

Guia do usuário

AWS CloudShell



AWS CloudShell: Guia do usuário

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

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que AWS CloudShell é	1
Atributos do AWS CloudShell	1
AWS Command Line Interface	2
Shells e ferramentas de desenvolvimento	2
Armazenamento persistente	2
CloudShell VPCambientes	3
Segurança	3
Opções de personalização	4
Restauração de sessão	4
Como começar com AWS CloudShell?	4
Preços para AWS CloudShell	7
AWS CloudShell Tópicos principais	7
FAQs	8
Como faço para começar a usar AWS CloudShell?	9
O que eu preciso acessar AWS CloudShell?	9
O que está AWS CloudShell no Console Toolbar?	9
Como faço para lançar AWS CloudShell no Console Toolbar?	9
O que Regiões da AWS está AWS CloudShell disponível em?	9
Que Região da AWS é atribuído se AWS CloudShell não estiver disponível na região selecionada quando você inicia CloudShell no Console Toolbar?	10
Quais tipos de shells posso usar no AWS CloudShell?	10
Com quais navegadores da web posso usar AWS CloudShell?	10
Como faço para criar e gerenciar meu AWS CloudShell ambiente?	10
Quais navegadores da web posso usar quando inicio AWS CloudShell no Console Toolbar?	10
Posso baixar arquivos do AWS CloudShell?	11
Qual software está pré-instalado no meu ambiente de shell?	11
Posso instalar software que não esteja disponível no ambiente shell?	11
Posso restringir as ações que os usuários podem realizar no AWS CloudShell?	11
Como posso mover dados do meu diretório inicial se eu quiser alterar o Região da AWS local onde estou usando AWS CloudShell?	12
Posso aumentar o limite que determina quando o AWS CloudShell atinge o tempo limite devido à inatividade do usuário?	12
Posso acessar AWS CloudShell a AWS Console Mobile Application partir da tela inicial?	12

Como posso lançar AWS CloudShell no AWS Console Mobile Application?	13
Posso usar teclas modificadoras nos meus teclados iOS e Android ao usar AWS CloudShell no? AWS Console Mobile Application	13
Posso dividir a exibição da AWS CloudShell guia em várias guias no AWS Console Mobile Application?	13
Posso acessar AWS CloudShell no Console Toolbar em um dispositivo móvel?	13
Quais são meus custos correntes CloudShell para minha AmazonVPC?	13
Posso criar mais de dois VPC ambientes por IAM principal?	14
Conceitos básicos	15
Pré-requisitos	15
Conteúdo	16
Etapa 1: faça login em AWS Management Console	16
Etapa 2: selecione uma região AWS CloudShell, inicie e escolha um shell	19
Etapa 3: baixar um arquivo do AWS CloudShell	22
Etapa 4: fazer upload de um arquivo para AWS CloudShell	24
Etapa 5: Remover um arquivo do AWS CloudShell	25
Etapa 6: criar um backup do diretório inicial	25
Etapa 7: reiniciar uma sessão de shell	27
Etapa 8: excluir um diretório inicial da sessão de shell	28
Etapa 9: editar o código do seu arquivo e executá-lo usando a linha de comando	29
Etapa 10: Use AWS CLI para adicionar o arquivo como um objeto em um bucket do Amazon S3	30
Tópicos relacionados	32
Tutoriais	33
Tutorial: como copiar vários arquivos	33
Como carregar e baixar vários arquivos usando o Amazon S3	34
Como carregar e baixar vários arquivos usando pastas compactadas	38
Tutorial: Usando CodeCommit	39
Pré-requisitos	39
Etapa 1: criar e clonar um repositório CodeCommit	39
Etapa 2: Prepare e confirme um arquivo antes de enviá-lo ao seu repositório CodeCommit	41
Tutorial: Criação de pré-assinados URLs	42
Pré-requisitos	42
Etapa 1: criar uma IAM função para conceder acesso ao bucket do Amazon S3	42
Gere o pré-assinado URL	44

Tutorial: Construindo um contêiner Docker interno AWS CloudShell e enviando para a Amazon ECR	45
Pré-requisitos	45
Procedimento tutorial	45
Limpeza	48
Tutorial: Implantando uma função Lambda usando o AWS CDK	48
Pré-requisitos	48
Procedimento tutorial	48
Limpeza	51
Trabalhando com AWS CloudShell	52
Navegando pela interface AWS CloudShell	52
.....	52
Trabalhando em Regiões da AWS	54
Especificando seu padrão Região da AWS para AWS CLI	54
Trabalhar com arquivos e armazenamento	56
Como trabalhar com o Docker	56
Atributos de acessibilidade	58
Navegação pelo teclado em CloudShell	58
CloudShell recursos de acessibilidade do terminal	58
Escolhendo tamanhos de fonte e temas de interface em CloudShell	58
Trabalhando com AWS serviços	60
AWS CLI exemplos de linha de comando para AWS serviços selecionados	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
Personalizando AWS CloudShell	64
Divisão da exibição da linha de comando em várias guias	64
Alteração do tamanho da fonte	65
Alteração do tema da interface	65
Uso do Safe Paste para texto de várias linhas	65
O uso do tmux para restaurar a sessão	66
Usando AWS CloudShell na Amazon Virtual Private Cloud (AmazonVPC)	67

Restrições operacionais	67
Criando um CloudShell VPC ambiente	68
IAMPermissões necessárias para criar e usar CloudShell VPC ambientes	69
IAMpolítica que concede CloudShell acesso total, incluindo acesso a VPC	70
Usando chaves de IAM condição para VPC ambientes	72
Exemplo de políticas com chaves de condição para VPC configurações	73
Segurança	3
Proteção de dados	79
Criptografia de dados	80
Identity and Access Management	80
Público	81
Autenticando com identidades	82
Gerenciando acesso usando políticas	85
Como AWS CloudShell funciona com IAM	88
Exemplos de políticas baseadas em identidade	95
Solução de problemas	98
Gerenciando AWS CloudShell o acesso e o uso com IAM políticas	100
Logging e monitoramento	114
Monitorando a atividade com CloudTrail	114
AWS CloudShell em CloudTrail	115
Validação de conformidade	117
Resiliência	122
Segurança da infraestrutura	123
Melhores práticas de segurança	124
Segurança FAQs	124
Quais AWS processos e tecnologias são usados quando você inicia CloudShell e inicia uma sessão de shell?	125
É possível restringir o acesso à rede CloudShell?	125
Posso personalizar meu CloudShell ambiente?	125
Onde meu diretório \$HOME está realmente armazenado no Nuvem AWS?	126
É possível criptografar meu diretório \$HOME?	126
Posso executar uma verificação de vírus no meu diretório \$HOME?	126
Posso restringir a entrada ou saída de dados para o meu? CloudShell	126
AWS CloudShell ambiente computacional	127
Recursos do ambiente de computação	127
CloudShell requisitos de rede	127

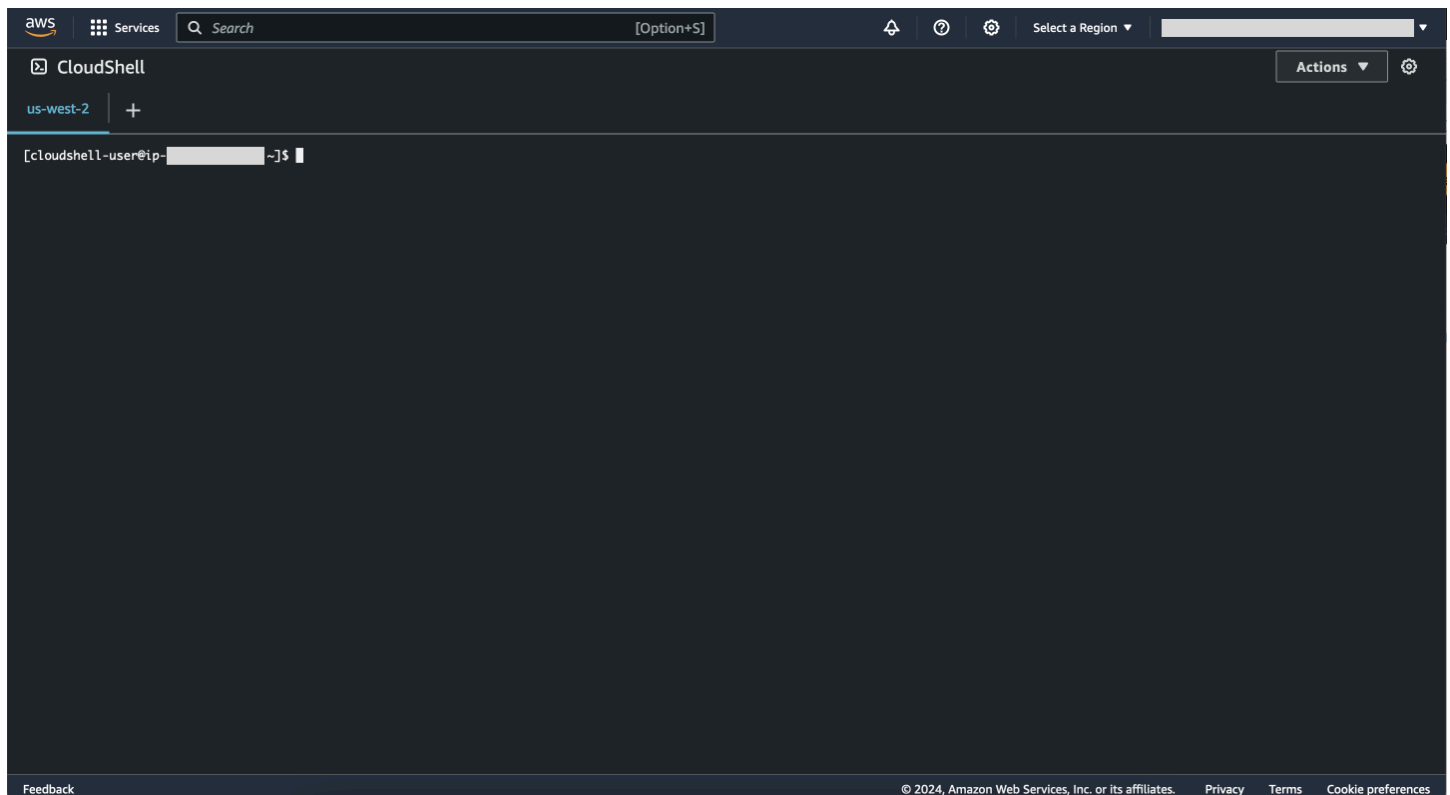
Software pré-instalado	128
Shells	129
AWS interfaces de linha de comando (CLI)	129
Tempos de execução e AWSSDKs: Node.js e Python 3	133
Ferramentas de desenvolvimento e utilitários de shell	136
AWS CLI Instalando em seu diretório inicial	144
Instalação de software de terceiros no ambiente do shell	146
Como modificar seu shell com scripts	146
Migração do Amazon Linux 2 para o Amazon Linux 2023	147
AWS CloudShell Migração FAQs	148
Solução de problemas	150
Solucionar de problemas de erros	150
Erro: “Não foi possível iniciar o ambiente. Para tentar novamente, atualize o navegador ou reinicie selecionando Ações, Reiniciar” AWS CloudShell	151
Erro: “Não foi possível iniciar o ambiente. Você não tem as permissões necessárias. Peça ao IAM administrador que conceda acesso a AWS CloudShell”	151
Não é possível acessar a linha de AWS CloudShell comando	151
Não é possível executar ping em endereços IP externos	152
Houve alguns problemas ao preparar seu terminal	152
As teclas de seta não funcionam corretamente em PowerShell	153
Web Sockets não suportados causam uma falha no início das sessões CloudShell	154
Não é possível importar o módulo <code>AWSPowerShell.NetCore</code>	155
O Docker não está em execução ao usar AWS CloudShell	156
O Docker ficou sem espaço em disco	156
<code>docker push</code> está atingindo o tempo limite e continua tentando novamente	156
Não consigo acessar recursos dentro VPC do meu AWS CloudShell VPC ambiente	157
O ENI usado AWS CloudShell pelo meu VPC ambiente não está limpo	157
O usuário com <code>CreateEnvironment</code> permissão somente para VPC ambientes também tem acesso a AWS CloudShell ambientes públicos	158
As credenciais não estão funcionando CloudShell	158
Regiões compatíveis	159
GovCloud Regiões	160
Service quotas e restrições	161
Armazenamento persistente	161
Uso mensal	162
Tamanho do comando	163

Shells simultâneos	163
Sessões de shell	163
Acesso à rede e transferência de dados	163
Restrições nos arquivos do sistema e nas páginas recarregadas	164
Histórico do documento	165
.....	clxix

O que AWS CloudShellé

AWS CloudShell é um shell pré-autenticado baseado em navegador que você pode iniciar diretamente do. AWS Management Console Você pode navegar CloudShell de AWS Management Console algumas maneiras diferentes. Para obter mais informações, consulte [Como começar a usar o AWS CloudShell?](#)

Você pode executar AWS CLI comandos usando seu shell preferido, como Bash, PowerShell, ou Z shell. E você pode fazer isso sem baixar ou instalar ferramentas de linha de comando.



Quando você inicia AWS CloudShell, um [ambiente computacional](#) baseado no Amazon Linux 2023 é criado. Nesse ambiente, você pode acessar uma [ampla variedade de ferramentas de desenvolvimento pré-instaladas](#), opções para [carregar](#) e [baixar](#) arquivos e [armazenamento de arquivos que persiste entre as sessões](#).

(Experimente agora: [Começando com AWS CloudShell](#))

Atributos do AWS CloudShell

O AWS CloudShell fornece os seguintes recursos:

AWS Command Line Interface

Você pode iniciar AWS CloudShell a partir do AWS Management Console. As AWS credenciais que você usou para entrar no console estão automaticamente disponíveis em uma nova sessão de shell. Como AWS CloudShell os usuários são pré-autenticados, você não precisa configurar as credenciais ao interagir com Serviços da AWS o uso da versão 2. AWS CLI O AWS CLI é pré-instalado no ambiente computacional do shell.

Para obter mais informações sobre como interagir com o Serviços da AWS uso da interface de linha de comando, consulte [Trabalhando com AWS serviços em AWS CloudShell](#).

Shells e ferramentas de desenvolvimento

Com o shell criado para AWS CloudShell sessões, você pode alternar facilmente entre os shells de linha de comando de sua preferência. Mais especificamente, você pode alternar entre Bash, PowerShell, e Z shell. Você também tem acesso a ferramentas e utilitários pré-instalados. Estes incluem git, make, pip, sudo, tar, tmux, vim, wget e zip.

O ambiente shell é pré-configurado com suporte para várias das principais linguagens de software, como Node.js e Python. Isso significa que, por exemplo, você pode executar Node.js e Python projetos sem primeiro realizar instalações em tempo de execução. PowerShell os usuários podem usar o .NET Core tempo de execução.

Você pode confirmar arquivos criados ou enviados AWS CloudShell para um repositório local antes de enviar esses arquivos para um repositório remoto gerenciado pelo. AWS CodeCommit

Para obter mais informações, consulte [AWS CloudShell ambiente computacional: especificações e software](#).

Armazenamento persistente

Com AWS CloudShell, você pode usar até 1 GB de armazenamento persistente em cada um sem Região da AWS custo adicional. O armazenamento persistente está localizado em seu diretório inicial (\$HOME) e é privado para você. Ao contrário dos recursos de ambiente temporários que são reciclados após o término de cada sessão do shell, os dados do diretório inicial persistem entre as sessões.

Para obter mais informações sobre a retenção de dados no armazenamento persistente, consulte [Armazenamento persistente](#).

Note

CloudShell VPCos ambientes não têm armazenamento persistente. O HOME diretório \$ é excluído quando seu VPC ambiente expira (após 20 a 30 minutos de inatividade) ou quando você exclui ou reinicia seu ambiente.

CloudShell VPCambientes

AWS CloudShell a nuvem privada virtual (VPC) permite que você crie um CloudShell ambiente em seuVPC. Para cada VPC ambiente, você pode atribuir umaVPC, adicionar uma sub-rede e associar um ou mais grupos de segurança. AWS CloudShell herda a configuração de rede do VPC e permite que você use AWS CloudShell com segurança na mesma sub-rede que outros recursos no. VPC

Segurança

O AWS CloudShell ambiente e seus usuários são protegidos por recursos de segurança específicos. Isso inclui recursos como gerenciamento de IAM permissões, restrições de sessão do shell e colagem segura para entrada de texto.

Gerenciamento de permissões com IAM

Como administrador, você pode conceder e negar permissões aos AWS CloudShell usuários usando IAM políticas. Você também pode criar políticas que especificam as ações específicas que os usuários podem realizar com o ambiente do shell. Para obter mais informações, consulte [Gerenciando AWS CloudShell o acesso e o uso com IAM políticas](#).

Gerenciamento de sessões do shell

Sessões inativas e de longa duração são automaticamente interrompidas e recicladas. Para obter mais informações, consulte [Sessões de shell](#).

Safe Paste para entrada de texto

O Safe Paste é habilitado por padrão. Esse atributo de segurança exige que você verifique se o texto de várias linhas que você deseja colar no shell não contém scripts maliciosos. Para obter mais informações, consulte [Uso do Safe Paste para texto de várias linhas](#).

Opções de personalização

Você pode personalizar sua AWS CloudShell experiência de acordo com sua preferência exata. Por exemplo, você pode alterar os layouts da tela (várias guias), os tamanhos dos textos exibidos e alternar entre os temas da interface clara e escura. Para obter mais informações, consulte [Personalizando sua experiência AWS CloudShell](#).

Você também pode estender seu ambiente de shell [instalando seu próprio software](#) e [modificando seu shell com scripts](#).

Restauração de sessão

A funcionalidade de restauração de sessão restaura as sessões que você estava executando em uma ou várias guias do navegador no CloudShell terminal. Se você atualizar ou reabrir as guias do navegador fechadas recentemente, essa funcionalidade retomará a sessão até que o shell seja interrompido devido à sessão inativa. Para continuar usando sua CloudShell sessão, pressione qualquer tecla na janela do terminal. Para obter mais informações sobre sessões de shell, consulte [Sessões de shell](#).

A restauração da sessão também restaura a saída mais recente do terminal e os processos em execução em cada guia do terminal.

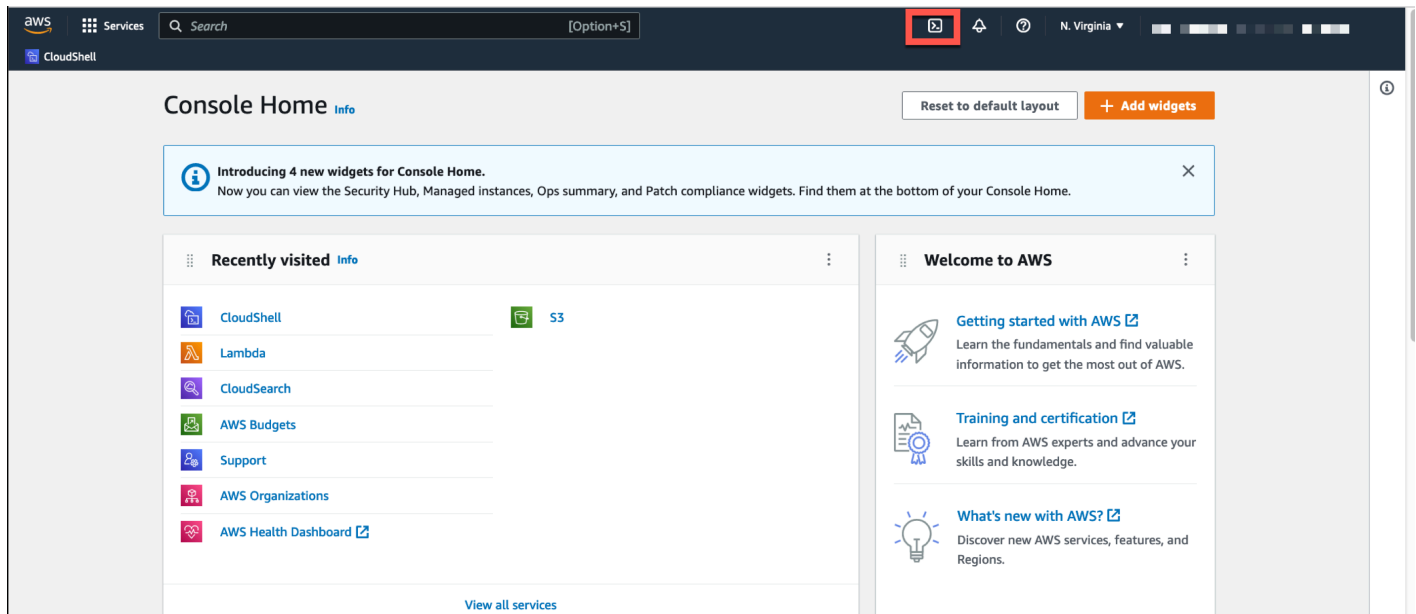
Note

A restauração de sessão não está disponível em aplicativos móveis.

Como começar com AWS CloudShell?

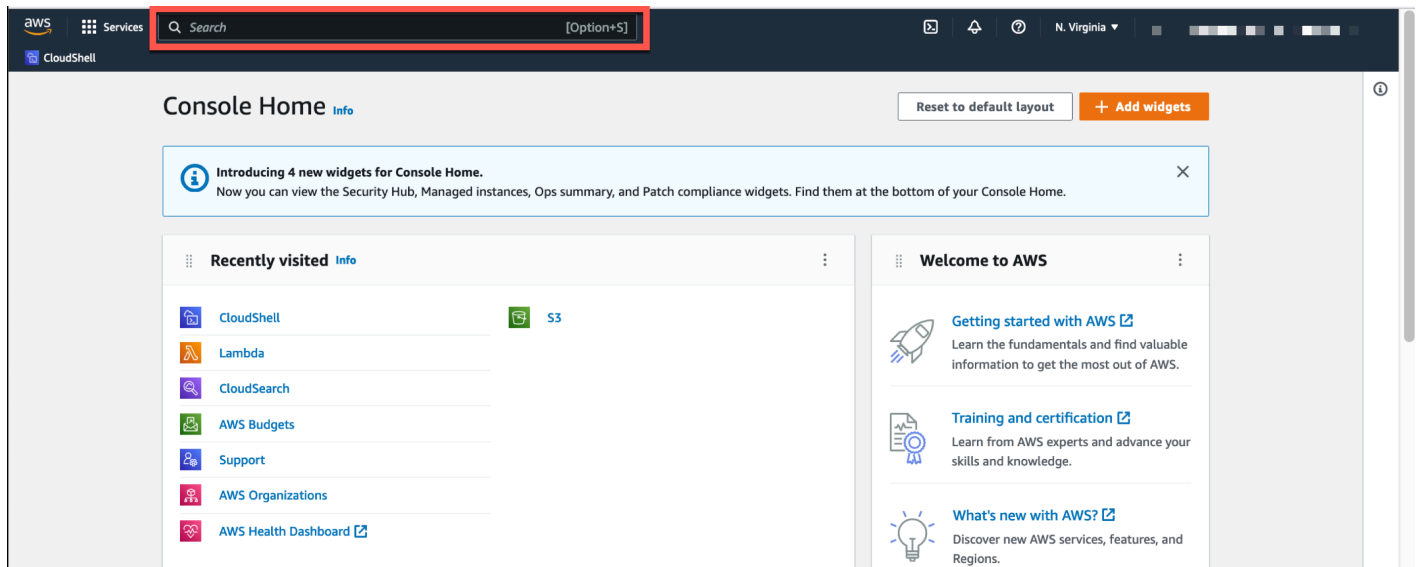
Para começar a trabalhar com o shell, faça login no AWS Management Console e escolha uma das seguintes opções:

- Na barra de navegação, escolha o CloudShell ícone.



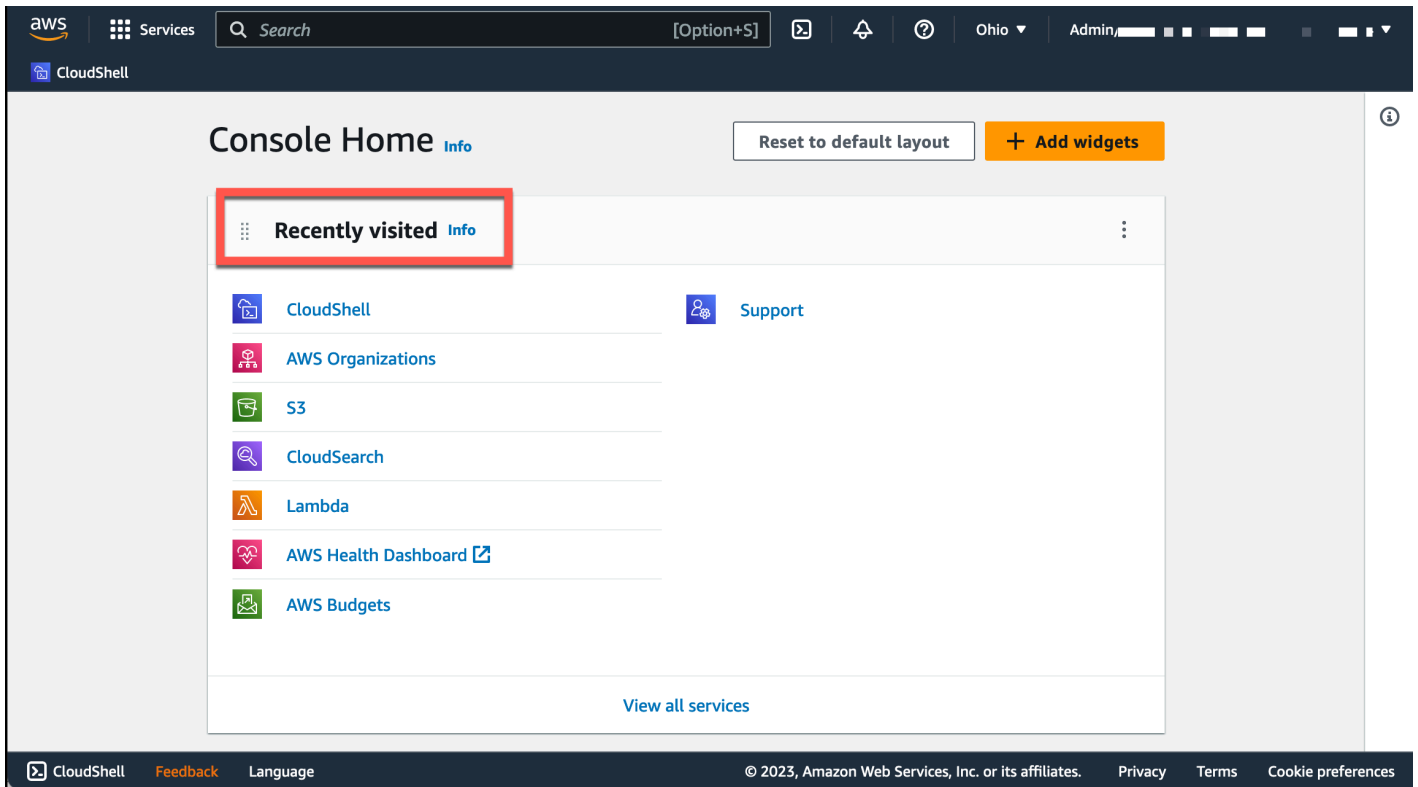
- Na caixa Pesquisar, digite “CloudShell” e escolha CloudShell.

Essa etapa abre sua CloudShell sessão em tela cheia.

