

# Instalación y configuración

Existen varias formas diferentes de instalar AWS CLI en su máquina, según el sistema operativo y el entorno que esté utilizando:

## Instalar en Windows

El AWS CLI es compatible con Microsoft Windows XP o posterior. Para los usuarios de Windows, el paquete de instalación MSI ofrece una forma familiar y conveniente de instalar AWS CLI sin instalar ningún otro requisito previo. Los usuarios de Windows deben usar el instalador de MSI a menos que ya estén usando pip para la administración de paquetes.

- [MSI Installer para Windows de 32 bits](#)
- [MSI Installer para Windows de 64 bits](#)

Ejecute el instalador MSI descargado. Siga las instrucciones que aparecen.

## Instalar en Linux

```
sudo apt install awscli
```

### Instalar en Linux (manualmente)

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

```
unzip awscliv2.zip
```

```
sudo ./aws/install
```

### Instalar utilizando pip:

Necesitará que se instale Python (versión 2, 2.6.5 +, 3 o 3.3+). Comprueba con

```
python --version
```

```
pip --help
```

Dado que ambos están instalados, use el siguiente comando para instalar aws cli.

```
sudo pip install awscli
```

## Para instalar AWS CLI utilizando el instalador incluido

Requisitos previos:

1. Descargue el instalador empaquetado de AWS CLI usando wget o curl.
2. Descomprima el paquete.
3. Ejecuta el ejecutable de instalación.

En Linux y OS X, aquí están los tres comandos que corresponden a cada paso:

```
$ curl "https://s3.amazonaws.com/aws-cli/awscli-bundle.zip" -o "awscli-bundle.zip"
```

```
$ unzip awscli-bundle.zip
```

```
$ sudo ./awscli-bundle/install -i /usr/local/aws -b /usr/local/bin/aws
```

## Instalar utilizando HomeBrew en OS X:

Otra opción para OS X

```
brew install awscli
```

## Probar la instalación de AWS CLI

Confirme que la CLI está instalada correctamente al ver el archivo de ayuda. Abra un terminal, shell o indicador de comando, ingrese la ayuda de aws y presione Enter:

```
$ aws help
```

## Configurando la CLI de AWS

Una vez que hayas terminado la instalación necesitas configurarla. Necesitará su clave de acceso y la clave secreta que obtendrá cuando cree su cuenta en aws. También puede especificar un [nombre de](#) región predeterminado y un tipo de salida predeterminado (texto | tabla | json).

```
$ aws configure
```

```
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
```

```
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

```
Default region name [None]: us-west-2
```

```
Default output format [None]: ENTER
```

## Creando un nuevo perfil

Para configurar un nuevo perfil de credencial con el nombre myprofile :

```
$ aws configure --profile myprofile
AWS Access Key ID [None]: ACCESSKEY
AWS Secret Access Key [None]: SECRETKEY
Default region name [None]: REGIONNAME
Default output format [None]: text | table | json
```

Para el ID y el secreto de la clave de acceso de AWS, cree un usuario de IAM en la consola de AWS y genere claves para ello.

La región será la región predeterminada para los comandos en el formato eu-west-1 o us-east-1 .

El formato de salida predeterminado puede ser text , table o json .

Ahora puede usar el nombre del perfil en otros comandos usando la opción --profile , por ejemplo:

```
$ aws ec2 describe-instances --profile myprofile
```

Las bibliotecas de AWS para otros idiomas (por ejemplo, aws-sdk para Ruby o boto3 para Python) también tienen opciones para usar el perfil que creas con este método. Por ejemplo, crear una nueva sesión en boto3 se puede hacer de esta manera, boto3.Session(profile\_name:'myprofile') y utilizará las credenciales que creó para el perfil.

Los detalles de su configuración de aws-cli se pueden encontrar en ~/.aws/config y ~/.aws/credentials (en linux y mac-os). Estos detalles se pueden editar manualmente desde allí.

## Usando los comandos aws cli

La sintaxis para usar aws cli es la siguiente:

```
aws [options] <command> <subcommand> [parameters]
```

Algunos ejemplos que usan el comando 'ec2' y el subcomando 'describe-instance':

```
aws ec2 describe-instances
```

```
aws ec2 describe-instances --instance-ids <your-id>
```

Ejemplo con una identificación falsa:

```
aws ec2 describe-instances --instance-ids i-c71r246a
```

## Listar cubos S3

```
aws s3 ls
```

Usar un perfil con nombre

```
aws --profile myprofile s3 ls
```

Enumere todos los objetos en un contenedor, incluidos los objetos en carpetas, con tamaño en formato legible para las personas y un resumen de las propiedades de los contenedores al final:

