling/ScalingStrategy paramètre pour contrôler la stratégie de dimensionnement du cluster à utiliser lors du lancement d'EC2 instances Amazon pour les nœuds de Slurm calcul. Les valeurs possibles sont all-or-nothing greedy-all-or-nothing best-effort ,, all-or-nothing par défaut.
- Ajoutez un HeadNode/SharedStorageType paramètre pour utiliser le EFS stockage plutôt que NFS les exportations

depuis le volume racine du nœud principal pour les ressources du système de fichiers partagées au sein du cluster : Intel ParallelCluster, Slurm et données. /home Cette amélioration réduit la charge sur le réseau du nœud principal.

- Autorisez le montage /home en tant que stockage partagé EFS ou FSx externe via la SharedStorage section du fichier de configuration.
- Ajoutez un nouveau paramètre SlurmSettings/MungeKeySecretArn pour autoriser l'utilisation d'une MUNGE clé externe définie par l'utilisateur depuis AWS Secrets Manager.
- Ajoutez un Monitoring/Alarms/Enabled paramètre pour activer Amazon CloudWatch Alarms pour le cluster.
- Ajoutez des alarmes du nœud principal pour surveiller les bilans de EC2 santé, l'CPU utilisation et l'état général du nœud principal d'Amazon, puis ajoutez-les au CloudWatch

tableau de bord créé avec le cluster.

- Ajoutez la prise en charge des associations de référentiels de données lors de l'utilisation de `PERSISTENT_STORAGE` for a managed FSx for Lustre.
- Ajoutez un `Scheduling/SlurmSettings/Database/DatabaseName` paramètre pour permettre aux utilisateurs de spécifier un nom personnalisé pour la base de données sur le serveur de base de données à utiliser pour la Slurm comptabilité.
- Créez `InstanceTypeCapacityReservationTarget/CapacityReservationId` dans la ressource de calcul.
- Ajout de la possibilité de spécifier un préfixe pour les IAM rôles et les politiques créés par AWS ParallelCluster API.
- Ajoutez la possibilité de spécifier une limite d'autorisations à appliquer aux IAM



rôles et aux politiques créés par AWS ParallelCluster API.

## Modifications

- Passez Slurm à la version 23.02.7 (au lieu de 23.02.6).
- Mettez à jour NVIDIA le pilote vers la version 535.129.03.
- Mettez à niveau le CUDA Toolkit vers la version 12.2.2.
- Utilisez des NVIDIA GPU pilotes Open Source (OpenRM) comme module NVIDIA noyau pour Linux au lieu d'un module source NVIDIA fermé.
- Supprimez la prise en charge du paramètre de `all_or_nothing_batch` configuration dans le programme de Slurm reprise, en faveur de la nouvelle configuration du `Scheduling/Scaling Strategy` cluster.
- La convention de dénomination des alarmes du cluster a été modifiée en « [cluster-name] - [component-name] - [metric] ».

- Modifiez les types de EBS volumes par défaut dans ADC les régions de gp2 à gp3, à la fois pour le volume racine et pour les volumes supplémentaires.
- La limite d'autorisation facultative pour le AWS ParallelCluster API est désormais appliquée à chaque IAM rôle créé par l'API infrastructure.
  - Mettez à niveau le EFA programme d'installation vers 1.29.1.
  - Pilote EFA : `efa-2.6.0-1`
  - Configuration EFA : `efa-config-1.15-1`
  - Profil EFA : `efa-profile-1.5-1`
  - LibFabric-AWS : `libfabric-aws-1.19.0-1`
  - Noyau RDMA : `rdma-core-46.0-1`
  - Ouvert MPI : `openmpi40-aws-4.1.6-1`
- Passez GDRCopy à la version 2.4 dans toutes les versions prises en charge OSes, sauf pour Centos 7 où la version 2.3.1 est utilisée.

- Passez `aws-cfn-bootstrap` à la version 2.0-28.
- Ajout du support pour Python 3.10 dans `aws-parallelcluster-batch-cli`

### Correctifs de bogues

- Corrigez une configuration de dimensionnement incohérente après l'annulation de la mise à jour du cluster lors de la modification de la liste des types d'instances déclarés dans les ressources de calcul.
- Corrigez la génération de SSH clés utilisateur lors du changement d'utilisateur sans privilège root dans des clusters intégrés à un LDAP serveur externe via des fichiers de configuration de cluster.
- Correction de la désactivation Slurm du mode économie d'énergie lors du réglage `Scaledown IdleTime = -1`.
- Corrigez le chemin codé en dur vers le répertoire Slurm d'installation dans `update_slurm_database_password.sh` le

script pour Slurm Accountin

g.

[AWS ParallelCluster version  
3.7.2 publiée](#)

AWS ParallelCluster version  
3.7.2 publiée.

25 octobre 2023

Changements :

- Passez Slurm à la version 23.02.6.

## [AWS ParallelCluster Sortie de la version 2023.10.0 de l'interface utilisateur](#)

AWS ParallelCluster La version 2023.10.0 de l'interface utilisateur est sortie.

20 octobre 2023

### Fonctionnalités :

- Ajout de la prise en charge de la ParallelCluster version 3.7.2 avec la parité des fonctionnalités dans l'assistant limitée au cache de FSx fichiers et à la compatibilité de planification basée sur la mémoire avec plusieurs types d'instances.

### Correctifs de bogue :

- Correction d'un problème provoquant des erreurs d'interface utilisateur lorsqu'il PCUI n'était pas autorisé à interagir avec Cost Explorer.

### Améliorations

- Sécurité améliorée en réduisant le jeton TTL d'accès de 10 minutes à 5 minutes.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers du [aws-](#)

[parallelcluster-ui](#) package sur GitHub.

[AWS ParallelCluster version 3.7.1 publiée](#)

AWS ParallelCluster version 3.7.1 publiée.

22 septembre 2023

Changements :

- Passez Slurm à la version 23.02.5 (au lieu de 23.02.4).
  - Mettez à jour Pmix vers la version 4.2.6 (à partir de la version 3.2.3).
  - Mettez à jour libjwt vers la version 1.15.3 (à partir de la version 1.12.0).
- Mettez à niveau le EFA programme d'installation vers 1.26.1, en résolvant le problème RDMA d'écriture de données dans P5.
  - Pilote EFA : `efa-2.5.0-1`
  - Configuration EFA : `efa-config-1.15-1`
  - Profil EFA : `efa-profile-1.5-1`
  - libFabric-AWS : `libfabric-aws-1.18.2-1`
  - ERdma-noyau : `rdma-core-46.0-1`
  - Ouvert MPI : `openmpi40-aws-4.1.5-4`

[AWS ParallelCluster version 3.7.0 publiée](#)

AWS ParallelCluster version 3.7.0 publiée.

30 août 2023

Améliorations :

- Support de la configuration des priorités de nœuds statiques et dynamiques dans les ressources de calcul à l'aide d'un YAML fichier AWS ParallelCluster de configuration.
- Ajout du support pour Ubuntu 22. RSAles touches ne sont pas prises en charge par défaut.
- Ajoutez le paramètre de configuration de la file d'attente `JobExclusiveAllocation` pour allouer les nœuds d'une partition exclusivement à une seule tâche à un moment donné.
- Autoriser le `aws-parallelcluster-node` package Override au moment de la création et de la mise à jour du cluster. Pour le nœud principal, cela s'applique à la mise à jour du cluster. Utile uniquement à des fins de développement.

- Évitez de démarrer le NFS serveur sur les nœuds de calcul.
- Ajoutez la prise en charge des nœuds de connexion.
- Autorisez la planification basée sur la mémoire lorsque plusieurs types d'instances sont spécifiés pour une ressource de Slurm calcul.
- Ajoutez la prise en charge du montage d'Amazon File Cache existant en tant que stockage partagé.

#### Changements :

- Attribuez aux nœuds Slurm dynamiques une priorité (poids) de 1 000 par défaut. Ce faisant, Slurm vous pouvez donner la priorité aux nœuds statiques inactifs par rapport aux nœuds dynamiques inactifs.
- `aws-parallelcluster-r-node` Les démons Make ne gèrent que les partitions AWS ParallelCluster gérées Slurm.
- Augmentez l'intervalle entre les sondages du EFS-`utils` chien de garde à 10 secondes. Cette modification



ion s'applique lorsque ce paramètre `EncryptionInTransit` est défini sur `true`, seule condition à l'origine de l'exécution du chien de garde.

