用户指南

AWS CloudShell

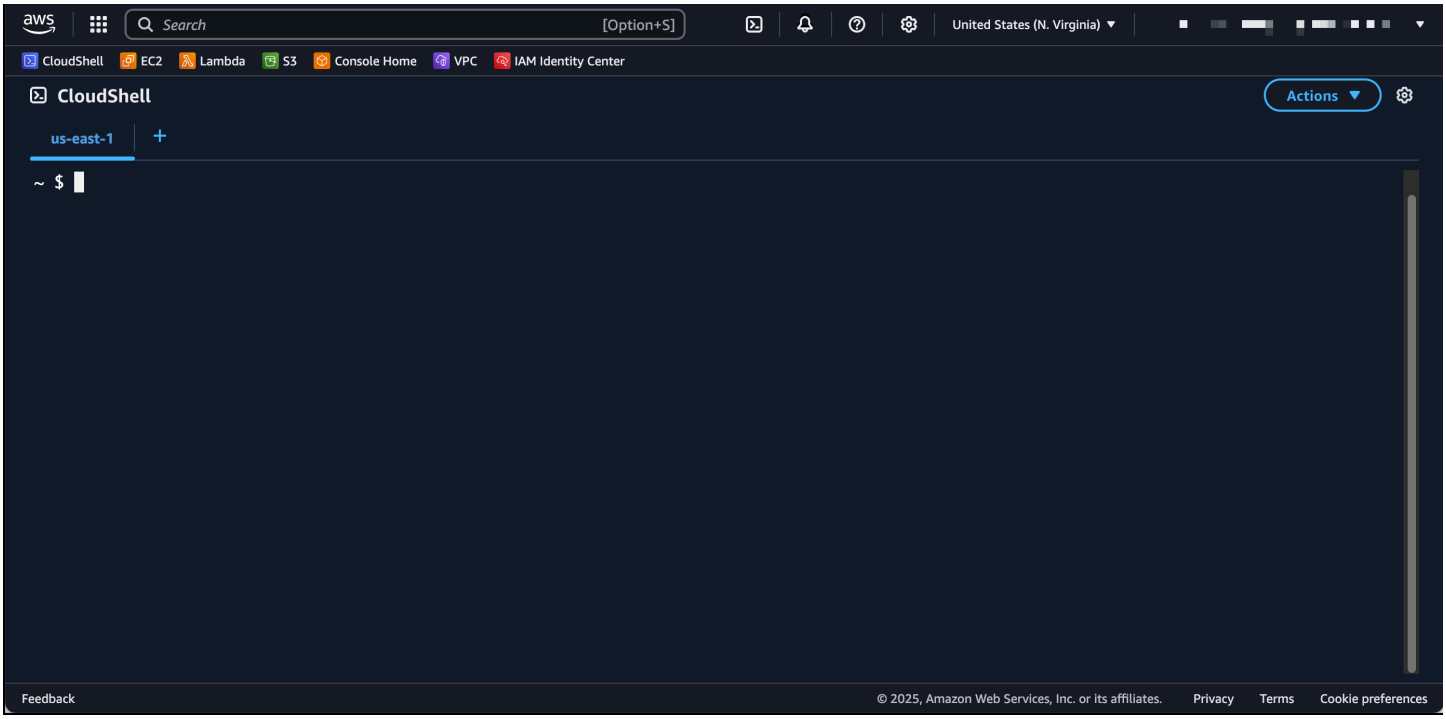
AWS CloudShell: 用户指南

Table of Contents

什么是 AWS CloudShell？

AWS CloudShell 是一个基于浏览器、经过预先验证的 shell，您可以直接从启动。 AWS 管理控制台 您可以通过 AWS 管理控制台 几种不同的方式导航到。 CloudShell 有关更多信息，请参阅入门 AWS CloudShell

您可以使用首选 shell 运行 AWS CLI 命令Bash，例如 PowerShell、或Z shell。而且您无需下载或安装命令行工具，即可完成此操作。



启动时 AWS CloudShell，将创建一个基于亚马逊 Linux 2023 的计算环境。在此环境中，您可以访问大量预安装的开发工具、用于上传和下载文件的选项，以及在会话之间保存的文件存储。你可以在最新版本的谷歌浏览器、Mozilla Firefox、Microsoft Edge 和苹果 Safari 浏览器中使用 CloudShell 。

（现在就试试吧：开始使用 AWS CloudShell）

AWS CloudShell的特征

AWS CloudShell 提供以下功能：

AWS Command Line Interface

您可以 AWS CloudShell 从启动 AWS 管理控制台。您用于登录控制台的 AWS 凭据将在新的 shell 会话中自动可用。由于 AWS CloudShell 用户已通过预先身份验证，因此在 AWS 服务 使用 AWS CLI 版本 2 进行交互时，您无需配置凭证。 AWS CLI 已预先安装在外壳的计算环境中。

有关 AWS 服务 使用命令行界面与进行交互的更多信息，请参阅在 CLI 中管理 AWS 服务 CloudShell。

Shell 和开发工具

使用为 AWS CloudShell 会话创建的 shell，您可以在首选的命令行 shell 之间无缝切换。更具体地说，您可以在Bash PowerShell、和之间切换Z shell。您还可以访问其他预安装工具和实用程序。其中包括 git、make、pip、sudo、tar、tmux、vim、wget 和 zip。

Shell 环境已预先配置为支持几种主要软件语言，例如 Node.js 和 Python。这意味着，例如，无需先执行运行时安装即可运行Node.js和Python项目。 PowerShell 用户可以使用.NET Core运行时。

有关更多信息，请参阅 AWS CloudShell 计算环境：规格和软件。

持久性存储

使用 AWS CloudShell，您可以免费使用每种 AWS 区域 存储空间中最多 1 GB 的永久存储空间。持久性存储位于您的主目录 ($HOME) 中，对您而言是私有的。与每个 Shell 会话结束后回收的临时环境资源不同的是，主目录中的数据会在不同会话之间保留。

有关在持久性存储中保留数据的更多信息，请参阅 持久性存储。

> **Note**
>
> CloudShell VPC 环境没有永久存储。当您的 VPC 环境超时（处于不活动状态 20-30 分钟后），或者当您删除或重启环境时，$HOME 目录将被删除。

CloudShell VPC 环境

AWS CloudShell 虚拟私有云 (VPC) 使您能够在 VPC 中创建 CloudShell 环境。对于每个 VPC 环境，您可以分配一个 VPC、添加子网以及关联一个或多个安全组。 AWS CloudShell 继承 VPC 的网络配置，使您能够与 VPC 中的其他资源在同一个子网中 AWS CloudShell 安全地使用。

安全性

AWS CloudShell 环境及其用户受到特定安全功能的保护。这包括 IAM 权限管理、Shell 会话限制和用于文本输入的安全粘贴等功能。

使用 IAM 进行权限管理

作为管理员，您可以使用 IAM 策略向 AWS CloudShell 用户授予和拒绝权限。您还可以创建策略来指定用户可以在 Shell 环境中执行的特定操作。有关更多信息，请参阅 使用 IAM 策略管理 AWS CloudShell 访问和使用情况。

Shell 会话管理

非活动和长时间运行的会话将自动停止和回收。有关更多信息，请参阅 Shell 会话。

用于文本输入的安全粘贴

默认情况下，“安全粘贴”处于启用状态。此安全功能要求您确认要粘贴到 Shell 中的多行文本不包含恶意脚本。有关更多信息，请参阅 对多行文本使用安全粘贴。

自定义选项

您可以根据自己的确切偏好自定义 AWS CloudShell 体验。例如，您可以更改屏幕布局（多个标签页）、显示的文本大小，以及在浅色和深色界面主题之间切换。有关更多信息，请参阅 自定义您的 AWS CloudShell 体验。

您还可以通过安装自己的软件和使用脚本修改 Shell 来扩展 Shell 环境。

会话恢复

会话恢复功能可以恢复您在 CloudShell 终端的单个或多个浏览器选项卡上运行的会话。如果您刷新或重新打开最近关闭的浏览器标签页，则此功能会恢复会话，直到 Shell 因会话处于非活动状态而停止。要继续使用您的 CloudShell 会话，请在终端窗口中按任意键。有关 Shell 会话的更多信息，请参阅 Shell 会话。

会话恢复还可以在每个终端标签页中恢复最新的终端输出和正在运行的进程。

> **Note**
>
> 会话恢复功能在移动应用程序中不可用。

的定价 AWS CloudShell

AWS CloudShell AWS 服务 是免费提供的。但是，您需要为运行的其他 AWS 资源付费 AWS CloudShell。此外，标准数据传输费用也适用。有关更多信息，请参阅AWS CloudShell 定价。

有关更多信息，请参阅 AWS CloudShell 的服务配额和限额。

关键 AWS CloudShell 话题

- 开始使用 AWS CloudShell
- AWS CloudShell 概念
- 在 CLI 中管理 AWS 服务 CloudShell
- 自定义您的 AWS CloudShell 体验
- AWS CloudShell 计算环境：规格和软件

开始使用 AWS CloudShell

本入门教程向您展示如何使用 Shell 命令行界面启动 AWS CloudShell 和执行关键任务。

首先，您登录 AWS 管理控制台 并选择一个 AWS 区域。然后，您可以在新的浏览器窗口中启动 CloudShell，并选择要使用的 Shell 类型。

接下来，在主目录中创建一个新文件夹，然后从本地计算机上传一个文件到该文件夹。在从命令行将该文件作为程序运行之前，您可以使用预安装的编辑器处理文件。最后，您可以调用 AWS CLI 命令以创建 Amazon S3 存储桶并将文件作为对象添加到该存储桶。

先决条件

IAM 权限

您可以通过将以下 AWS 托管策略附加到您的 IAM 身份（例如用户、角色或组）来获取 AWS CloudShell 的权限：

- AWSCloudShellFullAccess：为用户提供对 AWS CloudShell 及其功能的完全访问权限。

在本教程中，您还可以与 AWS 服务 交互。更具体地说，您可以通过创建 S3 存储桶并向该存储桶添加对象来与 Amazon S3 进行交互。您的 IAM 身份需要一个至少授予 `s3:CreateBucket` 和 `s3:PutObject` 权限的策略。

有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的 Amazon S3 操作。

练习文件

此练习还包括上传和编辑一个文件，然后在命令行界面中将该文件作为程序运行。在本地计算机上打开文本编辑器并添加以下代码段。

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

使用文件名 `add_prog.py` 保存该文件。

内容

步骤 1：登录到 AWS 管理控制台

此步骤包括输入您的 IAM 用户信息以访问 AWS 管理控制台。如果您已经在控制台中，请跳至步骤 2。

- 您可以使用 IAM 用户登录 URL 或前往主登录页面访问 AWS 管理控制台。

  IAM user sign-in URL

  - 打开浏览器并输入以下登录 URL。将 `account_alias_or_id` 替换为管理员提供的账户别名或账户 ID。

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

  - 输入您的 IAM 登录凭证，然后选择 登录。

  Main sign-in page

  - 打开 https://aws.amazon.com/console
  - 如果您之前没有使用此浏览器登录过，则会显示主登录页面。选择 IAM 用户，输入账户别名或账户 ID，然后选择下一步。
  - 如果您之前已经以 IAM 用户身份登录过，浏览器可能会记住 AWS 账户 的账户别名或账户 ID。如果是这一情况，输入您的 IAM 登录凭证，然后选择 登录。

> **Note**
>
> 您也可以以根用户身份登录。此身份具有对账户中所有 AWS 服务 和资源的完全访问权限。但是，我们强烈建议您不要使用根用户来执行日常任务，即使是管理任务。相反，请遵循仅使用根用户创建您的第一个 IAM 用户的最佳实践。

步骤 2：选择区域，启动 AWS CloudShell 并选择 Shell

在此步骤中，您将从控制台界面启动 CloudShell，选择可用的 AWS 区域，然后切换到您的首选 Shell，例如 Bash、PowerShell 或 Z shell。

1. 要选择将使用的 AWS 区域，请前往选择区域菜单，然后选择要使用的支持 AWS 区域。（可用区域突出显示。）

> **Important**
>
> 如果您切换区域，则界面会刷新，所选 AWS 区域 的名称将显示在命令行文本的上方。您添加到持久性存储空间的所有文件都仅在同一 AWS 区域 可用。如果您更改区域，则可以访问不同的存储空间和文件。

> **Important**
>
> 如果在控制台左下角的 Console Toolbar 上启动 CloudShell 时，CloudShell 在所选区域不可用，则默认区域将设置为最接近选定区域的区域。您可以运行该命令，该命令提供在与默认区域不同的区域中管理资源的权限。有关更多信息，请参阅使用 AWS 区域。
>
> Example
>
> 示例：
>
> 如果您选择欧洲（西班牙）eu-south-2，但 CloudShell 在欧洲（西班牙）eu-south-2 不可用，则默认区域设置为欧洲地区（爱尔兰）eu-west-1，该区域离欧洲（西班牙）eu-south-2 最近。

您将使用默认区域欧洲地区（爱尔兰）eu-west-1 的服务配额，并且将在所有地区恢复相同的 CloudShell 会话。默认区域可能会更改，您会在 CloudShell 浏览器窗口中收到通知。

2. 在 AWS 管理控制台 中，您可以选择以下选项之一来启动 CloudShell：

   1. 选择控制台导航栏中的 CloudShell 图标。
   2. 在搜索框中，键入“CloudShell”，然后选择 CloudShell。
   3. 在最新访问小部件中，选择 CloudShell。
   4. 在 Console Toolbar 中选择控制台左下角的 CloudShell。
      - 您可以通过拖动 = 来调整 CloudShell 会话的高度。
      - 您可以通过单击在新浏览器标签页中打开将您的 CloudShell 会话切换到全屏模式。

   当系统显示命令提示符时，表示 shell 已经准备就绪，可以进行交互。

   > **Note**
   >
   > 如果您遇到阻止成功启动 AWS CloudShell 或与之交互的问题，请检查信息以识别和解决故障排除 AWS CloudShell 中的这些问题。

3. 要选择要使用的预安装 Shell，请在命令行提示栏中输入以下程序名称之一：

   Bash

   ```
   bash
   ```

   如果切换到 Bash，则命令提示符处的符号将更新为 $。

   > **Note**
   >
   > Bash 是启动AWS CloudShell时正在运行的默认 Shell。

   PowerShell

   ```
   pwsh
   ```

如果切换到 PowerShell，则命令提示符处的符号将更新为 PS>。

Z shell

```
zsh
```

如果切换到 Z shell，则命令提示符处的符号将更新为 %。

有关 Shell 环境中预安装的版本的信息，请参阅 AWS CloudShell 计算环境一节中的 Shell 表格。

步骤 3：从 AWS CloudShell 下载文件

> **Note**
>
> 此选项不适用于 VPC 环境。

此步骤指导您完成下载文件的过程。

1. 要下载文件，请转到操作，然后从菜单中选择下载文件。

   将显示下载文件对话框。

2. 在下载文件对话框中，输入要下载的文件的路径。

   > **Note**
   >
   > 指定要下载的文件时，可以使用绝对路径或相对路径。对于相对路径名，默认情况下 `/home/cloudshell-user/` 会自动添加到开头。因此，要下载名为 `mydownload-file` 的文件，以下两个路径都是有效的路径：
   >
   > - 绝对路径：`/home/cloudshell-user/subfolder/mydownloadfile.txt`
   > - 相对路径：`subfolder/mydownloadfile.txt`

3. 选择下载。

   如果文件路径正确，则会显示一个对话框。您可以使用此对话框通过默认应用程序打开文件。或者，您可以将文件保存到本地计算机上的某个文件夹。

> **Note**
>
> 当您在 Console Toolbar 上启动 CloudShell 时，“下载” 选项不可用。您可以从 CloudShell 控制台或使用 Chrome 网络浏览器下载文件。

步骤 4：将文件上传到 AWS CloudShell

> **Note**
>
> 此选项不适用于 VPC 环境。

此步骤介绍如何上传文件，然后将其移至主目录中的新目录。

1. 要检查当前的工作目录，请在提示符下输入以下命令：

   ```
   pwd
   ```

   当您按 Enter 键时，Shell 会返回您当前的工作目录（例如，`/home/cloudshell-user`）。
2. 要将文件上传到该目录，请转至操作并从菜单中选择上传文件。

   将显示上传文件对话框。
3. 选择 Browse。
4. 在系统的文件上传对话框中，选择您为本教程（`add_prog.py`）创建的文本文件，然后选择打开。
5. 在 上传 对话框里，选择添加文件。

   进度条会跟踪上传情况。如果上传成功，则会显示一条消息，确认 `add_prog.py` 已添加到主目录的根目录中。
6. 要为文件创建目录，请输入 make directories 命令：`mkdir mysub_dir`。
7. 要将上传的文件从主目录的根目录移动到新目录，请使用 `mv` 命令：

   ```
   mv add_prog.py mysub_dir.
   ```
8. 要将工作目录更改为新目录，请输入 `cd mysub_dir`。

   命令提示符会更新以表明您已更改工作目录。

9. 要查看当前目录 mysub_dir 的内容，请输入 ls 命令。

   列出工作目录的内容。这包括您刚刚上传的文件。

步骤 5：从 AWS CloudShell 中移除文件

此步骤介绍如何从 AWS CloudShell 中移除文件。

1. 要从 AWS CloudShell 中移除文件，请使用标准 Shell 命令，例如 rm (删除)。

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

2. 要移除多个符合指定条件的文件，请运行 find 命令。

   以下示例删除了名称中包含后缀“.pdf”的所有文件。

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

> **Note**
>
> 假设您在特定 AWS 区域 中停止使用 AWS CloudShell。然后，该区域的持续性存储中的数据将在指定时间段后自动删除。更多信息，请参阅持久性存储。

步骤 6：创建主目录备份

此步骤介绍如何创建主目录备份。

1. 创建备份文件

   在主目录之外创建一个临时文件夹。

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

   您可以使用以下方法之一创建备份：

   a. 使用 tar 创建备份文件

      要使用 tar 创建备份文件，请输入以下命令：

```
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. 使用 zip 创建备份文件

要使用 zip 创建备份文件，请输入以下命令：

```
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. 将备份文件传输到 CloudShell 之外

您可以使用以下选项之一将备份文件传输到 CloudShell 之外：

a. 将备份文件下载到本地计算机上

您可以下载在先前步骤中创建的文件。有关如何从 CloudShell 下载文件的更多信息，请参阅从 AWS CloudShell 中下载文件。

在“下载文件”对话框中，输入要下载的文件的路径（例如，`/tmp/tmp.iA99tD9L98/home.tar.gz`）。

b. 将备份文件传输到 S3

要生成存储桶，请输入以下命令：

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

使用 AWS CLI 将文件复制到 S3 存储桶：

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

> **Note**
>
> 可能会收取数据传输费用。

3. 直接备份到 S3 存储桶

   要直接备份到 S3 存储桶，请输入以下命令：

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

步骤 7：重新启动 Shell 会话

此步骤介绍如何重新启动 shell 会话。

> **Note**
>
> 作为一项安全措施，如果您长时间不使用键盘或指针与 Shell 进行交互，则会话会自动停止。长时间运行的会话也会自动停止。有关更多信息，请参阅 Shell 会话。

