

# AWS CLI



AWS CLI es una forma cómoda de interactuar con AWS desde un terminal. Se ejecuta en Linux, macOS y Windows y proporciona una interfaz unificada para todos los servicios de AWS. A menos que se especifique lo contrario, la salida está por defecto en formato JSON.

## 1 Instalación de AWS CLI

La forma de proceder depende de su sistema operativo. Si se tienen dificultades para instalar la CLI, consultar [https://docs.aws.amazon.com/es\\_es/cli/latest/userguide/getting-started-prereqs.html#getting-started-prereqs-next](https://docs.aws.amazon.com/es_es/cli/latest/userguide/getting-started-prereqs.html#getting-started-prereqs-next) para obtener una descripción detallada de las opciones de instalación.

### Linux x86 (64 bits)

En un terminal, ejecutar los siguientes comandos:

```
$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"  
$ unzip awscliv2.zip  
$ sudo ./aws/install
```

Verifica tu instalación de AWS CLI ejecutando `aws --version` en tu terminal. La versión debe ser al menos 2.4.0.

### macOS

Los siguientes pasos le guiarán a través de la instalación de la CLI de AWS en macOS utilizando el instalador:

Descarga el instalador de la CLI de AWS en <https://awscli.amazonaws.com/AWSCLIV2.pkg>.

Ejecute el instalador descargado e instale la CLI siguiendo el asistente de instalación para todos los usuarios.

Verifique su instalación de la CLI de AWS ejecutando `aws --version` en su terminal. La versión debe ser al menos 2.4.0.

### Windows

Los siguientes pasos le guiarán a través de la instalación de la CLI de AWS en Windows utilizando el instalador MSI:

Información sobre AWS CLI: <https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

Descargar el instalador de la CLI de AWS en <https://awscli.amazonaws.com/AWSCLIV2.msi>

Ejecutar el instalador descargado e instala la CLI siguiendo el asistente de instalación.

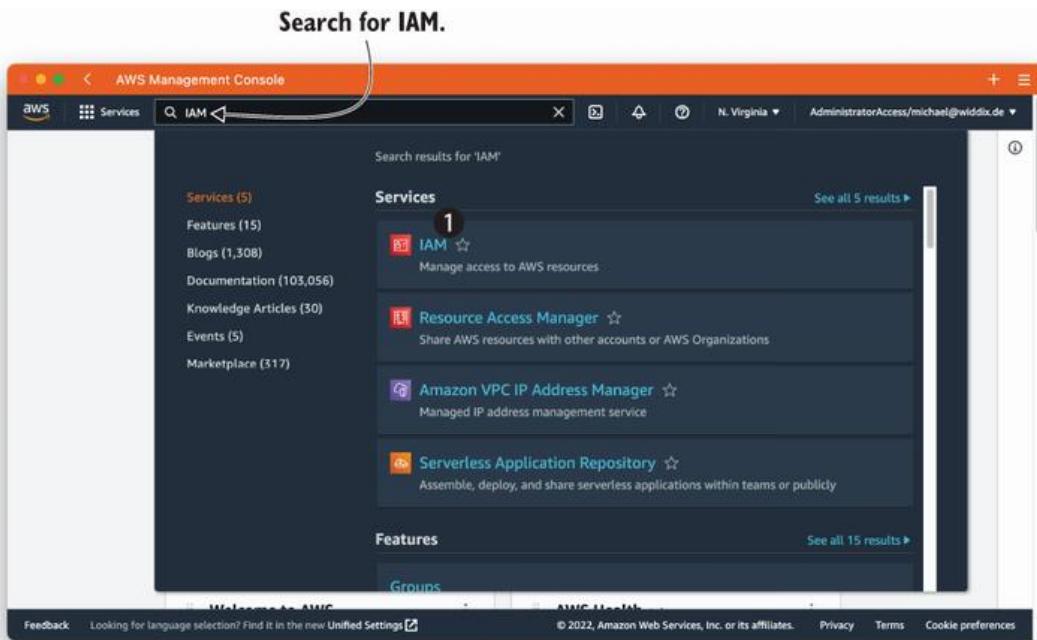
Ejecutar PowerShell como administrador buscando "PowerShell" en el menú Inicio y seleccionando *Ejecutar como administrador* en su menú contextual.

