Guía del usuario

AWS CloudShell

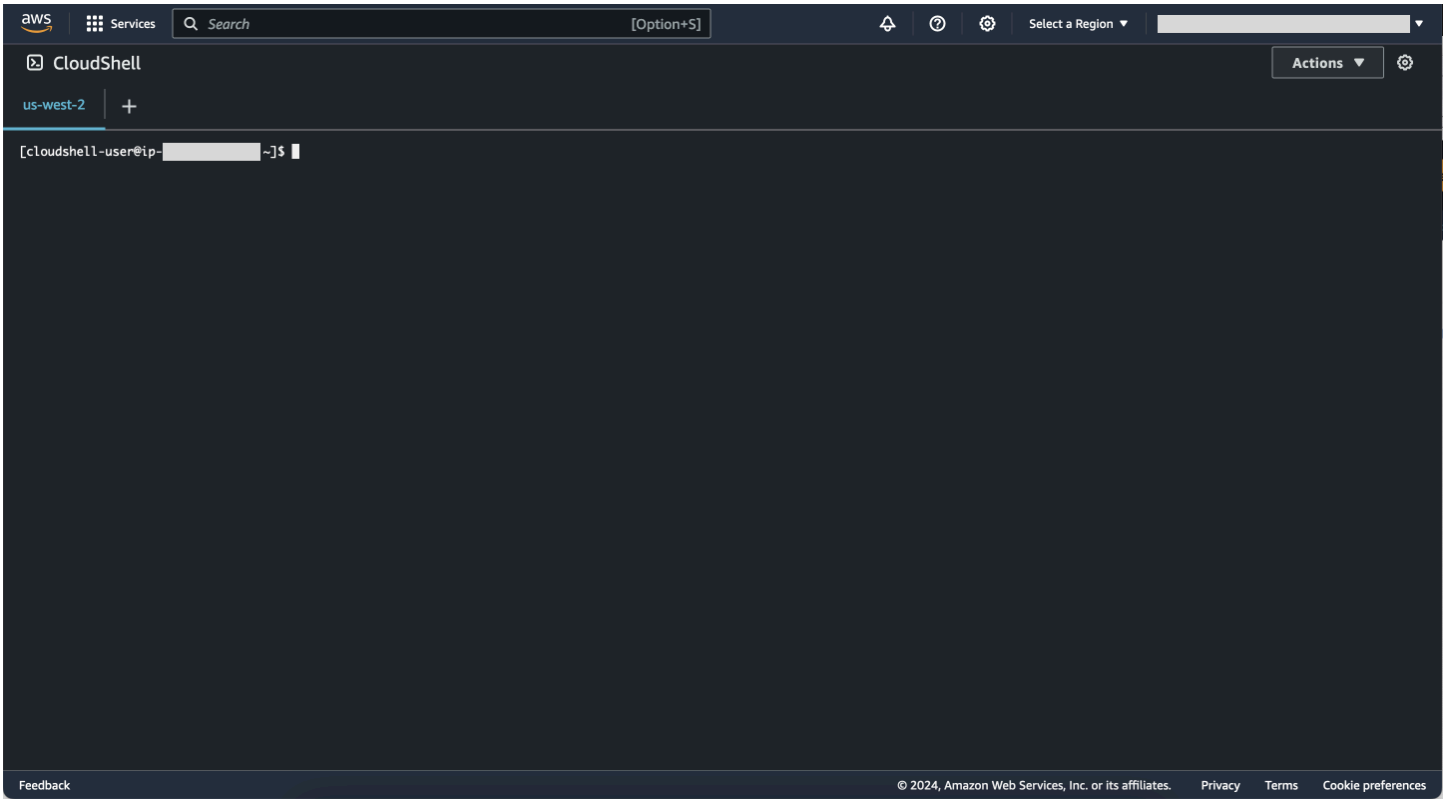
AWS CloudShell: Guía del usuario

Table of Contents

¿Qué es AWS CloudShell?

AWS CloudShell es un shell preautenticado y basado en un navegador que se puede iniciar directamente desde. AWS Management Console Puede navegar CloudShell desde varias formas diferentes AWS Management Console . Para obtener más información, consulte ¿Cómo empezar a usar AWS CloudShell?

Puede ejecutar AWS CLI comandos mediante el shell que prefiera, como Bash PowerShell, o Z shell. Y puede hacerlo sin descargar ni instalar herramientas de línea de comandos.



Cuando se lanza AWS CloudShell, se crea un entorno informático basado en Amazon Linux 2023. En este entorno, puede acceder a una amplia gama de herramientas de desarrollo preinstaladas, opciones para cargar y descargar archivos y al almacenamiento de archivos que persiste entre sesiones.

(Pruébelo ahora: Empezar con AWS CloudShell)

Características de AWS CloudShell

AWS CloudShell ofrece las siguientes características:

AWS Command Line Interface

Puede iniciar AWS CloudShell desde AWS Management Console. Las AWS credenciales que utilizó para iniciar sesión en la consola están disponibles automáticamente en una nueva sesión de shell. Como AWS CloudShell los usuarios están preautenticados, no es necesario configurar las credenciales al interactuar Servicios de AWS con la AWS CLI versión 2. Viene AWS CLI preinstalado en el entorno informático del shell.

Para obtener más información sobre cómo interactuar Servicios de AWS con la interfaz de línea de comandos, consulteTrabajar con AWS servicios en AWS CloudShell.

Intérprete de comandos y herramientas de desarrollo

Con el shell creado para AWS CloudShell las sesiones, puedes cambiar sin problemas entre los shell de línea de comandos que prefieras. Más específicamente, puedes cambiar entre Bash, PowerShell, y Z shell. También tiene acceso a las herramientas y utilidades preinstaladas. Estas incluyen git, make, pip, sudo, tar, tmux, vim, wget, y zip.

El entorno de shell está preconfigurado y es compatible con varios de los principales lenguajes de software, como Node.js y Python. Esto significa que, por ejemplo, puedes correr Node.js y Python proyectos sin realizar primero las instalaciones en tiempo de ejecución. PowerShell los usuarios pueden usar el .NET Core tiempo de ejecución.

Puedes confirmar los archivos que se crean o se AWS CloudShell cargan en un repositorio local antes de enviarlos a un repositorio remoto gestionado por AWS CodeCommit.

Para obtener más información, consulte AWS CloudShell entorno informático: especificaciones y software.

Almacenamiento persistente

Con AWS CloudShell, puedes usar hasta 1 GB de almacenamiento persistente en cada uno sin Región de AWS coste adicional. El almacenamiento persistente se encuentra en su directorio principal (`$HOME`) y es privado para usted. A diferencia de los recursos efímeros del entorno que se reciclan al finalizar cada sesión del intérprete de comandos, los datos del directorio principal persisten entre las sesiones.

Para obtener más información acerca de la retención de datos en el almacenamiento persistente, consulte Almacenamiento persistente.

> **Note**
>
> CloudShell VPClos entornos no tienen almacenamiento persistente. El HOME directorio $ se elimina cuando se agota el tiempo de espera del VPC entorno (tras 20 a 30 minutos de inactividad) o cuando se elimina o reinicia el entorno.

CloudShell VPCentornos

AWS CloudShell la nube privada virtual (VPC) le permite crear un CloudShell entorno en suVPC. Para cada VPC entorno, puede asignar unaVPC, añadir una subred y asociar uno o más grupos de seguridad. AWS CloudShell hereda la configuración de red del VPC y le permite utilizarlos de AWS CloudShell forma segura dentro de la misma subred que otros recursos del. VPC

Seguridad

El AWS CloudShell entorno y sus usuarios están protegidos por funciones de seguridad específicas. Esto incluye funciones como la administración de IAM permisos, las restricciones de sesión de shell y el pegado seguro para la entrada de texto.

Gestión de permisos con IAM

Como administrador, puede conceder y denegar permisos a AWS CloudShell los usuarios mediante IAM políticas. También puede crear políticas que especifiquen las acciones concretas que los usuarios pueden realizar en el entorno del intérprete de comandos. Para obtener más información, consulte Administrar el AWS CloudShell acceso y el uso con políticas IAM.

Administración de sesiones del intérprete de comandos

Las sesiones inactivas y de larga duración se detienen y reciclan automáticamente. Para obtener más información, consulte Sesiones del intérprete de comandos.

Pegado seguro para introducir texto

La opción de pegado seguro está habilitada de manera predeterminada. Esta característica de seguridad requiere que compruebe que el texto multilínea que desea pegar en el intérprete de comandos no contiene scripts maliciosos. Para obtener más información, consulte Uso de pegado seguro para texto de líneas múltiples.

Opciones de personalización

Puede personalizar su AWS CloudShell experiencia según sus preferencias exactas. Por ejemplo, puede cambiar el diseño de las pantallas (varias pestañas), los tamaños de los textos mostrados y alternar entre los temas de la interfaz claros y oscuros. Para obtener más información, consulte Personalización de tu experiencia AWS CloudShell.

También puede ampliar su entorno de shell instalando su propio software y modificando su shell con scripts.

Restauración de sesión

La funcionalidad de restauración de sesiones restaura las sesiones que estaba ejecutando en una o varias pestañas del navegador del CloudShell terminal. Si actualiza o vuelve a abrir las pestañas del navegador cerradas recientemente, esta funcionalidad reanuda la sesión hasta que el intérprete de comandos se detenga debido a una sesión inactiva. Para seguir utilizando la CloudShell sesión, pulse cualquier tecla de la ventana del terminal. Para obtener más información sobre las sesiones de intérprete de comandos, consulte Sesiones de intérprete de comandos.

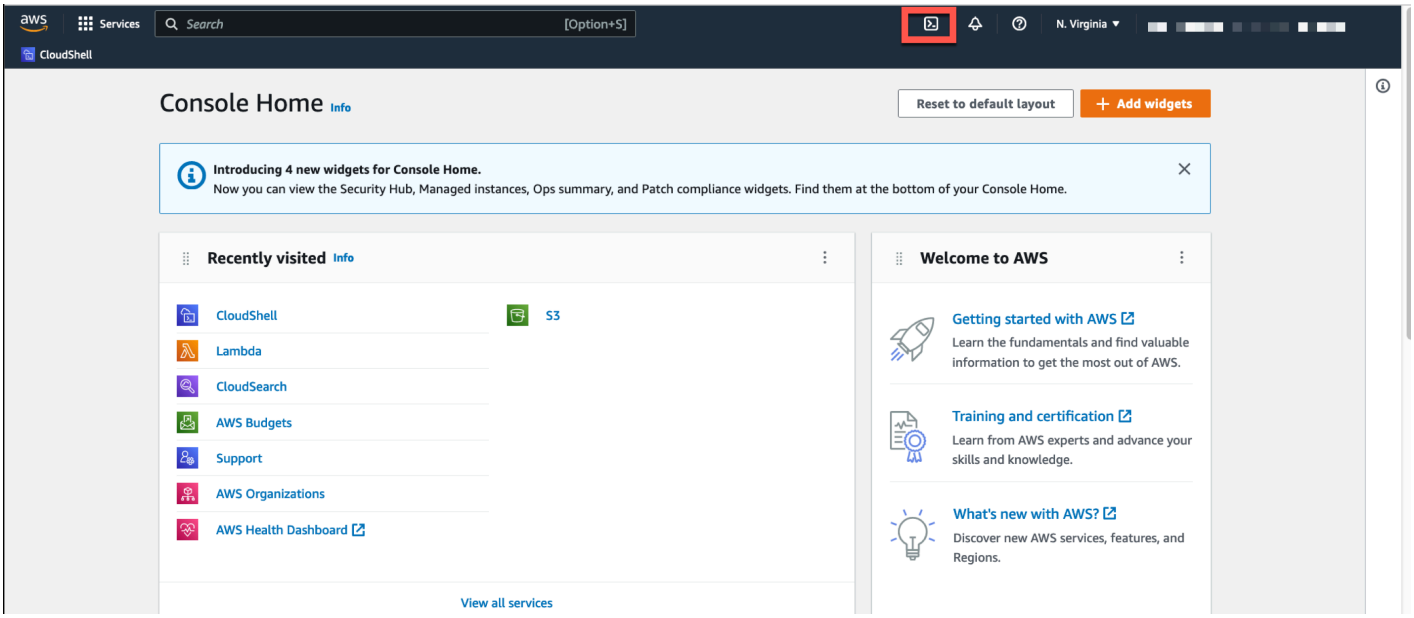
La restauración de sesiones también restaura la última salida del terminal y los procesos en ejecución en cada pestaña de terminal.

> **Note**
>
> La restauración de sesiones no está disponible en las aplicaciones móviles.

¿Cómo empezar AWS CloudShell?

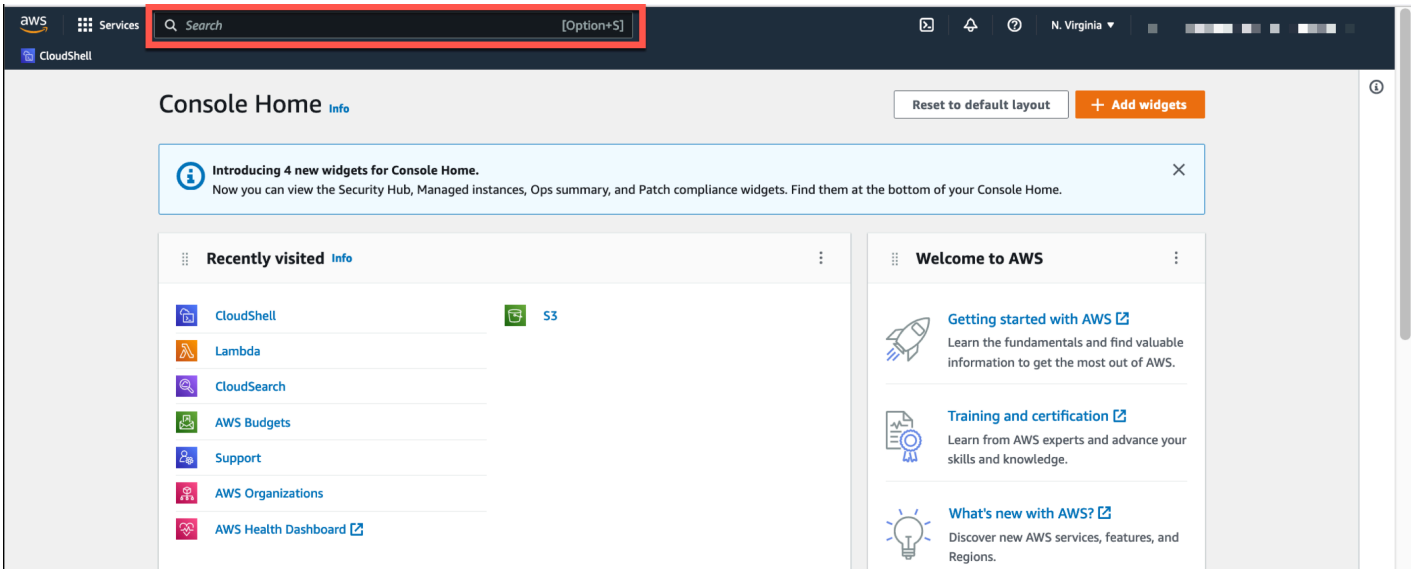
Para empezar a trabajar con el shell, inicie sesión en AWS Management Console y elija una de las siguientes opciones:

- En la barra de navegación, selecciona el CloudShellicono.



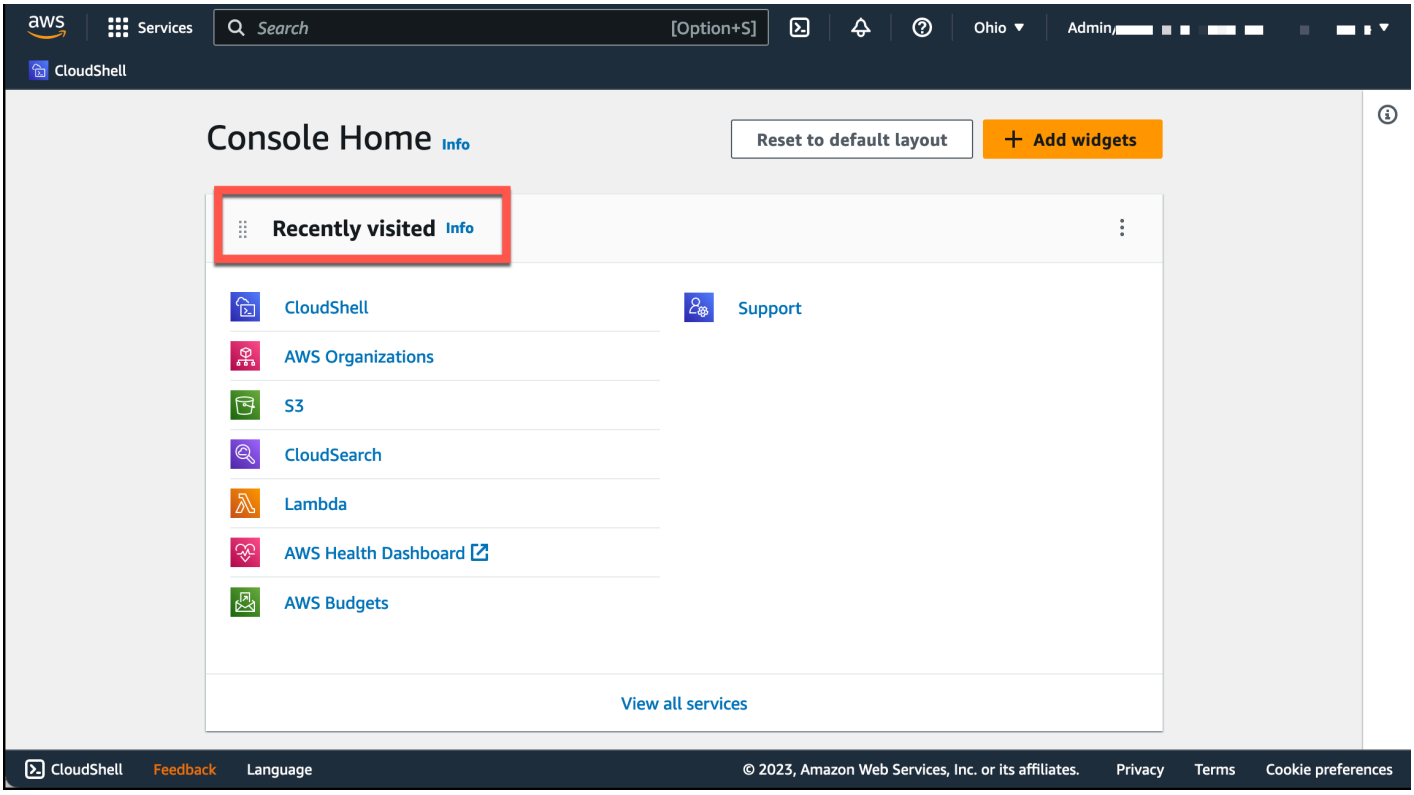
- En el cuadro de búsqueda, escriba «CloudShell» y, a continuación, elija CloudShell.

  Este paso abre la CloudShell sesión en pantalla completa.

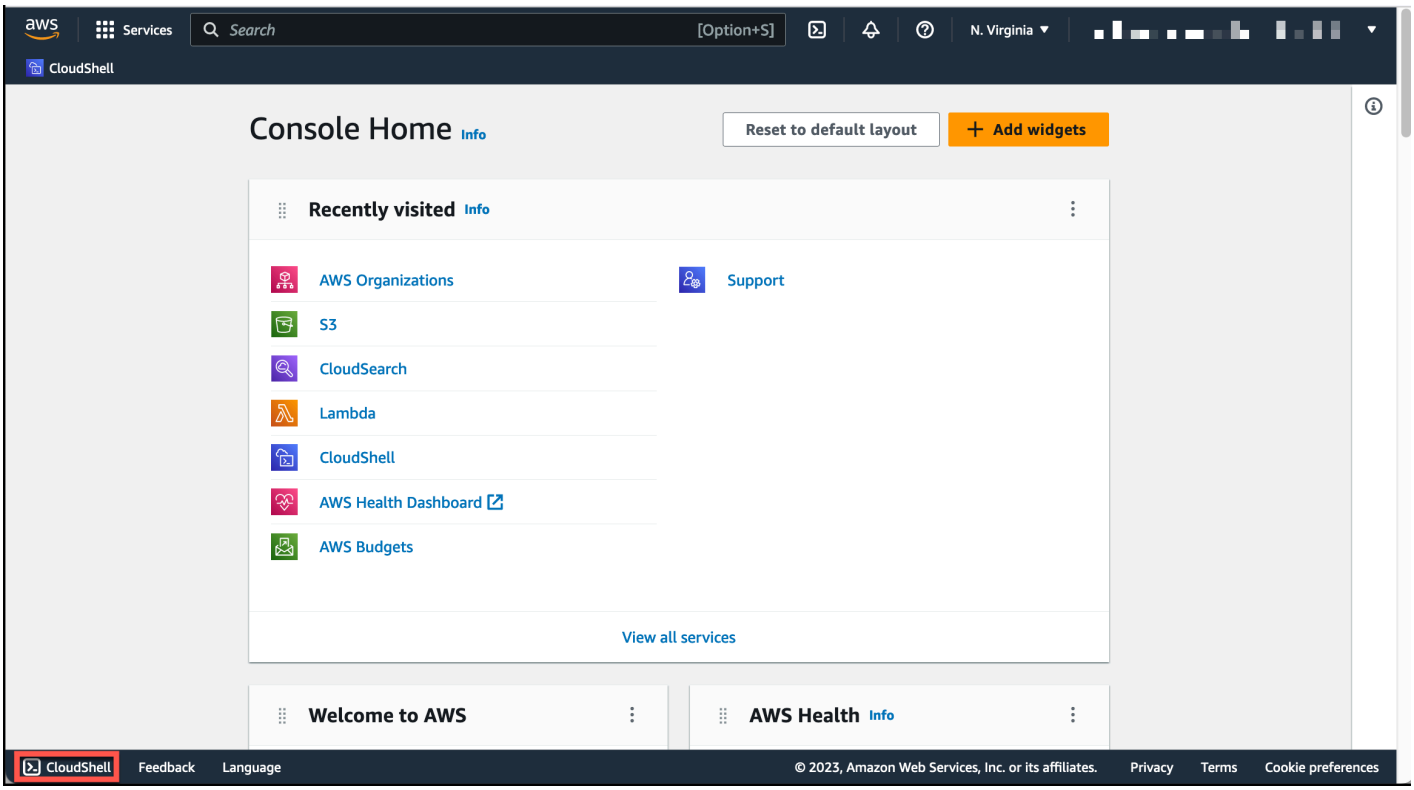


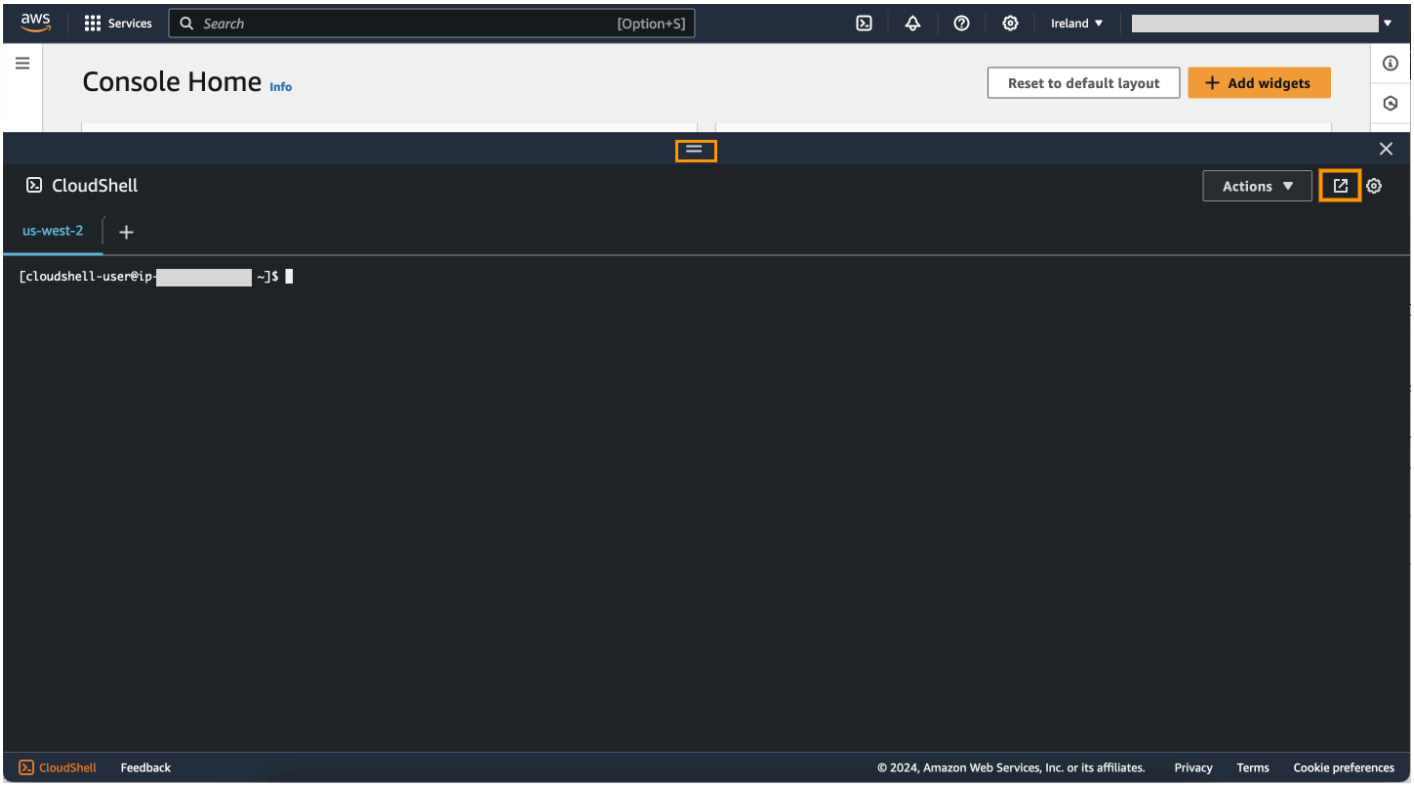
- En el widget Visitado recientemente, selecciona CloudShell.

  Este paso abre la CloudShell sesión en pantalla completa.



- Elige una CloudShellde las Console Toolbar, en la parte inferior izquierda de la consola. Puedes ajustar la altura de la CloudShell sesión arrastrándola=.





También puede cambiar la CloudShell sesión a pantalla completa haciendo clic en Abrir en una nueva pestaña del navegador.

Para obtener instrucciones sobre cómo iniciar sesión AWS Management Console y realizar las tareas clave con ella AWS CloudShell, consulte Primeros pasos con AWS CloudShell.

Precios para AWS CloudShell

AWS CloudShell es uno Servicio de AWS que está disponible sin cargo adicional. Sin embargo, pagas por otros AWS recursos con los que trabajas AWS CloudShell. Además, también se aplican las tarifas de transferencia de datos estándar. Para más información, consulte Precios de AWS CloudShell.

Para obtener más información, consulte Cuotas y restricciones de servicio para AWS CloudShell.

AWS CloudShell Temas clave

- Empezar con AWS CloudShell
- Trabajando con AWS CloudShell

- Trabajar con AWS servicios en AWS CloudShell
- Personalización de tu experiencia AWS CloudShell
- AWS CloudShell entorno informático: especificaciones y software

AWS CloudShell FAQs

Las siguientes son respuestas a algunas preguntas comunes sobre AWS CloudShell.

Para obtener más FAQs información sobre la seguridad, consulteAWS CloudShell Seguridad FAQs.

- ¿Cómo puedo empezar a usarlo AWS CloudShell?
- ¿A qué necesito acceder AWS CloudShell?
- ¿Qué hay AWS CloudShell en el Console Toolbar?
- ¿Cómo puedo realizar AWS CloudShell el lanzamiento en Console Toolbar?
- ¿Cómo creo y administro mi AWS CloudShell entorno?
- ¿Cuál Regiones de AWS está AWS CloudShell disponible en?
- ¿ Región de AWS Cuál se asigna si AWS CloudShell no está disponible en la región seleccionada cuando se lanza CloudShell en Console Toolbar?
- ¿Qué tipos de intérprete de comandos puedo utilizar en AWS CloudShell?
- ¿Con qué navegadores web puedo usar AWS CloudShell?
- ¿Qué navegadores web puedo usar al iniciar AWS CloudShell el Console Toolbar?
- ¿Puedo descargar un archivo al AWS CloudShell iniciarlo en Console Toolbar?
- ¿Qué software viene preinstalado en mi entorno de intérprete de comandos?
- ¿Puedo instalar software que no esté disponible en el entorno del intérprete de comandos?
- ¿Puedo restringir las acciones que pueden realizar los usuarios en AWS CloudShell?
- ¿Cómo puedo mover los datos de mi directorio principal si quiero cambiar el Región de AWS lugar donde los estoy usando AWS CloudShell?
- ¿Puedo aumentar el límite que determina cuando se agota el tiempo de espera de AWS CloudShell debido a la inactividad del usuario?
- ¿Puedo acceder AWS CloudShellAWS Console Mobile Application desde la pantalla de inicio?
- ¿Cómo puedo iniciar AWS CloudShell en el AWS Console Mobile Application?
- ¿Puedo usar las teclas modificadoras en mi teclado IOS y en el de Android cuando las uso AWS CloudShell en el? AWS Console Mobile Application

- ¿Puedo dividir la visualización de AWS CloudShell pestañas en varias pestañas? AWS Console Mobile Application
- ¿Puedo acceder a AWS CloudShell la barra de herramientas de la consola desde un dispositivo móvil?
- ¿Cuáles son los costes que CloudShell repercuten en mi AmazonVPC?
- ¿Puedo crear más de dos VPC entornos por IAM director?