1. 要重新启动 Shell 会话，请依次选择操作和重新启动。

   系统会通知您，重新启动 AWS CloudShell 会停止当前 AWS 区域 中的所有活动会话。

2. 要确认，请选择重新启动。

   界面会显示一条消息，指示 CloudShell 计算环境正在停止。环境停止并重新启动后，您可以开始在新会话中使用命令行。

> **Note**
>
> 在某些情况下，环境重新启动可能需要几分钟时间。

步骤 8：删除 Shell 会话主目录

此步骤介绍如何删除 shell 会话。

> **Note**
>
> 此选项不适用于 VPC 环境。重启 VPC 环境时，其主目录将被删除。

> **Warning**
>
> 删除主目录是一项不可逆的操作，即存储在主目录中的所有数据将被永久删除。但是，在以下情况下，您可以考虑使用此选项：
>
> - 您错误地修改了文件并且无法访问 AWS CloudShell 计算环境。删除您的主目录会将 AWS CloudShell 恢复其默认设置。
> - 您想立即从 AWS CloudShell 中删除所有数据。如果您停止在 AWS 区域中使用 AWS CloudShell，则持久性存储将在保留期结束时自动删除，除非您在该地区重新启动 AWS CloudShell。
>
> 如果您需要长期存储文件，请考虑使用 Amazon S3 等服务。

1. 要删除 Shell 会话，请选择操作、删除。

   系统会通知您，删除 AWS CloudShell 主目录会删除当前存储在您的 AWS CloudShell 环境中的所有数据。

   > **Note**
   >
   > 您不能撤消此操作。

2. 要确认删除操作，请在文本输入字段中输入位置名称，然后选择 删除。

   AWS CloudShell 将停止当前 AWS 区域中的所有活动会话。您可以创建一个新环境或设置 CloudShell VPC 环境。

3. 要创建新环境，请选择打开选项卡。
4. 要创建 CloudShell VPC 环境，请选择创建 VPC 环境。

手动退出 Shell 会话

使用命令行，您可以退出 Shell 会话并使用 `exit` 命令进行注销。然后，您可以按任意键重新连接并继续使用 AWS CloudShell。

步骤 9：编辑文件代码并使用命令行运行

此步骤演示如何使用预安装的 Vim 编辑器来处理文件。然后，您可以从命令行将该文件作为程序运行。

1. 要编辑您在上一步骤中上传的文件，请输入以下命令：

   ```
   vim add_prog.py
   ```

   Shell 界面将刷新以显示 Vim 编辑器。

2. 要在 Vim 中编辑文件，请按 I 键。现在编辑内容，让程序将三个数字而不是两个数字相加。

   ```
   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**
   >
   > 如果您将文本粘贴到编辑器中并启用了安全粘贴功能，则会显示一条警告。复制的多行文本可能包含恶意脚本。使用“安全粘贴”功能，您可以在粘贴之前验证完整的文本。如果您对文本安全感到满意，请选粘贴。

3. 编辑程序后，按下 Esc 进入 Vim 命令模式。然后，输入 `:wq` 命令保存文件，并退出编辑器。

   > **Note**
   >
   > 如果您不熟悉 Vim 命令模式，一开始可能会发现很难在命令模式和插入模式之间切换。保存文件和退出应用程序时使用命令模式。插入新文本时使用插入模式。要进入插入模式，

请按 I；要进入命令模式，请按 Esc。有关 Vim 及 AWS CloudShell 中提供的其他工具相关的更多信息，请参阅 开发工具和 Shell 实用程序。

4. 在主命令行界面上，运行以下程序并指定要输入的三个数字。语法如下所示。

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

命令行显示程序输出：`The sum is 15`。

步骤 10：使用 AWS CLI 将文件作为对象添加到Amazon S3 存储桶中

在此步骤中，您将创建一个 Amazon S3 存储桶，然后使用 PutObject 方法将代码文件作为对象添加到该存储桶中。

> **Note**
>
> 本教程说明如何使用 AWS CloudShell 中的 AWS CLI 与其他 Amazon Web Services 进行交互。使用此方法时，无需下载或安装任何其他资源。此外，由于已经在 Shell 中进行了身份验证，因此在进行调用之前无需配置凭证。

1. 要在指定 AWS 区域 创建存储桶，请输入以下命令：

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

> **Note**
>
> 如果您要在 `us-east-1` 区域之外创建存储桶，请使用 `LocationConstraint` 参数添加 `create-bucket-configuration` 以指定区域。以下是语法示例。
>
> ```
> $ aws s3api create-bucket --bucket my-bucket --region eu-west-1 --create-bucket-configuration LocationConstraint=eu-west-1
> ```

如果调用成功，命令行将显示来自服务的响应，输出与以下类似。

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

> **Note**
>
> 如果您不遵守 存储桶命名规则，则会显示以下错误：调用 createBucket 操作时出现错误 (invalidBucketName)：指定的存储桶无效。

2. 要上传文件并将该文件作为对象添加到您刚创建的存储桶，请调用 PutObject 方法。

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

将对象上传到 Amazon S3 存储桶后，命令行将显示来自服务的响应，类似于以下输出：

```
{"ETag": "\"ab123c1:w:wad4a567d8bfd9a1234ebeea56\""}
```

ETag 是存储的对象的哈希值。您可以使用此哈希值来 检查上传到 Amazon S3 的对象的完整性。

相关主题

- 在 CLI 中管理 AWS 服务 CloudShell
- 在本地计算机和 CloudShell 之间复制多个文件
- AWS CloudShell 概念
- 自定义您的 AWS CloudShell 体验

AWS CloudShell 教程

以下教程说明在使用 AWS CloudShell 时如何进行实验以及测试不同的功能和集成。

教程概述	了解更多
复制多个文件	the section called “教程：复制多个文件”
创建预签名 URL	???
在 AWS CloudShell 中构建 Docker 容器并将其推送到 Amazon ECR	???
使用 AWS CDK 部署 Lambda 函数	???

在本地计算机和 CloudShell 之间复制多个文件

本教程介绍如何在本地计算机和 CloudShell 之间复制多个文件。

使用 AWS CloudShell 界面，您可以在本地计算机和 Shell 环境之间一次上传或下载单个文件。要同时在 CloudShell 和本地计算机之间复制多个文件，请使用以下任意一种选项：

- Amazon S3：在本地计算机和 CloudShell 之间复制文件时，使用 S3 存储桶作为中间存储位置。
- Zip 文件：将多个文件压缩到一个压缩文件夹中，可以使用 CloudShell 界面上传或下载文件夹。

> **Note**
>
> 由于 CloudShell 不允许入站互联网通信，因此目前无法使用诸如 `scp` 或 `rsync` 之类的命令在本地计算机和 CloudShell 计算环境之间复制多个文件。

使用 Amazon S3 上传和下载多个文件

此步骤介绍如何使用 Amazon S3 上传和下载多个文件。

先决条件

要使用存储桶和对象，您需要一个 IAM policy 来授予执行以下 Amazon S3 API 操作的权限：

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

有关 Amazon S3 操作的完整列表，请参阅《Amazon Simple Storage Service API 参考》中操作部分。

使用 Amazon S3 将多个文件上传到 AWS CloudShell

此步骤介绍如何使用 Amazon S3 上传多个文件。

1. 在 AWS CloudShell 中，通过运行以下 `s3` 命令创建 S3 存储桶：

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

   如果调用成功，命令行将显示来自 S3 服务的响应：

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

2. 将目录中的文件从本地计算机上传到存储桶。请选择以下选项之一来上传文件：

   - AWS 管理控制台：使用拖放功能将文件夹和文件上传到 S3 存储桶
   - AWS CLI：在本地计算机上安装该版本版本的工具后，使用命令行将文件和文件夹上传到存储桶。

   Using the console

   - 通过 https://s3.console.aws.amazon.com/s3/ 打开 Amazon S3 控制台。

     （如果您正在使用 AWS CloudShell，则应该已经登录到控制台。）

- 在存储桶列表中，请选择要将文件夹和文件上传到其中的存储桶的名称。您也可以通过选择创建存储桶来创建自己选择的存储桶。
- 要将文件上传到文件夹，请选择要上传的文件，然后选择 上传。然后，将您选择的文件和文件夹拖放到列出目标存储桶中的对象的控制台窗口，或者选择添加文件或添加文件夹。

  上传页面上将列出所选文件。
- 选中复选框以指示要添加的文件。
- 要将所选文件添加到存储桶，请选择上传。

> **Note**
>
> 有关更多信息，请参阅《Amazon Simple Storage Service 控制台用户指南》中的如何将文件和文件夹上传到 S3 存储桶。

Using AWS CLI

> **Note**
>
> 对于此选项，您需要在本地计算机上安装 AWS CLI 工具，并配置用于调用 AWS 服务的凭证。有关更多信息，请参阅《AWS Command Line Interface 用户指南》。

- 启动 AWS CLI 工具并运行以下 `aws s3` 命令，将指定的存储桶与本地计算机上当前目录的内容同步：

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

如果同步成功，则会显示添加到存储桶的每个对象的上传消息。

3. 返回 CloudShell 命令行并输入以下命令，将 Shell 环境中的目录与 S3 存储桶的内容同步：

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

> **Note**
>
> 要执行模式匹配以排除或包含特定对象，您可以在 sync 命令中使用 `--exclude "<value>"` 和 `--include "<value>"` 参数。
> 有关更多信息，请参阅《AWS CLI 命令参考》中的使用 Exclude 和 Include 筛选条件。

如果同步成功，则会显示从存储桶下载到目录的每个文件的下载消息。

> **Note**
>
> 应用同步命令时，仅以递归方式将源目录中的新文件和更新过的文件复制到目标位置。

使用 Amazon S3 从 AWS CloudShell 下载多个文件

此步骤介绍如何使用 Amazon S3 下载多个文件。

1. 使用 AWS CloudShell 命令行，输入以下 `aws s3` 命令，将 S3 存储桶与 Shell 环境中当前目录的内容同步：

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

> **Note**
>
> 您还可以将 `--exclude "<value>"` 和 `--include "<value>"` 参数添加至 sync 命令来执行模式匹配以排除或包含特定文件或对象。
> 有关更多信息，请参阅《AWS CLI 命令参考》中的使用 Exclude 和 Include 筛选条件。

如果同步成功，则会显示添加到存储桶的每个对象的上传消息。

2. 将桶内容下载到本地计算机上。由于 Amazon S3 控制台不支持下载多个对象，因此您需要使用安装在本地计算机上的 AWS CLI 工具。

要通过 AWS CLI 工具的命令行下载，请运行以下命令。

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

如果同步成功，命令行将显示在目标目录中更新或添加的每个文件的下载消息。

> **Note**
>
> 对于此选项，您需要在本地计算机上安装 AWS CLI 工具，并配置用于调用 AWS 服务的凭证。有关更多信息，请参阅《AWS Command Line Interface 用户指南》。

使用压缩文件夹上传和下载多个文件

此步骤介绍如何使用压缩文件夹上传和下载多个文件。

使用 zip/unzip 实用程序，您可以压缩存档中的多个文件，这些文件可以视为单个文件。该实用程序已预先安装在 CloudShell 计算环境中。

有关预安装工具的更多信息，请参阅 开发工具和 Shell 实用程序。

使用压缩文件夹将多个文件上传到 AWS CloudShell

此步骤介绍如何使用压缩文件夹上传多个文件。

1. 在本地计算机上，将要上传的文件添加到压缩文件夹中。
2. 启动 CloudShell，然后选择操作、上传文件。
3. 在上传文件对话框中，选择选择文件，然后选择您刚创建的压缩文件夹。
4. 在上传文件对话框中，选择上传，将所选文件添加到 Shell 环境中。
5. 在 CloudShell 命令行中，运行以下命令将 zip 档案的内容解压缩到指定目录：

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

使用压缩文件夹从 AWS CloudShell 下载多个文件

此步骤介绍如何使用压缩文件夹下载多个文件。

1. 在 CloudShell 命令行中，运行以下命令，将当前目录中的所有文件添加到压缩文件夹中：

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

2. 选择操作、下载文件。

3. 在下载文件对话框中，输入压缩文件夹的路径（例如 `/home/cloudshell-user/zip-folder/zipped-archive.zip`），然后选择下载。

   如果路径正确，浏览器对话框将提供打开压缩文件夹或将其保存到本地计算机的选项。
4. 现在，您可以在本地计算机上解压缩下载的压缩文件夹中的内容。

使用 CloudShell 为 Amazon S3 对象创建预签名 URL

本教程向您展示如何创建预签名 URL 以便与他人共享 Amazon S3 对象。由于对象所有者在共享时会指定自己的安全凭证，因此任何收到预签名 URL 的人都可以在有限的时间内访问该对象。

先决条件

- 拥有 AWSCloudShellFullAccess 策略提供的访问权限的 IAM 用户。
- 有关创建预签名 URL 所需的 IAM 权限，请参阅《Amazon Simple Storage Service 用户指南》中的与其他用户共享对象。

步骤 1：创建 IAM 角色以授予对 Amazon S3 存储桶的访问权限

此步骤介绍如何创建 IAM 角色以授予对 Amazon S3 存储桶的访问权限。

1. 要获取可以共享的 IAM 详细信息，请从 AWS CloudShell 中调用 `get-caller-identity` 命令。

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

   如果调用成功，命令行将显示类似如下内容的响应：

   ```
   {
       "Account": "123456789012",
       "UserId": "AROAXXOZUUOTTWDCVIDZ2:redirect_session",
       "Arn": "arn:aws:sts::531421766567:assumed-role/Feder08/redirect_session"
   }
   ```
2. 获取您在上一步中获得的用户信息，并将其添加到 CloudFormation 模板中。此模板创建一个 IAM 角色。该角色向您的合作者授予对共享资源的最低权限。

   ```
   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. 将 CloudFormation 模板保存在名为 `template.yaml` 的文件中。
4. 使用模板部署堆栈并通过调用 `deploy` 命令创建 IAM 角色。

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

生成预签名 URL

此步骤介绍如何生成预签名 URL。

1. 在 AWS CloudShell 中使用编辑器，添加以下代码。此代码创建了一个 URL，为联合用户提供了直接访问 AWS 管理控制台 的权限。

```
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. 将代码保存在名为 share.py 的文件中。
3. 从命令行运行以下命令，以从 CloudFormation 中检索 IAM 角色的 Amazon 资源名称（ARN）。然后，在 Python 脚本中使用它来获取临时安全凭证。

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

该脚本返回一个 URL，合作者可以点击它，以进入到 AWS 管理控制台 中的 AWS CloudShell。在接下来的 3600 秒（1 小时）内，合作者可以完全控制在 Amazon S3 存储桶中的 myfolder/ 文件夹。凭证在 1 小时后到期。在此时间之后，合作者将无法再访问该存储桶。

在里面构建 Docker 容器 CloudShell并将其推送到 Amazon ECR 存储库

本教程向您展示如何定义和构建 Docker 容器 AWS CloudShell 并将其推送到 Amazon ECR 存储库。

先决条件

- 您必须拥有所需权限才能创建 Docker 容器并将其推送到 Amazon ECR 存储库。有关更多信息，请参阅《Amazon ECR 用户指南》中的 Amazon ECR 私有存储库。有关使用 Amazon ECR 推送映像所需权限的更多信息，请参阅《Amazon ECR 用户指南》中的推送映像所需的 IAM 权限。

教程

以下教程概述了如何使用该 CloudShell 接口构建 Docker 容器并将其推送到 Amazon ECR 存储库。

1. 在您的主目录中创建一个新文件夹。

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

2. 导航到所创建的文件夹。

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

3. 创建空 Dockerfile。

```
touch Dockerfile
```

4. 使用文本编辑器（如 `nano Dockerfile`）打开该文件并将以下内容粘贴到该文件中。

```
# 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. Dockerfile 现在已准备就绪，可用于构建。通过运行 `docker build` 构建容器。用 easy-to-type 名称标记容器，以便在 future 命令中使用。

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

请务必包含尾部句点（`.`）。
在 AWS CloudShell中运行的 Docker 构建命令的图片。

6. 现在，您可以测试该容器以检查其能否在 AWS CloudShell中正常运行。

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

里面的 docker 容器运行命令的图片 AWS CloudShell

7. 现在您已拥有正常运行的 Docker 容器，接下来需要将其推送到 Amazon ECR 存储库。如果您已有 Amazon ECR 存储库，则可以跳过此步骤。

运行以下命令，为本教程创建一个 Amazon ECR 存储库。

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

用于在内部创建 Amazon ECR 存储库的命令的图像 AWS CloudShell

8. 创建 Amazon ECR 存储库后，可以将 Docker 容器推送到该存储库。

   运行以下命令，获取 Docker 的 Amazon ECR 登录凭证。

```
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}
```

   用于获取 Docker 的 Amazon ECR 登录凭证的命令的图片。

> **Note**
>
> 如果未在您的中设置AWS_REGION环境变量， CloudShell 或者您想与其他环境中的资源进行交互 AWS 区域，请运行以下命令：
>
> ```
> AWS_REGION=<your-desired-region>
> ```

9. 使用目标 Amazon ECR 存储库标记映像，然后将其推送到该存储库。

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

   用于将映像标记为目标 Amazon ECR 存储库的命令的图片。

   如果您在尝试完成本教程时遇到错误或问题，请参阅本指南的故障排除部分寻求帮助。

清理

现在您已成功将 Docker 容器部署到 Amazon ECR 存储库。要从您的 AWS CloudShell 环境中删除您在本教程中创建的文件，请运行以下命令。

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

- 删除 Amazon ECR 存储库。

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

使用中的部署 Lambda 函数 AWS CDK CloudShell

本教程向您展示如何使用中的将 Lambda 函数部署到您的账户。 AWS Cloud Development Kit (AWS CDK) CloudShell

先决条件

- 引导您的帐户使用 AWS CDK。有关使用引导的信息 AWS CDK，请参阅 v2 开发者指南中的AWS CDK 引导。如果你还没有引导账户，你可以跑`cdk bootstrap`进去。 CloudShell

- 确保您拥有向您的账户部署资源的权限。建议使用管理员权限。

教程

以下教程概述了如何使用中的部署基于 Docker 容器的 Lambda 函数。 AWS CDK CloudShell

1. 在您的主目录中创建一个新文件夹。

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

2. 导航到所创建的文件夹。

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

3. 在本地安装 AWS CDK 依赖项。

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

   用于安装 AWS CDK 依赖项的命令的图像。

4. 在您创建的文件夹中创建一个基本 AWS CDK 项目。

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

5. 使用文本编辑器（如 `nano cdk.json`）打开该文件并将以下内容粘贴到该文件中。

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

```
}
```

6. 打开 lib/docker-tutorial.js 文件并将以下内容粘贴到该文件中。

```
// 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. 打开 lib/Dockerfile 并将以下内容粘贴到其中。

```
# 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. 打开 lib/hello.js 文件并将以下内容粘贴到该文件中。

```
// 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. 使用 C AWS CDK LI 合成项目并部署资源。您必须引导您的账户。

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

使用 AWS CDK CLI 合成项目和部署资源的命令的图像。

10. 调用 Lambda 函数进行确认和验证。

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

用于调用 Lambda 函数的命令的图片。

现在，您已经使用 AWS CDK成功部署了基于 Docker 容器的 Lambda 函数。有关的更多信息 AWS CDK，请参阅 AWS CDK v2 开发者指南。如果您在尝试完成本教程时遇到错误或问题，请参阅本指南的故障排除部分寻求帮助。

清理

现在，您已经使用 AWS CDK成功部署了基于 Docker 容器的 Lambda 函数。在 AWS CDK 项目内部，运行以下命令以删除关联的资源。系统将提示您确认删除。

