

Guide de l'utilisateur

AWS CloudShell



AWS CloudShell: Guide de l'utilisateur

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

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

Table of Contents

Qu'est-ce que c'est AWS CloudShell ?	1
Fonctionnalités d' AWS CloudShell	2
AWS Command Line Interface	2
Shell et outils de développement	2
Stockage permanent	2
CloudShell VPCenvironnements	3
Sécurité	3
Options de personnalisation	4
Restauration de session	4
Par où commencer AWS CloudShell ?	4
Tarification pour AWS CloudShell	7
AWS CloudShell Sujets clés	7
FAQs	8
Comment commencer à utiliser AWS CloudShell ?	9
À quoi ai-je besoin pour accéder AWS CloudShell ?	9
Qu'y a-t-il AWS CloudShell sur le Console Toolbar?	9
Comment puis-je me lancer AWS CloudShell sur le Console Toolbar?	9
Lequel Régions AWS est AWS CloudShell disponible en ?	10
Lequel Région AWS est attribué s'il AWS CloudShell n'est pas disponible dans la région sélectionnée lorsque vous lancez CloudShell le Console Toolbar?	10
Dans quels types de coques puis-je utiliser AWS CloudShell ?	10
Quels navigateurs Web puis-je utiliser AWS CloudShell ?	10
Comment créer et gérer mon AWS CloudShell environnement ?	10
Quels navigateurs Web puis-je utiliser lorsque je lance AWS CloudShell le Console Toolbar?	11
Puis-je télécharger des fichiers depuis AWS CloudShell ?	11
Quels logiciels sont préinstallés sur mon environnement shell ?	11
Puis-je installer un logiciel qui n'est pas disponible dans l'environnement shell ?	12
Puis-je restreindre les actions que les utilisateurs peuvent effectuer dans AWS CloudShell ?	12
Comment puis-je déplacer des données depuis mon répertoire personnel si je souhaite modifier l' Région AWS endroit où je les utilise AWS CloudShell ?	12
Puis-je augmenter la limite qui détermine le délai d'expiration AWS CloudShell dû à l'inactivité de l'utilisateur ?	12

Puis-je y accéder AWS CloudShellAWS Console Mobile Application depuis l'écran d'accueil ?	13
Comment puis-je me lancer AWS CloudShell dans le AWS Console Mobile Application ?	13
Puis-je utiliser les touches de modification de mes claviers iOS et Android lorsque je les utilise AWS CloudShell dans le ? AWS Console Mobile Application	13
Puis-je diviser l'affichage des AWS CloudShell onglets en plusieurs onglets sur le AWS Console Mobile Application ?	14
Puis-je accéder AWS CloudShell au Console Toolbar sur un appareil mobile ?	14
Quels sont les frais que je dois CloudShell payer à mon Amazon VPC ?	14
Puis-je créer plus de deux VPC environnements par IAM principal ?	14
Premiers pas	15
Prérequis	15
Table des matières	16
Étape 1 : Connectez-vous à AWS Management Console	16
Étape 2 : Sélectionnez une région AWS CloudShell, lancez et choisissez un shell	19
Étape 3 : Téléchargez un fichier depuis AWS CloudShell	22
Étape 4 : Chargez un fichier sur AWS CloudShell	24
Étape 5 : Supprimer un fichier de AWS CloudShell	25
Étape 6 : Création d'une sauvegarde du répertoire de base	25
Étape 7 : Redémarrer une session shell	27
Étape 8 : Supprimer le répertoire d'accueil d'une session shell	28
Étape 9 : Modifiez le code de votre fichier et exécutez-le à l'aide de la ligne de commande	29
Étape 10 : AWS CLI à utiliser pour ajouter le fichier en tant qu'objet dans un compartiment Amazon S3	31
Rubriques en relation	32
Didacticiels	33
Tutoriel : Copier plusieurs fichiers	33
Chargement et téléchargement de plusieurs fichiers à l'aide d'Amazon S3	34
Chargement et téléchargement de plusieurs fichiers à l'aide de dossiers zippés	38
Tutoriel : Utilisation CodeCommit	39
Prérequis	39
Étape 1 : créer et cloner un CodeCommit dépôt	40
Étape 2 : Préparez et validez un fichier avant de le transférer dans votre CodeCommit dépôt	41
Tutoriel : Création d'une version présignée URLs	42
Prérequis	42

Étape 1 : créer un IAM rôle pour accorder l'accès au compartiment Amazon S3	42
Générez le présigné URL	44
Tutoriel : Création d'un conteneur Docker à l'intérieur AWS CloudShell et transfert vers Amazon ECR	45
Prérequis	46
Procédure du didacticiel	46
Nettoyage	48
Tutoriel : Déploiement d'une fonction Lambda à l'aide du AWS CDK	48
Prérequis	48
Procédure du didacticiel	49
Nettoyage	51
Travailler avec AWS CloudShell	52
Navigation dans l'interface AWS CloudShell	52
.....	52
Travailler dans Régions AWS	54
Spécifier votre valeur par défaut Région AWS pour AWS CLI	55
Utilisation des fichiers et du stockage	56
Utilisation de Docker	56
Fonctionnalités d'accessibilité	58
Navigation au clavier dans CloudShell	58
CloudShell fonctionnalités d'accessibilité du terminal	58
Choix des tailles de police et des thèmes d'interface dans CloudShell	59
Travailler avec les AWS services	60
AWS CLI exemples de ligne de commande pour des AWS services sélectionnés	60
DynamoDB	61
AWS Cloud9	61
Amazon EC2	62
S3 Glacier	62
AWS Elastic Beanstalk CLI	62
Amazon ECS CLI	63
AWS SAM CLI	63
Personnalisation AWS CloudShell	65
Diviser l'affichage de la ligne de commande en plusieurs onglets	65
Modification de la taille de police	66
Modification du thème de l'interface	66
Utilisation du Safe Paste pour du texte multiligne	66

Utilisation tmux pour restaurer une session	67
Utilisation AWS CloudShell dans Amazon Virtual Private Cloud (AmazonVPC)	68
Contraintes d'exploitation	68
Création d'un CloudShell VPC environnement	70
IAMAutorisations requises pour créer et utiliser CloudShell VPC des environnements	71
IAMpolitique accordant un CloudShell accès complet, y compris l'accès à VPC	72
Utilisation de clés de IAM condition pour les VPC environnements	74
Exemples de politiques avec des clés de condition pour VPC les paramètres	75
Sécurité	3
Protection des données	81
Chiffrement des données	82
Gestion de l'identité et des accès	82
Public ciblé	83
Authentification par des identités	84
Gestion des accès à l'aide de politiques	88
Comment AWS CloudShell fonctionne avec IAM	90
Exemples de politiques basées sur l'identité	98
Résolution des problèmes	101
Gestion de AWS CloudShell l'accès et de l'utilisation à l'aide IAM de politiques	103
Journalisation et surveillance	118
Surveillance de l'activité avec CloudTrail	118
AWS CloudShell dans CloudTrail	118
Validation de conformité	121
Résilience	126
Sécurité de l'infrastructure	127
Bonnes pratiques de sécurité	128
Sûreté FAQs	128
Quels AWS processus et technologies sont utilisés lorsque vous lancez CloudShell et démarrez une session shell ?	129
Est-il possible de restreindre l'accès au réseau CloudShell ?	129
Puis-je personnaliser mon CloudShell environnement ?	129
Où est réellement stocké mon \$HOME répertoire dans le AWS Cloud ?	130
Est-il possible de chiffrer mon \$HOME répertoire ?	130
Puis-je lancer une analyse antivirus sur mon \$HOME répertoire ?	130
Puis-je restreindre l'entrée ou la sortie de données pour moi ? CloudShell	130
AWS CloudShell environnement informatique	131

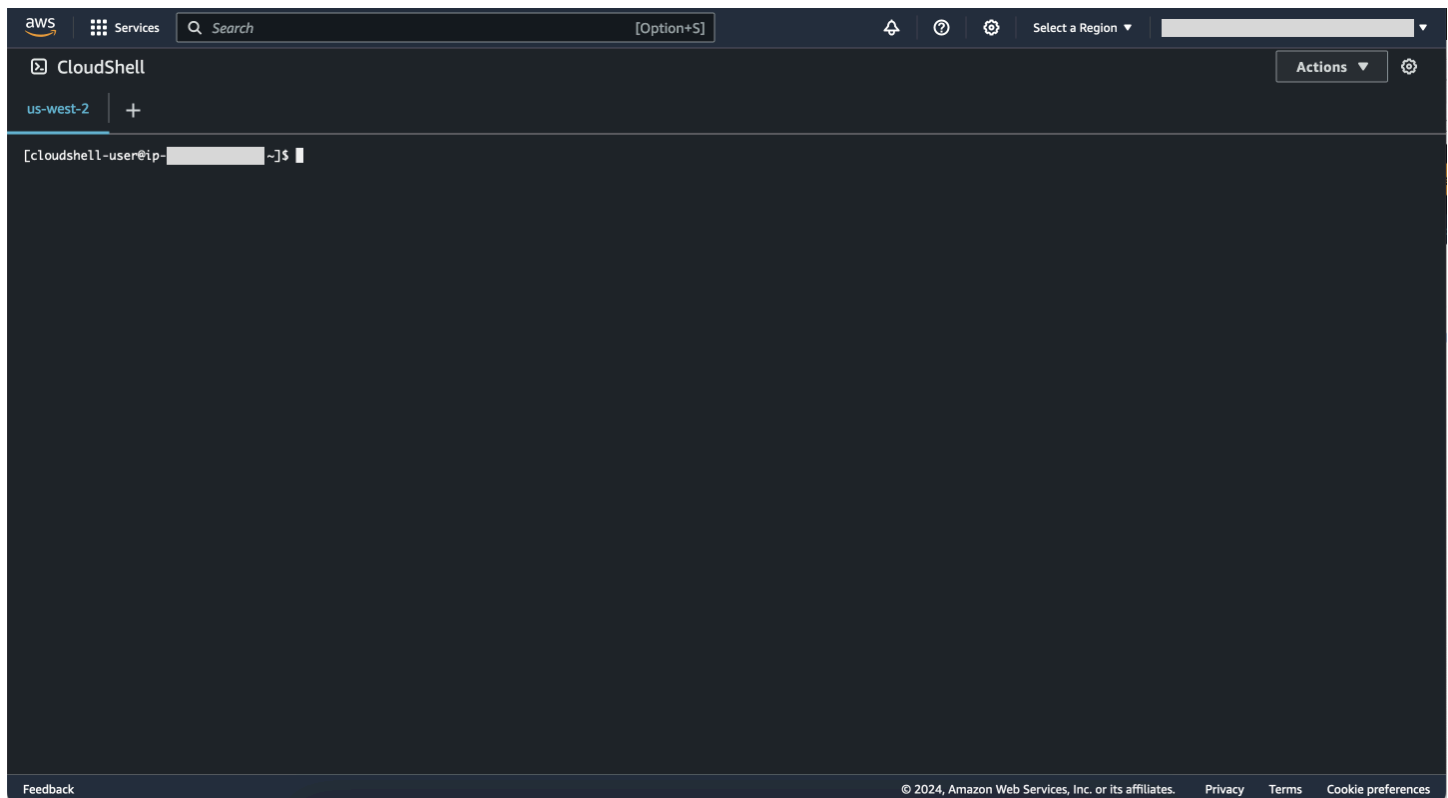
Ressources de l'environnement informatique	131
CloudShell exigences du réseau	131
Logiciel préinstallé	132
Coquillages	133
AWS interfaces de ligne de commande (CLI)	133
Runtimes et AWS SDKs : Node.js et Python 3	137
Outils de développement et utilitaires shell	140
Installation AWS CLI dans votre répertoire personnel	148
Installation de logiciels tiers sur votre environnement shell	150
Modifier votre shell à l'aide de scripts	151
Migration d'Amazon Linux 2 vers Amazon Linux 2023	152
AWS CloudShell Migration FAQs	152
Résolution des problèmes	154
Résolution des erreurs	154
Erreur : « Impossible de démarrer l'environnement. Pour réessayer, actualisez le navigateur ou redémarrez en sélectionnant « Actions, Redémarrer AWS CloudShell ».	155
Erreur : « Impossible de démarrer l'environnement. Vous n'avez pas les autorisations requises. Demandez à votre IAM administrateur d'accorder l'accès à AWS CloudShell »	155
Impossible d'accéder à AWS CloudShell la ligne de commande	155
Impossible d'envoyer un ping aux adresses IP externes	156
Des problèmes sont survenus lors de la préparation de votre terminal	156
Les touches fléchées ne fonctionnent pas correctement dans PowerShell	157
Les Web Sockets non pris en charge empêchent le démarrage des sessions CloudShell	158
Impossible d'importer le <code>AWSPowerShell.NetCore</code> module	159
Docker ne fonctionne pas lors de l'utilisation AWS CloudShell	160
Docker n'a plus d'espace disque	160
<code>docker push</code> le délai imparti est dépassé et continue de réessayer	161
Impossible d'accéder aux ressources VPC depuis mon AWS CloudShell VPC environnement	161
Le ENI fichier utilisé par AWS CloudShell pour mon VPC environnement n'est pas nettoyé .	161
Les utilisateurs <code>CreateEnvironment</code> autorisés uniquement à accéder aux VPC environnements ont également accès aux AWS CloudShell environnements publics	162
Les informations d'identification ne fonctionnent pas dans CloudShell	162
Régions prises en charge	164
GovCloud Régions	165
Quotas et restrictions de service	166

Stockage permanent	166
Utilisation mensuelle	167
Taille de commande	168
Coquilles simultanées	168
Sessions Shell	168
Accès au réseau et transfert de données	168
Restrictions relatives aux fichiers système et aux rechargements de pages	169
Historique de la documentation	170
.....	clxxiv

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

AWS CloudShell est un shell pré-authentifié basé sur un navigateur que vous pouvez lancer directement depuis le. AWS Management Console Vous pouvez y accéder CloudShell AWS Management Console de différentes manières. Pour plus d'informations, consultez [Comment démarrer AWS CloudShell ?](#)

Vous pouvez exécuter des AWS CLI commandes à l'aide de votre interpréteur de commandes préféré, comme Bash PowerShell, ou Z shell. Et vous pouvez le faire sans télécharger ni installer d'outils de ligne de commande.



Lors du lancement AWS CloudShell, un [environnement informatique](#) basé sur Amazon Linux 2023 est créé. [Dans cet environnement, vous pouvez accéder à une vaste gamme d'outils de développement préinstallés, à des options de chargement et de téléchargement de fichiers, ainsi qu'à un stockage de fichiers qui persiste entre les sessions.](#)

(Essayez-le maintenant : [Commencer avec AWS CloudShell](#))

Fonctionnalités d' AWS CloudShell

AWS CloudShell offre les fonctionnalités suivantes :

AWS Command Line Interface

Vous pouvez lancer AWS CloudShell depuis le AWS Management Console. Les AWS informations d'identification que vous avez utilisées pour vous connecter à la console sont automatiquement disponibles dans une nouvelle session shell. Les AWS CloudShell utilisateurs étant pré-authentifiés, il n'est pas nécessaire de configurer les informations d'identification lorsque vous interagissez avec la AWS CLI version Services AWS 2. Le AWS CLI est préinstallé sur l'environnement informatique du shell.

Pour plus d'informations sur l'interaction à Services AWS l'aide de l'interface de ligne de commande, consultez [Travailler avec des AWS services dans AWS CloudShell](#).

Shell et outils de développement

Avec le shell créé pour les AWS CloudShell sessions, vous pouvez passer facilement d'un shell de ligne de commande préféré à un autre. Plus précisément, vous pouvez basculer entre Bash, PowerShell, et Z shell. Vous avez également accès à des outils et utilitaires préinstallés. Il s'agit notamment git, make, pip, sudo, tar, tmux, vim, wget, et zip.

L'environnement shell est préconfiguré avec la prise en charge de plusieurs principaux langages logiciels, tels que Node.js and Python. Cela signifie que, par exemple, vous pouvez exécuter Node.js and Python projets sans effectuer au préalable des installations d'exécution. PowerShell les utilisateurs peuvent utiliser le .NET Core temps d'exécution.

Vous pouvez valider les fichiers créés ou chargés dans un référentiel local avant de transférer ces fichiers vers un référentiel distant géré par AWS CodeCommit. AWS CloudShell

Pour de plus amples informations, veuillez consulter [AWS CloudShell environnement informatique : spécifications et logiciels](#).

Stockage permanent

Avec AWS CloudShell, vous pouvez utiliser jusqu'à 1 Go de stockage persistant dans chacune Région AWS d'elles sans frais supplémentaires. Le stockage permanent se trouve dans votre répertoire personnel (\$HOME) et vous est réservé. Contrairement aux ressources environnementales

éphémères qui sont recyclées après la fin de chaque session du shell, les données de votre répertoire personnel persistent entre les sessions.

Pour plus d'informations sur la conservation des données dans le stockage persistant, consultez [Stockage permanent](#).

Note

CloudShell VPCles environnements ne disposent pas d'un stockage persistant. Le HOME répertoire \$ est supprimé lorsque votre VPC environnement expire (après 20 à 30 minutes d'inactivité), ou lorsque vous supprimez ou redémarrez votre environnement.

CloudShell VPCenvironnements

AWS CloudShell le cloud privé virtuel (VPC) vous permet de créer un CloudShell environnement dans votreVPC. Pour chaque VPC environnement, vous pouvez attribuer un sous-réseauVPC, ajouter un sous-réseau et associer un ou plusieurs groupes de sécurité. AWS CloudShell hérite de la configuration réseau du VPC et vous permet de l'utiliser AWS CloudShell en toute sécurité au sein du même sous-réseau que les autres ressources du. VPC

Sécurité

L' AWS CloudShell environnement et ses utilisateurs sont protégés par des dispositifs de sécurité spécifiques. Cela inclut des fonctionnalités telles que la gestion des IAM autorisations, les restrictions de session shell et le collage sécurisé pour la saisie de texte.

Gestion des autorisations avec IAM

En tant qu'administrateur, vous pouvez accorder ou refuser des autorisations aux AWS CloudShell utilisateurs à l'aide IAM de politiques. Vous pouvez également créer des politiques qui spécifient les actions spécifiques que les utilisateurs peuvent effectuer dans l'environnement shell. Pour de plus amples informations, veuillez consulter [Gestion de AWS CloudShell l'accès et de l'utilisation à l'aide IAM de politiques](#).

Gestion des sessions Shell

Les sessions inactives et de longue durée sont automatiquement arrêtées et recyclées. Pour de plus amples informations, veuillez consulter [Sessions Shell](#).

Coller en toute sécurité pour la saisie de texte

Le collage sécurisé est activé par défaut. Cette fonctionnalité de sécurité nécessite que vous vérifiiez que le texte multiligne que vous souhaitez coller dans le shell ne contient pas de scripts malveillants. Pour de plus amples informations, veuillez consulter [Utilisation du Safe Paste pour du texte multiligne](#).

Options de personnalisation

Vous pouvez personnaliser votre AWS CloudShell expérience selon vos préférences. Par exemple, vous pouvez modifier la mise en page de l'écran (plusieurs onglets), la taille du texte affiché et basculer entre les thèmes d'interface clairs et foncés. Pour de plus amples informations, veuillez consulter [Personnalisation de votre expérience AWS CloudShell](#).

Vous pouvez également étendre votre environnement shell en [installant votre propre logiciel](#) et [en modifiant votre shell à l'aide de scripts](#).

Restauration de session

La fonctionnalité de restauration de session restaure les sessions que vous exécutiez sur un ou plusieurs onglets de navigateur du CloudShell terminal. Si vous actualisez ou rouvrez des onglets de navigateur récemment fermés, cette fonctionnalité reprend la session jusqu'à ce que le shell soit arrêté en raison d'une session inactive. Pour continuer à utiliser votre CloudShell session, appuyez sur n'importe quelle touche de la fenêtre du terminal. Pour plus d'informations sur les sessions Shell, consultez la section [Sessions Shell](#).

La restauration de session restaure également les dernières sorties du terminal et les processus en cours d'exécution dans chaque onglet du terminal.

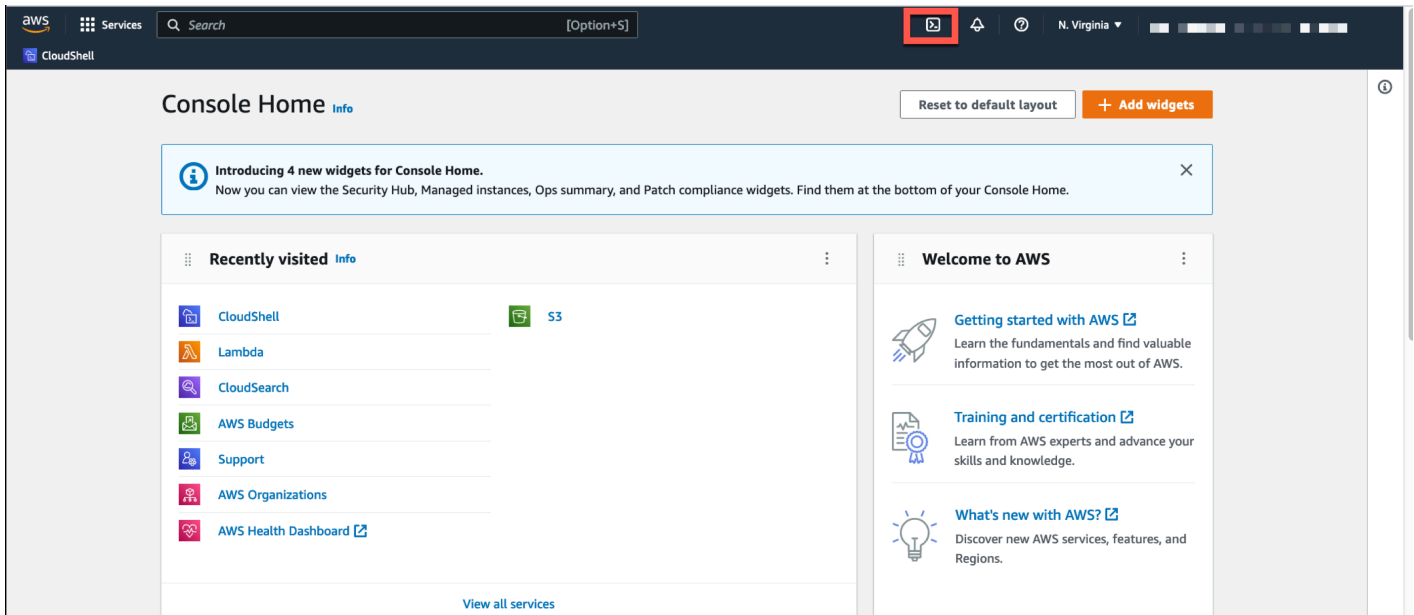
Note

La restauration de session n'est pas disponible dans les applications mobiles.

Par où commencer AWS CloudShell ?

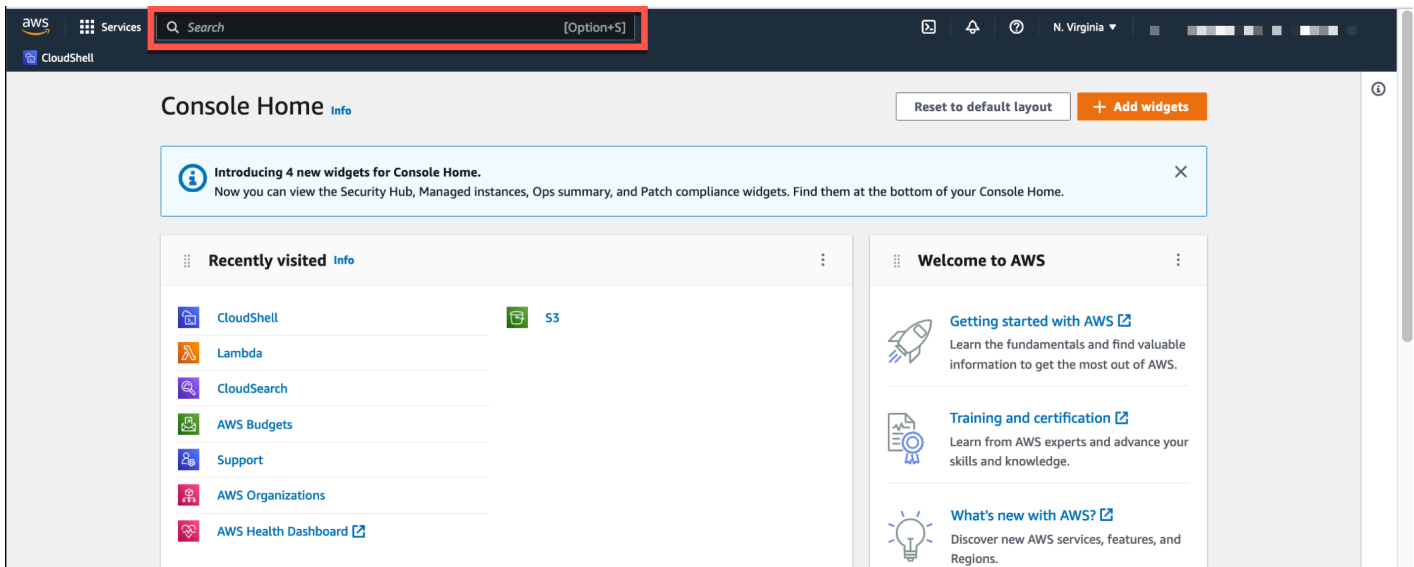
Pour commencer à utiliser le shell, connectez-vous au AWS Management Console et choisissez l'une des options suivantes :

- Dans la barre de navigation, choisissez l'CloudShellicône.



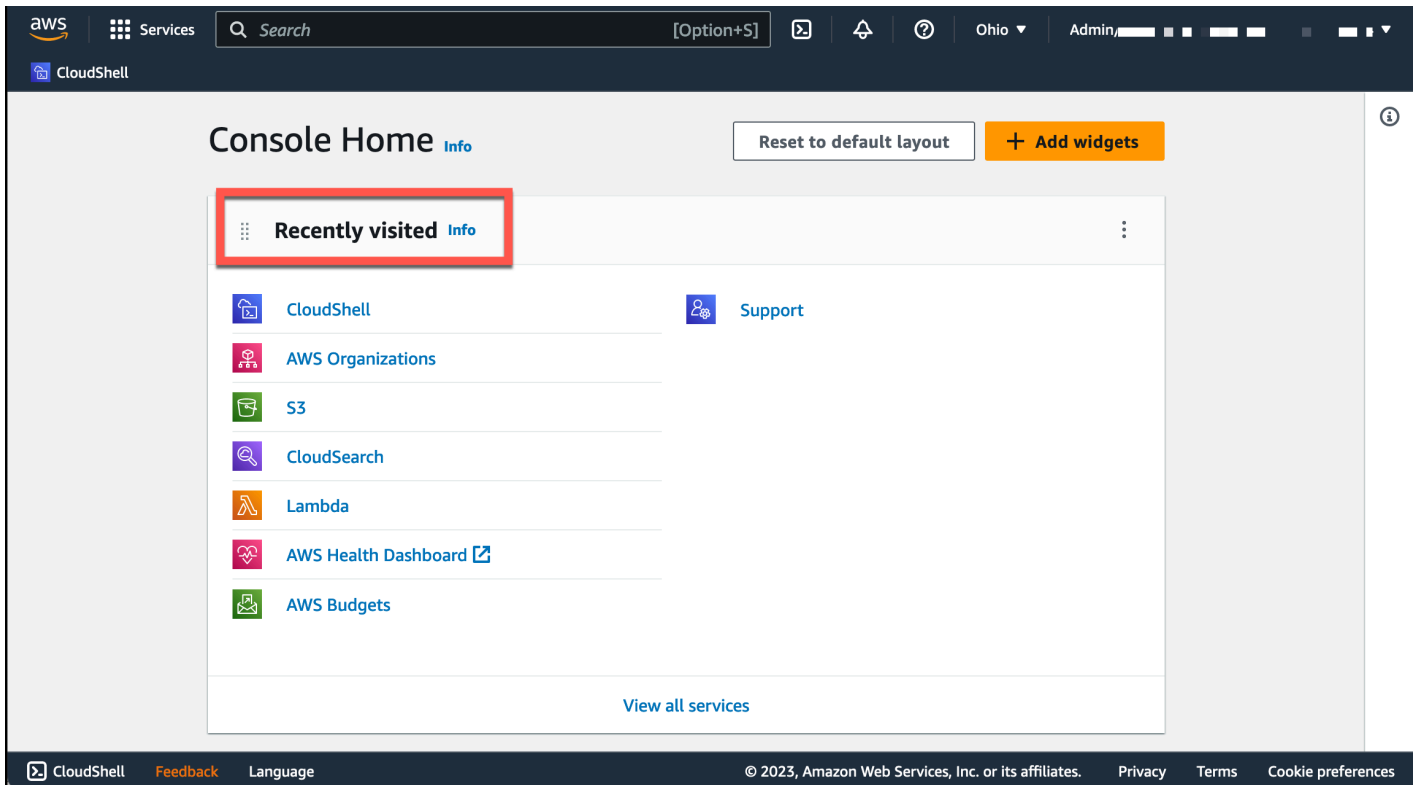
- Dans la zone de recherche, tapez « CloudShell », puis choisissez CloudShell.

Cette étape ouvre votre CloudShell session en plein écran.