- Mettez à niveau le EFA programme d'installation vers `1.25.1`.
  - EFA-driver : `efa-2.5.0-1` (à partir de `efa-2.1.1g`)
  - EFA-Config : `efa-config-1.15-1` (à partir de `efa-config-1.13-1`)
  - Profil EFA : `efa-profile-1.5-1` (pas de changement)
  - libFabric-aws :  
`libfabric-aws-1.18.1-0` (depuis `libfabric-aws-1.17.1-1`)
  - RDMA-Core : `rdma-core-46.0-1` (à partir de `rdma-core-43.0-1`)
  - Ouvert MPI : `openmpi40-aws-4.1.5-4` (à partir de `openmpi40-aws-4.1.5-1`)
- Passez Slurm à la version `23.02.4`.

- Modifiez la valeur par défaut de `Imds/` de `v1.0 ImdsSupport` à `v2.0`.
- Dépréciez Ubuntu 18.
- Mettez à jour la taille du volume racine par défaut à 40 Go pour tenir compte des limites de Centos 7.
- Limitez les autorisations sur le fichier `/tmp/wait_condition_handle.txt` dans le nœud principal afin que seul le root puisse le lire.
- Créez un JSON fichier de mappage Slurm partition `-nodelist` à utiliser par les démons du package de nœuds pour reconnaître les partitions et les listes de nœuds gérées par un PC. Slurm
- Mettez à jour NVIDIA le pilote vers la version 535.54.03.
- Mettez à niveau CUDA la bibliothèque vers la version 12.2.0.
- Mettez à niveau NVIDIA Fabric Manager vers `nvidia-fabricmanager-535`.
- Mettez à jour ARM PL vers la version 23.04.1 pour Ubuntu 22.04 uniquement.
- Passez NICE DCV à la version `2023.0-15487` .

- Serveur : 2023.0.15487-1
- xdcv : 2023.0.551-1
- GL : 2023.0.1039-1
- visualiseur Web : 2023.0.15487-1

#### Correctifs de bogue :

- Ajoutez une validation à la `ScaledownIdleTime` valeur pour éviter de définir une valeur inférieure à -1.
- Corrigez l'échec de création de clusters avec Ubuntu Deep Learning AMI sur GPU les instances DCV activées.
- Correction d'un problème à l'origine de la création de IAM politiques pendantes lors de la création d'un fournisseur de ressources ParallelCluster CloudFormation personnalisé avec `CustomLambdaRole`.
- Correction d'un problème qui provoquait un désalignement du DNS nom des nœuds de calcul sur les instances dotées de plusieurs interfaces réseau, lorsque l'on utilisait des valeurs `SlurmSettings/Dns/UseEc2Hostnames` égales à `True`

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers pour le cluster [aws-parallelcluster](#) et les packages sur. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

### Publication de documentation uniquement

AWS ParallelCluster publication du guide de l'utilisateur spécifique à la version 3.

17 juillet 2023

Publication de documentation uniquement :

- AWS ParallelCluster la version 3 possède son propre guide d'utilisation distinct.

[AWS ParallelCluster version  
3.6.1 publiée](#)

AWS ParallelCluster version  
3.6.1 publiée.

5 juillet 2023

Changements :

- Évitez la duplication des nœuds observée `clustermgtd` si des nœuds de calcul sont ajoutés à plusieurs Slurm partitions.

Correctifs de bogue :

- Supprimez le codage en dur du nom (`/dev/sda1` et `dev/xvda`) du périphérique du volume racine et récupérez-le à partir du AMI ou des numéros utilisés pendant le processus `create-cluster`.
- Corrigez l'échec de création du cluster lors de l'utilisation d'une ressource CloudFormation personnalisée avec `ElasticIp` défini sur `True`.
- Corrigez les échecs de création et de mise à jour de clusters lors de l'utilisation d'une ressource AWS CloudFormation personnalisée avec des fichiers de configuration volumineux.

- Correction d'un problème qui empêchait la désactivation de la protection sur Ubuntu et qui n'autorisait pas Cross Memory Attach (CMA) dans libfabric.
- Corrigez la logique de basculement rapide en cas d'insuffisance de capacité lorsque vous utilisez plusieurs types d'instances et qu'aucune instance n'est renvoyée.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers pour le cluster [aws-parallelcluster](#) et les packages sur. [aws-parallelcluster-cookbook](#) [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster La version 2023.06.0 de l'interface utilisateur est sortie](#)

AWS ParallelCluster La version 2023.06.0 de l'interface utilisateur a été publiée.

7 juin 2023

Changements :

- Mise à niveau de la AWS ParallelCluster API version par défaut vers la version 3.6.0.

Correctifs de bogue :

- Correction d'un déploiement interrompu pour la région AWS GovCloud (ouest des États-Unis).
- Le panneau partagé charge désormais correctement les détails du cluster une fois la création commencée.

Remarques :

- La fonctionnalité de surveillance des coûts n'est pas disponible dans AWS GovCloud (US) Regions.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers du [aws-parallelcluster-ui](#) package sur GitHub.

[AWS ParallelCluster version 3.6.0 publiée](#)

AWS ParallelCluster version 3.6.0 publiée.

22 mai 2023

Documentation :

- Ajoutez de la documentation pour [AWS ParallelCluster API de bibliothèque Python](#).

Améliorations :

- Ajoutez le support pour RHEL8.
- Ajoutez une [ressource AWS CloudFormation personnalisée](#) pour créer et gérer des clusters avec CloudFormation.
- Ajoutez la prise en charge de [la personnalisation de la Slurm configuration du cluster](#) dans le YAML fichier AWS ParallelCluster de configuration.
- Construisez Slurm avec le support de LUA.
- Augmentez la limite du nombre maximum de files d'attente par cluster de 10 à 50. Chaque file d'attente peut contenir jusqu'à 50 ressources de calcul. Chaque cluster peut



- disposer d'un maximum de 50 ressources de calcul.
- Ajoutez la prise en charge de la spécification d'une séquence de plusieurs [scripts d'action personnalisés](#) pour un événement configuré dans `OnNodeStart`, `OnNodeConfigured`, et `OnNodeUpdated` paramètres.
  - Ajoutez une nouvelle section de configuration `HealthChecks /Gpu`, pour appliquer des contrôles de GPU santé sur un nœud de calcul avant l'exécution d'une tâche.
  - Ajoutez le support pour `Tags` dans la `ComputeResources` configuration `SlurmQueues` and `SlurmQueues /`.
  - Ajoutez le support pour [DetailedMonitoring](#) dans la `Monitoring` configuration.
  - Ajoutez `mem_used_percent` `disk_used_percent` des indicateurs pour le suivi de la mémoire du nœud principal et de l'utilisation du disque du volume racine dans le AWS ParallelCluster [CloudWatch tableau de bord](#), et configure

z des alarmes pour surveiller ces indicateurs.

- Ajoutez la prise en charge [de la rotation des journaux](#) pour les journaux AWS ParallelCluster gérés.
- Suivez les erreurs courantes des nœuds de calcul et le temps d'inactivité le plus long des nœuds dynamiques dans le [CloudWatch tableau de bord](#).
- Obligez le serveur DCV d'authentification à utiliser au moins le TLS-1.2 protocole lors de la création du SSL socket.
- Installez le package [NVIDIA Data Center GPU Manager \(DCGM\)](#) sur tous les systèmes d'exploitation pris en charge, à l'exception de aarch64 centos7 et linux2.
- Chargez le module noyau [nvidia-vm](#) par défaut pour fournir la fonctionnalité Unified Virtual Memory (UVM) au pilote. CUDA
- Installez le [démon de NVIDIA persistance](#) en tant que service système.