-
```
npx cdk destroy DockerTutorialStack
```

- 要将您在本教程中创建的文件和资源从您的 AWS CloudShell 环境中删除，请运行以下命令。

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

AWS CloudShell 概念

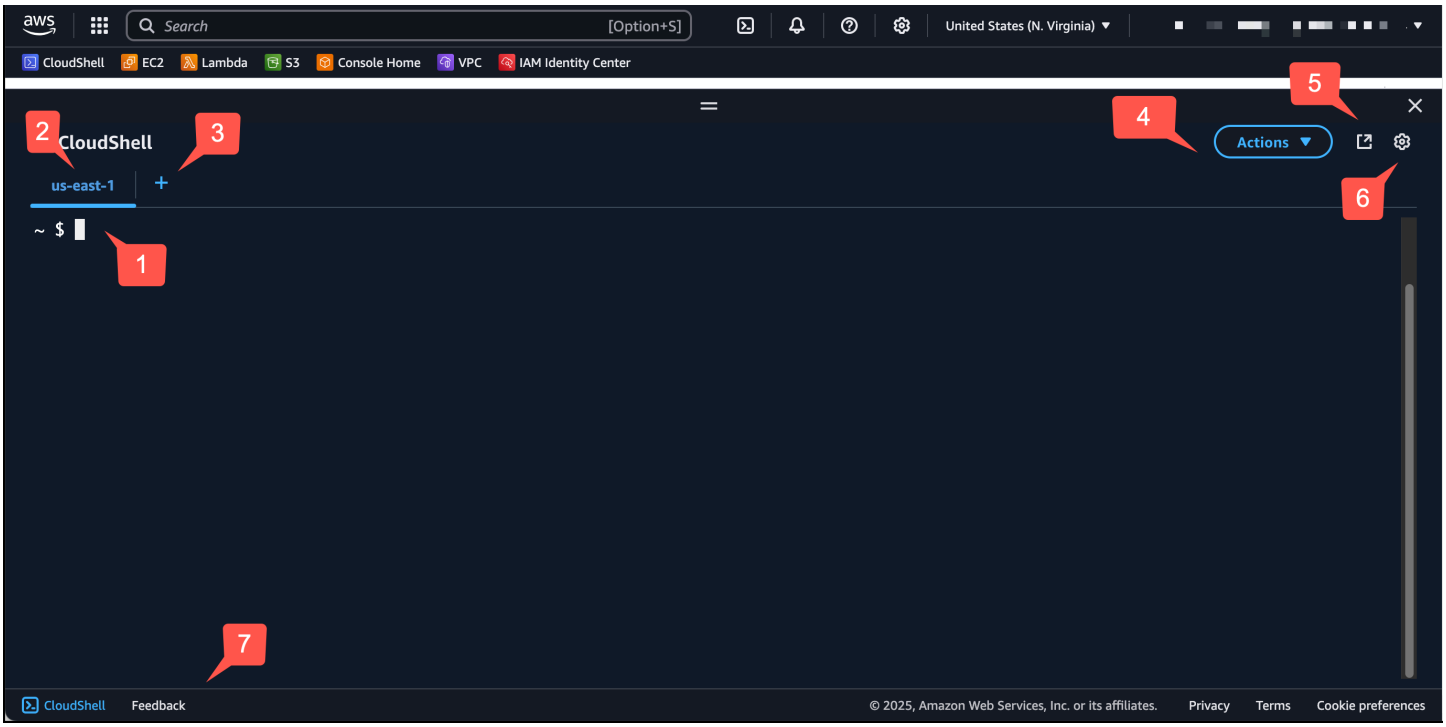
本节介绍如何与支持的应用程序交互 AWS CloudShell 并执行特定操作。

主题

浏览界面 AWS CloudShell

您可以从 AWS 管理控制台 和中浏览 CloudShell 界面功能Console Toolbar。

以下屏幕截图显示了几个关键的 AWS CloudShell 界面功能。



1. AWS CloudShell 命令行界面，用于使用首选 shell 运行命令。当前 Shell 类型由命令提示符指示。
2. 终端选项卡，它使用当前 AWS CloudShell 的运行 AWS 区域 位置。
3. + 图标是一个下拉菜单，包括创建、重启和删除环境的选项。

4. 操作菜单，提供更改屏幕布局、下载和上传文件、重启 AWS CloudShell和删除 AWS CloudShell 主目录的选项。

> **Note**
>
> 当您在 CloudShell 上启动时，“下载” 选项不可用Console Toolbar。

5. 在新浏览器中打开选项卡，提供全屏访问 CloudShell 会话的选项。
6. 首选项选项，可用于自定义 Shell 体验。
7. 底部栏提供以下选项：
   - CloudShell 从CloudShell图标启动。
   - 通过反馈图标提供反馈。选择您要提交的反馈类型，添加您的评论，然后选择提交。
     - 要提交反馈 CloudShell，请选择以下选项之一：
       - 从控制台启动 CloudShell并选择 “反馈”。添加您的评论，然后选择提交。
       - 选择CloudShell控制台左下角的Console Toolbar，然后选择 “在新浏览器选项卡中打开” 图标 “反馈”。添加您的评论，然后选择提交。

     > **Note**
     >
     > 当您在 CloudShell 上启动时，“反馈” 选项不可用Console Toolbar。
   - 了解我们的隐私政策和使用条款，并自定义 Cookie 首选项。

在 AWS 区域

您 AWS 区域 正在运行的当前显示为选项卡。

您可以通过使用区域选择器选择特定区域来选择 AWS 区域 要处理的区域。更改区域后，当您的 Shell 会话连接到在选定区域中运行的不同计算环境时，界面会刷新。

> **Important**
>
> - 每个存储空间最多可使用 1 GB 的永久存储空间 AWS 区域。持久性存储保存在您的主目录 (`$HOME`) 中。这意味着存储在您的主目录中的任何个人文件、目录、程序或脚本都位于同一个 AWS 区域中。此外，它们与位于主目录中但存储在不同区域中的那些文件不同。

文件在持久性存储中的长期保留也是按区域管理的。有关更多信息，请参阅 持久性存储。

- 永久存储不适用于 AWS CloudShell VPC 环境。

指定您的默认 AWS 区域 值 AWS CLI

您可以使用环境变量来指定访问 AWS 服务 所需的配置选项和凭据 AWS CLI。当您从中的特定区域启动时，或者在区域选择器 AWS CloudShell 中选择一个选项时，将在中设置 AWS 区域 用于指定 shell 会话默认值的环境变量。 AWS 管理控制台

环境变量优先于由更新的 AWS CLI 凭据文件`aws configure`。因此，您无法运行 `aws configure` 命令来更改由环境变量指定的区域。相反，要更改 AWS CLI 命令的默认区域，请为`AWS_REGION`环境变量赋值。在以下示例中，将 `us-east-1` 替换为您所在的区域。

Bash or Zsh

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

设置环境变量会更改使用的值，直到 Shell 会话结束或当您将该变量设置为其他值时。通过在 Shell 的启动脚本中设置变量，可使变量在未来的会话中继续有效。

PowerShell

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

如果在 PowerShell 提示符处设置环境变量，则环境变量仅在当前会话的持续时间内保存该值。或者，您可以通过将变量添加到您的 PowerShell 个人资料中来为所有未来 PowerShell 会话设置该变量。有关存储环境变量的更多信息，请参阅PowerShell 文档。

要确认您已更改默认区域，请运行`aws configure list`命令以显示当前的 AWS CLI 配置数据。

> **Note**
>
> 对于特定 AWS CLI 命令，您可以使用命令行选项覆盖默认区域`--region`。有关更多信息，请参阅 AWS Command Line Interface 用户指南中的 Command line options。

处理文件和存储

使用 AWS CloudShell的界面，您可以将文件上传到 shell 环境，也可以从 shell 环境下载文件。有关下载和上传文件的更多信息，请参阅入门 AWS CloudShell。

为了确保您添加的任何文件在会话结束后仍然可用，您应该知道持久性存储和临时存储之间的区别。

- 永久存储：每种存储空间都有 1 GB AWS 区域。持久性存储在您的主目录中。
- 临时存储：临时存储空间在会话结束时被回收。临时存储位于主目录之外的目录中。

> **Important**
>
> 确保将要保留并用于将来的 Shell 会话的文件保存在您的主目录中。例如，假设您通过运行 mv 命令将某个文件移出主目录。然后，当前 Shell 会话结束时，该文件将被回收。

CloudShell 在 Console Mobile Application 中访问

您可以 AWS Console Mobile Application 从主屏幕 CloudShell 中访问。在主屏幕上，您可以查看有关 CloudShell 和其他 AWS 服务的信息。有关更多信息，请参阅 AWS Console Mobile Application入门。要 CloudShell 在中启动 AWS Console Mobile Application，请选择以下选项之一：

- 选择导航栏底部的 CloudShell 图标。
- 在 “服务” 菜单CloudShell上选择。

您可以 CloudShell 随时选择 X 退出。

有关在 Console Mobile Application CloudShell 中进行访问的更多信息，请参阅访问 AWS CloudShell。

> **Note**
>
> 目前，您无法在 AWS Console Mobile Application中创建或启动 VPC 环境。

使用 Docker

AWS CloudShell 无需安装或配置即可完全支持 Docker。你可以在里面 AWS CloudShell定义、构建和运行 Docker 容器。您可以通过 AWS CDK 工具包部署基于 Docker 的资源，例如基于 Docker 容器的 Lambda 函数，也可以构建 Docker 容器并通过 Docker CLI 将其推送到 Amazon ECR 存储库。有关如何运行这两个部署的详细步骤，请参阅以下教程：

- 教程：使用部署 Lambda 函数 AWS CDK
- 教程：在里面构建 Docker 容器 AWS CloudShell 并将其推送到 Amazon ECR 存储库

通过 AWS CloudShell使用 Docker 有某些限制：

- Docker 在环境中的空间有限。如果您的单个映像较大，或者预先存在的 Docker 映像过多，则可能会导致无法拉取、构建或运行其他映像。有关 Docker 的更多信息，请参阅 Docker 文档指南。
- 除了 AWS GovCloud （美国） AWS 区域外，Docker 在所有区域都可用。有关可用 Docker 的区域列表，请参阅支持的 AWS 区域。 AWS CloudShell
- 如果您在使用 Docker 时遇到问题 AWS CloudShell，请参阅本指南的 “疑难解答” 部分，了解如何解决这些问题。

AWS CloudShell 的辅助功能

本主题介绍如何使用 CloudShell 的辅助功能。您可以使用键盘浏览页面上的可聚焦元素。您还可以自定义 CloudShell 的外观，包括字体大小和界面主题。

CloudShell 中的键盘导航

要浏览页面上的可聚焦元素，请按 Tab。

CloudShell 终端辅助功能

您可以通过以下方式使用 Tab 键：

- 终端模式（默认）——在此模式下，终端会捕获您的 Tab 密钥输入。将焦点放在终端上后，按下 Tab 即可仅访问终端的功能。
- 导航模式——在此模式下，终端不会捕获您的 Tab 密钥输入。按下 Tab 可浏览页面上的可聚焦元素。

要在终端模式和导航模式之间切换，请按 Ctrl+M。切换回去后，标题中会出现 Tab：导航，您可以使用 Tab 键在页面中导航。

要返回终端模式，请按 Ctrl+M。或者，选择 Tab：导航旁边的 X。

> **Note**
>
> 目前，CloudShell 终端的辅助功能在移动设备上不可用。

在 CloudShell 中选择字体大小和界面主题

您可以自定义 CloudShell 的外观以适应您的视觉偏好。

- 字体大小——在终端中从最小、小、中、大和最大字体大小中进行选择。有关更改字体大小的更多信息，请参阅 the section called “更改字体大小”。
- 主题——在浅色和深色界面主题之间进行选择。有关更改界面主题的更多信息，请参阅 the section called “更改界面主题”。

在 CLI 中管理 AWS 服务 CloudShell

的一个主要好处 AWS CloudShell 是，您可以使用它从命令行界面管理您的 AWS 服务。这意味着您不需要事先下载和安装工具或在本地配置您的凭证。启动时 AWS CloudShell，将创建一个已安装以下 AWS 命令行工具的计算环境：

- AWS CLI
- AWS Elastic Beanstalk CLI
- Amazon ECS CLI
- AWS SAM

而且，由于您已经登录 AWS，因此无需在使用服务之前在本地配置证书。您用于登录 AWS 管理控制台 的凭证会被转发到 AWS CloudShell。

如果要更改使用的默认 AWS 区域 AWS CLI，则可以更改分配给`AWS_REGION`环境变量的值。（有关更多信息，请参阅 指定您的默认 AWS 区域 值 AWS CLI。）

本主题的其余部分演示如何开始使用 AWS CloudShell 命令行与所选 AWS 服务进行交互。

AWS CLI 所选 AWS 服务的命令行示例

以下示例仅代表使用 AWS CLI 版本 2 中提供的命令可以使用的众多 AWS 服务中的一部分。有关完整列表，请参阅 AWS CLI 命令参考。

- DynamoDB
- Amazon EC2
- Amazon Glacier

DynamoDB

DynamoDB 是一项完全托管式 NoSQL 数据库服务，提供快速且可预测的性能，能够实现无缝扩展。该服务的 NoSQL 模式实现支持键值和文档数据结构。

以下`create-table`命令创建在您的账户中命名的 NoSQL 样式表`MusicCollection`。 AWS

```
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
```

有关更多信息，请参阅 AWS Command Line Interface 用户指南中的通过 AWS CLI使用 DynamoDB。

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) 是一项网络服务，可在云中提供安全且可调整大小的计算容量。该服务旨在降低网络规模级云计算的难度，并且更易访问。

以下 `run-instances` 命令在指定 VPC 子网中启动了一个 t2.micro 实例：

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

有关更多信息，请参阅《AWS Command Line Interface 用户指南》 AWS CLI中的 “ EC2使用 Amazon”。

Amazon Glacier

Amazon Glacier 和 Amazon Glacier Deep Archive 是一种安全、持久且成本极低的 Amazon S3 云存储类别，用于数据存档和长期备份。

以下 `create-vault` 命令创建一个文件库，即用于存储档案的容器：

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

有关更多信息，请参阅《AWS Command Line Interface 用户指南》中的通过 AWS CLI使用 Amazon Glacier。

AWS Elastic Beanstalk CLI

AWS Elastic Beanstalk CLI 提供了一个命令行界面，用于简化从本地存储库创建、更新和监控环境的过程。在这种情况下，环境是指运行应用程序版本的 AWS 资源集合。

以下 `create` 命令在自定义 Amazon Virtual Private Cloud (VPC) 中创建新环境。

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

有关更多信息，请参阅AWS Elastic Beanstalk 开发人员指南中的EB CLI 指令参考。

Amazon ECS CLI

Amazon Elastic Container Service (Amazon ECS) 命令行界面 (CLI) 提供多个高级命令。这些命令旨在简化从本地开发环境创建、更新和监视集群和任务的过程。（ Amazon ECS 集群是任务或服务的逻辑分组。 ）

以下 `configure` 命令将 Amazon ECS CLI 配置为创建名为 `ecs-cli-demo` 的集群配置。此集群配置使用 `FARGATE` 作为 `us-east-1 region` 中 `ecs-cli-demo` 集群的默认启动类型。

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

有关更多信息，请参阅《Amazon Elastic Container Service 开发人员指南》中的 Amazon ECS 命令行参考。

AWS SAM CLI

AWS SAM CLI 是一种在 AWS Serverless Application Model 模板和应用程序代码上运行的命令行工具。您可以使用它执行多项任务。其中包括在本地调用 Lambda 函数、为无服务器应用程序创建部署包以及将您的无服务器应用程序部署到云端。 AWS

以下 `init` 命令初始化一个新的 SAM 项目，并将所需参数作为参数传递：

```
sam init --runtime python3.9 --dependency-manager pip --app-template hello-world --name
 sam-app
```

有关更多信息，请参阅《AWS Serverless Application Model 开发人员指南》中的AWS SAM CLI 命令参考。

在 CloudShell 中使用 Amazon Q CLI

> **Important**
>
> 由于内部问题，AWS CloudShell 已暂时禁用 Amazon Q 聊天功能。我们正在积极调查，并将尽快恢复此功能。同时，您可以在 AWS 管理控制台中继续使用 Q 聊天。

Amazon Q CLI 是一个命令行界面，允许您与 Amazon Q 进行交互。有关更多信息，请参阅《Amazon Q 开发者版用户指南》中的在命令行上使用 Amazon Q 开发者版。

使用 CloudShell 中的 Amazon Q CLI，您可以通过终端在自然语言对话中进行互动、提问和接收 Amazon Q 的回复。您可以获得相关的 shell 命令，从而减少在终端中键入时搜索内容、记住语法和接收命令建议的需求。

> **Note**
>
> 目前，CloudShell 中的 Amazon Q CLI 功能在您的 CloudShell VPC 环境中不可用。

如果您在 CloudShell 中看不到 Amazon Q CLI 功能，请联系管理员，为您提供 IAM 权限。有关更多信息，请参阅《Amazon Q 开发者版用户指南》中的适用于 Amazon Q 开发者版的基于身份的策略示例。

本章介绍如何在 CloudShell 中使用 Amazon Q CLI 功能。

在 CloudShell 中使用 Amazon Q 内嵌建议

当您在终端中键入内容时，CloudShell 中的 Amazon Q 内嵌建议会为您提供命令建议。有关更多信息，请参阅《Amazon Q 开发者版用户指南》中的在命令行中使用 Amazon Q 内嵌建议。

在 CloudShell 中使用 Amazon Q 内嵌建议

1. 从 AWS 管理控制台中，选择 CloudShell。
2. 在 CloudShell 终端上，切换到 Z shell，然后开始键入。要切换到 Z shell，请在终端中键入 `zsh`，然后按 Enter。

> **Note**
>
> 目前，只有 Z shell 支持 Amazon Q 内嵌建议。

当您开始键入命令时，Amazon Q 会根据您当前的输入和之前的命令提出建议。内嵌建议会自动启用。

要禁用内嵌建议，请运行以下命令：