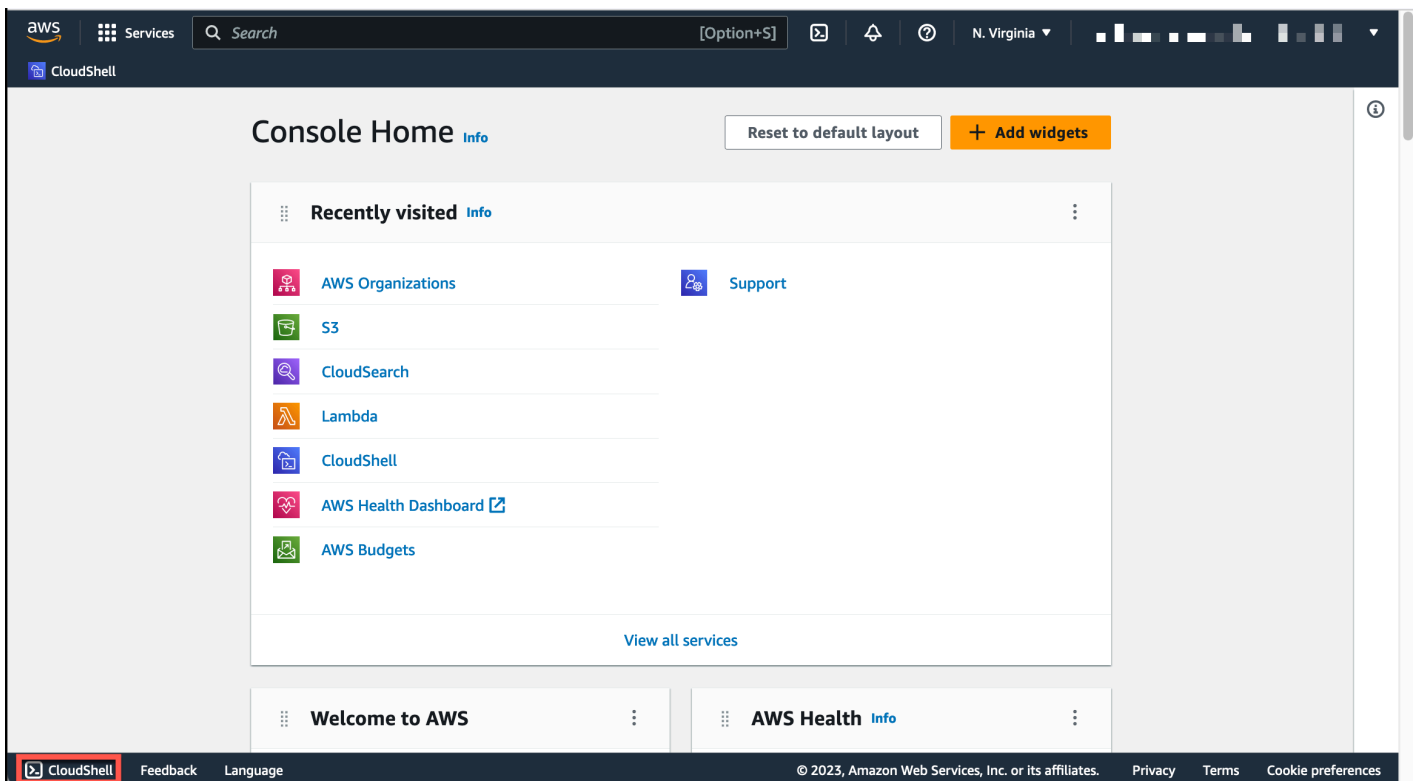


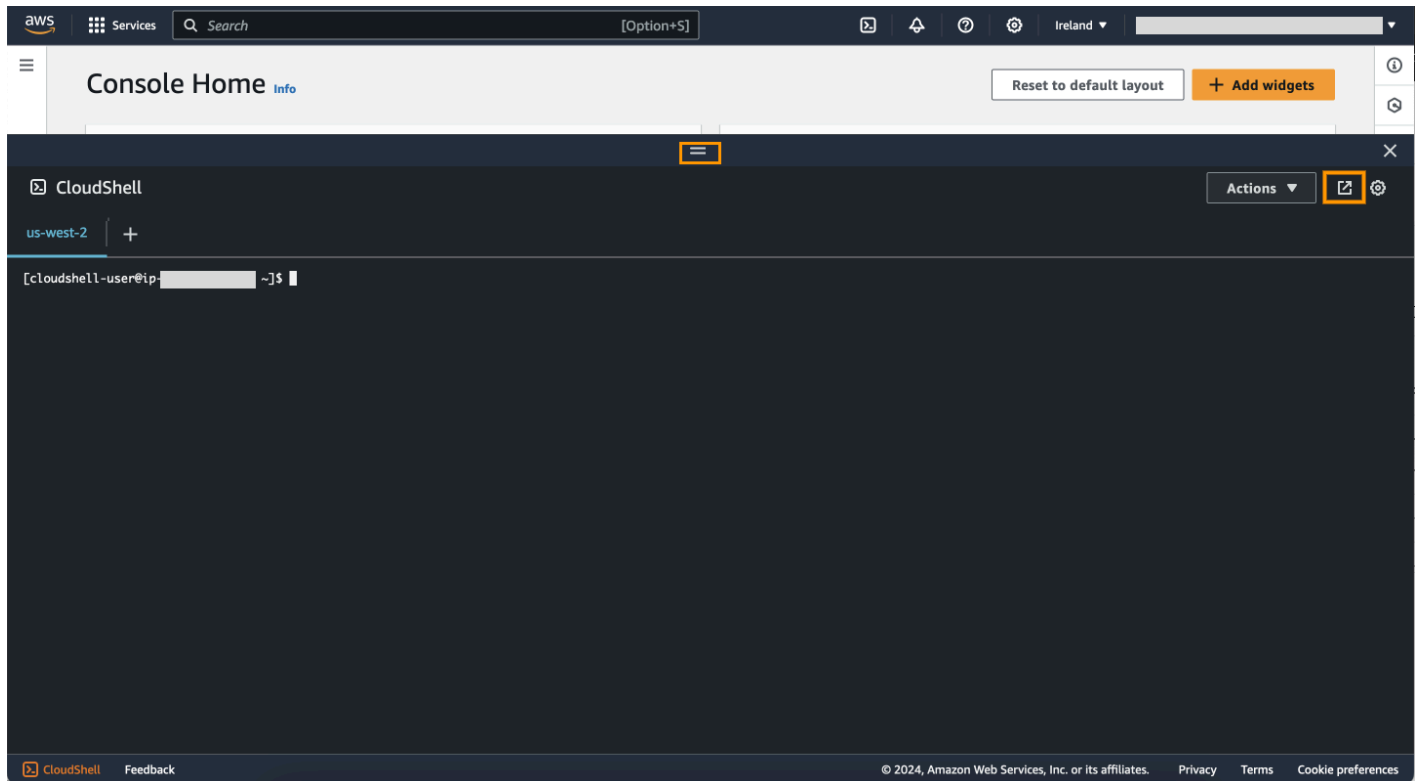
- Dans le widget Visites récentes, sélectionnez CloudShell.

Cette étape ouvre votre CloudShell session en plein écran.



- Choisissez CloudShell sur le Console Toolbar, en bas à gauche de la console. Vous pouvez régler la hauteur de votre CloudShell session en faisant glisser le pointeur. =





Vous pouvez également passer en CloudShell mode plein écran en cliquant sur Ouvrir dans un nouvel onglet du navigateur.

Pour obtenir des instructions sur la façon de se connecter à AWS Management Console et d'effectuer des tâches clés avec AWS CloudShell, voir [Getting started with AWS CloudShell](#).

Tarifcation pour AWS CloudShell

AWS CloudShell Service AWS est disponible sans frais supplémentaires. Cependant, vous payez pour les autres AWS ressources que vous utilisez AWS CloudShell. En outre, les [tarifs de transfert de données standard](#) s'appliquent également. Pour en savoir plus, consultez [Pricing AWS CloudShell](#) (Tarifcation).

Pour de plus amples informations, veuillez consulter [Quotas de service et restrictions pour AWS CloudShell](#).

AWS CloudShell Sujets clés

- [Commencer avec AWS CloudShell](#)

- [Travailler avec AWS CloudShell](#)
- [Travailler avec des AWS services dans AWS CloudShell](#)
- [Personnalisation de votre expérience AWS CloudShell](#)
- [AWS CloudShell environnement informatique : spécifications et logiciels](#)

AWS CloudShell FAQs

Vous trouverez ci-dessous les réponses aux questions les plus fréquemment posées sur AWS CloudShell.

Pour en savoir plus sur la sécurité, voir [AWS CloudShell Sécurité FAQs](#).

- [Comment puis-je commencer à l'utiliser AWS CloudShell ?](#)
- [À quoi ai-je besoin pour accéder AWS CloudShell ?](#)
- [Qu'y a-t-il AWS CloudShell sur le Console Toolbar?](#)
- [Comment puis-je me lancer AWS CloudShell sur le Console Toolbar?](#)
- [Comment créer et gérer mon AWS CloudShell environnement ?](#)
- [Lequel Régions AWS est AWS CloudShell disponible en ?](#)
- [Lequel Région AWS est attribué s'il AWS CloudShell n'est pas disponible dans la région sélectionnée lorsque vous lancez CloudShell le Console Toolbar?](#)
- [Dans quels types de coques puis-je utiliser AWS CloudShell ?](#)
- [Quels navigateurs Web puis-je utiliser AWS CloudShell ?](#)
- [Quels navigateurs Web puis-je utiliser lorsque je lance AWS CloudShell le Console Toolbar?](#)
- [Puis-je télécharger un fichier lorsque je lance AWS CloudShell le Console Toolbar?](#)
- [Quels logiciels sont préinstallés sur mon environnement shell ?](#)
- [Puis-je installer un logiciel qui n'est pas disponible dans l'environnement shell ?](#)
- [Puis-je restreindre les actions que les utilisateurs peuvent effectuer dans AWS CloudShell ?](#)
- [Comment puis-je déplacer des données depuis mon répertoire personnel si je souhaite modifier l' Région AWS endroit où je les utilise AWS CloudShell ?](#)
- [Puis-je augmenter la limite qui détermine le délai d'expiration AWS CloudShell dû à l'inactivité de l'utilisateur ?](#)
- [Puis-je y accéder AWS CloudShellAWS Console Mobile Application depuis l'écran d'accueil ?](#)

- [Comment puis-je me lancer AWS CloudShell dans le AWS Console Mobile Application ?](#)
- [Puis-je utiliser les touches de modification sur mon clavier IOS et sur mon clavier Android lorsque je les utilise AWS CloudShell dans le ? AWS Console Mobile Application](#)
- [Puis-je diviser l'affichage des AWS CloudShell onglets en plusieurs onglets sur le AWS Console Mobile Application ?](#)
- [Puis-je accéder à la AWS CloudShell barre d'outils de la console sur un appareil mobile ?](#)
- [Quels sont les frais que je dois CloudShell payer à mon Amazon VPC ?](#)
- [Puis-je créer plus de deux VPC environnements par IAM principal ?](#)

Comment commencer à utiliser AWS CloudShell ?

Vous pouvez commencer en vous lançant AWS CloudShell en quelques étapes à partir du AWS Management Console. Pour ce faire, connectez-vous à la console à l'aide de vos informations d'identification Compte AWS ou d'identification à la <https://console.aws.amazon.com/console/maison>.

Pour plus d'informations, consultez [Getting started with AWS CloudShell](#).

À quoi ai-je besoin pour accéder AWS CloudShell ?

Comme vous accédez AWS CloudShell depuis le AWS Management Console, vous devez être un IAM utilisateur capable de fournir un alias ou un identifiant de compte, un nom d'utilisateur et un mot de passe valides.

Pour lancer AWS CloudShell sur la console, vous devez disposer des IAM autorisations fournies par la politique ci-jointe. Pour de plus amples informations, veuillez consulter [Gestion de AWS CloudShell l'accès et de l'utilisation à l'aide IAM de politiques](#).

Qu'y a-t-il AWS CloudShell sur le Console Toolbar?

L' CloudShell icône en bas à gauche du AWS Management Console.

Comment puis-je me lancer AWS CloudShell sur le Console Toolbar?

Vous pouvez lancer AWS CloudShell sur le Console Toolbar en choisissant l'CloudShell icône en bas à gauche de la console.

Lequel Régions AWS est AWS CloudShell disponible en ?

Pour obtenir la liste des points de terminaison de service pris en charge Régions AWS et des points de terminaison associés, consultez la [AWS CloudShell page](#) du Référence générale d'Amazon Web Services.

Lequel Région AWS est attribué s'il AWS CloudShell n'est pas disponible dans la région sélectionnée lorsque vous lancez CloudShell le Console Toolbar?

La région par défaut est attribuée à la région la plus proche de la région sélectionnée. Pour plus d'informations, voir [Sélectionner une région AWS CloudShell, lancer et choisir un shell](#).

Vous pouvez exécuter la commande qui fournit des autorisations pour gérer les ressources dans une région différente de la région par défaut. Pour plus d'informations, consultez la section [Travailler dans Régions AWS](#).

Dans quels types de coques puis-je utiliser AWS CloudShell ?

Dans AWS CloudShell, vous pouvez exécuter des commandes à l'aide du Bash shell PowerShell, ou le Z shell. Pour changer de shell, entrez le nom du shell que vous souhaitez utiliser en utilisant le format suivant sur l'invite de commande :

- bash: Utilisez le Bash shell
- pwsh: Utilisation PowerShell
- zsh: Utilisez le Z shell

Quels navigateurs Web puis-je utiliser AWS CloudShell ?

AWS CloudShell prend en charge les versions les plus récentes des navigateurs Google Chrome, Mozilla Firefox, Microsoft Edge et Apple Safari.

Comment créer et gérer mon AWS CloudShell environnement ?

Votre AWS CloudShell environnement est créé et géré par nom IAM d'utilisateur par région. Vous pouvez le vérifier `UserID` en courant `aws sts get-caller-identity`. L'environnement

appartient à l'ID IAM utilisateur de cette région spécifique. Vous pourrez accéder à un autre AWS CloudShell environnement si vous changez de région IAM User Id ou de région.

Quels navigateurs Web puis-je utiliser lorsque je lance AWS CloudShell le Console Toolbar?

Vous pouvez lancer CloudShell sur le Console Toolbar en utilisant les versions les plus récentes des navigateurs Google Chrome, Microsoft Edge, Mozilla Firefox et Apple Safari.

Puis-je télécharger des fichiers depuis AWS CloudShell ?

Oui, vous pouvez télécharger un fichier lorsque vous lancez CloudShell le Console Toolbar ou depuis la page de CloudShell console à l'aide d'un navigateur. Vous pouvez télécharger un fichier à l'aide des versions les plus récentes des navigateurs Google Chrome et Microsoft Edge.

Actuellement, il n'est pas possible de télécharger un fichier à l'aide des navigateurs Mozilla Firefox et Apple Safari.

Note

L'option de téléchargement de fichiers n'est pas disponible pour AWS CloudShell VPC les environnements.

Quels logiciels sont préinstallés sur mon environnement shell ?

Avec le shell créé pour les AWS CloudShell sessions, vous pouvez passer facilement d'un interpréteur de commandes à un autre (Bash, PowerShell, et Z shell). Vous pouvez également avoir accès à des outils et utilitaires préinstallés tels que Make, pip, sudo, tar, tmux, Vim, Wget and Zip.

L'environnement shell est préconfiguré et prend en charge la plupart des principaux langages logiciels. Par exemple, vous pouvez l'utiliser pour exécuter Node.js and Python projets sans avoir à effectuer au préalable des installations d'exécution. PowerShell les utilisateurs peuvent utiliser le .NET Core temps d'exécution.

Vous pouvez ajouter des fichiers créés à l'aide du shell ou chargés via l'interface du shell dans un référentiel contrôlé par version géré à l'aide d'une version préinstallée de git.

Pour de plus amples informations, veuillez consulter [Logiciel préinstallé](#).

Puis-je installer un logiciel qui n'est pas disponible dans l'environnement shell ?

Oui, AWS CloudShell les utilisateurs ont sudo privilèges et peut installer des logiciels à partir de la ligne de commande. Pour de plus amples informations, veuillez consulter [Installation de logiciels tiers sur votre environnement shell](#).

Puis-je restreindre les actions que les utilisateurs peuvent effectuer dans AWS CloudShell ?

Oui, vous pouvez contrôler les actions que les utilisateurs peuvent effectuer dans AWS CloudShell. Par exemple, vous pouvez autoriser les utilisateurs à accéder à des fichiers AWS CloudShell mais les empêcher de charger ou de télécharger des fichiers dans l'environnement shell. Vous pouvez également empêcher complètement les utilisateurs d'y accéder AWS CloudShell. Pour de plus amples informations, veuillez consulter [Gestion de AWS CloudShell l'accès et de l'utilisation à l'aide IAM de politiques](#).

Comment puis-je déplacer des données depuis mon répertoire personnel si je souhaite modifier l' Région AWS endroit où je les utilise AWS CloudShell ?

Pour déplacer vos AWS CloudShell données d'une région Région AWS à une autre, téléchargez d'abord le contenu de votre répertoire personnel d'une région sur votre machine locale, puis téléchargez-le vers le répertoire de base d'une autre région. Pour de plus amples informations, veuillez consulter [???](#).

Note

Les options de chargement et de téléchargement ne sont pas disponibles pour AWS CloudShell VPC les environnements.

Puis-je augmenter la limite qui détermine le délai d'expiration AWS CloudShell dû à l'inactivité de l'utilisateur ?

Votre session shell se termine automatiquement au bout de 20 à 30 minutes environ si vous n'interagissez pas avec votre AWS CloudShell clavier ou votre pointeur. Les processus en cours

ne sont pas considérés comme des interactions. Étant donné qu' CloudShell il est conçu pour des activités ciblées et basées sur des tâches, il n'est actuellement pas prévu d'augmenter ce [délai](#) d'expiration.

Si vous souhaitez effectuer des tâches basées sur un terminal en utilisant des délais Service AWS d'expiration plus flexibles, nous vous recommandons d'utiliser notre solution basée sur le cloud IDE ou de lancer et de vous [connecter à une instance Amazon. AWS Cloud9](#)EC2

Puis-je y accéder AWS CloudShellAWS Console Mobile Application depuis l'écran d'accueil ?

Oui, vous pouvez y accéder AWS CloudShell en vous AWS Console Mobile Application connectant à l'Application Console Mobile. Pour plus d'informations, consultez le [AWS Console Mobile Application guide de l'utilisateur](#).

Comment puis-je me lancer AWS CloudShell dans le AWS Console Mobile Application ?

Vous pouvez lancer AWS CloudShell l'une des méthodes suivantes :

1. Sélectionnez l'AWS CloudShellicône en bas de la barre de navigation.
2. Sélectionnez AWS CloudShelle dans le menu Services.

Note

À l'heure actuelle, il est impossible de créer ou de lancer VPC des environnements dans le AWS Console Mobile Application.

Puis-je utiliser les touches de modification de mes claviers iOS et Android lorsque je les utilise AWS CloudShell dans le ? AWS Console Mobile Application

Oui, vous pouvez utiliser les touches de modification de vos claviers iOS et Android. Pour plus d'informations, consultez le [Guide de l'utilisateur de l'Application AWS Console Mobile](#).

Puis-je diviser l'affichage des AWS CloudShell onglets en plusieurs onglets sur le AWS Console Mobile Application ?

Non, il est actuellement impossible d'exécuter plusieurs AWS CloudShell onglets sur votre application mobile.

Puis-je accéder AWS CloudShell au Console Toolbar sur un appareil mobile ?

Non, vous ne pouvez actuellement pas accéder AWS CloudShell au Console Toolbar sur votre appareil mobile.

Quels sont les frais que je dois CloudShell payer à mon Amazon VPC ?

La connexion à votre compte privé VPC et l'accès aux ressources qu'il contient sont gratuits. Les transferts de données au sein de votre compte privé VPC sont inclus dans votre VPC facturation, et les transferts de données entre vos VPCs canaux CloudShell sont facturés au même prix que vos transferts actuels CloudShell.

Puis-je créer plus de deux VPC environnements par IAM principal ?

Non. Vous ne pouvez créer que deux VPC environnements au maximum.

Commencer avec AWS CloudShell

Ce didacticiel d'introduction explique comment lancer AWS CloudShell et exécuter des tâches clés à l'aide de l'interface de ligne de commande shell.

Tout d'abord, vous vous connectez au AWS Management Console et sélectionnez un Région AWS. Vous lancez CloudShell ensuite une nouvelle fenêtre de navigateur et un type de shell avec lequel vous pouvez travailler.

Ensuite, vous créez un nouveau dossier dans votre répertoire personnel et vous y chargez un fichier depuis votre ordinateur local. Vous travaillez sur ce fichier à l'aide d'un éditeur préinstallé avant de l'exécuter en tant que programme depuis la ligne de commande. Enfin, vous appelez des AWS CLI commandes pour créer un compartiment Amazon S3 et y ajouter votre fichier en tant qu'objet.

Prérequis

IAM autorisations

Vous pouvez obtenir des autorisations pour AWS CloudShell en associant la politique AWS gérée suivante à votre IAM identité (comme un utilisateur, un rôle ou un groupe) :

- `AWSCloudShellFullAccess`: fournit aux utilisateurs un accès complet AWS CloudShell à ses fonctionnalités.

Pour ce didacticiel, vous interagissez également avec Services AWS. Plus précisément, vous interagissez avec Amazon S3 en créant un compartiment S3 et en y ajoutant un objet. Votre IAM identité nécessite une politique qui accorde, au minimum, les `s3:PutObject` autorisations `s3:CreateBucket` et.

Pour plus d'informations, consultez [Amazon S3 Actions](#) dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Fichier d'exercices

Cet exercice implique également le téléchargement et la modification d'un fichier qui est ensuite exécuté en tant que programme à partir de l'interface de ligne de commande. Ouvrez un éditeur de texte sur votre ordinateur local et ajoutez l'extrait de code suivant.

```
import sys
```

```
x=int(sys.argv[1])
y=int(sys.argv[2])
sum=x+y
print("The sum is",sum)
```

Enregistrez le fichier sous le nom `add_prog.py`.

Table des matières

- [Étape 1 : Connectez-vous à AWS Management Console](#)
- [Étape 2 : Sélectionnez une région AWS CloudShell, lancez et choisissez un shell](#)
- [Étape 3 : Téléchargez un fichier depuis AWS CloudShell](#)
- [Étape 4 : Chargez un fichier sur AWS CloudShell](#)
- [Étape 5 : Supprimer un fichier de AWS CloudShell](#)
- [Étape 6 : Création d'une sauvegarde du répertoire de base](#)
- [Étape 7 : Redémarrer une session shell](#)
- [Étape 8 : Supprimer le répertoire d'accueil d'une session shell](#)
- [Étape 9 : Modifiez le code de votre fichier et exécutez-le depuis la ligne de commande](#)
- [Étape 10 : AWS CLI à utiliser pour ajouter le fichier en tant qu'objet dans un compartiment Amazon S3](#)

Étape 1 : Connectez-vous à AWS Management Console

Cette étape consiste à saisir vos informations IAM d'utilisateur pour accéder au AWS Management Console. Si vous êtes déjà dans la console, passez à l'[étape 2](#).

- Vous pouvez y accéder AWS Management Console en utilisant la connexion d'un IAM utilisateur URL ou en accédant à la page de connexion principale.

IAM user sign-in URL

- Ouvrez un navigateur et entrez le nom de connexion URL suivant.
account_alias_or_idRemplacez-le par l'alias de compte ou l'identifiant de compte fourni par votre administrateur.

```
https://account_alias_or_id.signin.aws.amazon.com/console/
```


- Entrez vos informations de IAM connexion et choisissez Se connecter.

Sign in as IAM user

Account ID (12 digits) or account alias

IAM user name

Password

Sign in

[Sign in using root user email](#)

[Forgot password?](#)

Main sign-in page

- Ouverte <https://aws.amazon.com/console/>.
- Si vous ne vous êtes pas connecté auparavant à l'aide de ce navigateur, la page de connexion principale apparaît. Choisissez un IAM utilisateur, entrez l'alias ou l'identifiant du compte, puis choisissez Next.

Sign in

Root user

Account owner that performs tasks requiring unrestricted access. [Learn more](#)

IAM user

User within an account that performs daily tasks. [Learn more](#)

Account ID (12 digits) or account alias

Next

- Si vous vous êtes déjà connecté en tant qu'IAM utilisateur auparavant. Votre navigateur se souvient peut-être de l'alias du compte ou de l'identifiant de compte du Compte AWS. Dans ce cas, entrez vos informations de IAM connexion et choisissez Se connecter.

Sign in as IAM user

Account ID (12 digits) or account alias

account_alias_or_id


IAM user name

Password

Sign in

[Sign in using root user email](#)

[Forgot password?](#)


 Note

Vous pouvez également vous connecter en tant qu'[utilisateur root](#). Cette identité donne un accès complet à toutes Services AWS les ressources du compte. Nous vous recommandons vivement de ne pas utiliser l'utilisateur root pour les tâches quotidiennes, même administratives. Adhérez plutôt à la meilleure pratique qui consiste à utiliser l'utilisateur root uniquement pour créer votre premier IAM utilisateur.

Étape 2 : Sélectionnez une région AWS CloudShell, lancez et choisissez un shell

Au cours de cette étape, vous lancez AWS CloudShell depuis l'interface de la console Région AWS, choisissez un shell disponible et passez à votre shell préféré, tel que Bash PowerShell, ou Z shell.

1. Pour choisir une Région AWS région dans laquelle travailler, allez dans le menu Sélectionnez une région et sélectionnez une [AWS région prise en charge](#) dans laquelle travailler. (Les régions disponibles sont surlignées.)

 Important

Si vous changez de région, l'interface est actualisée et le nom de la région sélectionnée Région AWS s'affiche au-dessus du texte de la ligne de commande. Tous les fichiers que vous ajoutez au stockage persistant ne sont disponibles que dans celui-ci Région AWS. Si vous changez de région, différents types de stockage et de fichiers sont accessibles.

 Important

S'il CloudShell n'est pas disponible dans la région sélectionnée lorsque vous le lancez CloudShell sur le Console Toolbar, dans le coin inférieur gauche de la console, la région par défaut est définie sur la région la plus proche de la région sélectionnée. Vous pouvez exécuter la commande qui fournit des autorisations pour gérer les ressources dans une

région différente de la région par défaut. Pour plus d'informations, consultez la section [Travailler dans Régions AWS](#).

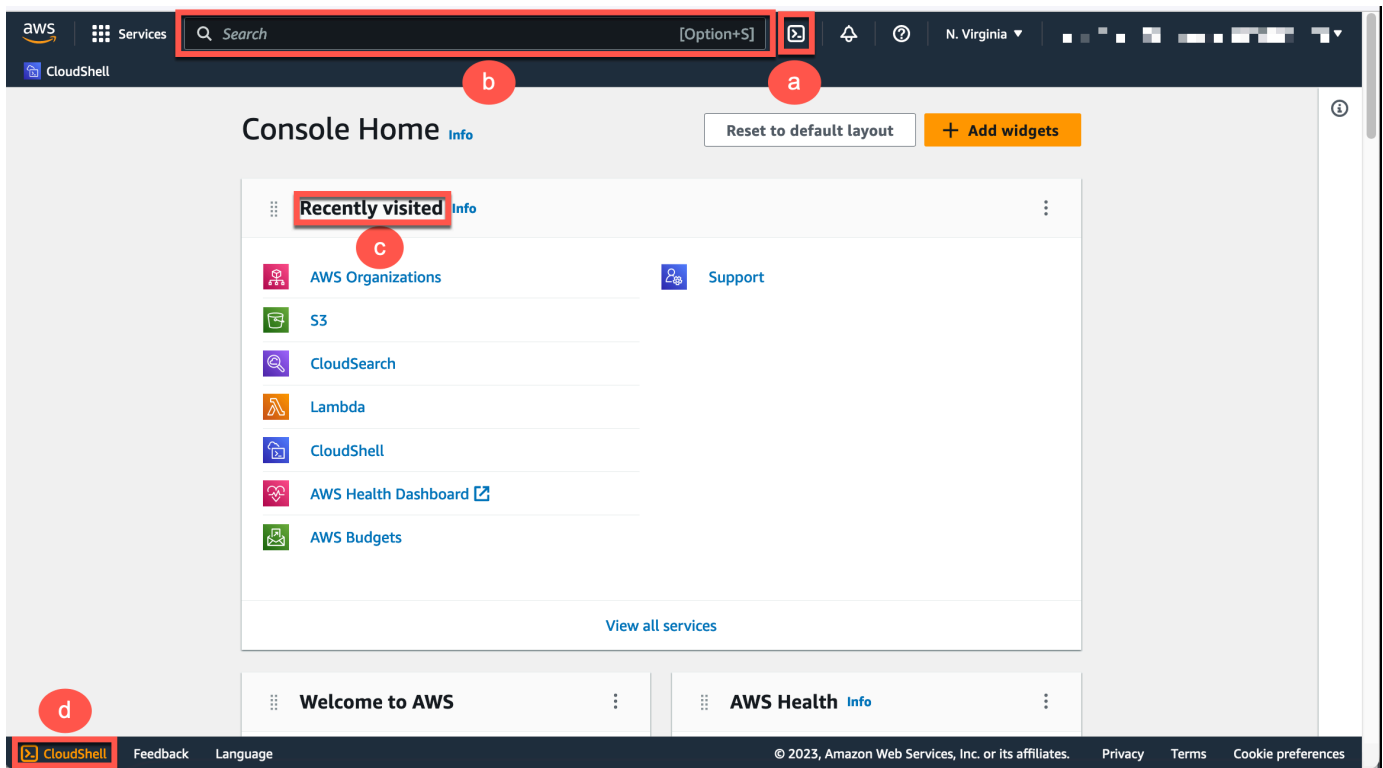
Exemple

Exemple

Si vous choisissez Europe (Espagne) eu-south-2 mais CloudShell n'est pas disponible en Europe (Espagne) eu-south-2, puis la région par défaut est définie sur Europe (Irlande) eu-west-1, la plus proche de l'Europe (Espagne) eu-south-2.

Vous utiliserez les quotas de service pour la région par défaut, Europe (Irlande) eu-west-1 et la même CloudShell session sera rétablie dans toutes les régions. La région par défaut peut être modifiée et vous en serez informé dans la fenêtre du CloudShell navigateur.

2. À partir de AWS Management Console, vous pouvez lancer le CloudShell jeu en choisissant l'une des options suivantes :
 1. Dans la barre de navigation, choisissez l'CloudShellicône.
 2. Dans la zone de recherche, tapez « CloudShell », puis choisissez CloudShell.
 3. Dans le widget Visites récentes, sélectionnez CloudShell.
 4. Choisissez CloudShell sur le Console Toolbar, en bas à gauche de la console.
 - Pour régler la hauteur de votre CloudShell session, faites glisser le pointeur=.
 - Pour passer en CloudShell mode plein écran, cliquez sur l'icône Ouvrir dans un nouvel onglet du navigateur.



Lorsque l'invite de commandes s'affiche, le shell est prêt pour l'interaction.

Note

Si vous rencontrez des problèmes qui vous empêchent de lancer ou d'interagir avec succès AWS CloudShell, consultez les informations permettant d'identifier et de résoudre ces problèmes dans [Résolution des problèmes AWS CloudShell](#).

3. Pour choisir un shell préinstallé avec lequel travailler, entrez le nom du programme sur la ligne de commande.

Bash

```
bash
```

Si vous passez à Bash, le symbole affiché à l'invite de commande prend la valeur\$.

Note

Bash est le shell par défaut qui s'exécute lorsque vous le lancez AWS CloudShell.

PowerShell

`pwsh`

Si vous passez à PowerShell. Le symbole affiché à l'invite de commande devient `PS>`.

Z shell

`zsh`

Si vous passez à Z shell, le symbole affiché à l'invite de commande prend la valeur `%`.

Pour plus d'informations sur les versions préinstallées dans votre environnement shell, consultez le [tableau des shells](#) dans la section relative à l'[environnement de AWS CloudShell calcul](#).

Étape 3 : Téléchargez un fichier depuis AWS CloudShell

Note

Cette option n'est pas disponible pour les VPC environnements.

Cette étape vous guide tout au long du processus de téléchargement d'un fichier.

1. Pour télécharger un fichier, allez dans Actions et choisissez Télécharger le fichier dans le menu.

La boîte de dialogue Télécharger le fichier s'affiche.

2. Dans la boîte de dialogue Télécharger le fichier, entrez le chemin du fichier à télécharger.

Download file



Download files from your AWS CloudShell to your local desktop. Folders are not supported.

Individual file path

You can copy the file path from the command-line and paste it below.

myfile.txt or /folder/myfile.txt.

Cancel

Download

Note

Vous pouvez utiliser des chemins absolus ou relatifs lorsque vous spécifiez un fichier à télécharger. Avec des chemins relatifs, `/home/cloudshell-user/` est ajouté automatiquement au début par défaut. Ainsi, pour télécharger un fichier appelé `mydownload-file`, les deux chemins suivants sont valides :

- Trajectoire absolue : `/home/cloudshell-user/subfolder/mydownloadfile.txt`
- Chemin relatif : `subfolder/mydownloadfile.txt`

3. Choisissez Téléchargement.

Si le chemin du fichier est correct, une boîte de dialogue s'affiche. Vous pouvez utiliser cette boîte de dialogue pour ouvrir le fichier avec l'application par défaut. Vous pouvez également enregistrer le fichier dans un dossier de votre ordinateur local.

Note

L'option de téléchargement n'est pas disponible lorsque vous lancez CloudShell le Console Toolbar. Vous pouvez télécharger un fichier depuis CloudShell la console ou à l'aide du navigateur Web Chrome. Pour plus d'informations sur le téléchargement d'un fichier, voir [Étape 3 : Télécharger un fichier depuis AWS CloudShell](#).

Étape 4 : Chargez un fichier sur AWS CloudShell

Note

Cette option n'est pas disponible pour les VPC environnements.

Cette étape décrit comment télécharger un fichier, puis le déplacer vers un nouveau répertoire de votre répertoire personnel.

1. Pour vérifier votre répertoire de travail actuel, entrez la commande suivante à l'invite :

```
pwd
```

Lorsque vous appuyez sur Entrée, le shell renvoie votre répertoire de travail actuel (par exemple, /home/cloudshell-user).

2. Pour télécharger un fichier dans ce répertoire, allez dans Actions et choisissez Charger le fichier dans le menu.

La boîte de dialogue Télécharger le fichier s'affiche.

3. Choisissez Parcourir.
4. Dans la boîte de dialogue de téléchargement de fichiers de votre système, sélectionnez le fichier texte que vous avez créé pour ce didacticiel (add_prog.py) et choisissez Ouvrir.
5. Dans la boîte de dialogue Charger un fichier, choisissez Charger.

Une barre de progression permet de suivre le téléchargement. Si le téléchargement est réussi, un message confirme qu'il add_prog.py a été ajouté à la racine de votre répertoire personnel.

6. Pour créer un répertoire pour le fichier, entrez la commande make directories :mkdir mysub_dir.
7. Pour déplacer le fichier téléchargé de la racine de votre répertoire personnel vers le nouveau répertoire, utilisez la mv commande suivante :

```
mv add_prog.py mysub_dir.
```

8. Pour remplacer votre répertoire de travail par le nouveau répertoire, entrez cd mysub_dir.

L'invite de commande est mise à jour pour indiquer que vous avez changé de répertoire de travail.

9. Pour afficher le contenu du répertoire en cours `mysub_dir`, entrez la `ls` commande.

Le contenu du répertoire de travail est répertorié. Cela inclut le fichier que vous venez de télécharger.

Étape 5 : Supprimer un fichier de AWS CloudShell

Cette étape décrit comment supprimer un fichier de AWS CloudShell.