- Escribir **Set-ExecutionPolicy Unrestricted** en PowerShell y pulse *Intro* para ejecutar el comando. Esto le permitirá ejecutar los scripts PowerShell sin firmar de nuestros ejemplos.
- Cerrar la ventana de PowerShell; ya no necesitamos trabajar como administrador.
- Ejecutar PowerShell seleccionando PowerShell en el menú Inicio.

Comprobar si AWS CLI funciona ejecutando `aws --version` en PowerShell. La versión debe ser al menos 2.4.0.

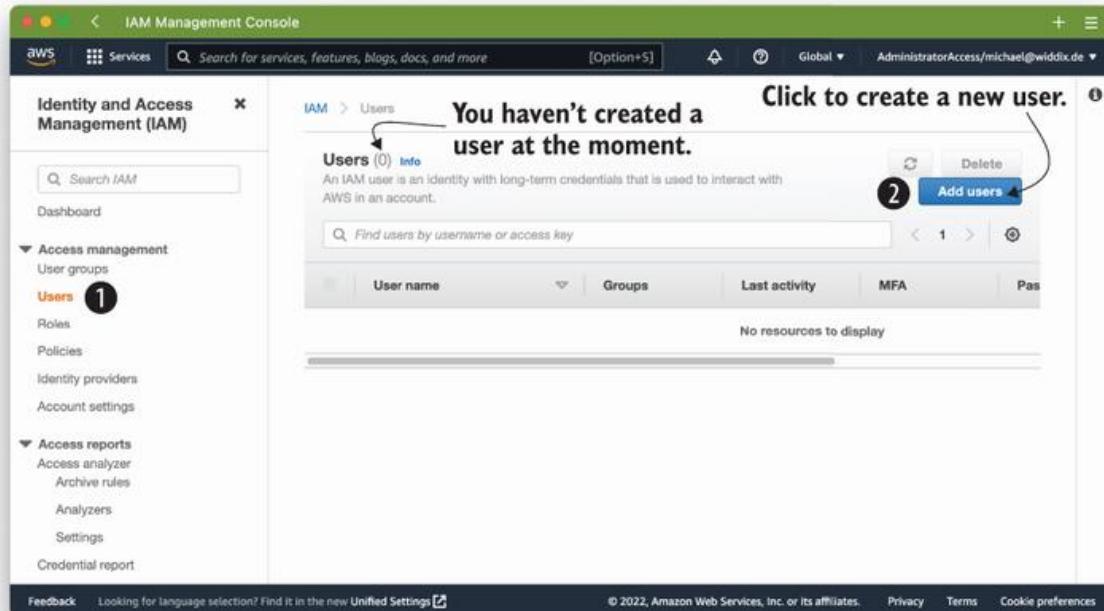
## 2 Configuración AWS CLI

Para usar el CLI, necesitamos autenticarnos. Hasta ahora, si usamos cuenta *root* de AWS se puede hacer de todo, bueno y malo. Es muy recomendable que no utilizar la cuenta raíz de AWS, así que vamos a crear un nuevo usuario. Para crear un nuevo usuario, sigue los siguientes pasos, que se ilustran en la imagen:



- Abrir la consola de administración de AWS en <https://console.aws.amazon.com>.
- Hacer clic en Servicios y buscar IAM.
- Abrir el servicio IAM.

Se abre una página como la que se muestra en la imagen; seleccionar Usuarios a la izquierda.