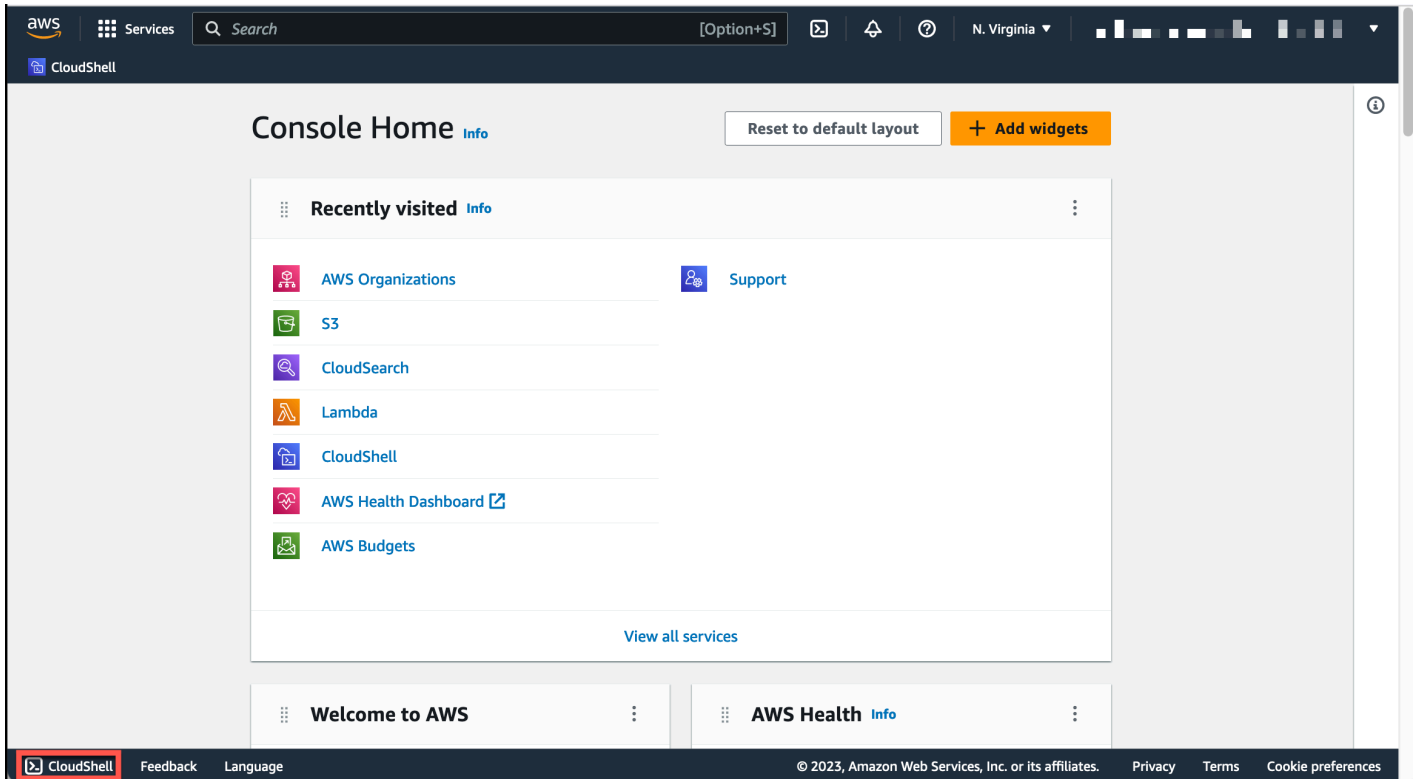


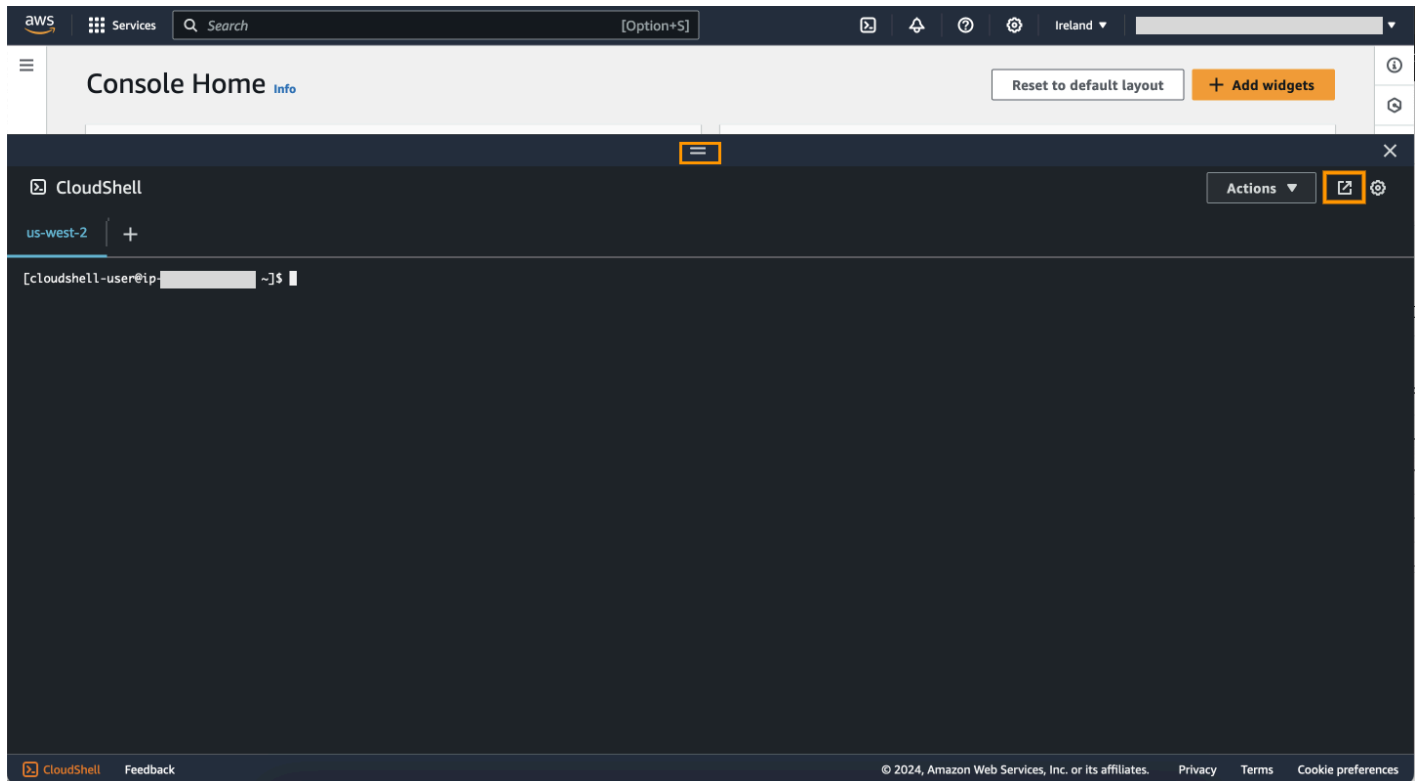
- No widget Visitado recentemente, escolha CloudShell.

Essa etapa abre sua CloudShell sessão em tela cheia.



- Escolha CloudShell no Console Toolbar, no canto inferior esquerdo do console. Você pode ajustar a altura da CloudShell sessão = arrastando.





Você também pode mudar sua CloudShell sessão para uma tela cheia clicando em Abrir na nova guia do navegador.

Para obter instruções sobre como fazer login AWS Management Console e realizar as principais tarefas com AWS CloudShell, consulte [Introdução](#) ao AWS CloudShell.

Preços para AWS CloudShell

AWS CloudShell é um AWS service (Serviço da AWS) que está disponível sem custo adicional. No entanto, você paga por outros AWS recursos com os quais trabalha AWS CloudShell. Além disso, [as taxas padrão de transferência de dados](#) também se aplicam. Para obter mais informações, consulte [Definição de preço do AWS CloudShell](#).

Para obter mais informações, consulte [Cotas e restrições de serviço para AWS CloudShell](#).

AWS CloudShell Tópicos principais

- [Começando com AWS CloudShell](#)
- [Trabalhando com AWS CloudShell](#)

- [Trabalhando com AWS serviços em AWS CloudShell](#)
- [Personalizando sua experiência AWS CloudShell](#)
- [AWS CloudShell ambiente computacional: especificações e software](#)

AWS CloudShell FAQs

A seguir estão as respostas para algumas perguntas comuns sobre AWS CloudShell.

Para mais FAQs informações sobre segurança, consulte [AWS CloudShell Segurança FAQs](#).

- [Como faço para começar a usar AWS CloudShell?](#)
- [O que eu preciso acessar AWS CloudShell?](#)
- [O que está AWS CloudShell no Console Toolbar?](#)
- [Como faço para lançar AWS CloudShell no Console Toolbar?](#)
- [Como faço para criar e gerenciar meu AWS CloudShell ambiente?](#)
- [O que Regiões da AWS está AWS CloudShell disponível em?](#)
- [Que Região da AWS é atribuído se AWS CloudShell não estiver disponível na região selecionada quando você inicia CloudShell no Console Toolbar?](#)
- [Quais tipos de shells posso usar no AWS CloudShell?](#)
- [Com quais navegadores da web posso usar AWS CloudShell?](#)
- [Quais navegadores da web posso usar quando inicio AWS CloudShell no Console Toolbar?](#)
- [Posso baixar um arquivo ao iniciar AWS CloudShell o Console Toolbar?](#)
- [Qual software está pré-instalado no meu ambiente de shell?](#)
- [Posso instalar software que não esteja disponível no ambiente shell?](#)
- [Posso restringir as ações que os usuários podem realizar no AWS CloudShell?](#)
- [Como posso mover dados do meu diretório inicial se eu quiser alterar o Região da AWS local onde estou usando AWS CloudShell?](#)
- [Posso aumentar o limite que determina quando o AWS CloudShell atinge o tempo limite devido à inatividade do usuário?](#)
- [Posso acessar AWS CloudShell a AWS Console Mobile Application partir da tela inicial?](#)
- [Como posso lançar AWS CloudShell no AWS Console Mobile Application?](#)

- [Posso usar teclas modificadoras no meu teclado IOS e no meu teclado do Android ao usar AWS CloudShell no? AWS Console Mobile Application](#)
- [Posso dividir a exibição da AWS CloudShell guia em várias guias no AWS Console Mobile Application?](#)
- [Posso acessar a barra AWS CloudShell de ferramentas do console em um dispositivo móvel?](#)
- [Quais são meus custos correntes CloudShell para minha AmazonVPC?](#)
- [Posso criar mais de dois VPC ambientes por IAM principal?](#)

Como faço para começar a usar AWS CloudShell?

Você pode começar lançando AWS CloudShell em algumas etapas a partir do AWS Management Console. Para fazer isso, entre no console usando suas IAM credenciais Conta da AWS ou suas credenciais em <https://console.aws.amazon.com/console/casa>.

Para obter mais informações, consulte [Conceitos básicos do AWS CloudShell](#).

O que eu preciso acessar AWS CloudShell?

Como você acessa a AWS CloudShell partir do AWS Management Console, você deve ser um IAM usuário que possa fornecer um alias ou ID de conta, nome de usuário e senha válidos.

Para iniciar AWS CloudShell no console, você precisa das IAM permissões fornecidas pela política anexada. Para obter mais informações, consulte [Gerenciando AWS CloudShell o acesso e o uso com IAM políticas](#).

O que está AWS CloudShell no Console Toolbar?

O CloudShell ícone no canto inferior esquerdo do AWS Management Console.

Como faço para lançar AWS CloudShell no Console Toolbar?

Você pode iniciar AWS CloudShell no Console Toolbar escolhendo o CloudShell ícone no canto inferior esquerdo do console.

O que Regiões da AWS está AWS CloudShell disponível em?

Para obter uma lista dos endpoints de serviço suportados Regiões da AWS e associados, consulte a [AWS CloudShell página](#) no Referência geral da Amazon Web Services.

Que Região da AWS é atribuído se AWS CloudShell não estiver disponível na região selecionada quando você inicia CloudShell no Console Toolbar?

A região padrão é atribuída a uma região mais próxima da região selecionada. Para obter mais informações, consulte [Selecionar uma região AWS CloudShell, iniciar e escolher um shell](#).

Você pode executar o comando que fornece permissões para gerenciar recursos em uma região diferente da região padrão. Para obter mais informações, consulte [Trabalhando em Regiões da AWS](#).

Quais tipos de shells posso usar no AWS CloudShell?

Em AWS CloudShell, você pode executar comandos usando o Bash shell, PowerShell, ou o Z shell. Para alternar shells, digite o nome do shell que você deseja usar usando o seguinte formato no prompt de comando:

- bash: Use o Bash shell
- pwsh: Use PowerShell
- zsh: Use o Z shell

Com quais navegadores da web posso usar AWS CloudShell?

AWS CloudShell suporta as versões mais recentes dos navegadores Google Chrome, Mozilla Firefox, Microsoft Edge e Apple Safari.

Como faço para criar e gerenciar meu AWS CloudShell ambiente?

Seu AWS CloudShell ambiente é criado e gerenciado por ID de IAM usuário por região. Você pode verificar o `UserId` executando `aws sts get-caller-identity`. O ambiente é de propriedade da ID IAM do usuário nessa região específica. Você poderá acessar um AWS CloudShell ambiente diferente se alterar a região IAM `UserId` ou.

Quais navegadores da web posso usar quando inicio AWS CloudShell no Console Toolbar?

Você pode iniciar CloudShell no Console Toolbar usando as versões mais recentes dos navegadores Google Chrome, Microsoft Edge, Mozilla Firefox e Apple Safari.

Posso baixar arquivos do AWS CloudShell?

Sim, você pode baixar um arquivo ao iniciar CloudShell o Console Toolbar ou na página do CloudShell console usando um navegador. Você pode baixar um arquivo usando as versões mais recentes dos navegadores Google Chrome e Microsoft Edge.

Atualmente, você não pode baixar um arquivo usando os navegadores Mozilla Firefox e Apple Safari.

Note

A opção de download de arquivos não está disponível para AWS CloudShell VPC ambientes.

Qual software está pré-instalado no meu ambiente de shell?

Com o shell criado para AWS CloudShell sessões, você pode alternar facilmente entre seus shells de linha de comando preferidos (Bash, PowerShell, e Z shell). Você também pode ter acesso a ferramentas e utilitários pré-instalados, como Make, pip, sudo, tar, tmux, Vim, Wget e Zip.

O ambiente shell é pré-configurado com suporte para a maioria das principais linguagens de software. Por exemplo, você pode usá-lo para executar Node.js e Python projetos sem precisar primeiro realizar instalações em tempo de execução. PowerShell os usuários podem usar o .NET Core tempo de execução.

Você pode adicionar arquivos que foram criados usando o shell ou carregados com a interface do shell em um repositório controlado por versão gerenciado usando uma versão pré-instalada do git.

Para obter mais informações, consulte [Software pré-instalado](#).

Posso instalar software que não esteja disponível no ambiente shell?

Sim, AWS CloudShell os usuários têm sudo privilégios e pode instalar software a partir da linha de comando. Para obter mais informações, consulte [Instalação de software de terceiros no ambiente do shell](#).

Posso restringir as ações que os usuários podem realizar no AWS CloudShell?

Sim, você pode controlar quais ações os usuários podem realizar no AWS CloudShell. Por exemplo, você pode permitir que os usuários acessem AWS CloudShell , mas impedir que eles carreguem ou

baixem arquivos dentro do ambiente shell. Ou, como alternativa, você pode impedir completamente o acesso dos usuários ao AWS CloudShell. Para obter mais informações, consulte [Gerenciando AWS CloudShell o acesso e o uso com IAM políticas](#).

Como posso mover dados do meu diretório inicial se eu quiser alterar o Região da AWS local onde estou usando AWS CloudShell?

Para mover seus AWS CloudShell dados de uma Região da AWS para outra região, primeiro baixe o conteúdo do seu diretório inicial em uma região para sua máquina local e, em seguida, faça o upload dele para o diretório inicial em outra região. Para obter mais informações, consulte [???](#).

Note

As opções de upload e download não estão disponíveis para AWS CloudShell VPC ambientes.

Posso aumentar o limite que determina quando o AWS CloudShell atinge o tempo limite devido à inatividade do usuário?

Sua sessão de shell termina automaticamente após aproximadamente 20 a 30 minutos se você não interagir com o AWS CloudShell uso do teclado ou do ponteiro. Os processos em execução não contam como interações. Como foi CloudShell projetado para atividades focadas e baseadas em tarefas, não há planos no momento para aumentar esse [limite de tempo limite](#).

Se você quiser realizar tarefas baseadas em terminais usando um AWS service (Serviço da AWS) com tempos limite mais flexíveis, recomendamos usar nosso sistema baseado em nuvem IDE ou iniciar e [conectar-se a](#) uma instância da Amazon. [AWS Cloud9](#)EC2

Posso acessar AWS CloudShell a AWS Console Mobile Application partir da tela inicial?

Sim, você pode acessar AWS CloudShell o AWS Console Mobile Application fazendo login no Console Mobile Application. Para obter mais informações, consulte o [Guia do usuário do AWS Console Mobile Application](#).

Como posso lançar AWS CloudShell no AWS Console Mobile Application?

Você pode iniciar AWS CloudShell usando um dos seguintes métodos:

1. Selecione o ícone do AWS CloudShell na parte inferior da barra de navegação.
2. Selecione o AWS CloudShell no menu Serviços.

Note

Atualmente, você não pode criar ou iniciar VPC ambientes no AWS Console Mobile Application.

Posso usar teclas modificadoras nos meus teclados iOS e Android ao usar AWS CloudShell no? AWS Console Mobile Application

Sim, você pode usar teclas modificadoras nos teclados iOS e Android. Para obter mais informações, consulte o [Guia do Usuário do AWS Console Mobile Application](#).

Posso dividir a exibição da AWS CloudShell guia em várias guias no AWS Console Mobile Application?

Não, atualmente você não pode executar várias AWS CloudShell guias em seu aplicativo móvel.

Posso acessar AWS CloudShell no Console Toolbar em um dispositivo móvel?

Não, atualmente você não pode acessar AWS CloudShell no Console Toolbar no seu dispositivo móvel.

Quais são meus custos correntes CloudShell para minha AmazonVPC?

Não há cobrança para se conectar à sua conta privada VPC e acessar os recursos nela contidos. As transferências de dados dentro do seu Private VPC estão incluídas em seu VPC faturamento, e as transferências de dados entre você e você CloudShell são cobradas VPCs pelo mesmo custo que as atuais CloudShell.

Posso criar mais de dois VPC ambientes por IAM principal?

Não. Você só pode criar até dois VPC ambientes.

Começando com AWS CloudShell

Este tutorial introdutório mostra como iniciar AWS CloudShell e executar tarefas importantes usando a interface de linha de comando do shell.

Primeiro, você faz login no AWS Management Console e seleciona um Região da AWS. Em seguida, você inicia CloudShell em uma nova janela do navegador e em um tipo de shell com o qual trabalhar.

Depois, você cria uma nova pasta no seu diretório inicial e carrega um arquivo nela a partir da sua máquina local. Você trabalha nesse arquivo usando um editor pré-instalado antes de executá-lo como um programa na linha de comando. Por fim, você chama AWS CLI comandos para criar um bucket do Amazon S3 e adicionar seu arquivo como um objeto ao bucket.

Pré-requisitos

IAMpermissões

Você pode obter permissões AWS CloudShell anexando a seguinte política AWS gerenciada à sua IAM identidade (como usuário, função ou grupo):

- `AWSCloudShellFullAccess`: fornece aos usuários acesso total AWS CloudShell a seus recursos.

Neste tutorial, você também interage com Serviços da AWS. Mais especificamente, você interage com o Amazon S3 criando um bucket do S3 e adicionando um objeto a esse bucket. Sua IAM identidade exige uma política que conceda, no mínimo, as `s3:PutObject` permissões `s3:CreateBucket` e.

Para obter mais informações, consulte [Ações do Amazon S3](#) no Guia do usuário do Amazon Simple Storage Service.

Arquivo de exercícios

Esse exercício também envolve carregar e editar um arquivo que é executado como um programa a partir da interface da linha de comando. Abra um editor de texto na máquina local e adicione o seguinte trecho de código.

```
import sys
```

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

Salve o arquivo com o nome `add_prog.py`.

Conteúdo

- [Etapa 1: faça login em AWS Management Console](#)
- [Etapa 2: selecione uma região AWS CloudShell, inicie e escolha um shell](#)
- [Etapa 3: baixar um arquivo do AWS CloudShell](#)
- [Etapa 4: fazer upload de um arquivo para AWS CloudShell](#)
- [Etapa 5: Remover um arquivo do AWS CloudShell](#)
- [Etapa 6: criar um backup do diretório inicial](#)
- [Etapa 7: reiniciar uma sessão de shell](#)
- [Etapa 8: excluir um diretório inicial da sessão de shell](#)
- [Etapa 9: editar o código do seu arquivo e executá-lo na linha de comando](#)
- [Etapa 10: Use AWS CLI para adicionar o arquivo como um objeto em um bucket do Amazon S3](#)

Etapa 1: faça login em AWS Management Console

Esta etapa envolve a inserção das informações IAM do usuário para acessar AWS Management Console o. Se você já estiver no console, vá para a [etapa 2](#).

- Você pode acessar o AWS Management Console usando o login de um IAM usuário URL ou acessando a página principal de login.

IAM user sign-in URL

- Abra um navegador e insira o seguinte loginURL. Substitua `account_alias_or_id` pelo alias ou ID da conta que seu administrador forneceu.

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

- Insira suas credenciais IAM de login e escolha Entrar.

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

- Aberto <https://aws.amazon.com/console/>.
- Se você não fez login anteriormente usando esse navegador, a página principal de login será exibida. Escolha IAM usuário, insira o alias da conta ou o ID da conta e escolha Avançar.

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

- Se você já fez login como IAM usuário antes. Talvez o seu navegador se lembre do alias ou do ID da conta da Conta da AWS. Se sim, insira suas credenciais IAM de login e escolha Entrar.

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?](#)


 Note

Você também pode fazer login como [usuário raiz](#). Essa identidade tem acesso completo a todos Serviços da AWS os recursos da conta. Recomendamos não usar o usuário raiz para suas tarefas diárias, nem mesmo as administrativas. Em vez disso, adote a prática recomendada de usar o usuário root somente para criar seu primeiro IAM usuário.

Etapa 2: selecione uma região AWS CloudShell, inicie e escolha um shell

Nesta etapa, você inicia a AWS CloudShell partir da interface do console, escolhe um disponível Região da AWS e alterna para o shell de sua preferência, como Bash, PowerShell, ou Z shell.

1. Para escolher uma Região da AWS para trabalhar, acesse o menu Selecionar uma região e selecione uma [AWS região suportada](#) para trabalhar. (As regiões disponíveis estão em destaque.)

 Important

Se você alternar entre regiões, a interface será atualizada e o nome da Região da AWS selecionada será exibido acima do texto da linha de comando. Todos os arquivos que você adiciona ao armazenamento persistente estão disponíveis somente nessa mesma Região da AWS. Se você alterar as regiões, diferentes armazenamentos e arquivos estarão acessíveis.

 Important

Se CloudShell não estiver disponível na região selecionada quando você inicia CloudShell no Console Toolbar, no canto inferior esquerdo do console, a região padrão é definida como a região mais próxima da região selecionada. Você pode executar o comando que fornece permissões para gerenciar recursos em uma região diferente da região padrão. Para obter mais informações, consulte [Trabalhando em Regiões da AWS](#).

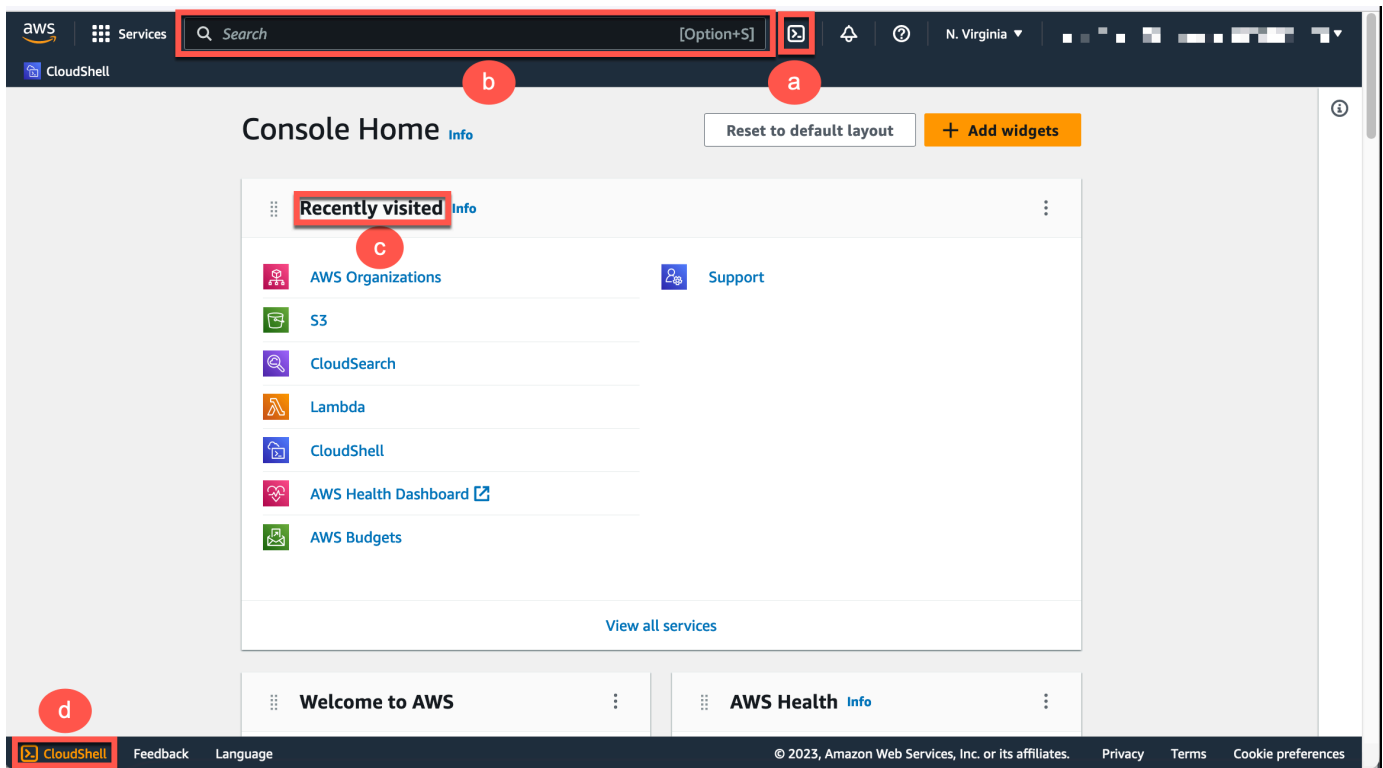
Example

Exemplo

Se você escolher Europa (Espanha) eu-south-2 mas CloudShell não está disponível na Europa (Espanha) eu-south-2, então a Região padrão é definida como Europa (Irlanda) eu-west-1, que está mais próxima da Europa (Espanha) eu-south-2.

Você usará as cotas de serviço para a região padrão, Europa (Irlanda) eu-west-1 e a mesma CloudShell sessão será restaurada em todas as regiões. A região padrão pode ser alterada e você será notificado na janela do CloudShell navegador.

2. A partir do AWS Management Console, você pode iniciar CloudShell escolhendo uma das seguintes opções:
 1. Na barra de navegação, escolha o CloudShell ícone.
 2. Na caixa Pesquisar, digite “CloudShell” e escolha CloudShell.
 3. No widget Visitado recentemente, escolha CloudShell.
 4. Escolha CloudShell no Console Toolbar, no canto inferior esquerdo do console.
 - Para ajustar a altura da CloudShell sessão, arraste=.
 - Para mudar sua CloudShell sessão para tela cheia, clique no ícone Abrir na nova guia do navegador.



Quando o prompt de comando for exibido, o shell estará pronto para interação.

Note

Se você encontrar problemas que o impeçam de iniciar ou interagir com sucesso AWS CloudShell, verifique as informações para identificar e resolver esses problemas em [Solução de problemas AWS CloudShell](#).