¿Cómo puedo empezar a usarlo AWS CloudShell?

Puedes empezar lanzándolo AWS CloudShell en unos pocos pasos desde el AWS Management Console. Para ello, inicia sesión en la consola con tus IAM credenciales Cuenta de AWS o las que tengas en https://console.aws.amazon.com/console/casa.

Para obtener más información, consulte Introducción a AWS CloudShell.

¿A qué necesito acceder AWS CloudShell?

Como accede AWS CloudShell desde el AWS Management Console, debe ser un IAM usuario que pueda proporcionar un alias o ID de cuenta, un nombre de usuario y una contraseña válidos.

Para iniciar AWS CloudShell en la consola, necesita los IAM permisos que proporciona la política adjunta. Para obtener más información, consulte Administrar el AWS CloudShell acceso y el uso con políticas IAM.

¿Qué hay AWS CloudShell en Console Toolbar?

El CloudShell icono de la parte inferior izquierda del AWS Management Console.

¿Cómo se inicia AWS CloudShell en el Console Toolbar?

Puede lanzarse AWS CloudShell en el Console Toolbar seleccionando el CloudShellicono de la parte inferior izquierda de la consola.

¿Cuál Regiones de AWS está AWS CloudShell disponible en?

Para obtener una lista de los puntos finales de servicio compatibles Regiones de AWS y asociados, consulte la AWS CloudShell página del Referencia general de Amazon Web Services.

Cuál Región de AWS se asigna si no AWS CloudShell está disponible en la región seleccionada cuando se inicia CloudShell en el Console Toolbar?

La región predeterminada se asigna a la región más cercana a la región seleccionada. Para obtener más información, consulta Seleccionar una región AWS CloudShell, lanzar y elegir un proyectil.

Puede ejecutar el comando que proporciona permisos para administrar los recursos en una región diferente a la región predeterminada. Para obtener más información, consulte Trabajar en Regiones de AWS.

¿Qué tipos de intérprete de comandos puedo utilizar en AWS CloudShell?

En AWS CloudShell, puede ejecutar comandos mediante el Bash shell, PowerShell, o el Z shell. Para cambiar de consola, introduzca el nombre de la consola que desee utilizar con el siguiente formato en la línea de comandos:

- `bash`: Usa el Bash shell
- `pwsh`: Utilice PowerShell
- `zsh`: Usa el Z shell

¿Con qué navegadores web puedo usar AWS CloudShell?

AWS CloudShell es compatible con las versiones más recientes de los navegadores Google Chrome, Mozilla Firefox, Microsoft Edge y Apple Safari.

¿Cómo creo y administro mi AWS CloudShell entorno?

Su AWS CloudShell entorno se crea y administra por ID IAM de usuario y región. Puede comprobarlo `UserId` ejecutando`aws sts get-caller-identity`. El entorno es propiedad del ID IAM de usuario de esa región específica. Podrá acceder a un AWS CloudShell entorno diferente si cambia de región IAM `UserId` o región.

¿Qué navegadores web puedo usar al iniciar AWS CloudShell el Console Toolbar?

Se puede iniciar CloudShell en el Console Toolbar utilizando las versiones más recientes de los navegadores Google Chrome, Microsoft Edge, Mozilla Firefox y Apple Safari.

¿Puedo descargar archivos desde AWS CloudShell?

Sí, puedes descargar un archivo al iniciarlo CloudShell en Console Toolbar o desde la página de la CloudShell consola mediante un navegador. Puedes descargar un archivo con las versiones más recientes de los navegadores Google Chrome y Microsoft Edge.

Actualmente, no puedes descargar un archivo con los navegadores Mozilla Firefox y Apple Safari.

> **Note**
>
> La opción de descarga de archivos no está disponible para AWS CloudShell VPC los entornos.

¿Qué software viene preinstalado en mi entorno de intérprete de comandos?

Con la consola creada para AWS CloudShell las sesiones, puedes cambiar sin problemas entre las capas de línea de comandos que prefieras (Bash, y PowerShell Z shell). También puede tener acceso a herramientas y utilidades preinstaladas, como Make, pip, sudo, tar, tmux, Vim, Wget y Zip.

El entorno del intérprete de comandos está preconfigurado y es compatible con la mayoría de los principales lenguajes de software. Por ejemplo, puede usarlo para ejecutar Node.js y Python proyectos sin tener que realizar primero instalaciones en tiempo de ejecución. PowerShell los usuarios pueden usar el .NET Core tiempo de ejecución.

Puede añadir los archivos que se crearon con el shell o que se cargaron con la interfaz del shell a un repositorio controlado por versiones gestionado mediante una versión preinstalada de git.

Para obtener más información, consulte Software preinstalado.

¿Puedo instalar software que no esté disponible en el entorno del intérprete de comandos?

Sí, los usuarios tienen AWS CloudShell sudo privilegios y pueden instalar software desde la línea de comandos. Para obtener más información, consulte Instalación de software de terceros en el entorno del intérprete de comandos.

¿Puedo restringir las acciones que pueden realizar los usuarios en AWS CloudShell?

Sí, puede controlar las acciones que pueden realizar los usuarios en AWS CloudShell. Por ejemplo, puede permitir el acceso de los usuarios, AWS CloudShell pero impedir que suban o descarguen archivos dentro del entorno de shell. O, como alternativa, puede impedir por completo el acceso a AWS CloudShell de los usuarios. Para obtener más información, consulte Administrar el AWS CloudShell acceso y el uso con políticas IAM.

¿Cómo puedo mover los datos de mi directorio principal si quiero cambiar el Región de AWS lugar donde los estoy usando AWS CloudShell?

Para mover AWS CloudShell los datos de una región Región de AWS a otra, descargue primero el contenido del directorio principal de una región a su máquina local y, a continuación, cárguelo al directorio principal de otra región. Para obtener más información, consulte ???.

> **Note**
>
> Las opciones de carga y descarga no están disponibles para AWS CloudShell VPC los entornos.

¿Puedo aumentar el límite que determina cuando se agota el tiempo de espera de AWS CloudShell debido a la inactividad del usuario?

Si no interactúas con AWS CloudShell el teclado o el puntero, tu sesión temporal finaliza automáticamente después de aproximadamente 20 o 30 minutos. Los procesos en ejecución no cuentan como interacciones. Debido a que CloudShell está diseñada para actividades específicas y basadas en tareas, por el momento no hay planes para aumentar este límite de tiempo de espera.

Si quieres realizar tareas basadas en terminales Servicio de AWS con tiempos de espera más flexibles, te recomendamos que utilices nuestra instancia basada en la nube o que lances una IDE instancia AWS Cloud9de Amazon y te conectes a ella. EC2

¿Puedo acceder AWS Console Mobile Application desde AWS CloudShell la pantalla de inicio?

Sí, puede acceder al AWS Console Mobile Application iniciando sesión AWS CloudShell en la aplicación Console Mobile Application. Si quiere obtener más información, consulte la Guía del usuario de AWS Console Mobile Application.

¿Cómo puedo iniciar AWS CloudShell el AWS Console Mobile Application?

Puede iniciarlo AWS CloudShell mediante uno de los siguientes métodos:

1. Seleccione el icono AWS CloudShell de la parte inferior de la barra de navegación.
2. Seleccione AWS CloudShell en el menú Servicios.

> **Note**
>
> Actualmente, no se pueden crear ni lanzar VPC entornos en AWS Console Mobile Application.

¿Puedo usar las teclas modificadoras en mis teclados iOS y Android cuando las uso AWS CloudShell en el? AWS Console Mobile Application

Sí, puede usar las teclas modificadoras en los teclados de iOS y Android. Para obtener más información, consulte la Guía del usuario de la aplicación AWS Console Mobile Application.

¿Puedo dividir la AWS CloudShell pantalla de pestañas en varias pestañas AWS Console Mobile Application?

No, actualmente no puedes ejecutar varias AWS CloudShell pestañas en tu aplicación móvil.

¿Puedo acceder AWS CloudShell a través del Console Toolbar ¿en un dispositivo móvil?

No, actualmente no puedes acceder AWS CloudShell en el Console Toolbar en tu dispositivo móvil.

¿Cuáles son los costes que CloudShell repercuten en mi AmazonVPC?

Conectarse a tu red privada VPC y acceder a los recursos que contiene es gratuita. Las transferencias de datos dentro de tu cuenta privada VPC están incluidas en tu VPC facturación, y las transferencias de datos entre tú y tú CloudShell se cobran al mismo coste que las actuales CloudShell. VPCs

¿Puedo crear más de dos VPC entornos por IAM director?

No. Solo puedes crear un máximo de dos VPC entornos.

Empezar con AWS CloudShell

En este tutorial introductorio se muestra cómo iniciar AWS CloudShell y realizar tareas clave mediante la interfaz de línea de comandos del shell.

Primero, inicia sesión en AWS Management Console y selecciona una Región de AWS. Luego, abre CloudShell una nueva ventana del navegador y un tipo de shell con el que trabajar.

A continuación, crea una nueva carpeta en su directorio principal y carga un archivo en ella desde su máquina local. Trabaja en ese archivo con un editor preinstalado antes de ejecutarlo como un programa desde la línea de comandos. Por último, debe AWS CLI ejecutar comandos para crear un bucket de Amazon S3 y añadir su archivo como objeto al bucket.

Requisitos previos

IAMpermisos

Puede obtener permisos AWS CloudShell adjuntando la siguiente política AWS administrada a su IAM identidad (por ejemplo, un usuario, un rol o un grupo):

- AWSCloudShellFullAccess: Proporciona a los usuarios acceso completo a sus funciones AWS CloudShell y a sus funciones.

En este tutorial, también interactúa con Servicios de AWS. Más específicamente, se interactúa con Amazon S3 creando un bucket de S3 y añadiendo un objeto a ese bucket. Tu IAM identidad requiere una política que otorgue, como mínimo, los `s3:PutObject` permisos `s3:CreateBucket` y.

Para obtener más información, consulte Acciones de Amazon S3 en la Guía del usuario de Amazon Simple Storage Service.

Archivo de ejercicios

Este ejercicio también implica cargar y editar un archivo que, a continuación, se ejecuta como un programa desde la interfaz de la línea de comandos. Abra un editor de texto en su equipo local y agregue el siguiente fragmento de código.

```
import sys
x=int(sys.argv[1])
```

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

Guarde el archivo con el nombre `add_prog.py`.

Contenido

Paso 1: inicie sesión en AWS Management Console

Este paso implica ingresar su información IAM de usuario para acceder al AWS Management Console. Si ya está en la consola, vaya al paso 2.

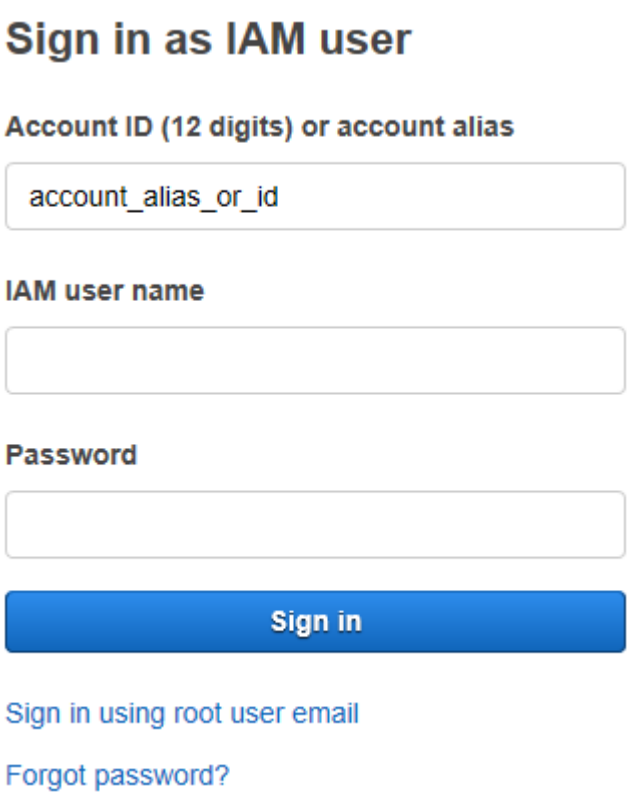
- Puede acceder a él AWS Management Console mediante el inicio de sesión de un IAM usuario URL o accediendo a la página principal de inicio de sesión.

  IAM user sign-in URL

  - Abre un navegador e introduce el siguiente inicio de sesión. URL Sustituya `account_alias_or_id` por el alias o el ID de cuenta que haya proporcionado el administrador.

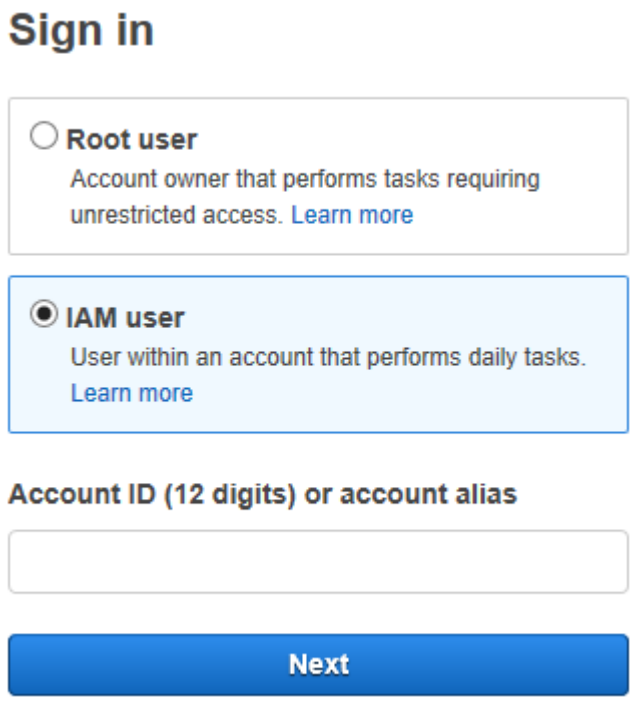
    ```
    https://account_alias_or_id.signin.aws.amazon.com/console/
    ```

  - Introduce tus credenciales IAM de inicio de sesión y selecciona Iniciar sesión.

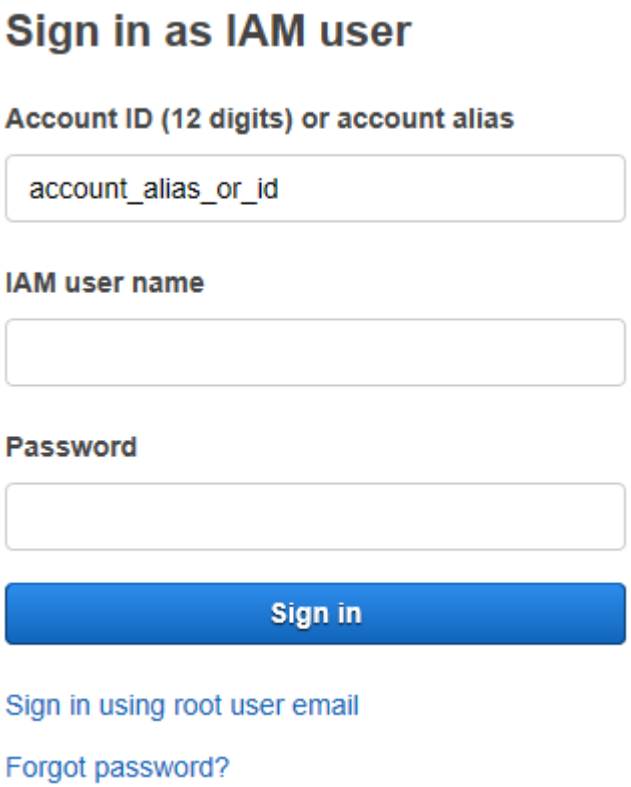


Main sign-in page

- Abrir https://aws.amazon.com/console/.
- Si no ha iniciado sesión anteriormente en este navegador, aparecerá la página principal de inicio de sesión. Elija el IAM usuario, introduzca el alias o el ID de la cuenta y pulse Siguiente.



- Si ya has iniciado sesión como IAM usuario anteriormente. Es posible que el navegador recuerde el alias o el ID de la cuenta de Cuenta de AWS. Si es así, introduce tus credenciales IAM de inicio de sesión y selecciona Iniciar sesión.



> **Note**
>
> También puede iniciar sesión como usuario raíz. Esta identidad tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Se recomienda encarecidamente no utilizar el usuario raíz para las tareas cotidianas, ni siquiera para las tareas administrativas. En su lugar, sigue la práctica recomendada de usar el usuario root solo para crear tu primer IAM usuario.

Paso 2: selecciona una región AWS CloudShell, lanza y elige un shell

En este paso, lo ejecutas AWS CloudShell desde la interfaz de la consola, eliges una disponible Región de AWS y cambias a la consola que prefieras, como Bash PowerShell, o Z shell.

1. Para elegir una región en la Región de AWS que trabajar, ve al menú Selecciona una región y selecciona una AWS región compatible en la que trabajar. (Las regiones disponibles aparecen resaltadas).

> **Important**
>
> Si cambia de región, la interfaz se actualiza y el nombre de la Región de AWS seleccionada aparece sobre el texto de la línea de comandos. Todos los archivos que añada al almacenamiento persistente solo estarán disponibles en esta misma Región de AWS. Si cambia de región, podrá acceder a diferentes archivos y almacenamiento.

> **Important**
>
> No CloudShell está disponible en la región seleccionada cuando lo CloudShell lanzas en Console Toolbar, en la parte inferior izquierda de la consola, la región predeterminada se establece en la región más cercana a la región seleccionada. Puede ejecutar el comando que proporciona permisos para administrar los recursos en una región diferente a la región predeterminada. Para obtener más información, consulte Trabajar en Regiones de AWS.

Example

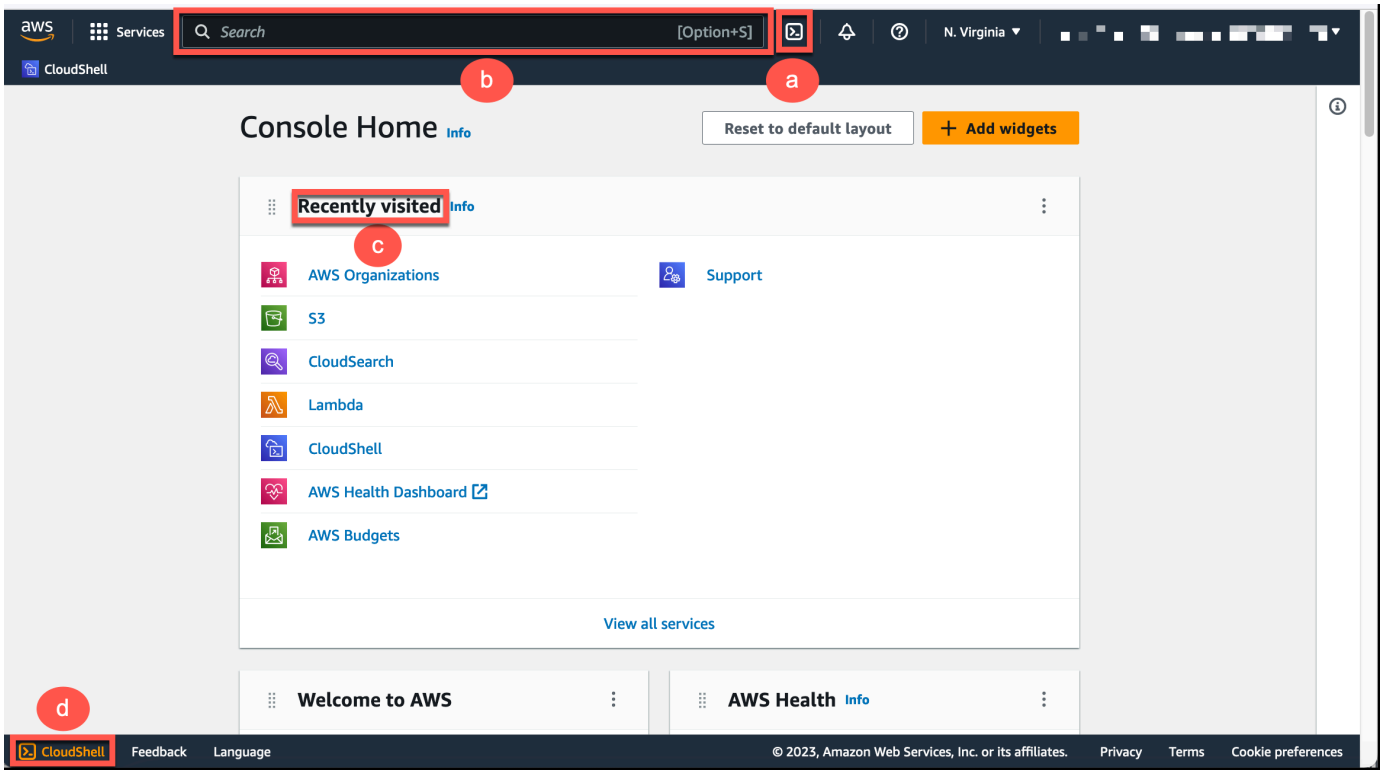
Ejemplo

Si eliges Europa (España) eu-south-2 pero CloudShell no está disponible en Europa (España) eu-south-2, entonces la región predeterminada se establece en Europa (Irlanda) eu-west-1, que está más cerca de Europa (España) eu-south-2.

Utilizará las cuotas de servicio de la región predeterminada, Europa (Irlanda) eu-west-1 y se restablecerá la misma CloudShell sesión en todas las regiones. Es posible que se cambie la región predeterminada y se le notificará en la ventana CloudShell del navegador.

2. Desde el AWS Management Console, puede iniciar CloudShell seleccionando una de las siguientes opciones:

   1. En la barra de navegación, selecciona el CloudShellicono.
   2. En el cuadro de búsqueda, escribe «CloudShell» y, a continuación, selecciona CloudShell.
   3. En el widget Visitados recientemente, selecciona CloudShell.
   4. Elige una CloudShellde las Console Toolbar, en la parte inferior izquierda de la consola.
      - Para ajustar la altura de la CloudShell sesión, arrastre=.
      - Para cambiar la CloudShell sesión a pantalla completa, haga clic en el icono Abrir en una nueva pestaña del navegador.



Cuando aparece el símbolo del sistema, el shell está listo para la interacción.

> **Note**
>
> Si encuentra problemas que le impiden iniciar o interactuar correctamente con ellos AWS CloudShell, busque información para identificarlos y solucionarlosSolución de problemas AWS CloudShell.

3. Para elegir un intérprete de comandos preinstalado con el que trabajar, introduzca uno de los siguientes nombres de programas en el símbolo del sistema.

   Bash

   ```
   bash
   ```

   Si cambias a Bash, el símbolo de la línea de comandos se actualiza a$.

> **Note**
>
> Bash es el shell predeterminado que se ejecuta al iniciar el juego AWS CloudShell.

PowerShell

```
pwsh
```

Si cambias a PowerShell, el símbolo de la línea de comandos se actualizará a`PS>`.

Z shell

```
zsh
```

Si cambia a Z shell, el símbolo de la línea de comandos se actualiza a`%`.

Para obtener información sobre las versiones preinstaladas en su entorno de shell, consulte la tabla de shells en la sección de entornos de AWS CloudShell cómputo.

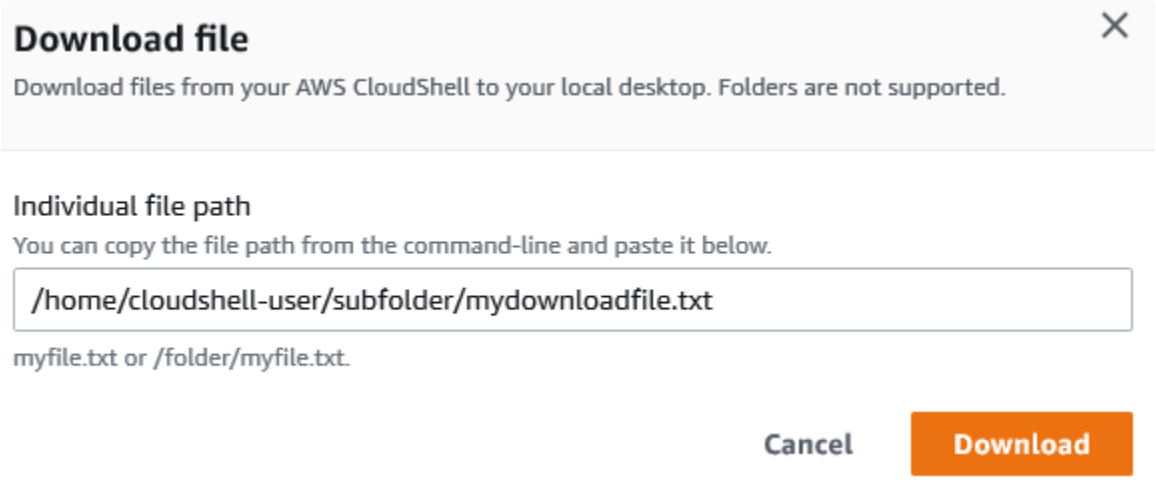
Paso 3: Descarga un archivo de AWS CloudShell

> **Note**
>
> Esta opción no está disponible para los VPC entornos.

En este paso, se detalla el proceso de descarga de un archivo.

1. Para descargar un archivo, ve a Acciones y selecciona Descargar archivo en el menú.

   Aparece el cuadro de diálogo Descargar archivo.
2. En el cuadro de diálogo Descargar archivo, introduzca la ruta del archivo que se va a descargar.



> **Note**
>
> Puede utilizar rutas absolutas o relativas al especificar un archivo para su descarga. Con nombres de ruta relativos, `/home/cloudshell-user/` se añade automáticamente al inicio de forma predeterminada. Por lo tanto, para descargar un archivo llamado "`mydownload-file`", las dos rutas siguientes son válidas:
>
> - Ruta absoluta: `/home/cloudshell-user/subfolder/mydownloadfile.txt`
> - Ruta relativa: `subfolder/mydownloadfile.txt`

3. Elija Descargar.

   Si la ruta del archivo es correcta, aparece un cuadro de diálogo. Puede utilizar este cuadro de diálogo para abrir el archivo con la aplicación por defecto. O puede guardar el archivo en una carpeta de su equipo local.

> **Note**
>
> La opción de descarga no está disponible cuando se inicia CloudShell en Console Toolbar. Puedes descargar un archivo desde la CloudShell consola o desde el navegador web Chrome. Para obtener más información sobre cómo descargar un archivo, consulta el paso 3: Descargar un archivo desde AWS CloudShell.

Paso 4: Sube un archivo a AWS CloudShell

> **Note**
>
> Esta opción no está disponible para los VPC entornos.

En este paso se describe cómo cargar un archivo y, a continuación, moverlo a un nuevo directorio del directorio principal.

1. Para comprobar su directorio de trabajo actual, introduzca el siguiente comando en la línea de comandos:

   ```
   pwd
   ```

   Al pulsar Intro, el intérprete de comandos devuelve su directorio de trabajo actual (por ejemplo, `/home/cloudshell-user`).
2. Para subir un archivo a este directorio, vaya a Acciones y seleccione Cargar archivo en el menú.

   Aparece el cuadro de diálogo Cargar archivo.
3. Elija Browse (Examinar).
4. En el cuadro de diálogo de Carga de archivos de su sistema, seleccione el archivo de texto que creó para este tutorial (`add_prog.py`) y elija Abrir.
5. En el cuadro de diálogo Añadir archivos, seleccione Cargar.

   Una barra de progreso registra la carga. Si la carga se ha realizado correctamente, un mensaje confirmará que `add_prog.py` se ha añadido a la raíz de su directorio principal.
6. Para crear un directorio para el archivo, introduzca el comando make directories: `mkdir mysub_dir`.
7. Para mover el archivo cargado de la raíz de su directorio principal al nuevo directorio, use el comando `mv`:

   ```
   mv add_prog.py mysub_dir.
   ```
8. Para cambiar el directorio de trabajo al nuevo directorio, introduzca `cd mysub_dir`.

   La línea de comandos se actualiza para indicar que ha cambiado el directorio de trabajo.
9. Para ver el contenido del directorio actual, `mysub_dir`, introduzca el comando `ls`.

Se muestra el contenido del directorio de trabajo. Esto incluye el archivo que acaba de cargar.

Paso 5: Eliminar un archivo de AWS CloudShell

En este paso se describe cómo eliminar un archivo de AWS CloudShell.

1. Para eliminar un archivo de AWS CloudShell, utilice comandos de shell estándar, como `rm` (eliminar).

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

2. Para eliminar varios archivos que cumplan los criterios especificados, ejecute el comando `find`.

   En el siguiente ejemplo, se eliminan todos los archivos que incluyen el sufijo “.pdf” en sus nombres.

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

> **Note**
>
> Supongamos que deja de usarlo AWS CloudShell en un lugar específico Región de AWS. Luego, los datos que se encuentran en el almacenamiento persistente de esa región se eliminan automáticamente después de un período específico. Para obtener más información, consulte Almacenamiento persistente.

Paso 6: cree una copia de seguridad del directorio principal

En este paso se describe cómo crear una copia de seguridad del directorio principal.

1. Crear una copia de seguridad

   Cree una carpeta temporal fuera del directorio principal.