```
aws s3 ls --recursive --summarize --human-readable s3://<bucket_name>/
```

## AWS completar para Ubuntu con Bash

La siguiente utilidad se puede utilizar para completar automáticamente los comandos:

```
$ which aws_completer  
/usr/bin/aws_completer
```

```
$ complete -C '/usr/bin/aws_completer' aws
```

Para futuras sesiones de shell, considere agregar esto a su `~ / .bashrc`

```
$ echo "complete -C '/usr/bin/aws_completer' aws" >> ~/.bashrc
```

Para comprobar, escriba:

```
$ aws ec
```

Presione la tecla [TAB], debería agregar 2 automáticamente:

```
$ aws ec2
```

# Hoja de trucos de AWS CLI - Lista de todos los comandos de CLI

## Preparar

### Instalar AWS CLI (Python)

AWS CLI es una herramienta común de CLI para administrar los recursos de AWS. Con esta única herramienta podemos gestionar todos los recursos de aws.

```
sudo apt-get install -y python-dev python-pip
sudo pip install awscli
aws --version
aws configure
```

### Bash one-liners

```
cat <file> # output a file
tee # split output into a file
cut -f 2 # print the 2nd column, per line
sed -n '5{p;q}' # print the 5th line in a file
sed 1d # print all lines, except the first
tail -n +2 # print all lines, starting on the 2nd
head -n 5 # print the first 5 lines
tail -n 5 # print the last 5 lines

expand # convert tabs to 4 spaces
unexpand -a # convert 4 spaces to tabs
wc # word count
tr ' '\t # translate / convert characters to other characters

sort # sort data
uniq # show only unique entries
paste # combine rows of text, by line
join # combine rows of text, by initial column value
```

# Cloudtrail - Registro y Auditoría

<http://docs.aws.amazon.com/cli/latest/reference/cloudtrail/>

5 Trails total, con soporte para permisos de nivel de recursos

# list all trails

```
aws cloudtrail describe-trails
```

# list all S3 buckets

```
aws s3 ls
```

# create a new trail

```
aws cloudtrail create-subscription \  
  --name awslog \  
  --s3-new-bucket awslog2016
```

# list the names of all trails

```
aws cloudtrail describe-trails --output text | cut -f 8
```

# get the status of a trail

```
aws cloudtrail get-trail-status \  
  --name awslog
```

# delete a trail

```
aws cloudtrail delete-trail \  
  --name awslog
```

# delete the S3 bucket of a trail

```
aws s3 rb s3://awslog2016 --force
```

# add tags to a trail, up to 10 tags

```
aws cloudtrail add-tags \  
  --resource-id awslog \  
  --tags-list "Key=log-type,Value=all"
```

# list the tags of a trail

```
aws cloudtrail list-tags \  
  --resource-id-list
```

# remove a tag from a trail

```
aws cloudtrail remove-tags \  
  --resource-id awslog \  
  --tags-list "Key=log-type,Value=all"
```

# IAM

## Usuarios

[How to Rotate Access Keys for IAM Users | AWS Security Blog](#)

[http://docs.aws.amazon.com/IAM/latest/UserGuide/reference\\_iam-limits.html](http://docs.aws.amazon.com/IAM/latest/UserGuide/reference_iam-limits.html)

Límites = 5000 usuarios, 100 grupos, 250 roles, 2 teclas de acceso / usuario

<http://docs.aws.amazon.com/cli/latest/reference/iam/index.html>

```
# list all user's info
```

```
aws iam list-users
```

```
# list all user's usernames
```

```
aws iam list-users --output text | cut -f 6
```

```
# list current user's info
```

```
aws iam get-user
```

```
# list current user's access keys
```

```
aws iam list-access-keys
```

```
# create new user
```

```
aws iam create-user \  
    --user-name aws-admin2
```

```
# create multiple new users, from a file
```

```
allUsers=$(cat ./user-names.txt)
```

```
for userName in $allUsers; do
```

```
    aws iam create-user \  
        --user-name $userName
```

```
done
```

```
# list all users
```

```
aws iam list-users --no-paginate
```

```
# get a specific user's info
```

```
aws iam get-user \  
    --user-name aws-admin2
```

```
# delete one user
```

```
aws iam delete-user \  
    --user-name aws-admin2
```

```
# delete all users
# allUsers=$(aws iam list-users --output text | cut -f 6);
allUsers=$(cat ./user-names.txt)
for userName in $allUsers; do
    aws iam delete-user \
        --user-name $userName
done
```

## Política de contraseñas

<http://docs.aws.amazon.com/cli/latest/reference/iam/>