3. Para escolher um shell pré-instalado com o qual trabalhar, digite o nome de programa no prompt da linha de comando.

Bash

```
bash
```

Se você mudar para Bash, o símbolo no prompt de comando é atualizado para\$.

Note

Bash é o shell padrão que está sendo executado quando você inicia AWS CloudShell.

PowerShell

```
pwsh
```

Se você mudar para PowerShell, o símbolo no prompt de comando será atualizado paraPS>.

Z shell

```
zsh
```

Se você mudar para Z shell, o símbolo no prompt de comando é atualizado para%.

Para obter informações sobre as versões pré-instaladas em seu ambiente de shell, consulte a [tabela de shells na seção](#) de ambiente de [AWS CloudShell computação](#).

Etapa 3: baixar um arquivo do AWS CloudShell

Note

Essa opção não está disponível para VPC ambientes.

Esta etapa orienta você no processo de download de um arquivo.

1. Para baixar um arquivo, acesse Ações e escolha Baixar arquivo no menu.

A caixa de diálogo Baixar arquivo é exibida.

2. Na caixa de diálogo Baixar arquivo, insira o caminho do arquivo a ser baixado.

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

Você pode usar caminhos absolutos ou relativos ao especificar um arquivo para download. Com nomes de caminhos relativos, `/home/cloudshell-user/` é adicionado automaticamente ao início por padrão. Portanto, para baixar um arquivo chamado `mydownload-file`, os dois caminhos a seguir são válidos:

- Caminho absoluto: `/home/cloudshell-user/subfolder/mydownloadfile.txt`
- Caminho relativo: `subfolder/mydownloadfile.txt`

3. Escolha Baixar.

Se o caminho do arquivo estiver correto, uma caixa de diálogo será exibida. Use essa caixa de diálogo para abrir o arquivo com o aplicativo padrão. Ou salve o arquivo em uma pasta na sua máquina local.

Note

A opção Download não está disponível quando você inicia CloudShell no Console Toolbar. Você pode baixar um arquivo do CloudShell console ou usando o navegador Chrome. Para obter mais informações sobre como baixar um arquivo, consulte [Etapa 3: Baixar um arquivo do AWS CloudShell](#).

Etapa 4: fazer upload de um arquivo para AWS CloudShell

Note

Essa opção não está disponível para VPC ambientes.

Esta etapa descreve como fazer upload de um arquivo e, em seguida, movê-lo para um novo diretório em seu diretório inicial.

1. Para verificar seu diretório de trabalho atual, no prompt, digite o seguinte comando:

```
pwd
```

Quando você pressiona Enter, o shell retorna seu diretório de trabalho atual (por exemplo, /home/cloudshell-user).

2. Para fazer upload de um arquivo para esse diretório, acesse Ações e escolha Carregar arquivo no menu.

A caixa de diálogo Carregar arquivo é exibida.

3. Escolha Navegar.
4. Na caixa de diálogo Upload de arquivo do seu sistema, selecione o arquivo de texto que você criou para este tutorial (add_prog.py) e escolha Abrir.
5. Na caixa de diálogo Carregar arquivo, escolha Carregar.

Uma barra de progresso rastreia o upload. Se o upload for bem-sucedido, uma mensagem confirmará que add_prog.py foi adicionado à raiz do seu diretório inicial.

6. Para criar um diretório para o arquivo, digite o comando make directories: `mkdir mysub_dir`.
7. Para mover o arquivo carregado da raiz do seu diretório inicial para o novo diretório, use o comando `mv`:

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

8. Para alterar seu diretório de trabalho para o novo diretório, digite `cd mysub_dir`.

O prompt de comando é atualizado para indicar que você alterou seu diretório de trabalho.

9. Para visualizar o conteúdo do diretório atual, `mysub_dir`, digite o comando `ls`.

O conteúdo do diretório de trabalho está listado. Isso inclui o arquivo que você acabou de carregar.

Etapa 5: Remover um arquivo do AWS CloudShell

Esta etapa descreve como remover um arquivo do AWS CloudShell.

1. Para remover um arquivo do AWS CloudShell, use comandos de shell padrão, como `rm` (remove).

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

2. Para remover vários arquivos que atendam aos critérios especificados, execute o comando `find`.

O exemplo a seguir remove todos os arquivos que incluem o sufixo “.pdf” em seus nomes.

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

Note

Suponha que você pare de usar AWS CloudShell em um específico Região da AWS. Em seguida, os dados que estão no armazenamento persistente dessa região são removidos automaticamente após um período especificado. Para obter informações, consulte [Armazenamento persistente](#).

Etapa 6: criar um backup do diretório inicial

Esta etapa descreve como criar um backup do diretório inicial.

1. Crie um arquivo de backup

Crie uma pasta temporária fora do diretório inicial.

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

Você pode usar uma das seguintes opções para criar um backup:

a. Criar um arquivo de backup usando tar

Para criar um arquivo de backup usando tar, insira o seguinte comando:

```
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. Criar um arquivo de backup usando zip

Para criar um arquivo de backup usando zip, insira o seguinte comando:

```
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. Transferir o arquivo de backup para fora CloudShell

Você pode usar uma das seguintes opções para transferir o arquivo de backup para fora CloudShell:

a. Baixar o arquivo de backup em sua máquina local

Você pode baixar o arquivo criado na etapa anterior. Para obter mais informações sobre como baixar um arquivo do CloudShell, consulte [Baixar um arquivo do AWS CloudShell](#).

Na caixa de diálogo do arquivo de download, insira o caminho do arquivo a ser baixado (por exemplo, /tmp/tmp.iA99tD9L98/home.tar.gz).


b. Transferir o arquivo de backup para o S3

Para gerar um bucket, insira este comando:

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

Use AWS CLI para copiar o arquivo para o bucket do S3:

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

 Note

Taxas de transferência de dados podem ser aplicadas.


3. Fazer backup diretamente em um bucket do S3

Para fazer backup diretamente em um bucket do S3, insira o seguinte comando:

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

Etapa 7: reiniciar uma sessão de shell

Esta etapa descreve como reiniciar uma sessão de shell.

 Note


Como medida de segurança, se você não interagir com o shell usando o teclado ou o ponteiro por um longo período, a sessão será interrompida automaticamente. Sessões de longa execução também são interrompidas automaticamente. Para obter mais informações, consulte [Sessões de shell](#).

1. Para reiniciar uma sessão de shell, escolha Ações, Reiniciar .

Você é notificado de que a reinicialização AWS CloudShell interrompe todas as sessões ativas na atual Região da AWS.

2. Para confirmar, escolha Reiniciar.


Uma interface exibe uma mensagem de que o ambiente CloudShell computacional está parando. Depois que o ambiente for interrompido e reiniciado, você poderá começar a trabalhar com a linha de comando em uma nova sessão.

 Note


Em alguns casos, pode levar alguns minutos para que o ambiente seja reiniciado.

Etapa 8: excluir um diretório inicial da sessão de shell

Esta etapa descreve como excluir uma sessão de shell.

 Note

Essa opção não está disponível para VPC ambientes. Quando você reinicia um VPC ambiente, seu diretório inicial é excluído.

 Warning


Excluir seu diretório inicial é uma ação irreversível em que todos os dados armazenados em seu diretório inicial são excluídos permanentemente. No entanto, considere essa opção nas seguintes situações:

- Você modificou um arquivo incorretamente e não consegue acessar o ambiente AWS CloudShell computacional. A exclusão do seu diretório pessoal retorna AWS CloudShell às configurações padrão.
- Você deseja remover todos os seus dados AWS CloudShell imediatamente. Se você parar de usar AWS CloudShell em uma AWS região, o armazenamento persistente [será automaticamente excluído no final do período de retenção](#), a menos que você inicie AWS CloudShell novamente na região.

Se você precisar de armazenamento de longo prazo para seus arquivos, considere um serviço como o Amazon S3 ou. CodeCommit

1. Para excluir uma sessão de shell, escolha Ações, Excluir.

Você é notificado de que a exclusão do diretório AWS CloudShell inicial exclui todos os dados atualmente armazenados em seu AWS CloudShell ambiente.

 Note

Não é possível desfazer essa ação.

2. Para confirmar a exclusão, insira excluir no campo de entrada de texto e selecione Excluir.

AWS CloudShell interrompe todas as sessões ativas na atual Região da AWS e cria um novo ambiente imediatamente.

Para sair manualmente das sessões do shell

Com a linha de comando, você pode sair de uma sessão de shell e fazer logout usando o comando `exit`. Em seguida, pressione qualquer tecla para se reconectar e continuar usando o AWS CloudShell.

Etapa 9: editar o código do seu arquivo e executá-lo usando a linha de comando

Esta etapa demonstra como usar o pré-instalado Vim editor para trabalhar com um arquivo. Em seguida, execute esse arquivo como programa da linha de comando.

1. Para editar o arquivo que você carregou na etapa anterior, insira o seguinte comando:

```
vim add_prog.py
```

A interface do shell é atualizada para exibir o Vim editor.

2. Para editar o arquivo em Vim, pressione a `I` tecla. Agora edite o conteúdo para que o programa some três números em vez de dois.

```
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

Se você colar o texto no editor e tiver o [atributo Safe Paste](#) ativado, um aviso será exibido. O texto de várias linhas copiado pode conter scripts maliciosos. Com o atributo Safe Paste, é possível verificar o texto completo antes de colá-lo. Se você estiver convencido de que o texto é seguro, escolha Colar.

3. Depois de editar o programa, pressione Esc para inserir a Vim modo de comando. Em seguida, insira o comando `:wq` para salvar o arquivo e sair do editor.

Note

Se você é novo no Vim modo de comando, você pode inicialmente achar difícil alternar entre o modo de comando e o modo de inserção. O modo de comando é usado ao salvar arquivos e sair do aplicativo. O modo de inserção é usado ao inserir um novo texto. Para entrar no modo de inserção, pressione `I` e, para entrar no modo de comando, pressione Esc. Para obter mais informações sobre Vim e outras ferramentas que estão disponíveis em AWS CloudShell, consulte [Ferramentas de desenvolvimento e utilitários de shell](#).

4. Na interface da linha de comando principal, execute o programa a seguir e especifique três números para entrada. A sintaxe é a seguinte.

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

A linha de comando exibe a saída do programa: `The sum is 15.`

Etapa 10: Use AWS CLI para adicionar o arquivo como um objeto em um bucket do Amazon S3

Nesta etapa, você cria um bucket do Amazon S3 e, em seguida, usa o `PutObject` método para adicionar seu arquivo de código como um objeto nesse bucket.

Note

Na maioria dos casos, você pode [Usando CodeCommit em AWS CloudShell](#) confirmar um arquivo de software em um repositório com controle de versão. Este tutorial mostra como você pode usar o AWS CLI in AWS CloudShell para interagir com outros AWS serviços. Usando este método, não é necessário baixar nem instalar nenhum recurso adicional. Além disso, como você já está autenticado no shell, não precisará configurar as credenciais antes de fazer chamadas.

1. Para criar um bucket em um determinado local Região da AWS, digite o seguinte comando:

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

Note

Se você estiver criando um bucket fora da us-east-1 Region, adicione create-bucket-configuration com o parâmetro LocationConstraint para especificar a região. Veja a seguir um exemplo de sintaxe.

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

Se a chamada tiver êxito, a linha de comando exibirá uma resposta do serviço semelhante à seguinte saída.

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

Note

Se você não seguir as [regras de nomeação de intervalos](#), o seguinte erro será exibido: Ocorreu um erro (InvalidBucketName) ao chamar a CreateBucket operação: O intervalo especificado não é válido.

2. Para fazer upload de um arquivo e adicioná-lo como um objeto ao bucket que você acabou de criar, chame o método PutObject.

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

Depois que o objeto é carregado no bucket do Amazon S3, a linha de comando exibe uma resposta do serviço semelhante à seguinte saída:

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

O ETag é o hash do objeto que foi armazenado. Você pode usar esse hash para [verificar a integridade do objeto carregado no Amazon S3](#).

Tópicos relacionados

- [Trabalhando com AWS serviços em AWS CloudShell](#)
- [Copiar vários arquivos entre sua máquina local e CloudShell](#)
- [Usando CodeCommit em AWS CloudShell](#)
- [Trabalhando com AWS CloudShell](#)
- [Personalizando sua experiência AWS CloudShell](#)

AWS CloudShell tutoriais

Os tutoriais a seguir mostram como experimentar e testar diferentes funcionalidades e integrações ao usar AWS CloudShell

Visão geral do tutorial	Saiba mais
Copiar vários arquivos	the section called “Tutorial: como copiar vários arquivos”
Usando CodeCommit	???
Criação de pré-assinados URLs	???
Construindo um contêiner Docker interno AWS CloudShell e enviando para a Amazon ECR	???
Implantando uma função Lambda usando o AWS CDK	???

Copiar vários arquivos entre sua máquina local e CloudShell

Este tutorial mostra como copiar vários arquivos entre sua máquina local CloudShell e.

Usando a AWS CloudShell interface, você pode carregar ou baixar um único arquivo entre sua máquina local e o ambiente de shell por vez. Para copiar vários arquivos entre CloudShell e sua máquina local ao mesmo tempo, use uma das seguintes opções:

- Amazon S3: use buckets do S3 como intermediário ao copiar arquivos entre sua máquina local e CloudShell
- Arquivos zip: compacte vários arquivos em uma única pasta compactada que pode ser carregada ou baixada usando a CloudShell interface.

Note

Como CloudShell não permite tráfego de entrada na Internet, atualmente não é possível usar comandos como `scp` ou `rsync` para copiar vários arquivos entre máquinas locais e o ambiente CloudShell computacional.

Como carregar e baixar vários arquivos usando o Amazon S3

Esta etapa descreve como fazer upload e download de vários arquivos usando o Amazon S3.

Pré-requisitos

Para trabalhar com buckets e objetos, você precisa de uma IAM política que conceda permissões para realizar as seguintes ações do Amazon API S3:

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

Para obter uma lista completa das ações do Amazon S3, consulte [Ações na Referência](#) do Amazon Simple Storage Service API.

Faça upload de vários arquivos para AWS CloudShell usar o Amazon S3

Esta etapa descreve como fazer upload de vários arquivos usando o Amazon S3.

1. Em AWS CloudShell, crie um bucket do S3 executando o seguinte `s3` comando:

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

Se a chamada tiver êxito, a linha de comando exibirá uma resposta do serviço S3:

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

2. Faça upload dos arquivos em um diretório da sua máquina local para o bucket. Escolha uma das seguintes opções para fazer upload de arquivos:
 - AWS Management Console: use drag-and-drop para fazer upload de arquivos e pastas para um bucket.
 - AWS CLI: com a versão da ferramenta instalada em sua máquina local, use a linha de comando para fazer upload de arquivos e pastas para o bucket.

Using the console

- Abra o console do Amazon S3 em. <https://s3.console.aws.amazon.com/s3/>

(Se você estiver usando AWS CloudShell, você já deve estar conectado ao console.)

- No painel de navegação à esquerda, escolha Buckets e, depois, o nome do bucket no qual você deseja carregar suas pastas ou arquivos. Você também pode criar um bucket de sua escolha selecionando Criar bucket.
- Para selecionar os arquivos e pastas para fazer upload, escolha Upload. Então, arraste e solte seus arquivos e pastas selecionados na janela do console que lista os objetos no bucket de destino ou escolha Adicionar arquivos ou Adicionar pastas.

Os arquivos que você escolheu estão listados na página Upload.

- Marque as caixas de seleção para indicar os arquivos a serem adicionados.
- Para adicionar os arquivos selecionados ao bucket, escolha Upload.

Note

Para obter informações sobre todas as opções de configuração ao usar o console, consulte [Como fazer upload de arquivos e pastas em um bucket do S3?](#) no Guia do usuário do Amazon Simple Storage Service.

Using AWS CLI

Note

Para essa opção, você precisa ter a AWS CLI ferramenta instalada em sua máquina local e ter suas credenciais configuradas para chamadas para AWS serviços.

Para obter mais informações, consulte o [Guia do usuário do AWS Command Line Interface](#).

- Inicie a AWS CLI ferramenta e execute o `aws s3` comando a seguir para sincronizar o bucket especificado com o conteúdo do diretório atual em sua máquina local:

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

Se a sincronização tiver êxito, as mensagens de upload serão exibidas para cada objeto adicionado ao bucket.

3. Volte para a linha de CloudShell comando e digite o seguinte comando para sincronizar o diretório no ambiente do shell com o conteúdo do bucket do S3:

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

Note

Você também pode adicionar os parâmetros `--exclude "<value>"` e `--include "<value>"` ao comando `sync` para realizar a correspondência de padrões e excluir ou incluir um objeto ou arquivo específico.

Para obter mais informações, consulte [Uso de filtros de exclusão e inclusão](#) na referência de comando da AWS CLI .

Se a sincronização tiver êxito, as mensagens de download serão exibidas para cada arquivo baixado do bucket para o diretório.

Note

O comando de sincronização copia apenas os arquivos novos e atualizados recursivamente a partir do diretório de origem para o destino.

Baixe vários arquivos AWS CloudShell usando o Amazon S3

Esta etapa descreve como baixar vários arquivos usando o Amazon S3.

1. Usando a linha de AWS CloudShell comando, insira o seguinte `aws s3` comando para sincronizar um bucket do S3 com o conteúdo do diretório atual no ambiente shell:

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

Note

Você também pode adicionar os parâmetros `--exclude "<value>"` e `--include "<value>"` ao comando `sync` para realizar a correspondência de padrões e excluir ou incluir um objeto ou arquivo específico.

Para obter mais informações, consulte [Uso de filtros de exclusão e inclusão](#) na referência de comando da AWS CLI .

Se a sincronização tiver êxito, as mensagens de upload serão exibidas para cada objeto adicionado ao bucket.

2. Faça o download do conteúdo do bucket em sua máquina local. Como o console do Amazon S3 não suporta o download de vários objetos, é preciso usar a ferramenta AWS CLI que está instalada na sua máquina local.

Na linha de comando da AWS CLI ferramenta, execute o seguinte comando:

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

Se a sincronização tiver êxito, a linha de comando exibirá uma mensagem de download para cada arquivo atualizado ou adicionado no diretório de destino.

Note

Para essa opção, você precisa ter a AWS CLI ferramenta instalada em sua máquina local e ter suas credenciais configuradas para chamadas para AWS serviços. Para obter mais informações, consulte o [Guia do usuário do AWS Command Line Interface](#).

Como carregar e baixar vários arquivos usando pastas compactadas

Esta etapa descreve como carregar e baixar vários arquivos usando pastas compactadas.

Com os utilitários zip/unzip, é possível compactar vários arquivos em um arquivo que pode ser tratado como um único arquivo. Os utilitários são pré-instalados no ambiente CloudShell computacional.

Para obter mais informações sobre ferramentas pré-instaladas, consulte [Ferramentas de desenvolvimento e utilitários de shell](#).

Faça upload de vários arquivos AWS CloudShell usando pastas compactadas

Esta etapa descreve como fazer upload de vários arquivos usando pastas compactadas.

1. Na sua máquina local, adicione os arquivos a serem carregados em uma pasta compactada.
2. Inicie o CloudShell, em seguida, escolha **Ações**, **Carregar arquivo**.
3. Na caixa de diálogo **Carregar arquivo**, escolha **Selecionar arquivo** e escolha a pasta compactada que você acabou de criar.
4. Na caixa de diálogo **Carregar arquivo**, escolha **Carregar** para adicionar o arquivo selecionado ao ambiente shell.
5. Na linha de comando do CloudShell, execute o comando a seguir para descompactar o conteúdo do arquivo zip em um diretório especificado:

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

Baixe vários arquivos AWS CloudShell usando pastas compactadas

Esta etapa descreve como baixar vários arquivos usando pastas compactadas.

1. Na linha de comando do CloudShell, execute o comando a seguir para adicionar todos os arquivos no diretório atual a uma pasta compactada:

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

2. Selecione **Ações**, **Baixar arquivo**.
3. Na caixa de diálogo **Baixar arquivo**, insira o caminho para a pasta compactada (`/home/cloudshell-user/zip-folder/zipped-archive.zip`, por exemplo) e escolha **Baixar**.

Se o caminho estiver correto, uma caixa de diálogo do navegador oferecerá a opção de abrir a pasta compactada ou salvá-la em sua máquina local.

4. Agora, em sua máquina local, você pode descompactar o conteúdo da pasta compactada baixada.

Usando CodeCommit em AWS CloudShell

CodeCommit é um serviço de controle de origem seguro, altamente escalável e gerenciado que hospeda repositórios Git privados. Usando AWS CloudShell, você pode trabalhar com CodeCommit na linha de comando usando o `git-remote-codecommit` utilitário. Esse utilitário vem pré-instalado no ambiente AWS CloudShell computacional e fornece um método simples para enviar e extrair código dos repositórios. CodeCommit Esse utilitário faz isso estendendo o Git. Para obter mais informações, consulte o [Guia do usuário do AWS CodeCommit](#).

Este tutorial descreve como criar um CodeCommit repositório e cloná-lo em seu ambiente AWS CloudShell computacional. Você também aprende como preparar e confirmar um arquivo em seu repositório clonado antes de enviá-lo para o repositório remoto que é gerenciado na nuvem. AWS

Pré-requisitos

Para obter informações sobre as permissões que um IAM usuário precisa usar AWS CloudShell, consulte a [seção de pré-requisitos no tutorial de introdução](#). Você também precisa de [IAMpermissões](#) para trabalhar com CodeCommit.

Além disso, antes de começar, certifique-se de ter o seguinte:

- Uma compreensão básica dos comandos do Git e dos conceitos de controle de versão
- Um arquivo no diretório inicial do seu shell que pode ser confirmado nos repositórios locais e remotos. Neste tutorial, ele é chamado de `my-git-file`.

Etapa 1: criar e clonar um repositório CodeCommit

Esta etapa descreve como criar e clonar um CodeCommit repositório.

1. Na interface da linha de CloudShell comando, digite o `codecommit` comando a seguir para criar um CodeCommit repositório chamado `MyDemoRepo`.

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

Se o repositório for criado com sucesso, a linha de comando exibirá a resposta do serviço.

```
{  
  "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. Usando a linha de comando, crie um novo diretório para seu repositório local e faça dele seu diretório de trabalho.

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

3. Para clonar o repositório remoto, use o comando `git clone`. (Ao trabalhar com `git-remote-codecommit`, use o URL estilo HTTPS (GRC)).

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

Se o repositório for clonado com sucesso, a linha de comando exibirá a resposta do serviço.

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

4. Para navegar até o repositório clonado, use o comando `cd`.

```
cd MyDemoRepo
```


Etapa 2: Prepare e confirme um arquivo antes de enviá-lo ao seu repositório CodeCommit

Esta etapa descreve como preparar e confirmar um arquivo antes de enviá-lo ao seu CodeCommit repositório.

1. Adicione um arquivo chamado `my-git-file` à `MyDemoRepo` pasta usando um editor Vim ou o recurso de upload de arquivo do AWS CloudShell. Para saber como usar os dois, consulte o [tutorial de introdução](#).

2. Para preparar seu arquivo no repositório, execute o comando `add` do git.

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

3. Para verificar se o arquivo foi preparado e está pronto para ser confirmado, execute o comando `status` do git.

```
git status
```

O `my-git-file` é listado como um novo arquivo e exibido em texto verde, indicando que está pronto para ser confirmado.

4. Confirme essa versão do arquivo preparado no repositório.

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

Note

Se forem solicitadas informações de configuração para concluir a confirmação, use o formato a seguir.

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

5. Para sincronizar seu repositório remoto com as alterações feitas no repositório local, envie as alterações para a ramificação `upstream`.

```
git push
```

Criação de um objeto pré-assinado URL para o Amazon S3 usando AWS CloudShell

Este tutorial mostra como criar um objeto pré-assinado URL para compartilhar um objeto do Amazon S3 com outras pessoas. Como os proprietários do objeto especificam suas próprias credenciais de segurança ao compartilhar, qualquer pessoa que receba o objeto pré-assinado URL pode acessar o objeto por um tempo limitado.

Pré-requisitos

- Um IAM usuário com permissões de acesso fornecidas pela `AWSCloudShellFullAccess` política.
- Para obter as IAM permissões necessárias para criar um objeto pré-assinado URL, consulte [Compartilhar um objeto com outras pessoas](#) no Guia do usuário do Amazon Simple Storage Service.

Etapa 1: criar uma IAM função para conceder acesso ao bucket do Amazon S3

Esta etapa descreve como criar uma IAM função para conceder acesso ao bucket do Amazon S3.

1. Para obter seus IAM detalhes que possam ser compartilhados, chame o `get-caller-identity` comando de AWS CloudShell.

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

Se a chamada tiver êxito, a linha de comando exibirá uma resposta semelhante à seguinte.

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

2. Pegue as informações do usuário que você obteve na etapa anterior e adicione-as a um modelo AWS CloudFormation . Esse modelo cria uma IAM função. Esse perfil concede ao seu colaborador permissões de privilégio mínimo para os recursos compartilhados.

```
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. Salve o AWS CloudFormation modelo em um arquivo chamado `template.yaml`.
4. Use o modelo para implantar a pilha e criar a IAM função chamando o `deploy` comando.

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

Gere o pré-assinado URL

Esta etapa descreve como gerar o pré-assinadoURL.

1. Usando seu editor em AWS CloudShell, adicione o código a seguir. Esse código cria um URL que fornece aos usuários federados acesso direto ao 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. Salve o código em um arquivo chamado `share.py`.
3. Execute o seguinte na linha de comando para recuperar o Amazon Resource Name (ARN) da IAM função de AWS CloudFormation. Em seguida, use-o no Python script para obter credenciais de segurança temporárias.

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

O script retorna um URL que um colaborador pode clicar para acessá-lo AWS CloudShell . AWS Management Console O colaborador tem controle total sobre a pasta `myfolder/` no bucket do Amazon S3 pelos próximos 3.600 segundos (uma hora). As credenciais expiram após uma hora. Após esse período, o colaborador não poderá mais acessar o bucket.

Construindo um contêiner Docker interno CloudShell e enviando-o para um repositório da Amazon ECR

Este tutorial mostra como definir e criar um contêiner Docker AWS CloudShell e enviá-lo para um ECR repositório da Amazon.

Pré-requisitos

- Você deve ter as permissões necessárias para criar e enviar para um ECR repositório da Amazon. Para obter mais informações sobre repositórios com a AmazonECR, consulte os [repositórios ECR privados da Amazon no Guia ECR](#) do usuário da Amazon. Para obter mais informações sobre as permissões necessárias para enviar imagens com a AmazonECR, consulte [IAMPermissões necessárias para enviar uma imagem no Guia ECR](#) do usuário da Amazon.

Procedimento tutorial

O tutorial a seguir descreve como usar a CloudShell interface para criar um contêiner Docker e enviá-lo para um repositório da AmazonECR.

1. Crie uma nova pasta no seu diretório pessoal.

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

2. Navegue até a pasta que você criou.

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

3. Crie um Dockerfile vazio.

```
touch Dockerfile
```

4. Usando um editor de texto, por exemplo nano Dockerfile, abra o arquivo e cole o conteúdo a seguir nele.