1. Pour supprimer un fichier AWS CloudShell, utilisez les commandes shell standard telles que `rm` (remove).

```
rm my-file-for-removal
```

2. Pour supprimer plusieurs fichiers répondant aux critères spécifiés, exécutez la `find` commande.

L'exemple suivant supprime tous les fichiers dont le nom contient le suffixe « `.pdf` ».

```
find -type f -name '*.pdf' -delete
```

Note

Supposons que vous arrêtez de l'utiliser AWS CloudShell dans un domaine spécifique Région AWS. Ensuite, les données qui se trouvent dans le stockage permanent de cette région sont automatiquement supprimées après une période spécifiée. Pour plus d'informations, consultez la section [Stockage persistant](#).

Étape 6 : Création d'une sauvegarde du répertoire de base

Cette étape décrit comment créer une sauvegarde du répertoire de base.

1. Création d'un fichier de sauvegarde

Créez un dossier temporaire en dehors du répertoire de base.

```
HOME_BACKUP_DIR=$(mktemp --directory)
```

Vous pouvez utiliser l'une des options suivantes pour créer une sauvegarde :

a. Création d'un fichier de sauvegarde à l'aide de tar

Pour créer un fichier de sauvegarde à l'aide de tar, entrez la commande suivante :

```
tar \
  --create \
  --gzip \
  --verbose \
  --file=${HOME_BACKUP_DIR}/home.tar.gz \
  [--exclude ${HOME}/.cache] \ // Optional
  ${HOME}/
echo "Home directory backed up to this file: ${HOME_BACKUP_DIR}/home.tar.gz"
```

b. Création d'un fichier de sauvegarde à l'aide de zip

Pour créer un fichier de sauvegarde à l'aide du format zip, entrez la commande suivante :

```
zip \
  --recurse-paths \
  ${HOME_BACKUP_DIR}/home.zip \
  ${HOME} \
  [--exclude ${HOME}/.cache/\*] // Optional
echo "Home directory backed up to this file: ${HOME_BACKUP_DIR}/home.zip"
```

2. Transférez le fichier de sauvegarde à l'extérieur CloudShell

Vous pouvez utiliser l'une des options suivantes pour transférer le fichier de sauvegarde à l'extérieur CloudShell :

a. Téléchargez le fichier de sauvegarde sur votre ordinateur local

Vous pouvez télécharger le fichier créé à l'étape précédente. Pour plus d'informations sur le téléchargement d'un fichier depuis CloudShell, voir [Télécharger un fichier depuis AWS CloudShell](#).

Dans la boîte de dialogue de téléchargement du fichier, entrez le chemin du fichier à télécharger (par exemple, /tmp/tmp.iA99tD9L98/home.tar.gz).


b. Transférez le fichier de sauvegarde vers S3

Pour générer un bucket, entrez la commande suivante :

```
aws s3 mb s3://${BUCKET_NAME}
```

AWSSCLIÀ utiliser pour copier le fichier dans le compartiment S3 :

```
aws s3 cp ${HOME_BACKUP_DIR}/home.tar.gz s3://${BUCKET_NAME}
```

 Note

Des frais de transfert de données peuvent s'appliquer.


3. Backup directement dans un compartiment S3

Pour effectuer une sauvegarde directement dans un compartiment S3, entrez la commande suivante :

```
aws s3 cp \  
  ${HOME}/ \  
  s3://${BUCKET_NAME} \  
  --recursive \  
  [--exclude .cache/\*] // Optional
```

Étape 7 : Redémarrer une session shell

Cette étape décrit comment redémarrer une session shell.

 Note

Par mesure de sécurité, si vous n'interagissez pas avec le shell à l'aide du clavier ou du pointeur pendant une période prolongée, la session s'arrête automatiquement. Les sessions de longue durée sont également automatiquement arrêtées. Pour de plus amples informations, veuillez consulter [Sessions Shell](#).

1. Pour redémarrer une session shell, choisissez Actions, Redémarrer.

Vous êtes informé que le redémarrage AWS CloudShell arrête toutes les sessions actives en cours Région AWS.

2. Pour confirmer, choisissez Redémarrer.

Une interface affiche un message indiquant que l'environnement CloudShell informatique s'arrête. Après l'arrêt et le redémarrage de l'environnement, vous pouvez commencer à travailler avec la ligne de commande dans une nouvelle session.

Note

Dans certains cas, le redémarrage de votre environnement peut prendre quelques minutes.

Étape 8 : Supprimer le répertoire d'accueil d'une session shell

Cette étape décrit comment supprimer une session shell.

Note

Cette option n'est pas disponible pour les VPC environnements. Lorsque vous redémarrez un VPC environnement, son répertoire de base est supprimé.

Warning

La suppression de votre répertoire personnel est une action irréversible dans laquelle toutes les données stockées dans votre répertoire personnel sont supprimées définitivement.


Toutefois, vous pouvez envisager cette option dans les situations suivantes :

- Vous avez modifié un fichier de manière incorrecte et vous ne pouvez pas accéder à l'environnement AWS CloudShell informatique. La suppression de votre répertoire personnel AWS CloudShell rétablit ses paramètres par défaut.
- Vous souhaitez supprimer toutes vos données AWS CloudShell immédiatement. Si vous arrêtez de l'utiliser AWS CloudShell dans une AWS région, le stockage persistant est [automatiquement supprimé à la fin de la période de conservation](#), sauf si vous le AWS CloudShell relancez dans la région.

Si vous avez besoin d'un stockage à long terme pour vos fichiers, pensez à un service tel qu'Amazon S3 ou CodeCommit.

1. Pour supprimer une session shell, choisissez Actions, Supprimer.

Vous êtes informé que la suppression du AWS CloudShell répertoire de base entraîne la suppression de toutes les données actuellement stockées dans votre AWS CloudShell environnement.

 Note

Vous ne pouvez pas annuler cette action.

2. Pour confirmer la suppression, saisissez Supprimer dans le champ de saisie de texte, puis choisissez Supprimer.

AWS CloudShell arrête toutes les sessions actives dans la session en cours Région AWS et crée immédiatement un nouvel environnement.

Quitter manuellement les sessions shell

Avec la ligne de commande, vous pouvez quitter une session shell et vous déconnecter à l'aide de la `exit` commande. Vous pouvez ensuite appuyer sur n'importe quelle touche pour vous reconnecter et continuer à l'utiliser AWS CloudShell.

Étape 9 : Modifiez le code de votre fichier et exécutez-le à l'aide de la ligne de commande

Cette étape montre comment utiliser le préinstallé Vim éditeur pour travailler avec un fichier. Vous exécutez ensuite ce fichier en tant que programme à partir de la ligne de commande.

1. Pour modifier le fichier que vous avez chargé à l'étape précédente, entrez la commande suivante :

```
vim add_prog.py
```

L'interface du shell est actualisée pour afficher le Vim éditeur.

2. Pour modifier le fichier dans Vim, appuyez sur la I touche. Modifiez maintenant le contenu pour que le programme ajoute trois chiffres au lieu de deux.

```
import sys
x=int(sys.argv[1])
y=int(sys.argv[2])
z=int(sys.argv[3])
sum=x+y+z
print("The sum is",sum)
```

Note

Si vous collez le texte dans l'éditeur et que la [fonction de collage sécurisé](#) est activée, un avertissement s'affiche. Le texte multiligne copié peut contenir des scripts malveillants. Grâce à la fonction Safe Paste, vous pouvez vérifier le texte complet avant qu'il ne soit collé. Si vous êtes convaincu que le texte est sécurisé, choisissez Coller.

3. Après avoir modifié le programme, appuyez Esc pour entrer Vim mode commande. Entrez ensuite la :wq commande pour enregistrer le fichier et quittez l'éditeur.

Note

Si vous êtes nouveau dans le Vim mode commande, vous pourriez avoir du mal au début à passer du mode commande au mode insertion. Le mode commande est utilisé lors de l'enregistrement des fichiers et de la sortie de l'application. Le mode Insertion est utilisé lors de l'insertion d'un nouveau texte. Pour passer en mode insertion, appuyez surl, et pour passer en mode commande, appuyez sur Esc. Pour plus d'informations sur Vim et d'autres outils disponibles dans AWS CloudShell, voir [Outils de développement et utilitaires shell](#).

4. Sur l'interface de ligne de commande principale, exécutez le programme suivant et spécifiez trois nombres à saisir. La syntaxe est la suivante.

```
python3 add_prog.py 4 5 6
```

La ligne de commande affiche le résultat du programme :The sum is 15.

Étape 10 : AWS CLI à utiliser pour ajouter le fichier en tant qu'objet dans un compartiment Amazon S3

Au cours de cette étape, vous créez un compartiment Amazon S3, puis vous utilisez la `PutObject` méthode pour ajouter votre fichier de code en tant qu'objet dans ce compartiment.

Note

Dans la plupart des cas, vous pouvez [Utilisation CodeCommit dans AWS CloudShell](#) valider un fichier logiciel dans un référentiel dont la version est contrôlée. Ce didacticiel montre comment vous pouvez utiliser AWS CLI in AWS CloudShell pour interagir avec d'autres AWS services. Avec cette méthode, vous n'avez pas besoin de télécharger ou d'installer de ressource supplémentaire. De plus, comme vous êtes déjà authentifié dans le shell, vous n'avez pas besoin de configurer les informations d'identification avant d'effectuer des appels.

1. Pour créer un bucket dans un compartiment spécifié Région AWS, entrez la commande suivante :

```
aws s3api create-bucket --bucket insert-unique-bucket-name-here --region us-east-1
```

Note

Si vous créez un bucket en dehors de la `us-east-1` région, ajoutez-le `create-bucket-configuration` avec le `LocationConstraint` paramètre pour spécifier la région. Voici un exemple de syntaxe.

```
$ aws s3api create-bucket --bucket my-bucket --region eu-west-1 --create-bucket-configuration LocationConstraint=eu-west-1
```

Si l'appel aboutit, la ligne de commande affiche une réponse du service similaire à la sortie suivante.

```
{
  "Location": "/insert-unique-bucket-name-here"
```

```
}
```

Note

Si vous ne respectez pas les [règles de dénomination des compartiments](#), le message d'erreur suivant s'affiche : Une erreur s'est produite (InvalidBucketName) lors de l'appel de l' CreateBucketopération : Le compartiment spécifié n'est pas valide.

2. Pour télécharger un fichier et l'ajouter en tant qu'objet au bucket que vous venez de créer, appelez la PutObject méthode.

```
aws s3api put-object --bucket insert-unique-bucket-name-here --key add_prog --body  
add_prog.py
```

Une fois l'objet chargé dans le compartiment Amazon S3, la ligne de commande affiche une réponse du service similaire à la sortie suivante :

```
{"ETag": "\"ab123c1:w:wad4a567d8bfd9a1234ebee56\""}
```

ETagIl s'agit du hachage de l'objet qui a été stocké. Vous pouvez utiliser ce hachage pour [vérifier l'intégrité de l'objet chargé sur Amazon S3](#).

Rubriques en relation

- [Travailler avec des AWS services dans AWS CloudShell](#)
- [Copier plusieurs fichiers entre votre machine locale et CloudShell](#)
- [Utilisation CodeCommit dans AWS CloudShell](#)
- [Travailler avec AWS CloudShell](#)
- [Personnalisation de votre expérience AWS CloudShell](#)

AWS CloudShell tutoriels

Les didacticiels suivants vous montrent comment expérimenter et tester différentes fonctionnalités et intégrations lors de leur utilisation AWS CloudShell.

Présentation du didacticiel	En savoir plus
Copier plusieurs fichiers	the section called “Tutoriel : Copier plusieurs fichiers”
En utilisant CodeCommit	???
Création d'une version présignée URLs	???
Création d'un conteneur Docker à l'intérieur AWS CloudShell et transfert vers Amazon ECR	???
Déploiement d'une fonction Lambda à l'aide du AWS CDK	???

Copier plusieurs fichiers entre votre machine locale et CloudShell

Ce didacticiel montre comment copier plusieurs fichiers entre votre machine locale et CloudShell.

À l'aide de l' AWS CloudShell interface, vous pouvez charger ou télécharger un seul fichier à la fois entre votre machine locale et l'environnement shell. Pour copier simultanément plusieurs fichiers entre CloudShell et votre machine locale, utilisez l'une des options suivantes :

- Amazon S3 : utilisez des compartiments S3 comme intermédiaire lorsque vous copiez des fichiers entre votre machine locale et CloudShell.
- Fichiers compressés : compressez plusieurs fichiers dans un seul dossier zippé qui peut être chargé ou téléchargé à l'aide de l' CloudShell interface.

Note

Comme le trafic Internet entrant CloudShell n'est pas autorisé, il n'est actuellement pas possible d'utiliser des commandes telles que `scp` ou `rsync` de copier plusieurs fichiers entre les machines locales et l'environnement CloudShell informatique.

Chargement et téléchargement de plusieurs fichiers à l'aide d'Amazon S3

Cette étape décrit comment charger et télécharger plusieurs fichiers à l'aide d'Amazon S3.

Prérequis

Pour travailler avec des compartiments et des objets, vous avez besoin d'une IAM politique qui accorde les autorisations nécessaires pour effectuer les API actions Amazon S3 suivantes :

- `s3:CreateBucket`
- `s3:PutObject`
- `s3:GetObject`
- `s3:ListBucket`

Pour obtenir la liste complète des actions Amazon S3, consultez la section [Actions](#) du manuel Amazon Simple Storage Service API Reference.

Chargez plusieurs fichiers AWS CloudShell vers Amazon S3

Cette étape décrit comment charger plusieurs fichiers à l'aide d'Amazon S3.

1. Dans AWS CloudShell, créez un compartiment S3 en exécutant la `s3` commande suivante :

```
aws s3api create-bucket --bucket your-bucket-name --region us-east-1
```

Si l'appel aboutit, la ligne de commande affiche une réponse du service S3 :

```
{
  "Location": "/your-bucket-name"
}
```

2. Téléchargez les fichiers dans un répertoire de votre machine locale vers le bucket. Choisissez l'une des options suivantes pour télécharger des fichiers :
 - AWS Management Console: drag-and-drop à utiliser pour télécharger des fichiers et des dossiers dans un bucket.
 - AWS CLI: Lorsque la version de l'outil est installée sur votre machine locale, utilisez la ligne de commande pour télécharger des fichiers et des dossiers dans le compartiment.

Using the console

- Ouvrez la console Amazon S3 à l'adresse <https://s3.console.aws.amazon.com/s3/>.

(Si vous utilisez AWS CloudShell, vous devriez déjà être connecté à la console.)

- Dans le volet de navigation de gauche, choisissez Buckets, puis le nom du bucket dans lequel vous souhaitez télécharger vos dossiers ou fichiers. Vous pouvez également créer un bucket de votre choix en choisissant Create bucket.
- Pour sélectionner les fichiers et les dossiers que vous souhaitez télécharger, choisissez Upload. Ensuite, glissez et déposez les fichiers et dossiers sélectionnés dans la fenêtre de console qui répertorie les objets du compartiment de destination, ou choisissez Ajouter des fichiers ou Ajouter des dossiers.

Les fichiers que vous avez choisis sont répertoriés dans la page Upload (Charger).

- Cochez les cases pour indiquer les fichiers à ajouter.
- Pour ajouter les fichiers sélectionnés au bucket, choisissez Upload.

Note

Pour plus d'informations sur la gamme complète des options de configuration lors de l'utilisation de la console, consultez [Comment télécharger des fichiers et des dossiers dans un compartiment S3 ?](#) dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Using AWS CLI

Note

Pour cette option, l' AWS CLI outil doit être installé sur votre ordinateur local et vos informations d'identification doivent être configurées pour les appels aux AWS services. Pour plus d'informations, consultez le [AWS Command Line Interface Guide de l'utilisateur](#) .

- Lancez l' AWS CLI outil et exécutez la `aws s3` commande suivante pour synchroniser le bucket spécifié avec le contenu du répertoire actuel sur votre machine locale :

```
aws s3 sync folder-path s3://your-bucket-name
```

Si la synchronisation est réussie, des messages de téléchargement sont affichés pour chaque objet ajouté au compartiment.

3. Revenez à la ligne de CloudShell commande et entrez la commande suivante pour synchroniser le répertoire dans l'environnement shell avec le contenu du compartiment S3 :

```
aws s3 sync s3://your-bucket-name folder-path
```

Note

Vous pouvez également ajouter `--exclude "<value>"` des `--include "<value>"` paramètres à la `sync` commande pour effectuer une correspondance de modèles afin d'exclure ou d'inclure un fichier ou un objet en particulier.

Pour plus d'informations, consultez la section [Utilisation des filtres d'exclusion et d'inclusion](#) dans la référence des AWS CLI commandes.

Si la synchronisation est réussie, des messages de téléchargement s'affichent pour chaque fichier téléchargé depuis le bucket vers le répertoire.

Note

Avec la commande `sync`, seuls les fichiers nouveaux et mis à jour sont copiés de manière récursive du répertoire source vers le répertoire de destination.

Téléchargez plusieurs fichiers à l' AWS CloudShell aide d'Amazon S3

Cette étape décrit comment télécharger plusieurs fichiers à l'aide d'Amazon S3.

1. À l'aide de la ligne de commande AWS CloudShell, entrez la `aws s3` commande suivante pour synchroniser un compartiment S3 avec le contenu du répertoire actuel dans l'environnement shell :

```
aws s3 sync folder-path s3://your-bucket-name
```

Note

Vous pouvez également ajouter `--exclude "<value>"` des `--include "<value>"` paramètres à la `sync` commande pour effectuer une correspondance de modèles afin d'exclure ou d'inclure un fichier ou un objet en particulier. Pour plus d'informations, consultez la section [Utilisation des filtres d'exclusion et d'inclusion](#) dans la référence des AWS CLI commandes.

Si la synchronisation est réussie, des messages de téléchargement sont affichés pour chaque objet ajouté au compartiment.

2. Téléchargez le contenu du bucket sur votre machine locale. La console Amazon S3 ne prenant pas en charge le téléchargement de plusieurs objets, vous devez utiliser l' AWS CLI outil installé sur votre machine locale.

À partir de la ligne de commande de l' AWS CLI outil, exécutez la commande suivante :

```
aws s3 sync s3://your-bucket-name folder-path
```

Si la synchronisation est réussie, la ligne de commande affiche un message de téléchargement pour chaque fichier mis à jour ou ajouté dans le répertoire de destination.

Note

Pour cette option, l' AWS CLI outil doit être installé sur votre ordinateur local et vos informations d'identification doivent être configurées pour les appels aux AWS services. Pour plus d'informations, consultez le [AWS Command Line Interface Guide de l'utilisateur](#) .

Chargement et téléchargement de plusieurs fichiers à l'aide de dossiers zippés

Cette étape décrit comment charger et télécharger plusieurs fichiers à l'aide de dossiers compressés.

Avec les utilitaires zip/unzip, vous pouvez compresser plusieurs fichiers dans une archive qui peut être traitée comme un seul fichier. Les utilitaires sont préinstallés dans l'environnement CloudShell informatique.

Pour plus d'informations sur les outils préinstallés, consultez [Outils de développement et utilitaires shell](#).

Chargez plusieurs fichiers dans des AWS CloudShell dossiers zippés

Cette étape décrit comment télécharger plusieurs fichiers à l'aide de dossiers compressés.

1. Sur votre ordinateur local, ajoutez les fichiers à télécharger dans un dossier compressé.
2. Lancez CloudShell, puis choisissez Actions, Télécharger le fichier.
3. Dans la boîte de dialogue Télécharger un fichier, choisissez Sélectionner un fichier, puis choisissez le dossier compressé que vous venez de créer.
4. Dans la boîte de dialogue Upload file, choisissez Upload pour ajouter le fichier sélectionné à l'environnement shell.
5. Dans la ligne de CloudShell commande, exécutez la commande suivante pour décompresser le contenu de l'archive zip dans un répertoire spécifié :

```
unzip zipped-files.zip -d my-unzipped-folder
```

Téléchargez plusieurs fichiers à AWS CloudShell l'aide de dossiers compressés

Cette étape décrit comment télécharger plusieurs fichiers à l'aide de dossiers compressés.

1. Dans la ligne de CloudShell commande, exécutez la commande suivante pour ajouter tous les fichiers du répertoire en cours dans un dossier compressé :

```
zip -r zipped-archive.zip *
```

2. Choisissez Actions, puis Télécharger le fichier.
3. Dans la boîte de dialogue Télécharger le fichier, entrez le chemin du dossier compressé (/home/cloudshell-user/zip-folder/zipped-archive.zippar exemple), puis choisissez Télécharger.

Si le chemin est correct, une boîte de dialogue du navigateur vous permet d'ouvrir le dossier compressé ou de l'enregistrer sur votre ordinateur local.

4. Sur votre machine locale, vous pouvez désormais décompresser le contenu du dossier compressé téléchargé.

Utilisation CodeCommit dans AWS CloudShell

CodeCommit est un service de contrôle de source sécurisé, hautement évolutif et géré qui héberge des référentiels Git privés. En utilisant AWS CloudShell, vous pouvez travailler CodeCommit sur la ligne de commande à l'aide de l'`git-remote-codecommit` utilitaire. Cet utilitaire est préinstallé dans l'environnement AWS CloudShell informatique et fournit une méthode simple pour transférer et extraire du code depuis des CodeCommit référentiels. Pour ce faire, cet utilitaire étend Git. Pour plus d'informations, consultez le [AWS CodeCommit Guide de l'utilisateur](#) .

Ce didacticiel explique comment créer un CodeCommit référentiel et le cloner dans votre environnement AWS CloudShell informatique. Vous apprenez également à préparer et à valider un fichier dans votre référentiel cloné avant de le transférer vers le référentiel distant géré dans le AWS Cloud.

Prérequis

Pour plus d'informations sur les autorisations dont un IAM utilisateur a besoin pour utiliser AWS CloudShell, consultez la [section sur les conditions préalables du didacticiel de mise en route](#). Vous avez également besoin [IAMd'autorisations](#) pour travailler avec CodeCommit.

De plus, avant de commencer, assurez-vous de disposer des éléments suivants :

- Compréhension de base des commandes Git et des concepts de contrôle de version
- Fichier situé dans le répertoire de base de votre shell qui peut être validé dans les référentiels locaux et distants. Dans ce didacticiel, il est appelé `my-git-file`.

Étape 1 : créer et cloner un CodeCommit dépôt

Cette étape décrit comment créer et cloner un CodeCommit dépôt.

1. Dans l'interface de ligne de CloudShell commande, entrez la `codecommit` commande suivante pour créer un CodeCommit référentiel appelé `MyDemoRepo`.

```
aws codecommit create-repository --repository-name MyDemoRepo --repository-
description "My demonstration repository"
```

Si le référentiel est créé avec succès, la ligne de commande affiche la réponse du service.

```
{
  "repositoryMetadata": {
    "accountId": "111122223333",
    "repositoryId": "0dcd29a8-941a-1111-1111-11111111111a",
    "repositoryName": "MyDemoRepo",
    "repositoryDescription": "My demonstration repository",
    "lastModifiedDate": "2020-11-23T20:38:23.068000+00:00",
    "creationDate": "2020-11-23T20:38:23.068000+00:00",
    "cloneUrlHttp": "https://git-codecommit.eu-west-1.amazonaws.com/v1/repos/
MyDemoRepo",
    "cloneUrlSsh": "ssh://git-codecommit.eu-west-1.amazonaws.com/v1/repos/
MyDemoRepo",
    "Arn": "arn:aws:codecommit:eu-west-1:111111111111:MyDemoRepo"
  }
}
```

2. À l'aide de la ligne de commande, créez un nouveau répertoire pour votre dépôt local et faites-en votre répertoire de travail.

```
mkdir my-shell-repo
cd my-shell-repo
```


3. Pour cloner le dépôt distant, utilisez la `git clone` commande. (Pendant que vous travaillez avec `git-remote-codecommit`, utilisez le URL style HTTPS (GRC)).

```
git clone codecommit::eu-west-1://MyDemoRepo
```

Si le dépôt est cloné avec succès, la ligne de commande affiche la réponse du service.

```
Cloning into 'MyDemoRepo'...  
warning: You appear to have cloned an empty repository.
```

4. Pour accéder au référentiel cloné, utilisez la `cd` commande.

```
cd MyDemoRepo
```

Étape 2 : Préparez et validez un fichier avant de le transférer dans votre CodeCommit dépôt

Cette étape décrit comment préparer et valider un fichier avant de le transférer dans votre CodeCommit dépôt.

1. Ajoutez un fichier appelé `MyDemoRepo` dans le dossier `my-git-file` à l'aide d'un éditeur Vim ou de la fonctionnalité de téléchargement de AWS CloudShell. Pour savoir comment utiliser les deux, consultez le [didacticiel de mise en route](#).
2. Pour placer votre fichier dans le dépôt, exécutez la `add` commande git.

```
git add my-git-file
```

3. Pour vérifier que le fichier a été préparé et qu'il est prêt à être validé, exécutez la `status` commande git.

```
git status
```

`my-git-file` est répertorié en tant que nouveau fichier et s'affiche en texte vert, indiquant qu'il est prêt à être validé.

4. Commettez cette version du fichier préparé dans le référentiel.

```
git commit -m "first commit to repo"
```

Note

Si des informations de configuration vous sont demandées pour terminer la validation, utilisez le format suivant.

```
$ git config --global user.name "Jane Doe"  
$ git config --global user.email janedoe@example.com
```

5. Pour synchroniser votre dépôt distant avec les modifications apportées à votre dépôt local, transférez les modifications vers la branche en amont.

```
git push
```

Création d'un objet présigné URL pour Amazon S3 à l'aide de AWS CloudShell

Ce didacticiel explique comment créer un objet présigné URL pour partager un objet Amazon S3 avec d'autres personnes. Étant donné que les propriétaires d'objets spécifient leurs propres informations de sécurité lors du partage, toute personne recevant le présigné URL peut accéder à l'objet pendant une durée limitée.

Prérequis

- Un IAM utilisateur disposant des autorisations d'accès prévues par la `AWSCloudShellFullAccess` politique.
- Pour connaître les IAM autorisations requises pour créer un objet présigné URL, consultez [Partager un objet avec d'autres personnes](#) dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Étape 1 : créer un IAM rôle pour accorder l'accès au compartiment Amazon S3

Cette étape décrit comment créer un IAM rôle pour accorder l'accès au compartiment Amazon S3.

1. Pour obtenir vos IAM informations qui peuvent être partagées, appelez la `get-caller-identity` commande depuis AWS CloudShell.

```
aws sts get-caller-identity
```

Si l'appel aboutit, la ligne de commande affiche une réponse similaire à la suivante.

```
{
  "Account": "123456789012",
  "UserId": "AROAXX0ZUU0TTWDCVIDZ2:redirect_session",
  "Arn": "arn:aws:sts::531421766567:assumed-role/Feder08/redirect_session"
}
```

2. Prenez les informations utilisateur que vous avez obtenues à l'étape précédente et ajoutez-les à un AWS CloudFormation modèle. Ce modèle crée un IAM rôle. Ce rôle accorde à votre collaborateur des autorisations de moindre privilège pour les ressources partagées.

```
Resources:
  CollaboratorRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              AWS: "arn:aws:iam::531421766567:role/Feder08"
            Action: "sts:AssumeRole"
      Description: Role used by my collaborators
      MaxSessionDuration: 7200
  CollaboratorPolicy:
    Type: AWS::IAM::Policy
    Properties:
      PolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Action:
              - 's3:*'
            Resource: 'arn:aws:s3:::<YOUR_BUCKET_FOR_FILE_TRANSFER>'
            Condition:
              StringEquals:
```

```
s3:prefix:
  - "myfolder/*"
PolicyName: S3ReadSpecificFolder
Roles:
  - !Ref CollaboratorRole
Outputs:
  CollaboratorRoleArn:
    Description: Arn for the Collaborator's Role
    Value: !GetAtt CollaboratorRole.Arn
```

3. Enregistrez le AWS CloudFormation modèle dans un fichier nommé `template.yaml`.
4. Utilisez le modèle pour déployer la pile et créer le IAM rôle en appelant la `deploy` commande.

```
aws cloudformation deploy --template-file ./template.yaml --stack-name
CollaboratorRole --capabilities CAPABILITY_IAM
```

Générez le présigné URL

Cette étape décrit comment générer le présigné URL.

1. À l'aide de votre éditeur AWS CloudShell, ajoutez le code suivant. Ce code crée un URL qui fournit aux utilisateurs fédérés un accès direct au AWS Management Console.

```
import urllib, json, sys
import requests
import boto3
import os

def main():
    sts_client = boto3.client('sts')
    assume_role_response = sts_client.assume_role(
        RoleArn=os.environ.get(ROLE_ARN),
        RoleSessionName="collaborator-session"
    )
    credentials = assume_role_response['Credentials']
    url_credentials = {}
    url_credentials['sessionId'] = credentials.get('AccessKeyId')
    url_credentials['sessionKey'] = credentials.get('SecretAccessKey')
    url_credentials['sessionToken'] = credentials.get('SessionToken')
    json_string_with_temp_credentials = json.dumps(url_credentials)
    print(f"json string {json_string_with_temp_credentials}")
```

```
request_parameters = f"?
Action=getSignInToken&Session={urllib.parse.quote(json_string_with_temp_credentials)}"
request_url = "https://signin.aws.amazon.com/federation" + request_parameters
r = requests.get(request_url)
signin_token = json.loads(r.text)
request_parameters = "?Action=login"
request_parameters += "&Issuer=Example.org"
request_parameters += "&Destination=" + urllib.parse.quote("https://us-
west-2.console.aws.amazon.com/cloudshell")
request_parameters += "&SignInToken=" + signin_token["SignInToken"]
request_url = "https://signin.aws.amazon.com/federation" + request_parameters

# Send final URL to stdout
print (request_url)

if __name__ == "__main__":
    main()
```

2. Enregistrez le code dans un fichier appelé `share.py`.
3. Exécutez la commande suivante depuis la ligne de commande pour récupérer le nom de ressource Amazon (ARN) du IAM rôle depuis AWS CloudFormation. Ensuite, utilisez-le dans Python script pour obtenir des informations d'identification de sécurité temporaires.

```
ROLE_ARN=$(aws cloudformation describe-stacks --stack-name CollaboratorRole --query
"Stacks[*].Outputs[?OutputKey=='CollaboratorRoleArn'].OutputValue" --output text)
python3 ./share.py
```

Le script renvoie un URL fichier sur lequel un collaborateur peut cliquer pour y accéder AWS CloudShell . AWS Management Console Le collaborateur a le contrôle total du `myfolder/` dossier dans le compartiment Amazon S3 pendant les 3 600 prochaines secondes (1 heure). Les informations d'identification expirent au bout d'une heure. Passé ce délai, le collaborateur ne pourra plus accéder au bucket.

Création d'un conteneur Docker à l'intérieur CloudShell et transfert vers un référentiel Amazon ECR

Ce didacticiel explique comment définir et créer un conteneur Docker AWS CloudShell et le transférer vers un ECR référentiel Amazon.

Prérequis

- Vous devez disposer des autorisations nécessaires pour créer un ECR référentiel Amazon et le transférer vers celui-ci. Pour plus d'informations sur les référentiels Amazon ECR, consultez les [référentiels ECR privés Amazon](#) dans le guide de l'utilisateur Amazon ECR. Pour plus d'informations sur les autorisations requises pour envoyer des images avec Amazon ECR, consultez la section [IAM Autorisations requises pour envoyer une image](#) dans le guide de l'utilisateur Amazon ECR.

Procédure du didacticiel

Le didacticiel suivant explique comment utiliser l'interface CloudShell pour créer un conteneur Docker et le transférer vers un ECR référentiel Amazon.

1. Créez un nouveau dossier dans votre répertoire personnel.

```
mkdir ~/docker-cli-tutorial
```

2. Accédez au dossier que vous avez créé.

```
cd ~/docker-cli-tutorial
```

3. Créez un Dockerfile vide.

```
touch Dockerfile
```

4. À l'aide d'un éditeur de texte nano Dockerfile, par exemple, ouvrez le fichier et collez-y le contenu suivant.

```
# Dockerfile

# Base this container on the latest Amazon Linux version
FROM public.ecr.aws/amazonlinux/amazonlinux:latest

# Install the cowsay binary
RUN dnf install --assumeyes cowsay

# Default entrypoint binary
ENTRYPOINT [ "cowsay" ]
```

```
# Default argument for the cowsay endpoint
CMD [ "Hello, World!" ]
```

5. Le Dockerfile est maintenant prêt à être créé. Construisez le conteneur en exécutant `docker build`. Marquez le conteneur avec un easy-to-type nom à utiliser dans les futures commandes.

```
docker build --tag test-container .
```

Assurez-vous d'inclure la période de fin (`.`).