Changements :

- Passez Slurm à la version 23.02.2 (à partir de la version 22.05.8).
- Mettez à jour munge vers la version 0.5.15 (à partir de la version 0.5.14).
- Réglez Slurm TreeWidth le sur 30.
- Définissez les epilog configurations Slurm prolog et sur le répertoire cible /opt/slurm/etc/scripts/prolog.d/ et /opt/slurm/etc/scripts/epilog.d/ respectivement.
- Réglé Slurm BatchStartTimeout sur 3 minutes maximum pour exécuter des Prolog scripts lors de l'enregistrement du nœud de calcul.
- Augmentez la durée par défaut RetentionInDays des CloudWatch journaux de 14 à 180 jours.
- Mettez à niveau le EFA programme d'installation vers 1.22.1.
  - Dimension : 2.8.3-2
  - EFA-driver : efa-2.1.1g (pas de changement)

- EFA-Config : efa-confi  
g-1.13-1 (pas de  
changement)
- Profil EFA : efa-profi  
le-1.5-1 (pas de  
changement)
- libFabric-aws :  
libfabric-aws-1.17  
.1-1 (depuis)  
libfabric-aws-1.17  
.0-1
- RDMA-Core : rdma-  
core-43.0-1 (aucun  
changement)
- Ouvert MPI : openmpi40  
-aws-4.1.5-1 (pas  
de changement)
- Mettez à niveau la version  
du client Lustre vers 2.12  
Amazon Linux 2. Le client  
Lustre 2.12 a été installé  
sur Ubuntu 20.04, 18.04 et  
CentOS >= 7.7.
- Mettez à niveau la version  
du client Lustre vers  
2.10.8 CentOS 7.6.
- Mettez à niveau le NVIDIA  
pilote vers la version  
470.182.03 (à partir de  
la version 470.141.03 ).
- Mettez à niveau le NVIDIA  
Fabric Manager vers  
la version 470.182.0  
3 (à partir de la  
version 470.141.03 ).

- Mettez à niveau le NVIDIA CUDA kit d'outils vers la version 11.8.0 (à partir de la version 11.7.1).
- Mettez à niveau l'NVIDIA Container Toolkit vers la version 11.8.0.
- Mettez à niveau la MPI bibliothèque Intel vers la version 2021 Update 9 (à partir de la version 2021 Update 6). Pour plus d'informations, consultez [la mise à jour 9 des MPI bibliothèques Intel® 2021](#).
- Passez NICE DCV à la version 2023.0-15022 (à partir de la version 2022.2-14521 ).
  - serveur : 2023.0.15022-1 (à partir de la version 2022.2-14521-1 ).
  - xdcv : 2023.0.547-1 (à partir de la version 2022.2.519-1 ).
  - gl : 2023.0.1027-1 (à partir de la version 2022.2.1012-1 ).
  - web\_viewer : 2023.0.15022-1 (à partir de la version 2022.2.14521-1 ).

- Passez `aws-cfn-bootstrap` à la version `2.0-24`.
- Image de mise à niveau utilisée par l' `CodeBuild` environnement lors de la création d'images de conteneur pour les `AWS Batch clusters` :
  - `aws/codebuild/amazonlinux2-x86_64-standard:4.0` (à partir de `aws/codebuild/amazonlinux2-x86_64-standard:3.0` ).
  - `aws/codebuild/amazonlinux2-aarch64-standard:2.0` (à partir de `aws/codebuild/amazonlinux2-aarch64-standard:1.0` ).

#### Correctifs de bogue :

- Corrigez EFS les validateurs de groupes de sécurité FSx réseau Amazon et Amazon pour éviter de signaler de fausses erreurs.
- Corrigez le balisage manquant des ressources créées par `Image Builder` pendant l'`build-image` opération.

- Corrige la politique de mise MaxCount à jour pour toujours effectuer des comparaisons numériques sur la MaxCount propriété.
- Corrigez l'alignement IP sur les instances de nœuds de calcul dotées de plusieurs cartes réseau.
- Corrigez le remplacement de StoragePass in `slurm_parallelcluster_slurmdbd.conf` lorsqu'une mise à jour des paramètres de file d'attente est effectuée et que les configurations Slurm comptables ne sont pas mises à jour.
- Corrigez le problème à l'origine de la création de groupes de sécurité suspendus lors de la création d'un cluster avec un système de EFS fichiers existant.
- Correction d'un problème à l'origine de l'échec du `cfn-hup` démon lors de son redémarrage.
- Considérez les nœuds dynamiques marqués d'un `INVALID_REG` indicateur comme des échecs d'amorçage en

mode Slurm protégé. Les nœuds statiques dont Slurm l'enregistrement échoue sont déjà traités comme des échecs de démarrage après `lenode_replacement_timeout` .

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers pour le cluster [aws-parallelcluster](#) et les packages sur. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub



[AWS ParallelCluster La version 2023.05.0 de l'interface utilisateur est sortie](#)

AWS ParallelCluster La version 2023.05.0 de l'interface utilisateur a été publiée.

16 mai 2023

Améliorations :

- À partir de AWS ParallelCluster la version 3.6.0, ajoutez le support pour RHEL 8.
- Ajoutez la surveillance des coûts du cluster.
- À partir de AWS ParallelCluster la version 3.6.0, augmentez les quotas de file d'attente et de ressources de calcul.

Changements :

- Amélioration de l'interface utilisateur de l'assistant de création de clusters.
- Augmentation de la vitesse de déploiement de l' AWS ParallelCluster interface utilisateur.
- Amélioration de l'interface pour l'ajout d'un nouvel utilisateur.
- Les files d'attente se trouvent par défaut dans le sous-réseau du nœud principal.

### Correctifs de bogue :

- Passez à la bonne région une fois la création du cluster terminée.
- Corrigez l'affichage de l'indicateur de chargement dans la fonction « Modifier le cluster ».
- Corrigez la création de clusters lorsque la EBS SnapshotId propriété est supprimée.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers du [aws-parallelcluster-ui](#) package sur GitHub.

[AWS ParallelCluster La version 2023.04.0 de l'interface utilisateur est sortie](#)

AWS ParallelCluster La version 2023.04.0 de l'interface utilisateur a été publiée.

17 avril 2023

Améliorations :

- Refonte de l'assistant de création de clusters.
- Refonte de la page des journaux du cluster.
- Ajoutez un paramètre de nom personnalisé pour le stockage partagé.
- Ajoutez une sélection de stockage multiple lors de l'ajout de stockage à un cluster.
- Ajoutez `DeletionPolicy` le support pour Amazon EFS et FSx Lustre.
- Ajoutez `ImdsSupport` un paramètre dans la configuration du cluster.
- Ajoutez la prise en charge des types d'instances C7.
- Tutoriel ajouté [Revenir à une version précédente d'un document AWS Systems Manager](#).

Changements :

- YAML Configuration du cluster d'une taille maximale de 1 Mo.
- L'utilisateur n'est pas déconnecté en raison d'une autorisation avec des informations d'identification IAM temporaires de Boto3.
- Les options multithreading sont désactivées lorsqu'une HPC instance est sélectionnée.
- Suppression de la désactivation de la restauration sur la page de création du cluster.
- L'utilisateur n'est pas autorisé à utiliser l' AWS ParallelCluster interface utilisateur tant que les informations requises ne sont pas fournies.
- Jusqu'à 10 files d'attente peuvent être ajoutées.
- Le SSM-SessionManagerRunShell document n'est pas remplacé lors de l'installation de l' AWS ParallelCluster interface utilisateur.

#### Correctifs de bogue :

- Corrigez le lien de réinitialisation du mot de passe cassé

- Correctif cassé delete stack dû au fait qu'il EcrPrivateRepository n'était pas vide
- Correction d'un problème d'initialisation de la case à cocher Générer des SSH clés dans la section Propriétés de gestion de plusieurs utilisateurs.
- Correction d'un crash causé par une tâche aux propriétés non définies.
- SCRATCHFSxRéglages fixes.
- Bouton de démarrage et d'arrêt des instances fixe, toujours activé après avoir cliqué une fois.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers du [aws-parallelcluster-ui](#) package sur GitHub.

[AWS ParallelCluster version  
3.5.1 publiée](#)

AWS ParallelCluster version  
3.5.1 publiée.

29 mars 2023

Améliorations :

- Ajoutez un [exécutable d'pclusterCLIinstallation](#) autonome.

Changements :

- Mettez à niveau le EFA programme d'installation vers 1.22.0.
  - EFA-driver : efa-2.1.1 g (à partir de) efa-2.1.1-1
  - EFA-Config : efa-config-1.13-1 (depuis efa-config-1.12-1)
  - Profil EFA : efa-profile-1.5-1 (pas de changement)
  - libFabric-aws : libfabric-aws-1.17.0-1 (depuis) libfabric-aws-1.16.1amzn3.0-1
  - RDMA-Core : rdma-core-43.0-1 (aucun changement)
  - Ouvert MPI : openmpi40-aws-4.1.5-1 (à

partir de `openmpi40-aws-4.1.4-3` )

Passez NICE DCV à la version `2022.2-14521` .