```
# 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 entrypoint
CMD [ "Hello, World!" ]
```

5. O Dockerfile agora está pronto para ser construído. Crie o contêiner executando `docker build`. Marque o contêiner com um `easy-to-type` nome para uso em comandos futuros.

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

Certifique-se de incluir o ponto final (.).

6. Agora você pode testar o contêiner para verificar se ele está funcionando corretamente no AWS CloudShell.

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

7. Agora que você tem um contêiner Docker em funcionamento, você precisa enviá-lo para um ECR repositório da Amazon. Se você já tem um ECR repositório da Amazon, pode pular esta etapa.

Execute o comando a seguir para criar um ECR repositório da Amazon para este tutorial.

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

8. Depois de criar o ECR repositório Amazon, você pode enviar o contêiner Docker para ele.

Execute o comando a seguir para obter as credenciais de ECR login da Amazon para o 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

Se a variável de `AWS_REGION` ambiente não estiver definida na sua CloudShell ou se você quiser interagir com recursos em outra Regiões da AWS, execute o seguinte comando:

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

9. Marque a imagem com o ECR repositório de destino da Amazon e, em seguida, envie-a para esse repositório.

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

Se você encontrar erros ou problemas ao tentar concluir este tutorial, consulte a seção [Solução de problemas](#) deste guia para obter ajuda.

Limpeza

Agora você implantou com sucesso seu contêiner Docker no seu repositório da AmazonECR. Para remover os arquivos que você criou neste tutorial do seu AWS CloudShell ambiente, execute o comando a seguir.

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

- Exclua o ECR repositório da Amazon.

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

## Implantando uma função Lambda usando o in AWS CDK AWS CloudShell

Este tutorial mostra como implantar uma função Lambda em sua conta usando o AWS Cloud Development Kit (AWS CDK) in. CloudShell

### Pré-requisitos

- Inicialize sua conta para uso com o. AWS CDK Para obter informações sobre como inicializar com AWS CDK, consulte [Bootstrapping](#) no Guia do desenvolvedor v2. AWS CDK Se você não inicializou a conta, pode acessá-la. `cdk bootstrap CloudShell`
- Verifique se você tem as permissões apropriadas para implantar recursos em sua conta. As permissões de administrador são recomendadas.

### Procedimento tutorial

O tutorial a seguir descreve como implantar uma função Lambda baseada em contêiner do Docker usando o in. AWS CDK CloudShell

1. Crie uma nova pasta no seu diretório pessoal.

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



2. Navegue até a pasta que você criou.

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

3. Instale as AWS CDK dependências localmente.

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

4. Crie um AWS CDK projeto básico na pasta que você criou.

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

5. Usando um editor de texto, por exemplo nano, abra o arquivo `cdk.json` e cole o conteúdo a seguir nele.

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

6. Abra o `lib/docker-tutorial.js` arquivo e cole o conteúdo a seguir nele.

```
// 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. Abra o `lib/Dockerfile` e cole o seguinte conteúdo nele.

```
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. Abra o `lib/hello.js` arquivo e cole o conteúdo a seguir nele.

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

9. Use o AWS CDK CLI para sintetizar o projeto e implantar os recursos. Você deve inicializar sua conta.

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

10. Invoque a função Lambda para confirmá-la e verificá-la.

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

Agora você implantou com sucesso uma função Lambda baseada em contêiner Docker usando o AWS CDK. Para obter mais informações sobre AWS CDK, consulte o [Guia do desenvolvedor AWS CDK v2](#). Se você encontrar erros ou problemas ao tentar concluir este tutorial, consulte a seção [Solução](#) de problemas deste guia para obter ajuda.

## Limpeza

Agora você implantou com sucesso uma função Lambda baseada em contêiner Docker usando o AWS CDK. Dentro do AWS CDK projeto, execute o comando a seguir para excluir os recursos associados. Você será solicitado a confirmar a exclusão.

- ```
npx cdk destroy DockerTutorialStack
```
- Para remover os arquivos e recursos que você criou neste tutorial do seu AWS CloudShell ambiente, execute o comando a seguir.

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

Trabalhando com AWS CloudShell

Esta seção descreve como interagir AWS CloudShell e realizar ações específicas com aplicativos compatíveis.

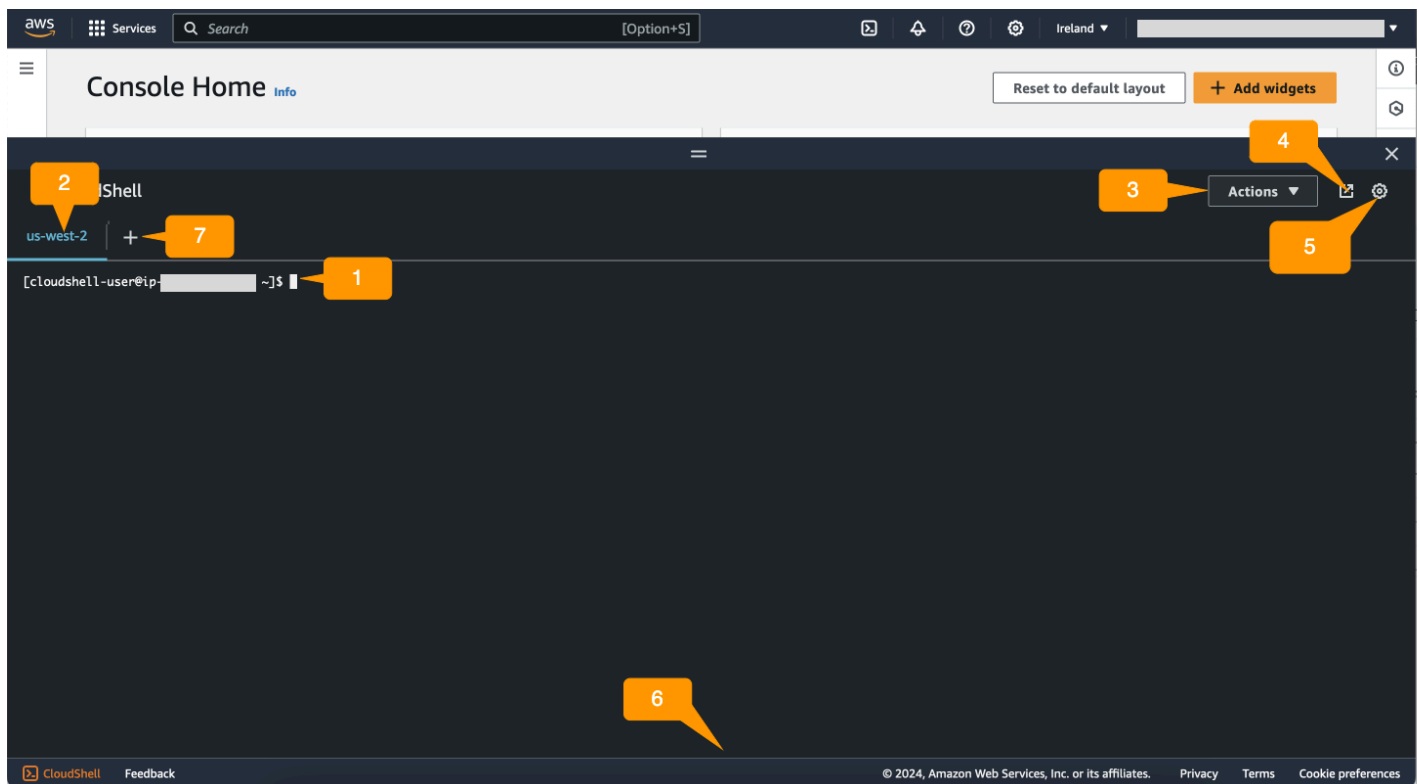
Tópicos

- [Navegando pela interface AWS CloudShell](#)
- [Trabalhando em Regiões da AWS](#)
- [Trabalhar com arquivos e armazenamento](#)
- [Como trabalhar com o Docker](#)


Navegando pela interface AWS CloudShell

Você pode navegar pelos recursos da CloudShell interface a partir do AWS Management Console e Console Toolbar.

A captura de tela a seguir indica vários recursos principais AWS CloudShell da interface.




1. AWS CloudShell interface de linha de comando que você usa para executar comandos usando [seu shell preferido](#). O tipo de shell atual é indicado pelo prompt de comando.
2. A guia do terminal, que usa o Região da AWS local em AWS CloudShell execução no momento.
3. O menu Ações, que fornece opções para [alterar o layout da tela](#), [baixar](#) e [carregar](#) arquivos, [reiniciar seu AWS CloudShell](#) e [excluir seu diretório inicial do AWS CloudShell](#).

 Note

A opção Download não está disponível quando você inicia CloudShell no Console Toolbar.

4. A guia Abrir em um novo navegador, que oferece a opção de acessar sua CloudShell sessão em tela cheia.
5. A opção Preferências, que você pode usar para [personalizar sua experiência do shell](#).
6. A barra inferior, que fornece as seguintes opções para:
 - Inicie CloudShell a partir do CloudShell ícone.
 - Forneça feedback usando o ícone Feedback. Escolha o tipo de feedback que você deseja enviar, adicione seus comentários e escolha Enviar.
 - Para enviar feedback CloudShell, escolha uma das seguintes opções:
 - No console CloudShell, inicie e escolha Feedback. Adicione seus comentários e escolha Enviar.
 - Escolha CloudShell no Console Toolbar, no canto inferior esquerdo do console, e escolha Abrir no ícone da nova guia do navegador, Feedback. Adicione seus comentários e escolha Enviar.

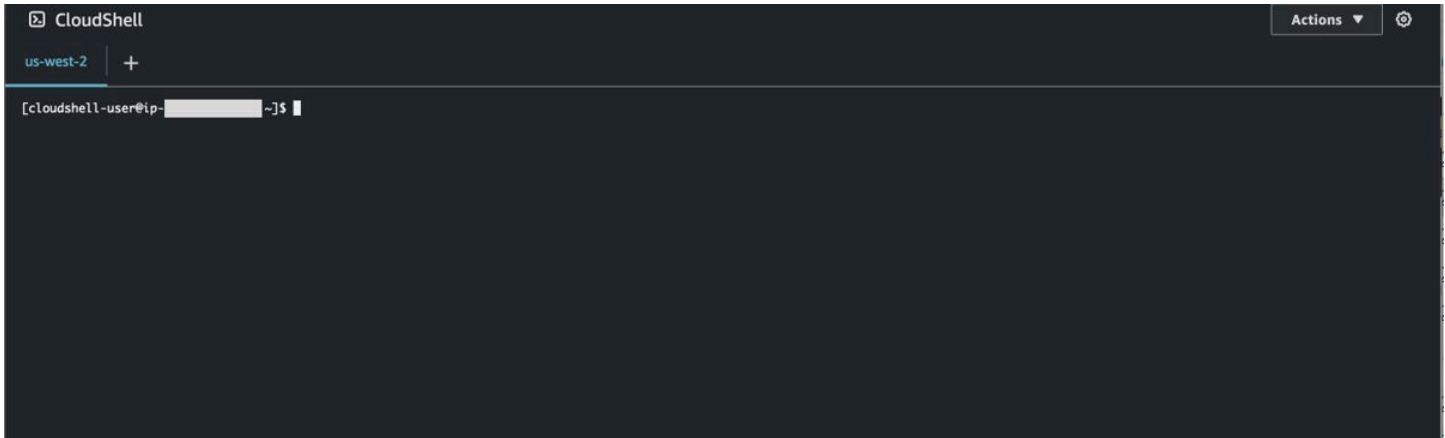
 Note

A opção Feedback não está disponível quando você inicia CloudShell o Console Toolbar.

- Saiba mais sobre nossa política de privacidade e termos de uso e personalize as preferências de cookies.
7. O ícone + é um menu suspenso que inclui opções para criar, reiniciar e excluir ambientes.

Trabalhando em Regiões da AWS

A corrente em Região da AWS que você está executando é exibida como uma guia.



Você pode escolher uma Região da AWS para trabalhar selecionando uma região específica usando o seletor de região. Depois de alterar as regiões, a interface é atualizada à medida que sua sessão de shell se conecta a um ambiente computacional diferente que está sendo executado na região selecionada.

Important

- Você pode usar até 1 GB de armazenamento persistente em cada um Região da AWS. O armazenamento persistente é armazenado em seu diretório inicial (\$HOME). Isso significa que todos os arquivos pessoais, diretórios, programas ou scripts armazenados em seu diretório inicial estão todos localizados em uma Região da AWS. Além disso, eles são diferentes daqueles que estão localizados no diretório inicial e armazenados em uma outra região.

A retenção a longo prazo dos arquivos no armazenamento persistente também é gerenciada com base na região. Para obter mais informações, consulte [Armazenamento persistente](#).

- O armazenamento persistente não está disponível para AWS CloudShell VPC ambientes.

Especificando seu padrão Região da AWS para AWS CLI

Você pode usar [variáveis de ambiente](#) para especificar as opções de configuração e as credenciais necessárias para acessar Serviços da AWS usando AWS CLI. A variável de ambiente que especifica

o padrão Região da AWS para sua sessão de shell é definida quando você inicia a AWS CloudShell partir de uma região específica no AWS Management Console ou quando você escolhe uma opção no seletor de região.

[As variáveis de ambiente têm precedência sobre AWS CLI os arquivos de credenciais](#) que são atualizados pelo `aws configure`. Portanto, você não pode executar o comando `aws configure` para alterar a região especificada pela variável de ambiente. Em vez disso, para alterar a região padrão dos AWS CLI comandos, atribua um valor à variável de ambiente `AWS_REGION`. Nos exemplos a seguir, substitua `us-east-1` pela região em que você está.

Bash or Zsh

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

Configurar a variável de ambiente altera o valor usado até o final da sua sessão de shell ou quando você define a variável como um valor diferente. Você pode tornar as variáveis persistentes em sessões futuras definindo-as no script de startup do shell.

PowerShell

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

Se você definir uma variável de ambiente no PowerShell prompt, a variável de ambiente salvará o valor somente durante a sessão atual. Como alternativa, você pode definir a variável para todas as PowerShell sessões futuras adicionando a variável ao seu PowerShell perfil. Para obter mais informações sobre como armazenar variáveis de ambiente, consulte a [PowerShell documentação](#).

Para confirmar que você alterou a região padrão, execute o `aws configure list` comando para exibir os dados de AWS CLI configuração atuais.

Note

Para AWS CLI comandos específicos, você pode substituir a região padrão usando a opção `--region` de linha de comando. Para obter mais informações, consulte [Opções de linha de comando](#) no Guia do usuário do AWS Command Line Interface .

Trabalhar com arquivos e armazenamento

Usando AWS CloudShell a interface, você pode fazer upload e baixar arquivos do ambiente shell. Para obter mais informações sobre como baixar e carregar arquivos, consulte [Introdução ao AWS CloudShell](#).

Para garantir que todos os arquivos adicionados estejam disponíveis após o término da sessão, você deve saber a diferença entre armazenamento persistente e temporário.

- Armazenamento persistente: você tem 1 GB de armazenamento persistente para cada um Região da AWS. O armazenamento persistente está no diretório inicial.
- Armazenamento temporário: o armazenamento temporário é reciclado ao final de uma sessão. O armazenamento temporário está nos diretórios que ficam fora do seu diretório inicial.

Important

Certifique-se de deixar os arquivos que você deseja manter e usar para futuras sessões de shell em seu diretório inicial. Por exemplo, suponha que você mova um arquivo para fora do seu diretório inicial executando o comando `mv`. Em seguida, esse arquivo é reciclado quando a sessão atual do shell termina.

Como trabalhar com o Docker

AWS CloudShell suporta totalmente o Docker sem instalação ou configuração. Você pode definir, criar e executar contêineres do Docker internamente AWS CloudShell. Você pode implantar recursos baseados em Docker, como funções Lambda baseadas em contêineres Docker, por meio do AWS CDK Toolkit, bem como criar contêineres Docker e enviá-los para repositórios da Amazon por meio do Docker. ECR CLI Para obter etapas detalhadas sobre como executar essas duas implantações, consulte os seguintes tutoriais:

- [Tutorial: Implantando uma função Lambda usando o AWS CDK](#)
- [Tutorial: Construindo um contêiner Docker interno AWS CloudShell e enviando-o para um repositório da Amazon ECR](#)

Há certas restrições e limitações no uso do Docker com AWS CloudShell:

- O Docker tem espaço limitado em um ambiente. Se você tiver imagens individuais grandes ou muitas imagens pré-existentes do Docker, isso pode causar problemas que podem impedir você de extrair, criar ou executar imagens adicionais. Para obter mais informações sobre o Docker, consulte o guia de [documentação do Docker](#).
- O Docker está disponível em todas as AWS regiões, exceto nas regiões AWS GovCloud (EUA). Para obter uma lista das regiões nas quais o Docker está disponível, consulte [AWS Regiões suportadas para AWS CloudShell](#).
- Se você encontrar problemas ao usar o Docker com AWS CloudShell, consulte a seção [Solução](#) de problemas deste guia para obter informações sobre como potencialmente resolver esses problemas.

Recursos de acessibilidade para AWS CloudShell

Este tópico descreve como usar os recursos de acessibilidade para CloudShell. Você pode usar um teclado para navegar pelos elementos que podem ser focados na página. Você também pode personalizar a aparência do CloudShell, incluindo tamanhos de fonte e temas de interface.

Navegação pelo teclado em CloudShell

Para navegar pelos elementos que podem ser focados na página, pressione Tab.

CloudShell recursos de acessibilidade do terminal

Você pode usar a tecla Tab nos modos a seguir:

- Modo terminal (padrão) — Nesse modo, o terminal captura sua entrada da tecla Tab. Depois que o foco estiver no terminal, pressione Tab para acessar somente a funcionalidade do terminal.
- Modo de navegação — Nesse modo, o terminal não captura a entrada da tecla Tab. Pressione Tab para navegar pelos elementos que podem ser focados na página.

Para alternar entre o modo terminal e o modo de navegação, pressione Ctrl +M. Depois de voltar, Tab: navegação aparece no cabeçalho e você pode usar a tecla Tab para navegar pela página.

Para retornar ao modo terminal, pressione Ctrl+M. Ou escolha X ao lado de Tab: navegação.

Note

Atualmente, os recursos de acessibilidade do CloudShell terminal não estão disponíveis em dispositivos móveis.

Escolhendo tamanhos de fonte e temas de interface em CloudShell

Você pode personalizar a aparência do CloudShell para acomodar suas preferências visuais.

- Tamanho da fonte — Escolha entre os tamanhos de fonte menor, pequeno, médio, grande e maior no terminal. Para obter mais informações sobre como alterar o tamanho da fonte, consulte [the section called “Alteração do tamanho da fonte”](#).

- Tema — Escolha entre temas de interface claro e escuro. Para obter mais informações sobre como alterar o tema da interface, consulte [the section called “Alteração do tema da interface”](#).

Trabalhando com AWS serviços em AWS CloudShell

Um dos principais benefícios AWS CloudShell é que você pode usá-lo para gerenciar seus AWS serviços a partir da interface da linha de comando. Isso significa que você não precisa baixar e instalar ferramentas ou configurar suas credenciais localmente com antecedência. Quando você inicia AWS CloudShell, é criado um ambiente computacional com as seguintes ferramentas de linha de comando já instaladas:

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

E como você já fez login AWS, não há necessidade de configurar suas credenciais localmente antes de usar os serviços. As credenciais que você usou para fazer login no AWS Management Console são encaminhadas para o AWS CloudShell.

Se quiser alterar a AWS região padrão usada para AWS CLI, você pode alterar o valor atribuído à variável de `AWS_REGION` ambiente. (Para ter mais informações, consulte [Especificando seu padrão Região da AWS para AWS CLI](#).)

O restante deste tópico demonstra como você pode começar a usar AWS CloudShell para interagir com AWS serviços selecionados a partir da linha de comando.

AWS CLI exemplos de linha de comando para AWS serviços selecionados

Os exemplos a seguir representam apenas alguns dos vários AWS serviços com os quais você pode trabalhar usando comandos disponíveis na AWS CLI versão 2. Para obter uma lista completa, consulte a [Referência de AWS CLI Comandos](#).

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

DynamoDB

O DynamoDB é um serviço SQL sem banco de dados totalmente gerenciado que fornece desempenho rápido e previsível com escalabilidade perfeita. A implementação desse serviço do SQL modo No oferece suporte a estruturas de dados de documentos e valores-chave.

O `create-table` comando a seguir cria uma tabela no SQL estilo No que é nomeada `MusicCollection` na sua AWS conta.

```
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
```

Para obter mais informações, consulte [Uso do DynamoDB com o AWS CLI](#) no Guia do usuário do AWS Command Line Interface .

AWS Cloud9

AWS Cloud9 é um ambiente de desenvolvimento integrado baseado em nuvem (IDE) que você pode usar para escrever, executar e depurar seu código em uma janela do navegador. O ambiente possui editor de código, depurador e terminal.

O `create-environment-ec2` comando a seguir cria um ambiente de AWS Cloud9 EC2 desenvolvimento com as configurações especificadas. Ele inicia uma EC2 instância da Amazon e, em seguida, se conecta da instância ao ambiente.

```
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
```

Para obter mais informações, consulte [Referência da linha de comando AWS Cloud9](#).

Amazon EC2

O Amazon Elastic Compute Cloud (AmazonEC2) é um serviço web que fornece capacidade computacional segura e redimensionável na nuvem. Ele foi projetado para tornar a computação em nuvem na escala da Web mais fácil e mais acessível.

O `run-instances` comando a seguir inicia uma instância `t2.micro` na sub-rede especificada de um 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
```

Para obter mais informações, consulte [Usando a Amazon EC2 com o AWS CLI](#) no Guia AWS Command Line Interface do usuário.

S3 Glacier

O S3 Glacier e o S3 Glacier Deep Archive são classes de armazenamento na nuvem do Amazon S3 seguras, duráveis e de custo extremamente baixo para arquivamento e back-up de dados a longo prazo.

O comando `create-vault` a seguir cria um cofre, um contêiner para armazenar arquivos:

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

Para obter mais informações, consulte [Uso do Amazon S3 Glacier com o AWS CLI](#) no Guia do usuário do AWS Command Line Interface .

AWS Elastic Beanstalk CLI

O AWS Elastic Beanstalk CLI fornece uma interface de linha de comando feita para simplificar a criação, atualização e monitoramento de ambientes a partir de um repositório local. Nesse contexto, um ambiente se refere a uma coleção de AWS recursos executando uma versão do aplicativo.

O `create` comando a seguir cria um novo ambiente em uma Amazon Virtual Private Cloud personalizada (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
```

Para obter mais informações, consulte a [referência do CLI comando EB](#) no Guia do AWS Elastic Beanstalk desenvolvedor.

Amazon ECS CLI

A interface de linha de comando (ECS) do Amazon Elastic Container Service (AmazonCLI) fornece vários comandos de alto nível. Eles foram projetados para simplificar os processos de criação, atualização e monitoramento de clusters e tarefas de um ambiente de desenvolvimento local. (Um ECS cluster da Amazon é um agrupamento lógico de tarefas ou serviços.)

O `configure` comando a seguir configura a Amazon ECS CLI para criar uma configuração de cluster chamada `ecs-cli-demo`. Essa configuração de cluster usa FARGATE como o tipo de inicialização padrão para o cluster `ecs-cli-demo` em `us-east-1` region.

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

Para obter mais informações, consulte a [Amazon ECS Command Line Reference](#) no Amazon Elastic Container Service Developer Guide.

AWS SAM CLI

AWS SAM CLI é uma ferramenta de linha de comando que opera em um AWS Serverless Application Model modelo e código de aplicativo. Você pode realizar várias tarefas usando-a. Isso inclui invocar funções Lambda localmente, criar um pacote de implantação para seu aplicativo sem servidor e implantar seu aplicativo sem servidor na nuvem. AWS

O `init` comando a seguir inicializa um novo SAM projeto com os parâmetros necessários passados como parâmetros:

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

Para obter mais informações, consulte a [referência de AWS SAM CLI comandos](#) no Guia do AWS Serverless Application Model desenvolvedor.

Personalizando sua experiência AWS CloudShell

Você pode personalizar os seguintes aspectos da sua AWS CloudShell experiência:

- [Layout das guias](#): divida a interface da linha de comando em várias colunas e linhas.
- [Tamanho da fonte](#): ajuste o tamanho do texto da linha de comando.
- [Tema de cores](#): alterne entre o tema claro e escuro.
- [Colagem segura](#): ative ou desative um atributo que exige que você verifique o texto em várias linhas antes de colá-lo.
- [Tmux para restauração de sessão](#): usar tmux restaura sua sessão até que ela fique inativa.

Você também pode estender seu ambiente de shell [instalando seu próprio software](#) e [modificando seu shell com scripts](#).

Divisão da exibição da linha de comando em várias guias

Execute vários comandos dividindo sua interface da linha de comando em vários painéis.

Note

Depois de abrir várias guias, selecione uma na qual deseja trabalhar clicando em qualquer lugar no painel de sua escolha. Feche uma guia escolhendo o símbolo x, que está ao lado do nome da região.

- Escolha Ações e uma das seguintes opções no Layout de guias:
 - Nova guia: adicione uma nova guia ao lado da que está ativa no momento.
 - Divisão em linhas: adicione uma nova guia em uma linha abaixo da que está ativa no momento.
 - Divisão em colunas: adicione uma nova guia em uma coluna ao lado da que está ativa no momento.

Se não houver espaço suficiente para exibir completamente cada guia, role para ver a guia inteira. Você também pode selecionar as barras de divisão que separam os painéis e arrastá-las usando o ponteiro para aumentar ou reduzir o tamanho do painel.

Alteração do tamanho da fonte

Aumente ou diminua o tamanho do texto exibido na interface da linha de comando.

1. Para alterar as configurações do AWS CloudShell terminal, acesse Configurações, Preferências.
2. Escolha um tamanho de texto. Suas opções são Menor, Pequeno, Médio, Grande e Maior.

Alteração do tema da interface

Altere entre o tema claro e escuro para a interface da linha de comando.

1. Para alterar o AWS CloudShell tema, acesse Configurações, Preferências.
2. Escolha Claro ou Escuro.

Uso do Safe Paste para texto de várias linhas

O Safe Paste é um atributo de segurança que solicita que você verifique se o texto de várias linhas que você está prestes a colar no shell não contém scripts maliciosos. O texto copiado de sites de terceiros pode conter código oculto que aciona comportamentos inesperados em seu ambiente de shell.

A caixa de diálogo Safe Paste exibe o texto completo que você copiou para a área de transferência. Se estiver convencido de que não há risco de segurança, escolha Colar.

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

Recomendamos que você ative o Safe Paste para detectar possíveis riscos de segurança em scripts. Você pode ativar ou desativar esse atributo escolhendo Preferências, Ativar Safe Paste e Desativar Safe Paste.

O uso do tmux para restaurar a sessão

AWS CloudShell usa o tmux para restaurar as sessões em uma ou várias guias do navegador. Se você atualizar as guias do navegador, a sessão será retomada até que ela fique inativa. Para obter mais informações, consulte [Restaurar sessão](#).

Usando AWS CloudShell na Amazon VPC

AWS CloudShell a nuvem privada virtual (VPC) permite que você crie um CloudShell ambiente em seu VPC. Para cada VPC ambiente, você pode atribuir uma VPC, adicionar uma sub-rede e associar até cinco grupos de segurança. AWS CloudShell herda a configuração de rede do VPC e permite que você use AWS CloudShell com segurança na mesma sub-rede que outros recursos no VPC e se conecte a eles.

Com a Amazon VPC, você pode lançar AWS recursos em uma rede virtual logicamente isolada que você definiu. Essa rede virtual se assemelha a uma rede tradicional que você operaria no seu data center, com os benefícios de usar a infraestrutura escalável da AWS. Para obter mais informações sobre VPC, consulte [Amazon Virtual Private Cloud](#).

Restrições operacionais

AWS CloudShell VPCs ambientes têm as seguintes restrições:

- Você pode criar no máximo dois VPC ambientes por IAM principal.
- Você pode atribuir no máximo cinco grupos de segurança para um VPC ambiente.
- Você não pode usar as opções de CloudShell upload e download no menu Ações para VPC ambientes.

Note

É possível fazer upload ou download de arquivos de VPC ambientes que tenham acesso à entrada/saída da internet por meio de outras ferramentas. CLI

- VPCs ambientes não oferecem suporte ao armazenamento persistente. O armazenamento é efêmero. Os dados e o diretório inicial são excluídos quando uma sessão do ambiente ativo termina.
- Seu AWS CloudShell ambiente só pode se conectar à Internet se estiver em uma VPC sub-rede privada.

Note

Os endereços IP públicos não são alocados aos CloudShell VPC ambientes por padrão. VPCs ambientes criados em sub-redes públicas com tabelas de roteamento configuradas

para rotear todo o tráfego para o Internet Gateway não terão acesso à Internet pública, mas sub-redes privadas configuradas com Network Address Translation (NAT) terão acesso à Internet pública. VPCs criados em tais sub-redes privadas terão acesso à Internet pública.

- Para fornecer um CloudShell ambiente gerenciado para sua conta, AWS pode provisionar acesso à rede aos seguintes serviços para o host de computação subjacente:
 - Amazon S3
 - VPC endpoints
 - com.amazonaws. <region>.mensagens ssm
 - com.amazonaws. <region>.registros
 - com.amazonaws. <region>.kms
 - com.amazonaws. <region>.execute-api
 - com.amazonaws. <region>.ecs-telemetry
 - 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

Você não pode restringir o acesso a esses endpoints modificando sua VPC configuração.

CloudShell VPC está disponível em todas as AWS regiões, exceto nas regiões AWS GovCloud (EUA). Para obter uma lista das regiões nas quais CloudShell VPC está disponível, consulte [AWS Regiões suportadas para AWS CloudShell](#).

Criando um CloudShell VPC ambiente


Este tópico explica as etapas para criar um VPC ambiente em CloudShell.

Pré-requisitos

Seu administrador deve fornecer as IAM permissões necessárias para que você possa criar VPC ambientes. Para obter mais informações sobre como habilitar permissões para criar CloudShell VPC ambientes, consulte [the section called “IAMPermissões necessárias para criar e usar CloudShell VPC ambientes”](#).

Para criar um CloudShell VPC ambiente

1. Na página do CloudShell console, escolha o ícone + e, em seguida, escolha Criar VPC ambiente no menu suspenso.
2. Na página Criar um VPC ambiente, insira um nome para seu VPC ambiente na caixa Nome.
3. Na lista suspensa Nuvem privada virtual (VPC), escolha uma. VPC
4. Na lista suspensa Sub-rede, escolha uma sub-rede.
5. Na lista suspensa Grupo de segurança, escolha um ou mais grupos de segurança que você deseja atribuir ao seu VPC ambiente.

 Note

Você pode escolher no máximo cinco grupos de segurança.

6. Escolha Criar para criar seu VPC ambiente.
7. (Opcional) Escolha Ações e, em seguida, escolha Exibir detalhes para revisar os detalhes do VPC ambiente recém-criado. O endereço IP do seu VPC ambiente é exibido no prompt da linha de comando.

Para obter informações sobre o uso de VPC ambientes, consulte [Conceitos básicos](#).

IAMPermissões necessárias para criar e usar CloudShell VPC ambientes

Para criar e usar CloudShell VPC ambientes, o IAM administrador deve habilitar o acesso a EC2 permissões VPC específicas da Amazon. Esta seção lista as EC2 permissões da Amazon necessárias para criar e usar VPC ambientes.

Para criar VPC ambientes, a IAM política atribuída à sua função deve incluir as seguintes EC2 permissões da Amazon:

- `ec2:DescribeVpcs`

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

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

Recomendamos incluir:

- `ec2>DeleteNetworkInterface`

Note

Essa permissão não é obrigatória, mas é necessária CloudShell para limpar o ENI recurso (ENI criado para CloudShell VPC ambientes marcados com `ManagedByCloudShell` chave) criado por ele. Se essa permissão não estiver habilitada, você deverá limpar manualmente o ENI recurso após cada uso do CloudShell VPC ambiente.

IAM política que concede CloudShell acesso total, incluindo acesso a VPC

O exemplo a seguir mostra como habilitar permissões completas, incluindo acesso VPC a 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": ""
      }
    }
  }
]
}

```

Usando chaves de IAM condição para VPC ambientes

Você pode usar chaves CloudShell de condição específicas para VPC configurações para fornecer controles de permissão adicionais para seus VPC ambientes. Você também pode especificar as sub-redes e os grupos de segurança que o VPC ambiente pode ou não usar.

