Guía del usuario de

AWS CloudShell

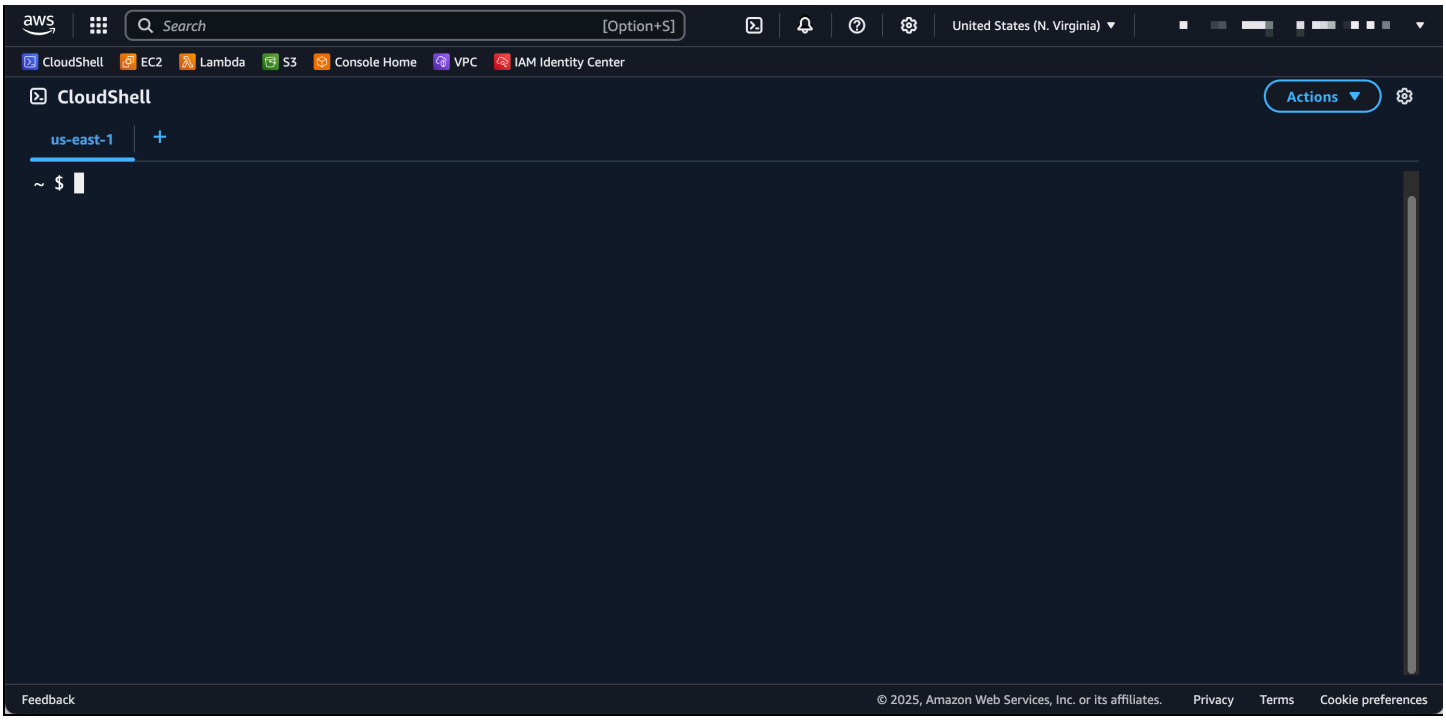
AWS CloudShell: Guía del usuario de

Table of Contents

¿Qué es AWS CloudShell?

AWS CloudShell es un shell preautenticado y basado en un navegador que se puede iniciar directamente desde. Consola de administración de AWS Puede navegar CloudShell desde varias formas diferentes Consola de administración de AWS . Para obtener más información, consulte Cómo empezar con AWS CloudShell

Puede ejecutar AWS CLI comandos con el shell que prefieraBash, como PowerShell, oZ shell. Y puede hacerlo sin necesidad de descargar ni instalar herramientas de la 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. Se puede utilizar CloudShell en las versiones más recientes de los navegadores Google Chrome, Mozilla Firefox, Microsoft Edge y Apple Safari.

(Pruébelo ahora: Introducción a con () AWS CloudShell)

Características de AWS CloudShell

AWS CloudShell ofrece las siguientes características:

AWS Command Line Interface

Puede iniciar AWS CloudShell desde Consola de administración de AWS. 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, consulteAdministre AWS los servicios desde CLI en 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, yZ shell. También tiene acceso a otras herramientas y utilidades pre instaladas. Se incluyen las siguientes: git, make, pip, sudo, tar, tmux, vim, wget y zip.

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

Para obtener más información, consulte Entorno de computación de AWS CloudShell: especificaciones y software.

Almacenamiento persistente

Con AWS CloudShell, puedes utilizar 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 Los entornos de VPC no tienen almacenamiento persistente. El directorio $HOME se elimina cuando se agota el tiempo de espera del entorno de VPC (después de unos 20-30 minutos de inactividad), o bien cuando elimina o reinicia su entorno.

CloudShell Entornos de VPC

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

Seguridad

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

Gestión de permisos con IAM

Como administrador, puede conceder y denegar permisos a los AWS CloudShell usuarios mediante las políticas de IAM. 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 las políticas de 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 su experiencia AWS CloudShell.

También puede ampliar su entorno de intérprete de comandos si instala su propio software y modifica el intérprete de comandos con scripts.

Restauración de sesión

La función 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.

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 Service Quotas y restricciones para AWS CloudShell.

AWS CloudShell Temas clave

- Introducción a con () AWS CloudShell

- AWS CloudShell Conceptos
- Administre AWS los servicios desde CLI en CloudShell
- Personalización de su experiencia AWS CloudShell
- Entorno de computación de AWS CloudShell: especificaciones y software

Introducción a con () AWS CloudShell

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

En primer lugar, debe iniciar sesión en la Consola de administración de AWS y seleccionar una Región de AWS. A continuación, se inicia CloudShell en una nueva ventana del navegador y en un tipo de intérprete de comandos 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 llamar a comandos de la AWS CLI para crear un bucket de Amazon S3 y añadir su archivo como objeto al bucket.

Requisitos previos

Permisos de IAM

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

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

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

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: regístrese en Consola de administración de AWS

Este paso implica introducir su información de usuario de IAM para acceder a la Consola de administración de AWS. Si ya está en la consola, vaya al paso 2.

- Puede obtener acceso a la Consola de administración de AWS mediante una URL de inicio de sesión de usuarios de IAM o en la página principal de inicio de sesión.

  IAM user sign-in URL

  - Abra un navegador y escriba la siguiente URL de inicio de sesión. 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/
    ```

- Introduzca sus credenciales de inicio de sesión de IAM y seleccione Iniciar sesión.

Main sign-in page

- Abra https://aws.amazon.com/console/.
- Si no ha iniciado sesión anteriormente en este navegador, aparecerá la página principal de inicio de sesión. Elija usuario de IAM, introduzca el alias de cuenta o el ID de cuenta y elija Siguiente.
- Si ha iniciado sesión anteriormente como usuario de IAM. Es posible que el navegador recuerde el alias o el ID de la cuenta de Cuenta de AWS. En ese caso, introduzca sus credenciales de inicio de sesión de IAM y seleccione Registrarse.

> **Note**
>
> También puede iniciar sesión como usuario raíz. Esta identidad tiene acceso completo a todos los Servicios de AWS y recursos de la cuenta. Se recomienda encarecidamente no utilizar el usuario raíz para las tareas cotidianas, ni siquiera para las tareas administrativas. En lugar de ello, es mejor ceñirse a la práctica recomendada de utilizar el usuario final exclusivamente para crear al primer usuario de IAM.

Paso 2: seleccione una región, inicie AWS CloudShell y elija un intérprete de comandos

En este paso, inicie CloudShell desde la interfaz de la consola, seleccione una Región de AWS disponible y cambie al intérprete de comandos que prefiera, por ejemplo Bash, PowerShell o Z shell.

1. Para elegir una Región de AWS en la que trabajar, vaya al menú Seleccionar una región y seleccione una región de AWS 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

Si CloudShell no está disponible en la región seleccionada al iniciar CloudShell en la 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 las Regiones de AWS.

Example

Ejemplo

Si elige Europa (España) eu-south-2 pero CloudShell no está disponible en Europa (España) eu-south-2, la región predeterminada se establece en Europa (Irlanda) eu-west-1, que es la más cercana a Europa (España) eu-south-2.

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

2. Desde la Consola de administración de AWS, puede iniciar CloudShell eligiendo una de las siguientes opciones:

   1. Elija el icono CloudShell en la barra de navegación.
   2. En el cuadro Buscar, escriba “CloudShell” y, a continuación, seleccione CloudShell.
   3. En el widget Visitas recientes, seleccione CloudShell.
   4. Elija el icono de CloudShell en la esquina inferior izquierda de la Console Toolbar de la consola.
      - Puede ajustar la altura de la sesión de CloudShell arrastrando =.
      - Puede cambiar la sesión de CloudShell a pantalla completa haciendo clic en 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 AWS CloudShell, busque información para identificarlos y solucionarlos en Solución 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 cambia a Bash, el símbolo de la línea de comandos se actualizará a $.

   > **Note**
   >
   > Bash es el intérprete de comandos predeterminado que se ejecuta cuando inicia AWS CloudShell.

   PowerShell

   ```
   pwsh
   ```

   Si cambia 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 actualizará a %.

   Para obtener información sobre las versiones preinstaladas en su entorno de intérprete de comandos, consulte la tabla de intérprete de comandos en la sección del entorno de computación AWS CloudShell.

Paso 3: descargue un archivo desde AWS CloudShell

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

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 Descargar no está disponible al iniciar CloudShell en la Console Toolbar. Puede descargar un archivo desde la consola de CloudShell o mediante el navegador web Chrome.

Paso 4: cargue un archivo en AWS CloudShell

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

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: elimine 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 del intérprete de comandos 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 usar AWS CloudShell en un Región de AWS específico. 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 fuera de CloudShell

Puede usar una de las siguientes opciones para transferir el archivo de copia de seguridad fuera de 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 de 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}
```

Utilice la CLI de AWS para copiar el archivo en el 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 S

   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 del intérprete de comandos.

> **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 del intérprete de comandos, seleccione Acciones, Reiniciar.

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

2. Para confirmar, seleccione Reiniciar.

   Una interfaz muestra un mensaje que indica que el entorno de computación de CloudShell 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 del intérprete de comandos.

> **Note**
>
> Esta opción no está disponible para los entornos de VPC. Al reiniciar un entorno de VPC, 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 de computación AWS CloudShell. Al eliminar el directorio principal, AWS CloudShell restablece su configuración predeterminada.
> - Quiere eliminar todos sus datos de AWS CloudShell de forma inmediata. Si deja de usar AWS CloudShell en una región de AWS, el almacenamiento persistente se eliminará automáticamente al final del período de retención, a menos que vuelva a iniciar AWS CloudShell en esa región.

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

1. Para eliminar una sesión del intérprete de comandos, elija Acciones, Eliminar.

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

   > **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 Región de AWS actual. Puede crear un nuevo entorno o configurar un entorno de VPC de CloudShell.

3. Para crear un nuevo entorno, elija Abrir una pestaña.
4. Para crear un entorno de VPC en CloudShell, elija Crear un entorno de VPC.

   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 editor Vim preinstalado 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 intérprete de comandos se actualiza para mostrar el editor Vim.

2. Para editar el archivo en Vim, pulse la tecla I. 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 entrar en el modo de comando Vim. A continuación, introduzca el comando `:wq` para guardar el archivo y salir del editor.

> **Note**
>
> Si es nuevo en el modo de comandos Vim, 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 sobre Vim y otras herramientas que están disponibles en AWS CloudShell, consulte Herramientas 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: use AWS CLI para añadir el archivo como un objeto en un bucket de Amazon S3

En este paso, cree un bucket de Amazon S3 y, a continuación, utilice el método PutObject para añadir el archivo de código como un objeto en ese depósito.

> **Note**
>
> En este tutorial, se muestra cómo puede usar la AWS CLI en AWS CloudShell para interactuar con otros servicios de AWS. 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 una cuenta específica de una Región de AWS, introduzca 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 cumple las reglas para asignar nombres a buckets, aparece el siguiente error: se ha producido un error (InvalidBucketName) al llamar a la operación CreateBucket: el bucket 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

- Administre AWS los servicios desde CLI en CloudShell
- Copia de varios archivos entre la máquina local y CloudShell
- AWS CloudShell Conceptos
- Personalización de su experiencia AWS CloudShell

AWS CloudShellTutoriales de

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

Información general del tutorial	Más información
Copia de varios archivos	the section called “Tutorial: copiar varios archivos”
Creación de URL prefirmadas	???
Creación de un contenedor de Docker en AWS CloudShell e inserción de este en Amazon ECR	???
Implementación de una función de Lambda mediante AWS CDK	???

Copia de varios archivos entre la máquina local y CloudShell

En este tutorial se muestra cómo copiar varios archivos entre la máquina local y CloudShell.

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

- Amazon S3: utilice buckets de S3 como intermediarios 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 de CloudShell.

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

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 política de IAM que conceda permisos para realizar las siguientes acciones de la API 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 Referencia de la API de Amazon Simple Storage Service.

Cargue varios archivos a AWS CloudShell mediante 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 comando `s3`:

   ```
   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:

   - Consola de administración de AWS: utilice la función de arrastrar y soltar para cargar archivos y carpetas en un bucket.
   - 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 de 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 esta opción, debe tener la herramienta de la AWS CLI instalada en su máquina local y tener sus credenciales configuradas para las llamadas a los servicios de AWS. Para obtener más información, consulte la Guía del usuario de AWS Command Line Interface.

