



**EBook Gratis**

# APRENDIZAJE

---

## aws-cli

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#aws-cli**

# Tabla de contenido

Acerca de.....	1
<b>Capítulo 1: Empezando con aws-cli.....</b>	<b>2</b>
Observaciones.....	2
<b>Descripción.....</b>	<b>2</b>
<b>Servicios soportados.....</b>	<b>2</b>
<b>Interfaz de línea de comandos de AWS en GitHub.....</b>	<b>2</b>
Versiones.....	2
Examples.....	2
Instalación y configuración.....	3
Creando un nuevo perfil.....	4
Usando los comandos aws cli.....	5
Listar cubos S3.....	5
AWS completer para Ubuntu con Bash.....	5
Hoja de trucos de AWS CLI - Lista de todos los comandos de CLI.....	6
Preparar.....	6
Instalar AWS CLI.....	6
Bash one-liners.....	6
Cloudtrail - Registro y Auditoría.....	7
YO SOY.....	7
Usuarios.....	7
Política de contraseñas.....	8
Teclas de acceso.....	9
Grupos, Políticas, Políticas Gestionadas.....	9
EC2.....	10
par de llaves.....	10
Grupos de seguridad.....	11
Instancias.....	12
Etiquetas.....	12
Cloudwatch.....	13
Grupos de registro.....	13

Secuencias de registro.....	13
<b>Capítulo 2: aws-codecommit para git local.....</b>	<b>15</b>
Observaciones.....	15
Examples.....	15
Configurar Codecommit para la línea de comandos git.....	15
Utilice SourceTree con AWS Codecommit.....	15
<b>Capítulo 3: ec2 describe-imagenes usos.....</b>	<b>19</b>
Examples.....	19
Describa la imagen por nombre AMI.....	19
<b>Capítulo 4: El parámetro --query.....</b>	<b>20</b>
Observaciones.....	20
Examples.....	20
Listado de instancias de una manera fácil de leer.....	20
<b>Creditos.....</b>	<b>22</b>

---

## Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [aws-cli](#)

It is an unofficial and free aws-cli ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official aws-cli.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Capítulo 1: Empezando con aws-cli

## Observaciones

---

## Descripción

La interfaz de línea de comandos (CLI) de AWS es una herramienta unificada para administrar sus servicios de AWS. Con solo una herramienta para descargar y configurar, puede controlar múltiples servicios de AWS desde la línea de comandos y automatizarlos a través de scripts.

AWS CLI presenta un nuevo conjunto de comandos de archivos simples para transferencias de archivos eficientes hacia y desde Amazon S3.

---

## Servicios soportados

Para obtener una lista de los servicios disponibles que puede usar con la interfaz de línea de comandos de AWS, consulte [Servicios disponibles](#) en la Referencia de comandos de la CLI de AWS.

---

## Interfaz de línea de comandos de AWS en GitHub

Puede ver, y bifurcar, el código fuente de la CLI de AWS en GitHub en el proyecto <https://github.com/aws/aws-cli> .

## Versiones

Versión	Fecha de lanzamiento
<a href="#">1.10.38</a>	2016-06-14
<a href="#">1.10.35</a>	2016-06-03
<a href="#">1.10.33</a>	2016-05-25
<a href="#">1.10.30</a>	2016-05-18

## Examples

## Instalación y configuración

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

En Microsoft Windows: use el instalador de MSI. En Linux, OS X o Unix: use pip (un administrador de paquetes para el software Python) o instale manualmente con el instalador incluido.

### 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
```

**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.

### Para instalar AWS CLI utilizando el instalador incluido

Requisitos previos:

- Linux, OS X o Unix
  - Python 2 versión 2.6.5+ o Python 3 versión 3.3+
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
```

## Actualizando la herramienta CLI

Amazon lanza periódicamente nuevas versiones de la herramienta AWS. Si la herramienta se instaló con la herramienta Python Pip, el siguiente comando comprobará el repositorio remoto en busca de actualizaciones y lo aplicará a su sistema local.

```
$ pip install awscli --upgrade
```

## 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 completer 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 `ec2` automáticamente:

```
$ aws ec2
```