CloudShell suporta as seguintes chaves de condição nas IAM políticas:

- `CloudShell:VpcIds`— Permitir ou negar um ou mais VPCs
- `CloudShell:SubnetIds`— Permitir ou negar uma ou mais sub-redes
- `CloudShell:SecurityGroupIds`— Permitir ou negar um ou mais grupos de segurança

Note

Se as permissões dos usuários com acesso a CloudShell ambientes públicos forem modificadas para adicionar restrições à `cloudshell:createEnvironment` ação, eles ainda poderão acessar o ambiente público existente. No entanto, se você quiser modificar uma IAM política com essa restrição e desativar seu acesso ao ambiente público existente, você deve primeiro atualizar a IAM política com a restrição e, em seguida, garantir que cada CloudShell usuário em sua conta exclua manualmente o ambiente público existente usando a interface de usuário da CloudShell web (Ações → Excluir CloudShell ambiente).

Exemplo de políticas com chaves de condição para VPC configurações

Os exemplos a seguir demonstram como usar chaves de condição para VPC configurações. Depois de criar uma instrução de política com as restrições desejadas, acrescente a instrução de política para o usuário ou a função de destino.

Garanta que os usuários criem somente VPC ambientes e neguem a criação de ambientes públicos

Para garantir que os usuários possam criar somente VPC ambientes, use a permissão de negação conforme mostrado no exemplo a seguir:

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

```

    "Null": {
      "cloudshell:VpcIds": "true"
    }
  }
}
]
}

```

Negar aos usuários acesso a grupos específicos VPCs, sub-redes ou grupos de segurança

Para negar aos usuários acesso a VPCs informações específicas, use `StringEquals` para verificar o valor da `cloudshell:VpcIds` condição. O exemplo a seguir nega aos usuários o acesso a `vpc-1` e `vpc-2`:

```

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

```

Para negar aos usuários acesso a VPCs informações específicas, use `StringEquals` para verificar o valor da `cloudshell:SubnetIds` condição. O exemplo a seguir nega aos usuários o acesso a `subnet-1` e `subnet-2`:

```

{

```

```

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

```

Para negar aos usuários acesso a VPCs informações específicas, use `StringEquals` para verificar o valor da `cloudshell:SecurityGroupIds` condição. O exemplo a seguir nega aos usuários o acesso a `sg-1` e `sg-2`:

```

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

```

```

    }
  ]
}

```

Permita que os usuários criem ambientes com VPC configurações específicas

Para permitir que os usuários acessem VPCs dados específicos, use `StringEquals` para verificar o valor da `cloudshell:VpcIds` condição. O exemplo a seguir permite que os usuários acessem `vpc-1` e `vpc-2`:

```

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

```

Para permitir que os usuários acessem VPCs dados específicos, use `StringEquals` para verificar o valor da `cloudshell:SubnetIds` condição. O exemplo a seguir permite que os usuários acessem `subnet-1` e `subnet-2`:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceStayInSpecificSubnets",
      "Action": [

```

```

    "cloudshell:CreateEnvironment"
  ],
  "Effect": "Allow",
  "Resource": "*",
  "Condition": {
    "ForAllValues:StringEquals": {
      "cloudshell:SubnetIds": [
        "subnet-1",
        "subnet-2"
      ]
    }
  }
}
]
}

```

Para permitir que os usuários acessem VPCs dados específicos, use `StringEquals` para verificar o valor da `cloudshell:SecurityGroupIds` condição. O exemplo a seguir permite que os usuários acessem `sg-1` e `sg-2`:

```

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

```

Segurança para AWS CloudShell

A segurança da nuvem na Amazon Web Services (AWS) é a nossa maior prioridade. Como AWS cliente, você se beneficia de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança. A segurança é uma responsabilidade compartilhada entre você AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como a Segurança da nuvem e a Segurança na nuvem.

Segurança da nuvem — AWS é responsável por proteger a infraestrutura que executa todos os serviços oferecidos na AWS nuvem e fornecer serviços que você possa usar com segurança. Nossa responsabilidade de segurança é a maior prioridade em AWS, e a eficácia de nossa segurança é regularmente testada e verificada por auditores terceirizados como parte dos [Programas de AWS Conformidade](#).

Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você está usando e por outros fatores, incluindo a sensibilidade de seus dados, os requisitos da sua organização e as leis e regulamentos aplicáveis.

AWS CloudShell segue o [modelo de responsabilidade compartilhada](#) por meio dos AWS serviços específicos que ele suporta. Para AWS obter informações sobre segurança do [AWS serviço](#), [consulte a página de documentação de segurança](#) do serviço e os [AWS serviços que estão no escopo dos esforços de AWS conformidade do programa de conformidade](#).

Os tópicos a seguir mostram como configurar para atender AWS CloudShell aos seus objetivos de segurança e conformidade.

Tópicos

- [Proteção de dados em AWS CloudShell](#)
- [Identity and Access Management para AWS CloudShell](#)
- [Registro e monitoramento em AWS CloudShell](#)
- [Validação de conformidade para AWS CloudShell](#)
- [Resiliência em AWS CloudShell](#)
- [Segurança da infraestrutura em AWS CloudShell](#)
- [Práticas recomendadas de segurança para AWS CloudShell](#)
- [AWS CloudShell Segurança FAQs](#)

Proteção de dados em AWS CloudShell

O modelo de [responsabilidade AWS compartilhada modelo](#) se aplica à proteção de dados em AWS CloudShell. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para obter mais informações sobre privacidade de dados, consulte [Privacidade de dados FAQ](#). Para obter informações sobre proteção de dados na Europa, consulte o [Modelo de Responsabilidade AWS Compartilhada e GDPR](#) a postagem no blog AWS de segurança.

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use a autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com AWS os recursos. Exigimos TLS 1,2 e recomendamos TLS 1,3.
- Configure API e registre as atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como [trabalhar com CloudTrail trilhas](#) no Guia AWS CloudTrail do usuário.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de FIPS 140-3 módulos criptográficos validados ao acessar AWS por meio de uma interface de linha de comando ou uma API, use um endpoint. FIPS Para obter mais informações sobre os FIPS endpoints disponíveis, consulte [Federal Information Processing Standard \(FIPS\) 140-3](#).

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de e-mail dos seus clientes, em marcações ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com AWS CloudShell ou Serviços da AWS usa o console, API, AWS CLI, ou AWS SDKs. Quaisquer dados inseridos em tags ou campos de texto de

formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, é altamente recomendável que você não inclua informações de credenciais no URL para validar sua solicitação para esse servidor.

Criptografia de dados

A criptografia de dados se refere à proteção dos dados quando estão em repouso enquanto estão armazenados AWS CloudShell e, quando em trânsito, eles viajam entre os AWS CloudShell terminais do serviço.

Criptografia em repouso usando AWS KMS

A criptografia em repouso refere-se à proteção de dados contra acesso não autorizado criptografando dados enquanto estão armazenados. Ao usar AWS CloudShell, você tem armazenamento persistente de 1 GB por AWS região sem nenhum custo. O armazenamento persistente está localizado em seu diretório inicial (\$HOME) e é privado para você. Ao contrário dos recursos de ambiente temporários que são reciclados após o término de cada sessão do shell, os dados do diretório inicial persistem.

A criptografia dos dados armazenados em AWS CloudShell é implementada usando chaves criptográficas fornecidas por AWS Key Management Service (AWS KMS). Esse é um AWS serviço gerenciado para criar e controlar as chaves mestras do cliente (CMKs) — as chaves de criptografia usadas para criptografar os dados do cliente que estão armazenados no AWS CloudShell ambiente. AWS CloudShell gera e gerencia chaves criptográficas para criptografar dados em nome dos clientes.

Criptografia em trânsito

Criptografia em trânsito refere-se a impedir os dados de serem interceptados enquanto eles se movem entre endpoints de comunicação.

Por padrão, toda comunicação de dados entre o computador do navegador da web do cliente e o computador baseado na nuvem AWS CloudShell é criptografada enviando tudo por meio de uma conexão HTTPS TLS /.

Você não precisa fazer nada para permitir o uso deHTTPS/TLSpara comunicação.

Identity and Access Management para AWS CloudShell

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. IAMos administradores controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar CloudShell os recursos. IAMé um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

- [Público](#)
- [Autenticando com identidades](#)
- [Gerenciando acesso usando políticas](#)
- [Como AWS CloudShell funciona com IAM](#)
- [Exemplos de políticas baseadas em identidade para AWS CloudShell](#)
- [Solução de problemas de identidade e acesso do AWS CloudShell](#)
- [Gerenciando AWS CloudShell o acesso e o uso com IAM políticas](#)

Público

A forma como você usa AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz CloudShell.

Usuário do serviço — Se você usar o CloudShell serviço para fazer seu trabalho, seu administrador fornecerá as credenciais e as permissões de que você precisa. À medida que você usa mais CloudShell recursos para fazer seu trabalho, talvez precise de permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se você não conseguir acessar um recurso no CloudShell, consulte [Solução de problemas de identidade e acesso do AWS CloudShell](#) .

Administrador de serviços — Se você é responsável pelos CloudShell recursos da sua empresa, provavelmente tem acesso total CloudShell a. É seu trabalho determinar quais CloudShell recursos e recursos seus usuários do serviço devem acessar. Em seguida, você deve enviar solicitações ao IAM administrador para alterar as permissões dos usuários do serviço. Revise as informações nesta página para entender os conceitos básicos doIAM. Para saber mais sobre como sua empresa pode usar IAM com CloudShell, consulte [Como AWS CloudShell funciona com IAM](#).

IAMadministrador — Se você for IAM administrador, talvez queira saber detalhes sobre como criar políticas para gerenciar o acesso CloudShell. Para ver exemplos de políticas CloudShell baseadas

em identidade que você pode usar em IAM, consulte. [Exemplos de políticas baseadas em identidade para AWS CloudShell](#)

Autenticando com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como IAM usuário ou assumindo uma IAM função. Usuário raiz da conta da AWS

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Os usuários (do IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você entra como uma identidade federada, seu administrador configurou previamente a federação de identidades usando IAM funções. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login em AWS, consulte [Como fazer login Conta da AWS](#) no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para você mesmo assinar solicitações, consulte [Assinar AWS API solicitações](#) no Guia IAM do usuário.

Independente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte [Autenticação multifator](#) no Guia AWS IAM Identity Center do usuário e [Uso da autenticação multifator \(MFA\) AWS no Guia do IAM usuário](#).

Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário raiz para tarefas diárias. Proteja as credenciais do usuário raiz e use-as para executar as tarefas que somente ele puder executar. Para ver a lista

completa de tarefas que exigem que você faça login como usuário raiz, consulte [Tarefas que exigem credenciais de usuário raiz](#) no Guia do IAM usuário.

Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas são acessadas Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos usar o AWS IAM Identity Center. Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter informações sobre o IAM Identity Center, consulte [O que é o IAM Identity Center?](#) no Guia do AWS IAM Identity Center usuário.

Grupos e usuários do IAM

Um [IAMusuário](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos confiar em credenciais temporárias em vez de criar IAM usuários com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com IAM os usuários, recomendamos que você alterne as chaves de acesso. Para obter mais informações, consulte [Altere as chaves de acesso regularmente para casos de uso que exigem credenciais de longo prazo](#) no Guia do IAMusuário.

Um [IAMgrupo](#) é uma identidade que especifica uma coleção de IAM usuários. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdminse conceder a esse grupo permissões para administrar IAM recursos.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários

têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte [Quando criar um IAM usuário \(em vez de uma função\)](#) no Guia do IAM usuário.

IAMfunções

Uma [IAMfunção](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas. É semelhante a um IAM usuário, mas não está associado a uma pessoa específica. Você pode assumir temporariamente uma IAM função no AWS Management Console [trocando de funções](#). Você pode assumir uma função chamando uma AWS API operação AWS CLI or ou usando uma personalizadaURL. Para obter mais informações sobre métodos de uso de funções, consulte [Usando IAM funções](#) no Guia IAM do usuário.

IAMfunções com credenciais temporárias são úteis nas seguintes situações:

- **Acesso de usuário federado:** para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter informações sobre funções para federação, consulte [Criação de uma função para um provedor de identidade terceirizado](#) no Guia IAM do usuário. Se você usa o IAM Identity Center, configura um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a uma função em. IAM Para obter informações sobre conjuntos de permissões, consulte [Conjuntos de Permissões](#) no Manual do Usuário do AWS IAM Identity Center .
- **Permissões temporárias IAM de IAM usuário** — Um usuário ou função pode assumir uma IAM função para assumir temporariamente permissões diferentes para uma tarefa específica.
- **Acesso entre contas** — Você pode usar uma IAM função para permitir que alguém (um diretor confiável) em uma conta diferente acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre funções e políticas baseadas em recursos para acesso entre contas, consulte [Acesso a recursos entre contas IAM no Guia](#) do IAM usuário.
- **Acesso entre serviços** — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicativos na Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões do principal de chamada, usando um perfil de serviço ou um perfil vinculado a serviço.

- Sessões de acesso direto (FAS) — Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. FAS as solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte [Encaminhar sessões de acesso](#).
- Função de serviço — Uma função de serviço é uma [IAM função](#) que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamente IAM. Para obter mais informações, consulte [Criação de uma função para delegar permissões a uma AWS service \(Serviço da AWS\)](#) no Guia do IAM usuário.
- Função vinculada ao serviço — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS). O serviço pode presumir a função de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.
- Aplicativos em execução na Amazon EC2 — Você pode usar uma IAM função para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma EC2 instância e fazendo AWS CLI AWS API solicitações. Isso é preferível a armazenar chaves de acesso na EC2 instância. Para atribuir uma AWS função a uma EC2 instância e disponibilizá-la para todos os aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que programas em execução na EC2 instância recebam credenciais temporárias. Para obter mais informações, consulte [Como usar uma IAM função para conceder permissões a aplicativos executados em EC2 instâncias da Amazon](#) no Guia IAM do usuário.

Para saber se usar IAM funções ou IAM usuários, consulte [Quando criar uma IAM função \(em vez de um usuário\)](#) no Guia do IAM usuário.

Gerenciando acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida

ou negada. A maioria das políticas é armazenada AWS como JSON documentos. Para obter mais informações sobre a estrutura e o conteúdo dos documentos de JSON política, consulte [Visão geral das JSON políticas](#) no Guia IAM do usuário.

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e funções não têm permissões. Para conceder permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

IAMas políticas definem permissões para uma ação, independentemente do método usado para realizar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função do AWS Management Console AWS CLI, do ou do AWS API.

Políticas baseadas em identidade

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Criação de IAM políticas no Guia](#) do IAMusuário.

As políticas baseadas em identidade podem ser categorizadas ainda adicionalmente como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte [Escolha entre políticas gerenciadas e políticas em linha no Guia](#) do IAMusuário.

Políticas baseadas no recurso

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado

pode executar nesse recurso e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas de uma política baseada IAM em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento JSON de política.

Amazon S3, AWS WAF, e Amazon VPC são exemplos de serviços que oferecem suporte. ACLs Para saber mais ACLs, consulte a [visão geral da lista de controle de acesso \(ACL\)](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- **Limites de permissões** — Um limite de permissões é um recurso avançado no qual você define as permissões máximas que uma política baseada em identidade pode conceder a uma IAM entidade (IAM usuário ou função). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade com seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte [Limites de permissões para IAM entidades](#) no Guia IAM do usuário.
- **Políticas de controle de serviço (SCPs)** — SCPs são JSON políticas que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em AWS Organizations. AWS Organizations é um serviço para agrupar e gerenciar centralmente várias Contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as suas contas. Os SCP limites de permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre Organizations e SCPs, consulte [Políticas de controle de serviços](#) no Guia AWS Organizations do Usuário.

- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em atributo. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte [Políticas de sessão](#) no Guia IAM do usuário.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de política estão envolvidos, consulte [Lógica de avaliação](#) de políticas no Guia IAM do usuário.

Como AWS CloudShell funciona com IAM

Antes de usar IAM para gerenciar o acesso ao CloudShell, saiba quais IAM recursos estão disponíveis para uso CloudShell.

IAMrecursos que você pode usar com AWS CloudShell

IAMrecurso	CloudShell apoio
Políticas baseadas em identidade	Sim
Políticas baseadas em recursos	Não
Ações das políticas	Sim
Atributos de políticas	Sim
Chaves de condição de política (específicas do serviço)	Sim
ACLs	Não
ABAC(tags nas políticas)	Não

IAMrecurso	CloudShell apoio
Credenciais temporárias	Sim
Sessões de acesso direto (FAS)	Não
Perfis de serviço	Não
Funções vinculadas ao serviço	Não

Para obter uma visão geral de como CloudShell e outros AWS serviços funcionam com a maioria dos IAM recursos, consulte [AWS os serviços que funcionam com IAM](#) no Guia do IAM usuário.

Políticas baseadas em identidade para CloudShell

Compatível com políticas baseadas em identidade: Sim

Políticas baseadas em identidade são documentos de políticas de JSON permissões que você pode anexar a uma identidade, como um IAM usuário, grupo de usuários ou função. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Criação de IAM políticas no Guia](#) do IAMusuário.

Com políticas IAM baseadas em identidade, você pode especificar ações e recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Você não pode especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou perfil ao qual ela está anexada. Para saber mais sobre todos os elementos que você pode usar em uma JSON política, consulte a [referência IAM JSON de elementos de política](#) no Guia IAM do usuário.

Exemplos de políticas baseadas em identidade para CloudShell

Para ver exemplos de políticas CloudShell baseadas em identidade, consulte. [Exemplos de políticas baseadas em identidade para AWS CloudShell](#)

Políticas baseadas em recursos dentro CloudShell

Suporte a políticas baseadas em recursos: não

Políticas baseadas em recursos são documentos JSON de política que você anexa a um recurso. Exemplos de políticas baseadas em recursos são políticas de confiança de IAM funções e políticas de bucket do Amazon S3. Em serviços que suportem políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações um principal especificado pode executar nesse recurso e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para habilitar o acesso entre contas, você pode especificar uma conta ou IAM entidades inteiras em outra conta como principal em uma política baseada em recursos. Adicionar uma entidade principal entre contas à política baseada em recurso é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um IAM administrador na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em recurso conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, [consulte Acesso a recursos entre contas IAM no](#) Guia do IAM usuário.

Ações políticas para CloudShell

Compatível com ações de políticas: Sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O `Action` elemento de uma JSON política descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da AWS API operação associada. Há algumas exceções, como ações somente com permissão que não têm uma operação correspondente. API Algumas operações também exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de CloudShell ações, consulte [Ações definidas por AWS CloudShell](#) na Referência de Autorização de Serviço. Algumas ações podem ter mais de uma API.

As ações de política CloudShell usam o seguinte prefixo antes da ação:

```
cloudshell
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

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

Para ver exemplos de políticas CloudShell baseadas em identidade, consulte [Exemplos de políticas baseadas em identidade para AWS CloudShell](#)

Recursos políticos para CloudShell

Compatível com recursos de políticas: Sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento Resource JSON de política especifica o objeto ou objetos aos quais a ação se aplica. As instruções devem incluir um elemento Resource ou NotResource. Como prática recomendada, especifique um recurso usando seu [Amazon Resource Name \(ARN\)](#). Isso pode ser feito para ações que oferecem compatibilidade com um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem compatibilidade com permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

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

Para ver uma lista dos tipos de CloudShell recursos e seus ARNs, consulte [Recursos definidos por AWS CloudShell](#) na Referência de Autorização de Serviço. Para saber com quais ações você pode especificar cada recurso, consulte [Ações definidas por AWS CloudShell](#). ARN

Para ver exemplos de políticas CloudShell baseadas em identidade, consulte [Exemplos de políticas baseadas em identidade para AWS CloudShell](#)

Chaves de condição de política para CloudShell

Compatível com chaves de condição de política específicas de serviço: Sim

Os administradores podem usar AWS JSON políticas para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Condition` (ou bloco `Condition`) permite que você especifique condições nas quais uma instrução estiver em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usem [agentes de condição](#), como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica AND. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma OR operação lógica. Todas as condições devem ser atendidas antes que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar condições. Por exemplo, você pode conceder permissão a um IAM usuário para acessar um recurso somente se ele estiver marcado com o nome de IAM usuário. Para obter mais informações, consulte [elementos de IAM política: variáveis e tags](#) no Guia IAM do usuário.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as [chaves de contexto de condição AWS global](#) no Guia IAM do usuário.

Para ver uma lista de chaves de CloudShell condição, consulte [Chaves de condição AWS CloudShell](#) na Referência de autorização de serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte [Ações definidas por AWS CloudShell](#).

Para ver exemplos de políticas CloudShell baseadas em identidade, consulte. [Exemplos de políticas baseadas em identidade para AWS CloudShell](#)

ACLsem CloudShell

SuportesACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento JSON de política.

ABAC com CloudShell

Suportes ABAC (tags nas políticas): Não

O controle de acesso baseado em atributos (ABAC) é uma estratégia de autorização que define permissões com base em atributos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a IAM entidades (usuários ou funções) e a muitos AWS recursos. Marcar entidades e recursos é a primeira etapa do ABAC. Em seguida, você cria ABAC políticas para permitir operações quando a tag do diretor corresponde à tag do recurso que ele está tentando acessar.

ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna complicado.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no [elemento de condição](#) de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial.

Para obter mais informações sobre ABAC, consulte [O que é ABAC?](#) no Guia do IAM usuário. Para ver um tutorial com etapas de configuração ABAC, consulte [Usar controle de acesso baseado em atributos \(ABAC\) no Guia](#) do IAM usuário.

Usando credenciais temporárias com CloudShell

Compatível com credenciais temporárias: Sim

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte [Serviços da AWS nesse trabalho IAM](#) no Guia do IAM usuário.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre a troca de funções, consulte [Alternando para uma função \(console\)](#) no Guia IAM do usuário.

Você pode criar manualmente credenciais temporárias usando o AWS CLI ou AWS API. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para obter mais informações, consulte [Credenciais de segurança temporárias em IAM](#).

Ao trocar de função, você estará usando um ambiente diferente. Você não pode trocar de função no mesmo AWS CloudShell ambiente.

Sessões de acesso direto para CloudShell

Suporta sessões de acesso direto (FAS): Não

Quando você usa um IAM usuário ou uma função para realizar ações em AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. FAS solicitações são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer FAS solicitações, consulte [Encaminhar sessões de acesso](#).

Funções de serviço para CloudShell

Compatível com perfis de serviço: não

Uma função de serviço é uma [IAM função](#) que um serviço assume para realizar ações em seu nome. Um IAM administrador pode criar, modificar e excluir uma função de serviço internamente em IAM. Para obter mais informações, consulte [Criação de uma função para delegar permissões a uma AWS service \(Serviço da AWS\)](#) no Guia do IAM usuário.

Warning

Alterar as permissões de uma função de serviço pode interromper CloudShell a funcionalidade. Edite as funções de serviço somente quando CloudShell fornecer orientação para fazer isso.

Funções vinculadas a serviços para CloudShell

Compatível com perfis vinculados ao serviço: Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode presumir a função de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um IAM administrador pode visualizar, mas não editar, as permissões das funções vinculadas ao serviço.

Exemplos de políticas baseadas em identidade para AWS CloudShell

Por padrão, usuários e funções não têm permissão para criar ou modificar CloudShell recursos. Eles também não podem realizar tarefas usando o AWS Management Console, AWS Command Line Interface (AWS CLI) ou AWS API. Para conceder permissão aos usuários para realizar ações nos recursos de que precisam, um IAM administrador pode criar IAM políticas. O administrador pode então adicionar as IAM políticas às funções e os usuários podem assumir as funções.

Para saber como criar uma política IAM baseada em identidade usando esses exemplos de documentos de JSON política, consulte [Criação de IAM políticas no Guia](#) do IAMusuário.

Para obter detalhes sobre ações e tipos de recursos definidos por CloudShell, incluindo o formato do ARNs para cada um dos tipos de recursos, consulte [Ações, recursos e chaves de condição AWS CloudShell na Referência de Autorização de Serviço](#).