- Inicie la AWS CLI y ejecute el siguiente comando `aws s3` para sincronizar el bucket 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 comandos de CloudShell e introduzca el siguiente comando para sincronizar el directorio del entorno del intérprete de comandos 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 de AWS CloudShell mediante Amazon S3

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

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

   ```
   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 herramienta de la AWS CLI, 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 herramienta de la AWS CLI instalada en su máquina local y tener sus credenciales configuradas para las llamadas a los servicios de AWS. Para obtener más información, consulte la Guía del usuario de AWS Command Line Interface.

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 de computación de CloudShell.

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 a través de carpetas comprimidas

Este paso 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 CloudShell y, a continuación, seleccione Acciones, 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 comandos de CloudShell, ejecute el siguiente comando para descomprimir el contenido del archivo zip en un directorio específico:

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

Descargue varios archivos desde AWS CloudShell mediante carpetas comprimidas

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

1. En la línea de comandos de CloudShell, 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.

Creación de una URL prefirmada para objetos de Amazon S3 mediante CloudShell

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

Requisitos previos

- Un usuario de IAM con permisos de acceso otorgados por la política AWSCloudShellFullAccess.
- Para conocer los permisos de IAM necesarios para crear una URL prefirmada, consulte Compartir un objeto con otros en la Guía del usuario de Amazon Simple Storage Service.

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

Este paso describe cómo crear un rol de IAM para conceder acceso a un bucket de Amazon S3.

1. Para obtener los detalles de IAM que se puedan compartir, llame al comando `get-caller-identity` desde 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 CloudFormation. Esta plantilla crea un rol de IAM. 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. Guarde la plantilla de CloudFormation en un archivo que se llame `template.yaml`.
4. Use la plantilla para implementar la pila y crear el rol de IAM mediante una llamada al comando `deploy`.

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

Generar una URL prefirmada

En este paso se describe cómo generar la URL prefirmada.

1. Con el editor de AWS CloudShell, añada el siguiente código. Este código crea una URL que facilita a los usuarios federados acceso directo a la Consola de administración de AWS.

```
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 rol de IAM de CloudFormation. A continuación, úselo en el script de Python 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 una URL en la que un colaborador puede hacer clic para acceder a AWS CloudShell en la Consola de administración de AWS. 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 repositorio de Amazon ECR.

Requisitos previos

- Debe tener los permisos necesarios para crear e insertar en un repositorio de Amazon ECR. Para obtener más información sobre los repositorios con Amazon ECR, consulte Repositorios privados de Amazon ECR en la Guía del usuario de Amazon ECR. Para obtener más información sobre los permisos necesarios para insertar imágenes con Amazon ECR, consulte los permisos de IAM necesarios para la inserción de una imagen en la Guía del usuario de Amazon ECR.

Procedimiento del tutorial

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

1. Cree una carpeta en el directorio principal.

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

2. Vaya a la carpeta que ha creado.

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

3. Cree un archivo Dockerfile vacío.

```
touch Dockerfile
```

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

```
# 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 archivo Dockerfile está ahora listo para crearse. Ejecute `docker build` para crear el contenedor. 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 (`.`).
   Imagen del comando docker build que se ejecuta en AWS CloudShell.

6. Ahora puede probar si el contenedor se ejecuta correctamente en AWS CloudShell.

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

   Imagen del comando docker container run en su interior AWS CloudShell

7. Ahora que tiene un contenedor de Docker en funcionamiento, debe insertarlo en un repositorio de Amazon ECR. Si ya tiene un repositorio de Amazon ECR puede omitir este paso.

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

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

   Imagen del comando utilizado para crear un repositorio de Amazon ECR en AWS CloudShell

8. Después de crear el repositorio de Amazon ECR, puede insertar el contenedor de Docker en este.

   Ejecute el siguiente comando para obtener las credenciales de inicio de sesión de Amazon ECR 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}
   ```

   Imagen del comando que se usa para obtener las credenciales de inicio de sesión de Amazon ECR para Docker.

> **Note**
>
> Si la variable de AWS_REGION entorno no está configurada en su CloudShell servidor 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 repositorio de Amazon ECR de destino y, a continuación, insértela en ese repositorio.

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

   Imagen del comando que se usa para etiquetar la imagen con el repositorio de Amazon ECR de destino.

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

Limpieza

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

- ```
  cd ~
  rm -rf ~/docker-cli-tutorial
  ```
- Elimine el repositorio de Amazon ECR.

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

Implementación de una función Lambda mediante el AWS CDK 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

- Arranque su cuenta para usarla con AWS CDK. Para 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

- Asegúrese de tener los permisos adecuados para implementar recursos en la cuenta. Se recomiendan permisos de administrador.

Procedimiento del tutorial

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

1. Cree una carpeta en el directorio principal.

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

2. Vaya a la carpeta que ha creado.

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

3. Instale las dependencias de forma local. AWS CDK

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

   Imagen del comando utilizado para instalar las AWS CDK dependencias.

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 este.

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

6. Abra el archivo `lib/docker-tutorial.js` y pegue en este el siguiente contenido.

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

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

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

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

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

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

7. Abra `lib/Dockerfile` y pegue en este el siguiente contenido.

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

# Copy the function code to the LAMBDA_TASK_ROOT directory
```

```
# This environment variable is provided by the lambda base image
COPY hello.js ${LAMBDA_TASK_ROOT}

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

8. Abra el archivo `lib/hello.js` y pegue en este el siguiente contenido.

```
// 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 la AWS CDK CLI para sintetizar el proyecto e implementar los recursos. Debe arrancar su cuenta.

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

Imagen del comando para usar la AWS CDK CLI para sintetizar el proyecto e implementar los recursos.

10. Invoque la función de Lambda para confirmar y verifíquela.

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

Imagen del comando que se usa para invocar la función de Lambda.

Acaba de implementar correctamente una función de Lambda basada en contenedores de Docker mediante AWS CDK. Para obtener más información AWS CDK, consulte la Guía para desarrolladores de AWS CDK la versión 2. Si encuentra errores o tiene problemas al intentar completar este tutorial, consulte la sección Solución de problemas de esta guía para obtener ayuda.

Limpieza

Acaba de implementar correctamente una función de 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
  ```
  ```

AWS CloudShell Conceptos

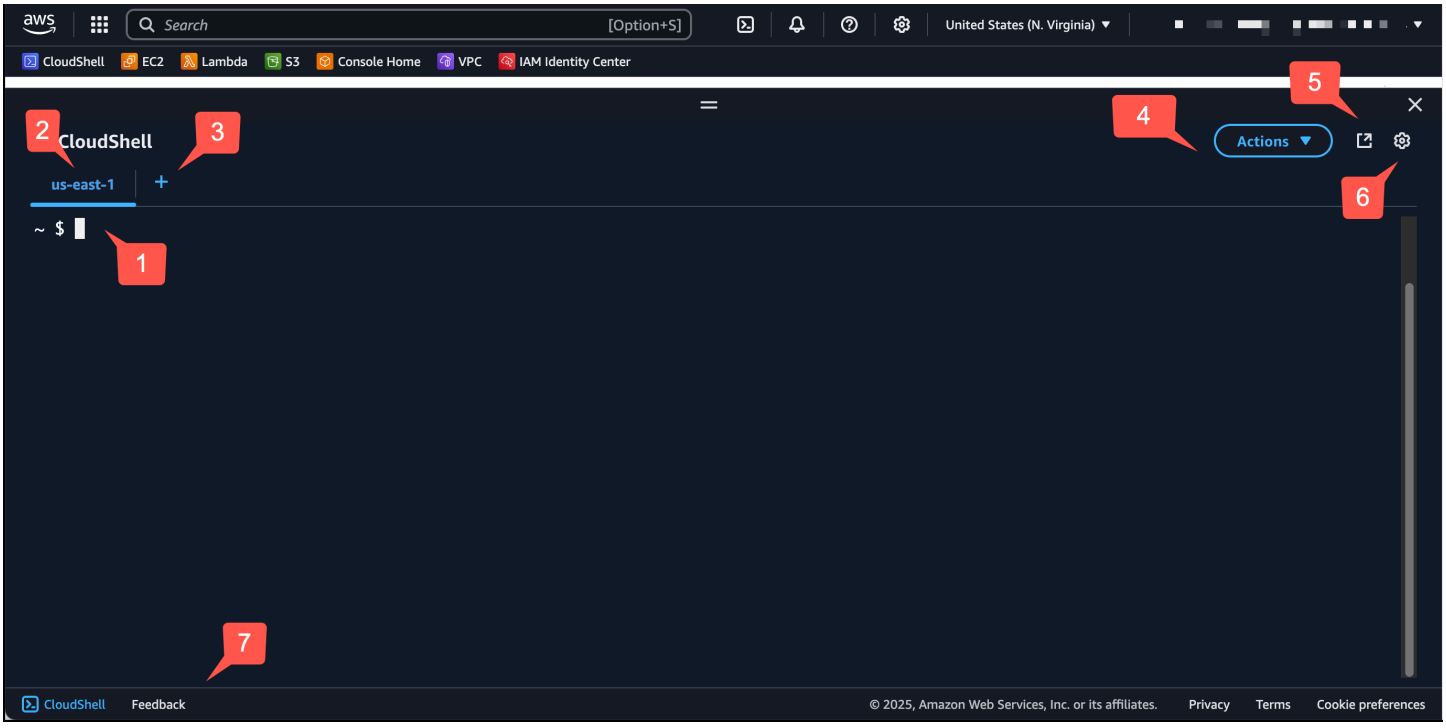
En esta sección se describe cómo interactuar con las aplicaciones compatibles AWS CloudShell y realizar acciones específicas con ellas.

Temas

Navegar por la interfaz AWS CloudShell

Puede navegar por las funciones de la CloudShell interfaz desde Consola de administración de AWS yConsole 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 Región de AWS dónde AWS CloudShell se está ejecutando actualmente.
3. El icono + es un menú desplegable que incluye opciones para crear entornos, reiniciarlos y eliminarlos.
4. 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 enConsole Toolbar.

5. La pestaña Abrir en un navegador nuevo, que ofrece la opción de acceder a la CloudShell sesión en pantalla completa.
6. La opción Preferencias, que puede utilizar para personalizar su experiencia de intérprete de comandos.
7. La barra inferior, que ofrece las siguientes opciones para:
   - Inicie CloudShell 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.
       - Selecciona CloudShellen la Console Toolbar esquina 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 Comentarios no está disponible cuando inicias CloudShell enConsole Toolbar.

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

Trabajando en Regiones de AWS

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

Puedes 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 los entornos de AWS CloudShell VPC.

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 Consola de administración de AWS 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.

Acceso CloudShell en la aplicación Console Mobile Application

Puede acceder AWS Console Mobile Application desde CloudShell la pantalla de inicio. Desde la pantalla de inicio, puede ver información CloudShell y otros AWS servicios. Para obtener más información, consulte Introducción a AWS Console Mobile Application. Para iniciar CloudShell en AWS Console Mobile Application, elige una de las siguientes opciones:

- Seleccione el icono CloudShell de la parte inferior de la barra de navegación.
- Seleccione CloudShellen el menú Servicios.

Puede salir CloudShell en cualquier momento seleccionando X.

Para obtener más información sobre el acceso a CloudShell la aplicación Console Mobile Application, consulte Acceso AWS CloudShell.

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

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, mediante el AWS CDK kit de herramientas, así como crear contenedores de Docker y enviarlos a los repositorios de Amazon ECR a través de la CLI de Docker. Para ver pasos detallados sobre cómo ejecutar ambas implementaciones, consulte los siguientes tutoriales:

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

Hay ciertas restricciones y limitaciones en 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, esto puede provocar problemas que le impidan extraer, crear o ejecutar imágenes adicionales. Para obtener más información sobre Docker, consulte la Guía de documentación de Docker.
- Docker está disponible en todas AWS las regiones, excepto en las regiones de AWS GovCloud (EE. UU.). Para ver una lista de las regiones en las que Docker está disponible, consulte 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.

Características de accesibilidad para AWS CloudShell

En este tema se describe cómo utilizar las funciones de accesibilidad de CloudShell. Puede utilizar un teclado para navegar por los elementos enfocables de la página. También puede personalizar el aspecto 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.

Funciones de accesibilidad del terminal CloudShell

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 terminales de CloudShell 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”.

Administre AWS los servicios desde CLI en CloudShell

Una ventaja clave AWS CloudShell es que puede usarla 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
- La CLI de Amazon ECS
- AWS SAM

Y como ya has iniciado sesión AWS, no es necesario que configures tus credenciales de forma local antes de usar los servicios. Las credenciales que utilizó para iniciar sesión en la Consola de administración de AWS 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 información sobre la sintaxis, consulte en la referencia de comandos de la CLI de AWS.

- DynamoDB
- Amazon EC2
- Amazon Glacier

DynamoDB

DynamoDB es un servicio de bases de datos NoSQL totalmente administrado que proporciona un rendimiento rápido y predecible, así como una perfecta escalabilidad. La implementación del modo NoSQL en este servicio admite estructuras de datos de documentos y valores clave.

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

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

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

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) 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 comando `run-instances` lanza una instancia t2.micro en la subred especificada en una 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.

Amazon Glacier

Amazon Glacier y Amazon Glacier Deep Archive son clases de almacenamiento en la nube de Amazon S3 seguras, duraderas y de muy bajo coste para el archivo de datos y las copias de seguridad 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 Glacier con la AWS CLI en la Guía del usuario de AWS Command Line Interface .

AWS CLI de Elastic Beanstalk

La AWS Elastic Beanstalk CLI proporciona una interfaz de línea de comandos diseñada para simplificar la creación, actualización y supervisión de 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 comando `create` crea un entorno nuevo en una nube privada virtual (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 la EB CLI en la Guía para desarrolladores de AWS Elastic Beanstalk .

La CLI de Amazon ECS

La interfaz de la línea de comandos (CLI) de Amazon Elastic Container Service (Amazon ECS) ofrece 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 clúster de Amazon ECS es una agrupación lógica de tareas o servicios).

El siguiente comando `configure` configura la CLI de Amazon ECS 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 comandos de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.

AWS SAM CLI

AWS SAM CLI es 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 comando `init` inicializa un nuevo proyecto SAM con los parámetros necesarios transferidos como parámetros:

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

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

Uso de la CLI de Amazon Q en CloudShell

> **Important**
>
> AWS CloudShell ha desactivado temporalmente la funcionalidad de chat de Amazon Q debido a un problema interno. Actualmente, lo estamos investigando y restableceremos esta funcionalidad lo antes posible. Mientras tanto, puede seguir utilizando el chat de Q en la Consola de administración de AWS.

La CLI de Amazon Q es una interfaz de la línea de comandos que permite interactuar con Amazon Q. Para obtener más información, consulte Uso de Amazon Q Developer en la línea de comandos en la Guía del usuario de Amazon Q Developer.

La CLI de Amazon Q en CloudShell permite interactuar con conversaciones en lenguaje natural, hacer preguntas y recibir respuestas de Amazon Q, todo desde su terminal. Puede obtener el comando del intérprete de comandos correspondiente, lo que reduce la necesidad de hacer búsquedas, recordar la sintaxis y recibir sugerencias de comandos a medida que escribe en el terminal.

> **Note**
>
> Actualmente, las características de la CLI de Amazon Q en CloudShell no están disponibles en su entorno de VPC de CloudShell.

Si no ve las características de la CLI de Amazon Q en CloudShell, póngase en contacto con el administrador para que le proporcione permisos de IAM. Para obtener más información, consulte Ejemplos de políticas basadas en identidad para Amazon Q Developer en la Guía del usuario de Amazon Q Developer.

En este capítulo se explica cómo puede utilizar las características de la CLI de Amazon Q en CloudShell.

Uso de las sugerencias en línea de Amazon Q en CloudShell

Las sugerencias en línea de Amazon Q en CloudShell proporcionan sugerencias de comandos a medida que escribe en el terminal. Para obtener más información, consulte Amazon Q insertado en la línea de comandos en la Guía del usuario de Amazon Q Developer.

Para usar las sugerencias en línea de Amazon Q en CloudShell

1. En la Consola de administración de AWS, elija CloudShell.
2. En el terminal de CloudShell, cambie al intérprete de comandos Z y empiece a escribir. Para cambiar al intérprete de comandos Z, escriba `zsh` en el terminal y, a continuación, pulse Intro.

   > **Note**
   >
   > Actualmente, Amazon Q en línea solo es compatible con el intérprete de comandos Z.

   Cuando empieza a escribir su comando, Amazon Q le hace sugerencias basadas en la entrada actual y en los comandos anteriores. Las sugerencias en línea se habilitan de forma automática.

Para deshabilitar las sugerencias en línea, ejecute el siguiente comando:

```
q inline disable
```

Para habilitar las sugerencias en línea, ejecute el siguiente comando:

```
q inline enable
```

Uso del comando q chat en CloudShell

El comando `q chat` permite hacer preguntas y recibir respuestas de Amazon Q, todo desde su terminal. Para iniciar una conversación con Amazon Q, ejecute el comando `q chat` en el terminal de CloudShell. Para obtener más información, consulte Chatting with Amazon Q in the CLI en la Guía del usuario de Amazon Q Developer.

Uso del comando q translate en CloudShell

El comando `q translate` permite escribir instrucciones en lenguaje natural. Para traducir con Amazon Q, ejecute el comando `q translate` en el terminal de CloudShell. Para obtener más

información, consulte Translating from natural language to bash en la Guía del usuario de Amazon Q Developer .

Finalización de comandos de la CLI en CloudShell

La finalización de la CLI en CloudShell proporciona sugerencias de comandos y opciones a medida que escribe en el terminal. Para obtener más información, consulte Generating command line completion en Amazon Q Developer User Guide.

Activación o desactivación de la CLI de Amazon Q

Puede activar o desactivar la CLI de Amazon Q seleccionando Preferencias, Activar la CLI de Amazon Q y Desactivar la CLI de Amazon Q. La CLI de Amazon Q le permite interactuar con instrucciones en lenguaje natural, hacer preguntas y obtener respuestas de Amazon Q, todo desde su terminal. Además, le proporciona sugerencias de comandos mientras escribe en el terminal. Cuando empieza a escribir en el terminal, Amazon Q sugiere opciones relevantes para completar su comando.

Política basada en identidad para la CLI de Amazon Q en CloudShell

Para usar la CLI de Amazon Q en CloudShell, asegúrese de tener los permisos de IAM necesarios. Para obtener más información, consulte Ejemplos de políticas basadas en identidad para Amazon Q Developer en la Guía del usuario de Amazon Q Developer.

Ejecución de un comando en CloudShell desde consolas de servicio de AWS

Puede ejecutar un comando en el terminal de CloudShell mediante las consolas Amazon ElastiCache y Amazon DocumentDB (compatible con MongoDB) en la Consola de administración de AWS.

Para ejecutar un comando en CloudShell desde otras consolas de servicio de AWS, La política de IAM asignada a su rol debe incluir permisos de `cloudshell:approveCommand`.

CloudShell se abre en la barra de herramientas de la consola y la ventana emergente Ejecutar comando aparece en CloudShell. En la ventana emergente Ejecutar comando, el comando aparece en el cuadro de comandos.

Para ejecutar un comando en el terminal de CloudShell, elija uno de los siguientes pasos:

1. Introduzca un nombre en el cuadro Nombre del nuevo entorno si no ha creado un entorno de VPC en CloudShell.

   Puede ver los detalles del entorno de VPC que se basan en los detalles de la VPC de su recurso.

   a. Elija Create and run.

      Este paso creará un nuevo entorno de VPC de CloudShell y ejecutará el comando en el terminal de CloudShell.
2. Puede ver el nombre del entorno de CloudShell si ya ha creado un entorno de VPC de CloudShell.

   > **Note**
   >
   > Si ya tiene un entorno de VPC de CloudShell, no puede crear uno nuevo.

   a. Seleccione Ejecutar.

      Este paso ejecutará el comando en el terminal de CloudShell en el entorno de VPC de CloudShell seleccionado.

> **Note**
>
> Si no tiene permiso para ver los entornos de VPC creados, póngase en contacto con su administrador para añadir el permiso de `cloudshell:describeEnvironments`. Para obtener más información, consulte Managing AWS CloudShell access and usage with IAM policies.

Puede seguir ejecutando comandos en el terminal de CloudShell.

Personalización de su experiencia AWS CloudShell

Puede personalizar los siguientes aspectos de su experiencia AWS CloudShell:

- 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.
- Amazon Q CLI: el uso de la CLI de Amazon Q le permite utilizar las características de esta CLI.

También puede ampliar su entorno de intérprete de comandos si instala su propio software y modifica el intérprete de comandos 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 terminal de AWS CloudShell, 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

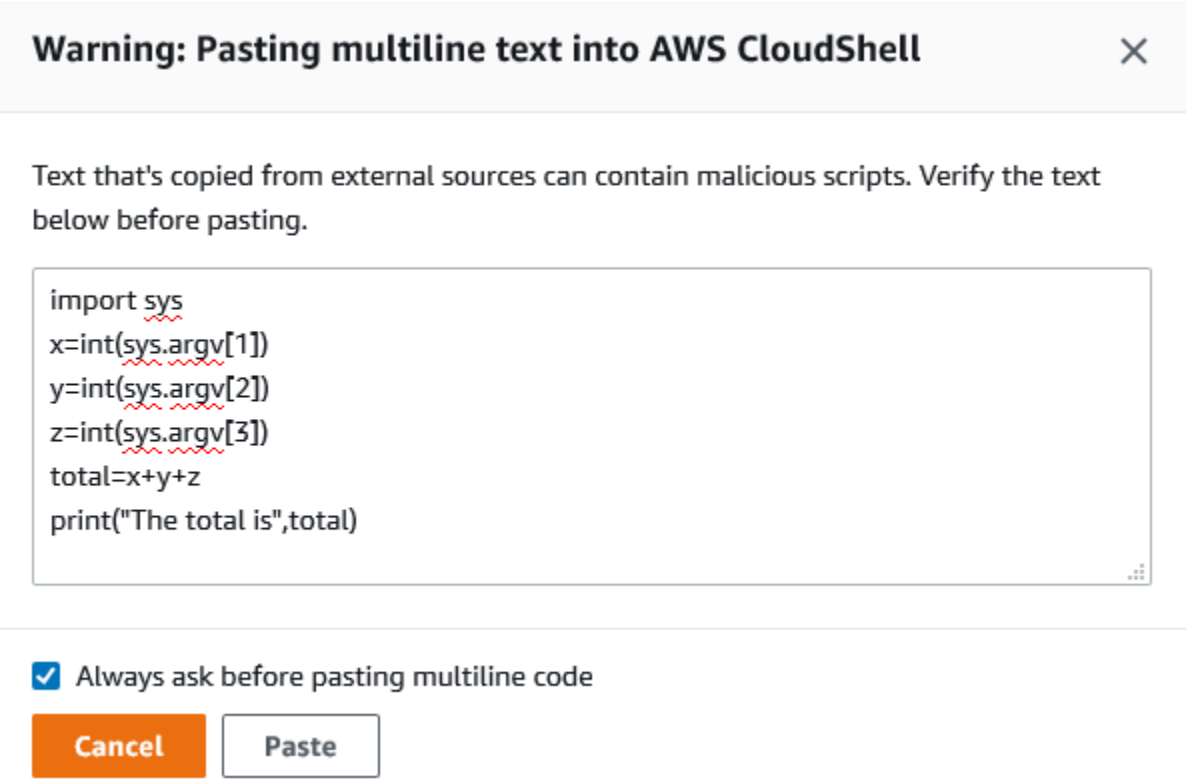
Cambie entre el tema claro y el oscuro en la interfaz de la línea de comandos.

1. Para cambiar el tema de AWS CloudShell, 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.

Uso de 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 de la CLI de Amazon Q

Puede activar o desactivar la CLI de Amazon Q seleccionando Preferencias, Activar la CLI de Amazon Q y Desactivar la CLI de Amazon Q. Para obtener más información, consulte Activación o desactivación de la CLI de Amazon Q.

Uso AWS CloudShell en Amazon VPC

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

Con Amazon VPC, 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 información sobre VPC, consulte Amazon Virtual Private Cloud.

Restricciones operativas

AWS CloudShell Los entornos de VPC tienen las siguientes restricciones:

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

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

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

> **Note**
>
> Las direcciones IP públicas no se asignan a los entornos de CloudShell VPC de forma predeterminada. Los entornos de VPC creados en subredes públicas con tablas de

enrutamiento configuradas para enrutar todo el tráfico a una puerta de enlace de Internet no tendrán acceso a la Internet pública, pero las subredes privadas configuradas con Traducción de direcciones de red (NAT) sí lo tienen. Los entornos de VPC creados en dichas subredes privadas tendrán acceso a la 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
  - Puntos de conexión de VPC
    - com.amazonaws.<región>.ssmmessages
    - com.amazonaws.<región>.logs
    - com.amazonaws.<región>.kms
    - com.amazonaws.<región>.execute-api
    - com.amazonaws.<región>.ecs-telemetry
    - com.amazonaws.<región>.ecs-agent
    - com.amazonaws.<región>.ecs
    - com.amazonaws.<región>.ecr.dkr
    - com.amazonaws.<región>.ecr.api
    - com.amazonaws.<región>.codecatalyst.packages
    - com.amazonaws.<región>.codecatalyst.git
    - aws.api.global.codecatalyst

  No puede restringir el acceso a estos puntos de conexión mediante la modificación de la configuración de la VPC.

  CloudShell La VPC está disponible en todas AWS las regiones y GovCloud regiones. Para ver una lista de las regiones en las que está disponible la CloudShell VPC, consulte AWS Regiones compatibles para. AWS CloudShell

Creación de un entorno CloudShell de VPC

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

Requisitos previos

El administrador debe proporcionarle los permisos de IAM necesarios para que pueda crear entornos de VPC. Para obtener más información sobre la habilitación de permisos para crear entornos de CloudShell VPC, consulte. the section called “Permisos de IAM necesarios para crear y usar entornos de CloudShell VPC”

Para crear un entorno de CloudShell VPC

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

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

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

Para obtener información sobre el uso de los entornos de VPC, consulte Introducción.

Permisos de IAM necesarios para crear y usar entornos de CloudShell VPC

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

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

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

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

Se recomienda incluir:

- ec2:DeleteNetworkInterface

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

Política de IAM que otorga CloudShell acceso total, incluido el acceso a la VPC

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

JSON

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