6. Vous pouvez maintenant tester le conteneur pour vérifier qu'il fonctionne correctement AWS CloudShell.

```
docker container run test-container
```

7. Maintenant que vous disposez d'un conteneur Docker fonctionnel, vous devez le transférer vers un ECR référentiel Amazon. Si vous possédez déjà un ECR référentiel Amazon, vous pouvez ignorer cette étape.

Exécutez la commande suivante pour créer un ECR référentiel Amazon pour ce didacticiel.

```
ECR_REPO_NAME=docker-tutorial-repo
aws ecr create-repository --repository-name ${ECR_REPO_NAME}
```

8. Après avoir créé le ECR référentiel Amazon, vous pouvez y transférer le conteneur Docker.

Exécutez la commande suivante pour obtenir les informations de ECR connexion Amazon pour Docker.

```
AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)
ECR_URL=${AWS_ACCOUNT_ID}.dkr.ecr.${AWS_REGION}.amazonaws.com
aws ecr get-login-password | docker login --username AWS --password-stdin
${ECR_URL}
```

Note

Si la variable d'`AWS_REGION` environnement n'est pas définie dans votre CloudShell ou si vous souhaitez interagir avec des ressources dans un autre Régions AWS, exécutez la commande suivante :

```
AWS_REGION=<your-desired-region>
```

- Marquez l'image avec le ECR référentiel Amazon cible, puis envoyez-la vers ce référentiel.

```
docker tag test-container ${ECR_URL}/${ECR_REPO_NAME}
docker push ${ECR_URL}/${ECR_REPO_NAME}
```

Si vous rencontrez des erreurs ou des problèmes en essayant de suivre ce didacticiel, consultez la section [Dépannage](#) de ce guide pour obtenir de l'aide.

Nettoyage

Vous avez maintenant déployé avec succès votre conteneur Docker dans votre ECR référentiel Amazon. Pour supprimer de votre AWS CloudShell environnement les fichiers que vous avez créés dans ce didacticiel, exécutez la commande suivante.

- ```
cd ~
rm -rf ~/docker-cli-tutorial
```

- Supprimez le ECR référentiel Amazon.

```
aws ecr delete-repository --force --repository-name ${ECR_REPO_NAME}
```

## Déploiement d'une fonction Lambda à l'aide du AWS CDK AWS CloudShell

Ce didacticiel explique comment déployer une fonction Lambda sur votre compte à l'aide de l' AWS Cloud Development Kit (AWS CDK) entrée. CloudShell

### Prérequis

- Démarrez votre compte pour l'utiliser avec le AWS CDK. Pour plus d'informations sur le démarrage avec AWS CDK, consultez la section [Bootstrapping](#) dans le guide du développeur de la AWS CDK version v2. Si vous n'avez pas démarré le compte, vous pouvez vous connecter. `cdk bootstrap CloudShell`



- Assurez-vous de disposer des autorisations appropriées pour déployer des ressources sur votre compte. Les autorisations d'administrateur sont recommandées.

## Procédure du didacticiel

Le didacticiel suivant explique comment déployer une fonction Lambda basée sur un conteneur Docker à l'aide du in. AWS CDK CloudShell

1. Créez un nouveau dossier dans votre répertoire personnel.

```
mkdir ~/docker-cdk-tutorial
```

2. Accédez au dossier que vous avez créé.

```
cd ~/docker-cdk-tutorial
```

3. Installez les AWS CDK dépendances localement.

```
npm install aws-cdk aws-cdk-lib
```

4. Créez un AWS CDK projet squelette dans le dossier que vous avez créé.

```
touch cdk.json
mkdir lib
touch lib/docker-tutorial.js lib/Dockerfile lib/hello.js
```

5. À l'aide d'un éditeur de texte nano `cdk.json`, par exemple, ouvrez le fichier et collez-y le contenu suivant.

```
{
 "app": "node lib/docker-tutorial.js"
}
```

6. Ouvrez le `lib/docker-tutorial.js` fichier et collez-y le contenu suivant.

```
// this file defines the CDK constructs we want to deploy

const { App, Stack } = require('aws-cdk-lib');
const { DockerImageFunction, DockerImageCode } = require('aws-cdk-lib/aws-lambda');
const path = require('path');
```

```
// create an application
const app = new App();

// define stack
class DockerTutorialStack extends Stack {
 constructor(scope, id, props) {
 super(scope, id, props);

 // define lambda that uses a Docker container
 const dockerfileDir = path.join(__dirname);
 new DockerImageFunction(this, 'DockerTutorialFunction', {
 code: DockerImageCode.fromImageAsset(dockerfileDir),
 functionName: 'DockerTutorialFunction',
 });
 }
}

// instantiate stack
new DockerTutorialStack(app, 'DockerTutorialStack');
```

7. Ouvrez le `lib/Dockerfile` et collez-y le contenu suivant.

```
Use a NodeJS 20.x runtime
FROM public.ecr.aws/lambda/nodejs:20

Copy the function code to the LAMBDA_TASK_ROOT directory
This environment variable is provided by the lambda base image
COPY hello.js ${LAMBDA_TASK_ROOT}

Set the CMD to the function handler
CMD ["hello.handler"]
```

8. Ouvrez le `lib/hello.js` fichier et collez-y le contenu suivant.

```
// define the handler
exports.handler = async (event) => {
 // simply return a friendly success response
 const response = {
 statusCode: 200,
 body: JSON.stringify('Hello, World!'),
 };
 return response;
};
```

```
};
```

- Utilisez le AWS CDK CLI pour synthétiser le projet et déployer les ressources. Vous devez démarrer votre compte.

```
npx cdk synth
npx cdk deploy --require-approval never
```

- Appelez la fonction Lambda pour la confirmer et la vérifier.

```
aws lambda invoke --function-name DockerTutorialFunction out.json
jq . out.json
```

Vous avez maintenant déployé avec succès une fonction Lambda basée sur un conteneur Docker à l'aide du. AWS CDK Pour plus d'informations AWS CDK, consultez le [guide du développeur de la AWS CDK version v2](#). Si vous rencontrez des erreurs ou des problèmes lorsque vous tentez de suivre ce didacticiel, consultez la section [Dépannage](#) de ce guide pour obtenir de l'aide.

## Nettoyage

Vous avez maintenant déployé avec succès une fonction Lambda basée sur un conteneur Docker à l'aide du. AWS CDK Dans le AWS CDK projet, exécutez la commande suivante pour supprimer les ressources associées. Il vous sera demandé de confirmer la suppression.

- ```
npx cdk destroy DockerTutorialStack
```
- Pour supprimer de votre AWS CloudShell environnement les fichiers et les ressources que vous avez créés dans ce didacticiel, exécutez la commande suivante.

```
cd ~  
rm -rf ~/docker-cli-tutorial
```

Travailler avec AWS CloudShell

Cette section décrit comment interagir avec les applications prises en charge AWS CloudShell et exécuter des actions spécifiques avec celles-ci.

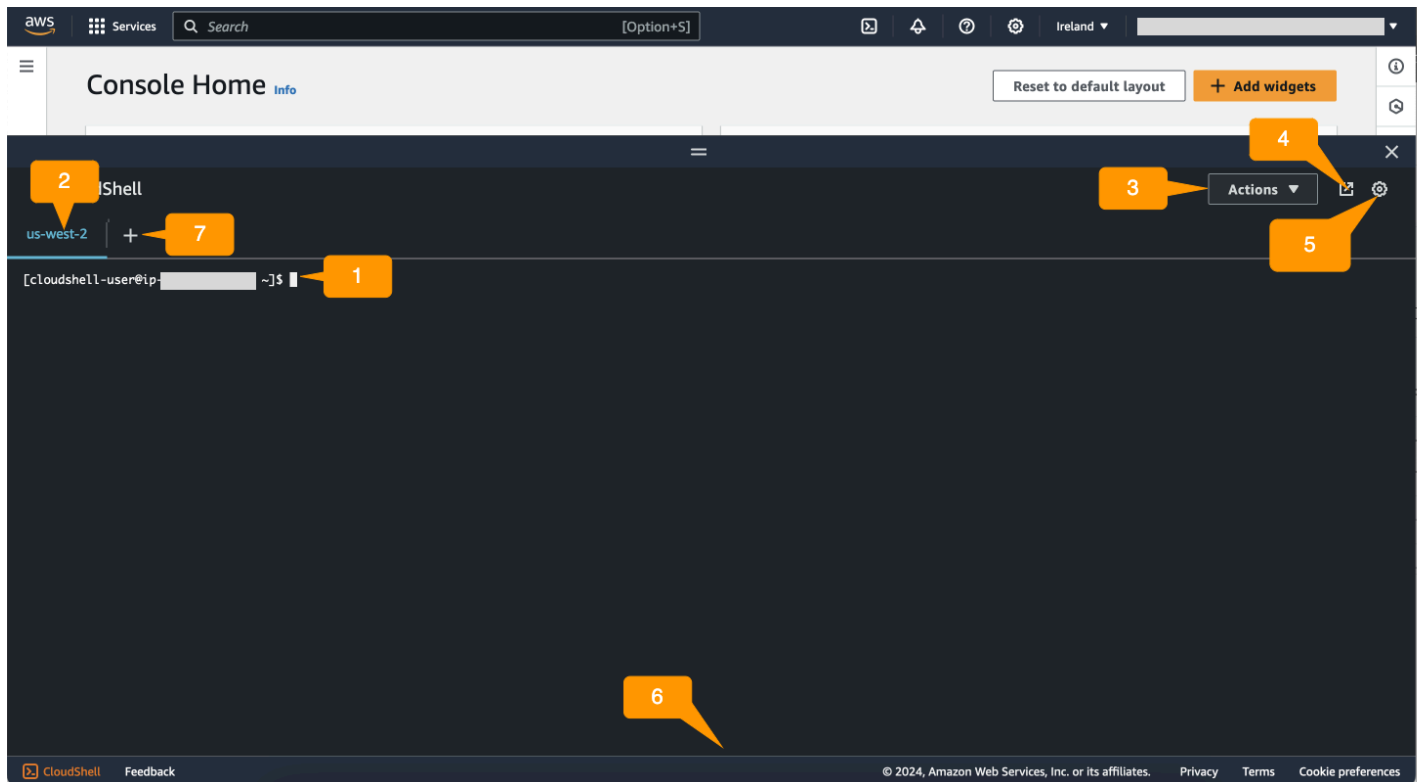
Rubriques

- [Navigation dans l'interface AWS CloudShell](#)
- [Travailler dans Régions AWS](#)
- [Utilisation des fichiers et du stockage](#)
- [Utilisation de Docker](#)


Navigation dans l'interface AWS CloudShell

Vous pouvez parcourir les fonctionnalités de l' CloudShell interface depuis AWS Management Console et Console Toolbar.

La capture d'écran suivante indique plusieurs fonctionnalités clés de AWS CloudShell l'interface.




1. AWS CloudShell interface de ligne de commande que vous utilisez pour exécuter des commandes à l'aide de [votre shell préféré](#). Le type de shell actuel est indiqué par l'invite de commande.
2. L'onglet du terminal, qui utilise l' Région AWS emplacement AWS CloudShell en cours d'exécution.
3. Le menu Actions, qui propose des options permettant de [modifier la disposition de l'écran](#), de [télécharger](#) et de [télécharger](#) des fichiers, de [redémarrer votre répertoire AWS CloudShell personnel](#) et de [supprimer celui-ci AWS CloudShell](#).

 Note

L'option de téléchargement n'est pas disponible lorsque vous lancez CloudShell le Console Toolbar.

4. L'onglet Ouvrir dans un nouveau navigateur, qui permet d'accéder à votre CloudShell session en plein écran.
5. L'option Préférences, que vous pouvez utiliser pour [personnaliser votre expérience shell](#).
6. La barre inférieure, qui propose les options suivantes pour :
 - Lancez CloudShell à partir de CloudShelll'icône.
 - Envoyez des commentaires à l'aide de l'icône Feedback. Choisissez le type de commentaire que vous souhaitez envoyer, ajoutez vos commentaires, puis choisissez Soumettre.
 - Pour envoyer des commentaires pour CloudShell, choisissez l'une des options suivantes :
 - Depuis la console CloudShell, lancez et choisissez Feedback. Ajoutez vos commentaires, puis choisissez Soumettre.
 - Choisissez CloudShellsur le Console Toolbar, dans le coin inférieur gauche de la console, puis choisissez l'icône Ouvrir dans un nouvel onglet du navigateur, Feedback. Ajoutez vos commentaires, puis choisissez Soumettre.

 Note

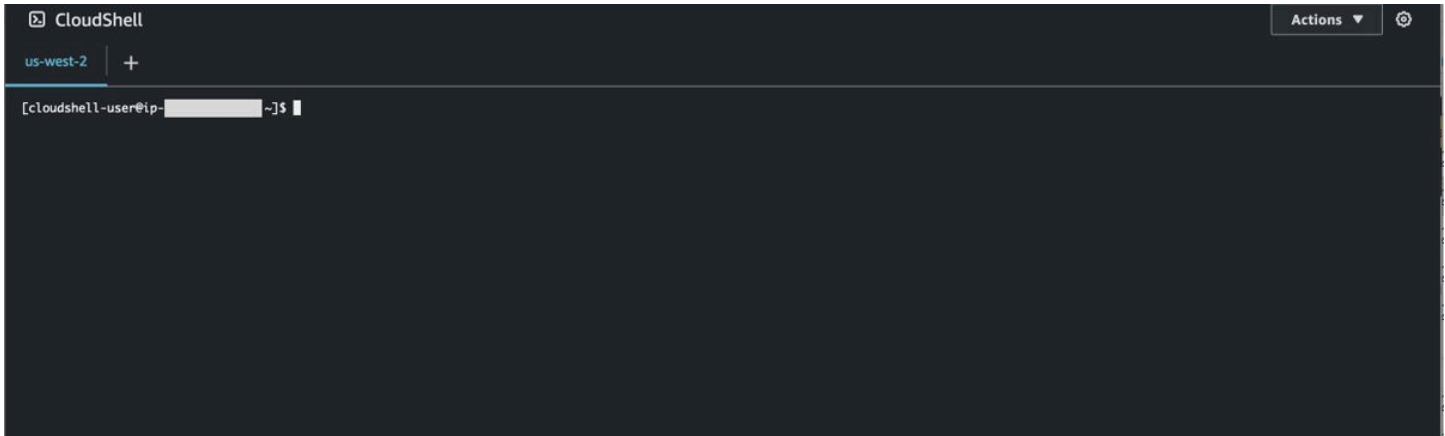
L'option Feedback n'est pas disponible lorsque vous lancez CloudShell le Console Toolbar.

- Découvrez notre politique de confidentialité et nos conditions d'utilisation, et personnalisez vos préférences en matière de cookies.

7. L'icône + est un menu déroulant qui inclut des options permettant de créer, de redémarrer et de supprimer des environnements.

Travailler dans Régions AWS

Le courant dans Région AWS lequel vous êtes en train de courir est affiché sous forme d'onglet.



Vous pouvez choisir une région dans laquelle Région AWS travailler en sélectionnant une région spécifique à l'aide du sélecteur de région. Une fois que vous avez changé de région, l'interface est actualisée au fur et à mesure que votre session shell se connecte à un autre environnement informatique qui s'exécute dans la région sélectionnée.

Important

- Vous pouvez utiliser jusqu'à 1 Go de stockage persistant dans chacune d'elles Région AWS. Le stockage permanent est stocké dans votre répertoire personnel (\$HOME). Cela signifie que tous les fichiers, répertoires, programmes ou scripts personnels stockés dans votre répertoire personnel se trouvent tous dans un seul répertoire Région AWS. De plus, ils sont différents de ceux qui se trouvent dans le répertoire personnel et qui sont stockés dans une région différente.

La conservation à long terme des fichiers dans le stockage persistant est également gérée par région. Pour de plus amples informations, veuillez consulter [Stockage permanent](#).

- Le stockage permanent n'est pas disponible pour AWS CloudShell VPC les environnements.

Spécifier votre valeur par défaut Région AWS pour AWS CLI

Vous pouvez utiliser des [variables d'environnement](#) pour spécifier les options de configuration et les informations d'identification requises pour accéder à Services AWS l'aide de AWS CLI. La variable d'environnement qui spécifie la valeur par défaut Région AWS pour votre session shell est définie soit lorsque vous lancez AWS CloudShell depuis une région spécifique dans le, AWS Management Console soit lorsque vous choisissez une option dans le sélecteur de région.

[Les variables d'environnement ont priorité sur les fichiers AWS CLI d'informations d'identification](#) mis à jour par `aws configure`. Vous ne pouvez donc pas exécuter la `aws configure` commande pour modifier la région spécifiée par la variable d'environnement. Pour modifier la région par défaut pour AWS CLI les commandes, attribuez plutôt une valeur à la variable d'`AWS_REGION` environnement. Dans les exemples suivants, remplacez `us-east-1` par la région dans laquelle vous vous trouvez.

Bash or Zsh

```
$ export AWS_REGION=us-east-1
```

La définition de la variable d'environnement modifie la valeur utilisée jusqu'à la fin de votre session shell ou jusqu'à ce que vous définissiez une valeur différente pour la variable. Vous pouvez définir des variables dans le script de démarrage de votre shell pour les rendre persistantes au cours des futures sessions.

PowerShell

```
PS C:\> $Env:AWS_REGION="us-east-1"
```

Si vous définissez une variable d'environnement à l'invite PowerShell, celle-ci enregistre la valeur uniquement pendant la durée de la session en cours. Vous pouvez également définir la variable pour toutes les PowerShell sessions futures en l'ajoutant à votre PowerShell profil. Pour plus d'informations sur le stockage des variables d'environnement, consultez la [PowerShell documentation](#).

Pour confirmer que vous avez modifié la région par défaut, exécutez la `aws configure list` commande pour afficher les données AWS CLI de configuration actuelles.

Note

Pour des AWS CLI commandes spécifiques, vous pouvez remplacer la région par défaut à l'aide de l'option `--region` de ligne de commande. Pour plus d'informations, consultez la section [Options de ligne de commande](#) dans le Guide de AWS Command Line Interface l'utilisateur.

Utilisation des fichiers et du stockage

À l'aide AWS CloudShell de l'interface, vous pouvez télécharger des fichiers vers et télécharger des fichiers depuis l'environnement shell. Pour plus d'informations sur le téléchargement et le chargement de fichiers, consultez [Getting started with AWS CloudShell](#).

Pour vous assurer que les fichiers que vous ajoutez sont disponibles à la fin de votre session, vous devez connaître la différence entre le stockage permanent et le stockage temporaire.

- Stockage persistant : vous disposez de 1 Go de stockage persistant pour chacun d'entre eux Région AWS. Le stockage permanent se trouve dans votre répertoire personnel.
- Stockage temporaire : le stockage temporaire est recyclé à la fin d'une session. Le stockage temporaire se trouve dans les répertoires situés en dehors de votre répertoire personnel.

Important

Assurez-vous de laisser les fichiers que vous souhaitez conserver et utiliser pour les futures sessions du shell dans votre répertoire personnel. Supposons, par exemple, que vous déplaçiez un fichier hors de votre répertoire personnel en exécutant la `mv` commande. Ce fichier est ensuite recyclé à la fin de la session shell en cours.

Utilisation de Docker

AWS CloudShell prend entièrement en charge Docker sans installation ni configuration. Vous pouvez définir, créer et exécuter des conteneurs Docker à l'intérieur AWS CloudShell. Vous pouvez déployer des ressources basées sur Docker, telles que des fonctions Lambda basées sur des conteneurs Docker, via le AWS CDK Toolkit, ainsi que créer des conteneurs Docker et les transférer vers des

référentiels Amazon via Docker. ECR CLI Pour obtenir des instructions détaillées sur la façon d'exécuter ces deux déploiements, consultez les didacticiels suivants :

- [Tutoriel : Déploiement d'une fonction Lambda à l'aide du AWS CDK](#)
- [Tutoriel : créer un conteneur Docker à l'intérieur AWS CloudShell et le transférer vers un référentiel Amazon ECR](#)

Certaines restrictions et limitations s'appliquent à l'utilisation de Docker avec AWS CloudShell :

- Docker dispose d'un espace limité dans un environnement. Si vous avez de grandes images individuelles ou un trop grand nombre d'images Docker préexistantes, cela peut entraîner des problèmes susceptibles de vous empêcher d'extraire, de créer ou d'exécuter des images supplémentaires. Pour plus d'informations sur Docker, consultez le guide de [documentation Docker](#).
- Docker est disponible dans toutes les AWS régions, à l'exception des régions AWS GovCloud (États-Unis). Pour obtenir la liste des régions dans lesquelles Docker est disponible, consultez la section [AWS Régions prises en charge pour AWS CloudShell](#).
- Si vous rencontrez des problèmes lors de l'utilisation de Docker avec AWS CloudShell, consultez la section [Dépannage](#) de ce guide pour savoir comment résoudre ces problèmes.

Fonctionnalités d'accessibilité pour AWS CloudShell

Cette rubrique décrit comment utiliser les fonctionnalités d'accessibilité pour CloudShell. Vous pouvez utiliser un clavier pour parcourir les éléments focalisables de la page. Vous pouvez également personnaliser l'apparence de CloudShell, notamment les tailles de police et les thèmes d'interface.

Navigation au clavier dans CloudShell

Pour parcourir les éléments focalisables de la page, appuyez sur Tab.

CloudShell fonctionnalités d'accessibilité du terminal

Vous pouvez utiliser la Tab touche dans les modes suivants :

- Mode terminal (par défaut) — Dans ce mode, le terminal capture votre saisie Tab clé. Lorsque le focus est sur le terminal, appuyez sur Tab cette touche pour accéder uniquement aux fonctionnalités du terminal.
- Mode de navigation — Dans ce mode, le terminal ne capture pas la saisie de votre Tab clé. Appuyez sur Tab ce bouton pour parcourir les éléments focalisables de la page.

Pour passer du mode terminal au mode navigation, appuyez sur Ctrl +M. Lorsque vous revenez en arrière, Tab : navigation apparaît dans l'en-tête et vous pouvez utiliser la Tab touche pour parcourir la page.

Pour revenir en mode terminal, appuyez sur Ctrl +M. Vous pouvez également choisir à X côté de Tab : navigation.

Note

À l'heure actuelle, les fonctionnalités d'accessibilité des CloudShell terminaux ne sont pas disponibles sur les appareils mobiles.

Choix des tailles de police et des thèmes d'interface dans CloudShell

Vous pouvez personnaliser l'apparence de CloudShell pour l'adapter à vos préférences visuelles.

- Taille de police — Choisissez entre les tailles de police les plus petites, petites, moyennes, grandes et les plus grandes dans le terminal. Pour plus d'informations sur la modification de la taille de police, consultez [the section called “Modification de la taille de police”](#).
- Thème — Choisissez entre les thèmes d'interface clairs et foncés. Pour plus d'informations sur la modification du thème de l'interface, consultez [the section called “Modification du thème de l'interface”](#).

Travailler avec des AWS services dans AWS CloudShell

L'un des principaux avantages AWS CloudShell est que vous pouvez l'utiliser pour gérer vos AWS services à partir de l'interface de ligne de commande. Cela signifie que vous n'avez pas besoin de télécharger et d'installer des outils ou de configurer vos informations d'identification localement au préalable. Lorsque vous lancez AWS CloudShell, un environnement de calcul est créé sur lequel les outils de ligne de commande AWS suivants sont déjà installés :

- [AWS CLI](#)
- [AWS Elastic Beanstalk CLI](#)
- [Amazon ECS CLI](#)
- [AWS SAM](#)

Et comme vous êtes déjà connecté AWS, il n'est pas nécessaire de configurer vos informations d'identification localement avant d'utiliser les services. Les informations d'identification que vous avez utilisées pour vous connecter AWS Management Console sont transmises à AWS CloudShell.

Si vous souhaitez modifier la AWS région par défaut utilisée pour AWS CLI, vous pouvez modifier la valeur attribuée à la variable d'environnement `AWS_REGION`. (Pour plus d'informations, consultez [Spécifier votre valeur par défaut Région AWS pour AWS CLI](#).)

Le reste de cette rubrique explique comment vous pouvez commencer AWS CloudShell à utiliser pour interagir avec des AWS services sélectionnés à partir de la ligne de commande.

AWS CLI exemples de ligne de commande pour des AWS services sélectionnés

Les exemples suivants ne représentent que certains des nombreux AWS services que vous pouvez utiliser à l'aide des commandes disponibles à partir de AWS CLI la version 2. Pour une liste complète, consultez la [référence des AWS CLI commandes](#).

- [DynamoDB](#)
- [AWS Cloud9](#)
- [Amazon EC2](#)
- [S3 Glacier](#)

DynamoDB

DynamoDB est un service SQL sans base de données entièrement géré qui fournit des performances rapides et prévisibles ainsi qu'une évolutivité sans faille. L'implémentation du SQL mode Non par ce service prend en charge les structures de données clé-valeur et de document.

La `create-table` commande suivante crée une table de SQL style Non nommée `MusicCollection` dans votre AWS compte.

```
aws dynamodb create-table \  
  --table-name MusicCollection \  
  --attribute-definitions AttributeName=Artist,AttributeType=S \  
  AttributeName=SongTitle,AttributeType=S \  
  --key-schema AttributeName=Artist,KeyType=HASH \  
  AttributeName=SongTitle,KeyType=RANGE \  
  --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5 \  
  --tags Key=Owner,Value=blueTeam
```

Pour plus d'informations, reportez-vous [à la section Utilisation de DynamoDB dans AWS CLI/AWS Command Line Interface le guide de l'utilisateur](#).

AWS Cloud9

AWS Cloud9 est un environnement de développement intégré basé sur le cloud (IDE) que vous pouvez utiliser pour écrire, exécuter et déboguer votre code dans une fenêtre de navigateur. L'environnement comprend un éditeur de code, un débogueur et un terminal.

La `create-environment-ec2` commande suivante crée un environnement de AWS Cloud9 EC2 développement avec les paramètres spécifiés. Il lance une EC2 instance Amazon, puis se connecte de l'instance à l'environnement.

```
aws cloud9 create-environment-ec2 --name my-demo-env --description "My demonstration development environment." --instance-type t2.micro --subnet-id subnet-1fab8aEX --automatic-stop-time-minutes 60 --owner-arn arn:aws:iam::123456789012:user/MyDemoUser
```

Pour plus d'informations, consultez la section Référence de la [AWS Cloud9 ligne de commande](#).

Amazon EC2

Amazon Elastic Compute Cloud (AmazonEC2) est un service Web qui fournit une capacité de calcul sécurisée et redimensionnable dans le cloud. Il est conçu pour rendre le cloud computing à l'échelle du Web plus facile et plus accessible.

La `run-instances` commande suivante lance une instance `t2.micro` dans le sous-réseau spécifié d'un : VPC

```
aws ec2 run-instances --image-id ami-xxxxxxx --count 1 --instance-type t2.micro --key-name MyKeyPair --security-group-ids sg-903004f8 --subnet-id subnet-6e7f829e
```

Pour plus d'informations, consultez la section [Utilisation EC2 d'Amazon AWS CLI dans le](#) guide de AWS Command Line Interface l'utilisateur.

S3 Glacier

S3 Glacier et S3 Glacier Deep Archive sont des classes de stockage cloud Amazon S3 sécurisées, durables et extrêmement économiques destinées à l'archivage des données et à la sauvegarde à long terme.

La `create-vault` commande suivante crée un coffre, un conteneur pour le stockage des archives :

```
aws glacier create-vault --vault-name my-vault --account-id -
```

Pour plus d'informations, consultez la section [Utilisation d'Amazon S3 Glacier AWS CLI dans le](#) guide de AWS Command Line Interface l'utilisateur.

AWS Elastic Beanstalk CLI

AWS Elastic Beanstalk CLIFournit une interface de ligne de commande conçue pour simplifier la création, la mise à jour et la surveillance d'environnements à partir d'un référentiel local. Dans ce contexte, un environnement fait référence à un ensemble de AWS ressources exécutant une version d'application.

La `create` commande suivante crée un nouvel environnement dans un Amazon Virtual Private Cloud personnalisé (VPC).

```
$ eb create dev-vpc --vpc.id vpc-0ce8dd99 --vpc.elbsubnets subnet-  
b356d7c6,subnet-02f74b0c --vpc.ec2subnets subnet-0bb7f0cd,subnet-3b6697c1 --  
vpc.securitygroup sg-70cff265
```

Pour plus d'informations, consultez la [référence des CLI commandes EB](#) dans le Guide du AWS Elastic Beanstalk développeur.

Amazon ECS CLI

L'interface de ligne de commande Amazon Elastic Container Service (AmazonECS) (CLI) fournit plusieurs commandes de haut niveau. Ils sont conçus pour simplifier les processus de création, de mise à jour et de surveillance des clusters et des tâches à partir d'un environnement de développement local. (Un ECS cluster Amazon est un regroupement logique de tâches ou de services.)

La configure commande suivante configure Amazon ECS CLI pour créer une configuration de cluster nommée `ecs-cli-demo`. Cette configuration de cluster utilise FARGATE comme type de lancement par défaut pour le `ecs-cli-demo` cluster dans le `us-east-1` region.

```
ecs-cli configure --region us-east-1 --cluster ecs-cli-demo --default-launch-type  
FARGATE --config-name ecs-cli-demo
```

Pour plus d'informations, consultez le manuel [Amazon ECS Command Line Reference](#) dans le manuel du développeur Amazon Elastic Container Service.

AWS SAM CLI

AWS SAM CLI est un outil de ligne de commande qui fonctionne sur un AWS Serverless Application Model modèle et un code d'application. Vous pouvez effectuer plusieurs tâches en l'utilisant. Il s'agit notamment d'invoquer des fonctions Lambda localement, de créer un package de déploiement pour votre application sans serveur et de déployer votre application sans serveur dans le cloud. AWS

La `init` commande suivante initialise un nouveau SAM projet avec les paramètres requis transmis en tant que paramètres :

```
sam init --runtime python3.7 --dependency-manager pip --app-template hello-world --name  
sam-app
```

Pour plus d'informations, consultez la [référence des AWS SAM CLI commandes](#) dans le Guide du AWS Serverless Application Model développeur.

Personnalisation de votre expérience AWS CloudShell

Vous pouvez personnaliser les aspects suivants de votre AWS CloudShell expérience :

- [Disposition des onglets](#) : divisez l'interface de ligne de commande en plusieurs colonnes et lignes.
- [Taille de police](#) : ajustez la taille du texte de la ligne de commande.
- [Thème de couleur](#) : bascule entre le thème clair et le thème foncé.
- [Collage sécurisé](#) : activez ou désactivez une fonctionnalité qui vous oblige à vérifier le texte multiligne avant qu'il ne soit collé.
- [Restauration de Tmux vers session](#) : l'utilisation de tmux restaure votre session jusqu'à ce qu'elle soit inactive.

Vous pouvez également étendre votre environnement shell en [installant votre propre logiciel](#) et [en modifiant votre shell à l'aide de scripts](#).

Diviser l'affichage de la ligne de commande en plusieurs onglets

Exécutez plusieurs commandes en divisant votre interface de ligne de commande en plusieurs volets.

Note

Après avoir ouvert plusieurs onglets, vous pouvez sélectionner celui dans lequel vous souhaitez travailler en cliquant n'importe où dans le volet de votre choix. Vous pouvez fermer un onglet en choisissant le symbole x, qui se trouve à côté du nom de la région.

- Choisissez Actions et l'une des options suivantes dans la disposition des onglets :
 - Nouvel onglet : ajoutez un nouvel onglet à côté de celui actuellement actif.
 - Diviser en lignes : ajoutez un nouvel onglet dans une ligne située en dessous de l'onglet actuellement actif.
 - Diviser en colonnes : ajoutez un nouvel onglet dans une colonne située à côté de l'onglet actuellement actif.

S'il n'y a pas assez d'espace pour afficher complètement chaque onglet, faites défiler l'écran pour afficher l'intégralité de l'onglet. Vous pouvez également sélectionner les barres séparées qui séparent les volets et les faire glisser à l'aide du pointeur pour augmenter ou réduire la taille du volet.

Modification de la taille de police

Augmentez ou diminuez la taille du texte affiché dans l'interface de ligne de commande.

1. Pour modifier les paramètres du AWS CloudShell terminal, allez dans Réglages, Préférences.
2. Choisissez une taille de texte. Vos options sont les plus petites, les plus petites, les moyennes, les plus grandes et les plus grandes.

Modification du thème de l'interface

Basculez entre le thème clair et le thème foncé pour l'interface de ligne de commande.