Tópicos

- [Melhores práticas de política](#)
- [Usando o CloudShell console](#)
- [Permitir que usuários visualizem suas próprias permissões](#)

Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir CloudShell recursos em sua conta. Essas ações podem incorrer em custos para seus Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso.

Para obter mais informações, consulte [políticas AWS gerenciadas](#) ou [políticas AWS gerenciadas para funções de trabalho](#) no Guia IAM do usuário.

- Aplique permissões com privilégios mínimos — Ao definir permissões com IAM políticas, conceda somente as permissões necessárias para realizar uma tarefa. Você faz isso definindo as ações que podem ser executadas em atributos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre IAM como usar para aplicar permissões, consulte [Políticas e permissões IAM no](#) Guia IAM do usuário.
- Use condições nas IAM políticas para restringir ainda mais o acesso — Você pode adicionar uma condição às suas políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte [elementos IAM JSON da política: Condição](#) no Guia IAM do usuário.
- Use o IAM Access Analyzer para validar suas IAM políticas e garantir permissões seguras e funcionais — o IAM Access Analyzer valida políticas novas e existentes para que as políticas sigam a linguagem da IAM política (JSON) e as melhores práticas. IAM IAMO Access Analyzer fornece mais de 100 verificações de políticas e recomendações práticas para ajudá-lo a criar políticas seguras e funcionais. Para obter mais informações, consulte [Validação da política do IAM Access Analyzer](#) no Guia do IAM Usuário.
- Exigir autenticação multifatorial (MFA) — Se você tiver um cenário que exija IAM usuários ou um usuário root Conta da AWS, ative MFA para obter segurança adicional. Para exigir MFA quando API as operações são chamadas, adicione MFA condições às suas políticas. Para obter mais informações, consulte [Configurando o API acesso MFA protegido](#) no Guia do IAM usuário.

Para obter mais informações sobre as melhores práticas em IAM, consulte [as melhores práticas de segurança IAM no](#) Guia IAM do usuário.

Usando o CloudShell console

Para acessar o AWS CloudShell console, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os CloudShell recursos em seu Conta da AWS. Se você criar uma política baseada em identidade que seja mais restritiva do que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para AWS CLI ou AWS API. Em vez disso, permita o acesso somente às ações que correspondam à API operação que eles estão tentando realizar.

Para garantir que usuários e funções ainda possam usar o CloudShell console, anexe também a política CloudShell *ConsoleAccess* ou a política *ReadOnly* AWS gerenciada às entidades. Para obter mais informações, consulte [Adicionar permissões a um usuário](#) no Guia do IAM usuário.

Permitir que usuários visualizem suas próprias permissões

Este exemplo mostra como você pode criar uma política que permita IAM aos usuários visualizar as políticas embutidas e gerenciadas que estão anexadas à identidade do usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando o AWS CLI ou AWS API

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupForUser",
        "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": "*"    
  }  
]  
}
```

Solução de problemas de identidade e acesso do AWS CloudShell

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com CloudShell e IAM.

Tópicos

- [Não estou autorizado a realizar uma ação em CloudShell](#)
- [Não estou autorizado a realizar iam: PassRole](#)
- [Quero permitir que pessoas fora da minha Conta da AWS acessem meus CloudShell recursos](#)

Não estou autorizado a realizar uma ação em CloudShell

Se você receber uma mensagem de erro informando que não tem autorização para executar uma ação, é preciso atualizar suas políticas para permitir que você realize a ação.

O exemplo de erro a seguir ocorre quando o usuário IAM mateojackson tenta usar o console para ver detalhes sobre um *my-example-widget* recurso fictício, mas não tem as permissões fictícias `aws:GetWidget`.

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

Nesse caso, a política do usuário mateojackson deve ser atualizada para permitir o acesso ao recurso *my-example-widget* usando a ação `aws:GetWidget`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Não estou autorizado a realizar iam: PassRole

Se você receber um erro informando que não está autorizado a realizar a `iam:PassRole` ação, suas políticas devem ser atualizadas para permitir que você transfira uma função para CloudShell o.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando um IAM usuário chamado `marymajor` tenta usar o console para realizar uma ação no CloudShell. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

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

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas fora da minha Conta da AWS acessem meus CloudShell recursos

Você pode criar um perfil que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se é CloudShell compatível com esses recursos, consulte [Como AWS CloudShell funciona com IAM](#).
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte [Fornecer acesso a um IAM usuário em outro Conta da AWS de sua propriedade](#) no Guia do IAM usuário.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte [Fornecer Contas da AWS acesso a terceiros](#) no Guia do IAM usuário.
- Para saber como fornecer acesso por meio da federação de identidades, consulte [Fornecendo acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do IAM usuário.

- Para saber a diferença entre usar funções e políticas baseadas em recursos para acesso entre contas, consulte Acesso a [recursos entre contas IAM no Guia](#) do IAM usuário.

Gerenciando AWS CloudShell o acesso e o uso com IAM políticas

Com os recursos de gerenciamento de acesso que podem ser fornecidos por AWS Identity and Access Management, os administradores podem conceder permissões aos IAM usuários. Dessa forma, esses usuários podem acessar AWS CloudShell e usar os recursos do ambiente. Os administradores também podem criar políticas que especifiquem em um nível granular quais ações esses usuários podem realizar com o ambiente shell.

A maneira mais rápida de um administrador conceder acesso aos usuários é por meio de uma política AWS gerenciada. Uma [política AWS gerenciada](#) é uma política autônoma criada e administrada pela AWS. A seguinte política AWS gerenciada para AWS CloudShell pode ser anexada às IAM identidades:

- **AWS CloudShellFullAccess**: concede permissão para uso AWS CloudShell com acesso total a todos os recursos.

A **AWS CloudShellFullAccess** política usa o caractere curinga (*) para dar à IAM identidade (usuário, função ou grupo) acesso total CloudShell e recursos. Para obter mais informações sobre essa política, consulte [AWS CloudShellFullAccess](#) Guia do usuário da política AWS gerenciada.

Note

IAM identidades com as seguintes políticas AWS gerenciadas também podem ser CloudShell lançadas. No entanto, essas políticas fornecem permissões abrangentes. Portanto, recomendamos que você conceda essas políticas somente se elas forem essenciais para a função profissional de um IAM usuário.

- **[Administrador](#)**: fornece IAM aos usuários acesso total e permite que eles deleguem AWS permissões a todos os serviços e recursos do.
- **[Usuário avançado do desenvolvedor](#)**: permite que IAM os usuários executem tarefas de desenvolvimento de aplicativos e criem e configurem recursos e serviços que suportem o desenvolvimento AWS consciente de aplicativos.

Para obter mais informações sobre como anexar políticas gerenciadas, consulte [Adicionar permissões de IAM identidade \(console\)](#) no Guia do IAM usuário.

Gerenciando ações permitidas no AWS CloudShell uso de políticas personalizadas

Para gerenciar as ações que um IAM usuário pode executar CloudShell, crie uma política personalizada que use a política CloudShellPolicy gerenciada como modelo. Como alternativa, edite uma [política embutida](#) que esteja incorporada na IAM identidade relevante (usuário, grupo ou função).

Por exemplo, você pode permitir que IAM os usuários acessem CloudShell, mas impedir que eles encaminhem as credenciais do CloudShell ambiente usadas para fazer login. AWS Management Console

Important

Para iniciar a AWS CloudShell partir do AWS Management Console, um IAM usuário precisa de permissões para as seguintes ações:

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

Se uma dessas ações não for explicitamente permitida por uma política anexada, um erro de IAM permissões será retornado quando você tentar CloudShell iniciá-la.

AWS CloudShell permissões

Nome	Descrição da permissão concedida	Necessário para o lançamento CloudShell?
<code>cloudshell:CreateEnvironment</code>	Cria um CloudShell ambiente, recupera o layout no início da CloudShell sessão e salva o layout atual do aplicativo web no back-end. Essa permissão só espera * o valor de Resource conforme descrito em the section called “Exemplos de IAM políticas para CloudShell” .	Sim
<code>cloudshell:CreateSession</code>	Conecta-se a um CloudShell ambiente a partir do AWS Management Console.	Sim
<code>cloudshell:GetEnvironmentStatus</code>	Leia o status de um CloudShell ambiente.	Sim
<code>cloudshell>DeleteEnvironment</code>	Exclui um CloudShell ambiente.	Não
<code>cloudshell:GetFileDownloadURLs</code>	Gera Amazon URLs S3 pré-assinado que é usado para baixar arquivos usando CloudShell CloudShell a interface web. Isso não está disponível para VPC ambientes.	Não

Nome	Descrição da permissão concedida	Necessário para o lançamento CloudShell?
<code>cloudshell:GetFileUploadUrls</code>	Gera Amazon URLs S3 pré-assinado que é usado para fazer upload de arquivos usando CloudShell CloudShell a interface web. Isso não está disponível para VPC ambientes.	Não
<code>cloudshell:DescribeEnvironments</code>	Descreve os ambientes.	Não
<code>cloudshell:PutCredentials</code>	Encaminha as credenciais usadas para fazer login no. AWS Management Console CloudShell	Não
<code>cloudshell:StartEnvironment</code>	Inicia um CloudShell ambiente que está parado.	Sim
<code>cloudshell:StopEnvironment</code>	Interrompe um CloudShell ambiente em execução.	Não

Exemplos de IAM políticas para CloudShell

Os exemplos a seguir mostram como as políticas podem ser criadas para restringir quem pode acessar CloudShell. Os exemplos também mostram as ações que podem ser executadas no ambiente shell.

A política a seguir impõe uma negação total do acesso CloudShell e de seus recursos.

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

```

    "Effect": "Deny",
    "Action": [
      "cloudshell:*"
    ],
    "Resource": "*"
  }]
}

```

A política a seguir permite que IAM os usuários acessem CloudShell , mas impede que eles gerem pré-assinados URLs para upload e download de arquivos. Os usuários ainda podem transferir arquivos de e para o ambiente, usando clientes como wget, por exemplo.

```

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

```

A política a seguir permite que IAM os usuários acessem CloudShell. No entanto, a política impede que as credenciais que você usou para fazer login sejam encaminhadas para o CloudShell ambiente. AWS Management Console IAM os usuários com essa política precisam configurar manualmente suas credenciais dentro CloudShell dela.

```

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

```



```

    {
      "Sid": "AllowUsingCloudshell",
      "Effect": "Allow",
      "Action": [
        "cloudshell:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DenyCredentialForwarding",
      "Effect": "Deny",
      "Action": [
        "cloudshell:PutCredentials"
      ],
      "Resource": "*"
    }
  ]
}

```

A política a seguir permite que IAM os usuários AWS CloudShell criem ambientes.

```

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

```

IAMPermissões necessárias para criar e usar CloudShell VPC ambientes

Para criar e usar CloudShell VPC ambientes, o IAM administrador deve habilitar o acesso a EC2 permissões VPC específicas da Amazon. Esta seção lista as EC2 permissões da Amazon necessárias para criar e usar VPC ambientes.

Para criar VPC ambientes, a IAM política atribuída à sua função deve incluir as seguintes EC2 permissões da Amazon:

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

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

Recomendamos também incluir:

- `ec2>DeleteNetworkInterface`

Note

Essa permissão não é obrigatória, mas é necessária CloudShell para limpar o ENI recurso (ENI criado para CloudShell VPC ambientes marcados com `ManagedByCloudShell` chave) criado por ele. Se essa permissão não estiver habilitada, você deverá limpar manualmente o ENI recurso após cada uso do CloudShell VPC ambiente.

IAM política que concede CloudShell acesso total, incluindo acesso a VPC

O exemplo a seguir mostra como habilitar permissões completas, incluindo acesso VPC a 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": ""
    }
  }
}
]
}

```

Usando chaves de IAM condição para VPC ambientes

Você pode usar chaves CloudShell de condição específicas para VPC configurações para fornecer controles de permissão adicionais para seus VPC ambientes. Você também pode especificar as sub-redes e os grupos de segurança que o VPC ambiente pode ou não usar.

CloudShell suporta as seguintes chaves de condição nas IAM políticas:

- `CloudShell:VpcIds`— Permitir ou negar um ou mais VPCs
- `CloudShell:SubnetIds`— Permitir ou negar uma ou mais sub-redes
- `CloudShell:SecurityGroupIds`— Permitir ou negar um ou mais grupos de segurança

Note

Se as permissões dos usuários com acesso a CloudShell ambientes públicos forem modificadas para adicionar restrições à `cloudshell:createEnvironment` ação, eles ainda poderão acessar o ambiente público existente. No entanto, se você quiser modificar uma IAM política com essa restrição e desativar seu acesso ao ambiente público existente, você deve primeiro atualizar a IAM política com a restrição e, em seguida, garantir que cada CloudShell usuário em sua conta exclua manualmente o ambiente público existente usando a interface de usuário da CloudShell web (Ações → Excluir CloudShell ambiente).

Exemplo de políticas com chaves de condição para VPC configurações

Os exemplos a seguir demonstram como usar chaves de condição para VPC configurações. Depois de criar uma instrução de política com as restrições desejadas, acrescente a instrução de política para o usuário ou a função de destino.

Garanta que os usuários criem somente VPC ambientes e neguem a criação de ambientes públicos

Para garantir que os usuários possam criar somente VPC ambientes, use a permissão de negação conforme mostrado no exemplo a seguir:

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

```

        "cloudshell:VpcIds": "true"
    }
}
]
}

```

Negar aos usuários acesso a grupos específicos VPCs, sub-redes ou grupos de segurança

Para negar aos usuários acesso a VPCs informações específicas, use `StringEquals` para verificar o valor da `cloudshell:VpcIds` condição. O exemplo a seguir nega aos usuários o acesso a `vpc-1` e `vpc-2`:

```

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

```

Para negar aos usuários acesso a VPCs informações específicas, use `StringEquals` para verificar o valor da `cloudshell:SubnetIds` condição. O exemplo a seguir nega aos usuários o acesso a `subnet-1` e `subnet-2`:

```

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

```

```

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

```

Para negar aos usuários acesso a VPCs informações específicas, use `StringEquals` para verificar o valor da `cloudshell:SecurityGroupIds` condição. O exemplo a seguir nega aos usuários o acesso a `sg-1` e `sg-2`:

```

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

```

```
}
```

Permita que os usuários criem ambientes com VPC configurações específicas

Para permitir que os usuários acessem VPCs dados específicos, use `StringEquals` para verificar o valor da `cloudshell:VpcIds` condição. O exemplo a seguir permite que os usuários acessem `vpc-1` e `vpc-2`:

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

Para permitir que os usuários acessem VPCs dados específicos, use `StringEquals` para verificar o valor da `cloudshell:SubnetIds` condição. O exemplo a seguir permite que os usuários acessem `subnet-1` e `subnet-2`:

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



```

    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
      "ForAllValues:StringEquals": {
        "cloudshell:SubnetIds": [
          "subnet-1",
          "subnet-2"
        ]
      }
    }
  ]
}

```

Para permitir que os usuários acessem VPCs dados específicos, use `StringEquals` para verificar o valor da `cloudshell:SecurityGroupIds` condição. O exemplo a seguir permite que os usuários acessem `sg-1` e `sg-2`:

```

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

```

Permissões para acessar Serviços da AWS

CloudShell usa as IAM credenciais que você usou para entrar no AWS Management Console

Note

Para usar IAM as credenciais que você usou para entrar no AWS Management Console, você deve ter `cloudshell:PutCredentials` permissão.

Esse recurso de pré-autenticação CloudShell facilita o uso AWS CLI. No entanto, um IAM usuário ainda precisa de permissões explícitas para Serviços da AWS as chamadas na linha de comando.

Por exemplo, suponha que IAM os usuários precisem criar buckets do Amazon S3 e fazer upload de arquivos como objetos para eles. Você pode criar uma política que permita explicitamente essas ações. O IAM console fornece um [editor visual](#) interativo que orienta o processo de criação de um documento JSON de política formatado. Depois que a política for criada, você poderá anexá-la à IAM identidade relevante (usuário, grupo ou função).

Para obter mais informações sobre como anexar políticas gerenciadas, consulte [Adicionar permissões de IAM identidade \(console\)](#) no Guia do IAM usuário.

Registro e monitoramento em AWS CloudShell

Este tópico descreve como você pode registrar e monitorar a AWS CloudShell atividade e o desempenho com CloudTrail.

Monitorando a atividade com CloudTrail

AWS CloudShell é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS service (Serviço da AWS) em AWS CloudShell. CloudTrail captura todas as API chamadas para AWS CloudShell eventos. As chamadas capturadas incluem chamadas do AWS CloudShell console e chamadas de código para AWS CloudShell API o.

Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon Simple Storage Service (Amazon S3). Isso inclui eventos para AWS CloudShell.

Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode descobrir uma variedade de informações sobre uma solicitação. Por exemplo, você pode determinar a solicitação que foi feita AWS CloudShell, saber o endereço IP do qual a solicitação foi feita, quem fez a solicitação e quando ela foi feita.

AWS CloudShell em CloudTrail

A tabela a seguir lista os AWS CloudShell eventos que são salvos no arquivo de CloudTrail log.

Note

AWS CloudShell evento que inclui:

- *indica que é uma chamada sem mutação (somente leitura). API
- A palavra `Environment` está relacionada ao ciclo de vida do ambiente computacional que hospeda a experiência do shell.
- A palavra `Layout` restaura todas as guias do navegador no CloudShell terminal.

CloudShell Eventos em CloudTrail

Nome do evento	Descrição
<code>createEnvironment</code>	Ocorre quando um CloudShell ambiente é criado.
<code>createSession</code>	Ocorre quando um CloudShell ambiente é conectado a partir do AWS Management Console.
<code>deleteEnvironment</code>	Ocorre quando um CloudShell ambiente é excluído.
<code>deleteSession</code>	Ocorre quando a sessão na CloudShell guia que está sendo executada na guia atual do navegador é excluída.
<code>getEnvironmentStatus*</code>	Ocorre quando o status de um CloudShell ambiente é recuperado.
<code>getFileDownloadUrls*</code>	Ocorre quando o Amazon URLs S3 pré-assinado que é usado para baixar arquivos usando

Nome do evento	Descrição
	CloudShell CloudShell a interface web é gerado.
<code>getFileUploadUrls*</code>	Ocorre quando o Amazon URLs S3 pré-assinado que é usado para fazer upload de arquivos usando CloudShell CloudShell a interface web é gerado.
<code>cloudshell:DescribeEnvironments</code>	Descreve os ambientes.
<code>getLayout*</code>	Ocorre quando o CloudShell layout no início da sessão é recuperado.
<code>putCredentials</code>	Ocorre quando as credenciais usadas para fazer login no AWS Management Console to CloudShell são encaminhadas.
<code>redeemCode*</code>	Ocorre quando o fluxo de trabalho para recuperar o token de atualização no CloudShell ambiente começa. Posteriormente, você pode usar esse token no <code>putCredentials</code> comando para acessar o CloudShell ambiente.
<code>sendHeartBeat</code>	Ocorre para confirmar que a CloudShell sessão está ativa.
<code>startEnvironment</code>	Ocorre quando um CloudShell ambiente é iniciado.
<code>stopEnvironment</code>	Ocorre quando um CloudShell ambiente em execução é interrompido.
<code>updateLayout</code>	Ocorre quando o layout atual do aplicativo web no back-end é salvo.

Eventos que incluem a palavra “Layout” restauram todas as guias do navegador no CloudShell terminal.

EventBridge regras para AWS CloudShell ações

Com EventBridge as regras, você especifica uma ação de destino a ser tomada ao EventBridge receber um evento que corresponda à regra. Você pode definir uma regra que especifique uma ação de destino a ser tomada com base em uma AWS CloudShell ação registrada como um evento em um arquivo de CloudTrail log.

Por exemplo, você pode [criar EventBridge regras AWS CLI](#) usando o `put-rule` comando. Uma `put-rule` chamada deve conter pelo menos um `EventPattern` ou `ScheduleExpression`. As regras com `EventPatterns` são acionadas quando um evento correspondente é observado. Os `EventPattern` quatro AWS CloudShell eventos:

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

Para obter mais informações, consulte [Eventos e padrões de eventos EventBridge no](#) Guia do EventBridge usuário da Amazon.

Validação de conformidade para AWS CloudShell

Audidores terceirizados avaliam a segurança e a conformidade dos AWS serviços como parte de vários programas de AWS conformidade.

AWS CloudShell está no escopo dos seguintes programas de conformidade:

SOC

AWS Os relatórios de controles do sistema e da organização (SOC) são relatórios independentes de exames terceirizados que demonstram como AWS alcança os principais controles e objetivos de conformidade.

Serviço	SDK	SOC1,2,3
AWS CloudShell	CloudShell	✓

PCI

O Payment Card Industry Data Security Standard (PCIDSS) é um padrão proprietário de segurança da informação administrado pelo PCI Security Standards Council, fundado pela American Express, Discover Financial Services, JCB International, MasterCard Worldwide e Visa Inc.

Serviço	SDK	PCI
AWS CloudShell	CloudShell	✓

ISOe CSA STAR certificações e serviços

AWS tem certificação para conformidade com ISO/IEC 27001:2013, 27017:2015, 27018:2019, 27701:2019, 22301:2019, 9001:2015 e v4.0. CSA STAR CCM

Serviço	SDK	ISOe CSA STAR certificações e serviços
AWS CloudShell	CloudShell	✓

FedRamp

O Programa Federal de Gerenciamento de Riscos e Autorizações (FedRAMP) é um programa do governo dos EUA que oferece uma abordagem padrão para avaliação de segurança, autorização e monitoramento contínuo de produtos e serviços em nuvem.

Serviço	SDK	Fed RAMP Moderado (Leste/Oeste)	RAMPA Alta do Fed (1GovCloud)
AWS CloudShell	CloudShell	✓	✓

DoD CC SRG

O Guia de Requisitos de Segurança de Computação em Nuvem () do Departamento de Defesa (DoDSRG) fornece um processo padronizado de avaliação e autorização para que os provedores de

serviços em nuvem (CSPs) obtenham uma autorização provisória do DoD, para que possam atender aos clientes do DoD.

Os serviços que passarem pela SRG avaliação e autorização do DoD CC terão o seguinte status:

- Avaliação da organização de avaliação terceirizada (3PAO): Este serviço está atualmente sendo avaliado por nosso avaliador terceirizado.
- Revisão do Conselho de Autorização Conjunta (JAB): Este serviço está passando por uma JAB revisão.
- Avaliação da Agência de Sistemas de Informação de Defesa (DISA): Este serviço está atualmente passando por uma DISA revisão.

Serviço	SDK	DoD CC SRG IL2 (Leste/Oeste)	DoD CC () SRG IL2 GovCloud	DoD CC () SRG IL4 GovCloud	DoD CC () SRG IL5 GovCloud	DoD CC SRG IL6 (Região AWS Secreta)
AWS CloudShell	CloudShell	3 PAO Avaliação	N/D	N/D	N/D	N/D

HIPAA BAA

A Lei de Portabilidade e Responsabilidade de Seguros de Saúde de 1996 (HIPAA) é uma lei federal que exigiu a criação de padrões nacionais para impedir que informações confidenciais de saúde do paciente fossem divulgadas sem o consentimento ou conhecimento do paciente.

AWS permite que as entidades cobertas e seus parceiros comerciais sujeitos HIPAA a processem, armazenem e transmitam com segurança informações de saúde protegidas (PHI). Além disso, a partir de julho de 2013, AWS oferece um Adendo de Associado Comercial padronizado (BAA) para esses clientes.

Serviço	SDK	HIPAA BAA
AWS CloudShell	CloudShell	✓

IRAP

O Programa de Avaliadores Registrados de Segurança da Informação (IRAP) permite que os clientes do governo australiano validem se os controles apropriados estão em vigor e determinem o modelo de responsabilidade apropriado para atender aos requisitos do Manual de Segurança da Informação do Governo Australiano (ISM) produzido pelo Centro Australiano de Segurança Cibernética (ACSC).

Serviço	Namespace*	IRAPprotegido
AWS CloudShell	N/D	✓

*Os namespaces ajudam você a identificar serviços em seu ambiente. AWS Por exemplo, quando você cria IAM políticas, trabalha com Amazon Resource Names (ARNs) e lê AWS CloudTrail registros.

MTCS

O Multi-Tier Cloud Security (MTCS) é um padrão operacional de gerenciamento de segurança de Cingapura (SPRINGSS 584), baseado nos padrões ISO 27001/02 do Sistema de Gerenciamento de Segurança da Informação (). ISMS

Serviço	SDK	Leste dos EUA (Ohio)	Leste dos EUA (N. da Virgínia)	Oeste dos EUA (Oregon)	Oeste dos EUA (N. da Califórnia)	Cingapura	Seul
AWS CloudShell	CloudShell	✓	✓	✓	N/D	N/D	N/D

C5

O Catálogo de Controles de Conformidade de Computação em Nuvem (C5) é um esquema de atestação apoiado pelo governo alemão introduzido na Alemanha pelo Escritório Federal de Segurança da Informação (BSI) para ajudar as organizações a demonstrar segurança

operacional contra ataques cibernéticos comuns ao usar serviços em nuvem dentro do contexto das “Recomendações de segurança para provedores de nuvem” do governo alemão.

Serviço	SDK	C5
AWS CloudShell	CloudShell	✓

ENSAItto

O esquema de acreditação ENS (Esquema Nacional de Seguridad) foi desenvolvido pelo Ministério das Finanças e Administração Pública e pelo CCN (Centro Criptológico Nacional). Isso inclui princípios básicos e requisitos mínimos necessários para a proteção adequada das informações.

Serviço	SDK	ENSAItto
AWS CloudShell	CloudShell	✓

FINMA

A Autoridade Supervisora do Mercado Financeiro Suíço (FINMA) é a reguladora independente dos mercados financeiros da Suíça. AWS O alinhamento da com FINMA os requisitos demonstra nosso compromisso contínuo em atender às elevadas expectativas dos provedores de serviços em nuvem estabelecidas pelos reguladores e clientes suíços de serviços financeiros.

Serviço	SDK	FINMA
AWS CloudShell	CloudShell	✓

PiTuKri

AWS o alinhamento com PiTuKri os requisitos demonstra nosso compromisso contínuo em atender às elevadas expectativas dos provedores de serviços em nuvem estabelecidas pela Agência Finlandesa de Transportes e Comunicações, Traficom.

Serviço	SDK	PiTuKri
AWS CloudShell	CloudShell	✓

Para obter uma lista de AWS serviços que estão no escopo de programas de conformidade específicos, consulte [AWSServiços no escopo do programa de conformidade AWS](#) . Para obter informações gerais, consulte Programas de [AWS conformidade Programas AWS](#) de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte [Baixando relatórios no AWS Artifact](#) .

Sua responsabilidade de conformidade ao usar AWS CloudShell é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentos aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- [Guias de início rápido sobre segurança e conformidade](#) de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos focados na segurança e na conformidade em. AWS
- Documento técnico [sobre arquitetura para HIPAA segurança e conformidade — Este whitepaper](#) descreve como as empresas podem usar AWS para criar aplicativos compatíveis. HIPAA
- AWS Recursos de <https://aws.amazon.com/compliance/resources/> de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- [Avaliação de recursos com regras](#) no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- [AWS Security Hub](#)— Esse AWS serviço fornece uma visão abrangente do seu estado de segurança interno, AWS que ajuda você a verificar sua conformidade com os padrões e as melhores práticas do setor de segurança.

Resiliência em AWS CloudShell

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade. AWS As regiões fornecem várias zonas de disponibilidade fisicamente separadas e isoladas, conectadas a redes de baixa latência, alta taxa de transferência e alta redundância. Com as zonas de disponibilidade, é possível projetar e operar aplicativos e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são altamente

disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de datacenters tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte [Infraestrutura AWS global](#).