```
  "Effect": "Allow",
  "Action": [
   "cloudshell:*"
  ],
  "Resource": "*"
 },
 {
  "Sid": "AllowDescribeVPC",
  "Effect": "Allow",
  "Action": [
   "ec2:DescribeSubnets",
   "ec2:DescribeSecurityGroups",
   "ec2:DescribeVpcs"
  ],
  "Resource": "*"
 },
 {
  "Sid": "AllowInspectVPCConfigurationViaCloudShell",
  "Effect": "Allow",
  "Action": [
   "ec2:DescribeDhcpOptions",
   "ec2:DescribeNetworkInterfaces"
  ],
  "Resource": "*",
  "Condition": {
   "ForAnyValue:StringEquals": {
    "aws:CalledVia": "cloudshell.amazonaws.com"
   }
  }
 },
 {
  "Sid": "AllowCreateTagWithCloudShellKeyViaCloudShell",
  "Effect": "Allow",
  "Action": [
   "ec2:CreateTags"
  ],
  "Resource": "arn:aws:ec2:*:*:network-interface/*",
  "Condition": {
   "StringEquals": {
    "ec2:CreateAction": "CreateNetworkInterface"
   },
   "ForAnyValue:StringEquals": {
    "aws:TagKeys": "ManagedByCloudShell",
    "aws:CalledVia": "cloudshell.amazonaws.com"
```

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

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

Uso de claves de condición de IAM para entornos de VPC

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

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

- `CloudShell:VpcIds`— Permitir o denegar una o más VPCs
- `CloudShell:SubnetIds`: permiten o deniegan una o varias subredes.
- `CloudShell:SecurityGroupIds`: permiten o deniegan uno o varios 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 política de IAM con esta restricción e inhabilitar su acceso al entorno público existente, primero debe actualizar la política de IAM 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).

Políticas de ejemplo con claves de condición para la configuración de la VPC

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

Cómo garantizar que los usuarios creen solo entornos de VPC y denieguen la creación de entornos públicos

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

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

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

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. En el ejemplo siguiente, se deniega a los usuarios el acceso a `vpc-1` y `vpc-2`:

Para denegar a los usuarios el acceso a una condición específica VPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:SubnetIds` condición. En el ejemplo siguiente, se deniega a los usuarios el acceso a `subnet-1` y `subnet-2`:

Para denegar a los usuarios el acceso a una condición específica VPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:SecurityGroupIds` condición. En el ejemplo siguiente, se deniega a los usuarios el acceso a `sg-1` y `sg-2`:

JSON

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

Cómo permitir a los usuarios crear entornos con configuraciones de VPC específicas

Para permitir que los usuarios accedan a una condición específica VPCs, utilice `StringEquals` para comprobar el valor de la `cloudshell:VpcIds` condición. En el ejemplo siguiente, se permite a los usuarios el acceso a `vpc-1` y `vpc-2`:

Para permitir a los usuarios acceder a una condición específica VPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:SubnetIds` condición. En el ejemplo siguiente, se permite a los usuarios el acceso a `subnet-1` y `subnet-2`:

JSON

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

Para permitir a los usuarios acceder a una condición específica VPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:SecurityGroupIds` condición. En el ejemplo siguiente, se permite a los usuarios el acceso a `sg-1` y `sg-2`:

JSON

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

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. Eres responsable de mantener el control sobre el contenido alojado en esta infraestructura. También eres 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 las Preguntas frecuentes sobre la privacidad de datos. Para obtener información sobre la protección de datos en Europa, consulta la publicación de blog sobre el Modelo de responsabilidad compartida de AWS y RGPD en el Blog de seguridad de AWS .

Con fines de protección de 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:

- Utiliza la autenticación multifactor (MFA) en cada cuenta.
- Se utiliza SSL/TLS para comunicarse con AWS los recursos. Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Configure la API y el registro de actividad de los usuarios 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.
- Utiliza servicios de seguridad administrados avanzados, como Amazon Macie, que lo ayuden a detectar y proteger los datos confidenciales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados por FIPS 140-3 para acceder a AWS través de una interfaz de línea de comandos o una API, utilice un punto final FIPS. Para obtener más información sobre los puntos de conexión de FIPS disponibles, consulta Estándar de procesamiento de la información federal (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, la API o. AWS CLI AWS SDKs Cualquier dato que introduzca en etiquetas o campos

de formato libre utilizados para los nombres se pueden emplear para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para 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 finales de servicio AWS CloudShell y cuando viajan entre ellos.

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. Cuando lo usas 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 AWS KMS keys 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 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 enviándolo todo a través de una HTTPS/TLS conexión.

No necesita hacer nada para habilitar el uso de HTTPS/TLS para la comunicación.

Identity and Access Management para AWS CloudShell

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

Temas

Público

La forma de usar AWS Identity and Access Management (IAM) varía según la función que desempeñes:

- Usuario del servicio: solicita permisos a su administrador si no puede acceder a las características (consulte Solución de problemas de CloudShell identidad y acceso a AWS).
- Administrador del servicio: determina el acceso de los usuarios y envía solicitudes de permiso (consulte Cómo CloudShell funciona AWS con IAM).
- Administrador de IAM: escribe políticas para administrar el acceso (consulte Ejemplos de políticas basadas en identidad para AWS CloudShell).

Autenticación con identidades

La autenticación es la forma en que inicias sesión AWS con tus credenciales de identidad. Debe autenticarse como usuario de Usuario raíz de la cuenta de AWS IAM o asumir una función de IAM.

Puede iniciar sesión como una identidad federada con las credenciales de una fuente de identidad, como AWS IAM Identity Center (IAM Identity Center), la autenticación de inicio de sesión único o las

credenciales. Google/Facebook Para obtener más información sobre el inicio de sesión, consulte Cómo iniciar sesión en su Cuenta de AWS en la Guía del usuario de AWS Sign-In .

Para el acceso programático, AWS proporciona un SDK y una CLI para firmar criptográficamente las solicitudes. Para obtener más información, consulte AWS Signature Version 4 para solicitudes de API en la Guía del usuario de IAM.

Cuenta de AWS usuario root

Al crear un Cuenta de AWS, se comienza con una identidad de inicio de sesión denominada usuario Cuenta de AWS raíz que tiene acceso completo a todos Servicios de AWS los recursos. Recomendamos encarecidamente que no utilice el usuario raíz para sus tareas diarias. Para ver las tareas que requieren credenciales de usuario raíz, consulte Tareas que requieren credenciales de usuario raíz en la Guía del usuario de IAM.

Identidad federada

Como práctica recomendada, exija a los usuarios humanos 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 empresarial, del proveedor de identidades web o al Directory Service que se accede Servicios de AWS mediante credenciales de una fuente de identidad. Las identidades federadas asumen roles que proporcionan credenciales temporales.

Para una administración de acceso centralizada, recomendamos AWS IAM Identity Center. Para obtener más información, consulte ¿Qué es el Centro de identidades de IAM? en la Guía del usuario de AWS IAM Identity Center .

Usuarios y grupos de IAM

Un usuario de IAM es una identidad que dispone de permisos específicos para una sola persona o aplicación. Recomendamos usar credenciales temporales en lugar de usuarios de IAM con credenciales a largo plazo. Para obtener más información, consulte Exigir a los usuarios humanos que utilicen la federación con un proveedor de identidad para acceder AWS mediante credenciales temporales en la Guía del usuario de IAM.

Un grupo de IAM especifica una colección de usuarios de IAM y facilita la administración de permisos para grandes conjuntos de usuarios. Para obtener más información, consulte Casos de usos para usuarios de IAM en la Guía del usuario de IAM.

Roles de IAM

Un rol de IAM es una identidad con permisos específicos que proporciona credenciales temporales. Puede asumir un rol cambiando de un rol de usuario a uno de IAM (consola) o llamando a una AWS CLI operación de AWS API. Para obtener más información, consulte Métodos para asumir un rol en la Guía del usuario de IAM.

Las funciones de IAM son útiles para el acceso de usuarios federados, los permisos de usuario de IAM temporales, el acceso entre cuentas, el acceso entre servicios y las aplicaciones que se ejecutan en Amazon. EC2 Para obtener más información, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Administración del acceso con políticas

El acceso se controla creando políticas y AWS adjuntándolas a identidades o recursos. AWS Una política define los permisos cuando están asociados a una identidad o un recurso. AWS evalúa estas políticas cuando un director hace una solicitud. La mayoría de las políticas se almacenan AWS como documentos JSON. Para obtener más información sobre los documentos de política JSON, consulte Información general de políticas de JSON en la Guía del usuario de IAM.

Mediante las políticas, los administradores especifican quién tiene acceso a qué, definiendo qué entidad principal puede realizar acciones sobre qué recursos y en qué condiciones.

De forma predeterminada, los usuarios y los roles no tienen permisos. Un administrador de IAM crea políticas de IAM y las agrega a roles, que los usuarios pueden asumir posteriormente. Las políticas de IAM definen permisos independientemente del método que se utilice para realizar la operación.

Políticas basadas en identidades

Las políticas basadas en identidad son documentos de política de permisos JSON que asocia a una identidad (usuario, grupo o rol). Estas políticas controlan qué acciones pueden realizar las identidades, en qué recursos y en qué condiciones. Para obtener más información sobre cómo crear una política basada en la identidad, consulte Definición de permisos de IAM personalizados con políticas administradas por el cliente en la Guía del usuario de IAM.

Las políticas basadas en identidad pueden ser políticas insertadas (incrustadas directamente en una sola identidad) o políticas administradas (políticas independientes asociadas a varias identidades). Para obtener más información sobre cómo elegir una política administrada o una política insertada, consulte Elegir entre políticas administradas y políticas insertadas en la Guía del usuario de IAM.

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de políticas JSON que se asocian a un recurso. Los ejemplos incluyen políticas de confianza de roles de IAM y 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. Debe especificar una entidad principal en una política basada en recursos.

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No puedes usar políticas AWS gestionadas de IAM en una política basada en recursos.

Otros tipos de políticas

AWS admite tipos de políticas adicionales que pueden establecer los permisos máximos que conceden los tipos de políticas más comunes:

- Límites de permisos: establecen los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM. Para obtener más información, consulte Límites de permisos para las entidades de IAM en la Guía del usuario de IAM.
- Políticas de control de servicios (SCPs): especifican los permisos máximos para una organización o unidad organizativa en AWS Organizations. Para obtener más información, consulte Políticas de control de servicios en la Guía del usuario de AWS Organizations .
- Políticas de control de recursos (RCPs): establece los permisos máximos disponibles para los recursos de tus cuentas. Para obtener más información, consulte Políticas de control de recursos (RCPs) en la Guía del AWS Organizations usuario.
- Políticas de sesión: políticas avanzadas que se transfieren como parámetro cuando se crea una sesión temporal para un rol o un usuario federado. Para obtener más información, consulte Políticas de sesión en la Guía del usuario de IAM.

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 usuario de IAM.

Cómo CloudShell funciona AWS con IAM

Antes de utilizar IAM para gestionar el acceso CloudShell, infórmese sobre las funciones de IAM disponibles para su uso. CloudShell

Características de IAM que puede usar con AWS CloudShell

Característica de IAM	CloudShell soporte
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 políticas)	No
Credenciales temporales	Sí
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 funciones de IAM, consulte AWS los servicios que funcionan con IAM en la Guía del usuario de IAM.

Políticas basadas en la identidad para CloudShell

Compatibilidad con las políticas basadas en identidad: sí

Las políticas basadas en identidad son documentos de políticas de permisos JSON que puede asociar a una identidad, como un usuario de IAM, 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 más información sobre cómo crear una política basada en la identidad, consulte Definición de permisos de IAM personalizados con políticas administradas por el cliente en la Guía del usuario de IAM.

Con las políticas basadas en identidades de IAM, puede especificar las acciones y los recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. Para obtener más información sobre los elementos que puede utilizar en una política de JSON, consulte Referencia de los elementos de la política de JSON de IAM en la Guía del usuario de IAM.

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 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 política JSON que se asocian a un recurso. Los ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM 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 basada en recursos. Los principales pueden incluir cuentas, usuarios, roles, usuarios federados o. Servicios de AWS

Para habilitar el acceso entre cuentas, puede especificar toda una cuenta o entidades de IAM de otra cuenta como la entidad principal de una política en función de recursos. Para obtener más información, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Acciones políticas para CloudShell

Compatibilidad con las acciones de políticas: sí

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

El elemento `Action` de una política JSON describe las acciones que puede utilizar para conceder o denegar el acceso en una política. Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

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

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 identidad para AWS CloudShell

Recursos de políticas para CloudShell

Compatibilidad con los recursos de políticas: sí

Los administradores pueden usar las políticas de AWS JSON 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` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Como práctica recomendada, especifique un recurso utilizando el Nombre de recurso de Amazon (ARN). En el caso de las acciones que no admiten permisos por recurso, 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 tipos ARNs, consulte Recursos definidos por AWS CloudShell en la Referencia de autorización de servicios. Para saber con qué acciones puede especificar el ARN de cada recurso, consulte Acciones definidas por AWS. CloudShell

Para ver ejemplos de políticas CloudShell basadas en la identidad, consulte. Ejemplos de políticas basadas en 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 las políticas de AWS JSON 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` especifica cuándo se ejecutan las instrucciones en función de criterios definidos. Puede crear expresiones condicionales que utilizan operadores de condición, tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud. Para ver todas las claves de condición AWS globales, consulte las claves de contexto de condición AWS globales en la Guía del usuario de IAM.

Para ver una lista de claves de CloudShell condición, consulte Claves de condición de AWS CloudShell en la Referencia de autorización de servicios. 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 identidad para AWS CloudShell

ACLs in CloudShell

Soporta ACLs: 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. ACLs son similares a las políticas basadas en recursos, aunque no utilizan el formato de documento de políticas JSON.

ABAC con CloudShell

Compatibilidad con ABAC (etiquetas en las políticas): no

El control de acceso basado en atributos (ABAC) es una estrategia de autorización que define permisos en función de atributos denominados etiquetas. Puede adjuntar etiquetas a las entidades y AWS los recursos de IAM y, a continuación, diseñar políticas de ABAC para permitir las operaciones cuando la etiqueta del principal coincida con la etiqueta del recurso.

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 sobre ABAC, consulte Definición de permisos con la autorización de ABAC en la Guía del usuario de IAM. Para ver un tutorial con los pasos para configurar ABAC, consulte Uso del control de acceso basado en atributos (ABAC) en la Guía del usuario de IAM.

Utilizar credenciales temporales con CloudShell

Compatibilidad con credenciales temporales: sí

Las credenciales temporales proporcionan acceso a AWS los recursos a corto plazo y se crean automáticamente cuando se utiliza la federación o se cambia de rol. 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 y Servicios de AWS que funcionan con IAM en la Guía del usuario de IAM.

Al cambiar de rol, utilizará un entorno diferente. No puede cambiar de rol en el mismo AWS CloudShell entorno.

Sesiones de acceso directo para CloudShell

Compatibilidad con las sesiones de acceso directo (FAS): no

Las sesiones de acceso directo (FAS) utilizan los permisos de la persona principal que llama Servicio de AWS, junto con la solicitud, Servicio de AWS para realizar solicitudes a los servicios descendentes. Para obtener información detallada sobre las políticas a la hora de realizar solicitudes de FAS, consulte Forward access sessions.

Roles de servicio para CloudShell

Compatible con roles de servicio: No

Un rol de servicio es un rol de IAM que asume un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte Crear un rol para delegar permisos a un Servicio de AWS en la Guía del usuario de IAM.

> **Warning**
>
> Cambiar los permisos de una función de servicio podría afectar a la CloudShell 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 administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

Ejemplos de políticas basadas en identidad para AWS CloudShell

De forma predeterminada, los usuarios y roles no tienen permiso para crear, ver ni modificar recursos de CloudShell. Un administrador de IAM puede crear políticas de IAM para conceder permisos a los usuarios para realizar acciones en los recursos que necesitan.

Para obtener información acerca de cómo crear una política basada en identidades de IAM mediante el uso de estos documentos de políticas JSON de ejemplo, consulte Creación de políticas de IAM (consola) en la Guía del usuario de IAM.

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

Temas

- Prácticas recomendadas sobre las políticas
- Mediante la consola de CloudShell
- 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 su 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. Con el fin de obtener más información, consulte las políticas administradas por AWS o las políticas administradas por AWS para funciones de tarea en la Guía de usuario de IAM.
- Aplique permisos de privilegio mínimo: cuando establezca permisos con políticas de IAM, 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. Con el fin de obtener más información sobre el uso de IAM para aplicar permisos, consulte Políticas y permisos en IAM en la Guía del usuario de IAM.
- Utilice condiciones en las políticas de IAM para restringir aún más el acceso: puede agregar una condición a sus políticas para limitar el acceso a las acciones y los recursos. Por ejemplo, puede escribir una condición de políticas para especificar que todas las solicitudes deben enviarse utilizando SSL. 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 CloudFormation. Para obtener más información, consulte Elementos de la política de JSON de IAM: Condición en la Guía del usuario de IAM.
- Utiliza el analizador de acceso de IAM para validar las políticas de IAM con el fin de garantizar la seguridad y funcionalidad de los permisos: el analizador de acceso de IAM valida políticas nuevas y existentes para que respeten el lenguaje (JSON) de las políticas de IAM y las prácticas recomendadas de IAM. El analizador de acceso de IAM proporciona más de 100 verificaciones de políticas y recomendaciones procesables para ayudar a crear políticas seguras y funcionales. Para más información, consulte Validación de políticas con el Analizador de acceso de IAM en la Guía del usuario de IAM.
- Requerir autenticación multifactor (MFA): si tiene un escenario que requiere usuarios de IAM o un usuario raíz en Cuenta de AWS su cuenta, active la MFA para mayor seguridad. Para exigir la MFA cuando se invoquen las operaciones de la API, añada condiciones de MFA a sus políticas. Para más información, consulte Acceso seguro a la API con MFA en la Guía del usuario de IAM.

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

Mediante la consola de CloudShell

Para acceder a la CloudShell consola de AWS, 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 solo realicen llamadas a la API AWS CLI o a la AWS API. En su lugar, permita el acceso únicamente a las acciones que coincidan con la operación de API que intentan realizar.

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

Cómo permitir a los usuarios consultar sus propios permisos

En este ejemplo, se muestra cómo podría crear una política que permita a los usuarios de IAM ver las políticas administradas e insertadas que se asocian a la identidad de sus usuarios. Esta política incluye permisos para completar esta acción en la consola o mediante programación mediante la API AWS CLI o AWS .