1. Pour changer de AWS CloudShell thème, allez dans Réglages, Préférences.
2. Choisissez Clair ou Foncé.

Utilisation du Safe Paste pour du texte multiligne

Le collage sécurisé est une fonctionnalité de sécurité qui vous invite à vérifier que le texte multiligne que vous êtes sur le point de coller dans le shell ne contient pas de scripts malveillants. Le texte copié depuis des sites tiers peut contenir du code masqué qui déclenche des comportements inattendus dans votre environnement shell.

La boîte de dialogue Safe Paste affiche le texte complet que vous avez copié dans votre presse-papiers. Si vous êtes convaincu qu'il n'y a aucun risque de sécurité, choisissez Coller.

Warning: Pasting multiline text into AWS CloudShell



Text that's copied from external sources can contain malicious scripts. Verify the text below before pasting.

```
import sys
x=int(sys.argv[1])
y=int(sys.argv[2])
z=int(sys.argv[3])
total=x+y+z
print("The total is",total)
```

Always ask before pasting multiline code

Cancel

Paste

Nous vous recommandons d'activer Safe Paste pour détecter les risques de sécurité potentiels dans les scripts. Vous pouvez activer ou désactiver cette fonctionnalité en choisissant Préférences, Activer le collage sécurisé et Désactiver le collage sécurisé.

Utilisation tmux pour restaurer une session

AWS CloudShell utilise tmux pour restaurer les sessions sur un ou plusieurs onglets de navigateur. Si vous actualisez les onglets du navigateur, votre session reprend jusqu'à ce qu'elle soit inactive. Pour plus d'informations, consultez la section [Restauration de session](#).

Utilisation AWS CloudShell sur Amazon VPC

AWS CloudShell le cloud privé virtuel (VPC) vous permet de créer un CloudShell environnement dans votre VPC. Pour chaque VPC environnement, vous pouvez attribuer un VPC, ajouter un sous-réseau et associer jusqu'à cinq groupes de sécurité. AWS CloudShell hérite de la configuration réseau du VPC et vous permet de l'utiliser AWS CloudShell en toute sécurité au sein du même sous-réseau que les autres ressources du VPC et de vous y connecter.

Avec AmazonVPC, vous pouvez lancer AWS des ressources dans un réseau virtuel logiquement isolé que vous avez défini. Ce réseau virtuel ressemble beaucoup à un réseau traditionnel que vous pourriez exécuter dans votre propre data center, et présente l'avantage d'utiliser l'infrastructure évolutive d' AWS. Pour plus d'informations sur Amazon Virtual Private CloudVPC, consultez [Amazon Virtual Private Cloud](#).

Contraintes d'exploitation

AWS CloudShell VPCles environnements présentent les contraintes suivantes :

- Vous pouvez créer un maximum de deux VPC environnements par IAM principal.
- Vous pouvez attribuer un maximum de cinq groupes de sécurité à un VPC environnement.
- Vous ne pouvez pas utiliser les options de CloudShell chargement et de téléchargement du menu Actions pour les VPC environnements.

Note

Il est possible de charger ou de télécharger des fichiers à partir d'VPCenvironnements ayant accès à l'entrée/à la sortie d'Internet via d'autres outils. CLI

- VPCles environnements ne prennent pas en charge le stockage persistant. Le stockage est éphémère. Les données et le répertoire de base sont supprimés à la fin d'une session d'environnement actif.
- Votre AWS CloudShell environnement ne peut se connecter à Internet que s'il se trouve dans un VPC sous-réseau privé.

Note

Les adresses IP publiques ne sont pas allouées aux CloudShell VPC environnements par défaut. VPCles environnements créés dans des sous-réseaux publics avec des tables de routage configurées pour acheminer tout le trafic vers Internet Gateway n'auront pas accès à l'Internet public, mais les sous-réseaux privés configurés avec Network Address Translation (NAT) auront accès à Internet public. VPCles environnements créés dans de tels sous-réseaux privés auront accès à l'Internet public.

- Pour fournir un CloudShell environnement géré à votre compte, vous AWS pouvez fournir un accès réseau aux services suivants pour l'hôte de calcul sous-jacent :
 - Amazon S3
 - VPCpoints de terminaison
 - com.amazonaws. <region>Messages .ssm
 - com.amazonaws. <region>.journaux
 - com.amazonaws. <region>.km
 - com.amazonaws. <region>.execute-api
 - com.amazonaws. <region>.ecs-télémetrie
 - com.amazonaws. <region>.ecs-agent
 - com.amazonaws. <region>.ecs
 - com.amazonaws. <region>.ecr .dkr
 - com.amazonaws. <region>.ecr.api
 - com.amazonaws. <region>.codecatalyst.packages
 - com.amazonaws. <region>.codecatalyst.git
 - aws.api.global.codecatalyst

Vous ne pouvez pas restreindre l'accès à ces points de terminaison en modifiant votre VPC configuration.

CloudShellVPCest disponible dans toutes les AWS régions, à l'exception des régions AWS GovCloud (États-Unis). Pour obtenir la liste des régions disponibles, consultez la section [AWS Régions prises en charge pour AWS CloudShell](#). CloudShell VPC

Création d'un CloudShell VPC environnement

Cette rubrique explique les étapes à suivre pour créer un VPC environnement dans CloudShell.

Prérequis

Votre administrateur doit fournir les IAM autorisations nécessaires pour que vous puissiez créer des VPC environnements. Pour plus d'informations sur l'activation des autorisations de création d' CloudShell VPC environnements, consultez [the section called “IAM Autorisations requises pour créer et utiliser CloudShell VPC des environnements”](#).

Pour créer un CloudShell VPC environnement

1. Sur la page de la CloudShell console, choisissez l'icône+, puis choisissez Créer un VPC environnement dans le menu déroulant.
2. Sur la page Créer un VPC environnement, entrez le nom de votre VPC environnement dans le champ Nom.
3. Dans la liste déroulante Virtual private cloud (VPC), sélectionnez un VPC.
4. Dans la liste déroulante Sous-réseau, sélectionnez un sous-réseau.
5. Dans la liste déroulante des groupes de sécurité, choisissez un ou plusieurs groupes de sécurité que vous souhaitez attribuer à votre VPC environnement.

Note

Vous pouvez choisir un maximum de cinq groupes de sécurité.

6. Choisissez Créer pour créer votre VPC environnement.
7. (Facultatif) Choisissez Actions, puis sélectionnez Afficher les détails pour consulter les détails de l'VPC environnement nouvellement créé. L'adresse IP de votre VPC environnement est affichée dans l'invite de ligne de commande.

Pour plus d'informations sur l'utilisation VPC des environnements, consultez [Premiers pas](#).

IAM Autorisations requises pour créer et utiliser CloudShell VPC des environnements

Pour créer et utiliser CloudShell VPC des environnements, l'IAM administrateur doit autoriser l'accès à VPC des EC2 autorisations Amazon spécifiques. Cette section répertorie les EC2 autorisations Amazon nécessaires pour créer et utiliser VPC des environnements.

Pour créer VPC des environnements, la IAM politique attribuée à votre rôle doit inclure les EC2 autorisations Amazon suivantes :

- `ec2:DescribeVpcs`
- `ec2:DescribeSubnets`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeDhcpOptions`
- `ec2:DescribeNetworkInterfaces`

- `ec2:CreateTags`
- `ec2:CreateNetworkInterface`
- `ec2:CreateNetworkInterfacePermission`

Nous vous recommandons d'inclure :

- `ec2:DeleteNetworkInterface`

Note

Cette autorisation n'est pas obligatoire, mais elle est requise CloudShell pour nettoyer la ENI ressource (ENI créée pour les CloudShell VPC environnements marqués d'une `ManagedByCloudShell` clé) créée par celle-ci. Si cette autorisation n'est pas activée, vous devez nettoyer manuellement la ENI ressource après chaque utilisation de CloudShell VPC l'environnement.

IAM politique accordant un CloudShell accès complet, y compris l'accès à VPC

L'exemple suivant montre comment activer les autorisations complètes, y compris l'accès à VPC, pour CloudShell :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudShellOperations",
      "Effect": "Allow",
      "Action": [
        "cloudshell:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowDescribeVPC",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowCreateTagWithCloudShellKey",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:*:network-interface/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateNetworkInterface"
        },
        "ForAnyValue:StringEquals": {
          "aws:TagKeys": "ManagedByCloudShell"
        }
      }
    }
  ]
}
```



```
    }
  },
  {
    "Sid": "AllowCreateNetworkInterfaceWithSubnetsAndSG",
    "Effect": "Allow",
    "Action": [
      "ec2:CreateNetworkInterface"
    ],
    "Resource": [
      "arn:aws:ec2:*:*:subnet/*",
      "arn:aws:ec2:*:*:security-group/*"
    ]
  },
  {
    "Sid": "AllowCreateNetworkInterfaceWithCloudShellTag",
    "Effect": "Allow",
    "Action": [
      "ec2:CreateNetworkInterface"
    ],
    "Resource": "arn:aws:ec2:*:*:network-interface/*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:TagKeys": "ManagedByCloudShell"
      }
    }
  },
  {
    "Sid": "AllowCreateNetworkInterfacePermissionWithCloudShellTag",
    "Effect": "Allow",
    "Action": [
      "ec2:CreateNetworkInterfacePermission"
    ],
    "Resource": "arn:aws:ec2:*:*:network-interface/*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/ManagedByCloudShell": ""
      }
    }
  },
  {
    "Sid": "AllowDeleteNetworkInterfaceWithCloudShellTag",
    "Effect": "Allow",
    "Action": [
      "ec2:DeleteNetworkInterface"
    ]
  }
}
```

```
    ],  
    "Resource": "arn:aws:ec2:*:*:network-interface/*",  
    "Condition": {  
      "StringEquals": {  
        "aws:ResourceTag/ManagedByCloudShell": ""  
      }  
    }  
  }  
]  
}
```

Utilisation de clés de IAM condition pour les VPC environnements

Vous pouvez utiliser des clés de condition CloudShell spécifiques pour les VPC paramètres afin de fournir des contrôles d'autorisation supplémentaires pour vos VPC environnements. Vous pouvez également spécifier les sous-réseaux et les groupes de sécurité que l'VPCenvironnement peut ou ne peut pas utiliser.

CloudShell prend en charge les clés de condition suivantes dans IAM les politiques :

- `CloudShell:VpcIds`— Autoriser ou refuser un ou plusieurs VPCs
- `CloudShell:SubnetIds`— Autoriser ou refuser un ou plusieurs sous-réseaux
- `CloudShell:SecurityGroupIds`— Autoriser ou refuser un ou plusieurs groupes de sécurité

Note

Si les autorisations accordées aux utilisateurs ayant accès aux CloudShell environnements publics sont modifiées pour ajouter une restriction à l'`cloudshell:createEnvironment`action, ils peuvent toujours accéder à leur environnement public existant. Toutefois, si vous souhaitez modifier une IAM politique comportant cette restriction et désactiver leur accès à l'environnement public existant, vous devez d'abord mettre à jour la IAM politique avec la restriction, puis vous assurer que chaque CloudShell utilisateur de votre compte supprime manuellement l'environnement public existant à l'aide de l'interface utilisateur CloudShell Web (Actions → Supprimer CloudShell l'environnement).

Exemples de politiques avec des clés de condition pour VPC les paramètres

Les exemples suivants montrent comment utiliser les clés de condition pour VPC les paramètres. Après avoir créé une instruction de politique avec les restrictions souhaitées, ajoutez l'instruction de politique pour l'utilisateur ou le rôle cible.

Assurez-vous que les utilisateurs créent uniquement VPC des environnements et interdisent la création d'environnements publics

Pour garantir que les utilisateurs ne peuvent créer que VPC des environnements, utilisez l'autorisation de refus, comme indiqué dans l'exemple suivant :

```
{
  "Statement": [
    {
      "Sid": "DenyCloudShellNonVpcEnvironments",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "Null": {
          "cloudshell:VpcIds": "true"
        }
      }
    }
  ]
}
```

Refuser aux utilisateurs l'accès à VPCs des sous-réseaux ou à des groupes de sécurité spécifiques

Pour refuser aux utilisateurs l'accès à une condition spécifique VPCs, utilisez cette option `StringEquals` pour vérifier la valeur de la `cloudshell:VpcIds` condition. L'exemple suivant refuse aux utilisateurs l'accès à `vpc-1` et `vpc-2` :

```
{
  "Version": "2012-10-17",
```

```

"Statement": [
  {
    "Sid": "EnforceOutOfVpc",
    "Action": [
      "cloudshell:CreateEnvironment"
    ],
    "Effect": "Deny",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "cloudshell:VpcIds": [
          "vpc-1",
          "vpc-2"
        ]
      }
    }
  }
]
}

```

Pour refuser aux utilisateurs l'accès à une condition spécifique VPCs, utilisez cette option `StringEquals` pour vérifier la valeur de la `cloudshell:SubnetIds` condition. L'exemple suivant refuse aux utilisateurs l'accès à `subnet-1` et `subnet-2` :

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceOutOfVpc",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloudshell:VpcIds": [
            "vpc-1",
            "vpc-2"
          ]
        }
      }
    }
  ]
}

```

```
]
}
```

Pour refuser aux utilisateurs l'accès à une condition spécifique VPCs, utilisez cette option `StringEquals` pour vérifier la valeur de la `cloudshell:SecurityGroupIds` condition. L'exemple suivant refuse aux utilisateurs l'accès à `sg-1` et `sg-2` :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceOutOfSecurityGroups",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "cloudshell:SecurityGroupIds": [
            "sg-1",
            "sg-2"
          ]
        }
      }
    }
  ]
}
```

Permettre aux utilisateurs de créer des environnements avec des VPC configurations spécifiques

Pour autoriser les utilisateurs à accéder à une condition spécifique VPCs, utilisez cette option `StringEquals` pour vérifier la valeur de la `cloudshell:VpcIds` condition. L'exemple suivant permet aux utilisateurs d'accéder à `vpc-1` et `vpc-2` :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceStayInSpecificVpc",
```

```
"Action": [
  "cloudshell:CreateEnvironment"
],
"Effect": "Allow",
"Resource": "*",
"Condition": {
  "StringEquals": {
    "cloudshell:VpcIds": [
      "vpc-1",
      "vpc-2"
    ]
  }
}
]
```

Pour autoriser les utilisateurs à accéder à une condition spécifique VPCs, utilisez cette option `StringEquals` pour vérifier la valeur de la `cloudshell:SubnetIds` condition. L'exemple suivant permet aux utilisateurs d'accéder à `subnet-1` et `subnet-2` :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceStayInSpecificSubnets",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "cloudshell:SubnetIds": [
            "subnet-1",
            "subnet-2"
          ]
        }
      }
    }
  ]
}
```

Pour autoriser les utilisateurs à accéder à une condition spécifique VPCs, utilisez cette option `StringEquals` pour vérifier la valeur de la `cloudshell:SecurityGroupIds` condition. L'exemple suivant permet aux utilisateurs d'accéder à `sg-1` et `sg-2` :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceStayInSpecificSecurityGroup",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "cloudshell:SecurityGroupIds": [
            "sg-1",
            "sg-2"
          ]
        }
      }
    }
  ]
}
```

Sécurité pour AWS CloudShell

Chez Amazon Web Services (AWS), la sécurité dans le cloud est la priorité principale. 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 cela comme la sécurité du cloud et la sécurité dans le cloud.

Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute tous les services proposés dans le AWS cloud et de vous fournir des services que vous pouvez utiliser en toute sécurité. Notre responsabilité en matière de sécurité est notre priorité absolue AWS, et l'efficacité de notre sécurité est régulièrement testée et vérifiée par des auditeurs tiers dans le cadre des [programmes de AWS conformité](#).

Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez et par d'autres facteurs, notamment la sensibilité de vos données, les exigences de votre organisation et les lois et réglementations applicables.

AWS CloudShell suit le [modèle de responsabilité partagée](#) à travers les AWS services spécifiques qu'il soutient. Pour obtenir des informations sur la sécurité des AWS services, consultez la [AWS page de documentation sur la sécurité AWS des services et les services concernés par les efforts de AWS conformité par programme de conformité](#).

Les rubriques suivantes expliquent comment procéder à la configuration AWS CloudShell pour atteindre vos objectifs de sécurité et de conformité.

Rubriques

- [Protection des données dans AWS CloudShell](#)
- [Identity and Access Management \(IAM\) pour AWS CloudShell](#)
- [Connexion et surveillance AWS CloudShell](#)
- [Validation de conformité pour AWS CloudShell](#)
- [Résilience dans AWS CloudShell](#)
- [Sécurité de l'infrastructure dans AWS CloudShell](#)
- [Bonnes pratiques de sécurité pour AWS CloudShell](#)
- [AWS CloudShell Sûreté FAQs](#)

Protection des données dans AWS CloudShell

Le [modèle de responsabilité AWS partagée](#) de s'applique à la protection des données dans AWS CloudShell. 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 responsabilitéAWS 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.
- UtilisezSSL/TLSpour 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. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section [Utilisation des CloudTrail sentiers](#) dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de FIPS 140 à 3 modules cryptographiques validés pour accéder AWS via une interface de ligne de commande ou unAPI, 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 CloudShell ou d'autres Services AWS utilisateurs de la consoleAPI, 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

Le chiffrement des données fait référence à la protection des données lorsqu'elles sont au repos, lorsqu'elles sont stockées AWS CloudShell et lorsqu'elles sont en transit entre les AWS CloudShell points de terminaison du service.

Chiffrement au repos à l'aide de AWS KMS

Le chiffrement au repos consiste à protéger vos données contre tout accès non autorisé en chiffrant les données stockées. Lors de l'utilisation AWS CloudShell, vous disposez gratuitement d'un espace de stockage permanent de 1 Go par AWS région. Le stockage permanent se trouve dans votre répertoire personnel (\$HOME) et vous est réservé. Contrairement aux ressources environnementales éphémères qui sont recyclées après la fin de chaque session du shell, les données de votre répertoire personnel persistent.

Le cryptage des données stockées dans AWS CloudShell est mis en œuvre à l'aide de clés cryptographiques fournies par AWS Key Management Service (AWS KMS). Il s'agit d'un AWS service géré permettant de créer et de contrôler les clés principales du client (CMKs), c'est-à-dire les clés de chiffrement utilisées pour chiffrer les données client stockées dans l' AWS CloudShell environnement. AWS CloudShell génère et gère des clés cryptographiques pour chiffrer les données pour le compte des clients.

Chiffrement en transit

Le chiffrement en transit consiste à protéger vos données contre l'interception pendant qu'elles se déplacent entre les points de terminaison de communication.

Par défaut, toutes les communications de données entre le navigateur Web du client et le cloud AWS CloudShell sont cryptées en envoyant le tout via une TLS connexion HTTPS /.

Vous n'avez rien à faire pour activer l'utilisation deHTTPS/TLSpour la communication.

Identity and Access Management (IAM) pour AWS CloudShell

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. IAMles administrateurs contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les CloudShell ressources. IAMest un Service AWS ventilateur que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- [Public ciblé](#)
- [Authentification par des identités](#)
- [Gestion des accès à l'aide de politiques](#)
- [Comment AWS CloudShell fonctionne avec IAM](#)
- [Exemples de politiques basées sur l'identité pour AWS CloudShell](#)
- [Résolution des problèmes d'identité et d'accès avec AWS CloudShell](#)
- [Gestion de AWS CloudShell l'accès et de l'utilisation à l'aide IAM de politiques](#)

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez CloudShell.

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

Administrateur du service — Si vous êtes responsable des CloudShell ressources de votre entreprise, vous avez probablement un accès complet à CloudShell. C'est à vous de déterminer les CloudShell fonctionnalités et les ressources auxquelles les utilisateurs de votre service doivent accéder. Vous devez ensuite envoyer des demandes à votre IAM administrateur pour modifier les autorisations des utilisateurs de votre service. Consultez les informations de cette page pour comprendre les concepts de base deIAM. Pour en savoir plus sur la façon dont votre entreprise peut utiliser IAM avec CloudShell, voir [Comment AWS CloudShell fonctionne avec IAM](#).

IAMadministrateur — Si vous êtes IAM administrateur, vous souhaitez peut-être en savoir plus sur la manière dont vous pouvez rédiger des politiques pour gérer l'accès à CloudShell. Pour consulter

des exemples de politiques CloudShell basées sur l'identité que vous pouvez utiliser dans IAM, consultez [Exemples de politiques basées sur l'identité pour AWS CloudShell](#)

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié (connecté à AWS) en tant que Utilisateur racine d'un compte AWS, en tant qu'IAMutilisateur ou en assumant un IAM rôle.

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

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

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

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

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé

de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui nécessitent que vous vous connectiez en tant qu'utilisateur root, consultez la section [Tâches nécessitant des informations d'identification utilisateur root](#) dans le Guide de IAM l'utilisateur.

Identité fédérée

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

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède à l'aide des informations d'identification fournies Services AWS par le biais d'une source d'identité. Lorsque des identités fédérées y accèdent Comptes AWS, elles assument des rôles, qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité afin de les utiliser dans toutes vos applications Comptes AWS et applications. Pour plus d'informations sur IAM Identity Center, consultez [Qu'est-ce qu'IAM Identity Center ?](#) dans le guide de AWS IAM Identity Center l'utilisateur.

Utilisateurs et groupes IAM

Un [IAMutilisateur](#) est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des IAM utilisateurs dotés d'informations d'identification à long terme, telles que des mots de passe et des clés d'accès. Toutefois, si vous avez des cas d'utilisation spécifiques qui nécessitent des informations d'identification à long terme auprès des IAM utilisateurs, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, voir [Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification à long terme](#) dans le Guide de IAM l'utilisateur.

Un [IAMgroupe](#) est une identité qui définit un ensemble d'IAMutilisateurs. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour

plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez nommer un groupe IAMAdminset lui donner les autorisations nécessaires pour administrer IAM des ressources.

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

IAMrôles

Un [IAMrôle](#) est une identité au sein de Compte AWS vous dotée d'autorisations spécifiques. Il est similaire à un IAM utilisateur, mais n'est pas associé à une personne en particulier. Vous pouvez assumer temporairement un IAM rôle dans le en AWS Management Console [changeant de rôle](#). Vous pouvez assumer un rôle en appelant une AWS API opération AWS CLI or ou en utilisant une option personnaliséeURL. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez la section [Utilisation IAM des rôles](#) dans le Guide de IAM l'utilisateur.

IAMles rôles dotés d'informations d'identification temporaires sont utiles dans les situations suivantes :

- Accès utilisateur fédéré : pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour plus d'informations sur les rôles pour la fédération, voir [Création d'un rôle pour un fournisseur d'identité tiers](#) dans le guide de IAM l'utilisateur. Si vous utilisez IAM Identity Center, vous configurez un ensemble d'autorisations. Pour contrôler les accès auxquels vos identités peuvent accéder après leur authentification, IAM Identity Center met en corrélation l'ensemble d'autorisations avec un rôle dans. IAM Pour plus d'informations sur les jeux d'autorisations, consultez [Jeux d'autorisations](#) dans le Guide de l'utilisateur AWS IAM Identity Center .
- Autorisations IAM utilisateur temporaires : un IAM utilisateur ou un rôle peut assumer un IAM rôle afin d'obtenir temporairement différentes autorisations pour une tâche spécifique.
- Accès entre comptes : vous pouvez utiliser un IAM rôle pour autoriser une personne (un mandant fiable) d'un autre compte à accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy).

Pour connaître la différence entre les rôles et les politiques basées sur les ressources pour l'accès entre comptes, voir [Accès aux ressources entre comptes IAM dans le guide](#) de l'IAM utilisateur.

- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
- Sessions d'accès transmises (FAS) — Lorsque vous utilisez un IAM utilisateur ou un rôle pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant au Service AWS, combinées à la demande Service AWS pour adresser des demandes aux services en aval. FAS les demandes ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur les politiques relatives FAS aux demandes, consultez la section [Transférer les sessions d'accès](#).
- Rôle de service — Un rôle de service est un [IAM rôle](#) qu'un service assume pour effectuer des actions en votre nom. Un IAM administrateur peut créer, modifier et supprimer un rôle de service de l'intérieur IAM. Pour plus d'informations, consultez [la section Création d'un rôle auquel déléguer des autorisations Service AWS](#) dans le Guide de IAM l'utilisateur.
- Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un Service AWS. Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un IAM administrateur peut consulter, mais pas modifier les autorisations pour les rôles liés à un service.
- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un IAM rôle pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une EC2 instance et qui font AWS CLI des AWS API demandes. Cela est préférable au stockage des clés d'accès dans l'EC2 instance. Pour attribuer un AWS rôle à une EC2 instance et le rendre disponible pour toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes exécutés sur l'EC2 instance d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez la section [Utilisation d'un IAM rôle pour accorder des autorisations aux applications exécutées sur des EC2 instances Amazon](#) dans le Guide de IAM l'utilisateur.

Pour savoir s'il faut utiliser IAM des rôles ou des IAM utilisateurs, voir [Quand créer un IAM rôle \(au lieu d'un utilisateur\)](#) dans le guide de IAM l'utilisateur.

Gestion des accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de JSON documents. Pour plus d'informations sur la structure et le contenu des documents de JSON politique, voir [Présentation des JSON politiques](#) dans le guide de IAM l'utilisateur.

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

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

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

Politiques basées sur l'identité

Les politiques basées sur l'identité sont JSON des documents de politique d'autorisation que vous pouvez joindre à une identité, telle qu'un IAM utilisateur, un groupe d'utilisateurs ou un rôle. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour savoir comment créer une politique basée sur l'identité, consultez la section [Création de IAM politiques](#) dans le Guide de l'IAMutilisateur.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs

utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour savoir comment choisir entre une politique gérée ou une politique intégrée, voir [Choisir entre des politiques gérées et des politiques intégrées dans le Guide](#) de l'IAMutilisateur.

Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents JSON de stratégie que vous attachez à une ressource. Les politiques de confiance dans les IAM rôles et les politiques relatives aux compartiments Amazon S3 sont des exemples de politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez [spécifier un principal](#) dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser de politiques AWS gérées depuis une IAM stratégie basée sur les ressources.

Listes de contrôle d'accès (ACLs)

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format du document JSON de stratégie.

Amazon S3 et Amazon VPC sont des exemples de services compatibles ACLs. AWS WAF Pour en savoir plus ACLs, consultez la [présentation de la liste de contrôle d'accès \(ACL\)](#) dans le guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- Limites d'autorisations — Une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une IAM entité (IAMutilisateur ou rôle). Vous pouvez définir une limite d'autorisations pour

une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez la section Limites d'[autorisations pour les IAM entités](#) dans le Guide de IAM l'utilisateur.

- **Politiques de contrôle des services (SCPs) :** SCPs JSON politiques qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans AWS Organizations. AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée Comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer des politiques de contrôle des services (SCPs) à l'un ou à l'ensemble de vos comptes. Les SCP limites d'autorisations pour les entités présentes dans les comptes des membres, y compris chacune d'entre elles Utilisateur racine d'un compte AWS. Pour plus d'informations sur les Organizations SCPs, voir [Politiques de contrôle des services](#) dans le Guide de AWS Organizations l'utilisateur.
- **Politiques de séance :** les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations, consultez la section [Politiques de session](#) dans le guide de IAM l'utilisateur.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section [Logique d'évaluation des politiques](#) dans le guide de IAM l'utilisateur.

Comment AWS CloudShell fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à CloudShell, découvrez quelles IAM fonctionnalités sont disponibles CloudShell.

IAM fonctionnalités que vous pouvez utiliser avec AWS CloudShell

IAM fonctionnalité	CloudShell soutien
Politiques basées sur l'identité	Oui
Politiques basées sur les ressources	Non
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique (spécifiques au service)	Oui
ACLs	Non
ABAC(balises dans les politiques)	Non
Informations d'identification temporaires	Oui
Transférer les sessions d'accès (FAS)	Non
Fonctions du service	Non
Rôles liés à un service	Non

Pour obtenir une vue d'ensemble du fonctionnement de la plupart des IAM fonctionnalités CloudShell et des autres AWS services, reportez-vous à la section [AWS Services compatibles IAM](#) dans le Guide de IAM l'utilisateur.

Politiques basées sur l'identité pour CloudShell

Prend en charge les politiques basées sur l'identité : oui

Les politiques basées sur l'identité sont JSON des documents de politique d'autorisation que vous pouvez joindre à une identité, telle qu'un IAM utilisateur, un groupe d'utilisateurs ou un rôle. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour savoir comment créer une politique basée sur l'identité, consultez la section [Création de IAM politiques](#) dans le Guide de l'IAM utilisateur.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier les actions et les ressources autorisées ou refusées ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité, car celle-ci s'applique à l'utilisateur ou au rôle auquel elle est attachée. Pour en savoir plus sur tous les éléments que vous pouvez utiliser dans une JSON politique, consultez la [référence aux éléments de IAM JSON politique](#) dans le Guide de IAM l'utilisateur.

Exemples de politiques basées sur l'identité pour CloudShell

Pour consulter des exemples de politiques CloudShell basées sur l'identité, consultez. [Exemples de politiques basées sur l'identité pour AWS CloudShell](#)

Politiques basées sur les ressources au sein de CloudShell

Prend en charge les politiques basées sur les ressources : non

Les politiques basées sur les ressources sont des documents JSON de stratégie que vous attachez à une ressource. Les politiques de confiance dans les IAM rôles et les politiques relatives aux compartiments Amazon S3 sont des exemples de politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez [spécifier un principal](#) dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour activer l'accès entre comptes, vous pouvez spécifier un compte entier ou IAM des entités d'un autre compte comme principal dans une politique basée sur les ressources. L'ajout d'un principal entre comptes à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un IAM administrateur du compte de confiance doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l'entité. Toutefois, si une politique basée sur des ressources accorde l'accès à un principal dans le même compte, aucune autre politique basée sur l'identité n'est requise. Pour plus d'informations, voir [Accès aux ressources entre comptes IAM dans](#) le Guide de IAM l'utilisateur.

Actions politiques pour CloudShell

Prend en charge les actions de politique : oui

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'Action élément d'une JSON politique décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès dans une politique. Les actions de stratégie portent généralement le même nom que l' AWS API opération associée. Il existe certaines exceptions, telles que les actions avec autorisation uniquement qui n'ont pas d'opération correspondante. API Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des CloudShell actions, reportez-vous à la section [Actions définies par AWS CloudShell](#) dans la référence d'autorisation de service. Certaines actions peuvent en comporter plusieurs API.

Les actions de politique en CloudShell cours utilisent le préfixe suivant avant l'action :

```
cloudshell
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [  
  "cloudshell:action1",  
  "cloudshell:action2"  
]
```

Pour consulter des exemples de politiques CloudShell basées sur l'identité, consultez. [Exemples de politiques basées sur l'identité pour AWS CloudShell](#)

Ressources politiques pour CloudShell

Prend en charge les ressources de politique : oui

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Resource` JSON de stratégie indique le ou les objets auxquels s'applique l'action. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de spécifier une ressource en utilisant son [Amazon Resource Name \(ARN\)](#). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

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

Pour consulter la liste des types de CloudShell ressources et leurs caractéristiques ARNs, consultez la section [Ressources définies par AWS CloudShell](#) dans la référence d'autorisation de service. Pour savoir avec quelles actions vous pouvez spécifier pour chaque ressource, consultez la ARN section [Actions définies par AWS CloudShell](#).

Pour consulter des exemples de politiques CloudShell basées sur l'identité, consultez. [Exemples de politiques basées sur l'identité pour AWS CloudShell](#)

Clés de conditions de politique pour CloudShell

Prend en charge les clés de condition de politique spécifiques au service : oui

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des [opérateurs de condition](#), tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération AND logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une OR opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez autoriser un IAM utilisateur à accéder à une ressource uniquement si celle-ci est étiquetée avec son nom IAM d'utilisateur. Pour plus d'informations, consultez [IAM la section Éléments de politique : variables et balises](#) dans le Guide de IAM l'utilisateur.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les [clés contextuelles de condition AWS globales](#) dans le guide de IAM l'utilisateur.

Pour consulter la liste des clés de CloudShell condition, reportez-vous à la section [Clés de condition pour AWS CloudShell](#) la référence d'autorisation de service. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez la section [Actions définies par AWS CloudShell](#).

Pour consulter des exemples de politiques CloudShell basées sur l'identité, consultez. [Exemples de politiques basées sur l'identité pour AWS CloudShell](#)

ACLs dans CloudShell

Supports ACLs : Non

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format du document JSON de stratégie.

ABAC avec CloudShell

Supports ABAC (balises dans les politiques) : Non

Le contrôle d'accès basé sur les attributs (ABAC) est une stratégie d'autorisation qui définit les autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises à IAM des entités (utilisateurs ou rôles) et à de nombreuses AWS ressources. Le balisage des entités et des ressources est la première étape de ABAC. Vous concevez ensuite des ABAC politiques pour autoriser les opérations lorsque le tag du principal correspond à celui de la ressource à laquelle il essaie d'accéder.