- serveur : `2022.2.14521-1`
- xdcv : `2022.2.519-1`
- GL : `2022.2.1012-1`
- visualiseur Web : `2022.2.14521-1`

Correctifs de bogue :

- Corrigez les défaillances potentielles de lancement de nœuds causées par la correspondance de modèles entre `MountDir` et `/etc/exports` lors de la suppression de EBS volumes Amazon partagés dans le cadre d'une mise à jour de cluster.
- Correctif pour empêcher la troncature du fichier `compute_console_output` journal à chaque `clustermgtd` itération.

Pour plus de détails sur les modifications, consultez les `CHANGELOG` fichiers pour le cluster [aws-parallelcluster](#) et les packages sur. [aws-paral](#)

[lelcluster-cookbookaws-para](#)

[lelcluster-node](#) GitHub



[AWS ParallelCluster version 3.5.0 publiée](#)

AWS ParallelCluster version 3.5.0 publiée.

20 février 2023

Améliorations :

- Accédez aux clusters et gérez-les à l'aide de l'[AWS ParallelCluster interface utilisateur](#).
- Ajoutez des AWS ParallelCluster politiques versionnés dans un CloudFormation modèle que vous pouvez référencer dans vos charges de travail.
- Ajoutez une bibliothèque AWS ParallelCluster Python que vous pouvez utiliser avec votre propre code.
- Ajoutez la journalisation de la sortie de la console du nœud de calcul à Amazon CloudWatch en cas de défaillance du bootstrap du nœud de calcul.
- Ajoutez un champ d'échec contenant le code d'échec et la raison à `describe-cluster` afficher en cas d'échec de la création du cluster.
- Ajoutez des validateurs pour empêcher toute injection de chaîne malveillante lors de

l'appel du module de sous-processus.

- Échec de la création du cluster si l'état du cluster change PROTECTED pendant le provisionnement des nœuds statiques.

#### Changements :

- Mise à niveau vers Slurm la version 22.05.8 (à partir de la version 22.05.7)
- Mettez à niveau le EFA programme d'installation vers 1.21.0.
  - EFA-driver : efa-2.1.1-1 (à partir de) efa-2.1
  - EFA-Config : efa-config-1.12-1 (depuis efa-config-1.11-1)
  - Profil EFA : efa-profile-1.5-1 (pas de changement)
  - libFabric-aws : libfabric-aws-1.16.1amzn3.0-1 (depuis) libfabric-aws-1.16.1
  - RDMA-Core : rdma-core-43.0-1 (à partir de) rdma-core-43.0-2
  - Ouvert MPI : openmpi40-aws-4.1.4-3 (pas de changement)

- Rendez les journaux Slurm du contrôleur plus détaillés et activez la journalisation supplémentaire pour le plug-in Slurm d'économie d'énergie.

#### Correctifs de bogue :

- Corrigez la création de la base de données de cluster en vérifiant que le nom du cluster ne dépasse pas 40 caractères lorsque la gestion des Slurm comptes est activée.
- Corrigez un problème en raison duquel `clusterimg` et les nœuds de calcul, redémarrés Slurm, étaient remplacés en cas d'échec des vérifications de statut des EC2 instances Amazon.
- Corrigez un problème qui empêchait le lancement de nœuds de calcul dont les réservations de capacité étaient partagées par d'autres comptes en raison d'une IAM politique incorrecte appliquée au nœud principal.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers pour

[aws-parallelcluster](#), [aws-parallelcluster-cookbookaws-parallelcluster-node](#), et les packages sur. [aws-parallelcluster-ui](#) GitHub

### [AWS ParallelCluster version 3.4.1 publiée](#)

AWS ParallelCluster version 3.4.1 publiée.

13 janvier 2023

Correctifs de bogues :

- Corrigez un problème du Slurm planificateur susceptible d'entraîner l'application incorrecte des mises à jour de son registre interne de nœuds de calcul. Par conséquent, en cas de problème, EC2 les instances peuvent devenir indisponibles ou être soutenues par un type d'instance incorrect.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers pour le cluster [aws-parallelcluster](#) et les packages sur. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster version 3.4.0 publiée](#)

AWS ParallelCluster version 3.4.0 publiée.

22 décembre 2022

Améliorations :

- Ajoutez la prise en charge du lancement de nœuds dans plusieurs zones de disponibilité afin d'augmenter la disponibilité des capacités.
- Ajoutez la prise en charge de la spécification de plusieurs sous-réseaux pour chaque file d'attente afin d'augmenter la disponibilité de la capacité.
- Ajoutez un nouveau paramètre de configuration dans [Iam/ResourcePrefix](#) pour spécifier un préfixe pour le chemin et le nom des IAM ressources créées par AWS ParallelCluster.
- Ajoutez une nouvelle section de configuration [DeploymentSettings/LambdaFunctionsVpcConfig](#) pour spécifier la configuration Vpc utilisée par les fonctions AWS ParallelCluster Lambda.

- Ajoutez la possibilité de spécifier un script personnalisé à exécuter dans le nœud principal lors d'une mise à jour du cluster. Le script peut être spécifié avec [HeadNode/CustomActions /OnNodeUpdated](#) lors de l'utilisation en Slurm tant que planificateur.

#### Changements :

- Supprimez la création de cibles de EFS montage Amazon pour les systèmes de fichiers existants.
- Montez les systèmes de EFS fichiers à l'aide de `amazon-efs-utils`. Les systèmes de fichiers peuvent être montés à l'aide d'un chiffrement en transit et d'un utilisateur IAM autorisé.
- Installez `stunnel` 5.67 sur CentOS7 et Ubuntu pour prendre EFS en charge le chiffrement en transit.
- Mettez à niveau le EFA programme d'installation vers `1.20.0` (depuis `1.18.0`).
  - EFA-driver : `efa-2.1` (à partir de) `efa-1.16.0-1`

- EFA-Config : efa-confi  
g-1.11-1 (pas de  
changement)
- Profil EFA : efa-profi  
le-1.5-1 (pas de  
changement)
- libFabric-aws :  
libfabric-  
aws-1.16.1 (depuis)  
libfabric-aws-1.16  
.0~amzn4.0-1
- RDMA-Core : rdma-core  
-43.0-2 de () rdma-  
core-41.0-2
- Ouvert MPI : openmpi40  
-aws-4.1.4-3 depuis  
(openmpi40-aws-4.1.  
4-2 )
- Passez Slurm à la  
version 22.05.7 (à partir  
de 22.05.5).
- Mettez à niveau Python  
vers 3.9.16 et 3.7.16. (de  
3.9.15 et 3.7.13).
- Avec Slurm 22.05.7, les  
nœuds dynamiques en  
IDLE+CLOUD+COMPLET  
ING+POWER\_DOWN+NOT  
\_RESPONDING état ne  
sont pas considérés comme  
défectueux.

Pour plus de détails sur les  
modifications, consultez les

CHANGELOG fichiers pour le cluster [aws-parallelcluster](#) et les packages sur. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) [GitHub](#)



[AWS ParallelCluster version 3.3.1 publiée](#)

AWS ParallelCluster version 3.3.1 publiée.

2 décembre 2022

Changements :

- AMIs Les AWS ParallelCluster produits officiels sont désormais disponibles après deux ans de EC2 dépréciation d'Amazon.
- Augmentez la taille de la mémoire du AWS ParallelCluster API Lambda à 2048 afin de réduire les pénalités de démarrage à froid et d'éviter les délais d'attente.

Correctifs de bogue :

- Empêchez le remplacement des systèmes de fichiers gérés FSx pour Lustre et la perte de données lors des mises à jour de clusters qui incluent des modifications de l'ID du sous-réseau du parc informatique.
- [SharedStorage DeletionPolicy](#) s'applique aux actions de mise à jour du cluster.