```
{
    "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 CloudShell identidad y acceso a AWS

Utilice la siguiente información como ayuda para diagnosticar y solucionar los problemas habituales que pueden surgir al trabajar con un CloudShell IAM.

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.

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

```
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 su AWS administrador. El administrador es la persona que le proporcionó las credenciales de inicio de sesión.

No estoy autorizado a realizar tareas como: PassRole

Si recibe un error que indica que no tiene autorización para realizar la acción `iam:PassRole`, las políticas deben actualizarse a fin de permitirle pasar un rol a CloudShell.

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 a un servicio. Para ello, debe tener permisos para transferir la función al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado `marymajor` intenta utilizar la consola para realizar una acción en 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 su AWS 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

Se puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Se 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 obtener más información, consulte lo siguiente:

- Para saber si CloudShell es compatible con estas funciones, consulte. Cómo CloudShell funciona AWS con IAM

- Para obtener información sobre cómo proporcionar acceso a los recursos de su Cuentas de AWS propiedad, consulte Proporcionar acceso a un usuario de IAM en otro usuario de su propiedad Cuenta de AWS en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso a tus recursos a terceros Cuentas de AWS, consulta Cómo proporcionar acceso a recursos que Cuentas de AWS son propiedad de terceros en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una federación de identidades, consulte Proporcionar acceso a usuarios autenticados externamente (federación de identidades) en la Guía del usuario de IAM.
- Para conocer sobre la diferencia entre las políticas basadas en roles y en recursos para el acceso entre cuentas, consulte Acceso a recursos entre cuentas en IAM en la Guía del usuario de IAM.

Administrar el AWS CloudShell acceso y el uso con las políticas de IAM

Con los recursos de administración de acceso que pueden proporcionar AWS Identity and Access Management, los administradores pueden conceder permisos a los usuarios de IAM. 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 administrada de AWS es una política independiente creada y administrada por AWS. La siguiente política AWS gestionada para se AWS CloudShell puede adjuntar a las identidades de IAM:

- AWS CloudShellFullAccess: concede permiso para usar AWS CloudShell con acceso completo a todas las características.

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

> **Note**
>
> También se pueden lanzar CloudShell identidades de IAM con las siguientes políticas AWS gestionadas. Sin embargo, estas políticas ofrecen amplios permisos. Por lo tanto, le

recomendamos que solo conceda estas políticas si son esenciales para el puesto de trabajo de un usuario de IAM.

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

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

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

Para gestionar las acciones con las que puede realizar un usuario de IAM CloudShell, cree una política personalizada que utilice la política CloudShellPolicy gestionada como plantilla. Alternativamente, edite una política en línea que esté incrustada en la identidad IAM relevante (usuario, grupo o rol).

Por ejemplo, puede permitir el acceso de los usuarios de IAM CloudShell, pero evitar que envíen las credenciales del CloudShell entorno que se utilizan para iniciar sesión. Consola de administración de AWS

Important

Para iniciar AWS CloudShell desde Consola de administración de AWS, un usuario de IAM 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 mostrará un error de permisos de IAM al intentar lanzarla. CloudShell

AWS CloudShell permisos

Name	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 espera * como valor para `Resource`, tal y como se describe en the section called “Ejemplos de políticas de IAM para CloudShell”.	Sí
`cloudshell:CreateSession`	Se conecta a un CloudShell entorno desde. Consola de administración de AWS	Sí
`cloudshell:GetEnvironmentStatus`	Lea el estado de un CloudShell entorno.	Sí
`cloudshell:DeleteEnvironment`	Elimina un CloudShell entorno.	No

Name	Descripción del permiso concedido	¿Necesario para el lanzamiento CloudShell?
`cloudshell:GetFileDownloadUrls`	Genera Amazon S3 prefirmado URLs que se utiliza para descargar archivos CloudShel l mediante la interfaz CloudShell web. Esto opción no está disponible para los entornos de VPC.	No
`cloudshell:GetFileUploadUrls`	Genera Amazon S3 prefirmado URLs que se utiliza para cargar archivos CloudShel l mediante la interfaz CloudShell web. Esto opción no está disponible para los entornos de VPC.	No
`cloudshell:DescribeEnvironments`	Describe los entornos.	No
`cloudshell:PutCredentials`	Reenvía las credencia les utilizadas para iniciar sesión en la Consola de administración de AWS carpeta. 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

Name	Descripción del permiso concedido	¿Necesario para el lanzamiento CloudShell?
`cloudshell:ApproveCommand`	Aprueba un comando enviado CloudShell desde otras consolas de AWS servicio.	No

Ejemplos de políticas de IAM 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.

JSON

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

La siguiente política permite a los usuarios de IAM 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`.

JSON

```
{
    "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 el acceso de los usuarios de IAM. CloudShell Sin embargo, la política impide que las credenciales que utilizó para iniciar sesión se reenvíen al CloudShell entorno. Consola de administración de AWS Los usuarios de IAM con esta política deben configurar manualmente sus credenciales en CloudShell ella.

JSON

```
{
    "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 los usuarios de IAM crear AWS CloudShell entornos.

JSON

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

Permisos de IAM necesarios para crear y usar entornos de CloudShell VPC

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

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

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

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

También se recomienda incluir:

- ec2:DeleteNetworkInterface

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

Política de IAM que otorga CloudShell acceso total, incluido el acceso a la VPC

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

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudShellOperations",
      "Effect": "Allow",
      "Action": [
```

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

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

Uso de claves de condición de IAM para entornos de VPC

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

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

- `CloudShell:VpcIds`— Permitir o denegar una o más VPCs
- `CloudShell:SubnetIds`: permiten o deniegan una o varias subredes.
- `CloudShell:SecurityGroupIds`: permiten o deniegan uno o varios 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 política de IAM con esta restricción e inhabilitar su acceso al entorno público existente, primero debe actualizar la política de IAM 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).