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

# Preparar

## Instalar AWS CLI

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"
```

## YO SOY

### Usuarios

<https://blogs.aws.amazon.com/security/post/Tx15CIT22V4J8RP/How-to-rotate-access-keys-for-IAM-users> [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-0000000

# open port 80, for everyone
aws ec2 authorize-security-group-ingress \
    --group-id sg-0000000 \
    --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-0000000 \
    --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 "syslog"
```



```
--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"
```

Lea Empezando con aws-cli en línea: <https://riptutorial.com/es/aws-cli/topic/1947/empezando-con-aws-cli>

# Capítulo 2: aws-codecommit para git local

## Observaciones

Prepárese configurando su máquina de desarrollo local con la [herramienta de línea de comandos aws](#) y el comando `git`.

## Examples

### Configurar Codecommit para la línea de comandos git

AWS Codecommit se puede utilizar como almacenamiento para repositorios GIT privados. La configuración incluye algunos pasos, asumiendo que ya tiene una cuenta de AWS válida.

1. Registrarse en [AWS Codecommit](#). Actualmente solo está disponible la región `us-east-1`.
2. Cree un [usuario IAM](#) que tendrá acceso a los repositorios, por ejemplo, `codecommit-user`
3. Adjunte el rol de permiso `AWSCodeCommitFullAccess` a este usuario
4. Cree una nueva `Access Key` para este usuario y anote la `key id` y el `secret code`
5. Ahora siga adelante y cree un [perfil de configuración de AWS](#) en su máquina local

```
$ aws configure --profile codecommit-user
```

En el siguiente paso, asociamos el comando `aws` con `git` como el ayudante de credenciales con los siguientes comandos:

```
$ git config --global credential.helper \  
  '!aws --profile codecommit-user codecommit credential-helper $@'  
$ git config --global credential.UseHttpPath true
```

Puedes verificar o editar esta configuración después:

```
$ git config --global --edit
```

Debes anotar una sección:

```
[credential]  
  helper = !aws --profile codecommit-user codecommit credential-helper $@  
  UseHttpPath = true
```

Ahora puedes usar `git` desde la línea de comandos como de costumbre.

### Utilice SourceTree con AWS Codecommit

Atlassian [SourceTree](#) es una herramienta visual para Mac y Windows para administrar los repositorios de código fuente. Se puede usar con Codecommit como un repositorio remoto, pero