```
q inline disable
```

要启用内嵌建议，请运行以下命令：

```
q inline enable
```

在 CloudShell 中使用 Q 聊天命令

该 `q chat` 命令允许您通过终端提问和接收来自 Amazon Q 的回复。要启动与 Amazon Q 的对话，请在 CloudShell 终端中运行 `q chat` 命令。有关更多信息，请参阅《Amazon Q 开发者版用户指南》中的在 CLI 中使用 Amazon Q 进行聊天。

在 CloudShell 中使用 Q 翻译命令

该 `q translate` 命令允许您编写自然语言指令。要使用 Amazon Q 进行翻译，请在 CloudShell 终端中运行 `q translate` 命令。有关更多信息，请参阅《Amazon Q 开发者版用户指南》中的从自然语言转换为 bash。

CloudShell 中的 CLI 命令补全

当您在终端中键入内容时，CloudShell 中的 CLI 补全会为您提供命令和选项建议。有关更多信息，请参阅《Amazon Q 开发者版用户指南》中的生成命令行补全。

启用或禁用 Amazon Q CLI

您可以通过选择首选项、启用 Amazon Q CLI 和禁用 Amazon Q CLI 来启用或禁用 Amazon Q CLI。使用 Amazon Q CLI，您可以通过终端与 Amazon Q 进行交互，通过自然语言指令提问并获取答案。

当您在终端中键入内容时，它还会为您提供命令建议。当您开始在终端中键入内容时，Amazon Q 会建议相关选项来完成您的命令。

CloudShell 中基于身份的 Amazon Q CLI 策略

要在 CloudShell 中使用 Amazon Q CLI，请确保具有所需的 IAM 权限。有关更多信息，请参阅《Amazon Q 开发者版用户指南》中的适用于 Amazon Q 开发者版的基于身份的策略示例。

从 AWS 服务控制台在 CloudShell 中运行命令

您可以通过 AWS 管理控制台中的 Amazon ElastiCache 和 Amazon DocumentDB（兼容 MongoDB）控制台，在 CloudShell 终端中运行命令。

要从其他 AWS 服务控制台在 CloudShell 中运行命令，分配给您的角色的 IAM 策略必须包含 `cloudshell:approveCommand` 权限。

CloudShell 在控制台工具栏上打开，运行命令弹出窗口将出现在 CloudShell 中。在运行命令弹出窗口中，命令出现在命令框中。

要在 CloudShell 终端中运行命令，请选择以下步骤之一：

1. 如果您尚未在 CloudShell 中创建 VPC 环境，请在新环境名称框中输入名称。

   您可以查看基于您资源的 VPC 详情构建的 VPC 环境详情。

   a. 选择创建并运行。

      此步骤将创建一个新的 CloudShell VPC 环境并在 CloudShell 终端中运行该命令。

2. 如果您已经创建了 CloudShell VPC 环境，则可以查看 CloudShell 环境名称。

   > **Note**
   >
   > 如果您已经有一个 CloudShell VPC 环境，则无法创建新 VPC 环境。

   a. 选择运行。

      此步骤将在所选 CloudShell VPC 环境的 CloudShell 终端中运行该命令。

      > **Note**
      >
      > 如果您没有查看已创建的 VPC 环境的权限，请联系您的管理员以添加 `cloudshell:describeEnvironments` 权限。有关更多信息，请参阅使用 IAM 策略管理 AWS CloudShell 访问和使用情况。

您可以继续在 CloudShell 终端中运行命令。

自定义您的 AWS CloudShell 体验

您可以自定义以下方面的 AWS CloudShell 体验：

- 标签页布局：将命令行界面拆分为多列和多行。
- 字体大小：调整命令行文本的大小。
- 颜色主题：在浅色和深色主题之间切换。
- 安全粘贴：开启或关闭一项要求您在粘贴多行文本之前对其进行验证的功能。
- Tmux 到会话恢复：使用 tmux 可以恢复会话，直到会话变为非活动状态。
- Amazon Q CLI：通过 Amazon Q CLI 您可以使用 Amazon Q CLI 功能。

您还可以通过安装自己的软件和使用脚本修改 Shell 来扩展 Shell 环境。

将命令行显示拆分成多个标签页

通过将命令行界面拆分为多个窗格来运行多个命令。

> **Note**
>
> 打开多个标签页后，您可以通过单击所选窗格中的任意位置来选择一个要处理的标签页。您可以通过选择区域名称旁边的 x 符号来关闭标签页。

- 从标签页布局中选择操作和以下选项之一：
  - 新建标签页：在当前活动标签页旁边添加一个新标签页。
  - 拆分成行：在当前活动标签页下方的行中添加一个新标签页。
  - 拆分为列：在当前活动标签页旁边的列中添加一个新标签页。

  如果空间不足，无法完全显示每个标签页，请滚动查看整个标签页。您也可以选择分隔窗格的分隔条，然后使用指针拖动分隔条来增加或减小窗格大小。

更改字体大小

增大或减小命令行界面中显示的文本的大小。

1. 要更改 AWS CloudShell 终端设置，请前往设置、首选项。
2. 选择文本大小。选项有最小、小、中、大和最大。

更改界面主题

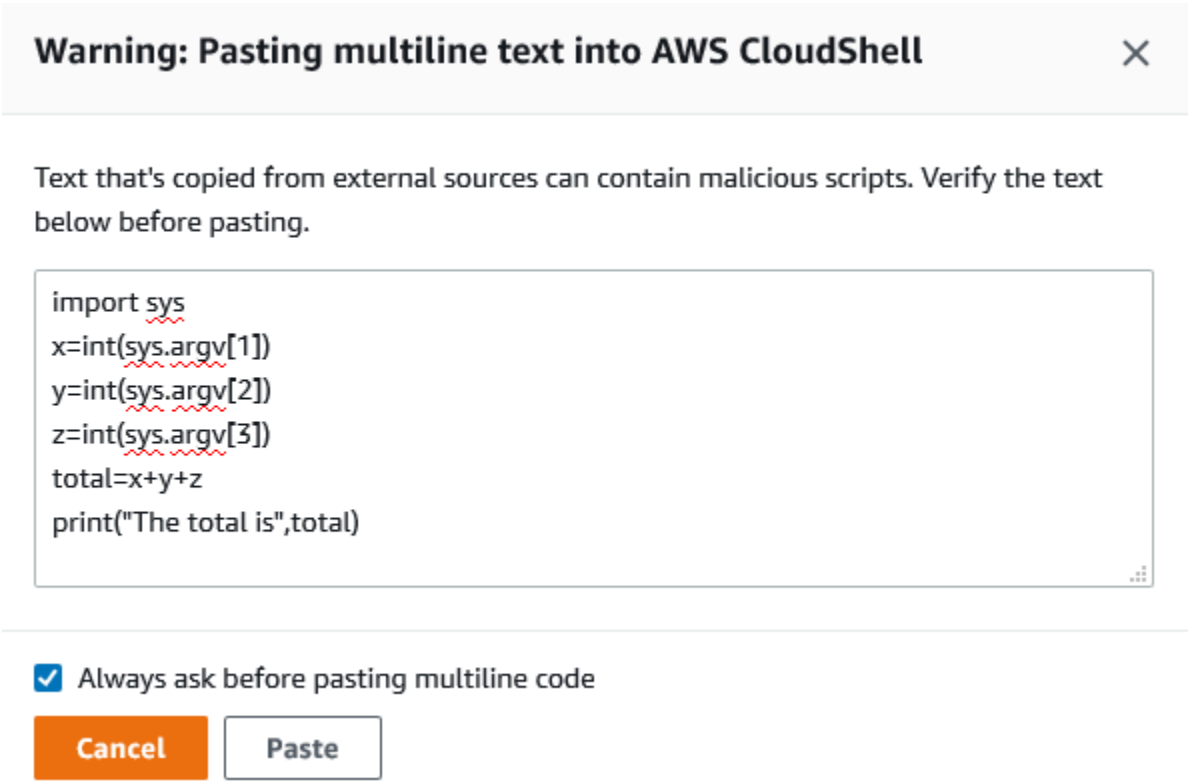
在命令行界面的浅色和深色主题之间切换。

1. 要更改 AWS CloudShell 主题，请前往设置、首选项。
2. 选择浅色或深色。

对多行文本使用安全粘贴

“安全粘贴”是一项安全功能，可提示您验证要粘贴到 Shell 中的多行文本是否包含恶意脚本。从第三方网站复制的文本可能包含隐藏代码，这些代码会在您的 Shell 环境中触发意外行为。

“安全粘贴”对话框显示您复制到剪贴板的完整文本。如果您对不存在安全风险感到满意，请选择粘贴。



我们建议您启用“安全粘贴”以捕获脚本中的潜在安全风险。您可以通过选择首选项、启用安全粘贴和禁用安全粘贴来打开或关闭此功能。

使用 tmux 进行会话恢复

AWS CloudShell 使用 tmux 在单个或多个浏览器标签页上恢复会话。如果您刷新浏览器标签页，它会恢复您的会话，直到会话变为非活动状态。有关更多信息，请参阅会话恢复。

使用 Amazon Q CLI

您可以通过选择首选项、启用 Amazon Q CLI 和禁用 Amazon Q CLI 来启用或禁用 Amazon Q CLI。有关更多信息，请参阅启用/禁用 Amazon Q CLI。

AWS CloudShell 在亚马逊 VPC 中使用

AWS CloudShell 虚拟私有云 (VPC) 使您能够在 VPC 中创建 CloudShell 环境。对于每个 VPC 环境，您可以分配一个 VPC、添加子网以及关联最多五个安全组。 AWS CloudShell 继承 VPC 的网络配置，使您能够与 VPC 中的其他资源在同一个子网中 AWS CloudShell 安全使用并连接到这些资源。

借助 Amazon VPC，您可以在您定义的逻辑隔离的虚拟网络中启动 AWS 资源。这个虚拟网络与您在数据中心中运行的传统网络极其相似，并会为您提供使用 AWS的可扩展基础设施的优势。有关 VPC 的更多信息，请参阅 Amazon Virtual Private Cloud。

操作限制

AWS CloudShell VPC 环境有以下限制：

- 对于每个 IAM 主体，最多只能创建两个 VPC 环境。
- 最多可将五个安全组分配给一个 VPC 环境。
- 您不能在 VPC 环境中使用 “操作” 菜单中的 CloudShell 上传和下载选项。

> **Note**
>
> 可以从可 ingress/egress 通过其他 CLI 工具访问互联网的 VPC 环境中上传或下载文件。

- VPC 环境不支持永久存储。支持临时性存储。活动环境会话结束后，数据和主目录将被删除。
- 您的 AWS CloudShell 环境只有在私有 VPC 子网中才能连接到互联网。

> **Note**
>
> 默认情况下，不向 CloudShell VPC 环境分配公有 IP 地址。在将路由表配置为将所有流量路由到互联网网关的公有子网中创建的 VPC 环境将无法访问公共互联网，但配置了网络地址转换（NAT）的私有子网可以访问公共互联网。在此类私有子网中创建的 VPC 环境将可以访问公共互联网。

- 要为您的账户提供托管 CloudShell 环境， AWS 可以为底层计算主机配置对以下服务的网络访问权限：
  - Amazon S3
  - VPC 端点

- com.amazonaws.<region>.ssmmessages
- com.amazonaws.<region>.logs
- 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

无法通过修改 VPC 配置来限制对这些端点的访问。

CloudShell VPC 在所有 AWS 地区和 GovCloud 地区都可用。有关可用 CloudShell VPC 的区域列表，请参阅支持的 AWS 区域 AWS CloudShell。

创建 V CloudShell PC 环境

本主题将引导您完成在中创建 VPC 环境的步骤 CloudShell。

先决条件

要创建 VPC 环境，管理员必须为您提供所需的 IAM 权限。有关启用创建 CloudShell VPC 环境的权限的更多信息，请参阅the section called “创建和使用 CloudShell VPC 环境所需的 IAM 权限”。

创建 CloudShell VPC 环境

1. 在 CloudShell 控制台页面上，选择 + 图标，然后从下拉菜单中选择创建 VPC 环境。
2. 在创建 VPC 环境页面上，在名称框中输入 VPC 环境的名称。
3. 在虚拟私有云（VPC）下拉列表中，选择一个 VPC。
4. 从子网下拉列表中，选择一个子网。
5. 从安全组下拉列表中，选择要分配给您的 VPC 环境的一个或多个安全组。

> **Note**
>
> 最多可以选择五个安全组。

6. 选择创建以创建您的 VPC 环境。
7. （可选）选择操作，然后选择查看详细信息，查看新创建的 VPC 环境的详细信息。您的 VPC 环境的 IP 地址将显示在命令行提示符中。

有关使用 VPC 环境的信息，请参阅开始使用。

创建和使用 CloudShell VPC 环境所需的 IAM 权限

要创建和使用 CloudShell VPC 环境，IAM 管理员必须允许访问特定于 VPC 的 Amazon EC2 权限。本节列出了创建和使用 VPC 环境所需的 Amazon EC2 权限。

要创建 VPC 环境，分配给您的角色的 IAM 策略必须包含以下 Amazon EC2 权限：

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

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

建议包括：

- ec2:DeleteNetworkInterface

> **Note**
>
> 此权限不是强制性的，但这是清理由 CloudShell 其创建的 ENI 资源（为 CloudShell VPC 环境 ENIs 创建的，标有ManagedByCloudShell密钥）所必需的。如果未启用此权限，则必须在每次 CloudShell VPC 环境使用后手动清理 ENI 资源。

授予完全 CloudShell 访问权限（包括 VPC 访问权限）的 IAM 策略

以下示例显示了如何启用完全权限，包括对 VPC 的访问权限 CloudShell：

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Sid": "AllowCloudShellOperations",
   "Effect": "Allow",
   "Action": [
    "cloudshell:*"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AllowDescribeVPC",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeSubnets",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeVpcs"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AllowInspectVPCConfigurationViaCloudShell",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeDhcpOptions",
    "ec2:DescribeNetworkInterfaces"
   ],
```

```
    "Resource": "*",
    "Condition": {
     "ForAnyValue:StringEquals": {
      "aws:CalledVia": "cloudshell.amazonaws.com"
     }
    }
   },
   {
    "Sid": "AllowCreateTagWithCloudShellKeyViaCloudShell",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:*:*:network-interface/*",
    "Condition": {
     "StringEquals": {
      "ec2:CreateAction": "CreateNetworkInterface"
     },
     "ForAnyValue:StringEquals": {
      "aws:TagKeys": "ManagedByCloudShell",
      "aws:CalledVia": "cloudshell.amazonaws.com"
     }
    }
   },
   {
    "Sid": "AllowCreateNetworkInterfaceWithSubnetsAndSGViaCloudShell",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateNetworkInterface"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:subnet/*",
     "arn:aws:ec2:*:*:security-group/*"
    ],
    "Condition": {
     "ForAnyValue:StringEquals": {
      "aws:CalledVia": "cloudshell.amazonaws.com"
     }
    }
   },
   {
    "Sid": "AllowCreateNetworkInterfaceWithCloudShellTagViaCloudShell",
    "Effect": "Allow",
    "Action": [
```

```
    "ec2:CreateNetworkInterface"
   ],
   "Resource": "arn:aws:ec2:*:*:network-interface/*",
   "Condition": {
    "ForAnyValue:StringEquals": {
     "aws:TagKeys": "ManagedByCloudShell",
     "aws:CalledVia": "cloudshell.amazonaws.com"
    }
   }
  },
  {
   "Sid": "AllowCreateNetworkInterfacePermissionWithCloudShellTagViaCloudShell",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateNetworkInterfacePermission"
   ],
   "Resource": "arn:aws:ec2:*:*:network-interface/*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceTag/ManagedByCloudShell": ""
    },
    "ForAnyValue:StringEquals": {
     "aws:CalledVia": "cloudshell.amazonaws.com"
    }
   }
  },
  {
   "Sid": "AllowDeleteNetworkInterfaceWithCloudShellTagViaCloudShell",
   "Effect": "Allow",
   "Action": [
    "ec2:DeleteNetworkInterface"
   ],
   "Resource": "arn:aws:ec2:*:*:network-interface/*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceTag/ManagedByCloudShell": ""
    },
    "ForAnyValue:StringEquals": {
     "aws:CalledVia": "cloudshell.amazonaws.com"
    }
   }
  }
 ]
```

```
}
```

将 IAM 条件键用于 VPC 环境

您可以在 VPC 设置中使用 CloudShell特定条件密钥，为您的 VPC 环境提供额外的权限控制。还可以指定可以使用和不能使用 VPC 环境的子网和安全组。

CloudShell 在 IAM 策略中支持以下条件键：

- `CloudShell:VpcIds`— 允许或拒绝一个或多个 VPCs
- `CloudShell:SubnetIds` – 允许或拒绝一个或多个子网
- `CloudShell:SecurityGroupIds` – 允许或拒绝一个或多个安全组

> **Note**
>
> 如果修改了有权访问公共 CloudShell 环境的用户的权限以增加对`cloudshell:createEnvironment`操作的限制，他们仍然可以访问其现有的公共环境。但是，如果您想修改具有此限制的 IAM 策略并禁用他们对现有公共环境的访问权限，则必须先使用该限制更新 IAM 策略，然后确保您账户中的每个 CloudShell 用户都使用 CloudShell Web 用户界面（操作 → 删除 CloudShell 环境）手动删除现有的公共环境。

带有用于 VPC 设置的条件键的策略示例

以下示例演示如何将条件键用于 VPC 设置。创建具有所需限制的策略语句后，为目标用户或角色附加策略语句。

确保用户仅创建 VPC 环境并拒绝创建公共环境

要确保用户只能创建 VPC 环境，请使用拒绝权限，如以下示例所示：

```
{
  "Statement": [
    {
      "Sid": "DenyCloudShellNonVpcEnvironments",
      "Action": [
        "cloudshell:CreateEnvironment"
```

```
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "Null": {
          "cloudshell:VpcIds": "true"
        }
      }
    }
  ]
}
```

拒绝用户访问特定 VPCs、子网或安全组

要拒绝用户访问特定的 VPCs内容，StringEquals请使用检查cloudshell:VpcIds条件的值。以下示例拒绝用户访问 vpc-1 和 vpc-2：

要拒绝用户访问特定的 VPCs内容，StringEquals请使用检查cloudshell:SubnetIds条件的值。以下示例拒绝用户访问 subnet-1 和 subnet-2：

要拒绝用户访问特定的 VPCs内容，StringEquals请使用检查cloudshell:SecurityGroupIds条件的值。以下示例拒绝用户访问 sg-1 和 sg-2：

JSON

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

```
      }
    }
  ]
}
```

允许用户使用特定 VPC 配置创建环境

要允许用户访问特定的 VPCs，StringEquals请使用检查cloudshell:VpcIds条件的值。以下示例允许用户访问 vpc-1 和 vpc-2：

要允许用户访问特定的 VPCs，StringEquals请使用检查cloudshell:SubnetIds条件的值。以下示例允许用户访问 subnet-1 和 subnet-2：

JSON

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

要允许用户访问特定的 VPCs，StringEquals请使用检查cloudshell:SecurityGroupIds条件的值。以下示例允许用户访问 sg-1 和 sg-2：

JSON

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

的安全性 AWS CloudShell

云安全性一直是 Amazon Web Services (AWS) 的重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。安全是双方共同承担 AWS 的责任。责任共担模式将其描述为云的安全性和云中的安全性。

云安全 — AWS 负责保护运行 AWS 云中提供的所有服务的基础架构，并为您提供可以安全使用的服务。我们的安全责任是重中之重 AWS，作为AWS 合规计划的一部分，第三方审计师定期测试和验证我们安全的有效性。

云端安全 — 您的责任由您使用的 AWS 服务以及其他因素决定，包括数据的敏感性、组织的要求以及适用的法律和法规。

AWS CloudShell 通过其支持的特定 AWS 服务遵循分担责任模式。有关 AWS 服务安全信息，请参阅AWS 服务安全文档页面和合规计划合 AWS 规工作范围内的AWS 服务。

以下主题向您介绍如何进行配置 AWS CloudShell 以满足您的安全和合规性目标。

主题

- 中的数据保护 AWS CloudShell
- 适用于 AWS 的 Identity and Access 管理 CloudShell
- 登录和监控 AWS CloudShell
- 合规性验证 AWS CloudShell
- 韧性在 AWS CloudShell
- 基础设施安全 AWS CloudShell
- 的安全最佳实践 AWS CloudShell
- AWS CloudShell 安全性 FAQs

中的数据保护 AWS CloudShell

分 AWS 担责任模型适用于中的数据保护 AWS CloudShell。如本模型所述 AWS ，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础结构上的内容的控制。您还负责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅 AWS Security Blog 上的 AWS Shared Responsibility Model and GDPR 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证（MFA）。
- 用于 SSL/TLS 与 AWS 资源通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。有关使用 CloudTrail 跟踪捕获 AWS 活动的信息，请参阅《AWS CloudTrail 用户指南》中的使用跟 CloudTrail 踪。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准（FIPS）第 140-3 版》https://aws.amazon.com/compliance/fips/。

强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括您使用控制台、API AWS CloudShell 或以其他 AWS 服务 方式使用控制台 AWS CLI、API 或时 AWS SDKs。在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供 URL，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

数据加密

数据加密是指在数据处于静止状态时， AWS CloudShell 以及数据在与服务端点 AWS CloudShell 之间传输时保护数据。

使用静态加密 AWS KMS

静态加密是指通过对存储数据进行加密来保护您的数据免受未经授权的访问。使用时 AWS CloudShell，您可以免费获得每个 AWS 区域 1 GB 的永久存储空间。持久性存储位于您的主目录 (`$HOME`) 中，对您而言是私有的。与每个 Shell 会话结束后回收的临时环境资源不同的是，主目录中的数据会保留。

存储在中的数据的加密 AWS CloudShell 是使用 AWS Key Management Service (AWS KMS) 提供的加密密钥实现的。这是一项用于创建和控制的托管 AWS 服务， AWS KMS keys即用于加密存储在 AWS CloudShell 环境中的客户数据的加密密钥。 AWS CloudShell 代表客户生成和管理用于加密数据的加密密钥。