Além da infraestrutura AWS global, AWS CloudShell oferece suporte aos seguintes recursos para atender às suas necessidades de resiliência e backup de dados:

- Confirme os arquivos que você cria e adiciona AWS CodeCommit. Este é um serviço de controle de versão hospedado pela Amazon Web Services que pode ser usado para armazenar e gerenciar ativos de maneira privada na nuvem. Esses ativos podem consistir em documentos, código-fonte e arquivos binários. Para obter mais informações, consulte [Usando CodeCommit em AWS CloudShell](#).
- Use AWS CLI chamadas para especificar arquivos em seu diretório inicial AWS CloudShell e adicioná-los como objetos nos buckets do Amazon S3. Para ver um exemplo, consulte [Introdução ao AWS CloudShell](#).

Segurança da infraestrutura em AWS CloudShell

Como serviço gerenciado, AWS CloudShell é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte [AWS Cloud Security](#). Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte [Proteção](#) de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa API chamadas AWS publicadas para acessar AWS CloudShell pela rede. Os clientes devem oferecer suporte para:

- Segurança da camada de transporte (TLS). Exigimos TLS 1,2 e recomendamos TLS 1,3.
- Suítes de criptografia com sigilo direto perfeito (), como (Ephemeral PFS Diffie-Hellman) ou DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Além disso, as solicitações devem ser assinadas usando uma ID de chave de acesso e uma chave de acesso secreta associada a um IAM principal. Ou você pode usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Note

Por padrão, instale AWS CloudShell automaticamente os patches de segurança para os pacotes do sistema de seus ambientes computacionais.

Práticas recomendadas de segurança para AWS CloudShell

As melhores práticas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas práticas recomendadas podem não ser apropriadas ou suficientes para seu ambiente, recomendamos que você as trate como considerações úteis em vez de prescrições.

Algumas práticas recomendadas de segurança para AWS CloudShell

- Use IAM permissões e políticas para controlar o acesso AWS CloudShell e garantir que os usuários possam realizar somente as ações (por exemplo, baixar e carregar arquivos) exigidas por sua função. Para obter mais informações, consulte [Gerenciando o AWS CloudShell acesso e o uso com IAM políticas](#).
- Não inclua dados confidenciais em suas IAM entidades, como usuários, funções ou nomes de sessões.
- Mantenha o atributo Safe Paste habilitado para detectar possíveis riscos de segurança no texto que você copiou de fontes externas. O Safe Paste é habilitado por padrão. Para obter mais informações sobre como usar a colagem segura para texto com várias linhas, consulte [Usando a colagem segura para texto com várias linhas](#).
- Familiarize-se com o [Modelo de Responsabilidade de Segurança Compartilhada](#) se você instalar aplicativos de terceiros no ambiente computacional do AWS CloudShell.
- Prepare mecanismos de reversão antes de editar scripts de shell que afetem a experiência de shell do usuário. Para obter mais informações sobre como modificar o ambiente de shell padrão, consulte [Modificar seu shell com scripts](#).
- Armazene seu código de forma segura em um sistema de controle de versão, por exemplo, [AWS CodeCommit](#).

AWS CloudShell Segurança FAQs

A seguir estão as respostas às perguntas mais frequentes sobre segurança para CloudShell.

- [Quais AWS processos e tecnologias são usados quando você inicia CloudShell e inicia uma sessão de shell?](#)
- [É possível restringir o acesso à rede CloudShell?](#)
- [Posso personalizar meu CloudShell ambiente?](#)
- [Onde meu diretório \\$HOME está realmente armazenado no Nuvem AWS?](#)
- [É possível criptografar meu diretório \\$HOME?](#)
- [Posso executar uma verificação de vírus no meu diretório \\$HOME?](#)

Quais AWS processos e tecnologias são usados quando você inicia CloudShell e inicia uma sessão de shell?

Ao fazer login AWS Management Console, você insere suas credenciais de IAM usuário. E, quando você inicia a CloudShell partir da interface do console, essas credenciais são usadas em chamadas para o CloudShell API que criam um ambiente computacional para o serviço. Em seguida, uma AWS Systems Manager sessão é criada para o ambiente computacional e CloudShell envia comandos para essa sessão.

[Voltar à lista de segurança FAQs](#)

É possível restringir o acesso à rede CloudShell?

Para ambientes públicos, não é possível restringir o acesso à rede. Se quiser restringir o acesso à rede, você deve habilitar a permissão para criar somente VPC ambientes e negar a criação de ambientes públicos.

Para obter mais informações, consulte [Garantir que os usuários criem somente VPC ambientes e neguem a criação de ambientes públicos](#).

Para CloudShell VPC ambientes, as configurações de rede são herdadas do seu VPC. O uso CloudShell em a VPC permite que você controle o acesso à rede do seu CloudShell VPC ambiente.

[Voltar à lista de segurança FAQs](#)

Posso personalizar meu CloudShell ambiente?

Você pode baixar e instalar utilitários e outros softwares de terceiros para o seu CloudShell ambiente. Somente o software instalado em seu diretório \$HOME persiste entre as sessões.

Conforme definido pelo [modelo de responsabilidade compartilhada da AWS](#), você é responsável pela configuração e gerenciamento necessários dos aplicativos que você instala.

[Voltar à lista de segurança FAQs](#)

Onde meu diretório **\$HOME** está realmente armazenado no Nuvem AWS?

Para ambientes públicos, a infraestrutura para armazenar dados em seu \$HOME é fornecida pelo Amazon S3.

Para VPC ambientes, seu \$HOME diretório é excluído quando seu VPC ambiente expira (após 20 a 30 minutos de inatividade) ou quando você exclui ou reinicia seu ambiente.

[Voltar à lista de segurança FAQs](#)

É possível criptografar meu diretório **\$HOME**?

Não, não é possível criptografar seu \$HOME diretório com sua própria chave. Mas CloudShell criptografa o conteúdo \$HOME do seu diretório enquanto o armazena no Amazon S3.

[Voltar à lista de segurança FAQs](#)

Posso executar uma verificação de vírus no meu diretório **\$HOME**?

No momento, não é possível executar uma verificação de vírus no seu diretório \$HOME. O suporte para esse atributo está sendo analisado.

[Voltar à lista de segurança FAQs](#)

Posso restringir a entrada ou saída de dados para o meu? CloudShell

Para restringir a entrada ou saída, recomendamos que você use um CloudShell VPC ambiente. O \$HOME diretório de um VPC ambiente é excluído quando seu VPC ambiente expira (após 20 a 30 minutos de inatividade) ou quando você exclui ou reinicia seu ambiente. No menu Ações, as opções de upload e download não estão disponíveis para VPC ambientes.

[Voltar à lista de segurança FAQs](#)

AWS CloudShell ambiente computacional: especificações e software

Quando você inicia AWS CloudShell, um ambiente computacional baseado no [Amazon Linux 2023](#) é criado para hospedar a experiência do shell. O ambiente é configurado com [recursos computacionais \(v CPU e memória\)](#) e fornece uma ampla variedade de [softwares pré-instalados](#) que podem ser acessados pela interface da linha de comando. Certifique-se de que qualquer software instalado no ambiente computacional tenha patches e esteja atualizado. Você também pode configurar seu ambiente padrão instalando software e modificando scripts de shell.

Recursos do ambiente de computação

Cada ambiente AWS CloudShell computacional recebe os seguintes recursos CPU e os seguintes recursos de memória:

- 1 v CPU (unidade de processamento central virtual)
- 2 GiB RAM

Além disso, o ambiente é provisionado com a seguinte configuração de armazenamento:

- Armazenamento persistente de 1 GB (o armazenamento persiste após o término da sessão)

Para obter mais informações, consulte [Armazenamento persistente](#).

CloudShell requisitos de rede

WebSockets

CloudShell depende do WebSocket protocolo, que permite a comunicação interativa bidirecional entre o navegador da web do usuário e o CloudShell serviço na AWS nuvem. Se você estiver usando um navegador em uma rede privada, o acesso seguro à Internet provavelmente é facilitado por servidores proxy e firewalls. WebSocket a comunicação geralmente pode atravessar servidores proxy sem problemas. Mas, em alguns casos, os servidores proxy WebSockets impedem o funcionamento correto. Se esse problema ocorrer, sua CloudShell interface relata o seguinte erro: `Failed to open sessions : Timed out while opening the session.`

Se esse erro ocorrer repetidamente, consulte a documentação do seu servidor proxy para garantir que ele esteja configurado para permitir WebSockets. Como alternativa, você pode entrar em contato com o administrador do sistema da sua rede.

Note

Se quiser definir permissões granulares por meio de listas de permissões específicas URLs, você pode adicionar parte da URL que a AWS Systems Manager sessão usa para abrir uma WebSocket conexão para enviar entradas e receber saídas. (Seus AWS CloudShell comandos são enviados para essa sessão do Systems Manager.)

O formato para isso StreamUrl usado pelo Systems Manager é `wss://ssmmessages.region.amazonaws.com/v1/data-channel/session-id?stream=(input|output)`.

A região representa o identificador de uma AWS região suportada por AWS Systems Manager, como `us-east-2` a região Leste dos EUA (Ohio).

Como o ID da sessão é criado após o início bem-sucedido de uma sessão específica do Systems Manager, você só pode especificar `wss://`

`ssmmessages.region.amazonaws.com` ao atualizar sua lista de permissões. URL

Para obter mais informações, consulte a [StartSession](#) operação na AWS Systems Manager API Referência.

Software pré-instalado

Note

Como o ambiente de AWS CloudShell desenvolvimento é atualizado regularmente para fornecer acesso ao software mais recente, não fornecemos números de versão específicos nesta documentação. Em vez disso, descrevemos como você pode verificar qual versão está instalada. Para verificar a versão instalada, insira o nome do programa seguido pela opção `--version` (por exemplo, `git --version`).

Shells

Shells pré-instalados

Nome	Descrição	Informações sobre a versão
Bash	O shell Bash é o aplicativo de shell padrão para AWS CloudShell.	<code>bash --version</code>
PowerShell (empurrão)	Oferecendo uma interface de linha de comando e suporte à linguagem de script, PowerShell é construído com base nos da Microsoft .NETTempo de execução da linguagem de comando. PowerShell usa comandos leves chamados cmdlets de aceitar e retornar. NETobjetos.	<code>pwsh --version</code>
Z Shell (zsh)	O Z Shell, também conhecido como zsh, é uma versão estendida do Bourne Shell que oferece suporte aprimorado à personalização de temas e plug-ins.	<code>zsh --version</code>

AWS interfaces de linha de comando (CLI)

CLI

Nome	Descrição	Informações sobre a versão
AWS CDK Kit de ferramentas CLI	O AWS CDK kit de ferramentas, o CLI comando, <code>cdk</code> , é a principal ferramenta que interage com seu AWS CDK	<code>cdk --version</code>

Nome	Descrição	Informações sobre a versão
	<p>aplicativo. Ele executa seu aplicativo, interroga o modelo de aplicativo que você definiu e produz e implanta os AWS CloudFormation modelos gerados pelo. AWS CDK</p> <p>Para obter mais informações, consulte AWS CDK Toolkit.</p>	
AWS CLI	<p>AWS CLI É uma interface de linha de comando que você pode usar para gerenciar vários AWS serviços a partir da linha de comando e automatizá-los usando scripts. Para obter mais informações, consulte Trabalhando com AWS serviços em AWS CloudShell.</p> <p>Para obter informações sobre como você pode garantir que está usando a maior parte da up-to-date AWS CLI versão 2, consulte AWS CLI Instalando em seu diretório inicial.</p>	<pre>aws --version</pre>

Nome	Descrição	Informações sobre a versão
EB CLI	<p>AWS Elastic Beanstalk CLI Ele fornece uma interface de linha de comando para simplificar a criação, a atualização e o monitoramento de ambientes a partir de um repositório local.</p> <p>Para obter mais informações, consulte Como usar a interface de linha de comando CLI (EB) do Elastic Beanstalk no Guia do desenvolvedor. AWS Elastic Beanstalk</p>	<code>eb --version</code>
Amazon ECS CLI	<p>A interface de linha de comando (ECS) do Amazon Elastic Container Service (Amazon CLI) fornece comandos de alto nível para simplificar a criação, a atualização e o monitoramento de clusters e tarefas.</p> <p>Para obter mais informações, consulte Como usar a interface de linha de ECS comando da Amazon no Amazon Elastic Container Service Developer Guide.</p>	<code>ecs-cli --version</code>

Nome	Descrição	Informações sobre a versão
AWS SAM CLI	<p>AWS SAM CLI é uma ferramenta de linha de comando que opera em um AWS Serverless Application Model modelo e código de aplicativo. Você pode realizar várias tarefas. Isso inclui invocar funções Lambda localmente, criar um pacote de implantação para seu aplicativo sem servidor e implantar seu aplicativo sem servidor na nuvem. AWS</p> <p>Para obter mais informações, consulte a referência de AWS SAM CLI comandos no Guia do AWS Serverless Application Model desenvolvedor.</p>	<pre>sam --version</pre>

Nome	Descrição	Informações sobre a versão
AWS Tools for PowerShell	<p>AWS Tools for PowerShell São PowerShell módulos que são construídos com base na funcionalidade exposta pelo AWS SDK for .NET. Com AWS Tools for PowerShell, você pode criar scripts de operações em seus AWS recursos a partir da linha de PowerShell comando.</p> <p>AWS CloudShell pré-instala a versão modularizada (AWS.Tools) do. AWS Tools for PowerShell</p> <p>Para obter mais informações, consulte Usando as AWS ferramentas PowerShell no Guia do AWS Tools for PowerShell usuário.</p>	<pre>powershell --Command ' Get-Module -ListAvailable -Name AWS.Tools .Common '</pre>

Tempos de execução e AWSSDKs: Node.js e Python 3

Tempos de execução e AWS SDKs

Nome	Descrição	Informações sobre a versão
Node.js (com npm)	<p>O Node.js é um JavaScript tempo de execução projetado para facilitar a aplicação de técnicas de programação assíncrona. Para obter mais informações, consulte a documentação no site oficial do Node.js.</p>	<ul style="list-style-type: none"> • Node.js: <code>node --version</code> • npm: <code>npm --version</code>

Nome	Descrição	Informações sobre a versão
	<p>O npm é um gerenciador de pacotes que fornece acesso a um registro on-line de JavaScript módulos. Para obter mais informações, consulte a documentação no site oficial do npm.</p>	
SDK para JavaScript em Node.js	<p>O kit de desenvolvimento de software (SDK) ajuda a simplificar a codificação fornecendo JavaScript objetos para AWS serviços como Amazon S3, Amazon EC2, DynamoDB e Amazon SWF. Para mais informações, consulte o Guia do desenvolvedor do AWS SDK for JavaScript.</p>	<pre>npm -g ls --depth 0 2>/dev/null grep aws-sdk</pre>

Nome	Descrição	Informações sobre a versão
Python	<p>O Python 3 está pronto para uso no ambiente shell. O Python 3 agora é considerado a versão padrão da linguagem de programação (o suporte ao Python 2 terminou em janeiro de 2020). Para obter mais informações, consulte a documentação no site oficial do Python.</p> <p>Além disso, o pip, o instalador de pacotes para Python, está pré-instalado. Você pode usar esse programa de linha de comando para instalar pacotes Python a partir dos índices on-line, como o Python Package Index. Para obter mais informações, consulte a documentação fornecida pela Python Packaging Authority.</p>	<ul style="list-style-type: none">• Python 3: <code>python3 --version</code>• pip: <code>pip3 --version</code>

Nome	Descrição	Informações sobre a versão
SDK para Python (Boto3)	<p>Boto é o kit de desenvolvimento de software (SDK) que os desenvolvedores de Python usam para criar, configurar e gerenciar Serviços da AWS, como Amazon EC2 e Amazon S3. O SDK fornece um acesso orientado a API por objetos e de baixo nível a Serviços da AWS.</p> <p>Para obter mais informações, consulte a documentação do Boto3.</p>	<code>pip3 list grep boto3</code>

Ferramentas de desenvolvimento e utilitários de shell

Ferramentas de desenvolvimento e utilitários de shell

Nome	Descrição	Informações sobre a versão
bash-completion	<p>O bash-completion é um conjunto de funções de shell que permitem o preenchimento automático de comandos ou argumentos parcialmente digitados pressionando a tecla Tab. Você pode encontrar os pacotes compatíveis com o bash-completion em <code>/usr/share/bash-completion/completions</code>.</p>	<code>dnf info bash-completion</code>

Nome	Descrição	Informações sobre a versão
	<p>Para configurar o preenchimento automático para os comandos de um pacote, o arquivo do programa deve ser originado. Por exemplo, para configurar o preenchimento automático para comandos do Git, adicione a seguinte linha <code>.bashrc</code> para que o recurso esteja disponível sempre que AWS CloudShell sua sessão começar:</p> <pre>source /usr/share/ bash-completion/ completions/git</pre> <p>Se você quiser usar scripts de preenchimento personalizados, adicione-os ao seu diretório inicial persistente (<code>\$HOME</code>) e origine-os diretamente no <code>.bashrc</code>.</p> <p>Para obter mais informações, consulte a README página do projeto em GitHub.</p>	

Nome	Descrição	Informações sobre a versão
CodeCommit utilitário para Git	<p>git-remote-codecommit é um utilitário que fornece um método simples para enviar e extrair código de CodeCommit repositórios por meio da extensão do Git. É o método recomendado para oferecer suporte a conexões feitas com acesso federado, provedores de identidade e credenciais temporárias.</p> <p>Para obter mais informações, consulte Etapas de configuração para HTTPS conexões AWS CodeCommit com git-remote-codecommit no Guia AWS CodeCommit do usuário.</p>	<pre>pip3 list grep git-remote-codecommit</pre>
Git	<p>O Git é um sistema de controle de versão distribuído que dá suporte às práticas modernas de desenvolvimento de software por meio de fluxos de trabalho de ramificações e preparação de conteúdo. Para obter mais informações, consulte a página de documentação no site oficial do Git.</p>	<pre>git --version</pre>

Nome	Descrição	Informações sobre a versão
iputils	O pacote iputils contém utilitários para redes Linux. Para obter mais informações sobre os utilitários fornecidos, consulte o repositório iputils em. GitHub	Exemplos de uma ferramenta iputils: <code>arping -V</code>
jq	O utilitário jq analisa dados JSON formatados para produzir uma saída modificada pelos filtros da linha de comando. Para obter mais informações, consulte o manual jq hospedado em. GitHub	<code>jq --version</code>
kubectl	kubectl é uma ferramenta de linha de comando para comunicação com o plano de controle de um cluster Kubernetes, usando o Kubernetes. API	<code>kubectl --version</code>
make	O utilitário make usa <code>makefiles</code> para automatizar conjuntos de tarefas e organizar a compilação de código. Para obter mais informações, consulte a documentação do GNU Make.	<code>make --version</code>

Nome	Descrição	Informações sobre a versão
man	O comando man fornece páginas do manual para utilitários e ferramentas de linha de comando. Por exemplo, <code>man ls</code> retorna a página de manual para o comando <code>ls</code> que lista os conteúdos dos diretórios. Para obter mais informações, consulte a entrada na Wikipédia na página man .	<code>man --version</code>
nano	O nano é um editor pequeno e fácil de usar para interface baseada em texto. Para obter mais informações, consulte a documentação do GNU nano .	<code>nano --version</code>
procps	O procps é um utilitário de administração do sistema que você pode usar para monitorar e interromper os processos atualmente em execução. Para obter mais informações, consulte o README arquivo que lista os programas que podem ser executados com procps .	<code>ps --version</code>

Nome	Descrição	Informações sobre a versão
SSHcliente	SSHos clientes usam o protocolo secure shell para comunicações criptografadas com um computador remoto. Open SSH é o SSH cliente pré-instalado. Para obter mais informações, consulte o SSHsite Open mantido pelo OpenBSD.	<code>ssh -V</code>
sudo	Com o utilitário sudo, os usuários podem executar um programa com as permissões de segurança de outro usuário, normalmente o superusuário. O Sudo é útil quando é necessário instalar aplicativos como administrador do sistema. Para obter mais informações, consulte o Manual do Sudo .	<code>sudo --version</code>
tar	O tar é um utilitário de linha de comando que você pode usar para agrupar vários arquivos em um único arquivo (geralmente chamado de tarball). Para obter mais informações, consulte a documentação do GNU tar .	<code>tar --version</code>

Nome	Descrição	Informações sobre a versão
tmux	O tmux é um multiplexer de terminal que você pode usar para executar diferentes programas simultaneamente em várias janelas. Para obter mais informações, consulte um blog que forneça uma introdução concisa ao tmux .	tmux -V
unzip	Para obter mais informações, consulte zip/unzip .	
vim	O vim é um editor personalizável com o qual você pode interagir por meio de uma interface baseada em texto. Para obter mais informações, consulte os recursos de documentação fornecidos em vim.org .	vim --version
wget	O wget é um programa de computador usado para recuperar conteúdo de servidores web especificados por endpoints na linha de comando. Para obter mais informações, consulte a documentação do GNU Wget .	wget --version

Nome	Descrição	Informações sobre a versão
zip/unzip	<p>Os utilitários zip/unzip usam um formato de arquivo que oferece compactação de dados sem perda de dados. Chame o comando zip para agrupar e compactar arquivos em um único arquivo. Use unzip para extrair arquivos de um arquivo em um diretório especificado.</p>	<pre>unzip --version zip --version</pre>

Nome	Descrição	Informações sobre a versão
Docker	<p>O Docker é uma plataforma aberta para desenvolvimento, envio e execução de aplicativos. O Docker permite que você separe seus aplicativos da sua infraestrutura para que você possa entregar software rapidamente. Ele permite que você crie Dockerfiles internamente e crie AWS CloudShell ativos do Docker com CDK. Para obter informações sobre quais AWS regiões são compatíveis com o Docker, consulte AWS Regiões suportadas para AWS CloudShell. Você deve estar ciente de que o Docker tem espaço limitado no ambiente. Se você tiver imagens individuais grandes ou muitas imagens pré-existentes do Docker, isso pode causar problemas. Para obter mais informações sobre o Docker, consulte o guia de documentação do Docker.</p>	<code>docker --version</code>

AWS CLI Instalando em seu diretório inicial

Como o resto do software pré-instalado em seu CloudShell ambiente, a AWS CLI ferramenta é atualizada automaticamente com atualizações programadas e patches de segurança. Se quiser garantir que você tenha a up-to-date versão mais recente do AWS CLI, você pode optar por instalar manualmente a ferramenta no diretório inicial do shell.

⚠ Important

Você precisa instalar manualmente sua cópia do AWS CLI no diretório inicial para que ela esteja disponível na próxima vez que você iniciar uma CloudShell sessão. Essa instalação é necessária porque os arquivos adicionados aos diretórios fora do \$HOME são excluídos após a conclusão de uma sessão de shell. Além disso, depois de instalar essa cópia do AWS CLI, ela não é atualizada automaticamente. Em outras palavras, é de sua responsabilidade gerenciar as atualizações e os patches de segurança.

Para obter mais informações sobre o Modelo de Responsabilidade AWS Compartilhada, consulte [Proteção de dados em AWS CloudShell](#).

Para instalar AWS CLI

1. Na linha de CloudShell comando, use o `curl` comando para transferir uma cópia compactada do AWS CLI instalado para o shell:

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

2. Descompacte a pasta compactada:

```
unzip awscliv2.zip
```

3. Para adicionar a ferramenta a uma pasta especificada, execute o AWS CLI instalador:

```
sudo ./aws/install --install-dir /home/cloudshell-user/usr/local/aws-cli --bin-dir /home/cloudshell-user/usr/local/bin
```

Se for instalado com sucesso, a linha de comando exibirá a seguinte mensagem:

```
You can now run: /home/cloudshell-user/usr/local/bin/aws --version
```

4. Para sua conveniência, recomendamos que você também atualize a variável ambiental `PATH` para não precisar especificar o caminho para a instalação da ferramenta ao executar comandos `aws`:

```
export PATH=/home/cloudshell-user/usr/local/bin:$PATH
```

Note

Se você desfizer essa alteração em `PATH`, os `aws` comandos que não apresentam um caminho especificado usarão a versão pré-instalada do AWS CLI por padrão.

Instalação de software de terceiros no ambiente do shell

Note

Recomendamos que você revise o [Modelo de Responsabilidade de Segurança Compartilhada](#) antes de instalar qualquer aplicativo de terceiros no ambiente computacional AWS CloudShell da empresa.

Por padrão, todos os AWS CloudShell usuários têm permissões `sudo`. Portanto, você pode usar o comando `sudo` para instalar software que ainda não esteja disponível no ambiente computacional do shell. Por exemplo, você pode usar `sudo` com o utilitário de DNF gerenciamento de pacotes para instalar `cowsay`, que gera imagens ASCII artísticas de uma vaca com uma mensagem:

```
sudo dnf install cowsay
```

Em seguida, você pode iniciar o programa recém-instalado digitando `echo "Welcome to AWS CloudShell" | cowsay`.

Important

Utilitários de gerenciamento de pacotes, como `dnf`, instalam programas em diretórios `/usr/bin` (por exemplo), que são reciclados quando sua sessão de shell termina. Isso significa que o software adicional é instalado e usado por sessão.

Como modificar seu shell com scripts

Se quiser modificar o ambiente de shell padrão, você pode editar um script de shell que é executado toda vez que o ambiente de shell é inicializado. O script `.bashrc` é executado sempre que o shell `bash` padrão é iniciado.

⚠ Warning

Se você modificar incorretamente o arquivo `.bashrc`, talvez não consiga acessar o ambiente do shell posteriormente. É uma boa prática fazer uma cópia do arquivo antes da edição. Você também pode reduzir o risco abrindo dois shells durante a edição do `.bashrc`. Se você perder o acesso em um shell, ainda terá o login no outro shell e poderá reverter quaisquer alterações.

Se você perder o acesso após modificar incorretamente `.bashrc` ou qualquer outro arquivo, poderá retornar AWS CloudShell às configurações padrão [excluindo seu](#) diretório pessoal.

No procedimento, você modificará o script `.bashrc` para que seu ambiente de shell alterne automaticamente para a execução do Z shell.

1. Abra o `.bashrc` usando um editor de texto (Vim, por exemplo):

```
vim .bashrc
```

2. Na interface do editor, pressione a tecla `I` para começar a editar e adicione o seguinte:

```
zsh
```

3. Para sair e salvar o arquivo `.bashrc` editado, pressione `Esc` para entrar no modo de comando do Vim e digite o seguinte:

```
:wq
```

4. Use o comando `source` para recarregar o arquivo `.bashrc`:

```
source .bashrc
```

Quando a interface da linha de comando estiver disponível novamente, o símbolo do prompt será alterado para `%` para indicar que agora você está usando o Z shell.

AWS CloudShell migrando de 0 AL2 para 023 AL2

AWS CloudShell, que foi baseado no Amazon Linux 2 (AL2), migrou para o Amazon Linux 2023 (AL2023). Para obter mais informações sobre AL2 023, consulte [O que é o Amazon Linux 2023 \(AL2023\) no Guia](#) do usuário do Amazon Linux 2023.

Com o AL2 023, você pode continuar acessando seu CloudShell ambiente existente com todas as ferramentas fornecidas pela CloudShell. Para obter mais informações sobre as ferramentas disponíveis, consulte [Software pré-instalado](#).

AL2O 023 fornece várias melhorias nas ferramentas de desenvolvimento, incluindo versões mais recentes de pacotes, como Node.js 18 e Python 3.9.

Note

Em AL2 203, Python 2 não é mais fornecido com seu CloudShell ambiente.

Para obter mais informações sobre as principais diferenças entre AL2 e AL2 023, consulte [Comparando o Amazon Linux 2 e o Amazon Linux 2023](#) no Guia do usuário do Amazon Linux 2023.

Se tiver dúvidas, entre em contato com o [AWS Support](#). Você também pode procurar respostas e postar dúvidas no [AWS re:Post](#). Ao entrar AWS re:Post, talvez seja necessário fazer login em AWS.