Seguir estos pasos para crear un usuario:

- Hacer clic en Añadir Usuarios para abrir la página mostrada en la figura 4.4.
- Introducir *mycli* como nombre de usuario.
- En AWS credential type, seleccionar Access Key-Programmatic Access.
- Haz clic en el botón Next: Permisos.

Add user

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\* mycli

[+ Add another user](#)

User name of the new user is mycli.

Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Select AWS credential type\*

Access key - Programmatic access  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

Password - AWS Management Console access  
Enables a **password** that allows users to sign-in to the AWS Management Console.

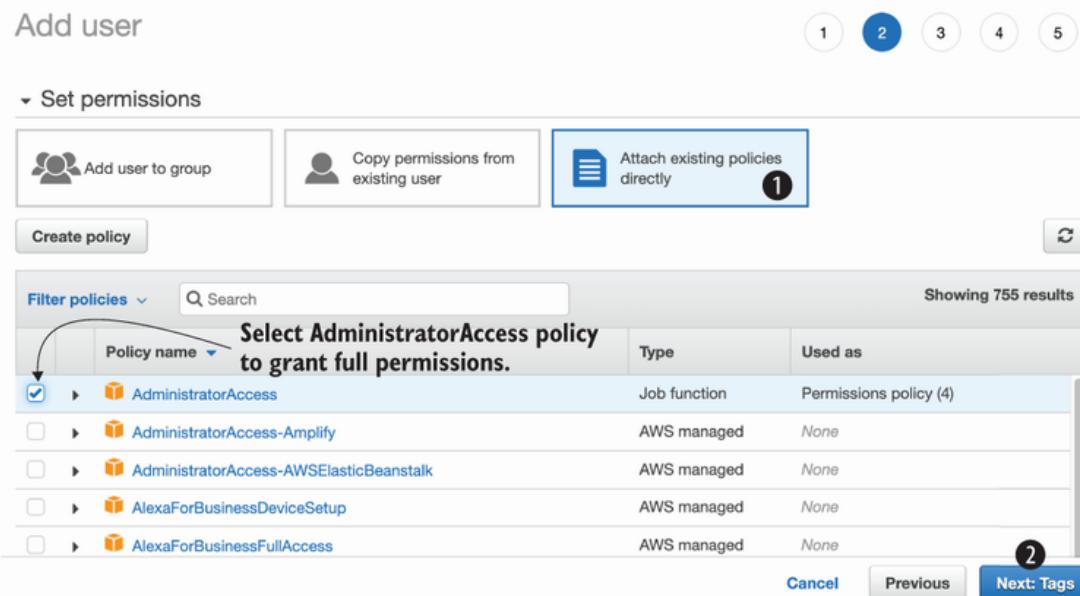
Check Programmatic access to generate an access key.

\* Required

Cancel Next: Permissions

En el siguiente paso, se tienen que definir los permisos para el nuevo usuario, como se muestra en la imagen:

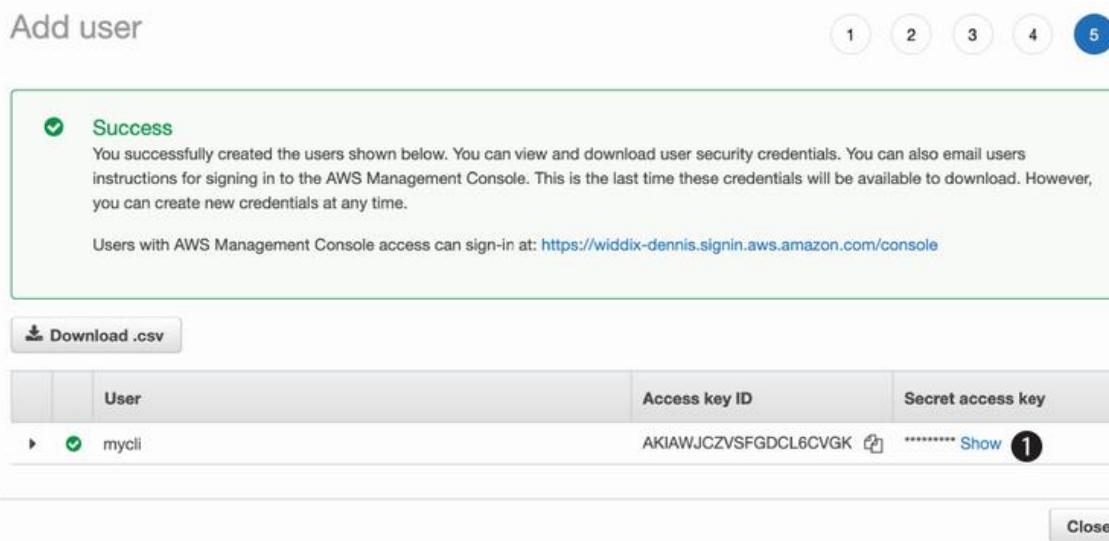
- Hacer clic en Adjuntar políticas existentes directamente.
- Seleccionar la política *AdministratorAccess*.
- Hacer clic en el botón Siguiente: *Etiquetas*.