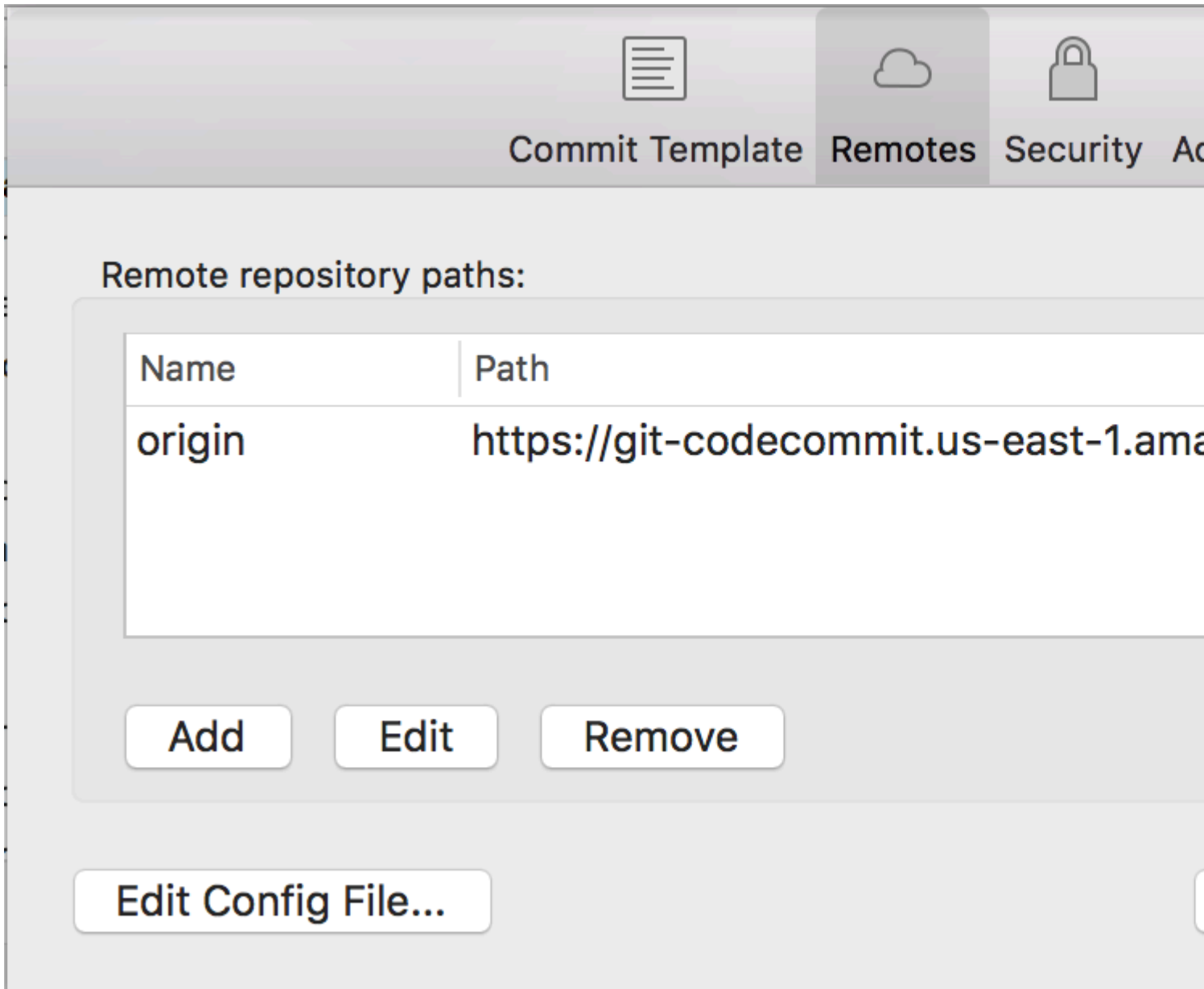
es necesario agregar una opción de configuración adicional al repositorio local en SourceTree para poder conectarse con codecommit.

Primero, configure Codecommit para git local.

Suponiendo que tiene un repositorio de git local que desea enviar al codecommit simplemente siga estos pasos:

1. Inicie sesión en AWS Codecommit utilizando la [consola web](#) .
2. Crear un nuevo repositorio, por ejemplo, `my-project`
3. Copie la URL de HTTPS, debería verse como `https://git-codecommit.us-east-1.amazonaws.com/v1/repos/my-project`
4. Ahora en SourceTree abre el panel Configuraciones / Remotos
5. Agregue un nuevo control remoto con nombre: `origin` y URL / Ruta: el enlace que copió antes
6. Finalmente, abra la opción Editar archivo de configuración y agregue el siguiente fragmento de código:

```
[credential]
  helper = /usr/local/bin/aws --profile codecommit-user codecommit credential-helper $@
  UseHttpPath = true
```



Después de guardar el archivo de configuración debería verse algo como esto:

```
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
  ignorecase = true
  precomposeunicode = true
[branch "master"]
  remote = origin
  merge = refs/heads/master
[remote "origin"]
  url = https://git-codecommit.us-east-1.amazonaws.com/v1/repos/digitaloffice.nu
  fetch = +refs/heads/*:refs/remotes/origin/*
[credential]
  helper = /usr/local/bin/aws --profile codecommit-user codecommit credential-helper $@
  UseHttpPath = true
```

Tenga en cuenta: esto se basa en la configuración de OS-X. Preste especial atención a la ruta de

aws (que es `/usr/local/bin/aws` en este caso) y seguramente será diferente en otras configuraciones de Unixes o Windows.

Lea `aws-codecommit para git local` en línea: <https://riptutorial.com/es/aws-cli/topic/6450/aws-codecommit-para-git-local>

---

# Capítulo 3: ec2 describe-imagenes usos

## Examples

### Describa la imagen por nombre AMI

```
aws ec2 describe-images --filters "Name=name,Values=${NAME_OF_AMI}"
```

Lea ec2 describe-imagenes usos en línea: <https://riptutorial.com/es/aws-cli/topic/9363/ec2-describe-imagenes-usos>

# Capítulo 4: El parámetro --query

## Observaciones

El parámetro `--query` menudo se pasa por alto, pero es increíblemente poderoso. Utiliza el lenguaje de consulta de [JMESPath](#) para filtrar las respuestas del servicio a lo que desea exactamente.