```
# list policy
# http://docs.aws.amazon.com/cli/latest/reference/iam/get-account-password-policy.html
aws iam get-account-password-policy
```

```
# set policy
# http://docs.aws.amazon.com/cli/latest/reference/iam/update-account-password-policy.html
aws iam update-account-password-policy \
    --minimum-password-length 12 \
    --require-symbols \
    --require-numbers \
    --require-uppercase-characters \
    --require-lowercase-characters \
    --allow-users-to-change-password
```

```
# delete policy
# http://docs.aws.amazon.com/cli/latest/reference/iam/delete-account-password-policy.html
aws iam delete-account-password-policy
```

## Teclas de acceso

<http://docs.aws.amazon.com/cli/latest/reference/iam/>

```
# list all access keys
aws iam list-access-keys
```

```
# list access keys of a specific user
aws iam list-access-keys \
    --user-name aws-admin2
```

```
# create a new access key
aws iam create-access-key \
    --user-name aws-admin2 \
    --output text | tee aws-admin2.txt
```

```
# list last access time of an access key
aws iam get-access-key-last-used \
```



```
--access-key-id AKIAINA6AJZY4EXAMPLE
```

```
# deactivate an access key
```

```
aws iam update-access-key \  
  --access-key-id AKIAI44QH8DHBEXAMPLE \  
  --status Inactive \  
  --user-name aws-admin2
```

```
# delete an access key
```

```
aws iam delete-access-key \  
  --access-key-id AKIAI44QH8DHBEXAMPLE \  
  --user-name aws-admin2
```

## **Grupos, Políticas, Políticas Gestionadas.**

[http://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_roles.html](http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html)

<http://docs.aws.amazon.com/cli/latest/reference/iam/>

```
# list all groups
```

```
aws iam list-groups
```

```
# create a group
```

```
aws iam create-group --group-name FullAdmins
```

```
# delete a group
```

```
aws iam delete-group \  
  --group-name FullAdmins
```

```
# list all policies
```

```
aws iam list-policies
```

```
# get a specific policy
```

```
aws iam get-policy \  
  --policy-arn <value>
```

```
# list all users, groups, and roles, for a given policy
```

```
aws iam list-entities-for-policy \  
  --policy-arn <value>
```

```
# list policies, for a given group
```

```
aws iam list-attached-group-policies \  
  --group-name FullAdmins
```

```
# add a policy to a group
```

```
aws iam attach-group-policy \  
  --group-name FullAdmins \  
  --policy-arn arn:aws:iam::aws:policy/AdministratorAccess
```

```
# add a user to a group
aws iam add-user-to-group \
  --group-name FullAdmins \
  --user-name aws-admin2
```

```
# list users, for a given group
aws iam get-group \
  --group-name FullAdmins
```

```
# list groups, for a given user
aws iam list-groups-for-user \
  --user-name aws-admin2
```

```
# remove a user from a group
aws iam remove-user-from-group \
  --group-name FullAdmins \
  --user-name aws-admin2
```

```
# remove a policy from a group
aws iam detach-group-policy \
  --group-name FullAdmins \
  --policy-arn arn:aws:iam::aws:policy/AdministratorAccess
```

```
# delete a group
aws iam delete-group \
  --group-name FullAdmins
```

# EC2

## par de llaves

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html>

```
# list all keypairs
# http://docs.aws.amazon.com/cli/latest/reference/ec2/describe-key-pairs.html
aws ec2 describe-key-pairs
```

```
# create a keypair
# http://docs.aws.amazon.com/cli/latest/reference/ec2/create-key-pair.html
aws ec2 create-key-pair \
    --key-name <value>
```

```
# create a new private / public keypair, using RSA 2048-bit
ssh-keygen -t rsa -b 2048
```

```
# import an existing keypair
# http://docs.aws.amazon.com/cli/latest/reference/ec2/import-key-pair.html
aws ec2 import-key-pair \
    --key-name keyname_test \
    --public-key-material file:///home/apollo/id_rsa.pub
```

```
# delete a keypair
# http://docs.aws.amazon.com/cli/latest/reference/ec2/delete-key-pair.html
aws ec2 delete-key-pair \
    --key-name <value>
```

## Grupos de seguridad

<http://docs.aws.amazon.com/cli/latest/reference/ec2/index.html>