传输中加密

传输中加密是指在通信终端节点之间移动数据时，保护您的数据免遭拦截。

默认情况下，客户端的 Web 浏览器计算机与基于云 AWS CloudShell 的计算机之间的所有数据通信都通过 HTTPS/TLS 连接发送所有内容进行加密。

您无需执行任何操作即可使用 HTTPS/TLS 进行通信。

适用于 AWS 的 Identity and Access 管理 CloudShell

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以进行身份验证（登录）和授权（拥有权限）使用 CloudShell 资源。您可以使用 IAM AWS 服务 ，无需支付额外费用。

主题

受众

您的使用方式 AWS Identity and Access Management (IAM) 因您的角色而异：

- 服务用户：如果您无法访问功能，请向管理员申请权限（请参阅对 AWS CloudShell 身份和访问进行故障排除）
- 服务管理员 - 确定用户访问权限并提交权限请求（请参阅 AWS 如何 CloudShell 与 IAM 合作）
- IAM 管理员 - 编写用于管理访问权限的策略（请参阅 AWS 基于身份的策略示例 CloudShell）

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份进行身份验证 AWS 账户根用户，或者通过担任 IAM 角色进行身份验证。

您可以使用来自身份源的证书 AWS IAM Identity Center （例如（IAM Identity Center）、单点登录身份验证或 Google/Facebook 证书，以联合身份登录。有关登录的更多信息，请参阅《AWS 登录 用户指南》中的如何登录您的 AWS 账户。

对于编程访问， AWS 提供 SDK 和 CLI 来对请求进行加密签名。有关更多信息，请参阅《IAM 用户指南》中的适用于 API 请求的AWS 签名版本 4。

AWS 账户 root 用户

创建时 AWS 账户，首先会有一个名为 AWS 账户 root 用户的登录身份，该身份可以完全访问所有资源 AWS 服务 和资源。我们强烈建议不要使用根用户进行日常任务。有关需要根用户凭证的任务，请参阅《IAM 用户指南》中的需要根用户凭证的任务。

联合身份

作为最佳实践，要求人类用户使用与身份提供商的联合身份验证才能 AWS 服务 使用临时证书进行访问。

联合身份是指来自您的企业目录、Web 身份提供商的用户 Directory Service ，或者 AWS 服务 使用来自身份源的凭据进行访问的用户。联合身份代入可提供临时凭证的角色。

要集中管理访问权限，建议使用。 AWS IAM Identity Center有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的什么是 IAM Identity Center？。

IAM 用户和群组

IAM 用户是对单个人员或应用程序具有特定权限的一个身份。建议使用临时凭证，而非具有长期凭证的 IAM 用户。有关更多信息，请参阅 IAM 用户指南中的要求人类用户使用身份提供商的联合身份验证才能 AWS 使用临时证书进行访问。

IAM 组指定一组 IAM 用户，便于更轻松地对大量用户进行权限管理。有关更多信息，请参阅《IAM 用户指南》中的 IAM 用户的使用案例。

IAM 角色

IAM 角色是具有特定权限的身份，可提供临时凭证。您可以通过从用户切换到 IAM 角色（控制台）或调用 AWS CLI 或 AWS API 操作来代入角色。有关更多信息，请参阅《IAM 用户指南》中的担任角色的方法。

IAM 角色对于联合用户访问、临时 IAM 用户权限、跨账户访问、跨服务访问以及在 Amazon 上运行的应用程序非常有用。 EC2有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略定义了与身份或资源关联时的权限。 AWS 在委托人提出请求时评估这些政策。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的更多信息，请参阅《IAM 用户指南》中的 JSON 策略概述。

管理员使用策略，通过定义哪个主体可以对什么资源以及在什么条件下执行操作，来指定谁有权访问什么内容。

默认情况下，用户和角色没有权限。IAM 管理员创建 IAM 策略并将其添加到角色中，然后用户可以代入这些角色。IAM 策略定义权限，而不考虑您使用哪种方法来执行操作。

基于身份的策略

基于身份的策略是您附加到身份（用户、组或角色）的 JSON 权限策略文档。这些策略控制身份可在何种条件下对哪些资源执行什么操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

基于身份的策略可以是内联策略（直接嵌入到单个身份中）或托管策略（附加到多个身份的独立策略）。要了解如何在托管策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的在托管策略与内联策略之间进行选择。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。您必须在基于资源的策略中指定主体。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

其他策略类型

AWS 支持其他策略类型，这些策略类型可以设置更常见的策略类型授予的最大权限：

- 权限边界 – 设置基于身份的策略可以授予 IAM 实体的最大权限。有关更多信息，请参阅《 IAM 用户指南》中的 IAM 实体的权限边界。
- 服务控制策略 (SCPs)-在中指定组织或组织单位的最大权限 AWS Organizations。有关更多信息，请参阅《AWS Organizations 用户指南》中的服务控制策略。
- 资源控制策略 (RCPs)-设置账户中资源的最大可用权限。有关更多信息，请参阅《AWS Organizations 用户指南》中的资源控制策略 (RCPs)。
- 会话策略 – 在为角色或联合用户创建临时会话时，作为参数传递的高级策略。有关更多信息，请参阅《IAM 用户指南》中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的策略评估逻辑。

AWS 如何 CloudShell 与 IAM 合作

在使用 IAM 管理访问权限之前 CloudShell，请先了解有哪些 IAM 功能可供使用 CloudShell。

您可以在 AWS 中使用的 IAM 功能 CloudShell

IAM 功能	CloudShell 支持
基于身份的策略	是
基于资源的策略	否
策略操作	是
策略资源	是
策略条件键（特定于服务）	是
ACLs	否
ABAC（策略中的标签）	否

IAM 功能	CloudShell 支持
临时凭证	是
转发访问会话 (FAS)	否
服务角色	否
服务关联角色	否

要全面了解 CloudShell 以及其他 AWS 服务如何与大多数 IAM 功能配合使用，请参阅 IAM 用户指南中的与 IAM 配合使用的AWS 服务。

基于身份的策略 CloudShell

支持基于身份的策略：是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素引用。

基于身份的策略示例 CloudShell

要查看 CloudShell 基于身份的策略的示例，请参阅。AWS 基于身份的策略示例 CloudShell

内部基于资源的政策 CloudShell

支持基于资源的策略：否

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户访问，您可以将整个账户或其他账户中的 IAM 实体指定为基于资源的策略中的主体。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

的政策行动 CloudShell

支持策略操作：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。在策略中包含操作以授予执行关联操作的权限。

要查看 CloudShell 操作列表，请参阅《服务授权参考》 CloudShell中的 AWS 定义的操作。某些操作可能有多个 API。

正在执行的策略操作在操作前 CloudShell 使用以下前缀：

```
cloudshell
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

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

要查看 CloudShell 基于身份的策略的示例，请参阅。AWS 基于身份的策略示例 CloudShell

的政策资源 CloudShell

支持策略资源：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

`Resource` JSON 策略元素指定要向其应用操作的一个或多个对象。作为最佳实践，请使用其 Amazon 资源名称（ARN）指定资源。对于不支持资源级权限的操作，请使用通配符 (*) 指示语句应用于所有资源。

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

要查看 CloudShell 资源类型及其列表 ARNs，请参阅《服务授权参考》 CloudShell中的 AWS 定义的资源。要了解您可以使用哪些操作来指定每种资源的 ARN，请参阅 AWS 定义的操作。 CloudShell

要查看 CloudShell 基于身份的策略的示例，请参阅。AWS 基于身份的策略示例 CloudShell

的策略条件密钥 CloudShell

支持特定于服务的策略条件键：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

`Condition` 元素指定语句何时根据定义的标准执行。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的AWS 全局条件上下文密钥。

要查看 CloudShell 条件密钥列表，请参阅《服务授权参考》 CloudShell中的 AWS 条件密钥。要了解您可以使用条件键的操作和资源，请参阅 AWS 定义的操作 CloudShell。

要查看 CloudShell 基于身份的策略的示例，请参阅。AWS 基于身份的策略示例 CloudShell

ACLs in CloudShell

支持 ACLs：否

访问控制列表 (ACLs) 控制哪些委托人（账户成员、用户或角色）有权访问资源。 ACLs 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

ABAC with CloudShell

支持 ABAC（策略中的标签）：否

基于属性的访问权限控制（ABAC）是一种授权策略，该策略基于称为标签的属性来定义权限。您可以将标签附加到 IAM 实体和 AWS 资源，然后设计 ABAC 策略以允许在委托人的标签与资源上的标签匹配时进行操作。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`aws:RequestTag/*key-name* 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的使用 ABAC 授权定义权限。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的使用基于属性的访问权限控制（ABAC）。

将临时证书与 CloudShell

支持临时凭证：是

临时证书提供对 AWS 资源的短期访问权限，并且是在您使用联合身份或切换角色时自动创建的。AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的临时安全凭证和使用 IAM 的。AWS 服务

当您切换角色时，将使用不同的环境。您不能在同一个 AWS CloudShell 环境中切换角色。

转发访问会话 CloudShell

支持转发访问会话（FAS）：否

转发访问会话 (FAS) 使用调用主体的权限 AWS 服务，再加上 AWS 服务 向下游服务发出请求的请求。有关发出 FAS 请求时的策略详细信息，请参阅转发访问会话。

CloudShell 的服务角色

支持服务角色：否

服务角色是由一项服务担任、代表您执行操作的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。

> **Warning**
>
> 更改服务角色的权限可能会中断 CloudShell 功能。只有在 CloudShell 提供操作指导时才编辑服务角色。

的服务相关角色 CloudShell

支持服务相关角色：否

服务相关角色是一种与服务相关联的 AWS 服务服务角色。服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务关联角色的权限。

AWS 基于身份的策略示例 CloudShell

默认情况下，用户和角色没有创建或修改 CloudShell 资源的权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅《IAM 用户指南》中的创建 IAM 策略（控制台）。

有关由 CloudShell定义的操作和资源类型（包括每种资源类型的格式）的详细信息，请参阅《服务授权参考》 CloudShell中的 AWS 操作、资源和条件密钥。 ARNs

主题

- 策略最佳实践
- 使用 CloudShell 控制台
- 允许用户查看他们自己的权限

策略最佳实践

基于身份的策略决定了某人是否可以在您的账户中创建、访问或删除 CloudShell 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的AWS 托管式策略或工作职能的AWS 托管式策略。
- 应用最低权限：在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。
- 使用 IAM 策略中的条件进一步限制访问权限：您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性：IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言（JSON）和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的使用 IAM Access Analyzer 验证策略。

- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的使用 MFA 保护 API 访问。

有关 IAM 中的最佳实操的更多信息，请参阅 IAM 用户指南中的 IAM 中的安全最佳实操。

使用 CloudShell 控制台

要访问 AWS CloudShell 控制台，您必须拥有一组最低权限。这些权限必须允许您列出和查看有关您的 CloudShell 资源的详细信息 AWS 账户。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（用户或角色），控制台将无法按预期正常运行。

对于仅调用 AWS CLI 或 AWS API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与其尝试执行的 API 操作相匹配的操作。

为确保用户和角色仍然可以使用 CloudShell 控制台，还需要将 CloudShell *ConsoleAccess*或*ReadOnly* AWS 托管策略附加到实体。有关更多信息，请参阅《IAM 用户指南》中的为用户添加权限。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
```

```
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

对 AWS CloudShell 身份和访问进行故障排除

使用以下信息来帮助您诊断和修复在使用 CloudShell 和 IAM 时可能遇到的常见问题。

主题

- 我无权在以下位置执行操作 CloudShell
- 我无权执行 iam：PassRole
- 我想允许我以外的人 AWS 账户 访问我的 CloudShell 资源

我无权在以下位置执行操作 CloudShell

如果您收到错误提示，指明您无权执行某个操作，则必须更新策略以允许执行该操作。

当 mateojackson IAM 用户尝试使用控制台查看有关虚构 *my-example-widget* 资源的详细信息，但不拥有虚构 awes:*GetWidget* 权限时，会发生以下示例错误。

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

在此情况下，必须更新 mateojackson 用户的策略，以允许使用 awes:*GetWidget* 操作访问 *my-example-widget* 资源。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我无权执行 iam：PassRole

如果您收到一个错误，表明您无权执行 `iam:PassRole` 操作，则必须更新策略以允许您将角色传递给。 CloudShell

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 `marymajor` 的 IAM 用户尝试使用控制台在 CloudShell 中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

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

在这种情况下，必须更新 Mary 的策略以允许她执行 `iam:PassRole` 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我想允许我以外的人 AWS 账户 访问我的 CloudShell 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以代入角色。对于支持基于资源的策略或访问控制列表 (ACLs) 的服务，您可以使用这些策略向人们授予访问您的资源的权限。

要了解更多信息，请参阅以下内容：

- 要了解是否 CloudShell 支持这些功能，请参阅AWS 如何 CloudShell 与 IAM 合作。
- 要了解如何提供对您拥有的资源的访问权限 AWS 账户 ，请参阅 IAM 用户指南中的向您拥有 AWS 账户 的另一个 IAM 用户提供访问权限。
- 要了解如何向第三方提供对您的资源的访问权限 AWS 账户，请参阅 IAM 用户指南中的向第三方提供访问权限。 AWS 账户
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的为经过外部身份验证的用户（身份联合验证）提供访问权限。
- 要了解使用角色和基于资源的策略进行跨账户访问之间的差别，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

使用 IAM 策略管理 AWS CloudShell 访问和使用情况

通过提供的访问管理资源 AWS Identity and Access Management，管理员可以向 IAM 用户授予权限。这样，这些用户就可以访问 AWS CloudShell 和使用环境的功能。管理员还可以创建策略，在精细级别上指定这些用户可以在 Shell 环境中执行哪些操作。

管理员向用户授予访问权限的最快方法是通过 AWS 托管策略。AWS 托管策略 是由 AWS创建和管理的独立策略。以下的 AWS 托管策略 AWS CloudShell 可以附加到 IAM 身份：

- AWS CloudShellFullAccess：授予使用 AWS CloudShell 的权限，并具有对所有功能的完全访问权限。

该AWS CloudShellFullAccess策略使用通配符 (*) 来授予 IAM 身份（用户、角色或群组）对 CloudShell和功能的完全访问权限。有关此策略的更多信息，请参阅AWS CloudShellFullAccess《AWS 托管策略用户指南》。

> **Note**
>
> 也可以启动具有以下 AWS 托管策略的 IAM 身份 CloudShell。但是，这些策略提供了广泛的权限。因此，我们建议您仅在这些策略对于 IAM 用户的工作角色必不可少的情况下才授予这些策略。
>
> - 管理员：为 IAM 用户提供完全访问权限，并允许他们向中的每项服务和资源委派权限 AWS。
> - 开发者高级用户：使 IAM 用户能够执行应用程序开发任务，创建和配置支持 AWS 感知型应用程序开发的资源和服务。
>
> 有关如何将托管策略附加到您的实体的更多信息，请参阅 IAM 用户指南中的添加 IAM 身份权限（控制台）。

AWS CloudShell 使用自定义策略管理允许的操作

要管理 IAM 用户可以执行的操作 CloudShell，请创建使用 CloudShellPolicy 托管策略作为模板的自定义策略。或者，编辑嵌入在相关 IAM 身份（用户、组或角色）中的内联策略。

例如，您可以允许 IAM 用户访问 CloudShell，但阻止他们转发用于登录的 CloudShell 环境证书 AWS 管理控制台。

Important

要 AWS CloudShell 从启动 AWS 管理控制台，IAM 用户需要执行以下操作的权限：

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

如果附加的策略未明确允许其中一项操作，则在您尝试启动时会返回 IAM 权限错误 CloudShell。

AWS CloudShell 权限

Name	对已授予权限的描述	需要启动 CloudShell吗？
`cloudshell:CreateEnvironment`	创建 CloudShell 环境，在会 CloudShell 话开始时检索布局，并将来自 Web 应用程序的当前布局保存在后端。如the section called “适用于 IAM 策略的示例 CloudShell”中所述，此权限仅期望将 * 作为 `Resource` 的值。	是
`cloudshell:CreateSession`	从连接到 CloudShell 环境 AWS 管理控制台。	是
`cloudshell:GetEnvironmentStatus`	读取 CloudShell 环境的状态。	是

Name	对已授予权限的描述	需要启动 CloudShell吗？
cloudshell:DeleteEnvironment	删除 CloudShell 环境。	否
cloudshell:GetFileDownloadUrls	生成预签名 URLs 的 Amazon S3，用于 CloudShell 通过 CloudShell 网络界面下载文件。这不适用于 VPC 环境。	否
cloudshell:GetFileUploadUrls	生成预签名 URLs 的 Amazon S3，用于 CloudShell 通过 CloudShell 网络界面上传文件。这不适用于 VPC 环境。	否
cloudshell:DescribeEnvironments	描述环境。	否
cloudshell:PutCredentials	将用于登录的凭据转发 AWS 管理控制台 到。CloudShell	否
cloudshell:StartEnvironment	启动已停止的 CloudShell 环境。	是
cloudshell:StopEnvironment	停止正在运行的 CloudShell 环境。	否
cloudshell:ApproveCommand	批准 CloudShell 从其他 AWS 服务控制台发送到的命令。	否

适用于 IAM 策略的示例 CloudShell

以下示例说明如何创建策略来限制谁可以访问 CloudShell。这些示例还显示可在 Shell 环境中执行的操作。

以下政策强制完全拒绝访问 CloudShell 及其功能。

JSON

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

以下策略允许 IAM 用户进行访问， CloudShell 但禁止他们生成 URLs 用于文件上传和下载的预签名。用户仍然可以将文件传入和传出环境，例如使用 `wget` 等客户端。

JSON

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

```
        "Action": [
            "cloudshell:GetFileDownloadUrls",
            "cloudshell:GetFileUploadUrls"
        ],
        "Resource": "*"
    }]
}
```

以下策略允许 IAM 用户进行访问 CloudShell。但是，该策略禁止将您用于登录的凭据转发 AWS 管理控制台 到 CloudShell 环境。使用此策略的 IAM 用户需要在其中手动配置其证书 CloudShell。

JSON

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

以下策略允许 IAM 用户创建 AWS CloudShell 环境。

JSON

```
{
```

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

创建和使用 CloudShell VPC 环境所需的 IAM 权限

要创建和使用 CloudShell VPC 环境，IAM 管理员必须允许访问特定于 VPC 的 Amazon EC2 权限。本节列出了创建和使用 VPC 环境所需的 Amazon EC2 权限。

要创建 VPC 环境，分配给您的角色的 IAM 策略必须包含以下 Amazon EC2 权限：

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

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

建议包括：

- ec2:DeleteNetworkInterface

> **Note**
>
> 此权限不是强制性的，但这是清理由 CloudShell 其创建的 ENI 资源（为 CloudShell VPC 环境 ENIs 创建的，标有ManagedByCloudShell密钥）所必需的。如果未启用此权限，则必须在每次 CloudShell VPC 环境使用后手动清理 ENI 资源。

授予完全 CloudShell访问权限（包括 VPC 访问权限）的 IAM 策略

以下示例显示了如何启用完全权限，包括对 VPC 的访问权限 CloudShell：

JSON

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

将 IAM 条件键用于 VPC 环境

您可以在 VPC 设置中使用 CloudShell特定条件密钥，为您的 VPC 环境提供额外的权限控制。还可以指定可以使用和不能使用 VPC 环境的子网和安全组。

CloudShell 在 IAM 策略中支持以下条件键：

- `CloudShell:VpcIds`— 允许或拒绝一个或多个 VPCs
- `CloudShell:SubnetIds` – 允许或拒绝一个或多个子网
- `CloudShell:SecurityGroupIds` – 允许或拒绝一个或多个安全组

> **Note**
>
> 如果修改了有权访问公共 CloudShell 环境的用户的权限以增加对`cloudshell:createEnvironment`操作的限制，他们仍然可以访问其现有的公共环境。但是，如果您想修改具有此限制的 IAM 策略并禁用他们对现有公共环境的访问权限，则必须先使用该限制更新 IAM 策略，然后确保您账户中的每个 CloudShell 用户都使用 CloudShell Web 用户界面（操作 → 删除 CloudShell环境）手动删除现有的公共环境。

带有用于 VPC 设置的条件键的策略示例

以下示例演示如何将条件键用于 VPC 设置。创建具有所需限制的策略语句后，为目标用户或角色附加策略语句。

确保用户仅创建 VPC 环境并拒绝创建公共环境

要确保用户只能创建 VPC 环境，请使用拒绝权限，如以下示例所示：

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

拒绝用户访问特定 VPCs、子网或安全组

要拒绝用户访问特定的 VPCs内容，StringEquals请使用检查cloudshell:VpcIds条件的值。以下示例拒绝用户访问 vpc-1 和 vpc-2：

要拒绝用户访问特定的 VPCs内容，StringEquals请使用检查cloudshell:SubnetIds条件的值。以下示例拒绝用户访问 subnet-1 和 subnet-2：

要拒绝用户访问特定的 VPCs内容，StringEquals请使用检查cloudshell:SecurityGroupIds条件的值。以下示例拒绝用户访问 sg-1 和 sg-2：

JSON

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

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

允许用户使用特定 VPC 配置创建环境

要允许用户访问特定的 VPCs，StringEquals请使用检查cloudshell:VpcIds条件的值。以下示例允许用户访问 vpc-1 和 vpc-2：

要允许用户访问特定的 VPCs，StringEquals请使用检查cloudshell:SubnetIds条件的值。以下示例允许用户访问 subnet-1 和 subnet-2：

JSON

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