ABAC est utile dans les environnements qui se développent rapidement et aide dans les situations où la gestion des politiques devient fastidieuse.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'[élément de condition](#) d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations ABAC, voir [Qu'est-ce que c'est ABAC ?](#) dans le guide de IAM l'utilisateur. Pour consulter un didacticiel présentant les étapes de configuration ABAC, voir [Utiliser le contrôle d'accès basé sur les attributs \(ABAC\)](#) dans le guide de l'IAM utilisateur.

Utilisation d'informations d'identification temporaires avec CloudShell

Prend en charge les informations d'identification temporaires : oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section [relative à l'utilisation IAM](#) dans le Guide de IAM l'utilisateur.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, consultez la section [Passage à un rôle \(console\)](#) dans le guide de IAM l'utilisateur.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide du AWS CLI ou AWS API. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez la section Informations [d'identification de sécurité temporaires dans IAM](#).

Lorsque vous changerez de rôle, vous utiliserez un environnement différent. Vous ne pouvez pas changer de rôle dans le même AWS CloudShell environnement.

Transférer les sessions d'accès pour CloudShell

Prend en charge les sessions d'accès transféré (FAS) : Non

Lorsque vous utilisez un IAM utilisateur ou un rôle pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant au Service AWS, combinées à la demande Service AWS pour adresser des demandes aux services en aval. FAS les demandes ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur les politiques relatives FAS aux demandes, consultez la section [Transférer les sessions d'accès](#).

Rôles de service pour CloudShell

Supporte les rôles de service : Non

Un rôle de service est un [IAM rôle](#) qu'un service assume pour effectuer des actions en votre nom. Un IAM administrateur peut créer, modifier et supprimer un rôle de service de l'intérieur IAM. Pour plus d'informations, consultez [la section Création d'un rôle auquel déléguer des autorisations Service AWS](#) dans le Guide de IAM l'utilisateur.

Warning

La modification des autorisations associées à un rôle de service peut perturber CloudShell les fonctionnalités. Modifiez les rôles de service uniquement lorsque CloudShell vous recevez des instructions à cet effet.

Rôles liés à un service pour CloudShell

Prend en charge les rôles liés à un service : non

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un IAM administrateur peut consulter, mais pas modifier les autorisations pour les rôles liés à un service.

Exemples de politiques basées sur l'identité pour AWS CloudShell

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier CloudShell des ressources. Ils ne peuvent pas non plus effectuer de tâches en utilisant le AWS Management Console, AWS Command Line Interface (AWS CLI) ou AWS API. Pour autoriser les utilisateurs à effectuer des actions sur les ressources dont ils ont besoin, un IAM administrateur peut créer des IAM politiques. L'administrateur peut ensuite ajouter les IAM politiques aux rôles, et les utilisateurs peuvent assumer les rôles.

Pour savoir comment créer une politique IAM basée sur l'identité à l'aide de ces exemples de documents de JSON stratégie, consultez la section [Création de IAM politiques](#) dans le guide de l'IAMutilisateur.

Pour plus de détails sur les actions et les types de ressources définis par CloudShell, y compris le format de ARNs pour chacun des types de ressources, voir [Actions, ressources et clés de condition AWS CloudShell](#) dans la référence d'autorisation de service.

Rubriques

- [Bonnes pratiques en matière de politiques](#)
- [Utilisation de la CloudShell console](#)
- [Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations](#)

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer CloudShell des ressources dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez [les politiques AWS gérées ou les politiques AWS gérées pour les fonctions professionnelles](#) dans le Guide de IAM l'utilisateur.

- Appliquer les autorisations du moindre privilège : lorsque vous définissez des autorisations à IAM l'aide de politiques, accordez uniquement les autorisations nécessaires à l'exécution d'une tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation IAM pour appliquer des autorisations, consultez la section [Politiques et autorisations IAM dans](#) le guide de IAM l'utilisateur.
- Utilisez des conditions dans IAM les politiques pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques pour limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez rédiger une condition de politique pour spécifier que toutes les demandes doivent être envoyées en utilisant SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que AWS CloudFormation. Pour plus d'informations, voir [Éléments IAM JSON de politique : Condition](#) dans le guide de IAM l'utilisateur.
- Utilisez IAM Access Analyzer pour valider vos IAM politiques afin de garantir des autorisations sécurisées et fonctionnelles. IAM Access Analyzer valide les politiques nouvelles et existantes afin qu'elles respectent le langage des politiques (JSON) et IAM les IAM meilleures pratiques. IAM Access Analyzer fournit plus de 100 vérifications des politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez la section [Validation des politiques d'IAM Access Analyzer](#) dans le guide de IAM l'utilisateur.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des IAM utilisateurs ou un utilisateur root Compte AWS, activez-le MFA pour une sécurité supplémentaire. Pour exiger le MFA moment où les API opérations sont appelées, ajoutez MFA des conditions à vos politiques. Pour plus d'informations, consultez [la section Configuration de l'API accès MFA protégé](#) dans le Guide de l'IAM utilisateur.

Pour plus d'informations sur les meilleures pratiques en matière de [sécurité IAM](#), consultez la section [Bonnes pratiques en matière](#) de sécurité IAM dans le Guide de IAM l'utilisateur.

Utilisation de la CloudShell console

Pour accéder à la AWS CloudShell console, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher les détails CloudShell des ressources de votre Compte AWS. Si vous créez une politique basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette politique.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement le AWS CLI ou le AWS API. Au lieu de cela, autorisez uniquement l'accès aux actions correspondant à l'API opération qu'ils tentent d'effectuer.

Pour garantir que les utilisateurs et les rôles peuvent toujours utiliser la CloudShell console, associez également la politique CloudShell *ConsoleAccess* ou la politique *ReadOnly* AWS gérée aux entités. Pour plus d'informations, consultez la section [Ajouter des autorisations à un utilisateur](#) dans le Guide de IAM l'utilisateur.

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux IAM utilisateurs de consulter les politiques intégrées et gérées associées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide du AWS CLI ou. AWS API

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",

```

```
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

Résolution des problèmes d'identité et d'accès avec AWS CloudShell

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec CloudShell et IAM.

Rubriques

- [Je ne suis pas autorisé à effectuer une action dans CloudShell](#)
- [Je ne suis pas autorisé à effectuer iam : PassRole](#)
- [Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes CloudShell ressources](#)

Je ne suis pas autorisé à effectuer une action dans CloudShell

Si vous recevez une erreur qui indique que vous n'êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit lorsque l'utilisateur `mateojacksonIAMutilisateur` essaie d'utiliser la console pour afficher les détails d'une `my-example-widget` ressource fictive mais ne dispose pas des `aws:GetWidget` autorisations fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
aws:GetWidget on resource: my-example-widget
```

Dans ce cas, la politique qui s'applique à l'utilisateur `mateojackson` doit être mise à jour pour autoriser l'accès à la ressource `my-example-widget` à l'aide de l'action `aws:GetWidget`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole`action, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle CloudShell.

Certains services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un IAM utilisateur nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans CloudShell. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes CloudShell ressources

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si ces fonctionnalités sont prises CloudShell en charge, consultez [Comment AWS CloudShell fonctionne avec IAM](#).
- Pour savoir comment donner accès à vos ressources sur un site Comptes AWS qui vous appartient, consultez la section [Fournir l'accès à un IAM utilisateur dans un autre site Compte AWS que vous possédez](#) dans le Guide de IAM l'utilisateur.

- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section [Fournir un accès à des ressources Comptes AWS détenues par des tiers](#) dans le Guide de IAM l'utilisateur.
- Pour savoir comment fournir un accès via la fédération d'identité, consultez la section [Fournir un accès aux utilisateurs authentifiés de manière externe \(fédération d'identité\)](#) dans le guide de l'IAMutilisateur.
- Pour connaître la différence entre l'utilisation de rôles et l'utilisation de politiques basées sur les ressources pour l'accès entre comptes, voir Accès aux [ressources entre comptes IAM dans le guide](#) de l'IAMutilisateur.

Gestion de AWS CloudShell l'accès et de l'utilisation à l'aide IAM de politiques

Grâce aux ressources de gestion des accès qui peuvent être fournies par AWS Identity and Access Management, les administrateurs peuvent accorder des autorisations aux IAM utilisateurs. De cette façon, ces utilisateurs peuvent accéder aux fonctionnalités de l'environnement AWS CloudShell et les utiliser. Les administrateurs peuvent également créer des politiques qui spécifient à un niveau granulaire les actions que ces utilisateurs peuvent effectuer dans l'environnement shell.

Le moyen le plus rapide pour un administrateur d'accorder l'accès aux utilisateurs est d'utiliser une politique AWS gérée. Une [politique AWS gérée](#) est une politique autonome créée et administrée par AWS. La politique AWS gérée suivante pour AWS CloudShell peut être attachée aux IAM identités :

- `AWS CloudShellFullAccess`: accorde l'autorisation d'utilisation AWS CloudShell avec un accès complet à toutes les fonctionnalités.

La `AWS CloudShellFullAccess` politique utilise le caractère générique (*) pour donner à l'IAMidentité (utilisateur, rôle ou groupe) un accès complet aux fonctionnalités CloudShell et aux fonctionnalités. Pour plus d'informations sur cette politique, consultez le Guide [AWS CloudShellFullAccess](#) de l'utilisateur de AWS Managed Policy.

Note

IAMles identités dotées des politiques AWS gérées suivantes peuvent également être lancées CloudShell. Toutefois, ces politiques prévoient des autorisations étendues. Nous vous recommandons donc de n'accorder ces politiques que si elles sont essentielles au rôle IAM professionnel de l'utilisateur.

- [Administrateur](#) : fournit IAM aux utilisateurs un accès complet et leur permet de déléguer des autorisations à chaque service et ressource qu'il contient AWS.
- [Développeur expérimenté : permet aux IAM utilisateurs d'](#)effectuer des tâches de développement d'applications et de créer et configurer des ressources et des services qui prennent en charge le développement d'applications en AWS toute connaissance de cause.

Pour plus d'informations sur l'attachement de politiques gérées, consultez la section [Ajout IAM d'autorisations d'identité \(console\)](#) dans le guide de IAM l'utilisateur.

Gestion des actions autorisées à AWS CloudShell l'aide de politiques personnalisées

Pour gérer les actions qu'un IAM utilisateur peut effectuer CloudShell, créez une politique personnalisée qui utilise la stratégie CloudShellPolicy gérée comme modèle. Vous pouvez également modifier [une politique](#) intégrée intégrée à l'IAMidentité appropriée (utilisateur, groupe ou rôle).

Par exemple, vous pouvez autoriser IAM les utilisateurs à accéder CloudShell, mais les empêcher de transmettre les informations d'identification de CloudShell l'environnement utilisées pour se connecter AWS Management Console.

Important

Pour lancer AWS CloudShell depuis le AWS Management Console, un IAM utilisateur doit disposer des autorisations nécessaires pour effectuer les actions suivantes :

- `CreateEnvironment`
- `CreateSession`
- `GetEnvironmentStatus`
- `StartEnvironment`

Si l'une de ces actions n'est pas explicitement autorisée par une politique attachée, une erreur d'IAM autorisation est renvoyée lorsque vous essayez de le lancer CloudShell.

AWS CloudShell autorisations

Nom	Description de l'autorisation accordée	Nécessaire pour le lancement CloudShell ?
<code>cloudshell:CreateEnvironment</code>	Crée un CloudShell environnement, récupère la mise en page au début de la CloudShell session et enregistre la mise en page actuelle depuis l'application Web dans le backend. Cette autorisation n'est attendue * que comme valeur pourResource, comme indiqué dans the section called "Exemples de IAM politiques pour CloudShell" .	Oui
<code>cloudshell:CreateSession</code>	Se connecte à un CloudShell environnement à partir du AWS Management Console.	Oui
<code>cloudshell:GetEnvironmentStatus</code>	Lisez l'état d'un CloudShell environnement.	Oui
<code>cloudshell>DeleteEnvironment</code>	Supprime un CloudShell environnement.	Non

Nom	Description de l'autorisation accordée	Nécessaire pour le lancement CloudShell ?
<code>cloudshell:GetFileDownloadUrls</code>	Génère un Amazon S3 pré-signé URLs qui est utilisé pour télécharger des fichiers CloudShell via l'interface CloudShell Web. Ceci n'est pas disponible pour les VPC environnements.	Non
<code>cloudshell:GetFileUploadUrls</code>	Génère un Amazon S3 pré-signé URLs qui est utilisé pour charger des fichiers CloudShell via l'interface CloudShell Web. Ceci n'est pas disponible pour les VPC environnements.	Non
<code>cloudshell:DescribeEnvironments</code>	Décrit les environnements.	Non
<code>cloudshell:PutCredentials</code>	Transfère les informations d'identification utilisées pour se connecter AWS Management Console au CloudShell.	Non
<code>cloudshell:StartEnvironment</code>	Démarre un CloudShell environnement qui est arrêté.	Oui
<code>cloudshell:StopEnvironment</code>	Arrête un CloudShell environnement en cours d'exécution.	Non

Exemples de IAM politiques pour CloudShell

Les exemples suivants montrent comment des politiques peuvent être créées pour restreindre les personnes autorisées à y accéder CloudShell. Les exemples montrent également les actions qui peuvent être effectuées dans l'environnement shell.

La politique suivante impose un déni complet de l'accès à CloudShell ses fonctionnalités.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "DenyCloudShell",
    "Effect": "Deny",
    "Action": [
      "cloudshell:*"
    ],
    "Resource": "*"
  }]
}
```

La politique suivante autorise IAM les utilisateurs à y accéder CloudShell mais les empêche de générer des documents pré-signés URLs pour le chargement et le téléchargement de fichiers. Les utilisateurs peuvent toujours transférer des fichiers vers et depuis l'environnement, en utilisant des clients par wget exemple.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowUsingCloudshell",
      "Effect": "Allow",
      "Action": [
        "cloudshell:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DenyUploadDownload",
      "Effect": "Deny",
      "Action": [
        "cloudshell:GetFileDownloadUrls",
        "cloudshell:GetFileUploadUrls"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "*"
  }]
}

```

La politique suivante autorise IAM les utilisateurs à y accéder CloudShell. Toutefois, la politique empêche le transfert vers l' CloudShell environnement des informations d'identification que vous avez utilisées pour vous connecter. AWS Management Console IAMles utilisateurs soumis à cette politique doivent configurer manuellement leurs informations d'identification dans cette politique CloudShell.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowUsingCloudshell",
      "Effect": "Allow",
      "Action": [
        "cloudshell:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DenyCredentialForwarding",
      "Effect": "Deny",
      "Action": [
        "cloudshell:PutCredentials"
      ],
      "Resource": "*"
    }
  ]
}

```

La politique suivante permet aux IAM utilisateurs de créer AWS CloudShell des environnements.

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "CloudShellUser",
    "Effect": "Allow",
    "Action": [
      "cloudshell:CreateEnvironment",
      "cloudshell:CreateSession",
      "cloudshell:GetEnvironmentStatus",

```

```
        "cloudshell:StartEnvironment"  
    ],  
    "Resource": "*" ]]  
}
```

IAMAutorisations requises pour créer et utiliser CloudShell VPC des environnements

Pour créer et utiliser CloudShell VPC des environnements, l'IAMadministrateur doit autoriser l'accès à VPC des EC2 autorisations Amazon spécifiques. Cette section répertorie les EC2 autorisations Amazon nécessaires pour créer et utiliser VPC des environnements.

Pour créer VPC des environnements, la IAM politique attribuée à votre rôle doit inclure les EC2 autorisations Amazon suivantes :

- ec2:DescribeVpcs
- ec2:DescribeSubnets
- ec2:DescribeSecurityGroups
- ec2:DescribeDhcpOptions
- ec2:DescribeNetworkInterfaces

- ec2:CreateTags
- ec2:CreateNetworkInterface
- ec2:CreateNetworkInterfacePermission

Nous vous recommandons également d'inclure :

- ec2>DeleteNetworkInterface

Note

Cette autorisation n'est pas obligatoire, mais elle est requise CloudShell pour nettoyer la ENI ressource (ENIscrée pour les CloudShell VPC environnements marqués d'une ManagedByCloudShell clé) créée par celle-ci. Si cette autorisation n'est pas activée, vous devez nettoyer manuellement la ENI ressource après chaque utilisation de CloudShell VPC l'environnement.

IAMpolitique accordant un CloudShell accès complet, y compris l'accès à VPC

L'exemple suivant montre comment activer les autorisations complètes, y compris l'accès à VPC, pour CloudShell :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudShellOperations",
      "Effect": "Allow",
      "Action": [
        "cloudshell:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowDescribeVPC",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowCreateTagWithCloudShellKey",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:*:network-interface/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateNetworkInterface"
        },
        "ForAnyValue:StringEquals": {
          "aws:TagKeys": "ManagedByCloudShell"
        }
      }
    }
  ],
}
```

```
{
  "Sid": "AllowCreateNetworkInterfaceWithSubnetsAndSG",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface"
  ],
  "Resource": [
    "arn:aws:ec2:*:*:subnet/*",
    "arn:aws:ec2:*:*:security-group/*"
  ]
},
{
  "Sid": "AllowCreateNetworkInterfaceWithCloudShellTag",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface"
  ],
  "Resource": "arn:aws:ec2:*:*:network-interface/*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "aws:TagKeys": "ManagedByCloudShell"
    }
  }
},
{
  "Sid": "AllowCreateNetworkInterfacePermissionWithCloudShellTag",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterfacePermission"
  ],
  "Resource": "arn:aws:ec2:*:*:network-interface/*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceTag/ManagedByCloudShell": ""
    }
  }
},
{
  "Sid": "AllowDeleteNetworkInterfaceWithCloudShellTag",
  "Effect": "Allow",
  "Action": [
    "ec2>DeleteNetworkInterface"
  ],
  "Resource": "arn:aws:ec2:*:*:network-interface/*",
```

```
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/ManagedByCloudShell": ""
      }
    }
  ]
}
```

Utilisation de clés de IAM condition pour les VPC environnements

Vous pouvez utiliser des clés de condition CloudShell spécifiques pour les VPC paramètres afin de fournir des contrôles d'autorisation supplémentaires pour vos VPC environnements. Vous pouvez également spécifier les sous-réseaux et les groupes de sécurité que l'VPCenvironnement peut ou ne peut pas utiliser.

CloudShell prend en charge les clés de condition suivantes dans IAM les politiques :

- `CloudShell:VpcIds`— Autoriser ou refuser un ou plusieurs VPCs
- `CloudShell:SubnetIds`— Autoriser ou refuser un ou plusieurs sous-réseaux
- `CloudShell:SecurityGroupIds`— Autoriser ou refuser un ou plusieurs groupes de sécurité

Note

Si les autorisations accordées aux utilisateurs ayant accès aux CloudShell environnements publics sont modifiées pour restreindre l'`cloudshell:createEnvironmentaction`, ils peuvent toujours accéder à leur environnement public existant. Toutefois, si vous souhaitez modifier une IAM politique comportant cette restriction et désactiver leur accès à l'environnement public existant, vous devez d'abord mettre à jour la IAM politique avec la restriction, puis vous assurer que chaque CloudShell utilisateur de votre compte supprime manuellement l'environnement public existant à l'aide de l'interface utilisateur CloudShell Web (Actions → Supprimer CloudShell l'environnement).

Exemples de politiques avec des clés de condition pour VPC les paramètres

Les exemples suivants montrent comment utiliser les clés de condition pour VPC les paramètres. Après avoir créé une instruction de politique avec les restrictions souhaitées, ajoutez l'instruction de politique pour l'utilisateur ou le rôle cible.

Assurez-vous que les utilisateurs créent uniquement VPC des environnements et interdisent la création d'environnements publics

Pour garantir que les utilisateurs ne peuvent créer que VPC des environnements, utilisez l'autorisation de refus, comme indiqué dans l'exemple suivant :

```
{
  "Statement": [
    {
      "Sid": "DenyCloudShellNonVpcEnvironments",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "Null": {
          "cloudshell:VpcIds": "true"
        }
      }
    }
  ]
}
```

Refuser aux utilisateurs l'accès à VPCs des sous-réseaux ou à des groupes de sécurité spécifiques

Pour refuser aux utilisateurs l'accès à une condition spécifique VPCs, utilisez cette option `StringEquals` pour vérifier la valeur de la `cloudshell:VpcIds` condition. L'exemple suivant refuse aux utilisateurs l'accès à `vpc-1` et `vpc-2` :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceOutOfVpc",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
```

```

        "cloudshell:VpcIds": [
            "vpc-1",
            "vpc-2"
        ]
    }
}
]
}

```

Pour refuser aux utilisateurs l'accès à une condition spécifique VPCs, utilisez cette option `StringEquals` pour vérifier la valeur de la `cloudshell:SubnetIds` condition. L'exemple suivant refuse aux utilisateurs l'accès à `subnet-1` et `subnet-2` :

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceOutOfVpc",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloudshell:VpcIds": [
            "vpc-1",
            "vpc-2"
          ]
        }
      }
    }
  ]
}

```

Pour refuser aux utilisateurs l'accès à une condition spécifique VPCs, utilisez cette option `StringEquals` pour vérifier la valeur de la `cloudshell:SecurityGroupIds` condition. L'exemple suivant refuse aux utilisateurs l'accès à `sg-1` et `sg-2` :

```

{
  "Version": "2012-10-17",

```

```
"Statement": [  
  {  
    "Sid": "EnforceOutOfSecurityGroups",  
    "Action": [  
      "cloudshell:CreateEnvironment"  
    ],  
    "Effect": "Deny",  
    "Resource": "*",  
    "Condition": {  
      "ForAnyValue:StringEquals": {  
        "cloudshell:SecurityGroupIds": [  
          "sg-1",  
          "sg-2"  
        ]  
      }  
    }  
  }  
]
```

Permettre aux utilisateurs de créer des environnements avec des VPC configurations spécifiques

Pour autoriser les utilisateurs à accéder à une condition spécifique VPCs, utilisez cette option `StringEquals` pour vérifier la valeur de la `cloudshell:VpcIds` condition. L'exemple suivant permet aux utilisateurs d'accéder à `vpc-1` et `vpc-2` :

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "EnforceStayInSpecificVpc",  
      "Action": [  
        "cloudshell:CreateEnvironment"  
      ],  
      "Effect": "Allow",  
      "Resource": "*",  
      "Condition": {  
        "StringEquals": {  
          "cloudshell:VpcIds": [  
            "vpc-1",  
            "vpc-2"  
          ]  
        }  
      }  
    }  
  ]  
}
```

```
    }
  }
]
}
```

Pour autoriser les utilisateurs à accéder à une condition spécifique VPCs, utilisez cette option `StringEquals` pour vérifier la valeur de la `cloudshell:SubnetIds` condition. L'exemple suivant permet aux utilisateurs d'accéder à `subnet-1` et `subnet-2` :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceStayInSpecificSubnets",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "cloudshell:SubnetIds": [
            "subnet-1",
            "subnet-2"
          ]
        }
      }
    }
  ]
}
```

Pour autoriser les utilisateurs à accéder à une condition spécifique VPCs, utilisez cette option `StringEquals` pour vérifier la valeur de la `cloudshell:SecurityGroupIds` condition. L'exemple suivant permet aux utilisateurs d'accéder à `sg-1` et `sg-2` :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceStayInSpecificSecurityGroup",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],

```

```
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
      "ForAllValues:StringEquals": {
        "cloudshell:SecurityGroupIds": [
          "sg-1",
          "sg-2"
        ]
      }
    }
  }
]
```

Autorisations d'accès Services AWS

CloudShell utilise les IAM informations d'identification que vous avez utilisées pour vous connecter au AWS Management Console.

Note

Pour utiliser les IAM informations d'identification que vous avez utilisées pour vous connecter au AWS Management Console, vous devez disposer d'une `cloudshell:PutCredentials` autorisation.

Cette fonctionnalité de pré-authentification CloudShell est pratique à utiliser AWS CLI. Cependant, un IAM utilisateur a toujours besoin d'autorisations explicites pour Services AWS les appels effectués depuis la ligne de commande.

Supposons, par exemple, que IAM les utilisateurs soient tenus de créer des compartiments Amazon S3 et d'y charger des fichiers sous forme d'objets. Vous pouvez créer une politique qui autorise explicitement ces actions. La IAM console fournit un [éditeur visuel](#) interactif qui guide le processus de création d'un document de politique JSON formaté. Une fois la politique créée, vous pouvez l'associer à IAM l'identité appropriée (utilisateur, groupe ou rôle).

Pour plus d'informations sur l'attachement de politiques gérées, consultez la section [Ajout IAM d'autorisations d'identité \(console\)](#) dans le guide de IAM l'utilisateur.

Connexion et surveillance AWS CloudShell

Cette rubrique décrit comment vous pouvez enregistrer et surveiller AWS CloudShell l'activité et les performances avec CloudTrail.

Surveillance de l'activité avec CloudTrail

AWS CloudShell est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou Service AWS dans AWS CloudShell. CloudTrail capture tous les API appels AWS CloudShell sous forme d'événements. Les appels capturés incluent des appels provenant de la AWS CloudShell console et des appels de code vers le AWS CloudShell API.

Si vous créez un suivi, vous pouvez activer la diffusion continue des CloudTrail événements vers un bucket Amazon Simple Storage Service (Amazon S3). Cela inclut les événements pour AWS CloudShell.

Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez découvrir diverses informations concernant une demande. Par exemple, vous pouvez déterminer la demande qui a été faite AWS CloudShell, vous pouvez connaître l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande et la date à laquelle elle a été faite.

AWS CloudShell dans CloudTrail

Le tableau suivant répertorie les AWS CloudShell événements enregistrés dans le fichier CloudTrail journal.

Note

AWS CloudShell événement qui inclut :

- *indique qu'il s'agit d'un appel non mutante (lecture seule). API
- Le terme fait Environment référence au cycle de vie de l'environnement informatique qui héberge l'expérience shell.
- Le mot Layout restaure tous les onglets du navigateur dans le CloudShell terminal.

CloudShell Événements à CloudTrail

Nom de l'événement	Description
<code>createEnvironment</code>	Survient lors de la création d'un CloudShell environnement.
<code>createSession</code>	Se produit lorsqu'un CloudShell environnement est connecté à partir du AWS Management Console.
<code>deleteEnvironment</code>	Survient lorsqu'un CloudShell environnement est supprimé.
<code>deleteSession</code>	Se produit lorsque la session de l' CloudShell onglet en cours d'exécution dans l'onglet actuel du navigateur est supprimée.
<code>getEnvironmentStatus*</code>	Se produit lorsque l'état d'un CloudShell environnement est récupéré.
<code>getFileDownloadUrls*</code>	Se produit lorsque des Amazon S3 URLs pré-signés utilisés pour télécharger des fichiers CloudShell via l'interface CloudShell Web sont générés.
<code>getFileUploadUrls*</code>	Se produit lorsque des Amazon S3 URLs pré-signés utilisés pour charger des fichiers CloudShell via l'interface CloudShell Web sont générés.
<code>cloudshell:DescribeEnvironments</code>	Décrit les environnements.
<code>getLayout*</code>	Se produit lorsque la CloudShell mise en page au début de la session est récupérée.
<code>putCredentials</code>	Se produit lorsque les informations d'identification utilisées pour se connecter AWS Management Console à CloudShell sont transférées.

Nom de l'événement	Description
<code>redeemCode*</code>	Se produit lorsque le flux de travail pour récupérer le jeton d'actualisation dans l' CloudShell environnement commence. Vous pouvez ensuite utiliser ce jeton dans la <code>putCredentials</code> commande pour accéder à l' CloudShell environnement.
<code>sendHeartBeat</code>	Survient pour confirmer que la CloudShell session est active.
<code>startEnvironment</code>	Se produit lors du démarrage d'un CloudShell environnement.
<code>stopEnvironment</code>	Se produit lorsqu'un CloudShell environnement en cours d'exécution est arrêté.
<code>updateLayout</code>	Se produit lorsque la mise en page actuelle de l'application Web dans le backend est enregistrée.

Les événements qui incluent le mot « Layout » restaurent tous les onglets du navigateur dans le CloudShell terminal.

EventBridge règles pour les AWS CloudShell actions

Avec EventBridge les règles, vous spécifiez une action cible à effectuer lors de la EventBridge réception d'un événement correspondant à la règle. Vous pouvez définir une règle qui spécifie une action cible à effectuer en fonction d'une AWS CloudShell action enregistrée en tant qu'événement dans un fichier CloudTrail journal.

Par exemple, vous pouvez [créer des EventBridge règles à AWS CLI](#) l'aide de la `put-rule` commande. Un `put-rule` appel doit contenir au moins un `EventPattern` ou `ScheduleExpression`. Les règles avec `EventPatterns` sont déclenchées lorsqu'un événement correspondant est observé. Les `EventPattern` pour les AWS CloudShell événements :