## Examples

### Listado de instancias de una manera fácil de leer

Las instancias tienen una gran cantidad de metadatos que se devuelven de una llamada a `describe-instances`, pero muchas veces solo quieres ver lo básico. Puede utilizar una consulta JMESPath combinada con la salida de la tabla para mostrar información de instancia concisa de una manera fácil de leer.

```
aws ec2 describe-instances --output table --query "Reservations[].Instances[].[Name: Tags[?Key == 'Name'].Value | [0], Id: InstanceId, State: State.Name, Type: InstanceType]"
```

```
-----  
|                               DescribeInstances                               |  
+-----+-----+-----+-----+  
|   Id   |   Name   |   State   |   Type   |  
+-----+-----+-----+-----+  
| i-abc123 | None     | stopped   | m3.large |  
| i-def456 | amazon linux | stopped   | t2.micro |  
| i-ghi789 | proxy    | running   | t2.micro |  
+-----+-----+-----+-----+
```

Ahora vamos a romper esa pieza por pieza. Primero, tenemos `--output table`. Esto produce una representación de tabla coloreada de la respuesta. En general, esto es más útil con comandos que devuelven pequeños conjuntos de datos o donde se han filtrado los datos.

Ahora en el `--query`. Este parece largo, pero en realidad es bastante simple. La primera parte es `Reservations[].Instances[]`. Esto devuelve una lista aplanada de todas las instancias devueltas.

La siguiente parte de la consulta se encapsula con `.{}`. Lo que está haciendo es crear un nuevo objeto json para cada elemento de la lista donde cada valor es una consulta JMESPath que se aplicará al objeto de origen (en este caso, una instancia). La mayoría de estos son muy simples, pero el `Name` es un poco más complejo.

La consulta completa para obtener `Name` es `Tags[?Key == 'Name'].Value | [0]`. La primera parte de eso, `Tags[?Key == 'Name']` es buscar en las etiquetas de la instancia una etiqueta cuya clave sea `Name`. La segunda mitad `.Value | [0]` es seleccionar los valores de cada una de esas etiquetas y luego tomar el primer elemento de la lista (en este caso, solo habrá uno).

Exactamente lo que quieres en esa mesa depende completamente de ti. Si quisiera agregar información de DNS, por ejemplo, podría agregar fácilmente una nueva clave `DNS: PublicDnsName :`

```
aws ec2 describe-instances --output table --query "Reservations[].Instances[].[Name: Tags[?Key == 'Name'].Value | [0], Id: InstanceId, State: State.Name, Type: InstanceType, DNS: PublicDnsName]"
```

```
-----  
-----  
|                                     DescribeInstances  
|  
+-----+-----+-----+-----+-----+  
|                                     | Id      | Name      | State     |  
Type      | DNS                                     |  
+-----+-----+-----+-----+-----+  
|                                     | i-abc123 | None      | stopped   |  
m3.large  |  
|                                     | i-def456 | amazon linux | stopped   |  
t2.micro  |  
| ec2-192-168-1-1.us-west-2.compute.amazonaws.com | i-ghi789 | proxy     | running   |  
t2.micro  |  
+-----+-----+-----+-----+-----+  
-----+
```

Lea El parámetro `--query` en línea: <https://riptutorial.com/es/aws-cli/topic/7306/el-parametro---query>



---

# Creditos

S. No	Capítulos	Contributors
1	Empezando con aws-cli	<a href="#">chenchuk</a> , <a href="#">Community</a> , <a href="#">Danny</a> , <a href="#">Esteban</a> , <a href="#">Nithin K Anil</a> , <a href="#">Paddez</a> , <a href="#">richardboydii</a> , <a href="#">Scroff</a> , <a href="#">Yaron Idan</a>
2	aws-codecommit para git local	<a href="#">jlapoutre</a>
3	ec2 describe-imagenes usos	<a href="#">Yuki Inoue</a>
4	El parámetro --query	<a href="#">Jordon Phillips</a>