Pour plus de détails sur les modifications, consultez le CHANGELOG fichier du package [aws-parallelcluster](#) sur. GitHub

[AWS ParallelCluster documentation uniquement \(note hpc6id\)](#)

AWS ParallelCluster mise à jour de documentation uniquement

2 décembre 2022

- AWS ParallelCluster ne prend pas en charge le type d'instance hpc6id pour le paramètre `/`.  
[HeadNodeInstanceType](#)

[AWS ParallelCluster version  
3.1.5 publiée](#)

AWS ParallelCluster version  
3.1.5 publiée.

16 novembre 2022

Améliorations :

- Corrigez le Slurm problème qui empêche la fermeture des nœuds inactifs.
- Mettre à jour le EFA programme d'installation vers la version 1.18.0
  - Pilote EFA : efa-1.16.0-1
  - EFA-Config : efa-config-1.11-1 (à partir de efa-config-1.9-1)
  - Profil EFA : efa-profile-1.5-1 (pas de changement)
  - LibFabric-aws : libfabric-aws-1.16.0~amzn4.0-1 (de) libfabric-1.13.2
  - RDMA-Core : rdma-core-41.0-2 (à partir de rdma-core-37.0)
  - Ouvert MPI : openmpi40-aws-4.1.4-2 (à partir de openmpi40-aws-4.1.1-2 )

Changements :

- Ajoutez `lambda:ListTags` et `lambda:UpdateResource` à celui `ParallelClusterUserRole` utilisé par la AWS ParallelCluster API pile pour une mise à jour du cluster.
- Mettez à niveau MPI la bibliothèque Intel vers la version 2021 Update 6 (à partir de la version 2021 Update 4). Pour plus d'informations, consultez la [mise à jour 6 des MPI bibliothèques Intel® 2021](#).
- Mettez à niveau NVIDIA le pilote vers la version 470.141.03 (à partir de 470.103.01).
- Mettez à niveau NVIDIA Fabric Manager vers la version 470.141.03 (à partir de 470.103.01).

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers pour le cluster [aws-parallelcluster](#) et les packages sur. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster version 3.3.0 publiée](#)

AWS ParallelCluster version 3.3.0 publiée.

2 novembre 2022

Améliorations :

- Ajoutez la prise en charge de la configuration d'allocation d'instances multiples pour une ressource de calcul lors de l'utilisation en Slurm tant que planificateur. Pour de plus amples informations, veuillez consulter [Allocation de plusieurs types d'instances avec Slurm](#).
- Ajoutez la prise en charge de l'ajout et de la suppression à l'[SharedStorage](#) aide d'une mise à jour du cluster, à l'aide d'une configuration mise à jour. Pour de plus amples informations, veuillez consulter [Stockage partagé](#).
- Ajoutez un nouveau paramètre de configuration [EfsDeletionPolicy](#) des paramètres de stockage [FsxLustre](#) \_partagé pour prendre en charge la rétention du stockage.

- Ajoutez la prise en charge de la Slurm comptabilité avec le nouveau paramètre de configuration [Scheduling SlurmSettings //Database](#). Pour de plus amples informations, veuillez consulter [Slurmcomptabilité avec AWS ParallelCluster](#).
- Ajoutez la prise en charge des réservations de capacité à la demande (ODCR) et des groupes de ressources de réservation de capacité. Pour de plus amples informations, veuillez consulter [Lancez des instances avec des réservations de capacité à la demande \(ODCR\)](#).
- Ajoutez un nouveau paramètre de configuration pour spécifier la IMDS version à prendre en charge dans un cluster ou créez une infrastructure d'images dans le cluster, [Imds/ImdsSupport](#), et créez, [Imds/ImdsSupport](#), des configurations.
- Ajoutez le support pour [Networking /Placement Group](#) dans la [ComputeResources](#) section [SlurmQueues](#) /.

- Ajoutez la prise en charge des instances dotées de plusieurs interfaces réseau limitées à une seule ENI par appareil.
- Améliorez la validation du réseau pour les systèmes de EFS fichiers Amazon externes en cochant le CIDR bloc dans le groupe de sécurité attaché.
- Ajoutez un validateur pour vérifier si les types d'instances configurés prennent en charge les groupes de placement.
- Configurez les NFS threads comme étant min (256, max (8, num\_cores \* 4)) pour garantir une meilleure stabilité et de meilleures performances.
- Déplacez NFS l'installation au moment de la création pour réduire le temps de configuration.
- Activez le chiffrement côté serveur pour le EcrImageBuilder SNS sujet créé lors du déploiement AWS ParallelCluster API et utilisé pour signaler les événements de création d'images docker.

Changements :

- Modifiez le comportement de [SlurmQueues /Networking /PlacementGroup /Enabled](#). Il crée désormais un groupe de placement géré unique pour chaque ressource de calcul au lieu d'un seul groupe de placement géré pour toutes les ressources de calcul.
- Ajoutez la prise en charge de [SlurmQueues Networking PlacementGroup](#) `///Name` comme méthode de dénomination préférée.
- Déplacez les balises du nœud principal du modèle de lancement vers la définition de l'instance pour éviter le remplacement du nœud principal lors des mises à jour des balises.
- Désactivez le multithreading via un script exécuté par le modèle de lancement `cloud-init` et non via `CpuOptions` un script défini dans le modèle de lancement.
- Mettez à niveau Python vers la version 3.9 et NodeJS vers la version 16 dans les ressources Lambda de



API l'infrastructureAPI, du conteneur Docker et du cluster.

- Suppression du support pour Python 3.6 dans `saws-parallelcluster-batch-cli`.
- Passez Slurm à la version 22.05.5 (à partir de 21.08.8-2).
- Mettez à niveau NVIDIA le pilote vers la version 470.141.03 (à partir de 470.129.06).
- Mettez à niveau NVIDIA Fabric Manager vers la version 470.141.03 (à partir de 470.129.06).
- Mettez à niveau le NVIDIA CUDA Toolkit vers la version 11.7.1 (from 11.4.4).
- Mettez à niveau le Python utilisé dans AWS ParallelCluster virtualenvs de 3.7.13 à 3.9.15.
- Mettez à niveau le EFA programme d'installation vers la version 1.18.0.
  - EFA-driver : `efa-1.16.0-1` (pas de changement)
  - Configuration EFA : `() efa-config-1.11-1`

```
from efa-confi
g-1.10-1
```

- Profil EFA : efa-profile-1.5-1 (pas de changement)
- LibFabric-aws :  
libfabric-aws-1.16.0~amzn4.0-1 (de).  
libfabric-aws-1.16.0~amzn2.0-1
- RDMA-Core : rdma-core-41.0-2 (à partir de rdma-core-37.0)
- Ouvert MPI : openmpi40-aws-4.1.4-2 (à partir de openmpi40-aws-4.1.1-2 )
- Passez NICE DCV à la version 2022.1-13300 (à partir de 2022.0-12760 ).
- Activez la suppression du SingleSubnetValidator formulaire Queues.
- Ne remplacez pas DRAIN les nœuds lorsqu'ils sont en COMPLETING état, car Epilog est peut-être toujours en cours d'exécution.

Correctifs de bogue :

- Corrigez la validation du paramètre des filtres dans la AWS ParallelCluster ListClusterLogStre

ams commande pour qu'elle échoue lorsque des filtres incorrects sont transmis.

- Corrige la validation du paramètre [SharedStorage /EfsSettings](#) pour qu'elle échoue lorsqu'elle le FileSystemId est spécifiée avec d'autres [EfsSettings](#) paramètres [SharedStorage /](#). Auparavant, FileSystemId n'était pas inclus.
- Corrigez la mise à jour du cluster lors de la modification de [SharedStorage](#) l'ordre en même temps que d'autres modifications de la configuration.
- Corrigez le UpdateParallelClusterLambda Role AWS ParallelCluster API dans lequel télécharger les journaux CloudWatch.
- Corrigez Cinc qui n'utilisait pas le bundle de certificats CA local lors de l'installation de packages avant l'exécution des livres de recettes.
- Corrigez un problème lors de la mise à niveau d'Ubuntu avec pcluster build-image when Build:UpdateOsPack

```
ages:Enabled:true is
set.
```

- Corrigez l'analyse de la configuration du YAML cluster en échouant sur des clés dupliquées.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers pour le cluster [aws-parallelcluster](#) et les packages sur. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster seule la API référence à la documentation a été ajoutée.](#)

AWS ParallelCluster mise à jour de documentation uniquement

27 octobre 2022

- La version 3 a été ajoutée [AWS ParallelCluster Référence d'API](#) à la documentation.