AWS CloudShell Migração FAQs

A seguir estão as respostas para algumas perguntas comuns sobre a migração de AL2 para AL2 023 com AWS CloudShell.

- [Essa migração afetará algum dos meus outros AWS recursos, como EC2 instâncias da Amazon em execuçãoAL2?](#)
- [Quais são os pacotes que serão alterados com a migração para AL2 023?](#)
- [Posso optar por não migrar?](#)
- [Posso criar um backup do meu ambiente AWS CloudShell ?](#)

A migração para AL2 023 afetará algum dos meus outros AWS recursos, como EC2 instâncias da Amazon em AL2 execução?

Nenhum serviço ou recurso além do seu AWS CloudShell ambiente é afetado por essa migração. Isso inclui recursos que você pode ter criado ou acessado internamente AWS CloudShell. Por exemplo, se você criou uma EC2 instância da Amazon em execução nela, AL2 ela não será migrada para AL2 023.

Quais são os pacotes que foram alterados com a migração para AL2 023?

AWS CloudShell atualmente, os ambientes incluem software pré-instalado. Para saber mais sobre a lista completa de softwares pré-instalados, consulte Software [pré-instalado](#). AWS CloudShell continuarão entregando esses pacotes, com exceção do Python 2. Para ver a diferença completa entre os pacotes fornecidos por AL2 e AL2 023, consulte [Comparando AL2 e AL2 023](#). Para clientes com requisitos específicos de pacotes e versões que não serão mais atendidos após a migração para o AL2 023, recomendamos entrar em contato com o AWS Support para enviar uma solicitação.

Posso optar por não migrar?

Não, você não pode optar por não migrar. AWS CloudShell os ambientes são gerenciados por AWS, portanto, todos os ambientes foram atualizados para AL2 023.

Posso criar um backup do meu AWS CloudShell ambiente?

AWS CloudShell continuará mantendo o diretório inicial do usuário. Para obter mais informações, consulte [Service Quotas e restrições para o AWS CloudShell](#). Se você tiver arquivos ou configurações armazenados em sua pasta inicial e quiser criar um backup para ela, conclua a [Etapa 6: criar um backup do diretório inicial](#).

Solução de problemas AWS CloudShell

Durante o uso AWS CloudShell, você pode encontrar problemas, como ao iniciar CloudShell ou executar tarefas importantes usando a interface de linha de comando do shell. As informações abordadas neste capítulo incluem como solucionar alguns dos problemas comuns que você pode encontrar.

Para obter respostas a uma variedade de perguntas sobre CloudShell, consulte [AWS CloudShell FAQs](#). Também é possível pesquisar respostas e postar perguntas no [Fórum de Discussão do AWS CloudShell](#). Ao entrar nesse fórum, pode ser que você precise fazer login na AWS. Você também pode [entrar em contato conosco](#) diretamente.

Solucionar de problemas de erros

Ao encontrar algum dos seguintes erros indexados, você pode usar as seguintes soluções para corrigi-los.

Tópicos

- [Erro: “Não foi possível iniciar o ambiente. Para tentar novamente, atualize o navegador ou reinicie selecionando Ações, Reiniciar” AWS CloudShell](#)
- [Erro: “Não foi possível iniciar o ambiente. Você não tem as permissões necessárias. Peça ao IAM administrador que conceda acesso a AWS CloudShell”](#)
- [Não é possível acessar a linha de AWS CloudShell comando](#)
- [Não é possível executar ping em endereços IP externos](#)
- [Houve alguns problemas ao preparar seu terminal](#)
- [As teclas de seta não funcionam corretamente em PowerShell](#)
- [Web Sockets não suportados causam uma falha no início das sessões CloudShell](#)
- [Não é possível importar o módulo AWSPowerShell.NetCore](#)
- [O Docker não está em execução ao usar AWS CloudShell](#)
- [O Docker ficou sem espaço em disco](#)
- [docker push está atingindo o tempo limite e continua tentando novamente](#)
- [Não consigo acessar recursos dentro VPC do meu AWS CloudShell VPC ambiente](#)

- [O ENI usado AWS CloudShell pelo meu VPC ambiente não está limpo](#)
- [O usuário com CreateEnvironment permissão somente para VPC ambientes também tem acesso a AWS CloudShell ambientes públicos](#)
- [As credenciais não estão funcionando CloudShell](#)

Erro: “Não foi possível iniciar o ambiente. Para tentar novamente, atualize o navegador ou reinicie selecionando Ações, Reiniciar” AWS CloudShell

Problema: Quando você tenta iniciar a AWS CloudShell partir do AWS Management Console, seu acesso é negado mesmo depois de ter exigido as permissões do IAM administrador e ter atualizado o navegador ou reiniciado. CloudShell

Solução: entre em contato com [AWS Support](#).

[\(Voltar ao início\)](#)

Erro: “Não foi possível iniciar o ambiente. Você não tem as permissões necessárias. Peça ao IAM administrador que conceda acesso a AWS CloudShell”

Problema: ao tentar iniciar a AWS CloudShell partir do AWS Management Console, você tem acesso negado e é notificado de que não tem as permissões necessárias.

Causa: A IAM identidade que você está usando para acessar AWS CloudShell não tem as IAM permissões necessárias.

Solução: solicite que seu IAM administrador forneça as permissões necessárias. Eles podem fazer isso adicionando uma política AWS gerenciada anexada (AWSCloudShellFullAccess) ou uma política embutida incorporada. Para obter mais informações, consulte [Gerenciando AWS CloudShell o acesso e o uso com IAM políticas](#).

[\(Voltar ao início\)](#)

Não é possível acessar a linha de AWS CloudShell comando

Problema: depois de modificar um arquivo usado pelo ambiente computacional, você não pode acessar a linha de comando em. AWS CloudShell

Solução: Se você perder o acesso após modificar incorretamente `.bashrc` ou qualquer outro arquivo, poderá retornar AWS CloudShell às configurações padrão [excluindo seu](#) diretório pessoal.

[\(Voltar ao início\)](#)

Não é possível executar ping em endereços IP externos

Problema: ao executar um comando ping na linha de comando (por exemplo, `ping amazon.com`), você recebe a seguinte mensagem.

```
ping: socket: Operation not permitted
```

Causa: O utilitário ping usa o Internet Control Message Protocol (ICMP) para enviar pacotes de solicitações de eco para um host de destino. Ele espera que um eco responda do destino. Como o ICMP protocolo não está habilitado AWS CloudShell, o utilitário ping não opera no ambiente computacional do shell.

Solução: Devido ao fato ICMP de não ser suportado no AWS CloudShell, você pode executar o seguinte comando para instalar o Netcat. O Netcat é um utilitário de rede de computadores para leitura e gravação em conexões de rede usando TCP ouUDP.

```
sudo yum install nc
nc -zv www.amazon.com 443
```

[\(Voltar ao início\)](#)

Houve alguns problemas ao preparar seu terminal

Problema: ao tentar acessar AWS CloudShell usando o navegador Microsoft Edge, você não consegue iniciar uma sessão de shell e o navegador exibe uma mensagem de erro.

Causa: AWS CloudShell não é compatível com versões anteriores do Microsoft Edge. Você pode acessar AWS CloudShell usando as quatro versões principais mais recentes dos navegadores compatíveis.

Solução: instale uma versão atualizada do navegador Edge do [site da Microsoft](#).

[\(Voltar ao início\)](#)

As teclas de seta não funcionam corretamente em PowerShell

Problema: em operação normal, você pode usar as teclas de seta para navegar pela interface da linha de comando e examinar seu histórico de comandos para trás e para frente. Mas, quando você pressiona as teclas de seta em determinadas versões de PowerShell ativado AWS CloudShell, as letras podem ser emitidas incorretamente.

Causa: A situação em que as teclas de seta produzem letras incorretamente é um problema conhecido nas versões PowerShell 7.2.x em execução no Linux.

Solução: Para remover as sequências de escape que modificam o comportamento das teclas de seta, edite o arquivo PowerShell de perfil e defina a `$PSStyle PlainText` variável como.

1. Na linha de AWS CloudShell comando, digite o comando a seguir para abrir o arquivo de perfil.

```
vim ~/.config/powershell/Microsoft.PowerShell_profile.ps1
```

Note

Se você já estiver conectado PowerShell, você também pode abrir o arquivo de perfil no editor com o comando a seguir.

```
vim $PROFILE
```

2. No editor, vá até o final do texto existente do arquivo, pressione `i` para entrar no modo de inserção e adicione a seguinte declaração.

```
$PSStyle.OutputRendering = 'PlainText'
```

3. Depois de fazer a edição, pressione `Esc` para entrar no modo de comando. Em seguida, insira o seguinte comando para salvar o arquivo e sair do editor.

```
:wq
```

Note

Suas alterações entrarão em vigor na próxima vez que você começar PowerShell.

[\(Voltar ao início\)](#)

Web Sockets não suportados causam uma falha no início das sessões CloudShell

Problema: Ao tentar iniciar AWS CloudShell, você recebe repetidamente a seguinte mensagem:`Failed to open sessions : Timed out while opening the session.`

Causa: CloudShell depende do WebSocket protocolo, que permite a comunicação interativa bidirecional entre seu navegador da web e AWS CloudShell. Se você estiver usando um navegador em uma rede privada, o acesso seguro à Internet provavelmente é facilitado por servidores proxy e firewalls. WebSocket a comunicação geralmente pode atravessar servidores proxy sem problemas. Mas, em alguns casos, os servidores proxy WebSockets impedem o funcionamento correto. Se esse problema ocorrer, não CloudShell será possível iniciar uma sessão de shell e a tentativa de conexão eventualmente expirará.

Solução: o tempo limite de conexão pode ser causado por um problema que não seja WebSockets incompatível. Se for esse o caso, primeiro atualize a janela do navegador onde a interface da linha de CloudShell comando está localizada.

Se você ainda estiver recebendo erros de tempo limite após a atualização, consulte a documentação do seu servidor proxy. E certifique-se de que seu servidor proxy esteja configurado para permitir Web Sockets. Como alternativa, consulte o administrador do sistema da sua rede.

Note

Digamos que você queira definir permissões granulares por meio de listas de permissões específicas. URLs Você pode adicionar parte do URL que a AWS Systems Manager sessão usa para abrir uma WebSocket conexão para enviar entradas e receber saídas. Seus AWS CloudShell comandos são enviados para essa sessão do Systems Manager.

O formato para StreamUrl isso usado pelo Systems Manager é `wss://ssmmessages.region.amazonaws.com/v1/data-channel/session-id?stream=(input|output)`.

A região representa o identificador de região de uma Região da AWS que é suportada por AWS Systems Manager. Por exemplo, `us-east-2` é o identificador de região para a região Leste dos EUA (Ohio).

Como o ID da sessão é criado após o início bem-sucedido de uma sessão específica do Systems Manager, você só pode especificar `wss://`

`ssmmessages.region.amazonaws.com` quando atualizar sua lista de permissões. URL Para obter mais informações, consulte a [StartSession](#) operação na AWS Systems Manager APIReferência.

[\(Voltar ao início\)](#)

Não é possível importar o módulo **AWSPowerShell.NetCore**

Problema: Quando você importa `AWSPowerShell o. NetCore` módulo in PowerShell by `Import-Module -Name AWSPowerShell.NetCore`, você recebe a seguinte mensagem de erro:

Import-Module: O módulo especificado ' . AWSPowerShell NetCore' não foi carregado porque nenhum arquivo de módulo válido foi encontrado em nenhum diretório do módulo.

Causa: O `AWSPowerShell.NetCore` módulo é substituído pelos módulos `AWS.Tools` por serviço em. AWS CloudShell

Solução: qualquer instrução de importação explícita pode não ser mais necessária ou precisar ser alterada para o módulo `AWS.Tools` por serviço relacionado.

Example

Example

- Na maioria dos casos, desde que nenhum tipo `.Net` seja usado, você não precisa de nenhuma instrução de importação explícita. Veja a seguir exemplos de instruções de importação.
 - `Get-S3Bucket`
 - `(Get-EC2Instance).Instances`
- Se forem usados tipos `.Net`, importe o módulo de nível de serviço (`AWS.Tools.<Service>`). Veja a seguir um exemplo de sintaxe.

```
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()
```

Para obter mais informações, consulte o [anúncio da versão 4](#) do AWS Tools for PowerShell.

[\(Voltar ao início\)](#)

O Docker não está em execução ao usar AWS CloudShell

Problema: o Docker não está funcionando corretamente durante o uso AWS CloudShell. Você recebe a seguinte mensagem de erro:`docker: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?.`

Solução: tente reiniciar seu ambiente. Essa mensagem de erro pode ocorrer quando você executa o Docker AWS CloudShell em uma GovCloud região que não oferece suporte a ele. Verifique se você está executando o Docker nas AWS regiões suportadas. Para obter uma lista das regiões nas quais o Docker está disponível, consulte [AWS Regiões suportadas](#) para AWS CloudShell

O Docker ficou sem espaço em disco

Problema: Você está recebendo a seguinte mensagem de erro:`ERROR: failed to solve: failed to register layer: write [...]: no space left on device.`

Causa: O Dockerfile está excedendo o espaço disponível em disco. AWS CloudShell Isso pode ser causado por grandes imagens individuais ou por muitas imagens pré-existentes do Docker.

Solução: Execute `df -h` para encontrar o uso do disco. Execute `sudo du -sh /folder/folder1` para avaliar o tamanho de determinadas pastas que você acha que podem ser grandes e considere excluir outros arquivos para liberar espaço. Uma opção seria considerar a remoção de imagens não utilizadas do Docker executando `docker rmi` Você deve estar ciente de que o Docker tem espaço limitado no ambiente. Para obter mais informações sobre o Docker, consulte o guia de [documentação do Docker](#).

docker push está atingindo o tempo limite e continua tentando novamente

Problema: quando você `docker push` executa, o tempo limite é atingido e continua tentando novamente sem sucesso.

Causa: Isso pode ser causado como resultado da falta de permissões, do envio para o repositório errado ou da falta de autenticação.

Solução: Para tentar resolver esse problema, verifique se você está enviando para o repositório correto. Execute `docker login` para autenticar corretamente. Certifique-se de ter todas as permissões necessárias para enviar para um ECR repositório da Amazon.

Não consigo acessar recursos dentro VPC do meu AWS CloudShell VPC ambiente

Problema: Não consigo acessar recursos VPC durante o uso do meu AWS CloudShell VPC ambiente.

Causa: Seu AWS CloudShell VPC ambiente herda as configurações de rede do seu VPC.

Solução: Para resolver esse problema, verifique se o seu VPC está configurado corretamente para acessar seus recursos. Para obter mais informações, consulte a VPC documentação [Connect your VPC to other networks](#) e a documentação Network Access Analyzer [Network Access Analyzer](#). Você pode encontrar o IPv4 endereço que o AWS CloudShell VPC ambiente está usando executando o comando `ip -a` dentro do seu ambiente no prompt da linha de comando ou na página do VPC console.

O ENI usado AWS CloudShell pelo meu VPC ambiente não está limpo

Problema: Não é possível limpar o ENI usado AWS CloudShell pelo meu VPC ambiente.

Causa: `ec2:DeleteNetworkInterface` a permissão não está habilitada para sua função.

Solução: para resolver esse problema, certifique-se de que a `ec2:DeleteNetworkInterface` permissão esteja habilitada para sua função, conforme mostrado no seguinte exemplo de script:

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

O usuário com **CreateEnvironment** permissão somente para VPC ambientes também tem acesso a AWS CloudShell ambientes públicos

Problema: o usuário restrito com `CreateEnvironment` permissão somente para VPC ambientes também pode acessar AWS CloudShell ambientes públicos.

Causa: Ao limitar `CreateEnvironment` as permissões somente para a criação de VPC ambientes e se você já tiver criado um ambiente público, você manterá seu acesso ao CloudShell ambiente público existente até que esse ambiente seja excluído usando a interface de usuário da web. Mas se você nunca usou CloudShell antes, não terá acesso a ambientes públicos.

Solução: para restringir o acesso a AWS CloudShell ambientes públicos, o IAM administrador deve primeiro atualizar a IAM política com a restrição e, em seguida, o usuário deve excluir manualmente o ambiente público existente usando a interface de usuário da AWS CloudShell web. (Ações → Excluir CloudShell ambiente).

As credenciais não estão funcionando CloudShell

Problema: Ao tentar fazer uma AWS CLI chamada de CloudShell, você recebe a mensagem de erro `Internal Server Error`.

Causa: A seguir estão as possíveis causas desse problema:

- A `putCredentials` API chamada CloudShell usada para atualizar as credenciais falhou. A API chamada pode falhar devido à falta de IAM permissões para `putCredentials` ação. Para obter mais informações, consulte [???](#). Se você já tiver IAM permissões para `putCredentials` ação, a API chamada poderá falhar devido a problemas de rede ou operacionais com CloudShell.
- Suas credenciais não são mais válidas porque a sessão do AWS console expirou, mas seu CloudShell ambiente ainda está em execução. Quando suas credenciais não são mais válidas, CloudShell não consegue fazer nenhuma API chamada.

Solução: tente atualizar a página da Web se você já tiver IAM as permissões necessárias para `putCredentials` a ação. Se o problema não for resolvido e você continuar recebendo o erro, entre em contato com o [AWS Support](#).

AWS Regiões suportadas para AWS CloudShell

Esta seção aborda a lista de AWS regiões suportadas e regiões de adesão para AWS CloudShell. Para obter uma lista de pontos finais de AWS serviço e cotas para CloudShell, consulte a [AWS CloudShell página](#) no. Referência geral da Amazon Web Services

A seguir estão as AWS regiões suportadas pelo CloudShell Docker e pelo CloudShell VPC ambiente:

- Leste dos EUA (Ohio)
- Leste dos EUA (N. da Virgínia)
- Oeste dos EUA (N. da Califórnia)
- Oeste dos EUA (Oregon)
- África (Cidade do Cabo)
- Ásia-Pacífico (Hong Kong)
- Ásia-Pacífico (Jacarta)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Osaka)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Singapura)
- Ásia-Pacífico (Sydney)
- Ásia-Pacífico (Tóquio)
- Canadá (Central)
- Europa (Frankfurt)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Milão)
- Europa (Paris)
- Europa (Estocolmo)
- Oriente Médio (Barém)
- Oriente Médio (UAE)
- América do Sul (São Paulo)

GovCloud Regiões

A seguir estão as GovCloud regiões com suporte para CloudShell:

- AWS GovCloud (Leste dos EUA)
- AWS GovCloud (Oeste dos EUA)

Atualmente, o Docker e o CloudShell VPC ambiente não estão disponíveis nas GovCloud regiões.

Cotas e restrições de serviço para AWS CloudShell

Esta página descreve as restrições e service quotas que se aplicam às seguintes áreas:

- [Armazenamento persistente](#)
- [Uso mensal](#)
- [Tamanho do comando](#)
- [Shells simultâneos](#)
- [Sessões de shell](#)
- [Acesso à rede e transferência de dados](#)
- [Arquivos do sistema e páginas recarregadas](#)

Armazenamento persistente

Com AWS CloudShell isso, você tem armazenamento persistente de 1 GB para cada um Região da AWS, sem nenhum custo. O armazenamento persistente está localizado em seu diretório pessoal (\$HOME) e é privado para você. Ao contrário dos recursos de ambiente temporários que são reciclados após o término de cada sessão do shell, os dados do diretório inicial persistem entre as sessões.

Note

CloudShell VPCos ambientes não têm armazenamento persistente. O HOME diretório \$ é excluído quando seu VPC ambiente expira (após 20 a 30 minutos de inatividade) ou quando você exclui seu ambiente.

Se você parar de usar AWS CloudShell em um Região da AWS, os dados serão retidos no armazenamento persistente dessa região por 120 dias após o final de sua última sessão. Após 120 dias, a menos que você tome alguma medida, seus dados serão automaticamente excluídos do armazenamento persistente dessa região. Você pode evitar a remoção iniciando o AWS CloudShell novamente nessa Região da AWS. Para obter mais informações, consulte [Etapa 2: selecionar uma região AWS CloudShell, iniciar e escolher um shell](#).

Note**Cenário de uso**

Márcia AWS CloudShell costumava armazenar arquivos em seus diretórios pessoais em dois Regiões da AWS: Leste dos EUA (Norte da Virgínia) e Europa (Irlanda). Ela então começou a usar AWS CloudShell exclusivamente na Europa (Irlanda) e parou de lançar sessões de shell no Leste dos EUA (Norte da Virgínia).

Antes do prazo final para excluir dados no Leste dos EUA (Norte da Virgínia), Márcia decide impedir que seu diretório pessoal seja reciclado abrindo AWS CloudShell e selecionando novamente a região Leste dos EUA (Norte da Virgínia). Como ela usa continuamente a Europa (Irlanda) para sessões de shell, seu armazenamento persistente nessa região não é afetado.

Uso mensal

Cada um Região da AWS dos seus Conta da AWS tem uma cota de uso mensal para AWS CloudShell. Essa cota combina o tempo total gasto com o uso CloudShell de todos os IAM diretores daquela região. Se você tentar acessar CloudShell depois de atingir a cota mensal dessa região, uma mensagem será exibida explicando por que o ambiente de shell não pode ser iniciado.

Note

Se você precisar aumentar suas cotas de uso mensal, entre em contato com o [AWS Support](#) com as seguintes informações:

- CloudShell Região de uso
- Seu caso de uso. Por exemplo, AWS CLI operação e execução de comandos Linux
- O número de CloudShell usuários. Por exemplo, 5-10
- A estimativa máxima de tempo que você usa CloudShell na região
- CloudShellVPCuso do ambiente

Podemos aprovar o aumento da estimativa de tempo máximo para 1000 horas por mês em comparação com o limite existente de 200 horas.

Tamanho do comando

O tamanho do comando não pode exceder 65412 caracteres.

Note

Se você pretende executar o comando que excede 65412 caracteres, crie um script com a linguagem de sua escolha e execute-o na interface da linha de comando. Para obter mais informações sobre a variedade de softwares pré-instalados que podem ser acessados pela interface da linha de comando, consulte [Software pré-instalado](#).

Para ver um exemplo de como criar um script e executá-lo na interface da linha de comando, consulte [Tutorial: introdução ao AWS CloudShell](#).

Shells simultâneos

- Projéteis simultâneos: você pode executar até 10 projéteis ao mesmo tempo em cada um Região da AWS para sua conta.

Sessões de shell

- Sessões inativas: AWS CloudShell é um ambiente de shell interativo — se você não interagir com ele usando o teclado ou o ponteiro por 20 a 30 minutos, sua sessão de shell será encerrada. Os processos em execução não contam como interações.
- Sessões de longa duração: uma sessão de shell que é executada continuamente por aproximadamente 12 horas termina automaticamente, mesmo que o usuário esteja interagindo regularmente com ela durante esse período.

Acesso à rede e transferência de dados

As restrições a seguir se aplicam ao tráfego de entrada e saída do seu ambiente AWS CloudShell :

- Saída: você pode acessar a Internet pública.
- Entrada: você não pode acessar as portas de entrada. Nenhum endereço IP público está disponível.

⚠ Warning

Com o acesso à Internet pública, há o risco de que certos usuários possam exportar dados do AWS CloudShell ambiente. Recomendamos que IAM os administradores gerenciem a lista de permissões de AWS CloudShell usuários confiáveis por meio de IAM ferramentas. Para obter informações sobre como o acesso de usuários específicos pode ser explicitamente negado, consulte [Gerenciando ações permitidas no AWS CloudShell uso de políticas personalizadas](#).

Transferência de dados: o upload e o download de arquivos de e para lá AWS CloudShell podem ser lentos para arquivos grandes. Como alternativa, você pode transferir arquivos para o seu ambiente a partir de um bucket do Amazon S3 usando a interface de linha de comando do shell.

Restrições nos arquivos do sistema e nas páginas recarregadas

- Arquivos do sistema: se você modificar incorretamente os arquivos exigidos pelo ambiente computacional, poderá ter problemas ao acessar ou usar o AWS CloudShell ambiente. Se isso ocorrer, talvez seja necessário [excluir seu diretório inicial](#) para recuperar o acesso.
- Recarregar páginas: para recarregar a interface do AWS CloudShell , use o botão atualizar no seu navegador em vez da sequência de teclas de atalho padrão do seu sistema operacional.

Histórico de documentos para o Guia AWS CloudShell do usuário

Atualizações recentes

A tabela a seguir descreve as alterações importantes feitas no Guia do usuário do AWS CloudShell .

Alteração	Descrição	Data
VPCSuporte da Amazon AWS CloudShell em determinadas regiões	Foi adicionado suporte para criar e usar AWS CloudShell VPC ambientes em determinadas regiões.	13 de junho de 2024
Novos tutoriais foram adicionados ao Guia do usuário AWS CloudShell	Foram adicionados dois novos tutoriais que detalham como criar um contêiner Docker dentro AWS CloudShell e enviá-lo para um ECR repositório da Amazon e como implantar uma função Lambda via. AWS CDK	27 de dezembro de 2023
Contêineres Docker suportados AWS CloudShell em determinadas regiões	Support para contêineres Docker AWS CloudShell foi adicionado em determinadas regiões.	27 de dezembro de 2023
AWS CloudShell migrou para agora usar o Amazon Linux 2023 (AL2023)	AWS CloudShell agora usa AL2 023 e migrou do Amazon Linux 2.	4 de dezembro de 2023
Novas AWS regiões para AWS CloudShell	AWS CloudShell agora está disponível ao público em geral nas seguintes AWS regiões:	16 de junho de 2023

- Oeste dos EUA (N. da Califórnia)
- Africa (Cape Town)
- Ásia Pacífico (Hong Kong)
- Ásia-Pacífico (Osaka)
- Ásia-Pacífico (Seul)
- Ásia-Pacífico (Jacarta)
- Ásia-Pacífico (Singapura)
- Europa (Paris)
- Europa (Estocolmo)
- Europa (Milão)
- Oriente Médio (Barém)
- Oriente Médio (UAE)

[Lançamento AWS CloudShell no Console Toolbar](#)

Lançamento CloudShell no Console Toolbar, no canto inferior esquerdo do console, escolhendo CloudShell.

28 de março de 2023

[Novas AWS regiões para AWS CloudShell](#)

AWS CloudShell agora está disponível nas seguintes AWS regiões:

6 de outubro de 2022

- Canadá (Central)
- Europa (Londres)
- América do Sul (São Paulo)

[AWS CloudShell suportado nos EUA AWS GovCloud](#)

AWS CloudShell agora é suportado na região AWS GovCloud (EUA).

29 de junho de 2022

[Segurança FAQs](#)

Mais FAQs focado em questões de segurança.

14 de abril de 2022

Web Sockets	Seção adicionada aos requisitos de rede CloudShell que explica o uso do WebSocket protocolo.	21 de março de 2022
Solução de problemas com teclas de seta PowerShell	Siga as etapas para corrigir as teclas de seta que produzem letras incorretamente quando pressionadas.	7 de fevereiro de 2022
Preenchimento automático da tecla Tab	Nova documentação que explica como usar o bash-completion, que permite o preenchimento automático de comandos ou argumentos parcialmente digitados pressionando a tecla Tab.	24 de setembro de 2021
Especificando regiões AWS	Documentação sobre como especificar o padrão Região da AWS para AWS CLI comandos.	11 de maio de 2021
Formatação nas versões Kindle PDF e Kindle	Tamanhos de imagem e texto fixos nas células da tabela.	10 de março de 2021

[Versão de disponibilidade geral \(GA\) de AWS CloudShell em AWS regiões selecionadas](#)

AWS CloudShell agora está disponível ao público em geral nas seguintes AWS regiões:

15 de dezembro de 2020

- Leste dos EUA (Ohio)
- Leste dos EUA (N. da Virgínia)
- Oeste dos EUA (Oregon)
- Ásia-Pacífico (Tóquio)
- Europa (Irlanda)
- Ásia-Pacífico (Mumbai)
- Ásia-Pacífico (Sydney)
- Europa (Frankfurt)

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.