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

   Puede utilizar uno de las siguientes opciones para crear una política de copia de seguridad:

a. Cree un archivo de respaldo con tar

Para crear un archivo de copia de seguridad mediante tar, escriba el siguiente comando:

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

b. Cree un archivo de respaldo mediante zip

Para crear un archivo de copia de seguridad mediante zip, escriba el siguiente comando:

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

2. Transfiera el archivo de respaldo al exterior CloudShell

Puede utilizar una de las siguientes opciones para transferir el archivo de copia de seguridad al exterior CloudShell:

a. Descargue el archivo de copia de seguridad en su máquina local

Puede descargar el archivo creado en el paso anterior. Para obtener más información sobre cómo descargar un archivo desde CloudShell, consulte Descargar un archivo desde AWS CloudShell.

En el cuadro de diálogo de descarga del archivo, introduzca la ruta del archivo que se va a descargar (por ejemplo, `/tmp/tmp.iA99tD9L98/home.tar.gz`).

b. Transfiera el archivo de copia de seguridad a S3

Escriba el siguiente comando para generar el bucket:

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

AWSCLIÚselo para copiar el archivo al bucket de S3:

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

> **Note**
>
> Es posible que se apliquen cargos por transferencia de datos.

3. Copia de seguridad directamente en un bucket de S3

   Para realizar una copia de seguridad directamente en un bucket de S3, escriba el siguiente comando:

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

Paso 7: reinicie una sesión del intérprete de comandos

En este paso se describe cómo reiniciar una sesión de shell.

> **Note**
>
> Como medida de seguridad, si no interactúa con el intérprete de comandos mediante el teclado o el puntero durante un período prolongado, la sesión se detiene automáticamente. Las sesiones de larga duración también se detienen automáticamente. Para obtener más información, consulte Sesiones del intérprete de comandos.

1. Para reiniciar una sesión de intérprete de comandos, seleccione Acciones, Reiniciar .

   Se le notifica que al reiniciar AWS CloudShell se detienen todas las sesiones activas de la sesión actual. Región de AWS

2. Para confirmar, seleccione Reiniciar.

   Una interfaz muestra un mensaje que indica que el entorno CloudShell informático se está deteniendo. Cuando el entorno se haya detenido y reiniciado, puede empezar a trabajar con la línea de comandos en una nueva sesión.

   > **Note**
   >
   > En algunos casos, es posible que el entorno tarde unos minutos en reiniciarse.

Paso 8: elimine el directorio principal de una sesión de intérprete de comandos

En este paso se describe cómo eliminar una sesión de shell.

> **Note**
>
> Esta opción no está disponible para los VPC entornos. Al reiniciar un VPC entorno, se elimina su directorio principal.

> **Warning**
>
> Eliminar el directorio principal es una acción irreversible en la que todos los datos almacenados en el directorio principal se eliminan de forma permanente. Sin embargo, es posible que desee considerar esta opción en las siguientes situaciones:
>
> - Modificó un archivo de forma incorrecta y no puede acceder al entorno AWS CloudShell informático. Al eliminar el directorio principal, AWS CloudShell se restablece su configuración predeterminada.
> - Quieres eliminar todos tus datos de AWS CloudShell forma inmediata. Si dejas de usarlo AWS CloudShell en una AWS región, el almacenamiento persistente se eliminará automáticamente al final del período de retención, a menos que AWS CloudShell vuelvas a iniciarlo en esa región.

Si necesita un almacenamiento prolongado para sus archivos, considere la posibilidad de utilizar un servicio como Amazon S3 o CodeCommit.

1. Para eliminar una sesión de shell, selecciona Acciones, Eliminar.

   Se le notifica que al eliminar el AWS CloudShell directorio principal se eliminan todos los datos almacenados actualmente en su AWS CloudShell entorno.

   > **Note**
   >
   > Esta acción no se puede deshacer.

2. Para confirmar la eliminación, escriba el nombre de la ubicación en el campo de entrada de texto y elija Eliminar.

   AWS CloudShell detiene todas las sesiones activas en la sesión actual Región de AWS y crea un nuevo entorno de forma inmediata.

   Salir manualmente de las sesiones del intérprete de comandos

   Con la línea de comandos, puede salir de una sesión de intérprete de comandos y cerrar sesión mediante el comando `exit`. A continuación, puede pulsar cualquier tecla para volver a conectarse y seguir utilizando AWS CloudShell.

Paso 9: edite el código de su archivo y ejecútelo usando la línea de comandos

En este paso se muestra cómo utilizar el dispositivo preinstalado Vim editor para trabajar con un archivo. A continuación, ejecute el archivo como un programa desde la línea de comandos.

1. Para editar el archivo que cargó en el paso anterior, introduzca el siguiente comando:

   ```
   vim add_prog.py
   ```

   La interfaz del shell se actualiza para mostrar el Vim editor.

2. Para editar el archivo en Vim, pulse la I tecla. Ahora edite el contenido para que el programa sume tres números en lugar de dos.

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

> **Note**
>
> Si pega el texto en el editor y tiene habilitada la característica de pegado seguro, aparecerá una advertencia. El texto de líneas múltiples que se copia puede contener scripts maliciosos. Con la característica de pegado seguro, puede verificar el texto completo antes de pegarlo. Si está seguro de que el texto es seguro, elija Pegar.

3. Tras editar el programa, pulse Esc para introducir la Vim modo de comando. A continuación, introduzca el comando `:wq` para guardar el archivo y salir del editor.

> **Note**
>
> Si eres nuevo en el Vim modo de comando, puede que al principio le resulte difícil cambiar entre el modo de comando y el modo de inserción. El modo de comando se utiliza al guardar archivos y salir de la aplicación. El modo de inserción se utiliza al insertar texto nuevo. Para entrar en el modo de inserción, pulse I, para entrar en el modo de comando, pulse Esc. Para obtener más información acerca de Vim y otras herramientas disponibles en AWS CloudShell, consulteHerramientas de desarrollo y utilidades de intérprete de comandos.

4. En la interfaz de la línea de comandos principal, ejecute el siguiente programa y especifique tres números para la entrada. La sintaxis es la siguiente.

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

La línea de comandos muestra el resultado del programa: `The sum is 15`.

Paso 10: Se utiliza AWS CLI para añadir el archivo como un objeto en un bucket de Amazon S3

En este paso, crea un depósito de Amazon S3 y, a continuación, utiliza el PutObjectmétodo para añadir el archivo de código como un objeto en ese depósito.

> **Note**
>
> En la mayoría de los casos, puede Uso CodeCommit en AWS CloudShell guardar un archivo de software en un repositorio con control de versiones. En este tutorial se muestra cómo se puede utilizar AWS CLI AWS CloudShell para interactuar con otros AWS servicios. Al usar este método, no necesita descargar o instalar recursos adicionales. Además, dado que ya está autenticado en el intérprete de comandos, no tiene que configurar las credenciales antes de realizar llamadas.

1. Para crear un bucket en un segmento específico Región de AWS, ingresa el siguiente comando:

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

> **Note**
>
> Si va a crear un depósito fuera de la región `us-east-1`, añada `create-bucket-configuration` con el parámetro `LocationConstraint` para especificar la región. A continuación, se muestra un ejemplo sintaxis .
>
> ```
> $ aws s3api create-bucket --bucket my-bucket --region eu-west-1 --create-
> bucket-configuration LocationConstraint=eu-west-1
> ```

Si la llamada se realiza correctamente, la línea de comandos muestra una respuesta del servicio similar a la siguiente salida.

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

> **Note**
>
> Si no sigue las reglas para asignar nombres a los depósitos, aparece el siguiente error: Se ha producido un error (InvalidBucketName) al llamar a la CreateBucket operación: el depósito especificado no es válido.

2. Para cargar un archivo y añadirlo como un objeto al bucket que acabas de crear, llama al método PutObject.

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

Después de cargar el objeto en el bucket de Amazon S3, la línea de comandos muestra una respuesta del servicio similar a la siguiente salida:

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

`ETag` es el hash del objeto que se almacenó. Puede usar este hash para comprobar la integridad del objeto cargado en Amazon S3.

Temas relacionados de

- Trabajar con AWS servicios en AWS CloudShell
- Copiar varios archivos entre su máquina local y CloudShell
- Uso CodeCommit en AWS CloudShell
- Trabajando con AWS CloudShell
- Personalización de tu experiencia AWS CloudShell

AWS CloudShell tutoriales

En los siguientes tutoriales se muestra cómo experimentar y probar diferentes funcionalidades e integraciones al AWS CloudShell utilizarlas.

Información general del tutorial	Más información
Copiar varios archivos	the section called “Tutorial: copiar varios archivos”
Usando CodeCommit	???
Creando prefirmado URLs	???
Construir un contenedor Docker en su interior AWS CloudShell y lanzarlo a Amazon ECR	???
Implementación de una función Lambda mediante AWS CDK	???

Copiar varios archivos entre su máquina local y CloudShell

Este tutorial muestra cómo copiar varios archivos entre su máquina local y CloudShell.

Mediante la AWS CloudShell interfaz, puede cargar o descargar un solo archivo entre su máquina local y el entorno de shell a la vez. Para copiar varios archivos entre CloudShell y su máquina local al mismo tiempo, utilice una de las siguientes opciones:

- Amazon S3: utilice buckets S3 como intermediario al copiar archivos entre su máquina local y. CloudShell
- Archivos zip: comprima varios archivos en una sola carpeta comprimida que se pueda cargar o descargar mediante la interfaz. CloudShell

> **Note**
>
> Como CloudShell no permite el tráfico entrante de Internet, actualmente no es posible utilizar comandos como `scp`, por ejemplo, `rsync` copiar varios archivos entre máquinas locales y el entorno CloudShell informático.

Carga y descarga de varios archivos mediante Amazon S3

En este paso se describe cómo cargar y descargar varios archivos mediante Amazon S3.

Requisitos previos

Para trabajar con buckets y objetos, necesita una IAM política que conceda permisos para realizar las siguientes API acciones de Amazon S3:

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

Para obtener una lista completa de las acciones de Amazon S3, consulte Acciones en la APIreferencia de Amazon Simple Storage Service.

Cargue varios archivos AWS CloudShell con Amazon S3

En este paso se describe cómo cargar varios archivos mediante Amazon S3.

1. En AWS CloudShell, cree un bucket de S3 ejecutando el siguiente `s3` comando:

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

   Si la llamada se realiza correctamente, la línea de comandos muestra una respuesta del servicio S3:

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

2. Cargue los archivos en un directorio desde el equipo local al bucket. Elija una de las siguientes opciones para cargar archivos:

   - AWS Management Console: Se utiliza drag-and-drop para cargar archivos y carpetas a un depósito.
   - AWS CLI: con la versión de la herramienta instalada en su máquina local, utilice la línea de comandos para cargar archivos y carpetas al bucket.

   Using the console

   - Abra la consola Amazon S3 en https://s3.console.aws.amazon.com/s3/.

     (Si la está utilizando AWS CloudShell, ya debería haber iniciado sesión en la consola).
   - En el panel de navegación izquierdo, elija Buckets, y después, elija el nombre del bucket en el que desea cargar sus carpetas o archivos. También puedes crear un depósito de tu elección seleccionando Crear bucket.
   - Para seleccionar un archivo en la carpeta, seleccione el archivo que desea cargar y elija Cargar. Después, arrastre y suelte los archivos y carpetas seleccionados en la ventana de la consola que indica los objetos en el bucket de destino, o seleccione Agregar archivos o Agregar carpetas.

     Los archivos seleccionados aparecen en la página Upload (Cargar).
   - Seleccione las casillas de verificación para indicar los archivos que se van a añadir.
   - Para añadir los archivos seleccionados al bucket, seleccione Cargar.

   > **Note**
   >
   > Para obtener información sobre todas las opciones de configuración al utilizar la consola, consulte ¿Cómo puedo cargar archivos y carpetas en un bucket de S3? en la Guía del usuario de Amazon Simple Storage Service.

Using AWS CLI

> **Note**
>
> Para utilizar esta opción, debe tener la AWS CLI herramienta instalada en su máquina local y configurar sus credenciales para las llamadas a AWS los servicios. Para obtener más información, consulte la AWS Command Line Interface Guía del usuario de .

- Inicie la AWS CLI herramienta y ejecute el siguiente `aws s3` comando para sincronizar el depósito especificado con el contenido del directorio actual de su máquina local:

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

Si la sincronización se realiza correctamente, se muestran los mensajes de carga para cada objeto añadido al bucket.

3. Vuelva a la línea de CloudShell comandos e introduzca el siguiente comando para sincronizar el directorio del entorno de shell con el contenido del bucket de S3:

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

> **Note**
>
> También puede añadir `--exclude "<value>"` y parámetros `--include "<value>"` al comando `sync` para realizar una concordancia de patrones para excluir o incluir un archivo u objeto concreto.
> Para obtener más información, consulte Uso de los filtros de exclusión e inclusión en la referencia de comandos de la AWS CLI .

Si la sincronización se realiza correctamente, se muestran mensajes de descarga para cada archivo descargado del bucket al directorio.

> **Note**
>
> Con el comando sync, solo los archivos nuevos y actualizados se copian recursivamente del directorio de origen al de destino.

Descargar varios archivos AWS CloudShell mediante Amazon S3

En este paso se describe cómo descargar varios archivos mediante Amazon S3.

1. Mediante la línea de AWS CloudShell comandos, introduzca el siguiente `aws s3` comando para sincronizar un bucket de S3 con el contenido del directorio actual en el entorno del shell:

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

   > **Note**
   >
   > También puede añadir `--exclude "<value>"` y parámetros `--include "<value>"` al comando `sync` para realizar una concordancia de patrones para excluir o incluir un archivo u objeto concreto.
   > Para obtener más información, consulte Uso de los filtros de exclusión e inclusión en la referencia de comandos de la AWS CLI .

   Si la sincronización se realiza correctamente, se muestran los mensajes de carga para cada objeto añadido al bucket.
2. Descargue el contenido del bucket a su equipo local. Como la consola Amazon S3 no admite la descarga de varios objetos, debe utilizar la AWS CLI que está instalada en su máquina local.

   Desde la línea de comandos de la AWS CLI herramienta, ejecute el siguiente comando:

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

   Si la sincronización se realiza correctamente, la línea de comandos muestra un mensaje de descarga para cada archivo actualizado o agregado en el directorio de destino.

> **Note**
>
> Para esta opción, debe tener la AWS CLI herramienta instalada en su máquina local y configurar sus credenciales para las llamadas a AWS los servicios. Para obtener más información, consulte la AWS Command Line Interface Guía del usuario de .

Cargue y descargue varios archivos mediante carpetas comprimidas

En este paso se describe cómo cargar y descargar varios archivos mediante carpetas comprimidas.

Con las utilidades de comprimir/descomprimir, puede comprimir varios archivos en un archivo que se puede tratar como un solo archivo. Las utilidades vienen preinstaladas en el entorno CloudShell informático.

Para obtener más información sobre las herramientas de pre-instalación, consulte Herramientas de desarrollo y utilidades de intérprete de comandos.

Cargue varios archivos a AWS CloudShell través de carpetas comprimidas

En este paso se describe cómo cargar varios archivos mediante carpetas comprimidas.

1. En su máquina local, añada los archivos que desee cargar a una carpeta comprimida.
2. Inicie y CloudShell, a continuación, seleccione Acciones y Cargar archivo.
3. En el cuadro de diálogo Cargar archivo, elija Seleccionar archivo y, a continuación, elija la carpeta comprimida que acaba de crear.
4. En el cuadro de diálogo Cargar archivo, elija Cargar para añadir el archivo seleccionado al entorno del intérprete de comandos.
5. En la línea de CloudShell comandos, ejecute el siguiente comando para descomprimir el contenido del archivo zip en un directorio específico:

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

Descarga varios archivos desde carpetas AWS CloudShell comprimidas

En este paso se describe cómo descargar varios archivos mediante carpetas comprimidas.

1. En la línea de CloudShell comandos, ejecute el siguiente comando para añadir todos los archivos del directorio actual a una carpeta comprimida:

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

2. Elija Acciones, Descargar archivo.
3. En el cuadro de diálogo Descargar archivo, introduzca la ruta de la carpeta comprimida (por ejemplo, `/home/cloudshell-user/zip-folder/zipped-archive.zip`) y, a continuación, seleccione Descargar.

   Si la ruta es correcta, un cuadro de diálogo del navegador ofrece la opción de abrir la carpeta comprimida o guardarla en el equipo local.
4. En su máquina local, ahora puede descomprimir el contenido de la carpeta comprimida descargada.

Uso CodeCommit en AWS CloudShell

CodeCommit es un servicio de control de código fuente seguro, altamente escalable y gestionado que aloja repositorios Git privados. Si lo usa AWS CloudShell, puede trabajar con él CodeCommit en la línea de comandos mediante la git-remote-codecommitutilidad. Esta utilidad viene preinstalada en el entorno AWS CloudShell informático y proporciona un método sencillo para introducir y extraer código de los CodeCommit repositorios. Esta utilidad lo hace ampliando Git. Para obtener más información, consulte la AWS CodeCommit Guía del usuario de .

En este tutorial, se describe cómo crear un CodeCommit repositorio y clonarlo en su entorno AWS CloudShell informático. También aprenderás a organizar y confirmar un archivo en tu repositorio clonado antes de enviarlo al repositorio remoto que se administra en AWS Cloud.

Requisitos previos

Para obtener información sobre los permisos que un IAM usuario debe usar AWS CloudShell, consulta la sección de requisitos previos del tutorial de introducción. También necesita IAMpermisos para trabajar con CodeCommit ellos.

Además, antes de empezar, asegúrese de tener lo siguiente:

- Una comprensión básica de los comandos de Git y los conceptos de control de versiones

- Un archivo en el directorio principal de su intérprete de comandos que se puede guardar en los repositorios locales y remotos. En este tutorial, se denomina “`my-git-file`“.

Paso 1: Crear y clonar un CodeCommit repositorio

En este paso se describe cómo crear y clonar un CodeCommit repositorio.

1. En la interfaz de línea de CloudShell comandos, introduzca el siguiente `codecommit` comando para crear un CodeCommit repositorio llamado`MyDemoRepo`.

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

   Si el repositorio se ha creado correctamente, la línea de comandos muestra la respuesta del servicio.

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

2. Con la línea de comandos, cree un nuevo directorio para su repositorio local y conviértalo en su directorio de trabajo.

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

3. Para clonar el repositorio remoto, utilice el comando `git clone`. (Mientras trabaja con él git-remote-codecommit, utilice el URL estilo HTTPS (GRC)).

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

Si el repositorio se clona correctamente, la línea de comandos muestra la respuesta del servicio.

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

4. Para navegar el repositorio clonado, utilice el comando `cd`.

```
cd MyDemoRepo
```

Paso 2: Organiza y confirma un archivo antes de subirlo a tu CodeCommit repositorio

En este paso se describe cómo organizar y confirmar un archivo antes de subirlo a tu CodeCommit repositorio.

1. Agrega un archivo llamado `my-git-file` a la `MyDemoRepo` carpeta mediante un editor de Vim o la función de carga de archivos de AWS CloudShell. Para obtener información acerca de cómo utilizar ambos, consulte el tutorial de primeros pasos.
2. Para almacenar su archivo en el repositorio, ejecute el comando `add` de git.

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

3. Para comprobar que el archivo se ha preparado y está listo para ser archivado, ejecute el comando `status` de git.

```
git status
```

`my-git-file` aparece como un archivo nuevo y se muestra en texto verde, lo que indica que está listo para ser confirmado.

4. Guarde esta versión del archivo preparado en el repositorio.

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

> **Note**
>
> Si se le solicita información de configuración para completar la confirmación, utilice el siguiente formato.
>
> ```
> $ git config --global user.name "Jane Doe"
> $ git config --global user.email janedoe@example.com
> ```

5. Para sincronizar su repositorio remoto con los cambios realizados en el repositorio local, envíe los cambios a la rama anterior.

```
git push
```

Creación de un objeto prefirmado URL para Amazon S3 mediante AWS CloudShell

En este tutorial, se muestra cómo crear un objeto prefirmado URL para compartir un objeto de Amazon S3 con otras personas. Como los propietarios de los objetos especifican sus propias credenciales de seguridad al compartir, cualquier persona que reciba el prefirmado URL puede acceder al objeto durante un tiempo limitado.

Requisitos previos

- Un IAM usuario con los permisos de acceso proporcionados por la AWSCloudShellFullAccesspolítica.
- Para conocer los IAM permisos necesarios para crear un objeto prefirmadoURL, consulte Compartir un objeto con otros en la Guía del usuario de Amazon Simple Storage Service.

Paso 1: Crear un IAM rol para conceder acceso al bucket de Amazon S3

En este paso se describe cómo crear un IAM rol para conceder acceso al bucket de Amazon S3.

1. Para obtener sus IAM datos y poder compartirlos, llame al `get-caller-identity` comando from AWS CloudShell.

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

Si la llamada se realiza correctamente, la línea de comandos muestra una respuesta similar a la siguiente:

```
{
    "Account": "123456789012",
    "UserId": "AROAXXOZUUOTTWDCVIDZ2:redirect_session",
    "Arn": "arn:aws:sts::531421766567:assumed-role/Feder08/redirect_session"
}
```

2. Tome la información de usuario que obtuvo en el paso anterior y agréguela a una plantilla de AWS CloudFormation . Esta plantilla crea un IAM rol. Este rol otorga a su colaborador los permisos con los privilegios mínimos para los recursos compartidos.

```
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. Guarda la AWS CloudFormation plantilla en un archivo con el nombre`template.yaml`.
4. Usa la plantilla para implementar la pila y crear el IAM rol mediante una llamada al `deploy` comando.

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

Genera el prefirmado URL

En este paso se describe cómo generar el URL prefirmado.

1. Con el editor incorporado AWS CloudShell, añade el siguiente código. Este código crea un URL que proporciona a los usuarios federados acceso directo al AWS Management Console.

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

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

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

  # Send final URL to stdout
  print (request_url)

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

2. Guarde el código en un archivo denominado `share.py`.
3. Ejecute lo siguiente desde la línea de comandos para recuperar el nombre de recurso de Amazon (ARN) del IAM rol AWS CloudFormation. A continuación, utilícelo en Python script para obtener credenciales de seguridad temporales.

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

   El script devuelve un archivo en el URL que un colaborador puede hacer clic para acceder a AWS CloudShell él. AWS Management Console El colaborador tiene el control total de la carpeta `myfolder/` del bucket de Amazon S3 durante los próximos 3600 segundos (1 hora). Las credenciales caducan después de una hora. Transcurrido este tiempo, el colaborador ya no podrá acceder al bucket.

Crear un contenedor Docker en su interior CloudShell y enviarlo a un repositorio de Amazon ECR

En este tutorial, se muestra cómo definir y crear un contenedor de Docker AWS CloudShell y cómo enviarlo a un ECR repositorio de Amazon.

Requisitos previos

- Debes tener los permisos necesarios para crear un ECR repositorio de Amazon e insertarlo en él. Para obtener más información sobre los repositorios de AmazonECR, consulta los repositorios ECR privados de Amazon en la Guía ECRdel usuario de Amazon. Para obtener más información sobre los permisos necesarios para enviar imágenes con AmazonECR, consulta IAMlos permisos necesarios para enviar una imagen en la Guía del ECR usuario de Amazon.

Procedimiento tutorial

El siguiente tutorial describe cómo usar la CloudShell interfaz para crear un contenedor de Docker y enviarlo a un ECR repositorio de Amazon.

1. Crea una nueva carpeta en tu directorio principal.

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

2. Navega hasta la carpeta que creaste.

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

3. Crea un Dockerfile vacío.

```
touch Dockerfile
```

4. Con un editor de texto, por ejemplo`nano Dockerfile`, abra el archivo y pegue el siguiente contenido en él.

```
# 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. El Dockerfile ya está listo para ser creado. Construya el contenedor ejecutándolo. `docker build` Etiquete el contenedor con un easy-to-type nombre para usarlo en futuros comandos.

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

   Asegúrese de incluir el punto final (`.`).

6. Ahora puede probar el contenedor para comprobar que funciona correctamente. AWS CloudShell

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

7. Ahora que tienes un contenedor de Docker en funcionamiento, necesitas subirlo a un ECR repositorio de Amazon. Si ya tienes un ECR repositorio de Amazon, puedes saltarte este paso.

   Ejecuta el siguiente comando para crear un ECR repositorio de Amazon para este tutorial.

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

8. Después de crear el ECR repositorio de Amazon, puedes insertar el contenedor de Docker en él.

   Ejecuta el siguiente comando para obtener las credenciales de inicio de ECR sesión de Amazon para Docker.

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

   > **Note**
   >
   > Si la variable de AWS_REGION entorno no está configurada en su archivo CloudShell o si desea interactuar con los recursos de otro Regiones de AWS, ejecute el siguiente comando:

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

9. Etiquete la imagen con el ECR repositorio de Amazon de destino y, a continuación, envíela a ese repositorio.

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

Si encuentra errores o tiene problemas al intentar completar este tutorial, consulte la sección de solución de problemas de esta guía para obtener ayuda.

Limpieza

Ya has implementado correctamente tu contenedor de Docker en tu ECR repositorio de Amazon. Para eliminar de su AWS CloudShell entorno los archivos que creó en este tutorial, ejecute el siguiente comando.

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

- Elimina el ECR repositorio de Amazon.

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

Implementación de una función Lambda mediante el AWS CDKAWS CloudShell

En este tutorial, se muestra cómo implementar una función de Lambda en su cuenta mediante el AWS Cloud Development Kit (AWS CDK) comando in. CloudShell

Requisitos previos

- Inicie su cuenta para usarla con. AWS CDKPara obtener información sobre el arranque con AWS CDK, consulte Bootstrapping en la Guía para desarrolladores de la versión 2.AWS CDK Si no has iniciado la cuenta, puedes entrar corriendo. `cdk bootstrap` CloudShell

- Asegúrate de tener los permisos adecuados para implementar recursos en tu cuenta. Se recomiendan permisos de administrador.

Procedimiento tutorial

El siguiente tutorial describe cómo implementar una función Lambda basada en contenedores de Docker mediante in. AWS CDK CloudShell

1. Cree una nueva carpeta en su directorio principal.

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

2. Navega hasta la carpeta que creaste.

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

3. Instala las AWS CDK dependencias de forma local.

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

4. Cree un AWS CDK proyecto básico en la carpeta que creó.

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

5. Con un editor de texto, por ejemplo`nano cdk.json`, abra el archivo y pegue el siguiente contenido en él.

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

6. Abre el `lib/docker-tutorial.js` archivo y pega el siguiente contenido en él.

```
// 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. Abre `lib/Dockerfile` y pega el siguiente contenido en él.

```
# 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. Abre el `lib/hello.js` archivo y pega el siguiente contenido en él.

```
// 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. Utilice AWS CDK CLI para sintetizar el proyecto y desplegar los recursos. Debe iniciar su cuenta.

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

10. Invoque la función Lambda para confirmarla y verificarla.

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

Ahora ha implementado correctamente una función Lambda basada en contenedores de Docker mediante. AWS CDKPara obtener más información AWS CDK, consulte la Guía para desarrolladores de la versión 2 AWS CDK. Si encuentra errores o problemas al intentar completar este tutorial, consulte la sección de solución de problemas de esta guía para obtener ayuda.

Limpieza

Ahora ha implementado correctamente una función Lambda basada en contenedores de Docker mediante. AWS CDK Dentro del AWS CDK proyecto, ejecute el siguiente comando para eliminar los recursos asociados. Se le pedirá que confirme la eliminación.