```
{ "source": [ "aws.cloudshell" ], "detail-type": [ "AWS API Call via CloudTrail" ],
  "detail": { "eventSource": [ "cloudshell.amazonaws.com" ] } }
```

Pour plus d'informations, consultez la section [Événements et modèles d'événements EventBridge](#) dans le guide de EventBridge l'utilisateur Amazon.

Validation de conformité pour AWS CloudShell

Des auditeurs tiers évaluent la sécurité et la conformité des AWS services dans le cadre de multiples programmes de AWS conformité.

AWS CloudShell est concerné par les programmes de conformité suivants :

SOC

AWS Les rapports de contrôle du système et de l'organisation (SOC) sont des rapports d'examen indépendants réalisés par des tiers qui montrent AWS comment les principaux contrôles et objectifs de conformité sont atteints.

Service	SDK	SOC1,2,3
AWS CloudShell	CloudShell	✓

PCI

La norme de sécurité des données de l'industrie des cartes de paiement (PCIDSS) est une norme exclusive de sécurité des informations administrée par le PCI Security Standards Council, fondé par American Express, Discover Financial Services, JCB International, MasterCard Worldwide et Visa Inc.

Service	SDK	PCI
AWS CloudShell	CloudShell	✓

ISOet CSA STAR certifications et services

AWS possède une certification de conformité aux normes ISO/IEC 27001:2013, 27017:2015, 27018:2019, 27701:2019, 22301:2019, 9001:2015, et v4.0. CSA STAR CCM

Service	SDK	ISOet CSA STAR certifications et services
AWS CloudShell	CloudShell	✓

FedRamp

Le Federal Risk and Authorization Management Program (FedRAMP) est un programme à l'échelle du gouvernement américain qui propose une approche standard en matière d'évaluation de la sécurité, d'autorisation et de surveillance continue des produits et services cloud.

Service	SDK	Fed RAMP modérée (Est/Ouest)	Red RAMP High (1GovCloud)
AWS CloudShell	CloudShell	✓	✓

DoD CC SRG

Le guide des exigences de sécurité du cloud computing () du ministère de la Défense (DoDSRG) fournit un processus d'évaluation et d'autorisation standardisé permettant aux fournisseurs de services cloud (CSPs) d'obtenir une autorisation provisoire du DoD, afin de pouvoir servir les clients du DoD.

Les services soumis à une SRG évaluation et à une autorisation du DoD CC auront le statut suivant :

- Organisation d'évaluation tierce (3PAO) Évaluation : Ce service fait actuellement l'objet d'une évaluation par notre évaluateur tiers.
- Examen conjoint par le Comité d'autorisation (JAB) : Ce service fait actuellement l'objet d'un JAB examen.
- Examen de la Defense Information Systems Agency (DISA) : Ce service fait actuellement l'objet d'un DISA examen.

Service	SDK	DoD CC SRG IL2 (Est/Ouest)	DoD CC () SRG IL2 GovCloud	DoD CC () SRG IL4 GovCloud	DoD CC () SRG IL5 GovCloud	DoD CC SRG IL6 (région AWS secrète)
AWS CloudShell	CloudShell	3 PAO Évaluation	N/A	N/A	N/A	N/A

HIPAA BAA

La Health Insurance Portability and Accountability Act de 1996 (HIPAA) est une loi fédérale qui a exigé la création de normes nationales pour protéger les informations sensibles sur la santé des patients contre toute divulgation à leur insu ou sans leur consentement.

AWS permet aux entités couvertes et à HIPAA leurs partenaires commerciaux concernés de traiter, de stocker et de transmettre en toute sécurité des informations de santé protégées (PHI). En outre, depuis juillet 2013, AWS propose un addendum normalisé pour les associés commerciaux (BAA) à ces clients.

Service	SDK	HIPAA BAA
AWS CloudShell	CloudShell	✓

IRAP

Le programme d'évaluateurs agréés en matière de sécurité de l'information (IRAP) permet aux clients du gouvernement australien de vérifier que les contrôles appropriés sont en place et de déterminer le modèle de responsabilité approprié pour répondre aux exigences du manuel de sécurité des informations du gouvernement australien (ISM) produit par le Centre australien de cybersécurité (ACSC).

Service	Espace de noms*	IRAPprotégé
AWS CloudShell	N/A	✓

*Les espaces de noms vous aident à identifier les services dans votre AWS environnement. Par exemple, lorsque vous créez des IAM politiques, travaillez avec Amazon Resource Names (ARNs) et lisez AWS CloudTrail des journaux.

MTCS

Le Multi-Tier Cloud Security (MTCS) est une norme opérationnelle de gestion de la sécurité de Singapour (SPRINGSS 584), basée sur les normes ISO 27001/02 du système de gestion de la sécurité de l'information (). ISMS

Service	SDK	USA Est (Ohio)	US-Est (Virginie du Nord)	US-Ouest (Oregon)	US-Ouest (Californie du Nord)	Singapour	Séoul
AWS CloudShell	CloudShell	✓	✓	✓	N/A	N/A	N/A

C5

Le Cloud Computing Compliance Controls Catalog (C5) est un système d'attestation soutenu par le gouvernement allemand introduit en Allemagne par l'Office fédéral de la sécurité de l'information (BSI) pour aider les entreprises à démontrer leur sécurité opérationnelle contre les cyberattaques courantes lors de l'utilisation de services cloud dans le contexte des « Recommandations de sécurité pour les fournisseurs de cloud » du gouvernement allemand.

Service	SDK	C5
AWS CloudShell	CloudShell	✓

ENSÉlevé

Le système d'accréditation ENS (Esquema Nacional de Seguridad) a été développé par le ministère des Finances et de l'Administration publique et le CCN (Centre national de cryptologie). Cela

comprend les principes de base et les exigences minimales nécessaires à une protection adéquate des informations.

Service	SDK	<u>ENSElevé</u>
AWS CloudShell	CloudShell	✓

FINMA

L'Autorité suisse de surveillance des marchés financiers (FINMA) est le régulateur indépendant des marchés financiers de la Suisse. AWS L'alignement sur les FINMA exigences démontre notre engagement continu à répondre aux attentes accrues des régulateurs des services financiers et des clients suisses à l'égard des fournisseurs de services cloud.

Service	SDK	<u>FINMA</u>
AWS CloudShell	CloudShell	✓

PiTuKri

AWS l'alignement sur les PiTuKri exigences démontre notre engagement continu à répondre aux attentes accrues des fournisseurs de services cloud définies par l'agence finlandaise des transports et des communications, Traficom.

Service	SDK	<u>PiTuKri</u>
AWS CloudShell	CloudShell	✓

Pour obtenir la liste des AWS services concernés par des programmes de conformité spécifiques, voir [AWSServices concernés par programme de conformité AWS](#) . Pour des informations générales, voir Programmes de [AWS conformité Programmes AWS](#) de .

Vous pouvez télécharger des rapports d'audit tiers en utilisant AWS Artifact. Pour plus d'informations, voir [Téléchargement de rapports dans AWS Artifact](#) .

Votre responsabilité en matière de conformité lors de l'utilisation AWS CloudShell est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Guides [de démarrage rapide sur la sécurité et la conformité](#) [Guides](#) sur la sécurité et la conformité — Ces guides de déploiement abordent les considérations architecturales et fournissent des étapes pour déployer des environnements de base axés sur la sécurité et sur la conformité sur AWS
- [Livre blanc sur l'architecture au service de la HIPAA sécurité et de la conformité](#) : ce livre blanc décrit comment les entreprises peuvent créer des applications AWS conformes. HIPAA
- AWS Ressources de <https://aws.amazon.com/compliance/resources/> de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- [Évaluation des ressources à l'aide des règles](#) du guide du AWS Config développeur : le AWS Config service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- [AWS Security Hub](#)— Ce AWS service fournit une vue complète de l'état de votre sécurité interne, AWS ce qui vous permet de vérifier votre conformité aux normes et aux meilleures pratiques du secteur de la sécurité.

Résilience dans AWS CloudShell

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. AWS Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, connectées par un réseau à faible latence, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section [Infrastructure AWS mondiale](#).

Outre l'infrastructure AWS mondiale, AWS CloudShell prend en charge les fonctionnalités suivantes pour répondre à vos besoins en matière de résilience et de sauvegarde des données :

- Validez les fichiers que vous créez et auxquels vous ajoutez des éléments AWS CodeCommit. Il s'agit d'un service de contrôle de version hébergé par Amazon Web Services que vous pouvez utiliser pour stocker et gérer des actifs de manière privée dans le cloud. Ces actifs peuvent être constitués de documents, de code source et de fichiers binaires. Pour de plus amples informations, veuillez consulter [Utilisation CodeCommit dans AWS CloudShell](#).
- Utilisez AWS CLI des appels pour spécifier les fichiers de votre répertoire personnel AWS CloudShell et les ajouter en tant qu'objets dans des compartiments Amazon S3. Pour un exemple, consultez la section [Getting started with AWS CloudShell](#).

Sécurité de l'infrastructure dans AWS CloudShell

En tant que service géré, AWS CloudShell il est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section [Sécurité du AWS cloud](#). Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section [Protection de l'infrastructure](#) dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez API les appels AWS publiés pour accéder AWS CloudShell via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Sécurité de la couche de transport (TLS). Nous avons besoin de la TLS version 1.2 et recommandons la TLS version 1.3.
- Des suites de chiffrement parfaitement confidentielles (PFS) telles que (Ephemeral Diffie-Hellman) ou DHE ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un identifiant de clé d'accès et d'une clé d'accès secrète associés à un IAM principal. Vous pouvez également utiliser [AWS Security Token Service](#) (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Note

Par défaut, installez AWS CloudShell automatiquement les correctifs de sécurité pour les packages système de vos environnements informatiques.

Bonnes pratiques de sécurité pour AWS CloudShell

Les bonnes pratiques suivantes doivent être considérées comme des instructions générales et ne représentent pas une solution de sécurité complète. Étant donné que ces meilleures pratiques peuvent ne pas être appropriées ou suffisantes pour votre environnement, nous vous recommandons de les considérer comme des considérations utiles plutôt que comme des prescriptions.

Quelques bonnes pratiques en matière de sécurité pour AWS CloudShell

- Utilisez IAM des autorisations et des politiques pour contrôler l'accès AWS CloudShell et garantir que les utilisateurs ne peuvent effectuer que les actions requises par leur rôle (par exemple, le téléchargement et le chargement de fichiers). Pour plus d'informations, consultez la section [Gestion de AWS CloudShell l'accès et de l'utilisation à l'aide IAM de politiques](#).
- N'incluez pas de données sensibles dans vos IAM entités, telles que les utilisateurs, les rôles ou les noms de session.
- Activez la fonction Safe Paste pour détecter les risques de sécurité potentiels dans le texte que vous avez copié à partir de sources externes. Le collage sécurisé est activé par défaut. Pour plus d'informations sur l'utilisation du collage sécurisé pour du texte multiligne, voir [Utilisation du collage sécurisé pour du texte multiligne](#).
- Familiarisez-vous avec le [modèle de responsabilité de sécurité partagée](#) si vous installez des applications tierces dans l'environnement informatique de AWS CloudShell.
- Préparez des mécanismes de restauration avant de modifier les scripts shell qui affectent l'expérience shell de l'utilisateur. Pour plus d'informations sur la modification de l'environnement shell par défaut, consultez [Modifier votre shell à l'aide de scripts](#).
- Stockez votre code de manière sécurisée dans un système de contrôle des versions, par exemple, [AWS CodeCommit](#).

AWS CloudShell Sûreté FAQs

Vous trouverez ci-dessous les réponses aux questions fréquemment posées sur la sécurité pour CloudShell.

- [Quels AWS processus et technologies sont utilisés lorsque vous lancez CloudShell et démarrez une session shell ?](#)
- [Est-il possible de restreindre l'accès au réseau CloudShell ?](#)
- [Puis-je personnaliser mon CloudShell environnement ?](#)

- [Où est réellement stocké mon \\$HOME répertoire dans le AWS Cloud ?](#)
- [Est-il possible de chiffrer mon \\$HOME répertoire ?](#)
- [Puis-je lancer une analyse antivirus sur mon \\$HOME répertoire ?](#)

Quels AWS processus et technologies sont utilisés lorsque vous lancez CloudShell et démarrez une session shell ?

Lorsque vous vous connectez AWS Management Console, vous entrez vos informations IAM d'identification d'utilisateur. Et, lorsque vous lancez le service CloudShell depuis l'interface de console, ces informations d'identification sont utilisées dans les appels CloudShell API qui créent un environnement informatique pour le service. Une AWS Systems Manager session est ensuite créée pour l'environnement informatique et CloudShell envoie des commandes à cette session.

[Retour à la liste des mesures de sécurité FAQs](#)

Est-il possible de restreindre l'accès au réseau CloudShell ?

Pour les environnements publics, il n'est pas possible de restreindre l'accès au réseau. Si vous souhaitez restreindre l'accès au réseau, vous devez autoriser la création d'VPC environnements uniquement et refuser la création d'environnements publics.

Pour plus d'informations, voir [Veiller à ce que les utilisateurs créent uniquement VPC des environnements et refuser la création d'environnements publics](#).

Pour les CloudShell VPC environnements, les paramètres réseau sont hérités de votre VPC. L'utilisation de CloudShell in a vous VPC permet de contrôler l'accès réseau de votre CloudShell VPC environnement.

[Retour à la liste des mesures de sécurité FAQs](#)

Puis-je personnaliser mon CloudShell environnement ?

Vous pouvez télécharger et installer des utilitaires et d'autres logiciels tiers adaptés à votre CloudShell environnement. Seuls les logiciels installés dans votre \$HOME répertoire sont conservés entre les sessions.

Comme le définit le [modèle de responsabilité AWS partagée](#), vous êtes responsable de la configuration et de la gestion nécessaires des applications que vous installez.

[Retour à la liste des mesures de sécurité FAQs](#)

Où est réellement stocké mon **\$HOME** répertoire dans le AWS Cloud ?

Pour les environnements publics, l'infrastructure de stockage des données dans votre environnement **\$HOME** est fournie par Amazon S3.

Pour les VPC environnements, votre **\$HOME** répertoire est supprimé lorsque le délai d'expiration de votre VPC environnement est expiré (après 20 à 30 minutes d'inactivité), ou lorsque vous supprimez ou redémarrez votre environnement.

[Retour à la liste des mesures de sécurité FAQs](#)

Est-il possible de chiffrer mon **\$HOME** répertoire ?

Non, il n'est pas possible de chiffrer votre **\$HOME** répertoire avec votre propre clé. Mais CloudShell chiffre le contenu de votre **\$HOME** répertoire tout en le stockant dans Amazon S3.

[Retour à la liste des mesures de sécurité FAQs](#)

Puis-je lancer une analyse antivirus sur mon **\$HOME** répertoire ?

À l'heure actuelle, il n'est pas possible d'exécuter une analyse antivirus de votre **\$HOME** répertoire. Support pour cette fonctionnalité est en cours de révision.

[Retour à la liste des mesures de sécurité FAQs](#)

Puis-je restreindre l'entrée ou la sortie de données pour moi ? CloudShell

Pour limiter les entrées ou les sorties, nous vous recommandons d'utiliser un CloudShell VPC environnement. Le **\$HOME** répertoire d'un VPC environnement est supprimé lorsque le délai d'expiration de votre VPC environnement est expiré (après 20 à 30 minutes d'inactivité), ou lorsque vous supprimez ou redémarrez votre environnement. Dans le menu Actions, les options de chargement et de téléchargement ne sont pas disponibles pour les VPC environnements.

[Retour à la liste des mesures de sécurité FAQs](#)

AWS CloudShell environnement informatique : spécifications et logiciels

Lors du lancement AWS CloudShell, un environnement informatique basé sur [Amazon Linux 2023](#) est créé pour héberger l'expérience shell. L'environnement est configuré avec [des ressources de calcul \(v CPU et mémoire\)](#) et fournit une large gamme de [logiciels préinstallés](#) accessibles depuis l'interface de ligne de commande. Assurez-vous que tous les logiciels que vous installez dans l'environnement informatique sont corrigés et à jour. Vous pouvez également configurer votre environnement par défaut en installant un logiciel et en modifiant les scripts shell.

Ressources de l'environnement informatique

Les ressources de mémoire CPU et de mémoire suivantes sont attribuées à chaque environnement de AWS CloudShell calcul :

- 1 v CPU (unité centrale virtuelle)
- 2 Go RAM

De plus, l'environnement est provisionné avec la configuration de stockage suivante :

- 1 Go de stockage persistant (le stockage persiste après la fin de la session)

Pour de plus amples informations, veuillez consulter [Stockage permanent](#).

CloudShell exigences du réseau

WebSockets

CloudShell dépend du WebSocket protocole, qui permet une communication interactive bidirectionnelle entre le navigateur Web de l'utilisateur et le CloudShell service dans le AWS Cloud. Si vous utilisez un navigateur sur un réseau privé, l'accès sécurisé à Internet est probablement facilité par des serveurs proxy et des pare-feux. WebSocket les communications peuvent généralement traverser les serveurs proxy sans problème. Mais dans certains cas, les serveurs proxy ne WebSockets peuvent pas fonctionner correctement. Si ce problème se produit, votre CloudShell interface signale l'erreur suivante :`Failed to open sessions : Timed out while opening the session.`

Si cette erreur se produit à plusieurs reprises, consultez la documentation de votre serveur proxy pour vous assurer qu'il est configuré pour autoriser WebSockets. Vous pouvez également contacter l'administrateur système de votre réseau.

Note

Si vous souhaitez définir des autorisations détaillées en répertoriant des autorisations spécifiquesURLs, vous pouvez ajouter une partie de celles URL que la AWS Systems Manager session utilise pour ouvrir une WebSocket connexion permettant d'envoyer des entrées et de recevoir des sorties. (Vos AWS CloudShell commandes sont envoyées à cette session Systems Manager.)

Le format StreamUrl utilisé par Systems Manager est `wss://ssmmessages.region.amazonaws.com/v1/data-channel/session-id?stream=(input|output)`.

La région représente l'identifiant de région d'une AWS région prise en charge par AWS Systems Manager, par exemple `us-east-2` pour la région USA Est (Ohio).

L'identifiant de session étant créé après le démarrage réussi d'une session Systems Manager spécifique, vous ne pouvez le spécifier que `wss://ssmmessages.region.amazonaws.com` lors de la mise à jour de votre liste d'autorisations. URL Pour plus d'informations, voir l'[StartSession](#) opération dans la AWS Systems Manager API référence.

Logiciel préinstallé

Note

L'environnement de AWS CloudShell développement étant régulièrement mis à jour pour permettre l'accès aux derniers logiciels, nous ne fournissons pas de numéros de version spécifiques dans cette documentation. Nous décrivons plutôt comment vous pouvez vérifier quelle version est installée. Pour vérifier la version installée, entrez le nom du programme suivi de l'`--version` option (par exemple, `git --version`).

Coquillages

Coques préinstallées

Name (Nom)	Description	Version information
Bash	Le shell Bash est l'application shell par défaut pour AWS CloudShell.	<code>bash --version</code>
PowerShell (mousqueton)	Offrant une interface de ligne de commande et un support de langage de script, PowerShell il s'appuie sur celui de Microsoft. NETExécution du langage de commande. PowerShell utilise des commandes légères appelées cmdlets accept et return. NETobjets.	<code>pwsh --version</code>
Coque Z (zsh)	Le Z Shell, également connu sous le nom de zsh, est une version étendue du Bourne Shell qui offre un support de personnalisation amélioré pour les thèmes et les plugins.	<code>zsh --version</code>

AWS interfaces de ligne de commande (CLI)

CLI

Name (Nom)	Description	Version information
AWS CDK Boîte à outils CLI	La AWS CDK boîte à outils, la CLI <code>commandcdk</code> , est le principal outil qui interagit avec votre AWS CDK application.	<code>cdk --version</code>

Name (Nom)	Description	Version information
	<p>Il exécute votre application, interroge le modèle d'application que vous avez défini, produit et déploie les AWS CloudFormation modèles générés par le. AWS CDK</p> <p>Pour plus d'informations, consultez la section AWS CDK Boîte à outils.</p>	
AWS CLI	<p>AWS CLI Il s'agit d'une interface de ligne de commande que vous pouvez utiliser pour gérer plusieurs AWS services à partir de la ligne de commande et les automatiser à l'aide de scripts. Pour de plus amples informations, veuillez consulter Travailler avec des AWS services dans AWS CloudShell.</p> <p>Pour plus d'informations sur la manière dont vous pouvez vous assurer que vous utilisez la up-to-date AWS CLI version 2 au maximum, voir Installation AWS CLI dans votre répertoire personnel.</p>	<pre>aws --version</pre>

Name (Nom)	Description	Version information
EB CLI	<p>AWS Elastic Beanstalk CLI Fournit une interface de ligne de commande pour simplifier la création, la mise à jour et la surveillance d'environnements à partir d'un référentiel local.</p> <p>Pour plus d'informations, consultez la section Utilisation de l'interface de ligne de commande CLI (EB) d'Elastic Beanstalk dans le manuel du développeur. AWS Elastic Beanstalk</p>	<code>eb --version</code>
Amazon ECS CLI	<p>L'interface de ligne de commande Amazon Elastic Container Service (AmazonECS) (CLI) fournit des commandes de haut niveau pour simplifier la création, la mise à jour et le suivi des clusters et des tâches.</p> <p>Pour plus d'informations, consultez la section Utilisation de l'interface de ligne de commande Amazon ECS dans le manuel du développeur Amazon Elastic Container Service.</p>	<code>ecs-cli --version</code>

Name (Nom)	Description	Version information
AWS SAM CLI	<p>AWS SAM CLI est un outil de ligne de commande qui fonctionne sur un AWS Serverless Application Model modèle et un code d'application. Vous pouvez effectuer plusieurs tâches. Il s'agit notamment d'invoquer des fonctions Lambda localement, de créer un package de déploiement pour votre application sans serveur et de déployer votre application sans serveur dans le cloud.</p> <p>AWS</p> <p>Pour plus d'informations, consultez la référence des AWS SAM CLI commandes dans le Guide du AWS Serverless Application Model développeur.</p>	<pre>sam --version</pre>

Name (Nom)	Description	Version information
AWS Tools for PowerShell	<p>Ces AWS Tools for PowerShell sont des PowerShell modules basés sur les fonctionnalités exposées par le AWS SDK for .NET. Avec AWS Tools for PowerShell, vous pouvez scripter des opérations sur vos AWS ressources à partir de la ligne de PowerShell commande.</p> <p>AWS CloudShell préinstalle la version modulaire (AWS.Tools) du. AWS Tools for PowerShell</p> <p>Pour plus d'informations, reportez-vous à la section Utilisation AWS des outils PowerShell dans le guide de AWS Tools for PowerShell l'utilisateur.</p>	<pre>pwsh --Command ' Get-Module -ListAvailable -Name AWS.Tools .Common '</pre>

Runtimes et AWS SDKs : Node.js et Python 3

Runtimes et AWS SDKs

Name (Nom)	Description	Version information
Node.js (avec npm)	<p>Node.js est un JavaScript environnement d'exécution conçu pour faciliter l'application de techniques de programmation asynchrones. Pour plus d'informations,</p>	<ul style="list-style-type: none"> Node.js : <code>node --version</code> npm : <code>npm --version</code>

Name (Nom)	Description	Version information
	<p>consultez la documentation sur le site officiel de Node.js.</p> <p>npm est un gestionnaire de paquets qui donne accès à un registre de JavaScript modules en ligne. Pour plus d'informations, consultez la documentation sur le site officiel de npm.</p>	
SDKpour JavaScript dans Node.js	<p>Le kit de développement logiciel (SDK) permet de simplifier le codage en fournissant JavaScript des objets pour AWS des services tels qu'Amazon S3, AmazonEC2, DynamoDB et Amazon. SWF Pour plus d'informations, consultez le Guide du développeur AWS SDK for JavaScript.</p>	<pre>npm -g ls --depth 0 2>/dev/null grep aws-sdk</pre>

Name (Nom)	Description	Version information
Python	<p>Python 3 est prêt à être utilisé dans l'environnement shell. Python 3 est désormais considéré comme la version par défaut du langage de programmation (le support de Python 2 a pris fin en janvier 2020). Pour plus d'informations, consultez la documentation sur le site officiel de Python.</p> <p>De plus, pip, le programme d'installation du package pour Python, est préinstallé. Vous pouvez utiliser ce programme en ligne de commande pour installer des packages Python à partir d'index en ligne tels que le Python Package Index. Pour plus d'informations, consultez la documentation fournie par la Python Packaging Authority.</p>	<ul style="list-style-type: none">• Python 3 : <code>python3 --version</code>• pépin : <code>pip3 --version</code>

Name (Nom)	Description	Version information
SDKpour Python (Boto3)	<p>Boto est le kit de développement logiciel (SDK) que les développeurs Python utilisent pour créer, configurer et gérer Services AWS, comme Amazon EC2 et Amazon S3. SDKII fournit un accès easy-to-use orienté objet API ainsi qu'un accès de bas niveau à Services AWS</p> <p>Pour plus d'informations, consultez la documentation de Boto3.</p>	<code>pip3 list grep boto3</code>

Outils de développement et utilitaires shell

Outils de développement et utilitaires shell

Name (Nom)	Description	Version information
bash-completion	<p>bash-completion est un ensemble de fonctions shell qui permettent l'autocomplétion de commandes ou d'arguments partiellement saisis en appuyant sur la touche Tab. Vous pouvez trouver les packages pris en charge par bash-completion dans <code>/usr/share/bash-completion/completions</code></p>	<code>dnf info bash-completion</code>

Name (Nom)	Description	Version information
	<p>Pour configurer la saisie semi-automatique des commandes d'un package, le fichier du programme doit être source. Par exemple, pour configurer la saisie semi-automatique pour les commandes Git, ajoutez la ligne suivante <code>.bashrc</code> afin que la fonctionnalité soit disponible chaque fois que votre AWS CloudShell session démarre :</p> <pre>source /usr/share/bash-completion/completions/git</pre> <p>Si vous souhaitez utiliser des scripts de complétion personnalisés, ajoutez-les à votre répertoire personnel persistant (<code>\$HOME</code>) et créez-les directement dans celui-ci <code>.bashrc</code>.</p> <p>Pour plus d'informations, consultez la README page du projet sur GitHub.</p>	

Name (Nom)	Description	Version information
CodeCommit utilitaire pour Git	<p>git-remote-codecommit est un utilitaire qui fournit une méthode simple pour transférer et extraire du code depuis des CodeCommit référentiels en étendant Git. Il s'agit de la méthode recommandée pour prendre en charge les connexions établies avec un accès fédéré, des fournisseurs d'identité et des informations d'identification temporaires.</p> <p>Pour plus d'informations, consultez les étapes de configuration des HTTPS connexions à AWS CodeCommit with git-remote-codecommit dans le Guide de l'utilisateur AWS CodeCommit.</p>	<pre>pip3 list grep git-remote-codecommit</pre>
Git	<p>Git est un système de contrôle de version distribué qui prend en charge les pratiques modernes de développement logiciel par le biais de flux de travail de branche et de mise en scène de contenu. Pour plus d'informations, consultez la page de documentation sur le site officiel de Git.</p>	<pre>git --version</pre>

Name (Nom)	Description	Version information
iputils	Le paquet iputils contient des utilitaires pour les réseaux Linux. Pour plus d'informations sur les utilitaires fournis, consultez le référentiel iputils sur GitHub	Exemples d'outil iputils : <code>arping -V</code>
jq	L'utilitaire jq analyse les données JSON formatées pour produire une sortie modifiée par des filtres de ligne de commande. Pour plus d'informations, consultez le manuel jq hébergé sur GitHub .	<code>jq --version</code>
kubectl	kubectl est un outil en ligne de commande permettant de communiquer avec le plan de contrôle d'un cluster Kubernetes, à l'aide de l'API Kubernetes.	<code>kubectl --version</code>
make	L'utilitaire make permet de automatiser des ensembles de tâches et d'organiser la compilation du code. Pour plus d'informations, consultez la documentation GNU Make .	<code>make --version</code>

Name (Nom)	Description	Version information
man	La commande man fournit des pages de manuel pour les utilitaires et outils de ligne de commande. man 1sRenvoie, par exemple, la page de manuel de la 1s commande répertoriant le contenu des répertoires. Pour plus d'informations, consultez l' entrée Wikipédia sur la page de manuel .	man --version
nano	nano est un petit éditeur convivial pour une interface texte. Pour plus d'informations, consultez la documentation sur les GNU nanotechnologies .	nano --version
procps	procps est un utilitaire d'administration système que vous pouvez utiliser pour surveiller et arrêter les processus en cours d'exécution. Pour plus d'informations, consultez le README fichier qui répertorie les programmes pouvant être exécutés avec procps .	ps --version

Name (Nom)	Description	Version information
SSHclient	SSHles clients utilisent le protocole Secure Shell pour les communications cryptées avec un ordinateur distant. Open SSH est le SSH client préinstallé. Pour plus d'informations, consultez le SSHsite Open géré par OpenBSD.	ssh -V
sudo	Avec l'utilitaire sudo, les utilisateurs peuvent exécuter un programme avec les autorisations de sécurité d'un autre utilisateur, généralement le superutilisateur. Sudo est utile lorsque vous devez installer des applications en tant qu'administrateur système. Pour plus d'informations, consultez le manuel de Sudo .	sudo --version
tar	tar est un utilitaire de ligne de commande que vous pouvez utiliser pour regrouper plusieurs fichiers dans un seul fichier d'archive (souvent appelé tarball). Pour plus d'informations, consultez la documentation GNU tar .	tar --version

Name (Nom)	Description	Version information
tmux	tmux est un multiplexeur de terminaux que vous pouvez utiliser pour exécuter différents programmes simultanément dans plusieurs fenêtres. Pour plus d'informations, consultez un blog qui fournit une introduction concise à tmux .	tmux -V
unzip	Pour plus d'informations, voir zip/unzip .	
vim	vim est un éditeur personnalisable avec lequel vous pouvez interagir via une interface texte. Pour plus d'informations, consultez les ressources de documentation fournies sur vim.org .	vim --version
wget	wget est un programme informatique utilisé pour récupérer du contenu à partir de serveurs Web spécifiés par des points de terminaison dans la ligne de commande. Pour plus d'informations, consultez la documentation de GNU Wget .	wget --version

Name (Nom)	Description	Version information
zip/décompressez	<p>Les utilitaires zip/unzip utilisent un format de fichier d'archive qui permet une compression des données sans perte de données. Appelez la commande zip pour regrouper et compresser des fichiers dans une seule archive. Utilisez unzip pour extraire des fichiers d'une archive dans un répertoire spécifique.</p>	<pre>unzip --version zip --version</pre>

Name (Nom)	Description	Version information
Docker	<p>Docker est une plateforme ouverte pour le développement, l'expédition et l'exécution d'applications. Docker vous permet de séparer vos applications de votre infrastructure afin de pouvoir fournir des logiciels rapidement. Il vous permet de créer des Dockerfiles à l'intérieur AWS CloudShell et de créer des actifs Docker avec CDK. Pour plus d'informations sur les AWS régions prises en charge par Docker, consultez la section AWS Régions prises en charge pour AWS CloudShell. Vous devez savoir que Docker dispose d'un espace limité dans l'environnement. Si vous avez de grandes images individuelles ou un trop grand nombre d'images Docker préexistantes, cela peut entraîner des problèmes. Pour plus d'informations sur Docker, consultez le guide de documentation Docker.</p>	<code>docker --version</code>

Installation AWS CLI dans votre répertoire personnel

Comme le reste des logiciels préinstallés dans votre CloudShell environnement, l' AWS CLI outil est mis à jour automatiquement avec des mises à niveau et des correctifs de sécurité planifiés. Pour

vous assurer que vous disposez de la up-to-date version la plus complète de AWS CLI, vous pouvez choisir d'installer manuellement l'outil dans le répertoire de base du shell.

Important

Vous devez installer manuellement votre copie de AWS CLI dans le répertoire de base afin qu'elle soit disponible au prochain démarrage d'une CloudShell session. Cette installation est nécessaire car les fichiers ajoutés à des répertoires situés en dehors de \$HOME sont supprimés une fois que vous avez terminé une session shell. De plus, après avoir installé cette copie de AWS CLI, elle n'est pas automatiquement mise à jour. En d'autres termes, il est de votre responsabilité de gérer les mises à jour et les correctifs de sécurité.

Pour plus d'informations sur le modèle de responsabilité AWS partagée, consultez [Protection des données dans AWS CloudShell](#).

Pour installer AWS CLI

1. Dans la ligne de CloudShell commande, utilisez la `curl` commande pour transférer une copie compressée du fichier AWS CLI installé dans le shell :

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

2. Décompressez le dossier zippé :

```
unzip awscliv2.zip
```

3. Pour ajouter l'outil dans un dossier spécifique, exécutez le AWS CLI programme d'installation :

```
sudo ./aws/install --install-dir /home/cloudshell-user/usr/local/aws-cli --bin-dir /home/cloudshell-user/usr/local/bin
```

S'il est correctement installé, la ligne de commande affiche le message suivant :

```
You can now run: /home/cloudshell-user/usr/local/bin/aws --version
```

4. Pour votre commodité, nous vous recommandons également de mettre à jour la variable d'PATHenvironnement afin de ne pas avoir à spécifier le chemin d'installation de l'outil lorsque vous exécutez des `aws` commandes :

```
export PATH=/home/cloudshell-user/usr/local/bin:$PATH
```

Note

Si vous annulez cette modification `PATH`, les `aws` commandes qui ne comportent pas de chemin spécifié utilisent la version préinstallée de AWS CLI par défaut.

Installation de logiciels tiers sur votre environnement shell

Note

Nous vous recommandons de consulter le [modèle de responsabilité partagée en matière de sécurité](#) avant d'installer des applications tierces dans l'environnement informatique AWS CloudShell de l'entreprise.

Par défaut, tous les AWS CloudShell utilisateurs disposent d'autorisations `sudo`. Par conséquent, vous pouvez utiliser la `sudo` commande pour installer un logiciel qui n'est pas encore disponible dans l'environnement informatique du shell. Par exemple, vous pouvez utiliser l'utilitaire `sudo` de DNF gestion des packages pour l'installer `cowsay`, qui génère des photos ASCII d'art d'une vache avec un message :

```
sudo dnf install cowsay
```

Vous pouvez ensuite lancer le programme nouvellement installé en tapant `echo "Welcome to AWS CloudShell" | cowsay`.

Important

Les utilitaires de gestion de packages tels que le `dnf` installent des programmes dans des répertoires (`/usr/bin` par exemple), qui sont recyclés à la fin de votre session shell. Cela signifie que des logiciels supplémentaires sont installés et utilisés par session.

Modifier votre shell à l'aide de scripts

Si vous souhaitez modifier l'environnement shell par défaut, vous pouvez modifier un script shell qui s'exécute à chaque démarrage de l'environnement shell. Le `.bashrc` script s'exécute chaque fois que le shell bash par défaut démarre.

Warning

Si vous modifiez votre `.bashrc` fichier de manière incorrecte, il se peut que vous ne puissiez pas accéder à votre environnement shell par la suite. Il est recommandé de faire une copie du fichier avant de le modifier. Vous pouvez également atténuer les risques en ouvrant deux coques lors de la modification `.bashrc`. Si vous perdez l'accès à un shell, vous êtes toujours connecté à l'autre shell et vous pouvez annuler toute modification.

Si vous perdez l'accès suite à une modification incorrecte `.bashrc` ou à tout autre fichier, vous pouvez AWS CloudShell rétablir ses paramètres par défaut en [supprimant votre répertoire personnel](#).

Au cours de la procédure, vous allez modifier le `.bashrc` script afin que votre environnement shell passe automatiquement à l'exécution du shell Z.

1. Ouvrez le `.bashrc` à l'aide d'un éditeur de texte (Vim, par exemple) :

```
vim .bashrc
```

2. Dans l'interface de l'éditeur, appuyez sur la touche `I` pour commencer à modifier, puis ajoutez ce qui suit :

```
zsh
```

3. Pour quitter et enregistrer le `.bashrc` fichier modifié, appuyez `Esc` pour passer en mode de commande Vim et entrez ce qui suit :

```
:wq
```

4. Utilisez la `source` commande pour recharger le `.bashrc` fichier :