```
          "cloudshell:SubnetIds": [
            "subnet-1",
            "subnet-2"
          ]
        }
      }
    }
  ]
}
```

要允许用户访问特定的 VPCs，StringEquals请使用检查cloudshell:SecurityGroupIds条件的值。以下示例允许用户访问 sg-1 和 sg-2：

JSON

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

访问权限 AWS 服务

CloudShell 使用您用于登录的 IAM 证书 AWS 管理控制台。

> **Note**
>
> 要使用您用于登录的 IAM 证书 AWS 管理控制台，您必须拥有`cloudshell:PutCredentials`权限。

的这种预身份验证功能 CloudShell 便于使用 AWS CLI。但是，IAM 用户仍然需要通过命令行调 AWS 服务 用的显式权限。

例如，假设 IAM 用户需要创建 Amazon S3 存储桶并将文件作为对象上传给这些存储桶。您可以创建明确允许这些操作的策略。IAM 控制台提供了一个交互式可视化编辑器，用于指导构建 JSON 格式的策略文档的过程。创建策略后，您可以将其附加到相关的 IAM 身份（用户、组或角色）。

有关如何将托管策略附加到您的实体的更多信息，请参阅 IAM 用户指南中的添加 IAM 身份权限（控制台）。

访问中 Amazon Q CLI 功能的权限 CloudShell

要在中使用 Amazon Q CLI 功能 CloudShell，例如内联建议、聊天和翻译，请确保您拥有所需的 IAM 权限。如果您无法访问中的 Amazon Q CLI 功能 CloudShell，请联系您的管理员为您提供必要的 IAM 权限。有关更多信息，请参阅《Amazon Q 开发者版用户指南》中的适用于 Amazon Q 开发者版的基于身份的策略示例。

登录和监控 AWS CloudShell

本主题介绍如何使用记录和监控 AWS CloudShell 活动和性能 CloudTrail。

使用监控活动 CloudTrail

AWS CloudShell 与 AWS CloudTrail一项服务集成，该服务提供用户、角色或 AWS 服务 中的操作记录 AWS CloudShell。 CloudTrail 将所有 API 调用捕获 AWS CloudShell 为事件。捕获的调用包括来自 AWS CloudShell 控制台的调用和对 AWS CloudShell API 的代码调用。

如果您创建了跟踪，则可以允许将 CloudTrail 事件持续传送到亚马逊简单存储服务 (Amazon S3) 存储桶。这包括以下活动： AWS CloudShell.

如果您不配置跟踪记录，则仍可在 CloudTrail 控制台的 Event history（事件历史记录）中查看最新事件。使用收集的信息 CloudTrail，您可以发现有关请求的各种信息。例如，您可以确定向 AWS 发出的请求 CloudShell，可以了解发出请求的 IP 地址、谁发出了请求以及何时发出。

AWS CloudShell in CloudTrail

下表列出了保存在 CloudTrail 日志文件中的 AWS CloudShell 事件。

> **Note**
>
> AWS CloudShell 活动包括：
>
> - * 表示它是一个非变易（只读）API 调用。
> - 单词 Environment 与托管 Shell 体验的计算环境的生命周期有关。
> - 这个词可以Layout恢复 CloudShell终端中的所有浏览器选项卡。

CloudShell 中的活动 CloudTrail

事件名称	说明
createEnvironment	在创建 CloudShell 环境时发生。
createSession	当 CloudShell 环境从连接时发生 AWS 管理控制台。
deleteEnvironment	在删除 CloudShell 环境时发生。
deleteSession	在当前浏览器 CloudShell 选项卡中运行的选项卡中的会话被删除时发生。
getEnvironmentStatus*	在检索 CloudShell 环境状态时发生。
getFileDownloadUrls*	生成用于通过 CloudShell CloudShell 网络界面下载文件的预签名 Amazon S3 URLs 时发生。
getFileUploadUrls*	生成用于使用 CloudShell 网页界面上传文件的预签名 Amazon S3 URLs 时发生。 CloudShell
cloudshell:DescribeEnvironments	描述环境。
getLayout*	在检索会话开始时的 CloudShell 布局时发生。

事件名称	说明
putCredentials	当用于登录到的凭据 CloudShell 被 AWS 管理控制台 转发时发生。
redeemCode*	在 CloudShell环境中检索刷新令牌的工作流程开始时发生。您稍后可以在putCredentials 命令中使用此令牌来访问 CloudShell 环境。
sendHeartBeat	用于确认会 CloudShell 话处于活动状态。
startEnvironment	在 CloudShell 环境启动时发生。
stopEnvironment	在运行 CloudShell 环境停止时发生。
updateLayout	保存后端 Web 应用程序的当前布局时发生。

包含 “布局” 一词的事件会恢复 CloudShell终端中的所有浏览器选项卡。

EventBridge AWS CloudShell 行动规则

使用 EventBridge 规则，您可以指定在 EventBridge 收到与规则匹配的事件时要采取的目标操作。您可以定义一条规则，根据 CloudTrail 日志文件中记录为事件的 AWS CloudShell 操作来指定要采取的目标操作。

例如，您可以使用put-rule命令创建 EventBridge 规则。 AWS CLI呼put-rule叫必须至少包含 EventPattern 或 ScheduleExpression。当观察到匹配的事件时，会触发带有 EventPatterns 的规则。EventPattern 对于 AWS CloudShell 活动：

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

有关更多信息，请参阅 Amazon EventBridge 用户指南 EventBridge中的事件和事件模式。

合规性验证 AWS CloudShell

作为多个合规计划的一部分，第三方审计师对 AWS 服务的安全性和 AWS 合规性进行评估。

AWS CloudShell 符合以下合规计划：

SOC

AWS 系统和组织控制 (SOC) 报告是独立的第三方检查报告，用于展示如何 AWS 实现关键合规控制和目标。

服务	SDK	SOC 1、2、3
AWS CloudShell	CloudShell	✓

PCI

支付卡行业数据安全标准（PCI DSS）是由PCI安全标准委员会管理的专有信息安全标准，该委员会由美国运通、Discover Financial Services、JCB International、Worldw MasterCard ide和Visa Inc.创立。

服务	SDK	PCI
AWS CloudShell	CloudShell	✓

ISO 和 CSA STAR 认证和服务

AWS 已获得 27001:2013、27017:2015、27018:2019、 ISO/IEC 27701:2019、22301:2019、9001:2015 和 CSA STAR CCM v4.0 的合规认证。

服务	SDK	ISO 和 CSA STAR 认证和服务
AWS CloudShell	CloudShell	✓

FedRamp

联邦风险与授权管理计划（FedRAMP）是一项美国政府层面的计划，它提供一种标准方法来对云产品和云服务进行安全性评估、授权以及持续监控。

服务	SDK	FedRAMP 中等（东部/西部）	FedRamp High () GovCloud
AWS CloudShell	CloudShell	✓	✓

DoD CC SRG

国防部 (DoD)《云计算安全要求指南》(SRG) 为云服务提供商 (CSPs) 提供了标准化的评估和授权流程，以获得国防部的临时授权，以便他们能够为国防部客户提供服务。

通过 DoD CC SRG 评估和授权的服务将具有以下状态：

- 第三方评估组织 (3PAO) 评估：该服务目前正在接受我们的第三方评估员的评估。
- 联合授权委员会 (JAB) 审查：该服务目前正在接受 JAB 审查。
- 美国国防信息系统局 (DISA) 审查：该服务目前正在接受 DISA 审查。

服务	SDK	国防部 CC SRG IL2（东/西）	国防部 CC SR IL2 G () GovCloud	国防部 CC SR IL4 G () GovCloud	国防部 CC SR IL5 G () GovCloud	国防部 CC SRG IL6（AWS 秘密区域）
AWS CloudShell	CloudShell	✓	✓	✓	✓	不适用

HIPAA BAA

1996 年的《健康保险流通与责任法案》(HIPAA) 是一项联邦法律，要求制定国家标准以保护敏感的患者健康信息，避免此类信息在未经患者同意或知情的情况下遭到披露。

AWS 使受 HIPAA 约束的受保实体及其业务伙伴能够安全地处理、存储和传输受保护的健康信息 (PHI)。此外，自 2013 年 7 月起， AWS 为此类客户提供标准化的商业伙伴附录 (BAA)。

服务	SDK	HIPAA BAA
AWS CloudShell	CloudShell	✓

IRAP

通过信息安全注册评估员计划 (IRAP)，澳大利亚政府客户能够验证适当的控制措施是否到位，并确定适当的责任模式，以满足澳大利亚网络安全中心 (ACSC) 编制的《澳大利亚政府信息安全手册》(ISM) 的要求。

服务	命名空间*	IRAP 受保护
AWS CloudShell	不适用	✓

*命名空间可帮助您识别环境中的服务。 AWS 例如，在创建 IAM 策略时，使用 Amazon 资源名称 (ARNs) 并读取 AWS CloudTrail 日志。

MTCS

多层云安全 (MTCS) 是一项可操作的新加坡安全管理标准（SPRING SS 584），基于 ISO 27001/02 信息安全管理系统 (ISMS) 标准。

服务	SDK	美国东部（俄亥俄州）	美国东部（弗吉尼亚州北部）	美国西部（俄勒冈州）	美国西部（北加利福尼亚）	新加坡	首尔
AWS CloudShell	CloudShell	✓	✓	✓	不适用	不适用	不适用

C5

云计算合规性控制目录 (C5) 是由德国联邦信息安全办公室 (BSI) 在德国推出的一项由德国政府支持的认证计划，旨在帮助企业在德国政府的“云提供商安全建议”所规定的环境中使用云服务时，展示企业在运营方面防范常见网络攻击的安全措施。

服务	SDK	C5
AWS CloudShell	CloudShell	✓

ENS High

ENS（Esquema Nacional de Seguridad）认证计划由金融和公共管理部与 CCN（国家密码学中心）共同制定。这包括充分保护信息所必需的基本原则和最低要求。

服务	SDK	ENS High
AWS CloudShell	CloudShell	✓

FINMA

瑞士金融市场监管局（FINMA）是瑞士独立的金融市场监管机构。 AWS符合 FINMA 要求表明我们持续致力于满足瑞士金融服务监管机构和客户对云服务提供商提出的更高期望。

服务	SDK	FINMA
AWS CloudShell	CloudShell	✓

PiTuKri

AWS 与 PiTuKri 需求保持一致表明了我们对满足芬兰交通和通信局Traficom对云服务提供商更高的期望的持续承诺。

服务	SDK	PiTuKri
AWS CloudShell	CloudShell	✓

有关特定合规计划范围内的 AWS 服务列表，请参阅按合规计划划分的 AWS 范围内的服务 AWS 按合规计划。有关一般信息，请参阅AWS 合规计划AWS。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅在 AWS Artifact 中下载报告。

您在使用 AWS CloudShell 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。 AWS 提供了以下资源来帮助实现合规性：

- 安全与合规性快速入门指南讨论了架构注意事项，并提供了在上面部署以安全为重点和以合规为重点的基准环境的步骤。 AWS

- HIPAA 安全与合规架构白皮书 — 本白皮书描述了公司如何使用来 AWS 创建符合 HIPAA 标准的应用程序。
- AWS 合规资源AWS — 此工作簿和指南集可能适用于您所在的行业和所在地区。
- 使用AWS Config 开发人员指南中的规则评估资源 — 该 AWS Config 服务评估您的资源配置在多大程度上符合内部实践、行业指导方针和法规。
- AWS Security Hub CSPM— 此 AWS 服务可全面了解您的安全状态 AWS ，帮助您检查是否符合安全行业标准和最佳实践。

韧性在 AWS CloudShell

AWS 全球基础设施是围绕 AWS 区域和可用区构建的。 AWS 区域提供多个物理隔离和隔离的可用区，这些可用区通过低延迟、高吞吐量和高度冗余的网络相连。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错能力和可扩展性。

有关 AWS 区域和可用区的更多信息，请参阅AWS 全球基础设施。

除了 AWS 全球基础架构外，还 AWS CloudShell 支持以下功能来支持您的数据弹性和备份需求：

- 使用 AWS CLI 调用在中指定主目录中的 AWS CloudShell 文件，并将它们作为对象添加到 Amazon S3 存储桶中。有关示例，请参阅开始使用 AWS CloudShell。

基础设施安全 AWS CloudShell

作为一项托管服务 AWS CloudShell ，受 AWS 全球网络安全的保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅AWS 云安全。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅 S AWS ecurity Pillar Well-Architected Fram ework 中的基础设施保护。

您可以使用 AWS 已发布的 API 调用 AWS CloudShell 通过网络进行访问。客户端必须支持以下内容：

- 传输层安全性协议（TLS）。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密（PFS）的密码套件，例如 DHE（临时 Diffie-Hellman）或 ECDHE（临时椭圆曲线 Diffie-Hellman）。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

> **Note**
>
> 默认情况下，AWS CloudShell 自动为计算环境的系统包安装安全补丁。

的安全最佳实践 AWS CloudShell

以下最佳实践是一般指导原则，并不代表完整安全解决方案。由于这些最佳实践可能不适合您的环境或无法满足您的环境要求，因此建议将其视为有用的考虑因素，而不是惯例。

以下方面的一些安全最佳实践 AWS CloudShell

- 使用 IAM 权限和策略来控制访问权限，AWS CloudShell 并确保用户只能执行其角色要求的操作（例如，下载和上传文件）。有关更多信息，请参阅使用 IAM 策略管理 AWS CloudShell 访问和使用情况。
- 请勿在您的 IAM 实体中包含敏感数据，例如用户、角色或会话名称。
- 启用“保持安全粘贴”功能，以捕获从外部源复制的文本中的潜在安全风险。默认情况下，“安全粘贴”处于启用状态。有关对多行文本使用安全粘贴的更多信息，请参阅对多行文本使用安全粘贴。
- 如果您在 AWS CloudShell的计算环境中安装了第三方应用程序，请了解共担安全责任模式。
- 在编辑会影响用户 Shell 体验的 Shell 脚本之前，请准备好回滚机制。有关修改默认 shell 环境的更多信息，请参阅使用脚本修改 shell。
- 将代码安全地存储在版本控制系统中。

AWS CloudShell 安全性 FAQs

以下是有关安全性的常见问题的答案 CloudShell。

- 启动和启动 shell 会话时使用了哪些 AWS 流程 CloudShell 和技术？
- 是否可以将网络访问限制为 CloudShell？
- 我可以自定义我的 CloudShell 环境吗？
- 我的 `$HOME` 目录实际上存储在 AWS Cloud中的什么地方？
- 可以加密我的 `$HOME` 目录吗？
- 我能否对我的 `$HOME` 目录进行病毒扫描？

启动和启动 shell 会话时使用了哪些 AWS 流程 CloudShell 和技术？

登录时 AWS 管理控制台，您需要输入您的 IAM 用户证书。而且，当您 CloudShell 从控制台界面启动时，这些凭据将用于调用为服务创建计算环境的 CloudShell API。然后为计算环境创建一个 AWS Systems Manager 会话，并 CloudShell 向该会话发送命令。

返回安全清单 FAQs

是否可以将网络访问限制为 CloudShell？

对于公共环境，无法限制网络访问。如果要限制网络访问，必须启用仅创建 VPC 环境的权限并拒绝创建公共环境。

有关更多信息，请参阅确保用户仅创建 VPC 环境并拒绝创建公共环境。

对于 CloudShell VPC 环境，网络设置继承自您的 VPC。 CloudShell 在 VPC 中使用可以控制您的 CloudShell VPC 环境的网络访问权限。

返回安全清单 FAQs

我可以自定义我的 CloudShell 环境吗？

您可以为您的 CloudShell 环境下载和安装实用程序和其他第三方软件。只有安装在 $HOME 目录中的软件才会在会话之间保留。

根据AWS 责任共担模式的定义，您负责对所安装的应用程序进行必要的配置和管理。

返回安全清单 FAQs

我的 **$HOME** 目录实际上存储在 AWS Cloud中的什么地方？

对于公共环境，用于在 $HOME 中存储数据的基础结构由 Amazon S3 提供。

对于 VPC 环境，当您的 VPC 环境超时（处于非活动状态 20-30 分钟后），或者当您删除或重启环境时，您的 $HOME 目录将被删除。

返回安全清单 FAQs

可以加密我的 **$HOME** 目录吗？

不可以，不能用自己的密钥加密您的 $HOME 目录。但是在将您的$HOME目录内容存储在 Amazon S3 中时会对其 CloudShell 进行加密。

返回安全清单 FAQs

我能否对我的 $HOME 目录进行病毒扫描？

目前，无法对您的 $HOME 目录进行病毒扫描。对此功能的支持正在审核中。

返回安全清单 FAQs

我能否限制我的数据入口或出口？ CloudShell

要限制入口或出口，我们建议您使用 VP CloudShell C 环境。当您的 VPC 环境超时（处于非活动状态 20-30 分钟后），或者当您删除或重启环境时，VPC 环境的 $HOME 目录将被删除。在操作菜单中，上传和下载选项不适用于 VPC 环境。

返回安全清单 FAQs

AWS CloudShell 计算环境：规格和软件

当您启动 AWS CloudShell 时，系统会创建一个基于 Amazon Linux 2023 的计算环境以托管 Shell 体验。该环境配置了计算资源（vCPU 和内存），并提供了大量可从命令行界面访问的预安装软件。确保您在计算环境中安装的任何软件都经过修补并处于最新状态。还可以通过安装软件和修改 Shell 脚本来配置默认环境。

计算环境资源

每个 AWS CloudShell 计算环境都分配了以下 CPU 和内存资源：

- 1 个 vCPU（虚拟中央处理器）
- 2 GiB RAM

而且，为环境配置了以下存储配置：

- 1GB 持久性存储空间（会话结束后存储空间仍保留）

有关更多信息，请参阅 持久性存储。

CloudShell 网络要求

WebSockets

CloudShell 依赖于 WebSocket 协议，该协议允许用户的 Web 浏览器与 AWS 云端的 CloudShell 服务进行双向交互式通信。如果您在专用网络中使用浏览器，则代理服务器和防火墙可能有助于安全访问互联网。WebSocket 通信通常可以毫无问题地遍历代理服务器。但是在某些情况下，代理服务器会阻止 WebSocket 正常运行。如果出现此问题，您的 CloudShell 界面会报告以下错误：`Failed to open sessions : Timed out while opening the session`。

如果此错误反复出现，请查看代理服务器的文档，确保其配置为允许 WebSocket。或者，您可以与网络的系统管理员联系。

> **Note**
>
> 如果要通过允许列出特定 URL 来定义精细权限，则可以添加 AWS Systems Manager 会话用于打开 WebSocket 连接以发送输入和接收输出的 URL 的一部分。（您的 AWS CloudShell 命令将发送到该 Systems Manager 会话。）
> Systems Manager 使用的此 StreamURL 的格式为 `wss://ssmmessages.region.amazonaws.com/v1/data-channel/session-id?stream=(input|output)`。
> 区域表示 AWS Systems Manager 支持的 AWS 区域的区域标识符，例如美国东部（俄亥俄州）区域的 `us-east-2`。
> 由于会话 ID 是在特定 Systems Manager 会话成功启动后创建的，因此您只能在更新 URL 允许列表时指定 `wss://ssmmessages.region.amazonaws.com`。有关更多信息，请参阅 AWS Systems Manager API 参考中的开始会话操作。

预安装的软件

> **Note**
>
> 由于 AWS CloudShell 开发环境会定期更新以提供对最新软件的访问权限，因此我们在本文档中不提供具体的版本号。相反，我们将介绍如何检查安装了哪个版本。要查看已安装的版本，请输入程序名称，然后选择 `--version` 选项（例如，`git --version`）。

Shell

预安装的 Shell

名称	描述	版本信息
Bash	Bash Shell 是 AWS CloudShell 的默认 Shell 应用程序。	`bash --version`
PowerShell (pwsh)	PowerShell 提供命令行界面和脚本语言支持，建立在 Microsoft 的 .NET 命令语言运行时之上。PowerShell 使用名	`pwsh --version`

名称	描述	版本信息
	为 cmdlets 的轻量级命令来接受和返回 .NET 对象。	
Z Shell (zsh)	Z Shell，也被称为 zsh，是 Bourne Shell 的扩展版本，它为主题和插件提供了增强的自定义支持。	`zsh --version`

AWS 命令行界面（CLI）

CLI

名称	描述	版本信息
AWS CDK Toolkit CLI	AWS CDK Toolkit，即 CLI 命令 cdk，是与您的 AWS CDK 应用程序交互的主要工具。它执行您的应用程序，查询您定义的应用程序模型，并生成和部署由 AWS CDK 生成的 AWS CloudFormation 模板。 有关更多信息，请参阅 AWS CDK 工具包。	`cdk --version`
AWS CLI	AWS CLI 是一个命令行界面，您可以使用它从命令行管理多个 AWS 服务，并利用脚本自动执行这些服务。有关更多信息，请参阅 在 CLI 中管理 AWS 服务 CloudShell。 有关如何确保使用的是最新版 AWS CLI 版本 2 的信息，请参阅 安装 AWS CLI 到主目录。	`aws --version`

名称	描述	版本信息
EB CLI	AWS Elastic Beanstalk CLI 提供了可简化从本地存储库创建、更新和监控环境的命令行界面。 有关 EB CLI 的更多信息，请参阅AWS Elastic Beanstalk 开发人员指南中的使用 Elastic Beanstalk 命令行界面（EB CLI）。	`eb --version`
Amazon ECS CLI	Amazon Elastic Container Service（Amazon ECS）命令行界面（CLI）提供高级命令，以简化集群和任务的创建、更新和监控。 有关更多信息，请参阅 Amazon Elastic Container Service 开发人员指南中的 Amazon ECS 命令行界面。	`ecs-cli --version`
AWS SAM CLI	AWS SAM CLI 是一个命令行工具，可在 AWS Serverless Application Model 模板和应用程序代码上运行。您可以执行多项任务。其中包括在本地调用 Lambda 函数、为无服务器应用程序创建部署包，以及将您的无服务器应用程序部署到 AWS 云端。 有关更多信息，请参阅AWS Serverless Application Model 开发人员指南中的AWS SAM CLI 指令参考。	`sam --version`

名称	描述	版本信息
AWS Tools for PowerShell	AWS Tools for PowerShell 是根据由 适用于 .NET 的 SDK 公开的功能构建的 PowerShell 模块。借助 AWS Tools for PowerShell，您可以从 PowerShell 命令行为 AWS 资源上操作编写脚本。 AWS CloudShell 预安装 AWS Tools for PowerShell 的模块化版本 (AWS.Tools)。更多信息，请参阅AWS Tools for PowerShell 用户指南中的使用 AWS Tools for PowerShell。	`pwsh --Command 'Get-AWSPowerShellVersion'`

运行时系统和 AWS SDK：Node.js 和 Python 3

运行时系统和 AWS SDK

名称	描述	版本信息
Node.js（带 npm）	Node.js 是一个 JavaScript 运行时系统，旨在让异步编程技术的应用变得更容易。有关更多信息，请参阅 Node.js 官方网站上的文档。 npm 是一个软件包管理器，提供对 JavaScript 模块在线注册表的访问。有关更多信息，请参阅 npm 官方网站上的文档。	• Node.js: `node --version` • npm：`npm --version`
Node.js 中的 SDK for JavaScript	软件开发工具包（SDK）通过为包括 Amazon S3、Amazon EC2、DynamoDB 和 Amazon	`npm -g ls --depth 0 2>/dev/null | grep aws-sdk`

名称	描述	版本信息
	SWF 在内的 Amazon Web Services 提供 JavaScript 对象，来帮助简化编码。有关更多信息，请参见适用于 JavaScript 的 AWS SDK 开发人员指南。	
Python	Python 3 随时可在 Shell 环境中使用。Python 3 现在被认为是该编程语言的默认版本（对 Python 2 的支持已于 2020 年 1 月结束）。有关更多信息，请参阅 Python 官方网站上的文档。 此外，预装的是 Pip，这是一款适用于 Python 的软件包安装程序。您可以使用此命令行程序从在线索引（例如 Python 程序包索引）中安装 Python 软件包。有关更多信息，请参阅 Python 打包权威机构提供的文档。	• Python 3: `python3 --version` • pip: `pip3 --version`
适用于 Python 的 SDK (Boto3)	Boto 是 Python 开发人员用来创建、配置和管理 AWS 服务（例如 Amazon EC2 和 Amazon S3）的软件开发工具包（SDK）。该 SDK 提供了易于使用的、面向对象的 API，以及对 AWS 服务 的低级别访问。 更多信息，请参阅 Boto3 文档。	`pip3 list | grep boto3`

开发工具和 Shell 实用程序

开发工具和 Shell 实用程序

名称	描述	版本信息
bash-completion	bash-completion 是一个 Shell 函数集合，允许通过按 Tab 键自动完成部分键入的命令或参数。您可以在 `/usr/share/bash-completion/completions` 中找到 bash-completion 支持的软件包。 要为软件包的命令设置自动完成功能，必须获取该程序文件。例如，要为 Git 命令设置自动完成功能，请在 `.bashrc` 中添加以下一行，这样每当您的 AWS CloudShell 会话开始时，该功能都可用： `source /usr/share/bash-completion/completions/git` 如果要使用自定义完成脚本，请将它们添加到持久性主目录 (`$HOME`) 中，然后直接从 `.bashrc` 中获取它们。 有关更多信息，请参阅 GitHub 上项目的 README 页面。	`dnf info bash-completion`
cqlsh-expansion	`cqlsh-expansion` 是一套工具包，包含 cqlsh 及相关辅助工具，这些工具已针对 Amazon Keyspaces 预	`cqlsh-expansion --version`

名称	描述	版本信息
	配置，同时保持与 Apache Cassandra 的完全兼容性。有关更多信息，请参阅《Amazon Keyspaces（Apache Cassandra 兼容）开发人员指南》中的使用 cqlsh 连接到 Amazon Keyspaces。	
Docker	Docker 是用于开发、发布和运行应用程序的开放平台。借助 Docker，您可以将应用程序与基础结构分开，以便快速交付软件，并且可以在 AWS CloudShell 内部构建 Dockerfiles，并使用 CDK 构建 Docker 资产。有关 Docker 支持哪些 AWS 区域的信息，请参阅 AWS CloudShell 支持的 AWS 区域。您应该知道，Docker 在环境中的空间有限。如果您的单个映像较大，或者预先存在的 Docker 映像过多，则可能会导致问题。有关 Docker 的更多信息，请参阅 Docker 文档指南。	`docker --version`
Git	Git 是一个分布式版本控制系统，它通过分支工作流程和内容暂存支持现代软件开发实践。有关更多信息，请参阅 Git 官方网站上的文档页面。	`git --version`

名称	描述	版本信息
iputils	iputils 软件包包含用于 Linux 联网的实用程序。有关提供的实用工具的更多信息，请参阅 GitHub 上的 iputils 存储库。	iputils 工具的示例：`arping -V`
jq	jq 实用程序解析 JSON 格式的数据，以生成由命令行过滤器修改的输出。有关更多信息，请参阅 GitHub 上托管的 jq 手册。	`jq --version`
kubectl	kubectl 是一个命令行工具，用于使用 Kubernetes API 与 Kubernetes 集群的控制面板进行通信。	`kubectl --version`
make	make 实用程序使用 `makefiles` 自动执行任务集并组织代码编译。有关更多信息，请参阅 GNU Make 文档	`make --version`
man	man 命令提供命令行实用程序和工具的手册页。例如，`man ls` 返回列出目录内容的 `ls` 命令的手册页。更多信息，请参阅 Wikipedia 的 man 词条。	`man --version`
nano	nano 是一款小型且用户友好的、基于文本界面的编辑器。有关更多信息，请参阅 GNU nano 文档。	`nano --version`

名称	描述	版本信息
OpenJDK 21	Amazon Corretto 21 是 OpenJDK 21 的长期支持（LTS）发行版。Amazon Corretto 是开放 Java 开发工具包（OpenJDK）的免费、多平台、生产就绪型分发版。有关更多信息，请参阅《Corretto 21 用户指南》中的什么是 Amazon Corretto 21？	`java -version`
procps	procps 是一个系统管理实用程序，可用于监控和停止当前正在运行的进程。有关更多信息，请参阅列出可以使用 procps 运行的程序的 README 文件。	`ps --version`
psql	PostgreSQL 是一个功能强大的开源数据库系统，它使用标准 SQL 功能，同时提供强大的功能来安全地管理和扩展复杂的数据操作。有关更多信息，请参阅什么是 PostgreSQL。	`psql --version`
SSH 客户端	SSH 客户端使用安全 Shell 协议与远程计算机进行加密通信。OpenSSH 是预安装的 SSH 客户端。有关更多信息，请参阅 OpenBSD 维护的 OpenSSH 网站。	`ssh -V`

名称	描述	版本信息
sudo	使用 sudo 实用程序，用户可以使用其他用户（通常是超级用户）的安全权限运行程序。当您需要以系统管理员身份安装应用程序时，Sudo 非常有用。有关更多信息，请参阅 Sudo 手册。	`sudo --version`
tar	tar 是一个命令行实用程序，可用于将多个文件分组到单个存档文件（通常称为 tarball）中。有关更多信息，请参阅 GNU tar 文档。	`tar --version`
tmux	tmux 是一个终端多路复用器，可用于在多个窗口中同时运行不同的程序。有关更多信息，请参阅简要介绍 tmux 的博客。	`tmux -V`
vim	Vim 是一个可自定义的编辑器，您可以通过基于文本的界面与之交互。更多信息，请参阅 vim.org 上提供的文档资源。	`vim --version`
wget	wget 是一种计算机程序，用于从命令行中的端点指定的 Web 服务器检索内容。有关更多信息，请参阅 GNU Wget 文档。	`wget --version`

名称	描述	版本信息

zip/unzip	zip/unzip 实用程序使用存档文件格式，可在不丢失数据的情况下提供无损数据压缩。调用 zip 命令将文件分组并压缩到单个存档中。使用 unzip 将存档中的文件解压缩到指定目录中。	`unzip --version`
		
		`zip --version`

安装 AWS CLI 到主目录

与 CloudShell 环境中预安装的其他软件一样，AWS CLI 工具会自动使用定期升级和安全补丁进行更新。如果要确保使用的是最新版本 AWS CLI，可以选择在 Shell 的主目录中手动安装该工具。

> **Important**
>
> 您需要在主目录中手动安装 AWS CLI 副本，以便下次启动 CloudShell 会话时可用。之所以需要此安装，是因为在完成 Shell 会话后，添加到 `$HOME` 之外的目录的文件将被删除。此外，安装此 AWS CLI 副本后，它不会自动更新。换句话说，管理更新和安全补丁是您的责任。
> 有关 AWS 责任共担模式的更多信息， 请参阅 中的数据保护 AWS CloudShell。

要安装 AWS CLI，请执行以下操作

1. 在 CloudShell 命令行中，使用 `curl` 命令将已安装 AWS CLI 的压缩副本传输到 Shell：

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

2. 对压缩文件夹进行解压缩：

   ```
   unzip awscliv2.zip
   ```

3. 要将工具添加到指定文件夹，请运行 AWS CLI 安装程序：

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

如果安装成功，命令行会显示以下消息：

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

4. 为了方便起见，我们建议您同时更新 PATH 环境变量，这样在运行 aws 命令时就无需指定工具的安装路径：

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

> **Note**
>
> 如果撤消对 PATH 的这一更改，则未采用指定路径的 aws 命令将默认使用 AWS CLI 的预装版本。

在 Shell 环境中安装第三方软件

> **Note**
>
> 建议您在向 AWS CloudShell 的计算环境安装任何第三方应用程序之前，先查看安全责任共担模式。

默认设置下，所有 AWS CloudShell 用户均无 sudo 权限。因此，您可以使用 sudo 命令来安装 Shell 计算环境中尚不可用的软件。例如，您可以结合使用 sudo 和 DNF 软件包管理实用程序来安装 cowsay，以便使用以下消息生成包含一只牛的 ASCII 格式的艺术图片：