-
```
npx cdk destroy DockerTutorialStack
```

- Para eliminar de su AWS CloudShell entorno los archivos y recursos que creó en este tutorial, ejecute el siguiente comando.

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

Trabajando con AWS CloudShell

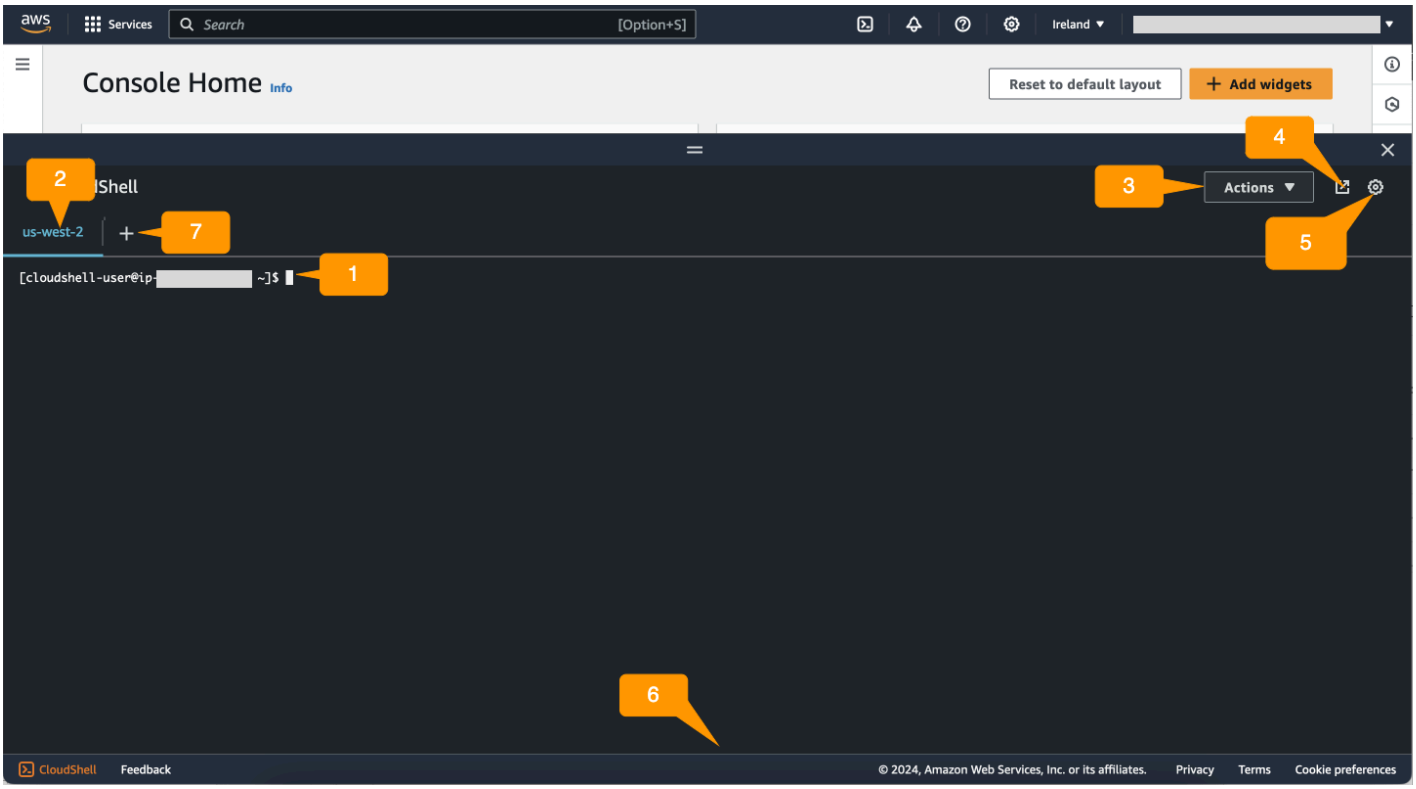
En esta sección se describe cómo interactuar con las aplicaciones compatibles AWS CloudShell y realizar acciones específicas con ellas.

Temas

Navegar por la interfaz AWS CloudShell

Puede navegar por las funciones CloudShell de la interfaz desde y AWS Management Console Console Toolbar.

La siguiente captura de pantalla muestra varias funciones clave AWS CloudShell de la interfaz.



1. AWS CloudShell interfaz de línea de comandos que se utiliza para ejecutar comandos mediante el shell que prefiera. El tipo de intérprete de comandos actual se indica en la línea de comandos.
2. La pestaña de terminal, que usa la ubicación Región de AWS en la que AWS CloudShell se está ejecutando actualmente.
3. El menú Acciones, que ofrece opciones para cambiar el diseño de la pantalla, descargar y cargar archivos, reiniciar su AWS CloudShell y eliminar su directorio principal de AWS CloudShell.

> **Note**
>
> La opción de descarga no está disponible cuando se inicia CloudShell en el Console Toolbar.

4. La pestaña Abrir en un navegador nuevo, que ofrece la opción de acceder a la CloudShell sesión en pantalla completa.
5. La opción Preferencias, que puede utilizar para personalizar su experiencia de intérprete de comandos.
6. La barra inferior, que ofrece las siguientes opciones para:
   - CloudShell Lánzala desde el CloudShellicono.
   - Envíe sus comentarios desde el icono Comentarios. Elija el tipo de comentarios que quiere enviar, añada sus comentarios y, a continuación, seleccione Enviar.
     - Para enviar comentarios CloudShell, elige una de las siguientes opciones:
       - Desde la consola CloudShell, inicia y selecciona Comentarios. Añada sus comentarios y, a continuación, seleccione Enviar.
       - Elige una CloudShellde las Console Toolbar, en la parte inferior izquierda de la consola y, a continuación, selecciona el icono Abrir en una nueva pestaña del navegador, Comentarios. Añada sus comentarios y, a continuación, seleccione Enviar.

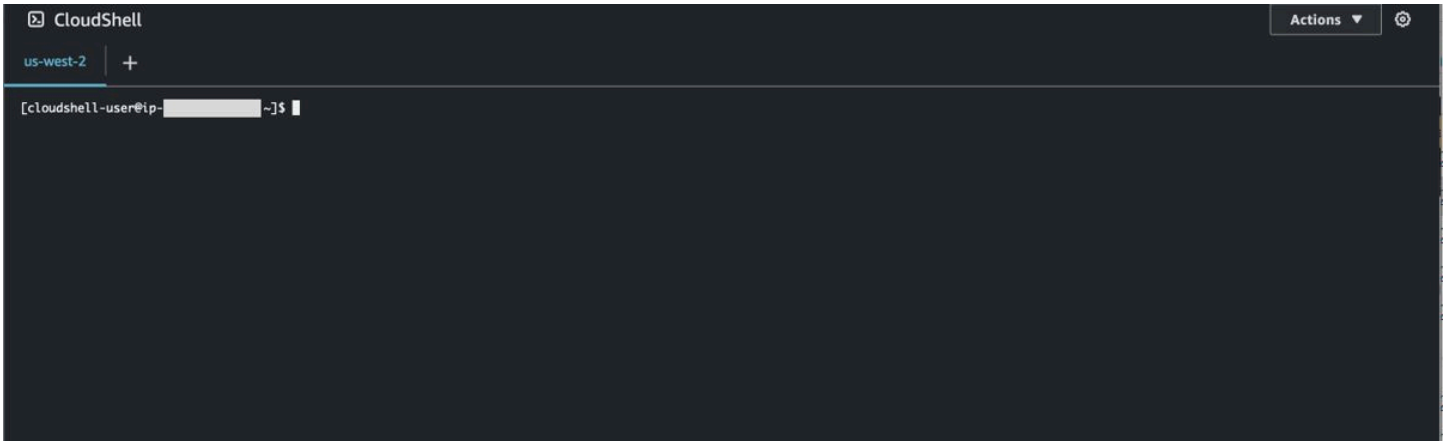
> **Note**
>
> La opción de comentarios no está disponible cuando se inicia CloudShell en Console Toolbar.

   - Obtenga información sobre nuestra política de privacidad y nuestras condiciones de uso, y personalice las preferencias de cookies.

7. El icono + es un menú desplegable que incluye opciones para crear, reiniciar y eliminar entornos.

¿Trabajando en Regiones de AWS

La corriente en la Región de AWS que te estás ejecutando se muestra como una pestaña.



Puede elegir una en Región de AWS la que trabajar seleccionando una región específica mediante el selector de regiones. Tras cambiar de región, la interfaz se actualiza a medida que la sesión del intérprete de comandos se conecta a un entorno de computación diferente que se ejecute en la región seleccionada.

Important

- Puedes usar hasta 1 GB de almacenamiento persistente en cada una Región de AWS. El almacenamiento persistente se guarda en su directorio principal (`$HOME`). Esto significa que todos los archivos, directorios, programas o scripts personales que estén almacenados en su directorio principal se encuentran todos en una Región de AWS. Además, son diferentes de los que se encuentran en el directorio principal y se almacenan en una región diferente.

  La retención de archivos en el ámbito de almacenamiento persistente a largo plazo también se gestiona por región. Para obtener más información, consulte Almacenamiento persistente.
- El almacenamiento persistente no está disponible para AWS CloudShell VPC los entornos.

Especifica tu valor predeterminado Región de AWS para AWS CLI

Puede utilizar variables de entorno para especificar las opciones de configuración y las credenciales necesarias para acceder a Servicios de AWS ellas AWS CLI. La variable de entorno que especifica el valor predeterminado Región de AWS de la sesión de shell se establece cuando se inicia AWS CloudShell desde una región específica AWS Management Console o cuando se selecciona una opción en el selector de regiones.

Las variables de entorno tienen prioridad sobre los archivos de AWS CLI credenciales que se actualizan medianteaws configure. Por lo tanto, no puede ejecutar el comando aws configure para cambiar la región especificada por la variable de entorno. En su lugar, para cambiar la región predeterminada de AWS CLI los comandos, asigne un valor a la variable de AWS_REGION entorno. En los ejemplos siguientes, sustituya us-east-1 por la región en la que se encuentre.

Bash or Zsh

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

La configuración de la variable de entorno cambia el valor usado hasta el finalice la sesión del intérprete de comandos o cuando se otorgue a la variable un valor diferente. Puede establecer variables en el script de inicio de su intérprete de comandos para que las variables persistan en futuras sesiones.

PowerShell

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

Si establece una variable de entorno en la PowerShell solicitud, la variable de entorno guarda el valor únicamente durante la sesión actual. Como alternativa, puede configurar la variable para todas las PowerShell sesiones futuras añadiendo la variable a su PowerShell perfil. Para obtener más información sobre el almacenamiento de variables de entorno, consulte la PowerShell documentación.

Para confirmar que ha cambiado la región predeterminada, ejecute el aws configure list comando para mostrar los datos de AWS CLI configuración actuales.

> **Note**
>
> Para AWS CLI comandos específicos, puede anular la región predeterminada mediante la opción `--region` de línea de comandos. Para obtener más información, consulte Opciones de la línea de comandos en la Guía del usuario de la AWS Command Line Interface .

Uso de archivos y almacenamiento

Mediante AWS CloudShell la interfaz, puede cargar y descargar archivos desde el entorno de shell. Para obtener más información sobre cómo descargar y cargar archivos, consulte Primeros pasos con AWS CloudShell.

Para asegurarse de que todos los archivos que añada estén disponibles cuando la sesión haya finalizado, debe conocer la diferencia entre almacenamiento persistente y temporal.

- Almacenamiento persistente: dispone de 1 GB de almacenamiento persistente para cada uno Región de AWS. El almacenamiento persistente se encuentra en el directorio principal.
- Almacenamiento temporal: el almacenamiento temporal se recicla al final de una sesión. El almacenamiento temporal se encuentra en los directorios que se encuentran fuera del directorio principal.

> **Important**
>
> Asegúrese de dejar los archivos que desee conservar y usar para futuras sesiones del intérprete de comandos en su directorio principal. Por ejemplo, supongamos que mueve un archivo fuera de su directorio principal ejecutando el comando `mv`. A continuación, ese archivo se recicla cuando finaliza la sesión del intérprete de comandos actual.

Uso de Docker

AWS CloudShell es totalmente compatible con Docker sin necesidad de instalación ni configuración. Puede definir, construir y ejecutar contenedores Docker en su interior. AWS CloudShell Puede implementar recursos basados en Docker, como funciones Lambda basadas en contenedores de Docker, a través del AWS CDK kit de herramientas, así como crear contenedores de Docker y

enviarlos a los repositorios de Amazon a través de Docker. ECR CLI Para ver los pasos detallados sobre cómo ejecutar estas dos implementaciones, consulte los siguientes tutoriales:

- Tutorial: Implementación de una función Lambda mediante AWS CDK
- Tutorial: Crear un contenedor Docker en su interior AWS CloudShell y subirlo a un repositorio de Amazon ECR

El uso de Docker con: AWS CloudShell

- El espacio de Docker en un entorno es limitado. Si tiene imágenes individuales de gran tamaño o demasiadas imágenes de Docker preexistentes, puede que se produzcan problemas que le impidan extraer, crear o ejecutar imágenes adicionales. Para obtener más información sobre Docker, consulta la guía de documentación de Docker.
- Docker está disponible en todas AWS las regiones, excepto en las regiones AWS GovCloud (EE. UU.). Para ver una lista de las regiones en las que Docker está disponible, consulta AWS las regiones compatibles para. AWS CloudShell
- Si tienes problemas al usar Docker con AWS CloudShell, consulta la sección de solución de problemas de esta guía para obtener información sobre cómo resolver estos problemas.

Funciones de accesibilidad para AWS CloudShell

En este tema se describe cómo utilizar las funciones de accesibilidad para CloudShell. Puede utilizar un teclado para navegar por los elementos enfocables de la página. También puede personalizar la apariencia de CloudShell, incluidos los tamaños de fuente y los temas de la interfaz.

Navegación por teclado en CloudShell

Para navegar por los elementos enfocables de la página, pulse Tab.

CloudShell funciones de accesibilidad del terminal

Puede usar la tecla Tab de las siguientes formas:

- Modo terminal (predeterminado): en este modo, el terminal captura la entrada de clave Tab. Cuando el foco esté en el terminal, pulse Tab para acceder únicamente a las funciones del terminal.
- Modo de navegación: en este modo, el terminal no captura la entrada de clave Tab. Pulse Tab para navegar por los elementos enfocables de la página.

Para cambiar entre el modo terminal y el modo de navegación, pulse Ctrl+M. Cuando vuelva a cambiarlo, aparecerá la pestaña: navegación en el encabezado y podrá usar la tecla Tab para navegar por la página.

Para volver al modo terminal, presione Ctrl+M. O bien, seleccione X junto a la pestaña: navegación.

> **Note**
>
> Actualmente, las funciones de accesibilidad de los CloudShell terminales no están disponibles en los dispositivos móviles.

Elegir tamaños de fuente y temas de interfaz en CloudShell

Puede personalizar la apariencia de CloudShell para adaptarla a sus preferencias visuales.

- Tamaño de fuente: elija entre los tamaños de fuente miniatura, pequeño, mediano, grande y extra grande del terminal. Para obtener más información acerca del cambio del tamaño de fuente, consulte the section called “Cambiar el tamaño de la fuente”.
- Tema: elija entre los temas de interfaz claros y oscuros. Para obtener más información acerca de cómo cambiar el tema de la interfaz, consulte the section called “Cambiar el tema de la interfaz”.

Trabajar con AWS servicios en AWS CloudShell

Una ventaja clave AWS CloudShell es que puede usarlo para administrar sus AWS servicios desde la interfaz de línea de comandos. Esto significa que no es necesario descargar e instalar herramientas o configurar las credenciales localmente de antemano. Al lanzarlo AWS CloudShell, se crea un entorno informático que ya tiene instaladas las siguientes herramientas de línea de AWS comandos:

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

Además, dado que ya ha iniciado sesión AWS, no es necesario configurar sus credenciales de forma local antes de utilizar los servicios. Las credenciales que utilizó para iniciar sesión en la AWS Management Console se reenvían a AWS CloudShell.

Si desea cambiar la AWS región predeterminada para la que se utiliza AWS CLI, puede cambiar el valor asignado a la variable de `AWS_REGION` entorno. (Para obtener más información, consulte Especifica tu valor predeterminado Región de AWS para AWS CLI).

En el resto de este tema se muestra cómo puede empezar AWS CloudShell a utilizarlos para interactuar con AWS los servicios seleccionados desde la línea de comandos.

AWS CLI ejemplos de línea de comandos para AWS servicios seleccionados

Los ejemplos siguientes representan solo algunos de los numerosos AWS servicios con los que puede trabajar mediante los comandos disponibles en la AWS CLI versión 2. Para obtener una lista completa, consulte la Referencia de AWS CLI comandos.

- DynamoDB
- AWS Cloud9
- Amazon EC2
- S3 Glacier

DynamoDB

DynamoDB es un servicio SQL sin base de datos totalmente gestionado que proporciona un rendimiento rápido y predecible con una escalabilidad perfecta. La implementación del SQL modo No en este servicio admite estructuras de datos de documentos y valores clave.

El siguiente `create-table` comando crea una tabla sin SQL estilo que se nombra `MusicCollection` en tu AWS cuenta.

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

Para obtener más información, consulte Utilización de DynamoDB con la AWS CLI en la Guía del usuario de AWS Command Line Interface .

AWS Cloud9

AWS Cloud9 es un entorno de desarrollo integrado basado en la nube (IDE) que puedes usar para escribir, ejecutar y depurar el código en una ventana del navegador. El entorno incluye un editor de código, un depurador y un terminal.

El siguiente `create-environment-ec2` comando crea un entorno de AWS Cloud9 EC2 desarrollo con la configuración especificada. Lanza una EC2 instancia de Amazon y, a continuación, se conecta desde la instancia al entorno.

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

Para obtener más información, consulte la referencia de la línea de comandos AWS Cloud9.

Amazon EC2

Amazon Elastic Compute Cloud (AmazonEC2) es un servicio web que proporciona una capacidad informática segura y de tamaño variable en la nube. Está diseñado para hacer más fácil y accesible la computación en la nube a escala web.

El siguiente `run-instances` comando lanza una instancia t2.micro en la subred especificada de un: VPC

```
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
```

Para obtener más información, consulta Cómo usar Amazon EC2 con el AWS CLI en la Guía del AWS Command Line Interface usuario.

S3 Glacier

S3 Glacier y S3 Glacier Deep Archive son clases de almacenamiento en la nube Amazon S3 seguras, duraderas y extremadamente económicas que permiten el archivado de datos y el respaldo a largo plazo.

El siguiente comando `create-vault` crea una bóveda, un contenedor para almacenar archivos:

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

Para obtener más información, consulte Uso de Amazon S3 Glacier con la AWS CLI en la Guía del usuario de AWS Command Line Interface .

AWS Elastic Beanstalk CLI

AWS Elastic Beanstalk CLIProporciona una interfaz de línea de comandos diseñada para simplificar la creación, la actualización y la supervisión de los entornos desde un repositorio local. En este contexto, un entorno se refiere a un conjunto de AWS recursos que ejecutan una versión de la aplicación.

El siguiente `create` comando crea un nuevo entorno en una Amazon Virtual Private Cloud (VPC) personalizada.

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

Para obtener más información, consulte la referencia de CLI comandos de EB en la Guía para AWS Elastic Beanstalk desarrolladores.

Amazon ECS CLI

La interfaz de línea de comandos de Amazon Elastic Container Service (AmazonECS) (CLI) proporciona varios comandos de alto nivel. Están diseñadas para simplificar los procesos de creación, actualización y monitoreo de clústeres y tareas desde un entorno de desarrollo local. (Un ECS clúster de Amazon es una agrupación lógica de tareas o servicios).

El siguiente `configure` comando configura Amazon ECS CLI para crear una configuración de clúster denominada`ecs-cli-demo`. Esta configuración de clúster utiliza `FARGATE` como tipo de lanzamiento predeterminado para el clúster `ecs-cli-demo` en `us-east-1 region`.

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

Para obtener más información, consulte la referencia de línea de ECS comandos de Amazon en la Guía para desarrolladores de Amazon Elastic Container Service.

AWS SAM CLI

AWS SAM CLIes una herramienta de línea de comandos que funciona con una AWS Serverless Application Model plantilla y un código de aplicación. Puede realizar varias tareas con ella. Estas incluyen la invocación local de las funciones de Lambda, la creación de un paquete de despliegue para la aplicación sin servidor y el despliegue de la aplicación sin servidor en la nube. AWS

El siguiente `init` comando inicializa un nuevo SAM proyecto con los parámetros necesarios pasados como parámetros:

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

Para obtener más información, consulte la referencia de AWS SAM CLI comandos en la Guía para AWS Serverless Application Model desarrolladores.

Personalización de tu experiencia AWS CloudShell

Puede personalizar los siguientes aspectos de su AWS CloudShell experiencia:

- Diseño de pestañas: divida la interfaz de la línea de comandos en varias columnas y filas.
- Tamaño de fuente: ajuste el tamaño del texto de la línea de comandos.
- Tema de color: alterne entre un tema claro y uno oscuro.
- Pegado seguro: active o desactive una función que requiere que verifique el texto multilínea antes de pegarlo.
- Tmux para restaurar la sesión: el uso de tmux restaura la sesión hasta que quede inactiva.

También puede ampliar su entorno de shell instalando su propio software y modificando su shell con scripts.

Dividir la pantalla de la línea de comandos en varias pestañas

Ejecute varios comandos dividiendo la interfaz de la línea de comandos en varios paneles.

> **Note**
>
> Tras abrir varias pestañas, puede seleccionar una en la que desee trabajar haciendo clic en cualquier parte del panel que desee. Puede cerrar una pestaña seleccionando el símbolo x, que se encuentra junto al nombre de la región.

- Seleccione Acciones y una de las siguientes opciones en el diseño de Pestañas:
  - Nueva pestaña: agrega una nueva pestaña que esté al lado de la que está activa actualmente.
  - Dividir en filas: agrega una nueva pestaña en una fila que esté por debajo de la que está activa actualmente.
  - Dividir en columnas: agrega una nueva pestaña en una columna que esté al lado de la que está activa actualmente.

Si no hay espacio suficiente para mostrar todas las pestañas por completo, desplácese para ver la pestaña completa. También puede seleccionar las barras divisorias que separan los paneles y arrastrarlas con el puntero para aumentar o reducir el tamaño del panel.

Cambiar el tamaño de la fuente

Aumente o disminuya el tamaño del texto que se muestra en la interfaz de la línea de comandos.

1. Para cambiar la configuración del AWS CloudShell terminal, vaya a Configuración, Preferencias.
2. Elija un tamaño de texto. Las opciones son la mínima, la pequeña, la mediana, la grande y la extra grande.

Cambiar el tema de la interfaz

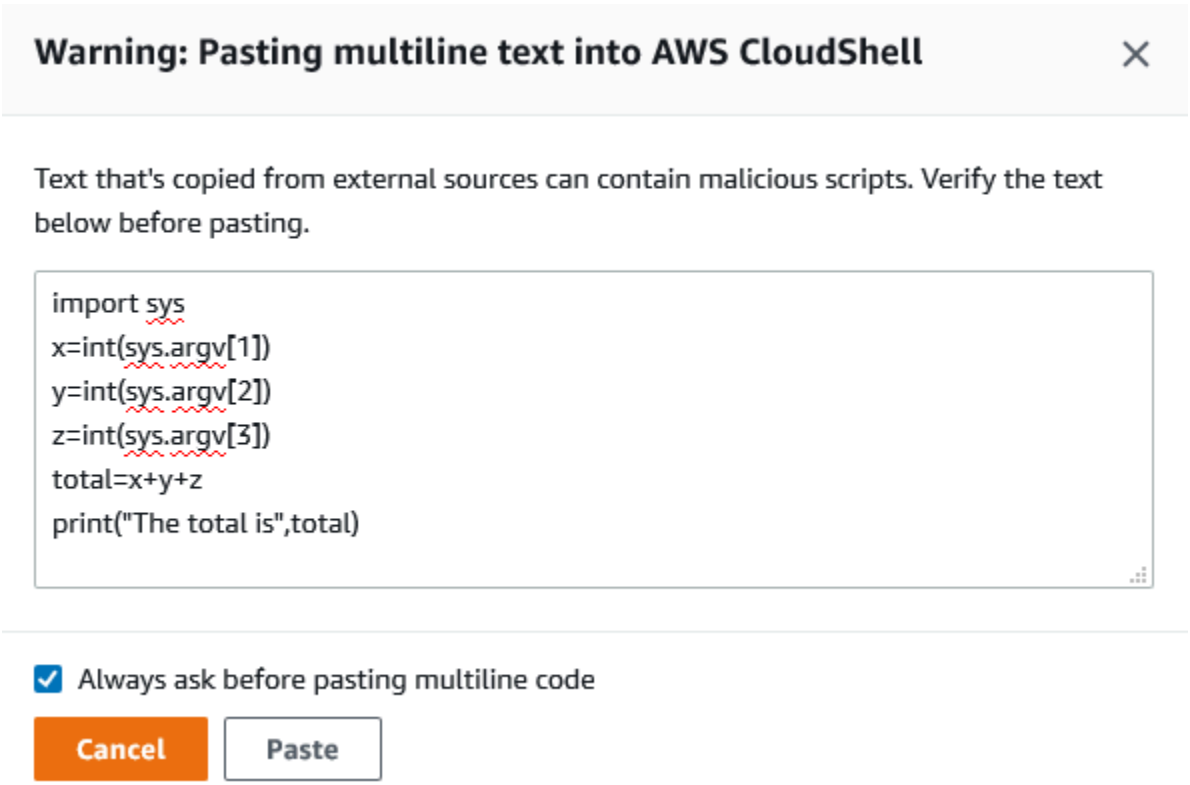
Cambie entre el tema claro y el oscuro en la interfaz de la línea de comandos.

1. Para cambiar el AWS CloudShell tema, vaya a Configuración, Preferencias.
2. Elija Claro u Oscuro.

Uso de pegado seguro para texto de líneas múltiples

El pegado seguro es una característica de seguridad que le solicita que compruebe que el texto multilínea que va a pegar en el intérprete de comandos no contiene scripts maliciosos. El texto que se copia de sitios de terceros puede contener código oculto que provoca comportamientos inesperados en el entorno del intérprete de comandos.

El cuadro de diálogo de pegado seguro muestra el texto completo que ha copiado en el portapapeles. Si está convencido de que no existe ningún riesgo de seguridad, elija Pegar.



Le recomendamos que active el pegado seguro para detectar posibles riesgos de seguridad en los scripts. Para activar o desactivar esta característica, seleccione Preferencias, Habilitar el pegado seguro y Desactivar el pegado seguro.

Utilización tmux para restaurar la sesión

AWS CloudShell usa tmux para restaurar las sesiones en una o varias pestañas del navegador. Si actualiza las pestañas del navegador, la sesión se reanudará hasta que quede inactiva. Para obtener más información, consulte Restauración de sesión.

Uso AWS CloudShell en Amazon VPC

AWS CloudShell la nube privada virtual (VPC) le permite crear un CloudShell entorno en suVPC. Para cada VPC entorno, puede asignar unaVPC, añadir una subred y asociar hasta cinco grupos de seguridad. AWS CloudShell hereda la configuración de red del VPC y le permite utilizarlos de AWS CloudShell forma segura dentro de la misma subred que otros recursos del mismo VPC y conectarse a ellos.

Con AmazonVPC, puede lanzar AWS recursos en una red virtual aislada de forma lógica que haya definido. Esta red virtual es muy similar a la red tradicional que usaría en su propio centro de datos, pero con los beneficios que supone utilizar la infraestructura escalable de AWS. Para obtener más informaciónVPC, consulte Amazon Virtual Private Cloud.

Restricciones operativas

AWS CloudShell VPClos entornos tienen las siguientes restricciones:

- Puede crear un máximo de dos VPC entornos por IAM principal.
- Puede asignar un máximo de cinco grupos de seguridad a un VPC entorno.
- No puede utilizar las opciones de CloudShell carga y descarga del menú Acciones para VPC los entornos.

> **Note**
>
> Es posible cargar o descargar archivos desde VPC entornos que tienen acceso a Internet (entrada/salida) a través de otras herramientas. CLI

- VPClos entornos no admiten el almacenamiento persistente. El almacenamiento es efímero. Los datos y el directorio principal se eliminan cuando finaliza una sesión de entorno activo.
- Su AWS CloudShell entorno solo puede conectarse a Internet si está en una VPC subred privada.

> **Note**
>
> Las direcciones IP públicas no se asignan a los CloudShell VPC entornos de forma predeterminada. VPClos entornos creados en subredes públicas con tablas de enrutamiento configuradas para enrutar todo el tráfico a Internet Gateway no tendrán

acceso a la Internet pública, pero las subredes privadas configuradas con la traducción de direcciones de red (NAT) sí tendrán acceso a la Internet pública. VPClos entornos creados en dichas subredes privadas tendrán acceso a Internet pública.

- Para proporcionar un CloudShell entorno gestionado para su cuenta, AWS puede proporcionar acceso de red a los siguientes servicios para el host informático subyacente:
  - Amazon S3
  - VPCpuntos finales
    - com.amazonaws. <region>Mensajes.ssm
    - com.amazonaws. <region>.logs
    - com.amazonaws. <region>.kms
    - com.amazonaws. <region>.execute-api
    - com.amazonaws. <region>.ecs-telemetría
    - 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

  No puede restringir el acceso a estos puntos finales modificando su configuración. VPC

  CloudShellVPCestá disponible en todas AWS las regiones, excepto en las regiones AWS GovCloud (EE. UU.). Para ver una lista de las regiones en las que CloudShell VPC está disponible, consulta AWS las regiones compatibles para AWS CloudShell.

Crear un CloudShell VPC entorno

En este tema se explican los pasos necesarios para crear un VPC entorno en CloudShell.

Requisitos previos

El administrador debe proporcionarle los IAM permisos necesarios para que pueda crear VPC entornos. Para obtener más información sobre cómo habilitar los permisos para crear CloudShell

VPC entornos, consultethe section called “IAMPermisos necesarios para crear y usar CloudShell VPC entornos”.

Para crear un CloudShell VPC entorno

1. En la página de la CloudShell consola, selecciona el icono + y, a continuación, selecciona Crear VPC entorno en el menú desplegable.
2. En la página Crear un VPC entorno, introduzca un nombre para su VPC entorno en el cuadro Nombre.
3. En la lista desplegable Nube privada virtual (VPC), elija unVPC.
4. En la lista desplegable de subredes, elija una subred.
5. En la lista desplegable de grupos de seguridad, elija uno o más grupos de seguridad que desee asignar a su entorno. VPC

   > **Note**
   >
   > Puede elegir un máximo de cinco grupos de seguridad.

6. Elija Crear para crear su VPC entorno.
7. (Opcional) Elija Acciones y, a continuación, elija Ver detalles para revisar los detalles del VPC entorno recién creado. La dirección IP de su VPC entorno se muestra en la línea de comandos.

Para obtener información sobre el uso de VPC entornos, consulteIntroducción.

IAMPermisos necesarios para crear y usar CloudShell VPC entornos

Para crear y usar CloudShell VPC entornos, el IAM administrador debe habilitar el acceso a EC2 permisos VPC específicos de Amazon. En esta sección se enumeran los EC2 permisos de Amazon necesarios para crear y usar VPC entornos.

Para crear VPC entornos, la IAM política asignada a su función debe incluir los siguientes EC2 permisos de Amazon:

- `ec2:DescribeVpcs`
- `ec2:DescribeSubnets`

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

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

Recomendamos incluir:

- ec2:DeleteNetworkInterface

> **Note**
>
> Este permiso no es obligatorio, pero es necesario CloudShell para limpiar el ENI recurso (ENIscreado para que los CloudShell VPC entornos estén etiquetados con una ManagedByCloudShell clave) creado por él. Si este permiso no está habilitado, debe limpiar el ENI recurso manualmente después de cada uso del CloudShell VPC entorno.

IAMpolítica que otorga CloudShell acceso completo, incluido el acceso a VPC

En el siguiente ejemplo, se muestra cómo habilitar todos los permisos, incluido el VPC acceso a CloudShell:

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