No se necesitan etiquetas, así que se hace clic en el botón Siguiente: *Revisar*.

La página de revisión resume lo que se ha configurado. Hacer clic en el botón Crear usuario para guardar. Finalmente, se verá la página mostrada en la imagen. Haga clic en el enlace *Mostrar* para mostrar el valor de las claves. Ahora se necesita copiar las credenciales a nuestra configuración de AWS CLI.

**ADVERTENCIA** Tratar el ID de la clave de acceso y la clave de acceso secreta con cuidado. Cualquiera que obtenga acceso a estas credenciales tendrá acceso de administrador a su cuenta de AWS.



Abrir el terminal de tu ordenador (PowerShell en Windows o un *shell* en Linux y macOS; no la consola de administración de AWS) y ejecutar `aws configure`. Se pedirán los siguientes cuatro datos:

ID de la clave de acceso de AWS: copia y pega este valor de la columna ID de la clave de acceso (la ventana de tu navegador).

- Clave de acceso secreta de AWS: copiar y pegar este valor de la columna Clave de acceso secreta (ventana del navegador).
- Nombre de región predeterminado: introducir us-east-1.
- Formato de salida predeterminado: introducimos json.
- Al final, el terminal debería tener un aspecto similar al siguiente:

```
$ aws configure
AWS Access Key ID [None]: AKIAIRUR3YLPOSVD7ZCA      ①
AWS Secret Access Key [None]:
→ SSKIng7jkAKERpcT3YphX4cD87sBYgWVw2enqBj7          ②
Default region name [None]: us-east-1
Default output format [None]: json
```

① ¡Tu valor será diferente! Cópialo desde la ventana de tu navegador.

② ¡Tu valor será diferente! Cópialo desde la ventana de tu navegador.