[AWS ParallelCluster version  
3.2.1 publiée](#)

AWS ParallelCluster version  
3.2.1 publiée.

3 octobre 2022

Améliorations :

- Améliorez la logique d'association des tables de routage de l'hôte aux différentes cartes réseau afin de mieux prendre en charge les EC2 instances Amazon dotées de plusieurs cartes réseauNICs.

Changements :

- Mettez à jour NVIDIA le pilote vers la version 470.141.03.
- Mettez à niveau NVIDIA Fabric Manager vers la version 470.141.03.
- Désactivez les cron tâches man-db et mlocate les tâches qui peuvent avoir un impact négatif sur les performances du nœud.
- Mettez à niveau la MPI bibliothèque Intel vers la version 2021.6.0.602.
- Mettez Python à niveau de la version 3.7.10 vers la version 3.7.13 pour faire face à ce risque de sécurité.

### Correctifs de bogue :

- Évitez d'échouer `DescribeCluster` lorsque la configuration du cluster n'est pas disponible.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers pour le cluster [aws-parallelcluster](#) et les packages sur. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster version  
3.2.0 publiée](#)

AWS ParallelCluster version  
3.2.0 publiée.

27 juillet 2022

Améliorations :

- Ajoutez la prise en charge de la [planification basée sur la mémoire dans Slurm](#).
  - Configurez la mémoire réelle des nœuds de calcul dans la configuration du Slurm cluster.
  - Ajoutez un nouveau paramètre de configuration [Scheduling / SlurmSettings / EnableMemoryBasedScheduling](#) pour activer la planification basée sur la mémoire dans Slurm.
  - Ajoutez un nouveau paramètre de configuration [Scheduling / SlurmQueues / ComputeResources / SchedulableMemory](#) pour remplacer la valeur par défaut de la mémoire vue par le planificateur sur les nœuds de calcul.

- Améliorez la flexibilité des mises à jour de configuration du cluster afin d'éviter autant que possible l'arrêt et le démarrage de l'ensemble du cluster. Ajoutez le nouveau paramètre de configuration [Scheduling /SlurmSettings /QueueUpdateStrategy](#) pour définir la stratégie préférée à utiliser lorsque les nœuds de calcul ont besoin d'une mise à jour de configuration et d'un remplacement.
- Améliorez le mécanisme de basculement sur les ressources de calcul disponibles en cas de problèmes de capacité insuffisante avec les instances AmazonEC2. [Désactivez les nœuds de calcul pendant une durée configurable lorsqu'un lancement de nœud échoue](#) en raison d'une capacité insuffisante.
- Ajoutez la prise en charge du montage de systèmes de [FSx for OpenZFS](#) fichiers [FSx for ONTAP](#) et de systèmes existants.
- Ajoutez la prise en charge du montage de plusieurs



instances de [systèmes de fichiers Amazon Elastic existants](#), [FSx pour LustreONTAP](#), [FSx pour et FSx pour les systèmes de ZFS fichiers ouverts](#).

- Ajoutez la prise en charge du [type FSx de déploiement Lustre Persistent\\_2](#) lors de la création d'un nouveau système de fichiers.
- Demandez à l'utilisateur d'activer EFA les types d'instances pris en charge lors de l'utilisation de l'[pcluster configure \\_assistant](#).
- Ajoutez la prise en charge du redémarrage des nœuds de calcul à l'aide de Slurm.
- Améliorez la gestion des états d'Slurmalimentation afin de prendre également en compte la mise hors tension manuelle des nœuds.
- Installez la version NVIDIA GDRCopy 2.3 dans le produit AMIs pour activer la copie en GPU mémoire à faible latence.

Changements :

- Mettez à niveau le EFA programme d'installation vers la version 1.17.2.
  - EFAchauffeur :  
efa-1.16.0-1
  - EFAconfiguration : efa-config-1.10-1
  - EFAprofil : efa-profile-1.5-1
  - Tissu Libfabric :  
libfabric-aws-1.16.0~amzn2.0-1
  - RDMAnoyau : rdma-core-41.0-2
  - Ouvert MPI : openmpi40-aws-4.1.4-2
- Passez NICE DCV à la version 2022.0-12760.
- Mettez à jour NVIDIA le pilote vers la version 470.129.06.
- Mettez à niveau NVIDIA Fabric Manager vers la version 470.129.06.
- Modifiez les types de EBS volumes par défaut de gp2 à gp3 à la fois dans le volume racine et dans les volumes supplémentaires.
- Modifications apportées FSx aux systèmes de fichiers Lustre créés par AWS ParallelCluster :

- Modifiez le type de déploiement par défaut en `Scratch_2`.
- Changez la version du serveur Lustre en `2.12`.
- Ne nécessite pas que `PlacementGroup /Enabled` soit défini sur `true` lors du passage d'un `PlacementGroup /existingId`.
- Ne permet pas de définir `PlacementGroup /Id` lorsque `PlacementGroup /Enabled` est explicitement défini sur `false`.
- Ajoutez une `parallelcluster:cluster-name` étiquette à toutes les ressources créées par AWS ParallelCluster.
- Ajouter `lambda:ListTags` et `lambda:UntagResource` à `ParallelClusterUserRole` utilisé par la AWS ParallelCluster API pile pour la mise à jour du cluster.
- Limitez IPv6 l'accès IMDS aux utilisateurs root et aux administrateurs du cluster uniquement lorsque le paramètre de configuration

HeadNodeImds//Securedest  
activé.

- Avec une option personnalisée AMI, utilisez la taille du volume AMI racine au lieu de la taille ParallelCluster par défaut de 35 GiB. La valeur peut être modifiée dans le fichier de configuration du cluster.
- Désactivation automatique du parc informatique lorsque le paramètre de configuration `Scheduling //SlurmQueues ComputeResources /SpotPrice` est inférieur au prix minimum requis pour le traitement des demandes ponctuelles.
- Afficher `requested_value` et `current_value` valeurs dans l'ensemble de modifications lors de l'ajout ou de la suppression d'une section lors d'une mise à jour.
- Désactivez le `aws-ubuntu-eni-helper` service, disponible dans Deep Learning AMIs, pour éviter les conflits configure `_nw_interface.sh` lors de la configuration d'instances avec plusieurs cartes réseau.

- Suppression du support pour Python 3.6.
- Définissez MTU la valeur 9001 pour toutes les interfaces réseau lors de la configuration d'instances avec plusieurs cartes réseau.
- Supprimez le point final lors de la configuration du nœud FQDN de calcul.
- Gérez les nœuds statiques dans `POWERING_DOWN` .
- Ne remplace pas le nœud dynamique dans `POWER_DOWN` car les tâches sont peut-être toujours en cours d'exécution.
- Redémarrage `clustermgtd` et `slurmctld` démons au moment de la mise à jour du cluster uniquement lorsque `Scheduling` les paramètres sont mis à jour dans la configuration du cluster.
- Fichiers de mise à jour `slurmctld` et de `slurmd` systemd service.
- Limitez IPv6 l'accès IMDS aux utilisateurs root et aux administrateurs du cluster uniquement lorsque le

paramètre de configuration  
HeadNodeImsds//Securedest  
activé.

- Définissez la Slurm configuration AuthInfo=cred\_expire=70 pour réduire le temps d'attente des tâches en attente avant de recommencer lorsque les nœuds ne sont pas disponibles.
- Mettez à niveau les dépendances des livres de recettes tiers :
  - apt-7.4.2 (à partir d'apt-7.4.0)
  - line-4.5.2 (à partir de line-4.0.1)
  - openssh-2.10.3 (depuis openssh-2.9.1)
  - pyenv-3.5.1 (à partir de pyenv-3.4.2)
  - selinux-6.0.4 (à partir de selinux-3.1.1)
  - yum-7.4.0 (à partir de yum-6.1.1)
  - yum-epel-4.5.0 (depuis yum-epel-4.1.2)

Correctifs de bogue :

- Corrigez le comportement par défaut pour ignorer les étapes AWS ParallelCluster de validation et de

test lors de la création d'une personnalisationAMI.