```
# list all security groups
aws ec2 describe-security-groups
```

```
# create a security group
aws ec2 create-security-group \
    --vpc-id vpc-1a2b3c4d \
    --group-name web-access \
    --description "web access"
```

```
# list details about a security group
aws ec2 describe-security-groups \
    --group-id sg-00000000
```

```
# open port 80, for everyone
aws ec2 authorize-security-group-ingress \
```

```

--group-id sg-00000000 \
--protocol tcp \
--port 80 \
--cidr 0.0.0.0/24

# get my public ip
my_ip=$(dig +short myip.opendns.com @resolver1.opendns.com);
echo $my_ip

# open port 22, just for my ip
aws ec2 authorize-security-group-ingress \
  --group-id sg-00000000 \
  --protocol tcp \
  --port 80 \
  --cidr $my_ip/24

# remove a firewall rule from a group
aws ec2 revoke-security-group-ingress \
  --group-id sg-00000000 \
  --protocol tcp \
  --port 80 \
  --cidr 0.0.0.0/24

# delete a security group
aws ec2 delete-security-group \
  --group-id sg-00000000

```

## Instancias

<http://docs.aws.amazon.com/cli/latest/reference/ec2/index.html>

```

# list all instances (running, and not running)
# http://docs.aws.amazon.com/cli/latest/reference/ec2/describe-instances.html
aws ec2 describe-instances

# create a new instance
# http://docs.aws.amazon.com/cli/latest/reference/ec2/run-instances.html
aws ec2 run-instances \
  --image-id ami-f0e7d19a \
  --instance-type t2.micro \
  --security-group-ids sg-00000000 \
  --dry-run

# stop an instance
# http://docs.aws.amazon.com/cli/latest/reference/ec2/terminate-instances.html
aws ec2 terminate-instances \
  --instance-ids <instance_id>

```

```
# list status of all instances
# http://docs.aws.amazon.com/cli/latest/reference/ec2/describe-instance-status.html
aws ec2 describe-instance-status
```

```
# list status of a specific instance
aws ec2 describe-instance-status \
    --instance-ids <instance_id>
```

## Etiquetas

```
# list the tags of an instance
# http://docs.aws.amazon.com/cli/latest/reference/ec2/describe-tags.html
aws ec2 describe-tags
```

```
# add a tag to an instance
# http://docs.aws.amazon.com/cli/latest/reference/ec2/create-tags.html
aws ec2 create-tags \
    --resources "ami-1a2b3c4d" \
    --tags Key=name,Value=debian
```

```
# delete a tag on an instance
# http://docs.aws.amazon.com/cli/latest/reference/ec2/delete-tags.html
aws ec2 delete-tags \
    --resources "ami-1a2b3c4d" \
    --tags Key=Name,Value=
```

# Cloudwatch

## Grupos de registro

<http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/WhatIsCloudWatchLogs.html> <http://docs.aws.amazon.com/cli/latest/reference/logs/index.html#cli-aws-logs>

crear un grupo

<http://docs.aws.amazon.com/cli/latest/reference/logs/create-log-group.html>

```
aws logs create-log-group \  
    --log-group-name "DefaultGroup"
```

listar todos los grupos de registro

<http://docs.aws.amazon.com/cli/latest/reference/logs/describe-log-groups.html>

```
aws logs describe-log-groups
```

```
aws logs describe-log-groups \  
    --log-group-name-prefix "Default"
```

borrar un grupo

<http://docs.aws.amazon.com/cli/latest/reference/logs/delete-log-group.html>

```
aws logs delete-log-group \  
    --log-group-name "DefaultGroup"
```

## Secuencias de registro

# Log group names can be between 1 and 512 characters long. Allowed  
# characters include a-z, A-Z, 0-9, '\_' (underscore), '-' (hyphen),  
# '/' (forward slash), and '.' (period).

# create a log stream

# <http://docs.aws.amazon.com/cli/latest/reference/logs/create-log-stream.html>

```
aws logs create-log-stream \  
    --log-group-name "DefaultGroup" \  
    --log-stream-name "syslog"
```

# list details on a log stream

# <http://docs.aws.amazon.com/cli/latest/reference/logs/describe-log-streams.html>

```
aws logs describe-log-streams \  
    --log-group-name "syslog"
```

```
aws logs describe-log-streams \  
    --log-stream-name-prefix "syslog"
```

```
# delete a log stream
# http://docs.aws.amazon.com/cli/latest/reference/logs/delete-log-stream.html
aws logs delete-log-stream \
    --log-group-name "DefaultGroup" \
    --log-stream-name "Default Stream"
```