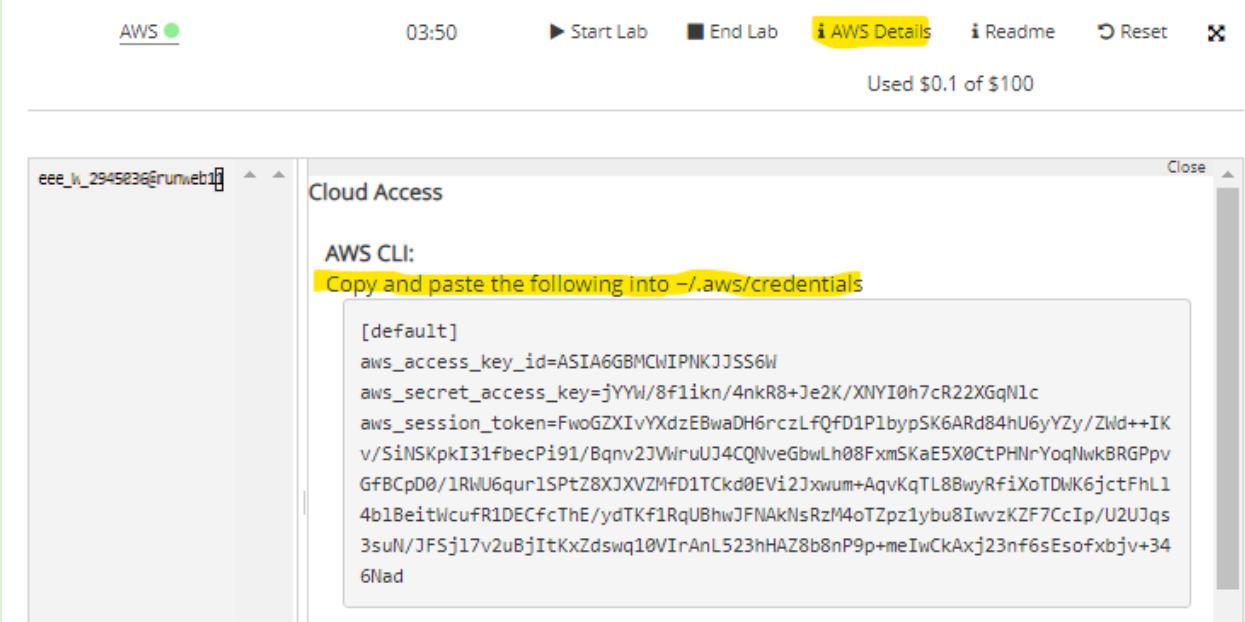
AWS CLI está ahora configurado para autenticarse como el usuario *mycli*. Volvemos a la ventana del navegador y hacemos clic en *Cerrar* para finalizar el asistente de creación de usuarios.

### !!! NOTA IMPORTANTE !!!!

Este comando almacena las credenciales en el fichero de configuración `~/.aws/credentials`.

En nuestro caso (AWS ACADEMY) no tenemos acceso a la mayoría de las configuraciones de usuarios (servicio IAM), por lo tanto, estas credenciales ya nos vienen dadas de antemano y tienen una configuración un poco distinta.

Para ello debemos ir a la ventana de lanzamiento del laboratorio y buscarlas donde se muestra en la imagen:



Como nos indica la imagen, debemos de copiar las tres claves que nos muestra con sus valores en el archivo **.aws/credentials** que se encuentra en la carpeta del perfil de nuestro usuario, que por tanto ha de quedar como muestra la imagen:

```
$ cat .aws/credentials
[default]
aws_access_key_id=ASIA6GBMCWIPNKKJSS6W
aws_secret_access_key=jYYW/8f1kn/4nkR8+Je2K/XNYI0h7cR22XGqNlc
aws_session_token=FwoGZXIvYXdzEBwaDH6rczLfqfd1PlbypSK6ARd84hU6yYZy/ZWd++IKv/SiNSKpkI31fbecPi91/Bqnv
2JVWruUJ4CQNveGbwlh08FxmSKaE5X0CtPHNrYoqNwkBRGPpvGfBCpD0/1RWU6qurLSPTZ8XJVZMF1TCkd0EVi2Jxwum+AqvK
qTL8BwyRfiXoTDWK6jctFhL14blBeitWcuFR1DECfcThE/ydTKf1RqUBhwJFNAkNsRzM4oTzp1ybu8IwvzKZF7CcIp/U2UJqs3
suN/JFSjl7v2uBjItKxZdsdq10VlrAnL523hHAZb8nP9p+meIwCkAxj23nf6sEsofxbjv+346Nad
```

Tener en cuenta que estas credenciales de AWS ACADEMY caducan y cambian en cada reinicio del laboratorio, por lo que tendremos que hacer este paso cada vez que lo arrancamos.