- Corrigez une fuite dans le descripteur de fichier `compute mgtd` .
- Corrigez un problème de course qui entraînait de manière sporadique la fermeture immédiate des instances lancées, car elles n'étaient pas encore disponibles dans la EC2 `DescribeInstances` réponse.
- Correction de la prise en charge du `DisableSimultaneousMultithreading` paramètre sur les types d'instances avec les processeurs Arm.
- Corrigez l'échec de mise à jour de la AWS ParallelCluster API pile lors de la mise à niveau depuis une version précédente. Ajoutez le modèle de ressource utilisé pour l'`ListImagePipelineImages` action dans le `EcrImageDeletionLambdaRole` .
- Correction de l' AWS ParallelCluster API ajout des autorisations manquantes nécessaires pour importer ou exporter depuis Amazon S3 lors de la création d'un

système de fichiers FSx  
pour Lustre.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers pour le cluster [aws-parallelcluster](#) et les packages sur. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) [GitHub](#)



[AWS ParallelCluster mises à jour uniquement relatives à la documentation depuis le début de l'année](#)

AWS ParallelCluster mises à jour relatives uniquement à la documentation.

6 juillet 2022

Nouvelles sections :

- [Bonnes pratiques : alertes budgétairesV3](#)
- [Bonnes pratiques : déplacement d'un cluster vers une nouvelle version AWS ParallelCluster mineure ou un correctifV3](#)
- [Utilisation des services dans Amazon S3V3](#)
- [Utilisation de instances SpotV3](#)
- [Slurmmode protégé par clusterV3](#)
- [AWS ParallelCluster ressources et balisageV3](#)
- [Tableau de CloudWatch bord AmazonV3](#)
- [Intégration avec Amazon CloudWatch Logs avec Amazon LogsV3](#)
- [Elastic Fabric AdapterV3](#)
- [AWS ParallelCluster Personnalisation de l'AMIV3](#)
- [Lancez des instances avec des réservations de capacité à la demande \(ODCR\)V3](#)

- [Corriger les AMI et remplacer les instances Amazon EC2](#)<sup>V3</sup>
- [Fonctionnement d'AWS ParallelCluster](#)<sup>V3</sup>
- [Configuration du chiffrement du stockage partagé à l'aide d'une AWS KMS clé](#)<sup>V3</sup>
- [Exécution de tâches dans un cluster en mode de files d'attente multiples](#)<sup>V3</sup>
- [Utilisation de l' AWS ParallelCluster API](#)<sup>V3</sup>

Mises à jour des sections :

- [Bonnes pratiques : performances du réseau](#)<sup>V3</sup> : ajout de meilleures pratiques pour l'utilisation d'Elastic Fabric Adaptor.
- [AWS Identity and Access Management autorisations dans AWS ParallelCluster](#)<sup>V3</sup> : Diverses mises à jour et ajouts [Politique AWS ParallelCluster pcluster utilisateur supplémentaire](#) lors de l'utilisation d'Amazon FSx for Lustre.
- [AWS ParallelCluster résolution des problèmes](#)<sup>V3</sup> : Diverses mises à jour.

[AWS ParallelCluster version  
3.1.4 publiée](#)

AWS ParallelCluster version  
3.1.4 publiée.

16 mai 2022

Améliorations :

- Ajoutez une validation pour [PasswordSecretArn](#) que [DirectoryService](#) / échoue si le secret n'existe pas.

Ajoutez le support pour activer JWT l'authentification Slurm.

Changements :

- Passez Slurm à la version 21.08.8-2.
- Construisez Slurm avec JWT du soutien.
- Ne nécessite pas que [PlacementGroup](#) / [Enabled](#) soit défini sur true lors du passage d'un [PlacementGroup](#) / [existentId](#).
- Ajouter `lambda:TagResource` à `ParallelClusterUserRole` utilisé par la ParallelCluster API pile pour la création de clusters et la création d'images.

### Correctifs de bogue :

- Corrigez la possibilité d'exporter les journaux d'un cluster lorsque vous utilisez la `export-cluster-log` commande avec l'`--filters` option.
- Corrigez le point d'entrée AWS Batch Docker pour utiliser un répertoire `/home` partagé pour coordonner l'exécution de tâches parallèles sur plusieurs nœuds.
- Réinitialisez l'adresse du nœud lorsque Slurm vous définit un nœud statique défectueux sur `Down` afin d'éviter de traiter le nœud statique défaillant avec une capacité insuffisante comme un nœud défaillant au démarrage.

Pour plus de détails sur les modifications, consultez les `CHANGELOG` fichiers pour le cluster [aws-parallelcluster](#) et les packages sur [aws-parallelcluster-cookbook](#) [aws-parallelcluster-node](#) GitHub

[AWS ParallelCluster version  
3.1.3 publiée](#)

AWS ParallelCluster version  
3.1.3 publiée.

20 avril 2022

Améliorations :

- Exécutez la création de SSH clés en même temps que la création d'un HOME répertoire, par exemple lors de la SSH connexion, lors du passage à un autre utilisateur et lors de l'exécution d'une commande en tant qu'autre utilisateur.
- Ajoutez la prise en charge des deux FQDN et des noms LDAP distinctifs dans le paramètre de configuration [Directory Service /DomainName](#). Le nouveau validateur vérifie désormais les deux syntaxes.
- Le nouveau `update_directory_service_password.sh` script déployé sur le nœud principal prend en charge la mise à jour manuelle du mot de passe Active Directory dans la SSSD configuration. Le mot de passe est récupéré par le AWS Secrets Manager as à

partir de la configuration du cluster.

- Ajoutez la prise en charge du déploiement de API l'infrastructure dans des environnements sans valeur par défautVPC.

#### Changements :

- Désactivez les C-States plus profonds dans x86\_64, officiels AMIs et AMIs créés par `build-image` commande, pour garantir des performances élevées et une faible latence.
- Mises à jour des packages du système d'exploitation et correctifs de sécurité.
- Modifiez les images de base d'Amazon Linux 2 à utiliser AMIs avec Kernel 5.10.

#### Correctifs de bogue :

- Corrigez la pile d'images `DELETE_FAILED` après la création réussie de l'image, en raison des nouvelles politiques d'EC2Image Builder.
- Corrigez le paramètre de configuration [Directory Service /DomainAdd](#) `r` conversion en SSSD

propriété `ldap_uri` lorsqu'il contient plusieurs adresses de domaine.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers du cluster [aws-parallelcluster](#) et les packages correspondants. [aws-parallelcluster-cookbook](#)  
GitHub

[AWS ParallelCluster version  
3.1.2 publiée](#)

AWS ParallelCluster version  
3.1.2 publiée.

2 mars 2022

Changements :

- Passez Slurm à la version 21.08.6 (à partir de 21.08.5).

Correctifs de bogue :

- Corrigez la mise à jour du /etc/hosts fichier sur les nœuds de calcul lorsqu'un cluster est déployé dans des sous-réseaux sans accès à Internet.
- Corrigez le bootstrap des nœuds de calcul pour attendre l'initialisation des disques éphémères avant de rejoindre le cluster.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers du package [aws-parallelcluster](#) sur GitHub



[AWS ParallelCluster version  
3.1.1 publiée](#)

AWS ParallelCluster version  
3.1.1 publiée.

10 février 2022

- Ajoutez la prise en charge des environnements de clusters à utilisateurs multiples en [intégrant les domaines Active Directory \(AD\)](#) gérés via AWS Directory Service.
- Ajoutez le support pour [UseEc2Hostnames](#) dans le fichier de configuration du cluster. Lorsque ce paramètre est défini sur true, utilisez les noms d'hôte EC2 par défaut d'Amazon (par exemple, ip-1-2-3-4) pour les nœuds de calcul.
- Ajoutez la prise en charge de la création de clusters dans [des sous-réseaux sans accès à Internet](#).
- Ajoutez la prise en charge de plusieurs types d'instances de calcul par file d'attente .
- Ajoutez la prise en charge GPU de la planification avec Slurm des ARM instances à l'aide NVIDIA de cartes.
- Ajoutez des drapeaux abrégés pour `cluster-name (-n), region`

(-r), image-id (-i)  
etcluster-configuration /image-configuration (-c) au.  
AWS ParallelCluster CLI

- Ajout du support pour NEW\_CHANGE\_DELETED l'option FSx pour le [AutoImportPolicy](#) paramètre Lustre.
- Ajoutez une parallelcluster:compute-resource-name balise aux EC2 LaunchTemplates ressources utilisées par les nœuds de calcul.
- Améliorez les groupes de sécurité créés au sein du cluster pour autoriser les connexions entrantes provenant de groupes de sécurité personnalisés lorsque des SecurityGroups paramètres sont spécifiés pour certains nœuds principaux et/ou files d'attente.
- Installez NVIDIA les pilotes et CUDA la bibliothèque pourARM.