```
    {
      "Sid": "AllowDescribeVPC",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeVpcs"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowCreateTagWithCloudShellKey",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:*:network-interface/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateNetworkInterface"
        },
        "ForAnyValue:StringEquals": {
          "aws:TagKeys": "ManagedByCloudShell"
        }
      }
    },
    {
      "Sid": "AllowCreateNetworkInterfaceWithSubnetsAndSG",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:security-group/*"
      ]
    },
    {
      "Sid": "AllowCreateNetworkInterfaceWithCloudShellTag",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface"
```

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

Uso de claves de IAM condición para VPC entornos

Puede usar claves de condición CloudShell específicas en la VPC configuración a fin de proporcionar controles de permisos adicionales para sus VPC entornos. También puede especificar las subredes y los grupos de seguridad que el VPC entorno puede y no puede usar.

CloudShell admite las siguientes claves de condición en las IAM políticas:

- `CloudShell:VpcIds`— Permitir o denegar una o más VPCs
- `CloudShell:SubnetIds`— Permitir o denegar una o más subredes
- `CloudShell:SecurityGroupIds`— Permitir o denegar uno o más grupos de seguridad

> **Note**
>
> Si los permisos de los usuarios con acceso a CloudShell entornos públicos se modifican para añadir restricciones a la `cloudshell:createEnvironment` acción, podrán seguir accediendo a su entorno público actual. Sin embargo, si desea modificar una IAM política con esta restricción e inhabilitar su acceso al entorno público existente, primero debe actualizar la IAM política con la restricción y, a continuación, asegurarse de que todos los CloudShell usuarios de su cuenta eliminen manualmente el entorno público existente mediante la interfaz de usuario CloudShell web (Acciones → Eliminar CloudShell entorno).

Ejemplos de políticas con claves de condición para la configuración VPC

Los siguientes ejemplos muestran cómo utilizar las claves de condición para la VPC configuración. Después de crear una instrucción de política con las restricciones deseadas, agregue la instrucción de política para el usuario o rol de destino.

Asegúrese de que los usuarios creen únicamente VPC entornos y denieguen la creación de entornos públicos

Para garantizar que los usuarios solo puedan crear VPC entornos, utilice el permiso de denegación como se muestra en el siguiente ejemplo:

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

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

Denegue a los usuarios el acceso a subredes o grupos de seguridad específicos VPCs

Para denegar a los usuarios el acceso a un VPCs contenido específico, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:VpcIds` condición. El siguiente ejemplo deniega a los usuarios el acceso a `vpc-1` y`vpc-2`:

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

Para denegar a los usuarios el acceso a determinadas condicionesVPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:SubnetIds` condición. El siguiente ejemplo deniega a los usuarios el acceso a `subnet-1` y`subnet-2`:

```
{
```

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

Para denegar a los usuarios el acceso a determinadas condicionesVPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:SecurityGroupIds` condición. El siguiente ejemplo deniega a los usuarios el acceso a `sg-1` y`sg-2`:

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

```
    }
  ]
}
```

Permita a los usuarios crear entornos con VPC configuraciones específicas

Para permitir a los usuarios acceder a determinadas condicionesVPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:VpcIds` condición. El siguiente ejemplo permite a los usuarios acceder a `vpc-1` y`vpc-2`:

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

Para permitir a los usuarios acceder a datos específicosVPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:SubnetIds` condición. El siguiente ejemplo permite a los usuarios acceder a `subnet-1` y`subnet-2`:

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

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

Para permitir a los usuarios acceder a datos específicosVPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:SecurityGroupIds` condición. El siguiente ejemplo permite a los usuarios acceder a `sg-1` y`sg-2`:

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

Seguridad para AWS CloudShell

La seguridad en la nube de Amazon Web Services (AWS) es la máxima prioridad. Como AWS cliente, usted se beneficia de una arquitectura de centro de datos y red diseñada para cumplir con los requisitos de las organizaciones más sensibles a la seguridad. La seguridad es una responsabilidad compartida entre usted AWS y usted. En el modelo de responsabilidad compartida, se habla de "seguridad de la nube" y "seguridad en la nube":

Seguridad de la nube: AWS se encarga de proteger la infraestructura en la que se ejecutan todos los servicios que se ofrecen en la AWS nube y de proporcionarle servicios que pueda utilizar de forma segura. Nuestra responsabilidad en materia de seguridad es nuestra máxima prioridad AWS, y auditores externos comprueban y verifican periódicamente la eficacia de nuestra seguridad como parte de los programas de AWS conformidad.

Seguridad en la nube: su responsabilidad viene determinada por el AWS servicio que utilice y otros factores, como la confidencialidad de sus datos, los requisitos de su organización y las leyes y reglamentos aplicables.

AWS CloudShell sigue el modelo de responsabilidad compartida a través de los AWS servicios específicos que respalda. Para obtener información sobre la seguridad de los AWS servicios, consulte la página de documentación sobre la seguridad del AWS servicio y AWS los servicios que se encuentran dentro del ámbito de aplicación de AWS los programas de cumplimiento.

En los temas siguientes, se muestra cómo configurarlo AWS CloudShell para cumplir sus objetivos de seguridad y conformidad.

Temas

Protección de datos en AWS CloudShell

El modelo de responsabilidad AWS compartida modelo se aplica a la protección de datos en AWS CloudShell. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta todos los Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Usted también es responsable de las tareas de administración y configuración de seguridad para los Servicios de AWS que utiliza. Para obtener más información sobre la privacidad de los datos, consulte la sección Privacidad de datos FAQ. Para obtener información sobre la protección de datos en Europa, consulte el modelo de responsabilidad AWS compartida y la entrada del GDPR blog sobre AWS seguridad.

Para proteger los datos, le recomendamos que proteja Cuenta de AWS las credenciales y configure los usuarios individuales con AWS IAM Identity Center o AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir sus obligaciones laborales. También recomendamos proteger sus datos de la siguiente manera:

- Utilice la autenticación multifactorial (MFA) con cada cuenta.
- UseSSL/TLSpara comunicarse con AWS los recursos. Necesitamos TLS 1.2 y recomendamos TLS 1.3.
- Configure API y registre la actividad del usuario con AWS CloudTrail. Para obtener información sobre el uso de CloudTrail senderos para capturar AWS actividades, consulte Cómo trabajar con CloudTrail senderos en la Guía del AWS CloudTrail usuario.
- Utilice soluciones de AWS cifrado, junto con todos los controles de seguridad predeterminados Servicios de AWS.
- Utilice servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita entre FIPS 140 y 3 módulos criptográficos validados para acceder a AWS través de una interfaz de línea de comandos o unaAPI, utilice un FIPS terminal. Para obtener más información sobre los FIPS puntos finales disponibles, consulte la Norma federal de procesamiento de información () FIPS 140-3.

Se recomienda encarecidamente no introducir nunca información confidencial o sensible, como, por ejemplo, direcciones de correo electrónico de clientes, en etiquetas o campos de formato libre, tales como el campo Nombre. Esto incluye cuando trabaja con AWS CloudShell o Servicios de AWS utiliza la consola, API AWS CLI, o. AWS SDKs Cualquier dato que ingrese en etiquetas o campos de formato libre utilizados para nombres se puede emplear para los registros de facturación

o diagnóstico. Si proporciona una URL a un servidor externo, le recomendamos encarecidamente que no incluya la información sobre las credenciales URL para validar la solicitud a ese servidor.

Cifrado de datos

El cifrado de datos se refiere a la protección de los datos cuando están en reposo, mientras están almacenados AWS CloudShell y cuando están en tránsito entre los puntos de conexión AWS CloudShell y los puntos de servicio.

Cifrado en reposo mediante AWS KMS

El cifrado en reposo hace referencia a la protección de sus datos del acceso no autorizado mediante el cifrado de datos mientras están almacenados. Al usarlo AWS CloudShell, dispones de un almacenamiento persistente de 1 GB por AWS región sin coste alguno. El almacenamiento persistente se encuentra en su directorio principal (`$HOME`) y es privado para usted. A diferencia de los recursos efímeros del entorno que se reciclan al finalizar cada sesión del intérprete de comandos, los datos del directorio principal persisten.

El cifrado de los datos almacenados en AWS CloudShell se implementa mediante claves criptográficas proporcionadas por AWS Key Management Service (AWS KMS). Se trata de un AWS servicio gestionado para crear y controlar las claves maestras de los clientes (CMKs), es decir, las claves de cifrado que se utilizan para cifrar los datos de los clientes que se almacenan en el AWS CloudShell entorno. AWS CloudShell genera y administra claves criptográficas para cifrar los datos en nombre de los clientes.

Cifrado en tránsito

El cifrado en tránsito se refiere a proteger sus datos de ser interceptados mientras se mueven entre los extremos de comunicación.

De forma predeterminada, todas las comunicaciones de datos entre el ordenador navegador web del cliente y el ordenador basado en la nube AWS CloudShell se cifran enviándolas a través de una conexiónHTTPS/TLS.

No necesita hacer nada para habilitar el uso deHTTPS/TLSpara la comunicación.

Identity and Access Management para AWS CloudShell

AWS Identity and Access Management (IAM) es una Servicio de AWS que ayuda al administrador a controlar de forma segura el acceso a AWS los recursos. IAMlos administradores controlan quién puede autenticarse (iniciar sesión) y quién puede autorizarse (tener permisos) para usar CloudShell los recursos. IAMes una Servicio de AWS que puede utilizar sin coste adicional.

Temas

Público

La forma de usar AWS Identity and Access Management (IAM) varía según el trabajo en el que se realice CloudShell.

Usuario del servicio: si utiliza el CloudShell servicio para realizar su trabajo, el administrador le proporcionará las credenciales y los permisos que necesita. A medida que vaya utilizando más CloudShell funciones para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarlo a solicitar los permisos correctos al administrador. Si no puede acceder a una función en CloudShell, consulteSolución de problemas de identidad y acceso en AWS CloudShell .

Administrador de servicios: si está a cargo de CloudShell los recursos de su empresa, probablemente tenga acceso total a ellos CloudShell. Su trabajo consiste en determinar a qué CloudShell funciones y recursos deben acceder los usuarios del servicio. A continuación, debe enviar solicitudes a su IAM administrador para cambiar los permisos de los usuarios del servicio. Revise la información de esta página para comprender los conceptos básicos deIAM. Para obtener más información sobre cómo su empresa puede utilizar IAM con CloudShell, consulte¿Cómo AWS CloudShell funciona con IAM.

IAMadministrador: si es IAM administrador, puede que desee obtener más información sobre cómo puede redactar políticas para administrar el acceso a ellas CloudShell. Para ver ejemplos de políticas

CloudShell basadas en la identidad que puede utilizarIAM, consulte. Ejemplos de políticas basadas en la identidad para AWS CloudShell

Autenticación con identidades

La autenticación es la forma de iniciar sesión AWS con sus credenciales de identidad. Debe estar autenticado (con quien haya iniciado sesión AWS) como IAM usuario o asumiendo un IAM rol. Usuario raíz de la cuenta de AWS

Puede iniciar sesión AWS como una identidad federada mediante las credenciales proporcionadas a través de una fuente de identidad. AWS IAM Identity Center Los usuarios (IAMIdentity Center), la autenticación de inicio de sesión único de su empresa y sus credenciales de Google o Facebook son ejemplos de identidades federadas. Al iniciar sesión como una identidad federada, el administrador configuró previamente la federación de identidades mediante roles. IAM Cuando accede AWS mediante la federación, asume indirectamente un rol.

Según el tipo de usuario que sea, puede iniciar sesión en el portal AWS Management Console o en el de AWS acceso. Para obtener más información sobre cómo iniciar sesión AWS, consulte Cómo iniciar sesión Cuenta de AWS en su Guía del AWS Sign-In usuario.

Si accede AWS mediante programación, AWS incluye un kit de desarrollo de software (SDK) y una interfaz de línea de comandos (CLI) para firmar criptográficamente sus solicitudes con sus credenciales. Si no utilizas AWS herramientas, debes firmar las solicitudes tú mismo. Para obtener más información sobre cómo usar el método recomendado para firmar las solicitudes usted mismo, consulte Firmar AWS API las solicitudes en la Guía del IAM usuario.

Independientemente del método de autenticación que use, es posible que deba proporcionar información de seguridad adicional. Por ejemplo, le AWS recomienda que utilice la autenticación multifactorial (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte Autenticación multifactorial en la Guía del AWS IAM Identity Center usuario y Uso de la autenticación multifactorial (MFA) AWS en la Guía del IAM usuario.

Cuenta de AWS usuario root

Al crear una Cuenta de AWS, comienza con una identidad de inicio de sesión que tiene acceso completo a todos Servicios de AWS los recursos de la cuenta. Esta identidad se denomina usuario Cuenta de AWS raíz y se accede a ella iniciando sesión con la dirección de correo electrónico y la contraseña que utilizaste para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Proteja las credenciales del usuario raíz y utilícelas solo para las tareas que solo el usuario raíz pueda realizar. Para ver la lista completa de tareas que requieren

que inicie sesión como usuario root, consulte Tareas que requieren credenciales de usuario root en la Guía del IAM usuario.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos, incluidos los que requieren acceso de administrador, que utilicen la federación con un proveedor de identidades para acceder Servicios de AWS mediante credenciales temporales.

Una identidad federada es un usuario del directorio de usuarios empresarial, un proveedor de identidades web AWS Directory Service, el directorio del Centro de Identidad o cualquier usuario al que acceda Servicios de AWS mediante las credenciales proporcionadas a través de una fuente de identidad. Cuando las identidades federadas acceden Cuentas de AWS, asumen funciones y las funciones proporcionan credenciales temporales.

Para una administración de acceso centralizada, le recomendamos que utilice AWS IAM Identity Center. Puede crear usuarios y grupos en IAM Identity Center, o puede conectarse y sincronizarse con un conjunto de usuarios y grupos de su propia fuente de identidad para usarlos en todas sus aplicaciones Cuentas de AWS . Para obtener información sobre IAM Identity Center, consulte ¿Qué es IAM Identity Center? en la Guía AWS IAM Identity Center del usuario.

Usuarios y grupos de IAM

Un IAMusuario es una identidad propia Cuenta de AWS que tiene permisos específicos para una sola persona o aplicación. Siempre que sea posible, recomendamos utilizar credenciales temporales en lugar de crear IAM usuarios con credenciales de larga duración, como contraseñas y claves de acceso. Sin embargo, si tiene casos de uso específicos que requieren credenciales a largo plazo con IAM los usuarios, le recomendamos que rote las claves de acceso. Para obtener más información, consulte Rotar las claves de acceso con regularidad para los casos de uso que requieran credenciales de larga duración en la Guía del IAM usuario.

Un IAMgrupo es una identidad que especifica un conjunto de IAM usuarios. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos para grandes conjuntos de usuarios. Por ejemplo, puede asignar un nombre a un grupo IAMAdminsy concederle permisos para administrar IAM los recursos.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales de larga duración permanentes; no obstante, los roles proporcionan credenciales

temporales. Para obtener más información, consulte Cuándo crear un IAM usuario (en lugar de un rol) en la Guía del IAM usuario.

IAMroles

Un IAMrol es una identidad dentro de ti Cuenta de AWS que tiene permisos específicos. Es similar a un IAM usuario, pero no está asociado a una persona específica. Puede asumir temporalmente un IAM rol en el AWS Management Console cambiando de rol. Puede asumir un rol llamando a una AWS API operación AWS CLI o o utilizando una operación personalizadaURL. Para obtener más información sobre los métodos de uso de roles, consulte Uso de IAM roles en la Guía del IAM usuario.

IAMlos roles con credenciales temporales son útiles en las siguientes situaciones:

- Acceso de usuario federado: para asignar permisos a una identidad federada, puede crear un rol y definir sus permisos. Cuando se autentica una identidad federada, se asocia la identidad al rol y se le conceden los permisos define el rol. Para obtener información sobre los roles para la federación, consulte Creación de un rol para un proveedor de identidad externo en la Guía del IAM usuario. Si usa IAM Identity Center, configura un conjunto de permisos. Para controlar a qué pueden acceder sus identidades después de autenticarse, IAM Identity Center correlaciona el conjunto de permisos con un rol en. IAM Para obtener información acerca de los conjuntos de permisos, consulte Conjuntos de permisos en la Guía del usuario de AWS IAM Identity Center .
- Permisos IAM de usuario temporales: un IAM usuario o rol puede asumir un IAM rol para asumir temporalmente diferentes permisos para una tarea específica.
- Acceso multicuenta: puedes usar un IAM rol para permitir que alguien (un responsable de confianza) de una cuenta diferente acceda a los recursos de tu cuenta. Los roles son la forma principal de conceder acceso entre cuentas. Sin embargo, con algunos Servicios de AWS, puedes adjuntar una política directamente a un recurso (en lugar de usar un rol como proxy). Para conocer la diferencia entre las funciones y las políticas basadas en recursos para el acceso multicuenta, consulta el tema sobre el acceso a los recursos entre cuentas IAM en la Guía del IAM usuario.
- Acceso entre servicios: algunos Servicios de AWS utilizan funciones en otros. Servicios de AWS Por ejemplo, cuando realizas una llamada en un servicio, es habitual que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado al servicio.
  - Sesiones de acceso directo (FAS): cuando utilizas un IAM usuario o un rol para realizar acciones en AWS ellas, se te considera director. Cuando utiliza algunos servicios, es posible que realice

una acción que desencadene otra acción en un servicio diferente. FASutiliza los permisos del principal que llama a an Servicio de AWS, junto con los que solicitan, Servicio de AWS para realizar solicitudes a los servicios descendentes. FASlas solicitudes solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros recursos Servicios de AWS o para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información detallada sobre la política a la hora de realizar FAS solicitudes, consulte Reenviar las sesiones de acceso.

- Función de servicio: una función de servicio es una IAMfunción que un servicio asume para realizar acciones en su nombre. Un IAM administrador puede crear, modificar y eliminar un rol de servicio desde dentroIAM. Para obtener más información, consulte Crear un rol para delegar permisos Servicio de AWS en un rol en el IAMManual del usuario.
- Función vinculada a un servicio: una función vinculada a un servicio es un tipo de función de servicio que está vinculada a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un IAM administrador puede ver los permisos de los roles vinculados al servicio, pero no editarlos.

- Aplicaciones que se ejecutan en Amazon EC2: puedes usar un IAM rol para administrar las credenciales temporales de las aplicaciones que se ejecutan en una EC2 instancia y que realizan AWS CLI o AWS API solicitan. Esto es preferible a almacenar las claves de acceso dentro de la EC2 instancia. Para asignar un AWS rol a una EC2 instancia y ponerlo a disposición de todas sus aplicaciones, debe crear un perfil de instancia adjunto a la instancia. Un perfil de instancia contiene el rol y permite que los programas que se ejecutan en la EC2 instancia obtengan credenciales temporales. Para obtener más información, consulte Uso de un IAM rol para conceder permisos a aplicaciones que se ejecutan en EC2 instancias de Amazon en la Guía del IAM usuario.

Para saber si se deben usar IAM roles o IAM usuarios, consulte Cuándo crear un IAM rol (en lugar de un usuario) en la Guía del IAM usuario.

Administración de acceso mediante políticas

El acceso se controla AWS creando políticas y adjuntándolas a AWS identidades o recursos. Una política es un objeto AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. AWS evalúa estas políticas cuando un director (usuario, usuario raíz o sesión de rol) realiza una solicitud. Los permisos en las políticas determinan si la solicitud se permite o se deniega. La mayoría de las políticas se almacenan AWS como JSON documentos. Para obtener más información sobre

la estructura y el contenido de los documentos de JSON políticas, consulte Descripción general de JSON las políticas en la Guía del IAM usuario.

Los administradores pueden usar AWS JSON las políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Para conceder a los usuarios permiso para realizar acciones en los recursos que necesitan, un IAM administrador puede crear IAM políticas. A continuación, el administrador puede añadir las IAM políticas a las funciones y los usuarios pueden asumir las funciones.

IAMlas políticas definen los permisos para una acción independientemente del método que se utilice para realizar la operación. Por ejemplo, suponga que dispone de una política que permite la acción `iam:GetRole`. Un usuario con esa política puede obtener información sobre el rol de AWS Management Console AWS CLI, el o el AWS API.

Políticas basadas en identidad

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener información sobre cómo crear una política basada en la identidad, consulte Creación de IAM políticas en la Guía del usuario. IAM

Las políticas basadas en identidades pueden clasificarse además como políticas insertadas o políticas administradas. Las políticas insertadas se integran directamente en un único usuario, grupo o rol. Las políticas administradas son políticas independientes que puede adjuntar a varios usuarios, grupos y funciones de su empresa. Cuenta de AWS Las políticas administradas incluyen políticas AWS administradas y políticas administradas por el cliente. Para saber cómo elegir entre una política gestionada o una política integrada, consulte Elegir entre políticas gestionadas y políticas integradas en la Guía del IAM usuario.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de JSON política que se adjuntan a un recurso. Algunos ejemplos de políticas basadas en recursos son las políticas de confianza de IAM roles y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad

principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puede usar políticas AWS administradas desde una política IAM basada en recursos.

Listas de control de acceso () ACLs

Las listas de control de acceso (ACLs) controlan qué responsables (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLsson similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de JSON políticas.

Amazon S3 AWS WAF y Amazon VPC son ejemplos de servicios compatiblesACLs. Para obtener más informaciónACLs, consulte la descripción general de la lista de control de acceso (ACL) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite tipos de políticas adicionales y menos comunes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le conceden.

- Límites de permisos: un límite de permisos es una función avanzada en la que se establecen los permisos máximos que una política basada en la identidad puede conceder a una IAM entidad (IAMusuario o rol). Puede establecer un límite de permisos para una entidad. Los permisos resultantes son la intersección de las políticas basadas en la identidad de la entidad y los límites de permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo `Principal` no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte los límites de los permisos para IAM las entidades en la Guía del IAMusuario.
- Políticas de control de servicios (SCPs): SCPs son JSON políticas que especifican los permisos máximos para una organización o unidad organizativa (OU) AWS Organizations. AWS Organizations es un servicio para agrupar y administrar de forma centralizada varios de los Cuentas de AWS que son propiedad de su empresa. Si habilitas todas las funciones de una organización, puedes aplicar políticas de control de servicios (SCPs) a una o a todas tus cuentas. SCPLimita los permisos de las entidades en las cuentas de los miembros, incluidas las de cada una Usuario raíz de la cuenta de AWS. Para obtener más información sobre OrganizationsSCPs, consulte las políticas de control de servicios en la Guía del AWS Organizations usuario.

- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidades del rol y las políticas de la sesión. Los permisos también pueden proceder de una política en función de recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información, consulte las políticas de sesión en la Guía del IAM usuario.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para saber cómo se AWS determina si se debe permitir una solicitud cuando se trata de varios tipos de políticas, consulte la lógica de evaluación de políticas en la Guía del IAM usuario.

¿Cómo AWS CloudShell funciona con IAM

Antes de utilizar IAM para administrar el acceso a CloudShell, infórmate sobre las IAM funciones disponibles para su uso CloudShell.

IAMfunciones que puedes usar con AWS CloudShell

IAMcaracterística	CloudShell apoyo
Políticas basadas en identidades	Sí
Políticas basadas en recursos	No
Acciones de políticas	Sí
Recursos de políticas	Sí
Claves de condición de política (específicas del servicio)	Sí
ACLs	No
ABAC(etiquetas en las políticas)	No
Credenciales temporales	Sí

IAMcaracterística	CloudShell apoyo
Sesiones de acceso directo (FAS)	No
Roles de servicio	No
Roles vinculados al servicio	No

Para obtener una visión general de cómo CloudShell funcionan otros AWS servicios con la mayoría de las IAM funciones, consulte AWS los servicios con los que funcionan IAM en la Guía del IAM usuario.

Políticas basadas en la identidad para CloudShell

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en la identidad son documentos de política de JSON permisos que se pueden adjuntar a una identidad, como un IAM usuario, un grupo de usuarios o un rol. Estas políticas controlan qué acciones pueden realizar los usuarios y los roles, en qué recursos y en qué condiciones. Para obtener información sobre cómo crear una política basada en la identidad, consulte Creación de IAM políticas en la Guía del usuario. IAM

Con las políticas IAM basadas en la identidad, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. No es posible especificar la entidad principal en una política basada en identidad porque se aplica al usuario o rol al que está adjunto. Para obtener más información sobre todos los elementos que puede utilizar en una JSON política, consulte la referencia sobre los elementos de la IAM JSON política en la Guía del IAMusuario.

Ejemplos de políticas basadas en la identidad para CloudShell

Para ver ejemplos de políticas CloudShell basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS CloudShell

Políticas basadas en recursos dentro de CloudShell

Admite políticas basadas en recursos: no

Las políticas basadas en recursos son documentos de JSON política que se adjuntan a un recurso. Algunos ejemplos de políticas basadas en recursos son las políticas de confianza de IAM roles y

las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se asocia la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe especificar una entidad principal en una política en función de recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar una cuenta completa o IAM entidades de otra cuenta como principales en una política basada en recursos. Añadir a una política en función de recursos una entidad principal entre cuentas es solo una parte del establecimiento de una relación de confianza. Cuando el principal y el recurso son diferentes Cuentas de AWS, el IAM administrador de la cuenta de confianza también debe conceder permiso a la entidad principal (usuario o rol) para acceder al recurso. Para conceder el permiso, adjunte la entidad a una política basada en identidad. Sin embargo, si la política en función de recursos concede el acceso a una entidad principal de la misma cuenta, no es necesaria una política basada en identidad adicional. Para obtener más información, consulte el tema Acceso a recursos entre cuentas IAM en la Guía del IAM usuario.

Acciones políticas para CloudShell

Compatibilidad con las acciones de política: sí

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El `Action` elemento de una JSON política describe las acciones que puede utilizar para permitir o denegar el acceso en una política. Las acciones de política suelen tener el mismo nombre que la AWS API operación asociada. Hay algunas excepciones, como las acciones que solo permiten permisos y que no tienen una operación coincidente. API También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

Para ver una lista de CloudShell acciones, consulte las acciones definidas AWS CloudShell en la Referencia de autorización de servicios. Algunas acciones pueden tener más de unaAPI.

Las acciones políticas CloudShell utilizan el siguiente prefijo antes de la acción:

```
cloudshell
```

Para especificar varias acciones en una única instrucción, sepárelas con comas.

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

Para ver ejemplos de políticas CloudShell basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS CloudShell

Recursos de políticas para CloudShell

Compatibilidad con los recursos de políticas: sí

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Resource` JSON de política especifica el objeto o los objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso mediante su nombre de recurso de Amazon (ARN). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

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

Para ver una lista de los tipos de CloudShell recursos y sus respectivos tiposARNs, consulte los recursos definidos AWS CloudShell en la Referencia de autorización de servicio. Para saber con qué acciones puede especificar cada recurso, consulte Acciones definidas por AWS CloudShell. ARN

Para ver ejemplos de políticas CloudShell basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS CloudShell

Claves de condición de la política para CloudShell

Compatibilidad con claves de condición de políticas específicas del servicio: sí

Los administradores pueden usar AWS JSON políticas para especificar quién tiene acceso a qué. Es decir, qué entidad principal puede realizar acciones en qué recursos y en qué condiciones.

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación `AND` lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición mediante una `OR` operación lógica. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder a un IAM usuario permiso para acceder a un recurso solo si está etiquetado con su nombre de IAM usuario. Para obtener más información, consulte los elementos IAM de la política: variables y etiquetas en la Guía del IAM usuario.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del IAMusuario.

Para ver una lista de claves de CloudShell condición, consulte las claves de condición AWS CloudShell en la Referencia de autorización de servicio. Para saber con qué acciones y recursos puede utilizar una clave de condición, consulte Acciones definidas por AWS CloudShell.

Para ver ejemplos de políticas CloudShell basadas en la identidad, consulte. Ejemplos de políticas basadas en la identidad para AWS CloudShell

ACLsen CloudShell

SoportesACLs: No

Las listas de control de acceso (ACLs) controlan qué directores (miembros de la cuenta, usuarios o roles) tienen permisos para acceder a un recurso. ACLsson similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de JSON políticas.

ABACcon CloudShell

Soportes ABAC (etiquetas en las políticas): No

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define los permisos en función de los atributos. En AWS, estos atributos se denominan etiquetas. Puede adjuntar etiquetas a IAM entidades (usuarios o roles) y a muchos AWS recursos. Etiquetar entidades y recursos es el primer paso deABAC. Luego, diseñe ABAC políticas para permitir las operaciones cuando la etiqueta del principal coincida con la etiqueta del recurso al que está intentando acceder.

ABACes útil en entornos de rápido crecimiento y ayuda en situaciones en las que la administración de políticas se vuelve engorrosa.

Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el elemento de condición de una política utilizando las claves de condición `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Si un servicio admite las tres claves de condición para cada tipo de recurso, el valor es Sí para el servicio. Si un servicio admite las tres claves de condición solo para algunos tipos de recursos, el valor es Parcial.

Para obtener más información al respectoABAC, consulte ¿Qué es? ABAC en la Guía IAM del usuario. Para ver un tutorial con los pasos de configuraciónABAC, consulte Usar el control de acceso basado en atributos (ABAC) en la Guía del IAMusuario.

Uso de credenciales temporales con CloudShell

Compatibilidad con credenciales temporales: sí

Algunas Servicios de AWS no funcionan cuando inicias sesión con credenciales temporales. Para obtener información adicional, incluida la información sobre cuáles Servicios de AWS funcionan con credenciales temporales, consulta la sección Servicios de AWS Cómo trabajar con credenciales temporales IAM en la Guía del IAM usuario.

Está utilizando credenciales temporales si inicia sesión en ellas AWS Management Console mediante cualquier método excepto un nombre de usuario y una contraseña. Por ejemplo, cuando accedes AWS mediante el enlace de inicio de sesión único (SSO) de tu empresa, ese proceso crea automáticamente credenciales temporales. También crea credenciales temporales de forma automática cuando inicia sesión en la consola como usuario y luego cambia de rol. Para obtener más información sobre el cambio de rol, consulte Cambiar a un rol (consola) en la Guía del IAMusuario.

Puede crear credenciales temporales manualmente con la tecla AWS CLI o AWS API. A continuación, puede utilizar esas credenciales temporales para acceder AWS. AWS recomienda generar credenciales temporales de forma dinámica en lugar de utilizar claves de acceso a largo plazo. Para obtener más información, consulte Credenciales de seguridad temporales en IAM.

Al cambiar de rol, utilizará un entorno diferente. No puede cambiar de rol dentro del mismo AWS CloudShell entorno.

Sesiones de acceso directo para CloudShell

Admite sesiones de acceso directo (FAS): No

Cuando utilizas un IAM usuario o un rol para realizar acciones en AWSél, se te considera director. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. FASutiliza los permisos del principal que llama a una Servicio de AWS, junto con los que solicita, Servicio de AWS para realizar solicitudes a los servicios descendentes. FASlas solicitudes solo se realizan cuando un servicio recibe una solicitud que requiere interacciones con otros recursos Servicios de AWS o para completarse. En este caso, debe tener permisos para realizar ambas acciones. Para obtener información detallada sobre la política a la hora de realizar FAS solicitudes, consulte Reenviar las sesiones de acceso.

Roles de servicio para CloudShell

Compatible con roles de servicio: No

Una función de servicio es una IAMfunción que un servicio asume para realizar acciones en su nombre. Un IAM administrador puede crear, modificar y eliminar un rol de servicio desde dentroIAM. Para obtener más información, consulte Crear un rol para delegar permisos Servicio de AWS en un rol en el IAMManual del usuario.

> **Warning**
>
> Cambiar los permisos de un rol de servicio podría afectar a CloudShell la funcionalidad. Edite las funciones de servicio solo cuando se CloudShell proporcionen instrucciones para hacerlo.

Funciones vinculadas al servicio para CloudShell

Compatibilidad con roles vinculados al servicio: no

Un rol vinculado a un servicio es un tipo de rol de servicio que está vinculado a un. Servicio de AWS El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados al servicio aparecen en usted Cuenta de AWS y son propiedad del servicio. Un IAM administrador puede ver los permisos de los roles vinculados al servicio, pero no editarlos.

Ejemplos de políticas basadas en la identidad para AWS CloudShell

De forma predeterminada, los usuarios y los roles no tienen permiso para crear o modificar CloudShell recursos. Tampoco pueden realizar tareas con AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS API. Para conceder a los usuarios permiso para realizar acciones en los recursos que necesitan, un IAM administrador puede crear IAM políticas. A continuación, el administrador puede añadir las IAM políticas a las funciones y los usuarios pueden asumir las funciones.

Para obtener información sobre cómo crear una política IAM basada en la identidad mediante estos documentos de JSON política de ejemplo, consulte Creación de IAM políticas en la Guía del IAMusuario.

Para obtener más información sobre las acciones y los tipos de recursos definidos CloudShell, incluido el formato de cada uno de los tipos de recursos, consulte las claves de condición, recursos y acciones de la Referencia AWS CloudShell de autorización de servicios. ARNs

Temas

- Prácticas recomendadas sobre las políticas
- Uso de la CloudShell consola
- Cómo permitir a los usuarios consultar sus propios permisos

Prácticas recomendadas sobre las políticas

Las políticas basadas en la identidad determinan si alguien puede crear CloudShell recursos de tu cuenta, acceder a ellos o eliminarlos. Estas acciones pueden generar costos adicionales para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comience con las políticas AWS administradas y avance hacia los permisos con privilegios mínimos: para empezar a conceder permisos a sus usuarios y cargas de trabajo, utilice las políticas AWS administradas que otorgan permisos para muchos casos de uso comunes. Están

disponibles en su. Cuenta de AWS Le recomendamos que reduzca aún más los permisos definiendo políticas administradas por el AWS cliente que sean específicas para sus casos de uso. Para obtener más información, consulte las políticas AWS gestionadas o las políticas AWS gestionadas para las funciones laborales en la Guía del IAM usuario.

- Aplique permisos con privilegios mínimos: cuando establezca permisos con IAM políticas, conceda solo los permisos necesarios para realizar una tarea. Para ello, debe definir las acciones que se pueden llevar a cabo en determinados recursos en condiciones específicas, también conocidos como permisos de privilegios mínimos. Para obtener más información sobre cómo IAM aplicar permisos, consulte Políticas y permisos IAM en la IAM Guía del usuario.
- Utilice las condiciones en IAM las políticas para restringir aún más el acceso: puede añadir una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de política para especificar que todas las solicitudes deben enviarse medianteSSL. También puedes usar condiciones para conceder el acceso a las acciones del servicio si se utilizan a través de una acción específica Servicio de AWS, por ejemplo AWS CloudFormation. Para obtener más información, consulte los elementos IAM JSON de la política: Condición en la Guía del IAM usuario.
- Utilice IAM Access Analyzer para validar sus IAM políticas y garantizar permisos seguros y funcionales: IAM Access Analyzer valida las políticas nuevas y existentes para que se ajusten al lenguaje de las políticas (JSON) y IAM a las IAM mejores prácticas. IAMAccess Analyzer proporciona más de 100 comprobaciones de políticas y recomendaciones prácticas para ayudarle a crear políticas seguras y funcionales. Para obtener más información, consulte la validación de políticas de IAM Access Analyzer en la Guía del IAM usuario.
- Requerir autenticación multifactorial (MFA): si se encuentra en una situación en la que se requieren IAM usuarios o un usuario raíz Cuenta de AWS, actívela MFA para aumentar la seguridad. Para solicitarlo MFA cuando se convoque a API las operaciones, añada MFA condiciones a sus políticas. Para obtener más información, consulte Configuración del API acceso MFA protegido en la Guía del IAM usuario.

Para obtener más información sobre las prácticas recomendadasIAM, consulte las prácticas recomendadas de seguridad IAM en la Guía del IAM usuario.

Uso de la CloudShell consola

Para acceder a la AWS CloudShell consola, debe tener un conjunto mínimo de permisos. Estos permisos deben permitirle enumerar y ver detalles sobre los CloudShell recursos de su cuenta Cuenta de AWS. Si crea una política basada en identidades que sea más restrictiva que el mínimo

de permisos necesarios, la consola no funcionará del modo esperado para las entidades (usuarios o roles) que tengan esa política.

No es necesario que concedas permisos mínimos de consola a los usuarios que realicen llamadas únicamente al AWS CLI o al AWS API. En su lugar, permita el acceso únicamente a las acciones que coincidan con la API operación que están intentando realizar.

Para garantizar que los usuarios y los roles puedan seguir utilizando la CloudShell consola, adjunte también la política *ReadOnly* AWS gestionada CloudShell *ConsoleAccess* o la política gestionada a las entidades. Para obtener más información, consulte Añadir permisos a un usuario en la Guía del IAM usuario.

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo se muestra cómo se puede crear una política que permita a IAM los usuarios ver las políticas integradas y administradas asociadas a su identidad de usuario. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la tecla o. AWS CLI AWS API

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

Solución de problemas de identidad y acceso en AWS CloudShell

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas más comunes que pueden surgir al trabajar con CloudShell yIAM.

Temas

- No estoy autorizado a realizar ninguna acción en CloudShell
- No estoy autorizado a realizar tareas como: PassRole
- Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis CloudShell recursos

No estoy autorizado a realizar ninguna acción en CloudShell

Si recibe un error que indica que no tiene autorización para realizar una acción, las políticas se deben actualizar para permitirle realizar la acción.

El siguiente ejemplo de error se produce cuando el `mateojackson` IAM usuario intenta usar la consola para ver los detalles de un *`my-example-widget`* recurso ficticio, pero no tiene los `awes:`*`GetWidget`* permisos ficticios.

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

En este caso, la política del usuario `mateojackson` debe actualizarse para permitir el acceso al recurso *`my-example-widget`* mediante la acción `awes:`*`GetWidget`*.

Si necesita ayuda, póngase en contacto con AWS el administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

No estoy autorizado a realizar tareas como: PassRole

Si recibes un mensaje de error que indica que no estás autorizado a realizar la `iam:PassRole` acción, debes actualizar tus políticas para que puedas transferirle CloudShell una función.

Algunas Servicios de AWS permiten transferir una función existente a ese servicio en lugar de crear una nueva función de servicio o una función vinculada al servicio. Para ello, debe tener permisos para transferir el rol al servicio.

El siguiente ejemplo de error se produce cuando un IAM usuario denominado `marymajor` intenta utilizar la consola para realizar una acción en ella. CloudShell Sin embargo, la acción requiere que el servicio cuente con permisos que otorguen un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

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

En este caso, las políticas de Mary se deben actualizar para permitirle realizar la acción `iam:PassRole`.

Si necesita ayuda, póngase en contacto con AWS el administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

Quiero permitir que personas ajenas a mí accedan Cuenta de AWS a mis CloudShell recursos

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que respaldan las políticas basadas en recursos o las listas de control de acceso (ACLs), puedes usar esas políticas para permitir que las personas accedan a tus recursos.

Para más información, consulte lo siguiente:

- Para saber si CloudShell es compatible con estas funciones, consulte. ¿Cómo AWS CloudShell funciona con IAM
- Para obtener información sobre cómo proporcionar acceso a los recursos de su propiedad, consulte Proporcionar acceso a un IAM usuario en otro Cuenta de AWS de su propiedad en la Guía del IAM usuario. Cuentas de AWS

- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo permitir el acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del IAM usuario.
- Para obtener información sobre cómo proporcionar acceso mediante la federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (federación de identidades) en la Guía del IAM usuario.
- Para saber la diferencia entre el uso de roles y políticas basadas en recursos para el acceso entre cuentas, consulte el acceso a recursos entre cuentas IAM en la Guía del usuario. IAM

Administrar el AWS CloudShell acceso y el uso con políticas IAM

Con los recursos de administración de acceso que pueden proporcionar AWS Identity and Access Management, los administradores pueden conceder permisos a IAM los usuarios. De esta forma, estos usuarios pueden acceder a las funciones del entorno AWS CloudShell y utilizarlas. Los administradores también pueden crear políticas que especifiquen de forma pormenorizada qué acciones pueden realizar esos usuarios en el entorno del intérprete de comandos.

La forma más rápida para que un administrador conceda acceso a los usuarios es mediante una política AWS gestionada. Una política AWS administrada es una política independiente creada y administrada por. AWS La siguiente política AWS administrada para se AWS CloudShell puede adjuntar a IAM las identidades:

- AWS CloudShellFullAccess: Concede permiso de uso AWS CloudShell con acceso completo a todas las funciones.

La AWS CloudShellFullAccesspolítica utiliza el carácter comodín (*) para dar a la IAM identidad (usuario, rol o grupo) acceso total a las funciones CloudShell y a las funciones. Para obtener más información sobre esta política, consulte AWS CloudShellFullAccessla Guía del usuario de políticas AWS administradas.

> **Note**
>
> IAMTambién se pueden lanzar identidades con las siguientes políticas AWS administradas CloudShell. Sin embargo, estas políticas ofrecen amplios permisos. Por lo tanto, te recomendamos que solo concedas estas políticas si son esenciales para el puesto de trabajo de un IAM usuario.

- Administrador: proporciona a IAM los usuarios acceso total y les permite delegar permisos en todos los servicios y recursos incluidos AWS.
- Desarrollador y usuario avanzado: permite a IAM los usuarios realizar tareas de desarrollo de aplicaciones y crear y configurar recursos y servicios que respalden AWS el desarrollo de aplicaciones inteligentes.

Para obtener más información sobre cómo adjuntar políticas gestionadas, consulte Añadir permisos de IAM identidad (consola) en la Guía del IAM usuario.

Administrar las acciones permitidas mediante el AWS CloudShell uso de políticas personalizadas

Para administrar las acciones con las que un IAM usuario puede realizar CloudShell, cree una política personalizada que utilice la política CloudShellPolicy administrada como plantilla. Como alternativa, edite una política en línea que esté integrada en la IAM identidad correspondiente (usuario, grupo o rol).

Por ejemplo, puedes permitir el acceso de IAM los usuarios CloudShell, pero evitar que reenvíen las credenciales del CloudShell entorno que se utilizan para iniciar sesión. AWS Management Console

Important

Para iniciar AWS CloudShell desde AWS Management Console, el IAM usuario necesita permisos para realizar las siguientes acciones:

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

Si una de estas acciones no está permitida explícitamente en una política adjunta, se IAM mostrará un error de permisos al intentar iniciarla CloudShell.

AWS CloudShell permisos

Nombre	Descripción del permiso concedido	¿Necesario para el lanzamiento CloudShell?
`cloudshell:CreateEnvironment`	Crea un CloudShell entorno, recupera el diseño al inicio de la CloudShell sesión y guarda el diseño actual de la aplicación web en el servidor. Este permiso solo es * el valor `Resource` indicado en. the section called “Ejemplos de IAM políticas para CloudShell”	Sí
`cloudshell:CreateSession`	Se conecta a un CloudShell entorno desde AWS Management Console.	Sí
`cloudshell:GetEnvironmentStatus`	Lea el estado de un CloudShell entorno.	Sí
`cloudshell:DeleteEnvironment`	Elimina un CloudShell entorno.	No
`cloudshell:GetFileDownloadUrls`	Genera Amazon S3 prefirmado URLs que se utiliza para descargar archivos CloudShel	No

Nombre	Descripción del permiso concedido	¿Necesario para el lanzamiento CloudShell?
	l mediante la interfaz CloudShell web. Esto no está disponible para VPC entornos.	
`cloudshell:GetFileUploadUrls`	Genera Amazon S3 prefirmado URLs que se utiliza para cargar archivos CloudShel l mediante la interfaz CloudShell web. Esto no está disponible para VPC entornos.	No
`cloudshell:DescribeEnvironments`	Describe los entornos.	No
`cloudshell:PutCredentials`	Reenvía las credencia les utilizadas para iniciar sesión en el AWS Management Console . CloudShell	No
`cloudshell:StartEnvironment`	Inicia un CloudShel l entorno que está detenido.	Sí
`cloudshell:StopEnvironment`	Detiene un CloudShel l entorno que se está ejecutando.	No

Ejemplos de IAM políticas para CloudShell

Los siguientes ejemplos muestran cómo se pueden crear políticas para restringir quién puede acceder CloudShell. Los ejemplos también muestran las acciones que se pueden realizar en el entorno del intérprete de comandos.

La siguiente política impone una denegación total del acceso a sus funciones CloudShell y a sus funciones.

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

La siguiente política permite a IAM los usuarios acceder CloudShell , pero les impide generar archivos prefirmados URLs para cargar y descargar archivos. Los usuarios pueden seguir transfiriendo archivos hacia y desde el entorno, utilizando clientes como, por ejemplo, `wget`.

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

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

La siguiente política permite a IAM los usuarios acceder CloudShell. Sin embargo, la política impide que las credenciales que utilizó para iniciar sesión AWS Management Console se reenvíen al CloudShell entorno. IAMlos usuarios con esta política deben configurar manualmente sus credenciales en ella CloudShell.

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

La siguiente política permite a IAM los usuarios crear AWS CloudShell entornos.

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

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

IAMPermisos necesarios para crear y usar CloudShell VPC entornos

Para crear y usar CloudShell VPC entornos, el IAM administrador debe habilitar el acceso a EC2 permisos VPC específicos de Amazon. En esta sección se enumeran los EC2 permisos de Amazon necesarios para crear y usar VPC entornos.

Para crear VPC entornos, la IAM política asignada a su función debe incluir los siguientes EC2 permisos de Amazon:

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

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

Recomendamos incluir también:

- ec2:DeleteNetworkInterface

> **Note**
>
> Este permiso no es obligatorio, pero es necesario CloudShell para limpiar el ENI recurso (ENIscreado para CloudShell VPC entornos que se etiquetan con una ManagedByCloudShell clave) creado por él. Si este permiso no está habilitado, debe limpiar el ENI recurso manualmente después de cada uso del CloudShell VPC entorno.

IAMpolítica que otorga CloudShell acceso completo, incluido el acceso a VPC

El siguiente ejemplo muestra cómo habilitar todos los permisos, incluido el VPC acceso a CloudShell:

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

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

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

Uso de claves de IAM condición para VPC entornos

Puede usar claves de condición CloudShell específicas en la VPC configuración a fin de proporcionar controles de permisos adicionales para sus VPC entornos. También puede especificar las subredes y los grupos de seguridad que el VPC entorno puede y no puede usar.

CloudShell admite las siguientes claves de condición en las IAM políticas:

- `CloudShell:VpcIds`— Permitir o denegar una o más VPCs
- `CloudShell:SubnetIds`— Permitir o denegar una o más subredes
- `CloudShell:SecurityGroupIds`— Permitir o denegar uno o más grupos de seguridad

> **Note**
>
> Si los permisos de los usuarios con acceso a CloudShell entornos públicos se modifican para añadir restricciones a la `cloudshell:createEnvironment` acción, podrán seguir accediendo a su entorno público actual. Sin embargo, si desea modificar una IAM política con esta restricción e inhabilitar su acceso al entorno público existente, primero debe actualizar la IAM política con la restricción y, a continuación, asegurarse de que todos los CloudShell usuarios de su cuenta eliminen manualmente el entorno público existente mediante la interfaz de usuario CloudShell web (Acciones → Eliminar CloudShell entorno).

Ejemplos de políticas con claves de condición para la configuración VPC

Los siguientes ejemplos muestran cómo utilizar las claves de condición para la VPC configuración. Después de crear una instrucción de política con las restricciones deseadas, agregue la instrucción de política para el usuario o rol de destino.

Asegúrese de que los usuarios creen únicamente VPC entornos y denieguen la creación de entornos públicos

Para garantizar que los usuarios solo puedan crear VPC entornos, utilice el permiso de denegación como se muestra en el siguiente ejemplo:

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

Denegue a los usuarios el acceso a subredes o grupos de seguridad específicos VPCs

Para denegar a los usuarios el acceso a un VPCs contenido específico, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:VpcIds` condición. El siguiente ejemplo deniega a los usuarios el acceso a `vpc-1` y`vpc-2`:

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

```
            "vpc-1",
            "vpc-2"
          ]
        }
      }
    }
  ]
}
```

Para denegar a los usuarios el acceso a determinadas condicionesVPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:SubnetIds` condición. El siguiente ejemplo deniega a los usuarios el acceso a `subnet-1` y`subnet-2`:

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

Para denegar a los usuarios el acceso a determinadas condicionesVPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:SecurityGroupIds` condición. El siguiente ejemplo deniega a los usuarios el acceso a `sg-1` y`sg-2`:

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

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

Permita a los usuarios crear entornos con VPC configuraciones específicas

Para permitir a los usuarios acceder a determinadas condicionesVPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:VpcIds` condición. El siguiente ejemplo permite a los usuarios acceder a `vpc-1` y`vpc-2`:

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