```
source .bashrc
```

Lorsque l'interface de ligne de commande est de nouveau disponible, le symbole d'invite a été modifié % pour indiquer que vous utilisez désormais le shell Z.

AWS CloudShell migration de vers AL2 023 AL2

AWS CloudShell, qui était basé sur Amazon Linux 2 (AL2), a migré vers Amazon Linux 2023 (AL2023). Pour plus d'informations sur la AL2 version 023, consultez la section [Qu'est-ce qu'Amazon Linux 2023 \(AL2023\)](#) dans le guide de l'utilisateur Amazon Linux 2023.

Avec AL2 023, vous pouvez continuer à accéder à votre CloudShell environnement existant avec tous les outils fournis par CloudShell. Pour plus d'informations sur les outils disponibles, consultez la section [Logiciels préinstallés](#).

AL2023 apporte plusieurs améliorations aux outils de développement, notamment des versions plus récentes de packages tels que Node.js 18 et Python 3.9.

Note

En AL2 203, Python 2 n'est plus expédié avec votre CloudShell environnement.

Pour plus d'informations sur les principales différences entre AL2 et AL2 023, consultez la section [Comparaison entre Amazon Linux 2 et Amazon Linux 2023](#) dans le guide de l'utilisateur Amazon Linux 2023.

Si vous avez des questions, contactez [AWS Support](#). Vous pouvez également rechercher des réponses et publier des questions dans [AWS re:Post](#). Lorsque vous entrez AWS re:Post, vous devrez peut-être vous connecter à AWS.

AWS CloudShell Migration FAQs

Vous trouverez ci-dessous les réponses aux questions les plus fréquemment posées sur la migration AL2 de AL2 023 avec AWS CloudShell.

- [Cette migration affectera-t-elle mes autres AWS ressources, telles que les EC2 instances Amazon exécutées sur celles-ci AL2 ?](#)
- [Quels sont les packages qui seront modifiés lors de la migration vers AL2 023 ?](#)

- [Puis-je me désinscrire de la migration ?](#)
- [Puis-je créer une sauvegarde de mon AWS CloudShell environnement ?](#)

La migration vers AL2 023 affectera-t-elle mes autres AWS ressources, telles que les EC2 instances Amazon exécutées sur celles-ci ? AL2

Aucun service ou ressource autre que votre AWS CloudShell environnement n'est affecté par cette migration. Cela inclut les ressources que vous avez peut-être créées ou auxquelles vous avez accédé depuis l'intérieur AWS CloudShell. Par exemple, si vous avez créé une EC2 instance Amazon exécutée sur AL2 celle-ci, elle ne sera pas migrée vers AL2 023.

Quels sont les packages qui ont été modifiés lors de la migration vers AL2 023 ?

AWS CloudShell les environnements incluent actuellement des logiciels préinstallés. Pour en savoir plus sur la liste complète des logiciels préinstallés, consultez la section Logiciels [préinstallés](#). AWS CloudShell continuera à fournir ces packages, à l'exception de Python 2. Pour connaître la différence complète entre les packages fournis par AL2 et AL2 023, voir [Comparaison AL2 et AL2 023](#). Pour les clients ayant des exigences spécifiques en matière de package et de version qui ne seront plus satisfaites après la migration vers la version AL2 023, nous recommandons de contacter le AWS Support pour soumettre une demande.

Puis-je me désinscrire de la migration ?

Non, vous ne pouvez pas vous désinscrire de la migration. AWS CloudShell les environnements sont gérés par AWS, par conséquent, tous les environnements ont été mis à niveau vers AL2 023.

Puis-je créer une sauvegarde de mon AWS CloudShell environnement ?

AWS CloudShell continuera à conserver le répertoire personnel de l'utilisateur. Pour plus d'informations, consultez la section [Quotas de service et restrictions pour AWS CloudShell](#). Si des fichiers ou des configurations sont stockés dans votre dossier personnel et que vous souhaitez créer une sauvegarde pour ceux-ci, suivez l'[étape 6 : Création d'une sauvegarde du répertoire personnel](#).

Résolution des problèmes AWS CloudShell

Lors de l'utilisation AWS CloudShell, vous pouvez rencontrer des problèmes, par exemple lorsque vous lancez CloudShell ou effectuez des tâches clés à l'aide de l'interface de ligne de commande du shell. Les informations présentées dans ce chapitre expliquent comment résoudre certains des problèmes courants que vous pourriez rencontrer.

Pour obtenir des réponses à diverses questions sur CloudShell, consultez le [AWS CloudShell FAQs](#). Vous pouvez également rechercher des réponses et poser des questions dans le [forum de AWS CloudShell discussion](#). Lorsque vous accédez à ce forum, vous devrez peut-être vous connecter à AWS. Vous pouvez également [nous contacter](#) directement.

Résolution des erreurs

Lorsque vous rencontrez l'une des erreurs indexées suivantes, vous pouvez utiliser les solutions suivantes pour les résoudre.

Rubriques

- [Erreur : « Impossible de démarrer l'environnement. Pour réessayer, actualisez le navigateur ou redémarrez en sélectionnant « Actions, Redémarrer AWS CloudShell ».](#)
- [Erreur : « Impossible de démarrer l'environnement. Vous n'avez pas les autorisations requises. Demandez à votre IAM administrateur d'accorder l'accès à AWS CloudShell »](#)
- [Impossible d'accéder à AWS CloudShell la ligne de commande](#)
- [Impossible d'envoyer un ping aux adresses IP externes](#)
- [Des problèmes sont survenus lors de la préparation de votre terminal](#)
- [Les touches fléchées ne fonctionnent pas correctement dans PowerShell](#)
- [Les Web Sockets non pris en charge empêchent le démarrage des sessions CloudShell](#)
- [Impossible d'importer le AWSPowerShell.NetCore module](#)
- [Docker ne fonctionne pas lors de l'utilisation AWS CloudShell](#)
- [Docker n'a plus d'espace disque](#)
- [docker push le délai imparti est dépassé et continue de réessayer](#)
- [Impossible d'accéder aux ressources VPC depuis mon AWS CloudShell VPC environnement](#)
- [Le ENI fichier utilisé par AWS CloudShell pour mon VPC environnement n'est pas nettoyé](#)

- [Les utilisateurs CreateEnvironment autorisés uniquement à accéder aux VPC environnements ont également accès aux AWS CloudShell environnements publics](#)
- [Les informations d'identification ne fonctionnent pas dans CloudShell](#)

Erreur : « Impossible de démarrer l'environnement. Pour réessayer, actualisez le navigateur ou redémarrez en sélectionnant « Actions, Redémarrer AWS CloudShell ».

Problème : Lorsque vous tentez AWS CloudShell de lancer depuis le AWS Management Console, l'accès vous est refusé même après avoir demandé l'autorisation de votre IAM administrateur et après avoir actualisé ou redémarré CloudShell votre navigateur.

Solution : contactez le [AWS Support](#).

[\(haut de la page\)](#)

Erreur : « Impossible de démarrer l'environnement. Vous n'avez pas les autorisations requises. Demandez à votre IAM administrateur d'accorder l'accès à AWS CloudShell »

Problème : Lorsque vous tentez AWS CloudShell de lancer depuis le AWS Management Console, l'accès vous est refusé et vous êtes informé que vous ne disposez pas des autorisations requises.

Cause : L'IAM identité que vous utilisez pour accéder AWS CloudShell ne dispose pas des IAM autorisations nécessaires.

Solution : demandez à votre IAM administrateur de vous fournir les autorisations nécessaires. Ils peuvent le faire soit en ajoutant une stratégie AWS gérée attachée (AWSCloudShellFullAccess), soit en ajoutant une politique intégrée. Pour de plus amples informations, veuillez consulter [Gestion de AWS CloudShell l'accès et de l'utilisation à l'aide IAM de politiques](#).

[\(haut de la page\)](#)

Impossible d'accéder à AWS CloudShell la ligne de commande

Problème : après avoir modifié un fichier utilisé par l'environnement informatique, vous ne pouvez pas accéder à la ligne de commande dans AWS CloudShell.

Solution : Si vous perdez l'accès suite à une modification incorrecte `.bashrc` ou à tout autre fichier, vous pouvez AWS CloudShell rétablir ses paramètres par défaut en [supprimant votre répertoire personnel](#).

[\(haut de la page\)](#)

Impossible d'envoyer un ping aux adresses IP externes

Problème : Lorsque vous exécutez une commande ping depuis la ligne de commande (par exemple, `ping amazon.com`), vous recevez le message suivant.

```
ping: socket: Operation not permitted
```

Cause : L'utilitaire ping utilise le protocole Internet Control Message Protocol (ICMP) pour envoyer des paquets de requêtes d'écho à un hôte cible. Il attend la réponse d'un écho de la cible. Comme le ICMP protocole n'est pas activé dans AWS CloudShell, l'utilitaire ping ne fonctionne pas dans l'environnement informatique du shell.

Solution : Étant donné que cela n'ICMP est pas pris en charge dans AWS CloudShell, vous pouvez exécuter la commande suivante pour installer Netcat. Netcat est un utilitaire de réseau informatique permettant de lire et d'écrire sur des connexions réseau à l'aide de TCP ou UDP.

```
sudo yum install nc
nc -zv www.amazon.com 443
```

[\(haut de la page\)](#)

Des problèmes sont survenus lors de la préparation de votre terminal

Problème : lorsque vous essayez d'accéder à l' AWS CloudShell aide du navigateur Microsoft Edge, vous ne pouvez pas démarrer de session shell et le navigateur affiche un message d'erreur.

Cause : AWS CloudShell n'est pas compatible avec les versions antérieures de Microsoft Edge. Vous pouvez y accéder AWS CloudShell en utilisant les quatre dernières versions majeures des navigateurs pris en charge.

Solution : installez une version mise à jour du navigateur Edge à partir du [site Microsoft](#).

[\(haut de la page\)](#)

Les touches fléchées ne fonctionnent pas correctement dans PowerShell

Problème : En fonctionnement normal, vous pouvez utiliser les touches fléchées pour naviguer dans l'interface de ligne de commande et parcourir l'historique de vos commandes dans les deux sens. Toutefois, lorsque vous appuyez sur les touches fléchées dans certaines versions de PowerShell ON AWS CloudShell, les lettres peuvent être mal sorties.

Cause : La situation dans laquelle les touches fléchées affichent des lettres de manière incorrecte est un problème connu avec les versions PowerShell 7.2.x exécutées sous Linux.

Solution : pour supprimer les séquences d'échappement qui modifient le comportement des touches fléchées, modifiez le fichier de PowerShell profil et définissez la `$PSStyle` variable sur `PlainText`.

1. Dans la ligne de AWS CloudShell commande, entrez la commande suivante pour ouvrir le fichier de profil.

```
vim ~/.config/powershell/Microsoft.PowerShell_profile.ps1
```

Note

Si vous êtes déjà connecté PowerShell, vous pouvez également ouvrir le fichier de profil dans l'éditeur à l'aide de la commande suivante.

```
vim $PROFILE
```

2. Dans l'éditeur, allez à la fin du texte existant du fichier, appuyez sur `i` pour passer en mode Insertion, puis ajoutez l'instruction suivante.

```
$PSStyle.OutputRendering = 'PlainText'
```

3. Après avoir effectué la modification, appuyez sur `Esc` ce bouton pour passer en mode commande. Entrez ensuite la commande suivante pour enregistrer le fichier et quitter l'éditeur.

```
:wq
```

Note

Vos modifications prendront effet au prochain démarrage PowerShell.

[\(haut de la page\)](#)

Les Web Sockets non pris en charge empêchent le démarrage des sessions CloudShell

Problème : Lorsque vous essayez de démarrer AWS CloudShell, vous recevez à plusieurs reprises le message suivant : `Failed to open sessions : Timed out while opening the session`

Cause : CloudShell dépend du WebSocket protocole, qui permet une communication interactive bidirectionnelle entre votre navigateur Web et AWS CloudShell. Si vous utilisez un navigateur sur un réseau privé, l'accès sécurisé à Internet est probablement facilité par des serveurs proxy et des pare-feux. WebSocket les communications peuvent généralement traverser les serveurs proxy sans problème. Mais, dans certains cas, les serveurs proxy ne WebSockets peuvent pas fonctionner correctement. Si ce problème se produit, CloudShell impossible de démarrer une session shell et la tentative de connexion finit par expirer.

Solution : Un délai d'expiration de connexion peut être dû à un problème autre qu'un problème non WebSockets pris en charge. Dans ce cas, actualisez d'abord la fenêtre du navigateur dans laquelle se trouve l'interface de ligne de CloudShell commande.

Si des erreurs de temporisation persistent après l'actualisation, consultez la documentation de votre serveur proxy. Assurez-vous également que votre serveur proxy est configuré pour autoriser les Web Sockets. Vous pouvez également contacter l'administrateur système de votre réseau.

Note

Supposons que vous souhaitiez définir des autorisations détaillées en indiquant des autorisations spécifiques. URLs Vous pouvez ajouter une partie de URL ce que la AWS Systems Manager session utilise pour ouvrir une WebSocket connexion afin d'envoyer des entrées et de recevoir des sorties. Vos AWS CloudShell commandes sont envoyées à cette session Systems Manager.

Le format utilisé par Systems Manager est `wss://ssmmessages.region.amazonaws.com/v1/data-channel/session-id?stream=(input|output)`. StreamUrl

La région représente l'identifiant de région pour une Région AWS région prise en charge par AWS Systems Manager. Par exemple, `us-east-2` est l'identifiant de région pour la région USA Est (Ohio).

L'identifiant de session étant créé après le démarrage réussi d'une session Systems Manager spécifique, vous ne pouvez le spécifier que `wss://ssmmessages.region.amazonaws.com` lorsque vous mettez à jour votre liste d'autorisations. URL Pour plus d'informations, voir l'[StartSession](#) opération dans la AWS Systems Manager API référence.

[\(haut de la page\)](#)

Impossible d'importer le **AWSPowerShell.NetCore** module

Problème : Lorsque vous importez le `AWSPowerShell.NetCore` module in PowerShell by `Import-Module -Name AWSPowerShell.NetCore`, vous recevez le message d'erreur suivant :

Import-Module : le module spécifié '`AWSPowerShell.NetCore`' n'a pas été chargé car aucun fichier de module valide n'a été trouvé dans aucun répertoire de modules.

Cause : Le `AWSPowerShell.NetCore` module est remplacé par les modules `AWS.Tools` par service dans `AWS CloudShell`

Solution : Il est possible que les instructions d'importation explicites ne soient plus requises ou qu'il soit nécessaire de les remplacer par le module `AWS.Tools` par service correspondant.

Exemple

Exemple

- Dans la plupart des cas, tant qu'aucun type `.Net` n'est utilisé, aucune instruction d'importation explicite n'est nécessaire. Voici des exemples de déclarations d'importation.
 - `Get-S3Bucket`
 - `(Get-EC2Instance).Instances`
- Si des types `.Net` sont utilisés, importez le module de niveau de service `(AWS.Tools.<Service>)`. Voici un exemple de syntaxe.

```
Import-Module -Name AWS.Tools.EC2
```

```
$InstanceTag = [Amazon.EC2.Model.Tag]::new("Environment", "Dev")
```

```
Import-Module -Name AWS.Tools.S3  
$LifecycleRule = [Amazon.S3.Model.LifecycleRule]::new()
```

Pour plus d'informations, consultez l'[annonce de la version 4](#) du AWS Tools for PowerShell.

[\(haut de la page\)](#)

Docker ne fonctionne pas lors de l'utilisation AWS CloudShell

Problème : Docker ne fonctionne pas correctement lors de l'utilisation AWS CloudShell. Le message d'erreur suivant s'affiche :`docker: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?.`

Solution : essayez de redémarrer votre environnement. Ce message d'erreur peut se produire lorsque vous exécutez Docker AWS CloudShell dans une GovCloud région qui ne le prend pas en charge. Assurez-vous que vous utilisez Docker dans les AWS régions prises en charge. Pour obtenir la liste des régions dans lesquelles Docker est disponible, consultez la section [AWS Régions prises en charge](#) pour AWS CloudShell

Docker n'a plus d'espace disque

Problème : Vous recevez le message d'erreur suivant :`ERROR: failed to solve: failed to register layer: write [...]: no space left on device.`

Cause : Le Dockerfile dépasse l'espace disque disponible en AWS CloudShell Cela peut être dû à de grandes images individuelles ou à un trop grand nombre d'images Docker préexistantes.

Solution : Exécutez `df -h` pour connaître l'utilisation du disque. Exécutez `sudo du -sh /folder/folder1` pour évaluer la taille de certains dossiers qui vous semblent volumineux et envisagez de supprimer d'autres fichiers pour libérer de l'espace. Une option serait d'envisager de supprimer les images Docker inutilisées en exécutant `docker rmi`. Vous devez savoir que Docker dispose d'un espace limité dans l'environnement. Pour plus d'informations sur Docker, consultez le guide de documentation [Docker](#).

docker push le délai imparti est dépassé et continue de réessayer

Problème : Lorsque vous l'exécutez `docker push`, le délai imparti est expiré et continue de réessayer sans succès.

Cause : Cela peut être dû à des autorisations manquantes, à un transfert vers le mauvais référentiel ou à un manque d'authentification.

Solution : Pour essayer de résoudre ce problème, assurez-vous d'effectuer le transfert vers le bon référentiel. Exécutez `docker login` pour vous authentifier correctement. Assurez-vous de disposer de toutes les autorisations requises pour le transfert vers un ECR référentiel Amazon.

Impossible d'accéder aux ressources VPC depuis mon AWS CloudShell VPC environnement

Problème : Impossible d'accéder aux ressources VPC pendant que j'utilise mon AWS CloudShell VPC environnement.

Cause : Votre AWS CloudShell VPC environnement hérite des paramètres réseau de votre VPC.

Solution : Pour résoudre ce problème, assurez-vous VPC que vous êtes correctement configuré pour accéder à vos ressources. Pour plus d'informations, consultez VPC la documentation [Connect your VPC to other networks](#) et la documentation [Network Access Analyzer](#). Vous pouvez trouver l'IPv4 adresse utilisée par l' AWS CloudShell VPC environnement en exécutant la commande `ip -a`` dans votre environnement dans l'invite de ligne de commande ou sur la page VPC Console.

Le ENI fichier utilisé par AWS CloudShell pour mon VPC environnement n'est pas nettoyé

Problème : Impossible de nettoyer le ENI fichier utilisé AWS CloudShell pour mon VPC environnement.

Cause : `ec2:DeleteNetworkInterface` l'autorisation n'est pas activée pour votre rôle.

Solution : pour résoudre ce problème, assurez-vous que `ec2:DeleteNetworkInterface` l'autorisation est activée pour votre rôle, comme indiqué dans l'exemple de script suivant :

```
{
  "Effect": "Allow",
```

```
"Action": [
  "ec2:DeleteNetworkInterface"
],
"Condition": {
  "StringEquals": {
    "aws:ResourceTag/ManagedByCloudShell": ""
  }
},
"Resource": "arn:aws:ec2:*:*:network-interface/*"
}
```

Les utilisateurs **CreateEnvironment** autorisés uniquement à accéder aux VPC environnements ont également accès aux AWS CloudShell environnements publics

Problème : les utilisateurs **CreateEnvironment** autorisés à accéder uniquement aux VPC environnements peuvent également accéder aux AWS CloudShell environnements publics.

Cause : Lorsque vous limitez **CreateEnvironment** les autorisations pour la création d'VPC environnements uniquement et si vous avez déjà créé un environnement public, vous conservez votre accès à l' CloudShell environnement public existant jusqu'à ce que cet environnement soit supprimé à l'aide de l'interface utilisateur Web. Mais si vous ne l'avez jamais utilisé CloudShell auparavant, vous n'aurez pas accès aux environnements publics.

Solution : pour restreindre l'accès aux AWS CloudShell environnements publics, l'IAM administrateur doit d'abord mettre à jour la IAM politique avec la restriction, puis l'utilisateur doit supprimer manuellement l'environnement public existant à l'aide de l'interface utilisateur AWS CloudShell Web. (Actions → Supprimer CloudShell l'environnement).

Les informations d'identification ne fonctionnent pas dans CloudShell

Problème : lorsque vous essayez de passer un AWS CLI appel depuis CloudShell, le message d'erreur Internal Server Error s'affiche.

Cause : Les causes possibles de ce problème sont les suivantes :

- L'`putCredentialsAPI` appel CloudShell utilisé pour actualiser les informations d'identification a échoué. L'API appel peut échouer en raison d'un manque d'IAM autorisations d'`putCredentials` action. Pour de plus amples informations, veuillez consulter [???](#). Si vous êtes

déjà IAM autorisé à `putCredentials` agir, l'API appel peut échouer en raison de problèmes de réseau ou de problèmes opérationnels liés à CloudShell.

- Vos informations d'identification ne sont plus valides car la session de AWS console a expiré, mais votre CloudShell environnement fonctionne toujours. Lorsque vos informations d'identification ne sont plus valides CloudShell, vous ne passez aucun API appel.

Solution : essayez d'actualiser la page Web si vous disposez déjà IAM des autorisations requises pour `putCredentials` agir. Si le problème n'est pas résolu et que le message d'erreur persiste, contactez le [AWS Support](#).

AWS Régions prises en charge pour AWS CloudShell

Cette section présente la liste des AWS régions prises en charge et des régions facultatives pour AWS CloudShell. Pour obtenir la liste des points de terminaison de AWS service et des quotas pour CloudShell, consultez la [AWS CloudShell page](#) du Référence générale d'Amazon Web Services.

Les AWS régions prises en charge pour Docker CloudShell, et l' CloudShell VPCenvironnement sont les suivantes :

- USA Est (Ohio)
- USA Est (Virginie du Nord)
- USA Ouest (Californie du Nord)
- USA Ouest (Oregon)
- Afrique (Le Cap)
- Asie-Pacifique (Hong Kong)
- Asie-Pacifique (Jakarta)
- Asie-Pacifique (Mumbai)
- Asie-Pacifique (Osaka)
- Asia Pacific (Seoul)
- Asie-Pacifique (Singapour)
- Asie-Pacifique (Sydney)
- Asie-Pacifique (Tokyo)
- Canada (Centre)
- Europe (Francfort)
- Europe (Irlande)
- Europe (Londres)
- Europe (Milan)
- Europe (Paris)
- Europe (Stockholm)
- Moyen-Orient (Bahreïn)
- Moyen-Orient (UAE)
- Amérique du Sud (São Paulo)

GovCloud Régions

Les GovCloud régions prises en charge sont les CloudShell suivantes :

- AWS GovCloud (USA Est)
- AWS GovCloud (US-Ouest)

Docker et son CloudShell VPC environnement ne sont actuellement pas disponibles dans GovCloud les régions.

Quotas de service et restrictions pour AWS CloudShell

Cette page décrit les quotas et restrictions de service qui s'appliquent aux domaines suivants :

- [Stockage permanent](#)
- [Utilisation mensuelle](#)
- [Taille de commande](#)
- [Coquilles simultanées](#)
- [Sessions Shell](#)
- [Accès au réseau et transfert de données](#)
- [Fichiers système et rechargements de pages](#)

Stockage permanent

Avec AWS CloudShell, vous disposez d'un stockage permanent de 1 Go pour chacun Région AWS sans frais. Le stockage permanent se trouve dans votre répertoire personnel (\$HOME) et vous est réservé. Contrairement aux ressources environnementales éphémères qui sont recyclées après la fin de chaque session du shell, les données de votre répertoire personnel persistent entre les sessions.

Note

CloudShell VPCles environnements ne disposent pas d'un stockage persistant. Le HOME répertoire \$ est supprimé lorsque votre VPC environnement expire (après 20 à 30 minutes d'inactivité) ou lorsque vous supprimez votre environnement.

Si vous arrêtez de AWS CloudShell les utiliser dans un Région AWS, les données sont conservées dans le stockage permanent de cette région pendant 120 jours après la fin de votre dernière session. Au bout de 120 jours, à moins que vous n'agissiez, vos données sont automatiquement supprimées du stockage persistant de cette région. Vous pouvez empêcher la suppression en le AWS CloudShell relançant Région AWS. Pour plus d'informations, voir [Étape 2 : sélectionner une région AWS CloudShell, lancer et choisir un shell](#).

Note

Scénario d'utilisation

Márcia avait l'habitude AWS CloudShell de stocker des fichiers dans ses répertoires personnels à deux endroits Régions AWS : l'est des États-Unis (Virginie du Nord) et l'Europe (Irlande). Elle a ensuite commencé à l'utiliser AWS CloudShell exclusivement en Europe (Irlande) et a cessé de lancer des sessions shell dans l'est des États-Unis (Virginie du Nord). Avant la date limite de suppression des données dans l'est des États-Unis (Virginie du Nord), Márcia décide d'empêcher le recyclage de son répertoire personnel en lançant AWS CloudShell et en sélectionnant à nouveau la région des États-Unis est (Virginie du Nord). Comme elle a toujours utilisé l'Europe (Irlande) pour ses sessions shell, son stockage persistant dans cette région n'est pas affecté.

Utilisation mensuelle

Chacun Région AWS de vous Compte AWS dispose d'un quota d'utilisation mensuel pour AWS CloudShell. Ce quota combine le temps total passé à l'utiliser CloudShell par tous les IAM directeurs de cette région. Si vous tentez d'y accéder CloudShell après avoir atteint le quota mensuel pour cette région, un message s'affiche pour expliquer pourquoi l'environnement shell ne peut pas être démarré.

Note

Si vous devez augmenter vos quotas d'utilisation mensuels, contactez le [AWSSupport en](#) fournissant les informations suivantes :

- CloudShell Région d'utilisation
- Votre cas d'utilisation. Par exemple, le AWS CLI fonctionnement et l'exécution de commandes Linux
- Le nombre d' CloudShell utilisateurs. Par exemple, 5-10
- L'estimation maximale du temps que vous passez CloudShell dans la région
- CloudShellVPCutilisation de l'environnement

Nous pouvons approuver l'augmentation de la durée maximale estimée à 1 000 heures par mois par rapport à la limite actuelle de 200 heures.

Taille de commande

La taille de la commande ne peut pas dépasser 65412 caractères.

Note

Si vous avez l'intention d'exécuter la commande qui dépasse 65 412 caractères, créez un script dans la langue de votre choix, puis exécutez-le depuis l'interface de ligne de commande. Pour plus d'informations sur la gamme de logiciels préinstallés accessibles depuis l'interface de ligne de commande, voir Logiciels [préinstallés](#).

Pour un exemple de création d'un script, puis de son exécution à partir de l'interface de ligne de commande, voir [Tutoriel : Commencer avec AWS CloudShell](#).

Coquilles simultanées

- Shells simultanés : vous pouvez exécuter jusqu'à 10 shells simultanément dans chacun d'eux Région AWS pour votre compte.

Sessions Shell

- Sessions inactives : AWS CloudShell environnement shell interactif. Si vous n'interagissez pas avec celui-ci à l'aide de votre clavier ou de votre pointeur pendant 20 à 30 minutes, votre session shell se termine. Les processus en cours ne sont pas considérés comme des interactions.
- Sessions de longue durée : une session shell qui s'exécute en continu pendant environ 12 heures se termine automatiquement même si l'utilisateur interagit régulièrement avec elle pendant cette période.

Accès au réseau et transfert de données

Les restrictions suivantes s'appliquent au trafic entrant et sortant de votre AWS CloudShell environnement :

- Sortant : vous pouvez accéder à l'Internet public.
- Entrant : vous ne pouvez pas accéder aux ports entrants. Aucune adresse IP publique n'est disponible.

⚠ Warning

Avec l'accès à l'Internet public, certains utilisateurs risquent d'exporter des données depuis l'AWS CloudShell environnement. Nous recommandons IAM aux administrateurs de gérer la liste autorisée des AWS CloudShell utilisateurs de confiance à l'aide IAM d'outils. Pour plus d'informations sur la manière dont l'accès peut être explicitement refusé à des utilisateurs spécifiques, consultez [Gestion des actions autorisées à AWS CloudShell l'aide de politiques personnalisées](#).

Transfert de données : le chargement et le téléchargement de fichiers depuis et vers des fichiers AWS CloudShell peuvent être lents pour les fichiers volumineux. Vous pouvez également transférer des fichiers vers votre environnement depuis un compartiment Amazon S3 à l'aide de l'interface de ligne de commande du shell.

Restrictions relatives aux fichiers système et aux rechargements de pages

- Fichiers système : si vous modifiez de manière incorrecte les fichiers requis par l'environnement informatique, vous risquez de rencontrer des problèmes lors de l'accès ou de l'utilisation de l'AWS CloudShell environnement. Dans ce cas, il se peut que vous [deviez supprimer votre répertoire personnel](#) pour y accéder de nouveau.
- Rechargement des pages : pour recharger l'AWS CloudShell interface, utilisez le bouton d'actualisation de votre navigateur au lieu de la séquence de touches de raccourci par défaut de votre système d'exploitation.

Historique du document pour le guide de AWS CloudShell l'utilisateur

Mises à jour récentes

Le tableau suivant décrit les modifications importantes apportées au Guide de l'utilisateur AWS CloudShell .

Modification	Description	Date
VPCAssistance Amazon pour AWS CloudShell certaines régions	Ajout de la prise en charge de la création et de l'utilisation d'AWS CloudShell VPCenvironnements dans certaines régions.	13 juin 2024
De nouveaux didacticiels ont été ajoutés au guide de AWS CloudShell l'utilisateur	Deux nouveaux didacticiels ont été ajoutés qui expliquent comment créer un conteneur Docker à l'intérieur AWS CloudShell et le transférer vers un ECR référentiel Amazon, et comment déployer une fonction Lambda via. AWS CDK	27 décembre 2023
Conteneurs Docker pris en charge AWS CloudShell dans certaines régions	Support pour les conteneurs Docker avec AWS CloudShell a été ajouté dans certaines régions.	27 décembre 2023
AWS CloudShell a migré pour utiliser désormais Amazon Linux 2023 (AL2023)	AWS CloudShell utilise désormais AL2 023 et a migré depuis Amazon Linux 2.	4 décembre 2023
De nouvelles AWS régions pour AWS CloudShell	AWS CloudShell est désormais généralement	16 juin 2023

disponible dans les AWS
régions suivantes :

- USA Ouest (Californie du Nord)
- Afrique (Le Cap)
- Asie-Pacifique (Hong Kong)
- Asie-Pacifique (Osaka)
- Asie-Pacifique (Séoul)
- Asie-Pacifique (Jakarta)
- Asie-Pacifique (Singapour)
- Europe (Paris)
- Europe (Stockholm)
- Europe (Milan)
- Moyen-Orient (Bahreïn)
- Moyen-Orient (UAE)

[Lancement AWS CloudShell sur Console Toolbar](#)

Lancement CloudShell sur Console Toolbar, en bas à gauche de la console en choisissant CloudShell.

28 mars 2023

[De nouvelles AWS régions pour AWS CloudShell](#)

AWS CloudShell est désormais disponible dans les AWS régions suivantes :

6 octobre 2022

- Canada (Centre)
- Europe (Londres)
- Amérique du Sud (São Paulo)

[AWS CloudShell pris en charge aux États-Unis AWS GovCloud](#)

AWS CloudShell est désormais pris en charge dans la région AWS GovCloud (États-Unis).

29 juin 2022

Sûreté FAQs	En outre, FAQs axé sur les problèmes de sécurité.	14 avril 2022
Sockets Web	Ajout d'une section aux exigences CloudShell du réseau expliquant l'utilisation du WebSocket protocole.	21 mars 2022
Résolution des problèmes liés aux touches fléchées PowerShell	Suivez les étapes pour corriger les touches fléchées qui n'affichent pas correctement les lettres lorsque vous appuyez dessus.	7 février 2022
Complétion automatique par touche de tabulation	Nouvelle documentation expliquant comment utiliser la complétion par bash, qui permet l'autocomplétion de commandes ou d'arguments partiellement saisis en appuyant sur la touche Tab.	24 septembre 2021
Spécification AWS des régions	Documentation sur la spécification des valeurs par défaut Région AWS pour AWS CLI les commandes.	11 mai 2021
Formatage dans PDF les versions Kindle	Tailles d'image et texte fixes dans les cellules du tableau.	10 mars 2021

[Version de disponibilité
générale \(GA\) AWS CloudShell
dans certaines AWS régions](#)

AWS CloudShell est désormais généralement disponible dans les AWS régions suivantes :

15 décembre 2020

- USA Est (Ohio)
- USA Est (Virginie du Nord)
- USA Ouest (Oregon)
- Asie Pacifique (Tokyo)
- Europe (Irlande)
- Asie-Pacifique (Mumbai)
- Asie-Pacifique (Sydney)
- Europe (Francfort)

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.