#### Changements :

- Passez Slurm à la version 21.08.5 (à partir de20.11.8).

- Mettre à jour Slurm le plugin vers la version 21.08 (à partir de 20.11).
- Passez NICE DCV à la version 2021.3-11591 (à partir de 2021.1-10851).
- Mettez à niveau NVIDIA le pilote vers la version 470.103.01 (à partir de 470.57.02).
- Mettez à niveau NVIDIA Fabric Manager vers la version 470.103.01 (à partir de 470.57.02).
- Passez CUDA à la version 11.4.4 (à partir de 11.4.0).
- [Intel](#) a MPI été mis à jour vers la version 2021 Update 4 (mise à jour à partir de la version 2019 Update 8). Pour plus d'informations, consultez la [mise à jour 4 des MPI bibliothèques Intel® 2021](#).
- Passez PMIx à la version 3.2.3 (à partir de 3.1.5).
- Supprimez le transfert des nœuds de calcul défaillants vers `/home/logs/compute`. Les fichiers journaux des nœuds de calcul sont disponibles dans CloudWatch et dans les

journaux de EC2 la console Amazon.

- Activez le potentiel de suppression `SlurmQueues` et de `ComputeResources` longueur des validateurs.
- Désactivez la mise à jour du package au moment du lancement de l'instance sur Amazon Linux 2.
- Désactivez les métadonnées d'image EC2 `ImageBuilder` améliorées d'Amazon lors de la création d'images AWS ParallelCluster personnalisées.
- Définissez explicitement la `cloud-init` source de données comme EC2 étant. Cela permet de gagner du temps de démarrage pour les plateformes Ubuntu et CentOS.
- Utilisez le nom de la ressource de calcul plutôt que le type d'instance dans le nom du modèle de lancement du parc de calcul.
- Redirigez `stderr` et `stdout` vers le fichier CLI journal pour empêcher le texte indésirable de figurer dans la sortie du `pcluster`. CLI

- Déplacez les recettes de configuration/installation pour séparer les livres de recettes appelés du livre principal. Les points d'entrée existants sont conservés et rétrocompatibles.
- Téléchargez les dépendances de la HPC plate-forme Intel pendant la AMI phase de création afin d'éviter de contacter Internet lors de la création du cluster.
- Ne supprimez pas le nom - de la ressource de calcul lors de la configuration des nœuds Slurm.
- Ne configurez GPUs pas Slurm lorsque NVIDIA le pilote n'est pas installé.
- Corrigez `ecs:ListContainerInstances` l'autorisation dans `BatchUserRole` .
- Corrige l'exportation des journaux de cluster lorsqu'aucun préfixe n'est spécifié, précédemment exporté vers un `None` préfixe.
- Corrige la non-exécution de la restauration en cas d'échec de la mise à jour du cluster.
- Corrigez `ecs:ListContainerInstances`

- instances l'autorisation dans `BatchUserRole` .
- Corrigez le `RootVolume` schéma `HeadNode` en déclenchant une erreur si une valeur non prise en charge `KmsKeyId` est spécifiée.
  - Corrigez les statistiques FSx manquantes d'Amazon à afficher dans le CloudWatch tableau de bord.
  - Corrigez `EfaSecurityGroupValidator` .  
Auparavant, il était susceptible de provoquer de fausses défaillances lorsque des groupes de sécurité personnalisés étaient fournis et EFA activés.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers pour le cluster [aws-parallelcluster](#) et les packages sur. [aws-parallelcluster-cookbookaws-parallelcluster-node](#) GitHub

[AWS ParallelCluster version 3.0.3 publiée](#)

AWS ParallelCluster version 3.0.3 publiée.

17 janvier 2022

- Désactivez `log4j-cve-2021-44228-hotpatch` l'agent (`Log4jHotPatch`) sur Amazon Linux 2 pour éviter une éventuelle dégradation des performances. Pour plus d'informations, consultez l'[annonce d'Amazon Linux Hotpatch pour Apache Log4j](#).

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers pour le cluster [aws-parallelcluster](#) et les packages sur. [aws-parallelcluster-cookbook](#) GitHub

[AWS ParallelCluster version 3.0.2 publiée](#)

AWS ParallelCluster version 3.0.2 publiée.

5 novembre 2021

Mettre à jour le [Elastic Fabric Adapter](#) programme d'installation vers la version 1.14.1

- EFAconfig : efa-config-1.9-1 (à partir de efa-config-1.9 )
- EFAprofil : efa-profile-1.5-1 (de efa-profile-1.5 )
- EFAModule noyau : efa-1.14.2 (à partir de efa-1.13.0 )
- RDMANoyau : rdma-core-37.0 (à partir de rdma-core-35 )
- Libfabric : libfabric-1.13.2 (de libfabric-1.13.0 )
- Ouvert MPI : openmpi40-aws-4.1.1-2 (pas de changement)

GPUDirectRDMA est toujours activé s'il est pris en charge par le type d'instance. L'option [GdrSupport](#) de configuration n'a aucun effet.

Pour plus de détails sur les modifications, consultez les



CHANGELOG fichiers du  
cluster [aws-parallelcluster](#) et  
les packages correspondants.  
[aws-parallelcluster-cookboo](#)  
[kaws-parallelcluster-node](#)  
GitHub

[AWS ParallelCluster version 3.0.1 publiée](#)

AWS ParallelCluster version 3.0.1 publiée.

27 octobre 2021

Outil de migration de configuration de clusters

- Les clients peuvent désormais migrer leurs configurations de cluster du format AWS ParallelCluster version 2 vers le format YAML basé sur la AWS ParallelCluster version 3. Pour de plus amples informations, veuillez consulter [pcluster3-config-converter](#).

Le nœud principal peut être arrêté

- Après l'arrêt du parc informatique, le nœud principal peut être arrêté puis redémarré à l'aide de la EC2 console Amazon ou de la commande [stop-instances](#) AWS CLI.

Région AWS Lecture par défaut à partir d'~/.aws/config un fichier

- Pour la [pcluster](#) commande, si le n' Région

AWS est pas spécifié dans le fichier de configuration, dans l'environnement ou sur la ligne de commande, la valeur par défaut Région AWS spécifiée dans le `region` paramètre de la `[default]` section du `~/.aws/config` fichier est utilisée.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers du cluster [aws-parallelcluster](#) et les packages correspondants. [aws-parallelcluster-cookboo](#)  
[kaws-parallelcluster-node](#)  
GitHub

[AWS ParallelCluster version  
3.0.0 publiée](#)

AWS ParallelCluster version  
3.0.0 publiée.

10 septembre 2021

Support pour la gestion des  
clusters via Amazon API  
Gateway

- Les clients peuvent désormais gérer et déployer des clusters via des HTTP points de terminaison avec Amazon API Gateway. Cela ouvre de nouvelles possibilités pour les flux de travail basés sur des scripts ou des événements.

L'interface de ligne de AWS ParallelCluster commande (CLI) a également été repensée pour être compatible avec cela API et inclut une nouvelle option JSON de sortie. Cette nouvelle fonctionnalité permet aux clients de mettre en œuvre des fonctionnalités de base similaires en utilisant CLI également le.

AMICréation personnalisée  
améliorée

- Les clients ont désormais accès à un processus plus

robuste pour créer et gérer des éléments personnalisés à AMIs l'aide d'EC2Image Builder. La personnalisation AMIs peut désormais être gérée via un fichier de AWS ParallelCluster configuration distinct et peut être créée à l'aide de la [pcluster build-image](#) commande dans l'interface de ligne de AWS ParallelCluster commande.

Pour plus de détails sur les modifications, consultez les CHANGELOG fichiers du cluster [aws-parallelcluster](#) et les packages correspondants. [aws-parallelcluster-cookboo](#)  
[kaws-parallelcluster-node](#)  
GitHub

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.