```
sudo dnf install cowsay
```

然后，您可以通过键入 `echo "Welcome to AWS CloudShell" | cowsay` 来启动新安装的程序。

> **Important**
>
> dnf 等软件包管理实用程序会将程序安装到目录（例如 /usr/bin），而这些程序会在 Shell 会话结束时被回收。这意味着要在每个会话的基础上安装和使用其他软件。

使用脚本修改 Shell

如果您要修改默认的 Shell 环境，可以编辑每次启动 Shell 环境时都会运行的 Shell 脚本。每当默认 bash shell 启动时，`.bashrc` 脚本就会运行。

> **Warning**
>
> 如果您错误地修改了 `.bashrc` 文件，则以后可能无法访问您的 Shell 环境。在编辑文件之前，最好先制作一份文件副本。您还可以在编辑 `.bashrc` 时通过打开两个 Shell 来降低风险。如果您在一个 Shell 中失去访问权限，则仍可以登录到另一个 Shell，并且可以回滚任何更改。
>
> 如果您在错误修改 `.bashrc` 或任何其他文件后确实失去了访问权限，则可以通过删除主目录将 AWS CloudShell 恢复其默认设置。

在此过程中，您将修改 `.bashrc` 脚本，以便您的 Shell 环境自动切换到运行 Z shell。

1. 使用文本编辑器（例如 Vim）打开 `.bashrc`：

   ```
   vim .bashrc
   ```

2. 在编辑器界面中，按 I 键开始编辑，然后添加以下内容：

   ```
   zsh
   ```

3. 要退出并保存编辑后的 `.bashrc` 文件，请 Esc 进入 Vim 命令模式并输入以下内容：

   ```
   :wq
   ```

4. 使用 `source` 命令重新加载 `.bashrc` 文件：

   ```
   source .bashrc
   ```

   当命令行界面再次可用时，提示符号已更改为 % 表示您现在正在使用 Z shell。

将 AWS CloudShell 从 AL2 迁移到 AL2023

AWS CloudShell 以前基于 Amazon Linux 2 (AL2)，现在已迁移到 Amazon Linux 2023 (AL2023)。有关更多信息，请参阅《Amazon Linux 2023 用户指南》中的什么是 Amazon Linux 2023 (AL2023)。

在 AL2023 中，您可以继续使用 CloudShell 提供的所有工具访问现有的 CloudShell 环境。有关可用工具的更多信息，请参阅预安装的软件。

AL2023 对开发工具进行了多项改进，包括更新版本的软件包，例如 Node.js 18 和 Python 3.9。

> **Note**
>
> 在 AL2023 中，Python 2 不再随 CloudShell 环境一起提供。

有关 AL2 和 AL2023 操作系统之间差异的详细信息，请参阅《Amazon Linux 2023 用户指南》中的比较 Amazon Linux 2 和 Amazon Linux 2023。

如果您有任何疑问，请联系 支持。 您还可以在 AWS re:Post 上搜索答案和发布问题。当您进入 AWS re:Post 时，可能会要求您登录到 AWS。

AWS CloudShell 迁移常见问题解答

以下是有关使用 AWS CloudShell 从 AL2 迁移到 AL2023 的一些常见问题的解答。

- 迁移到 AL2023 是否会影响我的任何其他 AWS 资源，例如在 AL2 上运行的 Amazon EC2 实？
- 迁移到 AL2023 后将更改哪些软件包？
- 我可以选择不迁移吗？
- 我能否为我的 AWS CloudShell 环境创建备份？

迁移到 AL2023 是否会影响我的任何其他 AWS 资源，例如在 AL2 上运行的 Amazon EC2 实例？

除了您的 AWS CloudShell 环境之外，没有任何服务或资源受到此次迁移的影响。这包括您可能已在 AWS CloudShell 内部创建或访问的资源。例如，如果您创建了一个在 AL2 上运行的 Amazon EC2 实例，则该实例将不会迁移到 AL2023。

迁移到 AL2023 后更改了哪些软件包？

AWS CloudShell 环境当前包括预安装的软件。要了解预安装软件的完整列表，请参阅预安装的软件。AWS CloudShell 将继续交付这些软件包，Python 2 除外。有关 AL2 和 AL2023 提供的软件包之间的完整区别，请参阅比较 AL2 和 AL2023。对于迁移到 AL2023 后其特定软件包和版本要求不再得到满足的客户，我们建议联系 AWS 支持部门提交申请。

我可以选择不迁移吗？

不可以，您不能选择不迁移。AWS CloudShell 环境由 AWS 进行管理，因此，所有环境都已升级到 AL2023。

我能否为我的 AWS CloudShell 环境创建备份？

AWS CloudShell 将继续保留用户主目录。有关更多信息，请参阅 AWS CloudShell 服务配额和限额。如果您在主文件夹中存储了任何文件或配置，并且想要为其创建备份，请完成步骤 6：创建主目录备份。

故障排除 AWS CloudShell

使用时 AWS CloudShell，您可能会遇到问题，例如使用 shell 命令行界面启动 CloudShell 或执行关键任务时。本章介绍的信息涵盖如何解决您可能遇到的一些常见问题。

有关各种问题的答案 CloudShell，请参阅AWS CloudShell FAQs。您还可以在AWS CloudShell 论坛上搜索答案和发布问题。当您进入此论坛时，可能会要求您登录 AWS。您也可以直接联系我们。

错误故障排除

当您遇到以下任何索引错误时，可以使用以下解决方案来解决这些错误。

主题

拒绝访问

问题：当您尝试 CloudShell从启动时 AWS 管理控制台，会收到消息 “无法启动环境。要重试，请刷新浏览器或通过选择‘操作’、‘重新启动 AWS CloudShell’来重新启动”。即使您获得了 IAM 管理员的所需权限，并且您已刷新或重新 CloudShell启动浏览器，您仍会被拒绝访问。

解决方案：联系 AWS 支持部门。

（回到顶部）

权限不足

问题：当您尝试 CloudShell从启动时 AWS 管理控制台，会收到消息 “无法启动环境。您没有所需的权限。要求您的 IAM 管理员授予对 AWS CloudShell“的访问权限。系统会拒绝您访问并通知您没有所需的权限。

原因：您用于访问的 IAM 身份 AWS CloudShell 缺少必要的 IAM 权限。

解决方案：请求您的 IAM 管理员为您提供必要的权限。他们可以通过添加附加的 AWS 托管策略 (AWSCloudShellFullAccess) 或嵌入式内联策略来实现此目的。有关更多信息，请参阅 使用 IAM 策略管理 AWS CloudShell 访问和使用情况。

（回到顶部）

无法访问 AWS CloudShell 命令行

问题：修改计算环境使用的文件后，您无法在中访问命令行 AWS CloudShell。

解决方案：如果修改错误`.bashrc`或任何其他文件后失去访问权限，则可以通过删除主目录 AWS CloudShell来恢复其默认设置。

（回到顶部）

无法 ping 外部 IP 地址

问题：从命令行（例如，`ping amazon.com`）运行 ping 命令时，您会收到以下消息。