Políticas de ejemplo con claves de condición para la configuración de la VPC

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

Cómo garantizar que los usuarios creen solo entornos de VPC y denieguen la creación de entornos públicos

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

```
{
  "Statement": [
    {
      "Sid": "DenyCloudShellNonVpcEnvironments",
```

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

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

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. En el ejemplo siguiente, se deniega a los usuarios el acceso a `vpc-1` y `vpc-2`:

Para denegar a los usuarios el acceso a una condición específica VPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:SubnetIds` condición. En el ejemplo siguiente, se deniega a los usuarios el acceso a `subnet-1` y `subnet-2`:

Para denegar a los usuarios el acceso a una condición específica VPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:SecurityGroupIds` condición. En el ejemplo siguiente, se deniega a los usuarios el acceso a `sg-1` y `sg-2`:

JSON

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

```
          "cloudshell:SecurityGroupIds": [
            "sg-1",
            "sg-2"
          ]
        }
      }
    }
  ]
}
```

Cómo permitir a los usuarios crear entornos con configuraciones de VPC específicas

Para permitir que los usuarios accedan a una condición específica VPCs, utilice `StringEquals` para comprobar el valor de la `cloudshell:VpcIds` condición. En el ejemplo siguiente, se permite a los usuarios el acceso a `vpc-1` y `vpc-2`:

Para permitir a los usuarios acceder a una condición específica VPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:SubnetIds` condición. En el ejemplo siguiente, se permite a los usuarios el acceso a `subnet-1` y `subnet-2`:

JSON

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

```
  ]
}
```

Para permitir a los usuarios acceder a una condición específica VPCs, utilice esta opción `StringEquals` para comprobar el valor de la `cloudshell:SecurityGroupIds` condición. En el ejemplo siguiente, se permite a los usuarios el acceso a `sg-1` y `sg-2`:

JSON

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

Permisos de acceso Servicios de AWS

CloudShell utiliza las credenciales de IAM que utilizó para iniciar sesión en. Consola de administración de AWS

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

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

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

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

Permisos para acceder a las funciones de la CLI de Amazon Q en CloudShell

Para utilizar las funciones de la CLI de Amazon Q CloudShell, como las sugerencias integradas, el chat y la traducción, asegúrese de tener los permisos de IAM necesarios. Si no puede acceder a las funciones de la CLI de Amazon Q CloudShell, póngase en contacto con su administrador para que le proporcione los permisos de IAM necesarios. Para obtener más información, consulte Ejemplos de políticas basadas en identidad para Amazon Q Developer en la Guía del usuario de Amazon Q Developer.

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 las llamadas a la API AWS CloudShell como eventos. Las llamadas

capturadas incluyen llamadas desde la AWS CloudShell consola y 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 configura un registro de seguimiento, puede ver los eventos más recientes en la consola de CloudTrail en el Event history (Historial de eventos). Con la información recopilada por CloudTrail, puede descubrir una variedad de información sobre una solicitud. Por ejemplo, puede determinar la solicitud que se realizó a AWS CloudShell, puede conocer la dirección IP desde la que se realizó la solicitud, quién la hizo y cuándo se hizo.

AWS CloudShell in CloudTrail

En la siguiente tabla se enumeran 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 a la API que no muta (solo lectura).
> - La palabra `Environment` se refiere al ciclo de vida del entorno de computación que aloja la experiencia del intérprete de comandos.
> - La palabra `Layout` restaura todas las pestañas del navegador en el CloudShell terminal.

CloudShell Eventos en CloudTrail

Nombre de evento	Description (Descripción)
`createEnvironment`	Se produce cuando se crea un CloudShell entorno.
`createSession`	Se produce cuando un CloudShell entorno se conecta desde Consola de administración de AWS.

Nombre de evento	Description (Descripción)
`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 Consola de administración de AWS .
`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.

Nombre de evento	Description (Descripción)
`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 signo 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 de sistemas y organizaciones (SOC) son informes de análisis independientes de terceros que demuestran cómo se AWS logran los principales controles y objetivos de cumplimiento.

Servicio	SDK	SOC 1,2,3
AWS CloudShell	CloudShell	✓

PCI

El estándar de seguridad de datos del sector de las tarjetas de pago (PCI DSS) es un estándar de seguridad de la información patentado administrado por el Consejo de Normas de Seguridad de la PCI, fundado por American Express, Discover Financial Services, JCB International, MasterCard Worldwide y Visa Inc.

Servicio	SDK	PCI
AWS CloudShell	CloudShell	✓

Certificaciones y servicios ISO y CSA STAR

AWS cuenta con la certificación ISO/IEC 27001:2013, 27017:2015, 27018:2019, 27701:2019, 22301:2019, 9001:2015y CSA STAR CCM v4.0.

Servicio	SDK	Certificaciones y servicios ISO y CSA STAR
AWS CloudShell	CloudShell	✓

FedRamp

El Programa Federal de Administración de Riesgos y Autorizaciones (FedRAMP) es un amplio programa gubernamental de EE. UU. que ofrece un enfoque estandarizado para la supervisión continua, la autorización y la evaluación de la seguridad de servicios y productos en la nube.

Servicio	SDK	FedRAMP Moderate (East/West)	FedRamp High () GovCloud
AWS CloudShell	CloudShell	✓	✓

DoD CC SRG

La Guía de requisitos de seguridad de la computación en la nube (SRG) del Departamento de Defensa (DoD) 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 evaluación y autorización de DoD CC SRG tendrán el siguiente estado:

- Evaluación de una organización de terceros (3PAO): el asesor de terceros está evaluando actualmente este servicio.
- Revisión de la Junta de Autorización Conjunta (JAB): este servicio se está sometiendo a una revisión de la JAB.
- Revisión de la Agencia de Sistemas de Información de Defensa (DISA): este servicio se encuentra actualmente en proceso de revisión de DISA.

Servicio	SDK	DoD CC SRG IL2 (Este/Oeste)	DoD CC IL2 SRG () GovCloud	DoD CC IL4 SRG () GovCloud	DoD CC IL5 SRG () GovCloud	DoD CC SRG IL6 (Región secreta)AWS
AWS CloudShell	CloudShell	✓	✓	✓	✓	N/A

HIPAA BAA

La Ley de Portabilidad y Responsabilidad de Seguros Médicos de 1996 (HIPAA) es una ley federal que exige la creación de estándares nacionales para proteger la información médica confidencial del paciente para evitar que se divulgue sin el consentimiento o el conocimiento del paciente.

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

Servicio	SDK	HIPAA BAA
AWS CloudShell	CloudShell	✓

IRAP

El Programa de Asesores Registrados de Seguridad de la Información (IRAP) permite a los clientes del gobierno australiano validar que existen controles apropiados y determinar el modelo de responsabilidad adecuado para cumplir los requisitos del Manual de Seguridad de la Información (ISM) del gobierno australiano producido por el Centro Australiano de Ciberseguridad (ACSC).

Servicio	Espacio de nombres*	Protección IRAP
AWS CloudShell	N/A	✓

*Los espacios de nombres le ayudan a identificar los servicios en todo su entorno. AWS Por ejemplo, al crear políticas de IAM, 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 (SPRING SS 584), basado en los estándares del Sistema de Gestión de la Seguridad de la Información (ISMS) ISO 27001/02.

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 conformidad de computación en la nube (C5) es un esquema de certificación respaldado por el gobierno alemán presentado en Alemania por la Oficina Federal de Seguridad de la Información (BSI) para ayudar a las organizaciones a demostrar la seguridad operativa frente a ciberataques comunes al utilizar servicios en la nube en el contexto de las "Recomendaciones de seguridad para proveedores de nube" del gobierno alemán.

Servicio	SDK	C5
AWS CloudShell	CloudShell	✓

ENS High

El sistema 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	ENS High
AWS CloudShell	CloudShell	✓

FINMA

La Autoridad Suiza de Supervisión de los Mercados Financieros (FINMA) es el regulador independiente de los mercados financieros de Suiza. La alineación de AWS con los requisitos de la FINMA demuestra nuestro compromiso continuo con el cumplimiento de las mayores expectativas para los proveedores de servicios en la nube establecidas por los reguladores de servicios financieros suizos y los clientes.

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 los programas de conformidad específicos, consulte Servicios de AWS incluidos . 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 cómo diseñar una arquitectura basada en la seguridad y el cumplimiento de la HIPAA: este documento técnico describe cómo pueden utilizar las empresas para crear aplicaciones que cumplan con la HIPAA. AWS
- AWS Recursos de cumplimiento Recursos de 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 CSPM— 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 es compatible con las siguientes funciones para satisfacer sus necesidades de respaldo y resiliencia de datos:

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

- Seguridad de la capa de transporte (TLS). Exigimos TLS 1.2 y recomendamos TLS 1.3.
- Conjuntos de cifrado con confidencialidad directa total (PFS) como DHE (Ephemeral Diffie-Hellman) o ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La mayoría de los sistemas modernos como Java 7 y posteriores son compatibles con estos modos.

> **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. Es posible que estas prácticas recomendadas no sean adecuadas o suficientes para su entorno, por lo que se recomienda abordarlas como consideraciones útiles en vez de normas.

Algunas prácticas recomendadas de seguridad para AWS CloudShell

- Utilice los permisos y las políticas de IAM 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 las políticas de IAM.
- No incluya datos confidenciales en sus entidades de IAM, como los usuarios, las funciones o los nombres de las sesiones.
- 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 varias líneas, consulte Uso 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 de computación 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 intérprete de comandos predeterminado, consulte Modificar el intérprete de comandos con scripts.
- Almacene el código de forma segura en un sistema de control de versiones.

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?

Al iniciar sesión Consola de administración de AWS, debe introducir sus credenciales de usuario de IAM. Además, cuando se inicia CloudShell desde la interfaz de la consola, estas credenciales se utilizan en las llamadas a la 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 quiere restringir el acceso a la red, debe habilitar el permiso para crear solo entornos de VPC y denegar la creación de entornos públicos.

Para obtener más información, consulte Cómo garantizar que los usuarios creen solo entornos de VPC y denieguen la creación de entornos públicos.

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

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?

Para entornos públicos, Amazon S3 proporciona la infraestructura para almacenar los datos en su `$HOME`.

En los entornos de VPC, el directorio `$HOME` se elimina cuando se agota el tiempo de espera del entorno de VPC (después de unos 20-30 minutos de inactividad), o bien cuando elimina o reinicia su entorno.

Volver a la lista de seguridad FAQs

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

No, no es posible cifrar el directorio `$HOME` con su clave propia. Sin embargo, CloudShell cifra el contenido del `$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 de VPC CloudShell . El directorio `$HOME` de un entorno de VPC se elimina cuando se agota el tiempo de espera del entorno de VPC (después de unos 20-30 minutos de inactividad), o bien cuando elimina o reinicia su entorno. En el menú Acciones, las opciones de carga y descarga no están disponibles para los entornos de VPC.

Volver a la lista de seguridad FAQs

Entorno de computación de AWS CloudShell: especificaciones y software

Al iniciar AWS CloudShell, se crea un entorno de computación basado en Amazon Linux 2023 para hospedar la experiencia del intérprete de comandos. El entorno está configurado con recursos de computación (vCPU y memoria) y ofrece una amplia gama de software preinstalado al que se puede acceder desde la interfaz de la línea de comandos. Asegúrese de que todo software que instale en el entorno de computación tenga aplicados los parches correspondientes y esté 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 de computación de AWS CloudShell se le asignan los siguientes recursos de CPU y memoria:

- 1 vCPU (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.

Requisitos de red de CloudShell

WebSockets

CloudShell depende del protocolo WebSocket, que permite la comunicación interactiva bidireccional entre el navegador web del usuario y el servicio CloudShell en la nube de AWS. Si utiliza un navegador en una red privada, es probable que los servidores proxy y los firewalls faciliten el acceso seguro a Internet. La comunicación de WebSocket normalmente puede atravesar los servidores proxy sin problemas. Sin embargo, en algunos casos, los servidores proxy impiden que los WebSockets funcionen correctamente. Si se produce este problema, su interfaz de CloudShell

informa del siguiente error: `Failed to open sessions : Timed out while opening the session`.

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

> **Note**
>
> Si desea definir permisos detallados permitiendo URL específicas, puede añadir parte de la URL que utilice la sesión de AWS Systems Manager para abrir una conexión WebSocket para enviar entradas y recibir salidas. (Sus comandos de AWS CloudShell se envían a esa sesión de Systems Manager).
> El formato de esta StreamUrl utilizada por 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 compatible con AWS Systems Manager, como `us-east-2` para 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 especificar `wss://ssmmessages.region.amazonaws.com` cuando actualice su lista de direcciones URL permitidas. Para obtener más información, consulte la operación StartSession en la Referencia API de AWS Systems Manager.