```
    }
  ]
}
```

Para permitir a los usuarios acceder a datos específicosVPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:SubnetIds` condición. El siguiente ejemplo permite a los usuarios acceder a `subnet-1` y`subnet-2`:

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

Para permitir a los usuarios acceder a datos específicosVPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:SecurityGroupIds` condición. El siguiente ejemplo permite a los usuarios acceder a `sg-1` y`sg-2`:

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

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

Permisos de acceso Servicios de AWS

CloudShell utiliza las IAM credenciales que utilizó para iniciar sesión en AWS Management Console.

> **Note**
>
> Para utilizar las IAM credenciales que utilizó para iniciar sesión en AWS Management Console, debe tener `cloudshell:PutCredentials` permiso.

Esta función de autenticación previa CloudShell hace que sea cómoda de usar AWS CLI. Sin embargo, IAM el usuario sigue necesitando permisos explícitos para Servicios de AWS las llamadas desde la línea de comandos.

Por ejemplo, supongamos que IAM los usuarios deben crear buckets de Amazon S3 y cargarles archivos como objetos. Puede crear una política que permita esas acciones de forma explícita. La IAM consola proporciona un editor visual interactivo que guía a través del proceso de creación de un documento de JSON política con formato. Una vez creada la política, puede adjuntarla a la IAM identidad correspondiente (usuario, grupo o rol).

Para obtener más información sobre cómo adjuntar políticas administradas, consulte Añadir permisos de IAM identidad (consola) en la Guía del IAM usuario.

Inicio de sesión y supervisión AWS CloudShell

En este tema se describe cómo puede registrar y supervisar AWS CloudShell la actividad y el rendimiento con CloudTrail.

Supervisar la actividad con CloudTrail

AWS CloudShell está integrado con AWS CloudTrail un servicio que proporciona un registro de las acciones realizadas por un usuario, un rol o una Servicio de AWS persona AWS CloudShell. CloudTrail captura todas API las llamadas AWS CloudShell como eventos. Las llamadas capturadas incluyen las llamadas desde la AWS CloudShell consola y las llamadas en código a la AWS CloudShell API.

Si crea un registro, puede habilitar la entrega continua de CloudTrail eventos a un bucket de Amazon Simple Storage Service (Amazon S3). Esto incluye eventos para AWS CloudShell.

Si no configuras una ruta, podrás ver los eventos más recientes en la CloudTrail consola, en el historial de eventos. Con la información recopilada por CloudTrail, puedes descubrir distintos tipos de información sobre una solicitud. Por ejemplo, puede determinar el destinatario de la solicitud AWS CloudShell, conocer la dirección IP desde la que se realizó la solicitud, quién la hizo y cuándo se hizo.

AWS CloudShell en CloudTrail

La siguiente tabla muestra los AWS CloudShell eventos que se guardan en el archivo de CloudTrail registro.

> **Note**
>
> AWS CloudShell evento que incluye:
>
> - *indica que se trata de una llamada no mutante (de solo lectura). API
> - La palabra `Environment` se refiere al ciclo de vida del entorno informático que aloja la experiencia de shell.
> - La palabra `Layout` restaura todas las pestañas del navegador en el CloudShell terminal.

CloudShell Eventos en CloudTrail

Nombre de evento	Descripción
`createEnvironment`	Se produce cuando se crea un CloudShell entorno.
`createSession`	Se produce cuando un CloudShell entorno se conecta desde AWS Management Console.
`deleteEnvironment`	Se produce cuando se elimina un CloudShell entorno.
`deleteSession`	Se produce cuando se elimina la sesión de la CloudShell pestaña que se está ejecutando en la pestaña actual del navegador.
`getEnvironmentStatus*`	Se produce cuando se recupera el estado de un CloudShell entorno.
`getFileDownloadUrls*`	Se produce cuando se genera Amazon S3 prefirmado URLs que se utiliza para descargar archivos CloudShell mediante la interfaz CloudShell web.
`getFileUploadUrls*`	Se produce cuando se generan Amazon S3 prefirmados URLs que se utilizan para cargar archivos CloudShell mediante la interfaz CloudShell web.
`cloudshell:DescribeEnvironments`	Describe los entornos.
`getLayout*`	Se produce cuando se recupera el CloudShell diseño del inicio de la sesión.
`putCredentials`	Se produce cuando se CloudShell reenvían las credenciales utilizadas para iniciar sesión en el AWS Management Console .

Nombre de evento	Descripción
`redeemCode*`	Se produce cuando comienza el flujo de trabajo para recuperar el token de actualización en el CloudShell entorno. Más adelante, puede utilizar este token en el `putCredentials` comando para acceder al CloudShell entorno.
`sendHeartBeat`	Se produce para confirmar que la CloudShell sesión está activa.
`startEnvironment`	Se produce cuando se inicia un CloudShell entorno.
`stopEnvironment`	Se produce cuando se detiene un CloudShell entorno en ejecución.
`updateLayout`	Se produce cuando se guarda el diseño actual de la aplicación web en el backend.

Los eventos que incluyen la palabra «Diseño» restauran todas las pestañas del navegador en el CloudShell terminal.

EventBridge reglas de AWS CloudShell acción

Con EventBridge las reglas, se especifica la acción objetivo que se debe realizar cuando se EventBridge recibe un evento que coincide con la regla. Puedes definir una regla que especifique la acción objetivo que se debe realizar en función de una AWS CloudShell acción que se registre como un evento en un archivo de CloudTrail registro.

Por ejemplo, puede crear EventBridge reglas AWS CLI con el `put-rule` comando. Una `put-rule` llamada debe contener al menos un EventPattern o ScheduleExpression. Las reglas con EventPatterns se activan cuando se observa un evento coincidente. Los EventPattern cuatro AWS CloudShell eventos:

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

Para obtener más información, consulte Eventos y patrones de eventos EventBridge en la Guía del EventBridge usuario de Amazon.

Validación de conformidad para AWS CloudShell

Los auditores externos evalúan la seguridad y el cumplimiento de AWS los servicios como parte de varios programas de AWS cumplimiento.

AWS CloudShell está dentro del ámbito de aplicación de los siguientes programas de cumplimiento:

SOC

AWS Los informes de controles del sistema y la organización (SOC) son informes de examen independientes de terceros que demuestran cómo se AWS logran los principales controles y objetivos de cumplimiento.

Servicio	SDK	SOC1,2,3
AWS CloudShell	CloudShell	✓

PCI

El Estándar de Seguridad de Datos de la Industria de Tarjetas de Pago (PCIDSS) es un estándar de seguridad de la información patentado administrado por el Consejo de Normas de PCI Seguridad, que fue fundado por American Express, Discover Financial Services, JCB International, MasterCard Worldwide y Visa Inc.

Servicio	SDK	PCI
AWS CloudShell	CloudShell	✓

ISOy CSA STAR certificaciones y servicios

AWS cuenta con la certificación de conformidad con las normasISO/IEC27001:2013, 27017:2015, 27018:2019, 27701:2019, 22301:2019, 9001:2015y v4.0. CSA STAR CCM

Servicio	SDK	ISOy CSA STAR certificaciones y servicios
AWS CloudShell	CloudShell	✓

FedRamp

El Programa Federal de Gestión de Riesgos y Autorizaciones (FedRAMP) es un programa que abarca todo el gobierno de EE. UU. y que ofrece un enfoque estándar para la evaluación de la seguridad, la autorización y el monitoreo continuo de los productos y servicios en la nube.

Servicio	SDK	Reserva Federal RAMP Moderada (Este/Oeste)	RAMPMáximo de la Reserva Federal () GovCloud
AWS CloudShell	CloudShell	✓	✓

DoD CC SRG

La Guía de requisitos de seguridad de la computación en la nube () del Departamento de Defensa (DoDSRG) proporciona un proceso estandarizado de evaluación y autorización para que los proveedores de servicios en la nube (CSPs) obtengan una autorización provisional del DoD, de modo que puedan atender a los clientes del DoD.

Los servicios que se sometan a la SRG evaluación y autorización del DoD CC tendrán el siguiente estado:

- Evaluación de una organización de evaluación externa (3PAO): Nuestro evaluador externo está evaluando actualmente este servicio.
- Revisión de la Junta de Autorización Conjunta (JAB): Este servicio se encuentra actualmente en proceso de JAB revisión.
- Revisión de la Agencia de Sistemas de Información de Defensa (DISA): Este servicio se encuentra actualmente en proceso de DISA revisión.

Servicio	SDK	DoD CC SRG IL2 (Este/Oeste)	DoD CC () SRG IL2 GovCloud	DoD CC () SRG IL4 GovCloud	DoD CC () SRG IL5 GovCloud	DoD CC SRG IL6 (región AWS secreta)
AWS CloudShell	CloudShell	3PAO. Evaluación	N/A	N/A	N/A	N/A

HIPAA BAA

La Ley de Portabilidad y Responsabilidad de los Seguros Médicos de 1996 (HIPAA) es una ley federal que exige la creación de normas nacionales para proteger la información de salud confidencial de los pacientes para que no se divulgue sin el consentimiento o el conocimiento del paciente.

AWS permite a las entidades cubiertas y a sus socios comerciales sujetos HIPAA a ellas procesar, almacenar y transmitir de forma segura la información de salud protegida (PHI). Además, a partir de julio de 2013, AWS ofrece un apéndice estandarizado para socios comerciales (BAA) para dichos clientes.

Servicio	SDK	HIPAA BAA
AWS CloudShell	CloudShell	✓

IRAP

El Programa de evaluadores registrados en materia de seguridad de la información (IRAP) permite a los clientes del gobierno australiano validar que se han establecido los controles adecuados y determinar el modelo de responsabilidad adecuado para cumplir con los requisitos del Manual de seguridad de la información del Gobierno australiano (ISM) elaborado por el Centro de Ciberseguridad de Australia (). ACSC

Servicio	Espacio de nombres*	IRAPprotegido
AWS CloudShell	N/A	✓

*Los espacios de nombres le ayudan a identificar los servicios en todo su entorno. AWS Por ejemplo, al crear IAM políticas, trabajar con Amazon Resource Names (ARNs) y leer AWS CloudTrail registros.

MTCS

La seguridad en la nube de varios niveles (MTCS) es un estándar operativo de gestión de la seguridad de Singapur (SPRINGSS 584), basado en los estándares del Sistema de Gestión de la Seguridad de la Información ISO 27001/02 (). ISMS

Servicio	SDK	Este de EE. UU. (Ohio)	Este de EE. UU. (Norte de Virginia)	Oeste de EE. UU. (Oregón)	Oeste de EE. UU. (Norte de California)	Singapur	Seúl
AWS CloudShell	CloudShell	✓	✓	✓	N/A	N/A	N/A

C5

El catálogo de controles de cumplimiento de la informática en la nube (C5) es un esquema de certificación respaldado por el gobierno alemán introducido en Alemania por la Oficina Federal de Seguridad de la Información (BSI) para ayudar a las organizaciones a demostrar su seguridad operativa contra los ciberataques comunes cuando utilizan servicios en la nube en el contexto de las «Recomendaciones de seguridad para los proveedores de nube» del Gobierno alemán.

Servicio	SDK	C5
AWS CloudShell	CloudShell	✓

ENS¡Alto!

El esquema de acreditación ENS (Esquema Nacional de Seguridad) ha sido desarrollado por el Ministerio de Hacienda y Administración Pública y el CCN (Centro Criptológico Nacional). Se

compone de los principios básicos y los requisitos mínimos necesarios para la protección adecuada de la información.

Servicio	SDK	ENSAlto
AWS CloudShell	CloudShell	✓

FINMA

La Autoridad de Supervisión de los Mercados Financieros de Suiza (FINMA) es el regulador independiente de los mercados financieros de Suiza. AWS El cumplimiento de FINMA los requisitos demuestra nuestro compromiso continuo de cumplir con las altas expectativas que los reguladores de los servicios financieros y los clientes suizos han fijado para los proveedores de servicios en la nube.

Servicio	SDK	FINMA
AWS CloudShell	CloudShell	✓

PiTuKri

AWS El cumplimiento de PiTuKri los requisitos demuestra nuestro compromiso continuo de cumplir con las altas expectativas de los proveedores de servicios en la nube establecidas por la Agencia Finlandesa de Transporte y Comunicaciones, Traficom.

Servicio	SDK	PiTuKri
AWS CloudShell	CloudShell	✓

Para ver una lista de AWS los servicios que están incluidos en el ámbito de aplicación de programas de cumplimiento específicos, consulte AWSServicios incluidos en el ámbito de aplicación por programa AWS . Para obtener información general, consulte Programas de AWS cumplimiento > Programas AWS .

Puede descargar informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte Descarga de informes en AWS Artifact .

Su responsabilidad de cumplimiento al AWS CloudShell utilizarlos viene determinada por la confidencialidad de sus datos, los objetivos de cumplimiento de su empresa y las leyes y reglamentos aplicables. AWS proporciona los siguientes recursos para ayudar con el cumplimiento:

- Guías de inicio rápido sobre sobre seguridad y cumplimiento: estas guías de implementación analizan las consideraciones arquitectónicas y proporcionan los pasos para implementar entornos básicos centrados en la seguridad y el cumplimiento. AWS
- Documento técnico sobre la arquitectura basada en HIPAA la seguridad y el cumplimiento: en este documento técnico se describe cómo pueden utilizar las empresas para crear aplicaciones que cumplan con las normas. AWS HIPAA
- AWS Recursos de cumplimiento Recursos AWS : esta colección de libros de trabajo y guías puede aplicarse a su sector y ubicación.
- Evaluación de los recursos con las reglas de la guía para AWS Config desarrolladores: el AWS Config servicio evalúa en qué medida las configuraciones de los recursos cumplen con las prácticas internas, las directrices del sector y las normas.
- AWS Security Hub— Este AWS servicio proporciona una visión integral del estado de su seguridad AWS que le ayuda a comprobar su conformidad con los estándares y las mejores prácticas del sector de la seguridad.

Resiliencia en AWS CloudShell

La infraestructura AWS global se basa en AWS regiones y zonas de disponibilidad. AWS Las regiones proporcionan varias zonas de disponibilidad aisladas y separadas físicamente, que están conectadas mediante redes de baja latencia, alto rendimiento y alta redundancia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una conmutación por error automática entre las zonas sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de uno o varios centros de datos.

Para obtener más información sobre AWS las regiones y las zonas de disponibilidad, consulte Infraestructura global.AWS

Además de la infraestructura AWS global, AWS CloudShell admite las siguientes funciones para satisfacer sus necesidades de respaldo y resiliencia de datos:

- Confirme los archivos que cree y a los que AWS CodeCommit añada. Esta es la documentación oficial de Amazon Web Services que puede utilizar para almacenar y administrar activos en la

nube. Estos activos pueden consistir en documentos, código fuente y archivos binarios. Para obtener más información, consulte Uso CodeCommit en AWS CloudShell.

- Utilice AWS CLI las llamadas para especificar los archivos de su directorio principal AWS CloudShell y añadirlos como objetos en los buckets de Amazon S3. Para ver un ejemplo, consulte la sección Cómo empezar con AWS CloudShell.

Seguridad de la infraestructura en AWS CloudShell

Como servicio gestionado, AWS CloudShell está protegido por la seguridad de la red AWS global. Para obtener información sobre los servicios AWS de seguridad y cómo se AWS protege la infraestructura, consulte Seguridad AWS en la nube. Para diseñar su AWS entorno utilizando las mejores prácticas de seguridad de la infraestructura, consulte Protección de infraestructuras en un marco de buena AWS arquitectura basado en el pilar de la seguridad.

Utiliza las API llamadas AWS publicadas para acceder a AWS CloudShell través de la red. Los clientes deben admitir lo siguiente:

- Seguridad de la capa de transporte (TLS). Necesitamos TLS 1.2 y recomendamos TLS 1.3.
- Cifre suites con perfecto secreto (PFS), como (Ephemeral Diffie-Hellman) o DHE ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

Además, las solicitudes deben firmarse con un identificador de clave de acceso y una clave de acceso secreta que esté asociada a un director. IAM También puede utilizar AWS Security Token Service (AWS STS) para generar credenciales de seguridad temporales para firmar solicitudes.

> **Note**
>
> De forma predeterminada, instale AWS CloudShell automáticamente los parches de seguridad para los paquetes del sistema de sus entornos informáticos.

Prácticas recomendadas de seguridad para AWS CloudShell

Las siguientes prácticas recomendadas son directrices generales y no constituyen una solución de seguridad completa. Dado que estas prácticas recomendadas pueden no ser adecuadas o

suficientes para su entorno, le recomendamos que las trate como consideraciones útiles y no como prescripciones.

Algunas prácticas recomendadas de seguridad para AWS CloudShell

- Utilice IAM los permisos y las políticas para controlar el acceso AWS CloudShell y garantizar que los usuarios solo puedan realizar las acciones requeridas por su función (por ejemplo, descargar y cargar archivos). Para obtener más información, consulte Administrar el AWS CloudShell acceso y el uso con IAM políticas.
- No incluya datos confidenciales en sus IAM entidades, como usuarios, funciones o nombres de sesión.
- Mantenga habilitada la Función de pegado seguro para detectar posibles riesgos de seguridad en el texto que ha copiado de fuentes externas. La opción de pegado seguro está habilitada de manera predeterminada. Para obtener más información sobre el uso de la función de pegado seguro para texto de líneas múltiples, consulte Uso de la función de pegado seguro para texto de líneas múltiples.
- Familiarícese con el modelo de responsabilidad de seguridad compartida si instala aplicaciones de terceros en el entorno informático de. AWS CloudShell
- Prepare los mecanismos de reversión antes de editar los scripts del intérprete de comandos que afecten a la experiencia del usuario con el intérprete de comandos. Para obtener más información sobre cómo modificar el entorno de shell predeterminado, consulte Modificar el shell con scripts.
- Almacene su código de forma segura en un sistema de control de versiones, por ejemplo, AWS CodeCommit.

AWS CloudShell Seguridad FAQs

Las siguientes son respuestas a las preguntas más frecuentes sobre la seguridad de CloudShell.

- ¿Qué AWS procesos y tecnologías se utilizan al lanzar CloudShell e iniciar una sesión provisional?
- ¿Es posible restringir el acceso a la red a CloudShell?
- ¿Puedo personalizar mi CloudShell entorno?
- ¿Dónde está realmente almacenado mi directorio `$HOME` en Nube de AWS?
- ¿Es posible cifrar mi directorio `$HOME`?
- ¿Puedo ejecutar un análisis de virus en mi directorio `$HOME`?

¿Qué AWS procesos y tecnologías se utilizan al lanzar CloudShell e iniciar una sesión provisional?

Al iniciar sesión AWS Management Console, debe introducir sus credenciales IAM de usuario. Además, cuando se inicia CloudShell desde la interfaz de la consola, estas credenciales se utilizan en las llamadas CloudShell API que crean un entorno informático para el servicio. A continuación, se crea una AWS Systems Manager sesión para el entorno informático y se CloudShell envían los comandos a esa sesión.

Volver a la lista de seguridad FAQs

¿Es posible restringir el acceso a la red a CloudShell?

En los entornos públicos, no es posible restringir el acceso a la red. Si desea restringir el acceso a la red, debe habilitar el permiso para crear únicamente VPC entornos y denegar la creación de entornos públicos.

Para obtener más información, consulte Garantizar que los usuarios creen únicamente VPC entornos y denegar la creación de entornos públicos.

En el CloudShell VPC caso de los entornos, la configuración de red se hereda de suVPC. El uso de CloudShell in a VPC le permite controlar el acceso a la red de su CloudShell VPC entorno.

Volver a la lista de seguridad FAQs

¿Puedo personalizar mi CloudShell entorno?

Puede descargar e instalar utilidades y otro software de terceros para su CloudShell entorno. Solo el software que está instalado en su directorio `$HOME` se conserva entre sesiones.

Según lo definido en el modelo de responsabilidad compartida de AWS, usted es responsable de la configuración y administración necesarias de las aplicaciones que instale.

Volver a la lista de seguridad FAQs

¿Dónde está realmente almacenado mi directorio **`$HOME`** en Nube de AWS?

En el caso de los entornos públicos, Amazon S3 proporciona la infraestructura para almacenar los datos en sus `$HOME` entornos.

En el VPC caso de los entornos, el $HOME directorio se elimina cuando se agota el tiempo de espera (después de 20 a 30 minutos de inactividad) o cuando se elimina o reinicia el entorno. VPC

Volver a la lista de seguridad FAQs

¿Es posible cifrar mi directorio **$HOME**?

No, no es posible cifrar el $HOME directorio con una clave propia. Sin embargo, CloudShell cifra el contenido del $HOME directorio mientras lo almacena en Amazon S3.

Volver a la lista de seguridad FAQs

¿Puedo ejecutar un análisis de virus en mi directorio **$HOME**?

Por el momento, no es posible realizar un análisis de virus en su directorio $HOME. Se está revisando la compatibilidad con esta característica.

Volver a la lista de seguridad FAQs

¿Puedo restringir la entrada o salida de datos para mí? CloudShell

Para restringir la entrada o la salida, le recomendamos que utilice un entorno. CloudShell VPC
El $HOME directorio de un VPC entorno se elimina cuando se agota el tiempo de espera del VPC entorno (después de 20 a 30 minutos de inactividad) o cuando se elimina o reinicia el entorno. En el menú Acciones, las opciones de carga y descarga no están disponibles para los VPC entornos.

Volver a la lista de seguridad FAQs

AWS CloudShell entorno informático: especificaciones y software

Cuando se lanza AWS CloudShell, se crea un entorno informático basado en Amazon Linux 2023 para alojar la experiencia de shell. El entorno está configurado con recursos de cómputo (v CPU y memoria) y proporciona una amplia gama de software preinstalado al que se puede acceder desde la interfaz de línea de comandos. Asegúrese de que todo el software que instale en el entorno informático esté parcheado y actualizado. También puede configurar su entorno predeterminado instalando software y modificando los scripts del intérprete de comandos.

Recursos del entorno de computación

A cada entorno AWS CloudShell informático se le asignan los siguientes recursos de memoria CPU y los siguientes:

- 1 v CPU (unidad central de procesamiento virtual)
- 2 GiB RAM

Además, el entorno se aprovisiona con la siguiente configuración de almacenamiento:

- 1 GB de almacenamiento persistente (el almacenamiento persiste después de finalizar la sesión)

Para obtener más información, consulte Almacenamiento persistente.

CloudShell requisitos de red

WebSockets

CloudShell depende del WebSocket protocolo, que permite la comunicación interactiva bidireccional entre el navegador web del usuario y el CloudShell servicio en la AWS nube. Si utilizas un navegador en una red privada, es probable que los servidores proxy y los firewalls faciliten el acceso seguro a Internet. WebSocket Por lo general, la comunicación puede atravesar los servidores proxy sin problemas. Sin embargo, en algunos casos, los servidores proxy WebSockets impiden que funcionen correctamente. Si ocurre este problema, su CloudShell interfaz informa del siguiente error:`Failed to open sessions : Timed out while opening the session`.

Si este error se produce repetidamente, consulte la documentación de su servidor proxy para asegurarse de que esté configurado para permitir WebSockets. Como alternativa, puede ponerse en contacto con el administrador del sistema de su red.

> **Note**
>
> Si quieres definir los permisos detallados mediante listas de permisos específicasURLs, puedes añadir parte de los URL que utiliza la AWS Systems Manager sesión para abrir una WebSocket conexión para enviar entradas y recibir salidas. (Los AWS CloudShell comandos se envían a esa sesión de Systems Manager).
> El formato que StreamUrl utiliza Systems Manager es`wss://ssmmessages.`**`region`**`.amazonaws.com/v1/data-channel/`**`session-id`**`?stream=(input|output)`.
> La región representa el identificador de AWS región de una región compatible AWS Systems Manager, como `us-east-2` la región EE.UU. Este (Ohio).
> Como el identificador de sesión se crea después de que una sesión concreta de Systems Manager se haya iniciado correctamente, solo puede especificarlo `wss://ssmmessages.region.amazonaws.com` al actualizar URL la lista de permitidos. Para obtener más información, consulte la StartSessionoperación en la Referencia.AWS Systems Manager API

Software preinstalado

> **Note**
>
> Como el entorno de AWS CloudShell desarrollo se actualiza periódicamente para proporcionar acceso al software más reciente, no proporcionamos números de versión específicos en esta documentación. En su lugar, describimos cómo puede comprobar qué versión está instalada. Para comprobar la versión instalada, introduzca el nombre del programa seguido de la opción `--version` (por ejemplo, `git --version`).

Intérpretes de comandos

Intérpretes de comandos preinstalados

Nombre	Descripción	Version information
Bash	El shell Bash es la aplicación de shell predeterminada para AWS CloudShell.	`bash --version`
PowerShell (pwsh)	Al ofrecer una interfaz de línea de comandos y compatibilidad con el lenguaje de secuencia s de comandos, PowerShel l se basa en los de Microsoft . NETTiempo de ejecución del lenguaje de comandos. PowerShell utiliza comandos ligeros llamados `cmdlets` aceptar y devolver. NETobjeto s.	`pwsh --version`
Z Shell (zsh)	Z Shell, también conocido como `zsh`, es una versión ampliada de Bourne Shell que ofrece un soporte de personali zación mejorado para temas y complementos.	`zsh --version`

AWS interfaces de línea de comandos (CLI)

CLI

Nombre	Descripción	Version information
AWS CDK Kit de herramientas CLI	El AWS CDK kit de herramien tas, el CLI comando`cdk`, es la herramienta principal que	`cdk --version`

Nombre	Descripción	Version information
	interactúa con la aplicación. AWS CDK Ejecuta tu aplicación, consulta el modelo de aplicación que has definido y produce e implementa las plantillas generadas por el AWS CloudFormation . AWS CDK Para obtener más información, consulte Kit de herramientas AWS CDK.	
AWS CLI	AWS CLI Se trata de una interfaz de línea de comandos que puede utilizar para gestionar varios AWS servicios desde la línea de comandos y automatizarlos mediante scripts. Para obtener más información, consulte Trabajar con AWS servicios en AWS CloudShell. Para obtener información acerca de cómo puede asegurarse de utilizar la mayoría de la up-to-date AWS CLI versión 2, consulteInstalación en AWS CLI su directorio principal.	`aws --version`

Nombre	Descripción	Version information
EB CLI	AWS Elastic Beanstalk CLIProporciona una interfaz de línea de comandos para simplificar la creación, actualización y supervisión de los entornos desde un repositorio local. Para obtener más información, consulte Uso de la interfaz de línea de comandos CLI (EB) de Elastic Beanstalk en la Guía para desarrolladores.AWS Elastic Beanstalk	`eb --version`
Amazon ECS CLI	La interfaz de línea de comandos de Amazon Elastic Container Service (Amazon ECSCLI) () proporciona comandos de alto nivel para simplificar la creación, actualización y supervisión de clústeres y tareas. Para obtener más informaci ón, consulte Uso de la interfaz de línea de ECS comandos de Amazon en la Guía para desarrolladores de Amazon Elastic Container Service.	`ecs-cli --version`

Nombre	Descripción	Version information
AWS SAM CLI	AWS SAM CLIes una herramienta de línea de comandos que funciona con una AWS Serverless Application Model plantilla y un código de aplicación. Puede realizar varias tareas. Estas incluyen la invocación local de las funciones de Lambda, la creación de un paquete de despliegue para la aplicación sin servidor y el despliegue de la aplicación sin servidor en la nube. AWS Para obtener más información, consulte la referencia de AWS SAM CLI comandos en la Guía para desarrolladores.AWS Serverless Application Model	`sam --version`

Nombre	Descripción	Version information
AWS Tools for PowerShell	AWS Tools for PowerShell Se trata de PowerShell módulos que se basan en la funcional idad expuesta en el AWS SDK for .NET. Con AWS Tools for PowerShell, puede programar operaciones en sus AWS recursos desde la línea de PowerShell comandos. AWS CloudShell preinstal a la versión modularizada (AWS.Tools) del. AWS Tools for PowerShell Para obtener más informaci ón, consulte Uso de las AWS herramientas de la Guía del usuario PowerShell.AWS Tools for PowerShell	`pwsh --Command ' Get-Module -ListAvai lable -Name AWS.Tools .Common'`

Tiempos de ejecución y AWSSDKs: Node.js y Python 3

Tiempos de ejecución y AWS SDKs

Nombre	Descripción	Version information
Node.js (con npm)	Node.js es un JavaScript motor de ejecución diseñado para facilitar la aplicación de técnicas de programac ión asíncrona. Para obtener más información, consulte la documentación del sitio oficial de Node.js.	• Node.js: `node --version` • npm: `npm --version`

Nombre	Descripción	Version information
	npm es un administrador de paquetes que proporcio na acceso a un registro de módulos en línea. JavaScript Para obtener más información, consulte la documentación en el sitio web de npm.	
SDKpara JavaScript en Node.js	El kit de desarrollo de software (SDK) ayuda a simplificar la codificación al proporcionar JavaScript objetos para AWS servicios como Amazon S3, AmazonEC2, DynamoDB y Amazon. SWF Para obtener más información, consulte la Guía para desarrolladores de AWS SDK for JavaScript.	`npm -g ls --depth 0 2>/dev/null | grep aws-sdk`

Nombre	Descripción	Version information
Python	Python 3 está listo para usarse en el entorno de shell. Python 3 ahora se considera la versión predeterminada del lenguaje de programación (Python 2 dejó de recibir soporte en enero de 2020). Para obtener más información, consulte la documentación del sitio oficial de Python. Además, viene preinstalado pip, el instalador de paquetes para Python. Puede usar este programa de la línea de comandos para instalar paquetes de Python desde los índices en línea, como el Python Package Index. Para obtener más información, consulte la documentación de Python Packaging Authority.	• Python 3: `python3 --version` • pip: `pip3 --version`

Nombre	Descripción	Version information
SDKpara Python (Boto3)	Boto es el kit de desarrollo de software (SDK) que los desarrolladores de Python utilizan para crear, configura r y administrar Servicios de AWS, como Amazon EC2 y Amazon S3. SDKPropor ciona un acceso easy-to-use orientado a objetos API y de bajo nivel. Servicios de AWS Para obtener más información, consulte la documentación de Boto3.	`pip3 list | grep boto3`

Herramientas de desarrollo y utilidades de intérprete de comandos

Herramientas de desarrollo y utilidades de intérprete de comandos

Nombre	Descripción	Version information
bash-completion	bash-completion es un conjunto de funciones de intérprete de comandos que permite completar automáticamente comandos o argumentos escritos parcialme nte pulsando la tecla Tab. Puede encontrar los paquetes compatibles con bash-comp letion en `/usr/share/ bash-completion/ completions` .	`dnf info bash-comp letion`

Nombre	Descripción	Version information
	Para configurar la función de autocompletar los comandos de un paquete, el archivo del programa debe ser el origen. Por ejemplo, para configurar la función de autocompletar para los comandos de Git, añade la siguiente línea para `.bashrc` que la función esté disponible siempre que se inicie la AWS CloudShell sesión: `source /usr/share/bash-completion/completions/git` Si quiere usar scripts de finalización personalizados, agréguelos a su directorio principal persistente (`$HOME`) y búsquelos directamente en `.bashrc`. Para obtener más información, consulta la README página del proyecto en GitHub.	

Nombre	Descripción	Version information
CodeCommit utilidad para Git	git-remote-codecommit es una utilidad que proporcio na un método sencillo para insertar y extraer código de los CodeCommit repositor ios mediante la extensión de Git. Es el método recomenda do para admitir conexione s realizadas con acceso federado, proveedores de identidad y credenciales temporales. Para obtener más informaci ón, consulta los pasos de configuración de HTTPS las conexiones AWS CodeCommi t a y git-remote-codecom mit en la Guía del AWS CodeCommit usuario.	`pip3 list | grep git-remote-codecommit`
Git	Git es un sistema de control de versiones distribuido que apoya las prácticas modernas de desarrollo de software a través de los flujos de trabajo de las sucursales y la puesta en escena del contenido. Para obtener más informaci ón, consulte la página de documentación del sitio oficial de Git.	`git --version`

Nombre	Descripción	Version information
iputils	El paquete iputils contiene utilidades para redes Linux. Para obtener más información sobre las utilidades incluidas , consulte el repositorio iputils en. GitHub	Ejemplos de una herramienta iputils: `arping -V`
jq	La utilidad jq analiza los datos con JSON formato para producir resultados que se modifican mediante filtros de línea de comandos. Para obtener más información, consulte el manual de jq alojado en. GitHub	`jq --version`
kubectl	kubectl es una herramien ta de línea de comandos para comunicarse con el plano de control de un clúster de Kubernetes mediante Kubernetes. API	`kubectl --version`
make	La utilidad make utiliza `makefiles` para automatiz ar conjuntos de tareas y organizar la compilación de código. Para obtener más información, consulta la documentación de Make. GNU	`make --version`

Nombre	Descripción	Version information
man	El comando man proporcio na páginas de manual para utilidades y herramientas de la línea de comandos. Por ejemplo, `man ls` vuelve a la página del manual del comando `ls` que muestra el contenido de los directorios. Para obtener más informaci ón, consulte la entrada man en Wikipedia.	`man --version`
nano	nano es un editor pequeño y fácil de usar para una interfaz basada en texto. Para obtener más información, consulte la documentación sobre GNU nano.	`nano --version`
procps	procps es una utilidad de administración del sistema que puede utilizar para supervisar y detener los procesos que se estén ejecutando actualmente. Para obtener más información, consulte el README archivo que muestra los programas que se pueden ejecutar con procps.	`ps --version`

Nombre	Descripción	Version information
SSHcliente	SSHlos clientes utilizan el protocolo shell seguro para las comunicaciones cifradas con un ordenador remoto. Open SSH es el SSH cliente que viene preinstalado. Para obtener más información, consulte el SSHsitio de Open mantenido por OpenBSD.	`ssh -V`
sudo	Con la utilidad sudo, los usuarios pueden ejecutar un programa con los permisos de seguridad de otro usuario, normalmente el superusua rio. Sudo resulta útil cuando necesita instalar aplicacio nes como administrador del sistema. Para obtener más información, consulte el manual de Sudo.	`sudo --version`
tar	tar es una utilidad de la línea de comandos que se puede utilizar para agrupar varios archivos en un único archivo (a menudo denominado tarball). Para obtener más información, consulte la documentación de GNU tar.	`tar --version`

Nombre	Descripción	Version information
tmux	tmux es un multiplexor de terminal que puede utilizar para ejecutar diferentes programas simultáneamente en varias ventanas. Para obtener más información, consulte un blog que ofrece una introducción concisa a tmux.	`tmux -V`
unzip	Para obtener más información, consulte zip/unzip.	
vim	vim es un editor personalizable con el que puedes interactuar a través de una interfaz basada en texto. Para obtener más información, consulte la documentación del proveedor de recursos de vim.org.	`vim --version`
wget	wget es un programa informático utilizado para recuperar contenido de servidores web especificados por puntos de conexión en la línea de comandos. Para obtener más información, consulte la documentación de GNU Wget.	`wget --version`

Nombre	Descripción	Version information

zip/unzip	Las utilidades zip/unzip utilizan un formato de archivo comprimido que ofrece una compresión de datos sin pérdida de datos. Ejecute el comando zip para agrupar y comprimir archivos en un único archivo. Use unzip para extraer archivos de un archivo a un directorio específico.	`unzip --version`
		
		`zip --version`

Nombre	Descripción	Version information
Docker	Docker es una plataform a abierta para desarrollar, enviar y ejecutar aplicacio nes. Docker le permite separar sus aplicaciones de su infraestructura para que pueda entregar software rápidamente. Le permite crear Dockerfiles internamente AWS CloudShell y crear activos de Docker con ellos. CDK Para obtener información sobre qué AWS regiones son compatibl es con Docker, consulte Regiones compatibles para. AWSAWS CloudShell Debe tener en cuenta que el espacio de Docker en el entorno es limitado. Si tiene imágenes individuales de gran tamaño o demasiadas imágenes de Docker preexistentes, pueden producirse problemas. Para obtener más información sobre Docker, consulta la guía de documentación de Docker.	`docker --version`

Instalación en AWS CLI su directorio principal

Al igual que el resto del software preinstalado en su CloudShell entorno, la AWS CLI herramienta se actualiza automáticamente con actualizaciones programadas y parches de seguridad. Si quiere asegurarse de disponer de la up-to-date versión más completa de AWS CLI, puede optar por instalar la herramienta manualmente en el directorio principal del shell.

> **Important**
>
> Debe instalar manualmente la copia de AWS CLI en el directorio principal para que esté disponible la próxima vez que inicie una CloudShell sesión. Esta instalación es necesaria porque los archivos que se añaden a directorios externos a `$HOME` se eliminan al finalizar una sesión del intérprete de comandos. Además, después de instalar esta copia de AWS CLI, no se actualiza automáticamente. Es decir, es su responsabilidad administrar las actualizaciones y los parches de seguridad.
> Para obtener más información sobre el modelo de responsabilidad AWS compartida, consulte[Protección de datos en AWS CloudShell](#).

Para instalar AWS CLI

1. En la línea de CloudShell comandos, utilice el `curl` comando para transferir una copia comprimida del archivo AWS CLI instalado al shell:

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

2. Descomprima la carpeta comprimida:

   ```
   unzip awscliv2.zip
   ```

3. Para añadir la herramienta a una carpeta específica, ejecute el AWS CLI instalador:

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

   Si se ha instalado correctamente, la línea de comandos muestra el siguiente mensaje:

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

4. Para su comodidad, le recomendamos que actualice también la variable de entorno de `PATH` para no tener que especificar la ruta de instalación de la herramienta al ejecutar los comandos `aws`:

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

> **Note**
>
> Si deshace este cambio a`PATH`, `aws` los comandos que no incluyan una ruta específica utilizarán la versión preinstalada de AWS CLI por defecto.

Instalación de software de terceros en el entorno del intérprete de comandos

> **Note**
>
> Le recomendamos que revise el modelo de responsabilidad de seguridad compartida antes de instalar cualquier aplicación de terceros en el entorno informático AWS CloudShell del dispositivo.

De forma predeterminada, todos los AWS CloudShell usuarios tienen permisos de sudo. Por lo tanto, puede usar el comando `sudo` para instalar software que aún no esté disponible en el entorno de computación del intérprete de comandos. Por ejemplo, puede utilizarlos `sudo` con la utilidad de DNF administración de paquetes para instalarlo`cowsay`, que genera imágenes ASCII artísticas de una vaca con el siguiente mensaje:

```
sudo dnf install cowsay
```

A continuación, puede iniciar el programa recién instalado escribiendo `echo "Welcome to AWS CloudShell" | cowsay`.

> **Important**
>
> Las utilidades de administración de paquetes`/usr/bin`, como dnf, instalan programas en directorios (por ejemplo), que se reciclan cuando finaliza la sesión de shell. Esto significa que se instala y utiliza software adicional por sesión.

Modificar el intérprete de comandos con scripts

Si desea modificar el entorno del intérprete de comandos predeterminado, puede editar un script del intérprete de comandos que se ejecute cada vez que se inicie el entorno del intérprete de comandos. El script `.bashrc` se ejecuta cada vez que se inicia el intérprete de comandos bash predeterminado.

> **Warning**
>
> Si modifica el archivo `.bashrc` de forma incorrecta, es posible que no pueda acceder al entorno del intérprete de comandos más adelante. Se recomienda hacer una copia del archivo antes de editarlo. También puede evitar riesgos si abre dos intérpretes de comandos al editar `.bashrc`. Si pierde el acceso a un intérprete de comandos, su sesión seguirá iniciada en otro intérprete de comandos y podrá deshacer cualquier cambio.
> Si pierde el acceso después de modificar incorrectamente `.bashrc` o cualquier otro archivo, puede volver AWS CloudShell a su configuración predeterminada borrando su directorio principal.

En el proceso, modificará el script `.bashrc` para que su entorno del intérprete de comandos pase automáticamente a ejecutar el intérprete de comandos Z.

1. Abra `.bashrc` con un editor de texto (Vim, por ejemplo):

   ```
   vim .bashrc
   ```

2. En la interfaz del editor, pulse la tecla I para iniciar la edición y, a continuación, añada lo siguiente:

   ```
   zsh
   ```

3. Para salir del archivo editado `.bashrc` y guardarlo, pulse Esc para entrar en el modo de comando Vim e introduzca lo siguiente:

   ```
   :wq
   ```

4. Utilice el comando `source` para volver a cargar el archivo `.bashrc`:

   ```
   source .bashrc
   ```

Cuando la interfaz de la línea de comandos vuelva a estar disponible, el símbolo del indicador cambiará a % para indicar que ahora está utilizando el intérprete de comandos Z.

AWS CloudShell migrar de 0 AL2 a 023 AL2

AWS CloudShell, que estaba basado en Amazon Linux 2 (AL2), migró a Amazon Linux 2023 (AL2023). Para obtener más información acerca de AL2 023, consulte Qué es Amazon Linux 2023 (AL2023) en la Guía del usuario de Amazon Linux 2023.

Con AL2 023, puede seguir accediendo a su CloudShell entorno actual con todas las herramientas que ofrece. CloudShell Para obtener más información sobre las herramientas disponibles, consulte Software preinstalado.

AL2023 proporciona varias mejoras a las herramientas de desarrollo, incluidas las versiones más recientes de paquetes, como Node.js 18 y Python 3.9.

> **Note**
>
> En AL2 203, Python 2 ya no se envía con su CloudShell entorno.

Para obtener más información sobre las principales diferencias entre Amazon Linux AL2 y AL2 023, consulte Comparación de Amazon Linux 2 y Amazon Linux 2023 en la Guía del usuario de Amazon Linux 2023.

Si tiene alguna pregunta, póngase en contacto con AWS Support. También puede buscar respuestas y publicar preguntas en AWS re:Post. Cuando entres AWS re:Post, es posible que tengas que iniciar sesión en. AWS

AWS CloudShell Migración FAQs

Las siguientes son respuestas a algunas preguntas frecuentes sobre la migración del AL2 0 AL2 al 023 con AWS CloudShell.

- ¿Afectará esta migración a alguno de mis otros AWS recursos, como las EC2 instancias de Amazon en las que se estén ejecutandoAL2?
- ¿Cuáles son los paquetes que se cambiarán con la migración a la versión AL2 023?

¿Afectará la migración a AL2 023 a alguno de mis otros AWS recursos, como las EC2 instancias de Amazon en AL2 las que se estén ejecutando?

Esta migración no afecta a ningún servicio o recurso que no sea su AWS CloudShell entorno. Esto incluye los recursos que podría haber creado o a los que haya accedido desde dentro AWS CloudShell. Por ejemplo, si has creado una EC2 instancia de Amazon que se ejecute en ellaAL2, no se migrará a AL2 023.

¿Cuáles son los paquetes que se han modificado con la migración a la AL2 023?

AWS CloudShell los entornos actualmente incluyen software preinstalado. Para obtener más información sobre la lista completa de software preinstalado, consulte Software preinstalado. AWS CloudShell seguirá entregando estos paquetes, con la excepción de Python 2. Para ver la diferencia completa entre los paquetes proporcionados por AL2 y AL2 023, consulte Comparación AL2 y AL2 023. Para los clientes con requisitos específicos de paquete y versión que dejarán de cumplirse tras la migración a la versión AL2 023, les recomendamos que se pongan en contacto con AWS Support para enviar una solicitud.

¿Puedo no participar en la migración?

No, no puedes excluirte de la migración. AWS CloudShell los entornos se administran mediante AWS, por lo tanto, todos los entornos se han actualizado a la versión AL2 023.

¿Puedo crear una copia de seguridad de mi AWS CloudShell entorno?

AWS CloudShell seguirá conservando el directorio principal del usuario. Para obtener más información, consulte Service Quotas y limitaciones para AWS CloudShell. Si tiene algún archivo o configuración almacenado en su carpeta principal y desea crear una copia de seguridad del mismo, complete el Paso 6: cree una copia de seguridad del directorio principal.

Solución de problemas AWS CloudShell

Durante el uso AWS CloudShell, es posible que se produzcan problemas, por ejemplo, al iniciar CloudShell o realizar tareas clave mediante la interfaz de línea de comandos del shell. La información que se trata en este capítulo describe cómo solucionar algunos de los problemas comunes que es posible que encuentre.

Para obtener respuestas a una variedad de preguntas sobre CloudShell, consulte la AWS CloudShell FAQs. También puede buscar respuestas y publicar preguntas en los foros de debate de AWS CloudShell. Cuando acceda al foro, es posible que se requiera que inicie sesión en AWS. También puede ponerse en contacto con nosotros directamente.

Solución de errores

Cuando se encuentre ante alguno de los siguientes errores de indexación, puede utilizar las siguientes soluciones que se indican a continuación para corregirlos.

Temas

- El usuario con CreateEnvironment permiso exclusivo para VPC entornos también tiene acceso a AWS CloudShell entornos públicos
- Las credenciales no funcionan en CloudShell

Error: «No se puede iniciar el entorno. Para volver a intentarlo, actualiza el navegador o reinicia seleccionando «Acciones, Reiniciar AWS CloudShell »

Problema: cuando intentas iniciar AWS CloudShell desde el AWS Management Console, se te deniega el acceso incluso después de haber obtenido los permisos necesarios de tu IAM administrador y de haber actualizado o reiniciado el navegador. CloudShell

Solución: póngase en contacto con AWS Support.

(Volver arriba)

Error: «No se puede iniciar el entorno. No dispone del permiso necesario. Pida a su IAM administrador que le conceda acceso a AWS CloudShell»

Problema: cuando intentas iniciar AWS CloudShell desde AWS Management Console, se te deniega el acceso y se te notifica que no tienes los permisos necesarios.

Causa: la IAM identidad a la que estás accediendo AWS CloudShell carece de los IAM permisos necesarios.

Solución: solicite al IAM administrador que le proporcione los permisos necesarios. Para ello, pueden añadir una política AWS gestionada adjunta (AWSCloudShellFullAccess) o una política integrada integrada. Para obtener más información, consulte Administrar el AWS CloudShell acceso y el uso con políticas IAM.

(Volver arriba)

No se puede acceder a la línea de AWS CloudShell comandos

Problema: después de modificar un archivo que utiliza el entorno informático, no se puede acceder a la línea de comandos AWS CloudShell.

Solución: si pierde el acceso después de modificar `.bashrc` o modificar cualquier otro archivo de forma incorrecta, puede volver AWS CloudShell a su configuración predeterminada borrando el directorio principal.

(Volver arriba)

No se puede hacer ping a las direcciones IP externas

Problema: cuando ejecuta un comando ping desde la línea de comandos (por ejemplo, `ping amazon.com`), recibe el siguiente mensaje.

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

Causa: la utilidad ping utiliza el Protocolo de mensajes de control de Internet (ICMP) para enviar paquetes de solicitudes de eco a un host de destino. Espere a que se produzca un eco desde el objetivo para responder. Como el ICMP protocolo no está habilitado en AWS CloudShell, la utilidad ping no funciona en el entorno informático del shell.

Solución: debido a que no ICMP es compatible AWS CloudShell, puede ejecutar el siguiente comando para instalar Netcat. Netcat es una utilidad de red informática para leer y escribir en conexiones de red mediante TCP o. UDP

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

(Volver arriba)

Se han producido algunos problemas al preparar el terminal

Problema: al intentar acceder AWS CloudShell mediante el navegador Microsoft Edge, no puede iniciar una sesión de shell y el navegador muestra un mensaje de error.

Causa: AWS CloudShell no es compatible con versiones anteriores de Microsoft Edge. Puedes acceder AWS CloudShell mediante las cuatro versiones principales más recientes de los navegadores compatibles.

Solución: instale una versión actualizada del navegador Edge desde el sitio de Microsoft.

(Volver arriba)

Las teclas de flecha no funcionan correctamente en PowerShell

Problema: En condiciones normales de funcionamiento, puede utilizar las teclas de dirección para navegar por la interfaz de la línea de comandos y explorar el historial de comandos hacia atrás y

hacia delante. Sin embargo, al presionar las teclas de flecha en ciertas versiones de PowerShell on AWS CloudShell, es posible que las letras se muestren incorrectamente.

Causa: la situación en la que las teclas de flecha imprimen letras de forma incorrecta es un problema conocido en las versiones PowerShell 7.2.x que se ejecutan en Linux.

Solución: para eliminar las secuencias de escape que modifican el comportamiento de las teclas de flecha, edite el archivo PowerShell de perfil y establezca la `$PSStyle` variable en. `PlainText`

1. En la línea de AWS CloudShell comandos, introduzca el siguiente comando para abrir el archivo de perfil.

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

   > **Note**
   >
   > Si ya está dentro PowerShell, también puede abrir el archivo de perfil en el editor con el siguiente comando.
   >
   > ```
   > vim $PROFILE
   > ```

2. En el editor, vaya al final del texto existente en el archivo, presione i para entrar en el modo Insertar y, a continuación, añada la siguiente declaración.

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

3. Tras realizar la edición, pulse Esc para entrar en el modo de comando. A continuación, introduzca el siguiente comando para guardar el archivo y salir del editor.

   ```
   :wq
   ```

> **Note**
>
> Los cambios surtirán efecto la próxima vez que comience PowerShell.

(Volver arriba)

Los Web Sockets no compatibles provocan un error al iniciar las sesiones CloudShell

Problema: Al intentar iniciar AWS CloudShell, recibe repetidamente el siguiente mensaje:`Failed to open sessions : Timed out while opening the session`.

Causa: CloudShell depende del WebSocket protocolo, que permite la comunicación interactiva bidireccional entre su navegador web y AWS CloudShell. Si utilizas un navegador en una red privada, es probable que los servidores proxy y los firewalls faciliten el acceso seguro a Internet. WebSocket Por lo general, la comunicación puede atravesar los servidores proxy sin problemas. Sin embargo, en algunos casos, los servidores proxy WebSockets impiden que funcionen correctamente. Si se produce este problema, no se CloudShell puede iniciar una sesión de shell y, finalmente, se agota el tiempo de espera del intento de conexión.

Solución: el tiempo de espera de la conexión puede deberse a un problema que no WebSockets sea compatible. Si este es el caso, actualice primero la ventana del navegador donde se encuentra la interfaz de línea de CloudShell comandos.

Si sigue recibiendo errores de tiempo de espera después de la actualización, consulte la documentación de su servidor proxy. Además, asegúrese de que su servidor proxy esté configurado para permitir Web Sockets. También puede ponerse en contacto con el administrador del sistema de la red.

> **Note**
>
> Supongamos que desea definir permisos granulares mediante una lista de permisos específicaURLs. Puede añadir parte de los URL que utiliza la AWS Systems Manager sesión para abrir una WebSocket conexión para enviar entradas y recibir salidas. Los AWS CloudShell comandos se envían a esa sesión de Systems Manager.
> El formato StreamUrl que utiliza Systems Manager es `wss://ssmmessages.`**`region`**`.amazonaws.com/v1/data-channel/`**`session-id`**`?stream=(input|output)`.
> La región representa el identificador de región de una Región de AWS región compatible con AWS Systems Manager. Por ejemplo, `us-east-2` es el identificador de región de la región Este de EE. UU. (Ohio).
> Como el identificador de sesión se crea después de que una sesión concreta de Systems Manager se haya iniciado correctamente, solo puede especificarlo `wss://ssmmessages.region.amazonaws.com` cuando actualice URL la lista de permitidos. Para

obtener más información, consulte la StartSessionoperación en la Referencia.AWS Systems Manager API

(Volver arriba)

No se pudo importar el módulo **AWSPowerShell.NetCore**

Problema: Al importar el AWSPowerShell. NetCoreen el módulo PowerShell de`Import-Module -Name AWSPowerShell.NetCore`, recibirá el siguiente mensaje de error:

Import-Module: el módulo especificado '. AWSPowerShell NetCore'no se cargó porque no se encontró un archivo de módulo válido en ningún directorio de módulos.

Causa: el `AWSPowerShell.NetCore` módulo se sustituye por los módulos AWS.Tools por servicio de. AWS CloudShell

Solución: es posible que las instrucciones de importación explícitas ya no sean necesarias o deban cambiarse al módulo .Tools correspondiente por servicio AWS.

Example

Example

- En la mayoría de los casos, siempre que no se utilice ningún tipo .Net, no necesitará ninguna declaración de importación explícita. Los siguientes son ejemplos de declaraciones de importación.
  - `Get-S3Bucket`
  - `(Get-EC2Instance).Instances`

- Si se utilizan los tipos .Net, importe el módulo de nivel de servicio (`AWS.Tools.<Service>`). A continuación, se muestra un ejemplo sintaxis .

```
Import-Module -Name AWS.Tools.EC2
$InstanceTag = [Amazon.EC2.Model.Tag]::new("Environment","Dev")
```

```
Import-Module -Name AWS.Tools.S3
$LifecycleRule = [Amazon.S3.Model.LifecycleRule]::new()
```

Para obtener más información, consulte el anuncio de la versión 4 para AWS Tools for PowerShell.

(Volver arriba)

Docker no se ejecuta cuando se usa AWS CloudShell

Problema: Docker no funciona correctamente cuando se usa. AWS CloudShell Recibe el siguiente mensaje de error:`docker: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?`.

Solución: intente reiniciar el entorno. Este mensaje de error puede aparecer al ejecutar Docker AWS CloudShell en una GovCloud región que no lo admite. Asegúrese de ejecutar Docker en las regiones compatibles AWS . Para ver una lista de las regiones en las que Docker está disponible, consulta las regiones compatibles AWS para AWS CloudShell

Docker se ha quedado sin espacio en disco

Problema: recibe el siguiente mensaje de error:`ERROR: failed to solve: failed to register layer: write [...]: no space left on device`.

Causa: el Dockerfile supera el espacio disponible en disco. AWS CloudShell Esto puede deberse a imágenes individuales de gran tamaño o a demasiadas imágenes de Docker preexistentes.

Solución: ejecute `df -h` para averiguar el uso del disco. Ejecute `sudo du -sh /folder/folder1` para evaluar el tamaño de ciertas carpetas que considere grandes y considere eliminar otros archivos para liberar espacio. Una opción sería considerar la posibilidad de eliminar las imágenes de Docker no utilizadas mediante la ejecución. `docker rmi` Debe tener en cuenta que el espacio de Docker en el entorno es limitado. Para obtener más información sobre Docker, consulte la guía de documentación de Docker.

`docker push`se está agotando el tiempo de espera y lo sigue intentando

Problema: cuando `docker push` lo ejecutas, se agota el tiempo de espera y sigue intentándolo sin éxito.

Causa: esto puede deberse a la falta de permisos, al envío a un repositorio incorrecto o a una falta de autenticación.

Solución: Para intentar resolver este problema, asegúrate de que estás accediendo al repositorio correcto. Ejecute `docker login` para autenticarse correctamente. Asegúrate de tener todos los permisos necesarios para acceder a un ECR repositorio de Amazon.

No puedo acceder a los recursos VPC de mi AWS CloudShell VPC entorno

Problema: no puedo acceder a los recursos VPC mientras uso mi AWS CloudShell VPC entorno.

Causa: su AWS CloudShell VPC entorno hereda su configuración de red. VPC

Solución: Para resolver este problema, asegúrese de que VPC está configurado correctamente para acceder a sus recursos. Para obtener más información, consulte VPC la documentación Conéctese VPC a otras redes y la documentación Network Access Analyzer y la documentación Network Access Analyzer. Para encontrar la IPv4 dirección que utiliza el AWS CloudShell VPC entorno, ejecute el comando `ip -a` dentro de su entorno en la línea de comandos o en la página de la VPC consola.

La ENI utilizada por AWS CloudShell para mi VPC entorno no está limpia

Problema: No se puede limpiar lo ENI usado por AWS CloudShell para mi VPC entorno.

Causa: `ec2:DeleteNetworkInterface` el permiso no está habilitado para su función.

Solución: para resolver este problema, asegúrese de que el `ec2:DeleteNetworkInterface` permiso esté habilitado para su rol, como se muestra en el siguiente script de ejemplo:

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

El usuario con **`CreateEnvironment`** permiso exclusivo para VPC entornos también tiene acceso a AWS CloudShell entornos públicos

Problema: Los usuarios restringidos con `CreateEnvironment` permiso exclusivo para VPC entornos también pueden acceder a AWS CloudShell entornos públicos.

Causa: si limita `CreateEnvironment` los permisos para la creación de VPC entornos únicamente y si ya ha creado un entorno público, conservará el acceso al CloudShell entorno público existente hasta que lo elimine mediante la interfaz de usuario web. Sin embargo, si nunca lo ha utilizado CloudShell antes, no tendrá acceso a los entornos públicos.

Solución: para restringir el acceso a los AWS CloudShell entornos públicos, el IAM administrador primero debe actualizar la IAM política con la restricción y, a continuación, el usuario debe eliminar manualmente el entorno público existente mediante la interfaz de usuario AWS CloudShell web. (Acciones → Eliminar CloudShell entorno).

Las credenciales no funcionan en CloudShell

Problema: Al intentar realizar una AWS CLI llamada CloudShell, aparece el mensaje de error Error interno del servidor.

Causa: las posibles causas de este problema son las siguientes:

- Se produjo un error en la `putCredentials` API llamada que se CloudShell utiliza para actualizar las credenciales. La API llamada puede fallar debido a la falta de IAM permisos para `putCredentials` actuar. Para obtener más información, consulte ???. Si ya tienes IAM permisos para `putCredentials` actuar, la API llamada puede fallar debido a problemas de red o problemas operativos con ella CloudShell.
- Sus credenciales ya no son válidas porque la sesión de la AWS consola ha caducado, pero su CloudShell entorno sigue funcionando. Cuando sus credenciales ya no sean válidas, CloudShell no podrá realizar ninguna API llamada.

Solución: intente actualizar la página web si ya tiene los IAM permisos necesarios para `putCredentials` actuar. Si el problema no se resuelve y sigues recibiendo el error, ponte en contacto con AWS Support.

AWS Regiones compatibles para AWS CloudShell

En esta sección se incluye la lista de AWS regiones compatibles y regiones en las que se ha optado por AWS CloudShell participar. Para obtener una lista de los puntos finales de AWS servicio y las cuotas correspondientes CloudShell, consulte la AWS CloudShell página del. Referencia general de Amazon Web Services

Las siguientes son las AWS regiones compatibles para CloudShell Docker y CloudShell VPC el entorno:

- US East (Ohio)
- Este de EE. UU. (Norte de Virginia)
- Oeste de EE. UU. (Norte de California)
- Oeste de EE. UU. (Oregón)
- África (Ciudad del Cabo)
- Asia-Pacífico (Hong Kong)
- Asia-Pacífico (Yakarta)
- Asia Pacific (Bombay)
- Asia-Pacífico (Osaka)
- Asia-Pacífico (Seúl)
- Asia-Pacífico (Singapur)
- Asia-Pacífico (Sídney)
- Asia-Pacífico (Tokio)
- Canadá (centro)
- Europa (Fráncfort)
- Europa (Irlanda)
- Europa (Londres)
- Europa (Milán)
- Europa (París)
- Europa (Estocolmo)
- Medio Oriente (Baréin)
- Oriente Medio () UAE

- América del Sur (São Paulo)

GovCloud Regiones

Las siguientes son las GovCloud regiones compatibles para CloudShell:

- AWS GovCloud (Este de EE. UU.)
- AWS GovCloud (Estados Unidos-Oeste)

Actualmente, Docker y el CloudShell VPC entorno no están disponibles en GovCloud las regiones.

Cuotas y restricciones de servicio para AWS CloudShell

En esta página se describen las Service Quotas y restricciones que se aplican a las siguientes áreas:

- Almacenamiento persistente
- Uso mensual
- Tamaño del comando
- Intérprete de comandos simultáneos
- Sesiones del intérprete de comandos
- Acceso a la red y transferencia de datos
- Archivos del sistema y recargas de páginas

Almacenamiento persistente

Con AWS CloudShell, dispone de un almacenamiento persistente de 1 GB para cada uno sin Región de AWS coste alguno. El almacenamiento persistente se encuentra en tu directorio principal ($HOME) y es privado para ti. A diferencia de los recursos efímeros del entorno que se reciclan al finalizar cada sesión del intérprete de comandos, los datos del directorio principal persisten entre las sesiones.

> **Note**
>
> CloudShell VPClos entornos no tienen almacenamiento persistente. El HOME directorio $ se elimina cuando se agota el tiempo de espera del VPC entorno (tras 20 a 30 minutos de inactividad) o cuando se elimina el entorno.

Si deja de usarlo AWS CloudShell en una región Región de AWS, los datos se conservan en el almacenamiento persistente de esa región durante 120 días después del final de la última sesión. Transcurridos 120 días, a menos que realice alguna acción, sus datos se eliminarán automáticamente del almacenamiento persistente de esa región. Puede evitar que se eliminen iniciando AWS CloudShell de nuevo en esa Región de AWS. Para obtener más información, consulta el paso 2: selecciona una región AWS CloudShell, lanza y elige un shell.

> **Note**
>
> Escenario de uso
> Márcia AWS CloudShell solía almacenar archivos en sus directorios principales en dos de ellos Regiones de AWS: EE.UU. Este (Norte de Virginia) y Europa (Irlanda). Luego comenzó a usar AWS CloudShell exclusivamente en Europa (Irlanda) y dejó de lanzar sesiones ficticias en el este de EE. UU. (Virginia del Norte).
> Antes de que venza la fecha límite para eliminar datos en EE.UU. Este (Norte de Virginia), Márcia decide impedir que su directorio principal sea reciclado abriendo AWS CloudShell y seleccionando de nuevo la región EE.UU. Este (Norte de Virginia). Como ha utilizado continuamente Europa (Irlanda) para las sesiones de intérprete de comandos, su almacenamiento persistente en esa región no se ve afectado.

Uso mensual

Cada una Región de AWS de las tuyas Cuenta de AWS tiene una cuota de uso mensual de AWS CloudShell. Esta cuota combina el tiempo total dedicado CloudShell al consumo por todos los IAM directores de esa región. Si intentas acceder CloudShell después de haber alcanzado la cuota mensual de esa región, aparecerá un mensaje en el que se explica por qué no se puede iniciar el entorno shell.

> **Note**
>
> Si necesita aumentar sus cuotas de uso mensuales, póngase en contacto con AWSSupport con la siguiente información:
>
> - CloudShell Región de uso
> - Tu caso de uso. Por ejemplo, AWS CLI operación y ejecución de comandos de Linux
> - El número de CloudShell usuarios. Por ejemplo, 5-10
> - La estimación máxima del tiempo que pasas CloudShell en la región
> - CloudShellVPCuso del entorno
>
> Podemos aprobar el aumento del tiempo máximo estimado a 1000 horas por mes, en comparación con el límite actual de 200 horas.

Tamaño del comando

El tamaño del comando no puede superar los 65412 caracteres.

> **Note**
>
> Si pretende ejecutar un comando que supere los 65412 caracteres, cree un script con el lenguaje de programación que prefiera y ejecútelo desde la interfaz de la línea de comandos. Para obtener más información sobre la gama de software preinstalado al que se puede acceder desde la interfaz de la línea de comandos, consulte la sección Software preinstalado.
>
> Para ver un ejemplo de cómo crear un script y, a continuación, ejecutarlo desde la interfaz de la línea de comandos, consulte Tutorial: Primeros pasos con AWS CloudShell.

Intérprete de comandos simultáneos

- Proyectiles simultáneos: puedes lanzar hasta 10 proyectiles al mismo tiempo en cada uno de ellos Región de AWS para tu cuenta.

Sesiones del intérprete de comandos

- Sesiones inactivas: AWS CloudShell es un entorno de shell interactivo. Si no interactúas con él con el teclado o el puntero durante 20 o 30 minutos, la sesión de shell finaliza. Los procesos en ejecución no cuentan como interacciones.
- Sesiones de larga duración: una sesión del intérprete de comandos que se ejecute de forma continua durante aproximadamente 12 horas finaliza automáticamente aunque el usuario interactúe habitualmente con ella durante ese período.

Acceso a la red y transferencia de datos

Las siguientes restricciones se aplican tanto al tráfico entrante como al saliente de su entorno AWS CloudShell :

- Saliente: puede acceder a la red de Internet público.

- Entrante: no puede acceder a los puertos entrantes. No hay ninguna dirección IP pública disponible.

> **Warning**
>
> Con el acceso a la Internet pública, existe el riesgo de que algunos usuarios exporten datos del entorno. AWS CloudShell Recomendamos que IAM los administradores gestionen la lista de AWS CloudShell usuarios de confianza permitidos mediante IAM herramientas. Para obtener información sobre cómo se puede denegar el acceso de forma explícita a usuarios específicos, consulte Administrar las acciones permitidas mediante el AWS CloudShell uso de políticas personalizadas.

Transferencia de datos: la carga y descarga de archivos de origen y destino AWS CloudShell puede ser lenta en el caso de archivos de gran tamaño. Como alternativa, puede transferir archivos a su entorno desde un bucket de Amazon S3 mediante la interfaz de la línea de comandos del intérprete de comandos.

Restricciones en los archivos del sistema y en la recarga de páginas

- Archivos del sistema: si modifica incorrectamente los archivos necesarios para el entorno informático, es posible que tenga problemas al acceder o utilizar el AWS CloudShell entorno. Si esto ocurre, es posible que tenga que eliminar su directorio principal para recuperar el acceso.
- Recargar páginas: para volver a cargar la interfaz de AWS CloudShell , utilice el botón de actualización del navegador en lugar de utilizar la secuencia de atajos de teclado predeterminada del sistema operativo.

Historial de documentos de la Guía AWS CloudShell del usuario

Actualizaciones recientes

En la siguiente tabla se describen cambios importantes en la Guía del usuario de AWS CloudShell .

Cambio	Descripción	Fecha
VPCSoporte de Amazon AWS CloudShell en determinadas regiones	Se ha añadido soporte para la creación y el uso de AWS CloudShell VPC entornos en determinadas regiones.	13 de junio de 2024
Se han añadido nuevos tutoriales a la Guía AWS CloudShell del usuario	Se han agregado dos nuevos tutoriales que detallan cómo crear un contenedor Docker en su interior AWS CloudShel l y enviarlo a un ECR repositor io de Amazon, y cómo implementar una función Lambda mediante. AWS CDK	27 de diciembre de 2023
En determinadas regiones, los contenedores Docker son compatibles AWS CloudShell	Se AWS CloudShell ha añadido soporte para contenedores Docker con en algunas regiones.	27 de diciembre de 2023
AWS CloudShell ha migrado para usar ahora Amazon Linux 2023 (AL2023)	AWS CloudShell ahora usa AL2 023 y ha migrado desde Amazon Linux 2.	4 de diciembre de 2023
Nuevas AWS regiones para AWS CloudShell	AWS CloudShell ahora está disponible de forma general en las siguientes AWS regiones:	16 de junio de 2023

	• Oeste de EE. UU. (Norte de California) • África (Ciudad del Cabo) • Asia Pacific (Hong Kong) • Asia-Pacífico (Osaka) • Asia-Pacífico (Seúl) • Asia-Pacífico (Yakarta) • Asia-Pacífico (Singapur) • Europa (París) • Europa (Estocolmo) • Europe (Milan) • Middle East (Bahrain) • Oriente Medio (UAE)	
Lanzamiento AWS CloudShell en el Console Toolbar	CloudShell Lánzalo en el Console Toolbar, en la parte inferior izquierda de la consola, seleccionando CloudShell.	28 de marzo de 2023
Nuevas AWS regiones para AWS CloudShell	AWS CloudShell ya está disponible en las siguientes AWS regiones: • Canadá (centro) • Europe (Londres) • América del Sur (São Paulo)	6 de octubre de 2022
AWS CloudShell compatible en EE. UU. AWS GovCloud	AWS CloudShell ahora es compatible en la región AWS GovCloud (EE. UU.).	29 de junio de 2022
Seguridad FAQs	Además, FAQs se centra en cuestiones de seguridad.	14 de abril de 2022

Web Sockets	Se agregó una sección a los requisitos CloudShell de la red que explica el uso del WebSocket protocolo.	21 de marzo de 2022
Solución de problemas con las teclas de flecha en PowerShell	Siga los pasos para corregir las teclas de flecha que generan letras de forma incorrecta al presionarlas.	7 de febrero de 2022
La tecla de tabulación se completa automáticamente	Nueva documentación que explica cómo usar bash-completion, que permite completar automáticamente comandos o argumentos escritos parcialmente pulsando la tecla Tab.	24 de septiembre de 2021
Especificación de AWS regiones	Documentación sobre la especificación de Región de AWS los AWS CLI comandos por defecto.	11 de mayo de 2021
Formateo PDF y versiones Kindle	Tamaños de imagen y texto fijos en las celdas de la tabla.	10 de marzo de 2021

Versión de disponibilidad general (GA) AWS CloudShell en determinadas AWS regiones	AWS CloudShell ahora está disponible de forma general en las siguientes AWS regiones: • US East (Ohio) • Este de EE. UU. (Norte de Virginia) • Oeste de EE. UU. (Oregón) • Asia-Pacífico (Tokio) • Europe (Irlanda) • Asia Pacific (Bombay) • Asia-Pacífico (Sídney) • Europe (Fráncfort)	15 de diciembre de 2020

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.