```
ping: socket: Operation not permitted
```

原因：ping 实用程序使用 Internet 控制消息协议 (ICMP) 向目标主机发送回显请求数据包。它会等待目标的回显回复。由于中未启用 ICMP 协议 AWS CloudShell，因此 ping 实用程序无法在外壳的计算环境中运行。

解决方案：由于中不支持 ICMP AWS CloudShell，您可以运行以下命令来安装 Netcat。Netcat 是一款计算机网络实用程序，用于使用 TCP 或 UDP 读取和写入网络连接。

```
sudo yum install nc
nc -zv www.amazon.com 443
```

（回到顶部）

准备您的终端时遇到了一些问题

问题：尝试 AWS CloudShell 使用 Microsoft Edge 浏览器进行访问时，你无法启动 shell 会话，并且浏览器会显示一条错误消息。

原因：与早期版本 AWS CloudShell 的 Microsoft Edge 不兼容。您可以使用支持的浏览器的最新四个主要版本进行访问 AWS CloudShell 。

解决方案：从 Microsoft 网站安装更新版本的 Edge 浏览器。

（回到顶部）

箭头键在中无法正常工作 PowerShell

问题：在正常操作中，您可以使用箭头键浏览命令行界面，并来回扫描命令历史记录。但是，在某些版本的 PowerShell on 中按箭头键时 AWS CloudShell，可能会错误地输出字母。

原因：在 Linux 上运行的 PowerShell 7.2.x 版本中，箭头键错误输出字母的情况是一个已知问题。

解决方案：要删除修改箭头键行为的转义序列，请编辑 PowerShell 配置文件并将`$PSStyle`变量设置为`PlainText`。

1. 在 AWS CloudShell 命令行中，输入以下命令以打开配置文件。

   ```
   vim ~/.config/powershell/Microsoft.PowerShell_profile.ps1
   ```

   > **Note**
   >
   > 如果您已经在编辑器中 PowerShell，也可以使用以下命令在编辑器中打开配置文件。
   >
   > ```
   > vim $PROFILE
   > ```

2. 在编辑器中，转到文件现有文本的末尾，按 i 进入插入模式，然后添加以下语句。

```
$PSStyle.OutputRendering = 'PlainText'
```

3. 编辑完成后，按下 Esc 进入命令模式。接下来，请输入以下命令保存文件，然后退出编辑器。

```
:wq
```

> **Note**
>
> 您的更改将在下次开始时生效 PowerShell。

（回到顶部）

不支持的 Web 套接字会导致无法启动 CloudShell 会话

问题：当你尝试启动时 AWS CloudShell，你会反复收到以下消息：`Failed to open sessions : Timed out while opening the session`。

原因: CloudShell 取决于WebSocket 协议, 它允许在您的Web浏览器和之间进行双向交互式通信 AWS CloudShell. 如果您在专用网络中使用浏览器，则代理服务器和防火墙可能有助于安全访问互联网。 WebSocket 通信通常可以毫无问题地通过代理服务器。但是，在某些情况下，代理服务器 WebSockets 无法正常工作。如果出现此问题，则 CloudShell 无法启动 shell 会话，连接尝试最终会超时。

解决方案：连接超时可能是由不支持 WebSockets以外的其他问题引起的。如果是这种情况，请先刷新 CloudShell 命令行界面所在的浏览器窗口。

如果刷新后仍然出现超时错误，请参阅代理服务器的文档。并且，请确保您的代理服务器已配置为允许 WebSocket。或者，请与您网络的系统管理员联系。

> **Note**
>
> 假设您想通过特定许可名单来定义精细权限。 URLs您可以添加 AWS Systems Manager 会话用来打开 WebSocket连接以发送输入和接收输出的部分 URL。您的 AWS CloudShell 命令将发送到该 Systems Manager 会话。
> Systems Manager 使用的格式是 `wss://ssmmessages.region.amazonaws.com/v1/data-channel/session-id?stream=(input|output)`。 StreamUrl

该区域表示支持的区域标 AWS 区域 识符 AWS Systems Manager。例如，us-east-2 是美国东部（俄亥俄州）区域的区域标识符。
由于会话 ID 是在特定 Systems Manager 会话成功启动后创建的，因此您只能在更新 URL 允许列表时指定 wss://ssmmessages.region.amazonaws.com。有关更多信息，请参阅 AWS Systems Manager API 参考中的StartSession操作。

（回到顶部）

无法导入 AWSPowerShell.NetCore 模块

问题：导入 AWSPower命令行管理程序时。 NetCore在 b PowerShell y 的模块中Import-Module -Name AWSPowerShell.NetCore，您会收到以下错误消息：

导入模块：指定的模块 “Shell” AWSPower。 NetCore' 未加载，因为在任何模块目录中都找不到有效的模块文件。

原因：该AWSPowerShell.NetCore模块已被中每个服务的 AWS.Tools 模块所取代。 AWS CloudShell

解决方案：可能不再需要任何显式导入语句，或者需要将其更改为相关的每项服务 AWS.Tools 模块。

Example

Example

- 在大多数情况下，只要不使用 .Net 类型，就不需要任何显式导入语句。以下是导入语句的示例。
  - Get-S3Bucket
  - (Get-EC2Instance).Instances
- 如果使用 .Net 类型，请导入服务级别模块 (AWS.Tools.<Service>)。以下是语法示例。

```
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()
```

有关更多信息，请参阅 AWS Tools for PowerShell的版本 4 公告。

（回到顶部）

使用 AWS CloudShell时 Docker 未运行

问题：使用 AWS CloudShell时 Docker 无法正常运行。您会收到以下错误：`docker: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?`。

解决方案：尝试重启您的环境。当您在某个 GovCloud区域中 AWS CloudShell 运行 Docker 时，可能会出现此错误消息。请确保您在支持的 AWS 区域中运行 Docker。有关可用 Docker 的区域列表，请参阅支持的 AWS 区域。 AWS CloudShell

Docker 的磁盘空间已耗尽

问题：您收到以下错误消息：`ERROR: failed to solve: failed to register layer: write [...]: no space left on device`。

原因：Dockerfile 超出了中的可用磁盘空间。 AWS CloudShell这可能是由于单个映像过大或预先存在的 Docker 映像过多造成的。

解决方案：运行 `df -h` 以查找磁盘使用情况。运行 `sudo du -sh /folder/folder1` 来评估某些您认为可能很大的文件夹的大小，并考虑删除其他文件以释放空间。一种选择是考虑通过运行 `docker rmi` 来删除未使用的 Docker 映像。您应该知道 Docker 在环境中的空间有限，有关 Docker 的更多信息，请参阅 Docker 文档指南。

`docker push` 超时并且一直在重试

问题：当您运行 `docker push` 时，它会超时，并且会继续重试，但没有成功。

原因：这可能是由于缺少权限、推送到错误的存储库或缺乏身份验证所致。

解决方案：要尝试解决此问题，请确保推送到正确的存储库。运行 `docker login` 以正确进行身份验证。确保您拥有推送到 Amazon ECR 存储库的所有必需权限。

无法从我的 VPC 环境中访问 V AWS CloudShell PC 内的资源

问题：使用我的 VPC 环境时无法访问 V AWS CloudShell PC 内的资源。

原因：您的 AWS CloudShell VPC 环境继承了 VPC 的网络设置。

解决方案：要解决此问题，请确保正确设置您的 VPC 以访问您的资源。有关更多信息，请参阅 VPC 文档将您的 VPC 连接到其他网络，以及网络访问分析器文档网络访问分析器。您可以通过在环境中的命令`ip -a`行提示符下或在 AWS CloudShell VPC 控制台页面上运行命令来找到 VPC 环境正在使用 IPv4 的地址。

AWS CloudShell 用于我的 VPC 环境的 ENI 未被清除

问题：无法清理 AWS CloudShell 用于我的 VPC 环境的 ENI。

原因：您的角色未启用 `ec2:DeleteNetworkInterface` 权限。

解决方案：要解决此问题，请确保您的角色已启用 `ec2:DeleteNetworkInterface` 权限，如以下脚本示例所示：

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

仅具有 VPC 环境**`CreateEnvironment`**权限的用户也可以访问公共 AWS CloudShell 环境

问题：仅获得 VPC 环境`CreateEnvironment`权限限制的用户也可以访问公共 AWS CloudShell 环境。

原因：如果您仅限制创建 VPC 环境的`CreateEnvironment`权限，并且您已经创建了公共环境，则在使用 Web 用户界面删除该 CloudShell环境之前，您将保留对现有公共环境的访问权限。但是，如果您 CloudShell 以前从未使用过，则将无法访问公共环境。

解决方案：要限制对公共 AWS CloudShell 环境的访问，IAM 管理员必须先使用限制更新 IAM 策略，然后用户必须使用 AWS CloudShell Web 用户界面手动删除现有的公共环境。（操作 → 删除 CloudShell 环境）。

AWS CloudShell 支持的 AWS 区域

本节介绍 AWS CloudShell 支持的 AWS 区域和选择加入区域的列表。有关 CloudShell 的 AWS 服务终端点和配额的列表，请参阅 Amazon Web Services 一般参考 中的AWS CloudShell页面。

以下是 CloudShell、Docker 和 CloudShell VPC 环境支持的 AWS 区域：

- 美国东部（俄亥俄州）
- 美国东部（弗吉尼亚州北部）
- 美国西部（加利福尼亚北部）
- 美国西部（俄勒冈州）
- 非洲（开普敦）
- 亚太地区（香港）
- 亚太地区（雅加达）
- 亚太地区（孟买）
- 亚太地区（大阪）
- 亚太地区（首尔）
- 亚太地区（新加坡）
- 亚太地区（悉尼）
- 亚太地区（东京）
- 加拿大（中部）
- 欧洲地区（法兰克福）
- 欧洲地区（爱尔兰）
- 欧洲地区（伦敦）
- 欧洲地区（米兰）
- 欧洲地区（巴黎）
- 欧洲地区（斯德哥尔摩）
- 中东（巴林）
- 中东（阿联酋）
- 南美洲（圣保罗）

GovCloud 区域

以下是 CloudShell 支持的 GovCloud 区域：

- AWS GovCloud (美国东部)
- AWS GovCloud（美国西部）

> **Note**
>
> Docker 和 CloudShell VPC 环境在 GovCloud 区域中可用。

AWS CloudShell 的服务配额和限额

本页介绍适用于以下领域的服务配额和限额：

持久性存储

使用 AWS CloudShell，您可在每个 AWS 区域中拥有最高 1GB 的持久性存储。持久性存储位于您的主目录 ($HOME) 中，对您而言是私有的。与每个 Shell 会话结束后回收的临时环境资源不同的是，主目录中的数据会在不同会话之间保留。

> **Note**
>
> CloudShell VPC 环境没有永久存储空间。当您的 VPC 环境超时（处于非活动状态 20-30 分钟之后）或者您删除环境时，$HOME 目录将被删除。

如果您停止在 AWS 区域 中使用 AWS CloudShell，则在上次会话结束后，数据将在该区域的持久性存储中保留 120 天。120 天后，除非您采取措施，否则您的数据将自动从该地区的持久性存储中删除。您可以通过在 AWS 区域 中再次启动 AWS CloudShell 来阻止删除。有关更多信息，请参阅步骤 2：选择区域，启动 AWS CloudShell 并选择 Shell。

> **Note**
>
> 使用场景
> Márcia 使用 AWS CloudShell 将文件存储在两个 AWS 区域 的主目录中：美国东部（弗吉尼亚州北部）和欧洲地区（爱尔兰）。然后，她开始只在欧洲地区（爱尔兰）使用 AWS CloudShell，并停止在美国东部（弗吉尼亚州北部）启动 Shell 会话。

在美国东部（弗吉尼亚州北部）删除数据的截止日期之前，Márcia 决定再次启动 AWS CloudShell 并选择美国东部（弗吉尼亚州北部）区域，以防止其主目录被回收。由于她一直使用欧洲地区（爱尔兰）进行 Shell 会话，因此她在该地区的持久性存储不会受到影响。

每月使用量

您的 AWS 区域中的每一个 AWS 账户都有 AWS CloudShell 月度使用限额。该限额汇总了该区域所有 IAM 主体使用 CloudShell 所花费的总时间。如果您在达到该区域的月度限额后尝试访问 CloudShell，则会显示一条消息，解释无法启动 Shell 环境的原因。

使用服务限额控制台请求提高限制

您可以通过打开服务配额控制台来申请增加每月使用配额。有关更多信息，请参阅《服务配额用户指南》中的 Requesting a quota increase。

并发 Shell

您最多可以为自己的账户在每个 AWS 区域中同时运行 10 个 Shell。

使用服务限额控制台请求提高限制

您可以通过打开服务配额控制台来申请增加每个区域的配额。有关更多信息，请参阅《服务配额用户指南》中的 Requesting a quota increase。

命令大小

命令大小不能超过 65412 个字符。

Note

如果您打算执行超过 65412 个字符的命令，请使用您选择的语言创建脚本，然后从命令行界面执行该脚本。有关可从命令行界面访问的预安装软件范围的更多信息，请参阅预安装的软件。要查看如何创建脚本然后从命令行界面执行脚本的示例，请参阅教程：AWS CloudShell 入门。

Shell 会话

- 非活动会话：AWS CloudShell是一个交互式 Shell 环境——如果您在 20—30 分钟内没有使用键盘或指针与其交互，则 Shell 会话将结束。正在运行的进程不算作交互。

  如果您希望使用 AWS 服务执行基于终端的任务，并且需要更灵活的超时设置，建议您启动并连接到一个 Amazon EC2 实例。
- 长时间运行的会话：持续运行大约 12 小时的 Shell 会话会自动结束，即使用户在此期间定期与其交互。

VPC 环境

对于每个 IAM 主体，最多只能创建两个 VPC 环境。

> **Note**
>
> 连接到您的私有 VPC 并访问其中的资源是免费的。您的私有 VPC 内部的数据传输已包含在您的 VPC 账单中，通过 CloudShell 在您的 VPC 之间传输数据的费用与您当前所用 CloudShell 的费用相同。

网络访问和数据传输

以下限制适用于您的 AWS CloudShell 环境的入站和出站通信：

- 出站：您可以访问公共互联网。
- 入站：您无法访问入站端口。公共 IP 地址不可用。

> **Warning**
>
> 访问公共互联网时，某些用户从 AWS CloudShell 环境中导出数据时可能会有风险。我们建议 IAM 管理员通过 IAM 工具管理受信任 AWS CloudShell 用户的允许列表。有关如何明确拒绝特定用户访问的信息，请参阅 AWS CloudShell 使用自定义策略管理允许的操作。

数据传输：对于大文件，将文件上传到 AWS CloudShell 或从中下载文件可能会很慢。或者，您可以使用 Shell 的命令行界面将文件从 Amazon S3 存储桶传输到您的环境。

对系统文件和页面重新加载的限制

- 系统文件：如果您错误地修改了计算环境所需的文件，则在访问或使用 AWS CloudShell 环境时可能会遇到问题。如果发生这种情况，您可能需要删除主目录才能重新获得访问权限。
- 重新加载页面：要重新加载 AWS CloudShell 界面，请使用浏览器中的刷新按钮，而不是操作系统的默认快捷键序列。

AWS CloudShell 用户指南的文档历史记录

最近的更新

下表介绍了《AWS CloudShell 用户指南》的重要更改。

变更	说明	日期
AWS CloudShell 中的 Amazon Q CLI	增加了对在 AWS CloudShell 中使用 Amazon Q CLI 功能的支持。	2024 年 10 月 2 日
Amazon VPC 在某些区域支持 AWS CloudShell	增加了对在某些区域创建和使用 AWS CloudShell VPC 环境的支持。	2024 年 6 月 13 日
新教程已添加到《AWS CloudShell 用户指南》中	新增了两个教程，详细介绍了如何在 AWS CloudShell 内部构建 Docker 容器并将其推送到 Amazon ECR 存储库，以及如何通过 AWS CDK 部署 Lambda 函数。	2023 年 12 月 27 日
在某些区域，AWS CloudShell 支持 Docker 容器	在某些区域已添加了 AWS CloudShell 对 Docker 容器的支持。	2023 年 12 月 27 日
AWS CloudShell 已迁移，现在使用 Amazon Linux 2023（AL2023）	AWS CloudShell 现在使用 AL2023 并已从 Amazon Linux 2 迁移。	2023 年 12 月 4 日
适用于 AWS CloudShell 的新 AWS 区域	AWS CloudShell 现在已在以下 AWS 区域正式发布： • 美国西部（加利福尼亚北部） • 非洲（开普敦） • 亚太地区（香港）	2023 年 6 月 16 日

	• 亚太地区（大阪） • 亚太地区（首尔） • 亚太地区（雅加达） • 亚太地区（新加坡） • 欧洲地区（巴黎） • 欧洲地区（斯德哥尔摩） • 欧洲地区（米兰） • 中东（巴林） • 中东（阿联酋）	
在 AWS CloudShell 上启动 Console Toolbar	在 Console Toolbar 上启动 CloudShell（方法是在控制台左下角选择 CloudShell）。	2023 年 3 月 28 日
适用于 AWS CloudShell 的新 AWS 区域	现在以下 AWS 区域提供 AWS CloudShell： • 加拿大（中部） • 欧洲地区（伦敦） • 南美洲（圣保罗）	2022 年 10 月 6 日
AWS CloudShell美国 AWS GovCloud 支持的	AWS GovCloud（美国）区域现在支持 AWS CloudShell。	2022 年 6 月 29 日
安全常见问题解答	其他以安全问题为重点的常见问题解答。	2022 年 4 月 14 日
Web 套接字	在网络要求中添加了一节，解释了 CloudShell 对 WebSocket 协议的使用。	2022 年 3 月 21 日
排除 PowerShell 中的箭头键故障	按照步骤修复按下箭头键时错误输出字母的问题。	2022 年 2 月 7 日

Tab 键自动完成	解释如何使用 bash-completion 的新文档，它允许通过按 Tab 键自动完成部分键入的命令或参数。	2021 年 9 月 24 日
指定 AWS 区域	有关为 AWS CLI 命令指定默认 AWS 区域 的文档。	2021 年 5 月 11 日
在 PDF 和 Kindle 版本中进行格式化	修复了表格单元格中的图像大小和文本。	2021 年 3 月 10 日
在选定 AWS 区域正式发布 (GA) AWS CloudShell 版本	AWS CloudShell 现在已在以下 AWS 区域正式发布： • 美国东部（俄亥俄州） • 美国东部（弗吉尼亚州北部） • 美国西部（俄勒冈） • 亚太地区（东京） • 欧洲地区（爱尔兰） • 亚太地区（孟买） • 亚太地区（悉尼） • 欧洲地区（法兰克福）	2020 年 12 月 15 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。