Software preinstalado

> **Note**
>
> Como el entorno de desarrollo AWS CloudShell se actualiza periódicamente para otorgar acceso al software más reciente, no incluimos 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 intérprete de comandos Bash es la aplicación de intérprete de comandos predeterminada para AWS CloudShell.	`bash --version`
PowerShell (pwsh)	PowerShell, que ofrece una interfaz de la línea de comandos y compatibilidad con lenguajes de scripting , se basa en el entorno de ejecución del lenguaje de comandos .NET de Microsoft. PowerShell utiliza comandos ligeros denominados “`cmdlets`“ que aceptan y devuelven objetos .NET.	`pwsh --version`
Z Shell (zsh)	Z Shell, también conocido como `zsh`, es una versión ampliada de Bourne Shell que ofrece un soporte de personalización mejorado para temas y complementos.	`zsh --version`

Interfaz de la línea de comandos (CLI) de AWS

CLI

Nombre	Descripción	Version information
Kit de herramientas de AWS CDK: la CLI	El kit de herramientas AWS CDK, el comando de la CLI, `cdk`, es la herramienta principal que interactúa con su aplicación de AWS CDK. Ejecuta su aplicación, consulta el modelo de aplicación que ha definido y produce e implementa las plantillas de AWS CloudFormation generadas por el AWS CDK. Para obtener más información, consulte Kit de herramientas AWS CDK.	`cdk --version`
AWS CLI	La AWS CLI es una interfaz de la línea de comandos para administrar múltiples servicios de AWS desde la línea de comandos y automatizarlos mediante scripts. Para obtener más información, consulte Administre AWS los servicios desde CLI en CloudShell. Para obtener información sobre cómo puede asegurarse de que está utilizando la versión de la AWS CLI 2 más actualizada, consulte	`aws --version`

Nombre	Descripción	Version information
	Instalación de la AWS CLI en el directorio de inicio.	
CLI DE EB	La CLI de AWS Elastic Beanstalk es una interfaz de la línea de comandos de que simplifican la creación, actualización y monitoreo de entornos desde un repositorio local. Para obtener más informaci ón sobre la EB CLI, consulte Uso de Elastic Beanstalk Command Line Interface (EB CLI) en la Guía para desarrolladores de AWS Elastic Beanstalk.	`eb --version`
CLI de Amazon ECS	La interfaz de la línea de comandos (CLI) de Amazon Elastic Container Service (Amazon ECS) proporciona comandos de alto nivel que simplifican la creación, la actualización y el monitoreo de clústeres y tareas. Para obtener más informaci ón, consulte la Usar la interfaz de la línea de comandos de Amazon ECS en la Guía para desarrolladores de Amazon Elastic Container Service.	`ecs-cli --version`

Nombre	Descripción	Version information
AWS SAM CLI	La CLI de AWS SAM es una herramienta de la línea de comandos que opera sobre una plantilla de AWS Serverless Application Model y código de la aplicación. Puede realizar varias tareas. Estos incluyen la invocación de funciones de Lambda localmente, la creación de un paquete de implementación para su aplicación sin servidor y la implementación de su aplicación sin servidor en la nube de AWS. Para obtener más información, consulte la referencia de la CLI de AWS SAM en la Guía para desarrolladores de AWS Serverless Application Model.	`sam --version`

Nombre	Descripción	Version information
Herramientas de AWS para PowerShell	Herramientas de AWS para PowerShell son módulos de PowerShell basados en la funcionalidad expuesta por SDK para .NET. Con Herramientas de AWS para PowerShell, se permite realizar operaciones mediante scripts en sus recursos de AWS desde la línea de comandos de PowerShell. AWS CloudShell preinstala la versión modularizada (AWS.Tools) de Herramientas de AWS para PowerShell. Para obtener más información, consulte Uso de las herramientas de AWS para PowerShell en la Guía del usuario de Herramientas de AWS para PowerShell.	`pwsh --Command 'Get-AWSPowerShellVersion'`

Entornos de ejecución y SDK de AWS: Node.js y Python 3

Entornos de ejecución y SDK de AWS

Nombre	Descripción	Version information
Node.js (con npm)	Node.js es un entorno de ejecución de JavaScript diseñado para facilitar la aplicación de técnicas de programación asíncrona. Para obtener más información,	• Node.js: `node --version` • npm: `npm --version`

Nombre	Descripción	Version information
	consulte la documentación del sitio oficial de Node.js. npm es un administrador de paquetes que proporciona acceso a un registro en línea de módulos de JavaScript. Para obtener más información, consulte la documentación en el sitio web de npm.	
SDK para JavaScript en Node.js	El kit de desarrollo de software (SDK) ayuda a simplificar la codificación al proporcionar objetos de JavaScript para los servicios de AWS, incluidos Amazon S3, Amazon EC2, DynamoDB y Amazon SWF. Para obtener más información, consulte la Guía para desarrolladores de AWS SDK para JavaScript.	`npm -g ls --depth 0 2>/dev/null | grep aws-sdk`

Nombre	Descripción	Version information
Python	Python 3 está listo para su uso en el entorno del intérprete de comandos. 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
SDK para 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. El SDK proporcio na una API fácil de usar y orientada a objetos, así como un acceso de bajo nivel a los 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 característica de autocompletar para los comandos de Git, añada la siguiente línea para `.bashrc` que la característica esté disponible siempre que se inicie la sesión de AWS CloudShell: `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, consulte la página README en GitHub.	

Nombre	Descripción	Version information
cqlsh-expansion	`cqlsh-expansion` es un conjunto de herramientas que incluye cqlsh y ayudantes preconfigurados para Amazon Keyspaces, al tiempo que mantiene la total compatibilidad con Apache Cassandra. Para obtener más información, consulte Using cqlsh to connect to Amazon Keyspaces en Amazon Keyspaces (for Apache Cassandra) Developer Guide.	`cqlsh-expansion --version`

Nombre	Descripción	Version information
Docker	Docker es una plataform a abierta para desarrollar, enviar y ejecutar aplicaciones. Docker le permite separar las aplicaciones de la infraestr uctura para que pueda entregar software de forma rápida. Permite crear Dockerfil es en AWS CloudShell y crear recursos de Docker con CDK. Para obtener informaci ón sobre qué regiones de AWS son compatibles con Docker, consulte Regiones de AWS compatibles con AWS 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, consulte la Guía de documentación de Docker.	`docker --version`

Nombre	Descripción	Version information
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`
iputils	El paquete iputils contiene utilidades para redes Linux. Para obtener más información sobre las utilidades proporcio nadas, consulte el repositorio de iputils en GitHub.	Ejemplos de una herramienta iputils: `arping -V`
jq	La utilidad jq analiza los datos con formato JSON para producir una salida que se modifica mediante filtros de la línea de comandos. Para obtener más informaci ón, consulte el manual de jq alojado en GitHub.	`jq --version`
kubectl	kubectl es una herramienta de la línea de comandos para comunicarse con el plano de control de un clúster de Kubernetes mediante la API de Kubernetes.	`kubectl --version`

Nombre	Descripción	Version information
make	La utilidad make utiliza `makefiles` para automatizar conjuntos de tareas y organizar la compilación de código. Para obtener más información, consulte la Documentación de GNU Make.	`make --version`
man	El comando man proporciona 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 información, 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 Documentación GMU nano.	`nano --version`

Nombre	Descripción	Version information
OpenJDK 21	Amazon Corretto 21 es una distribución de OpenJDK 21 con soporte a largo plazo (LTS). Amazon Corretto es una distribución sin costo, multiplataforma y lista para producción de Open Java Development Kit (OpenJDK) . Para obtener más informaci ón, consulte What is Amazon Corretto 21? en Corretto 21 User Guide.	`java -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 archivo README que muestra los programas que se pueden ejecutar con procps.	`ps --version`
psql	PostgreSQL es un potente sistema de bases de datos de código abierto que utiliza capacidades SQL estándar y ofrece características robustas para gestionar y escalar de forma segura operaciones de datos complejas. Para obtener más información, consulte What is PostgreSQL.	`psql --version`

Nombre	Descripción	Version information
Cliente SSH	Los clientes SSH utilizan el protocolo del intérprete de comandos seguro para las comunicaciones cifradas con un equipo remoto. OpenSSH es el cliente SSH que viene preinstalado. Para obtener más información, consulte el sitio OpenSSH administrado 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 --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`
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`

Instalación de la AWS CLI en el directorio de inicio

Al igual que el resto del software preinstalado en el entorno CloudShell, la herramienta de la AWS CLI se actualiza automáticamente con actualizaciones programadas y parches de seguridad. Si quiere asegurarse de tener la versión más actualizada de la AWS CLI, puede optar por instalar la herramienta manualmente en el directorio principal del intérprete de comandos.

> **Important**
>
> Debe instalar manualmente la copia de la AWS CLI en el directorio principal para que esté disponible la próxima vez que inicie una sesión de CloudShell. 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 que instale esta copia de la AWS CLI, no se actualiza automáticamente. Es decir, es su responsabilidad administrar las actualizaciones y los parches de seguridad.
> Para más información sobre el modelo de responsabilidad compartida de AWS, consulte Protección de datos en AWS CloudShell.

Para instalar AWS CLI

1. En la línea de comandos de CloudShell, utilice el comando `curl` para transferir una copia comprimida de la AWS CLI instalada al intérprete de comandos:

```
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 instalador la AWS CLI:

```
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`, los comandos de `aws` que no incluyan una ruta específica utilizarán la versión preinstalada de la 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 de computación AWS CloudShell del dispositivo.

De forma predeterminada, todos los usuarios de AWS CloudShell tienen permisos. 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 usar `sudo` con la utilidad de administración de paquetes DNF para instalar `cowsay`, que genera imágenes de arte en formato ASCII 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 como dnf instalan programas en directorios (`/usr/bin`, por ejemplo), que se reciclan cuando finaliza la sesión del intérprete de comandos. 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 `.bashrc` o modificar cualquier otro archivo de forma incorrecta, puede recuperar la configuración predeterminada de AWS CloudShell eliminando el 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.

Migración de AWS CloudShell de AL2 a AL2023

AWS CloudShell, que se basaba en Amazon Linux 2 (AL2), se ha migrado a Amazon Linux 2023 (AL2023). Para obtener más información sobre AL2023, consulte Qué es Amazon Linux 2023 (AL2023) en la Guía del usuario de Amazon Linux 2023.

Con AL2023, puede seguir accediendo a su entorno de CloudShell existente con todas las herramientas que proporciona CloudShell. Para obtener más información sobre las herramientas disponibles, consulte Software preinstalado.

AL2023 proporciona varias mejoras en las herramientas de desarrollo, incluidas las versiones más recientes de paquetes, como Node.js 18 y Python 3.9.

> **Note**
>
> En AL2023, Python 2 ya no se incluye con su entorno de CloudShell.

Para obtener más información sobre las principales diferencias entre AL2 y AL2023, consulte Comparación entre Amazon Linux 2 y Amazon Linux 2023 en la Guía del usuario de Amazon Linux 2023.

Si tiene alguna pregunta, póngase en contacto con Soporte. También puede buscar respuestas y publicar preguntas en AWS re:Post. Cuando acceda a AWS re:Post, es posible que se requiera que inicie sesión en AWS.

Preguntas frecuentes sobre migración de AWS CloudShell

Las siguientes son respuestas a algunas preguntas frecuentes sobre la migración de AL2 a AL2023 con AWS CloudShell.

- Afectará la migración a AL2023 a alguno de mis otros recursos de AWS, como las instancias de Amazon EC2 que se ejecutan en AL2?
- Cuáles son los paquetes que se cambiarán con la migración a AL2023?
- Puedo no participar en la migración?
- ¿Puedo crear una copia de seguridad de mi entorno AWS CloudShell?

¿Afectará la migración a AL2023 a alguno de mis otros recursos de AWS, como las instancias de Amazon EC2 que se ejecutan en AL2?

Esta migración no afecta a ningún servicio o recurso que no sea su entorno AWS CloudShell. Esto incluye los recursos que puede haber creado o a los que haya accedido desde AWS CloudShell. Por ejemplo, si ha creado una instancia de Amazon EC2 que se ejecuta en AL2, no se migrará a AL2023.

¿Cuáles son los paquetes que han cambiado con la migración a AL2023?

Los entornos AWS CloudShell actualmente incluyen software preinstalado. Para obtener 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 AL2023, consulte Comparación entre AL2 y AL2023. Para los clientes con requisitos específicos de paquete y versión que dejarán de cumplirse tras la migración a AL2023, les recomendamos que se pongan en contacto con AWS Support para enviar una solicitud.

¿Puedo no participar en la migración?

No, no puede no participar en la migración. AWS administra los entornos de AWS CloudShell, por lo tanto, todos los entornos se han actualizado a la versión AL2023.

¿Puedo crear una copia de seguridad de mi entorno AWS CloudShell?

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

Acceso denegado

Problema: Al intentar iniciar CloudShell desde el Consola de administración de AWS, aparece el mensaje «No se puede iniciar el entorno». Para volver a intentarlo, actualiza el navegador o reinicia seleccionando Acciones, reinicia AWS CloudShell. Se le deniega el acceso incluso después de haber solicitado los permisos de su administrador de IAM y de haber actualizado o reiniciado el navegador. CloudShell

Solución: póngase en contacto con AWS Support.