Es hora de probar si la CLI funciona. Cambiamos a la ventana de terminal e introduzcamos `aws ec2 describe-regions` para obtener una lista de todas las regiones disponibles, como se muestra aquí:

```
$ aws ec2 describe-regions
{
  "Regions": [
    {
      "Endpoint": "ec2.eu-north-1.amazonaws.com",
      "RegionName": "eu-north-1",
      "OptInStatus": "opt-in-not-required"
    },
    [...]
    {
      "Endpoint": "ec2.us-west-2.amazonaws.com",
      "RegionName": "us-west-2",
      "OptInStatus": "opt-in-not-required"
    }
  ]
}
```

¡Funciona! Ya se podemos empezar a utilizar la CLI.

### 3 Uso de AWS CLI

Lista completa de comandos AWS CLI:

<https://awscli.amazonaws.com/v2/documentation/api/latest/index.html#>

Supongamos que se desea obtener una lista de todas las instancias EC2 en ejecución de tipo `t2.micro` para que pueda ver lo que se está ejecutando en su cuenta de AWS. Ejecute el siguiente comando en su terminal:

```
$ aws ec2 describe-instances --filters "Name=instance-type,Values=t2.micro"
```

```
{  
  "Reservations": []      ①  
}  
}
```

① La lista está vacía porque no has creado una instancia EC2.

Para utilizar AWS CLI, se necesita especificar un servicio y una acción. En el ejemplo anterior, el servicio es `ec2` y la acción es `describe-instances`. Puedes añadir opciones con el valor `--name` de la siguiente manera:

```
$ aws <service> <action> [--name value ...]
```

Una característica importante de la CLI es la palabra clave `help`. Puede obtener ayuda en los siguientes tres niveles de detalle:

`aws help` -Muestra todos los servicios disponibles

`aws <service> help` -Muestra todas las acciones disponibles para un determinado servicio

`aws <service> <action> help` -Muestra todas las opciones disponibles para la acción de un servicio concreto.

Una característica importante de AWS CLI necesita explicación antes de empezar. La opción `--query` utiliza la sintaxis JMESPath, que es un lenguaje de consulta para JSON, para extraer datos del resultado. Esto puede ser útil porque normalmente sólo se necesita un campo específico del resultado. El siguiente comando CLI obtiene una lista de todas las AMI en formato JSON:

```
$ aws ec2 describe-images --filters \  
→  "Name=name,Values=amzn2-ami-hvm-2.0.202*-x86_64-gp2"      ①  
{  
  "Images": [  
    {  
      "ImageId": "ami-0ce1e3f77cd41957e",  
      "State": "available"  
      [...]  
    },  
    [...]  
    {  
      "ImageId": "ami-08754599965c30981",  
      "State": "available"  
    }  
  ]
```

```
}
```

- ① El filtro sólo devuelve imágenes de Amazon Linux 2 para AMD/Intel.

La salida es abrumadora. Para iniciar una instancia EC2, necesitas el ImageId sin toda la demás información. Con JMESPath, puedes extraer sólo esa información, así:

```
$ aws ec2 describe-images --filters \  
→ "Name=name,Values=amzn2-ami-hvm-2.0.202*-x86_64-gp2" \  
→ --query "Images[0].ImageId"          ①  
"ami-146e2a7c"
```

- ① Devuelve el primer ID de imagen de la lista.

La salida aparece entre comillas. Esto es causado por la configuración predeterminada de la CLI de AWS para dar salida a todos los datos en formato JSON y las cadenas JSON se encierran entre comillas. Para cambiar esto, se puede utilizar la opción `--output text` de la siguiente manera para formatear la salida como múltiples líneas de valores de cadena separados por tabuladores. Esta configuración puede ser útil para pasar la salida a un procesador de texto, como grep, sed o awk:

```
$ aws ec2 describe-images --filters \  
→ "Name=name,Values=amzn2-ami-hvm-2.0.202*-x86_64-gp2" \  
→ --query "Images[0].ImageId" --output text          ①  
ami-146e2a7c
```

- ① Establece el formato de salida en texto plano.