(Volver arriba)

Permisos insuficientes

Problema: Al intentar iniciar el entorno CloudShell desde Consola de administración de AWS, aparece el mensaje «No se ha podido iniciar el entorno». No dispone del permiso necesario. Pida al administrador de IAM que le conceda acceso a AWS CloudShell». Se le deniega el acceso y se le notifica que no dispone de los permisos necesarios.

Causa: la identidad de IAM a la que está accediendo AWS CloudShell carece de los permisos de IAM necesarios.

Solución: solicite al administrador de IAM 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 las políticas de 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 protocolo ICMP no está activado AWS CloudShell, la utilidad ping no funciona en el entorno informático del shell.

Solución: debido a que el ICMP no es compatible AWS CloudShell, puede ejecutar el siguiente comando para instalar Netcat. Netcat es una utilidad de redes computacionales para leer conexiones de red y escribir en ellas 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: cuando intenta 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 específica URLs. Puede añadir parte de la 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 especificar `wss://ssmmessages.region.amazonaws.com` cuando actualice su lista de direcciones URL

permitidas. Para obtener más información, consulta la StartSessionoperación en la referencia de la AWS Systems Manager API.

(Volver arriba)

No se pudo importar el módulo **AWSPowerShell.NetCore**

Problema: al importar el AWSPower Shell. NetCoreEn el módulo PowerShell by`Import-Module -Name AWSPowerShell.NetCore`, recibirá el siguiente mensaje de error:

Import-Module: el módulo especificado, «Shell». AWSPower 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 Herramientas de AWS para PowerShell.

(Volver arriba)

Docker no se ejecuta cuando se usa AWS CloudShell

Problema: Docker no se ejecuta de forma correcta cuando se usa AWS CloudShell. Se 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 región. GovCloud Asegúrese de ejecutar Docker en las regiones compatibles AWS . Para ver una lista de las regiones en las que Docker está disponible, consulta AWS las regiones compatibles para. AWS CloudShell

Docker se ha quedado sin espacio en disco

Problema: se 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 que haya imágenes individuales de gran tamaño o 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 crea que pueden ser grandes y considere la posibilidad de eliminar otros archivos para liberar espacio. Una opción sería considerar la posibilidad de eliminar las imágenes de Docker no utilizadas mediante la ejecución de `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.

Se está agotando el tiempo de espera de **`docker push`** y sigue intentándolo

Problema: cuando se ejecuta `docker push`, se agota el tiempo de espera de este y sigue intentándolo sin éxito.

Causa: esto puede deberse a la falta de permisos, a la inserción en un repositorio incorrecto o a la falta de autenticación.

Solución: para intentar resolver este problema, asegúrese de realizar la inserción en el repositorio correcto. Ejecute `docker login` para autenticar de forma adecuada. Asegúrese de tener todos los permisos necesarios para insertar en un repositorio de Amazon ECR.

No puedo acceder a los recursos de la VPC desde mi entorno de AWS CloudShell VPC

Problema: no puedo acceder a los recursos de la VPC mientras utilizo mi entorno de VPC AWS CloudShell .

Causa: el entorno de AWS CloudShell VPC hereda la configuración de red de la VPC.

Solución: para resolver este problema, asegúrese de que la VPC esté configurada de forma correcta para acceder a sus recursos. Para obtener más información, consulte en la documentación de la VPC el tema Conectar la VPC a otras redes y la documentación del Analizador de acceso a la red. Para encontrar la IPv4 dirección que utiliza el entorno de AWS CloudShell VPC, ejecute el comando `ip -a` dentro de su entorno en la línea de comandos o en la página de la consola de VPC.

El ENI utilizado AWS CloudShell por mi entorno de VPC no está limpio

Problema: no se puede limpiar la interfaz de red elástica que AWS CloudShell utiliza para mi entorno de VPC.

Causa: el permiso `ec2:DeleteNetworkInterface` no está habilitado para su rol.

Solución: para resolver este problema, asegúrese de que el permiso `ec2:DeleteNetworkInterface` esté habilitado para su rol, tal y 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 solo para entornos de VPC también tiene acceso a entornos públicos. AWS CloudShell

Problema: El usuario restringido con `CreateEnvironment` permiso solo para entornos de VPC también puede acceder a los entornos públicos AWS CloudShell .

Causa: si limita `CreateEnvironment` los permisos para la creación de entornos de VPC únicamente y si ya ha creado un entorno público, conservará el acceso al CloudShell entorno público existente hasta que se elimine este entorno mediante la interfaz de usuario web. Sin embargo, si nunca los ha utilizado CloudShell antes, no tendrá acceso a los entornos públicos.

Solución: para restringir el acceso a los AWS CloudShell entornos públicos, el administrador de IAM primero debe actualizar la política de IAM 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).

Regiones de AWS compatibles con AWS CloudShell

En esta sección se incluye la lista de regiones de AWS compatibles y regiones registradas para AWS CloudShell. Para obtener una lista de puntos de conexión y cuotas de servicios de AWS para CloudShell, consulte la página AWS CloudShell en Referencia general de Amazon Web Services.

A continuación se indican las regiones de AWS compatibles con el entorno de VPC de CloudShell, Docker y CloudShell:

- Este de EE. UU. (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-Pacífico (Mumbai)
- 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)
- Medio Oriente (EAU)
- América del Sur (São Paulo)

Regiones de GovCloud

Las siguientes son las regiones de GovCloud compatibles con CloudShell:

- AWS GovCloud (Este de EE. UU.)
- AWS GovCloud (Oeste de EE. UU.)

> **Note**
>
> El entorno de VPC de CloudShell y Docker está disponible en las regiones de GovCloud.

Service Quotas y restricciones para AWS CloudShell

En esta página se describen las Service Quotas y restricciones que se aplican a las siguientes áreas:

- Almacenamiento persistente
- Uso mensual
- Intérprete de comandos simultáneos
- Tamaño del comando
- Sesiones del intérprete de comandos
- Entornos de VPC
- Acceso a la red y transferencia de datos
- Archivos del sistema y recargas de páginas

Almacenamiento persistente

Con AWS CloudShell, tiene un almacenamiento persistente de 1 GB para cada Región de AWS sin costo 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 entre las sesiones.

> **Note**
>
> Los entornos de VPC de CloudShell no tienen almacenamiento persistente. El directorio $HOME se elimina cuando se agota el tiempo de espera del entorno de VPC (después de unos 20-30 minutos de inactividad), o bien cuando elimina su entorno.

Si deja de usar AWS CloudShell en una Región de AWS, los datos se conservan en el almacenamiento persistente de esa región durante 120 días después del final de su 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, consulte el Paso 2: seleccione una región, inicie AWS CloudShell y elija un intérprete de comandos.

> **Note**
>
> Escenario de uso
> Márcia ha usado AWS CloudShell para almacenar archivos en sus directorios principales en dos Regiones de AWS: Este de EE. UU. (Norte de Virginia) y Europa (Irlanda). Luego comenzó a usar AWS CloudShell exclusivamente en Europa (Irlanda) y dejó de iniciar sesiones del intérprete de comandos en el Este de EE. UU. (Norte de Virginia).
> Antes de que venza la fecha límite para eliminar datos en el Este de EE. UU. (Norte de Virginia), Márcia decide impedir que su directorio principal sea reciclado abriendo AWS CloudShell y seleccionando de nuevo la región del Este de EE. UU. (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 Región de AWS de su Cuenta de AWS tiene una cuota de uso mensual para AWS CloudShell. Esta cuota combina el tiempo total dedicado a usar CloudShell por todas las entidades principales de IAM en esa región. Si intenta acceder a CloudShell después de haber alcanzado la cuota mensual para esa región, aparece un mensaje en el que se explica por qué no se puede iniciar el entorno del intérprete de comandos.

Para solicitar un aumento, visite la Consola de Service Quotas

Para solicitar un aumento de las cuotas de uso mensuales, abra la consola de Service Quotas. Para obtener más información, consulte Solicitud de aumento de cuota en la Guía del usuario de Service Quotas.

Intérprete de comandos simultáneos

Puede iniciar hasta 10 intérpretes de comandos al mismo tiempo en cada Región de AWS para su cuenta.

Para solicitar un aumento, visite la Consola de Service Quotas

Puede solicitar un aumento de la cuota para cada región abriendo la consola de Service Quotas. Para obtener más información, consulte Solicitud de aumento de cuota en la Guía del usuario de Service Quotas.

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.

Sesiones del intérprete de comandos

- Sesiones inactivas: AWS CloudShell es un entorno de intérprete de comandos interactivo. Si no interactúa con él a través del teclado o el puntero durante 20 o 30 minutos, se cerrará la sesión del intérprete de comandos. Los procesos en ejecución no cuentan como interacciones.

  Si desea realizar tareas basadas en terminal utilizando un servicio de AWS con tiempos de espera más flexibles, le recomendamos que inicie y se conecte a una instancia de Amazon EC2.
- 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.

Entornos de VPC

Solo puede crear hasta dos entornos de VPC por entidad principal de IAM.

> **Note**
>
> No se aplica ningún cargo por la conexión a su VPC privada y el acceso a los recursos que contiene. Las transferencias de datos dentro de su VPC privada se incluyen en la facturación

de la VPC y las que tienen lugar entre sus VPC a través de CloudShell se cobran al mismo costo que en su instancia de CloudShell actual.

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 red de Internet público, existe el riesgo de que algunos usuarios exporten datos del entorno AWS CloudShell. Recomendamos que los administradores de IAM gestionen la lista de usuarios de AWS CloudShell de confianza permitidos mediante las herramientas de IAM. 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 a y de 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 de computación, es posible que tenga problemas al acceder o utilizar el entorno AWS CloudShell. 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 para la guía del usuario de AWS CloudShell

Actualizaciones recientes

En la siguiente tabla se describen cambios importantes en la Guía del usuario de AWS CloudShell.

Cambio	Descripción	Fecha
CLI de Amazon Q en AWS CloudShell	Se ha agregado compatibilidad para usar las características de la CLI de Amazon Q en AWS CloudShell.	2 de octubre de 2024
Compatibilidad con Amazon VPC para AWS CloudShell en determinadas regiones	Se ha agregado compatibilidad para la creación y el uso de entornos de VPC de AWS CloudShell en determinadas regiones.	13 de junio de 2024
Adición de nuevos tutoriales a la Guía del usuario de AWS CloudShell	Se han agregado dos nuevos tutoriales en los que se detalla cómo crear un contenedor de Docker en AWS CloudShell y cómo insertarlo en un repositorio de Amazon ECR, además de cómo implementar una función de Lambda mediante AWS CDK.	27 de diciembre de 2023
Compatibilidad de los contenedores de Docker con AWS CloudShell en determinadas regiones	Se ha agregado compatibilidad para los contenedores de Docker con AWS CloudShell en algunas regiones.	27 de diciembre de 2023

AWS CloudShell se ha migrado y ahora usa Amazon Linux 2023 (AL2023)	AWS CloudShell ahora usa AL2023 y se ha migrado desde Amazon Linux 2.	4 de diciembre de 2023
Nuevas regiones de AWS para AWS CloudShell	AWS CloudShell está disponible en las siguientes regiones de AWS: • Oeste de EE. UU. (Norte de California) • Africa (Cape Town) • 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) • Medio Oriente (EAU)	16 de junio de 2023
Iniciar AWS CloudShell en la Console Toolbar	Inicie CloudShell en Console Toolbar, en la parte inferior izquierda de la consola seleccionando CloudShell.	28 de marzo de 2023
Nuevas regiones de AWS para AWS CloudShell	AWS CloudShell está disponible en las siguientes regiones de AWS: • Canadá (centro) • Europa (Londres) • América del Sur (São Paulo)	6 de octubre de 2022

AWS CloudShell compatible con AWS GovCloud de EE. UU	AWS CloudShell ya se admite en la región AWS GovCloud (EE. UU.).	29 de junio de 2022
Preguntas frecuentes sobre seguridad	Preguntas frecuentes adicionales centradas en cuestiones de seguridad.	14 de abril de 2022
Web Sockets	Se agregó una sección a los requisitos de red que explica el uso del protocolo WebSocket por parte de CloudShell.	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 regiones de AWS	Documentación sobre la especificación de la Región de AWS por defecto para comandos de la AWS CLI.	11 de mayo de 2021
Formateo en las versiones PDF y Kindle	Tamaños de imagen y texto fijos en las celdas de la tabla.	10 de marzo de 2021

Versión de disponibilidad general (GA) de AWS CloudShell en regiones de AWS seleccionadas	AWS CloudShell está disponible en las siguientes regiones de AWS: • Este de EE. UU. (Ohio) • Este de EE. UU. (Norte de Virginia) • Oeste de EE. UU. (Oregón) • Asia-Pacífico (Tokio) • Europa (Irlanda) • Asia-Pacífico (Mumbai) • Asia-Pacífico (Sídney) • Europa (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.