



**EBook Gratuito**

# APPENDIMENTO

## aws-cli

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

**#aws-cli**

# Sommario

Di.....	1
<b>Capitolo 1: Iniziare con aws-cli.....</b>	<b>2</b>
Osservazioni.....	2
<b>Descrizione.....</b>	<b>2</b>
<b>Servizi supportati.....</b>	<b>2</b>
<b>Interfaccia della riga di comando AWS su GitHub.....</b>	<b>2</b>
Versioni.....	2
Examples.....	2
Installazione e configurazione.....	2
Creare un nuovo profilo.....	4
Usando i comandi cli di aws.....	5
Elenca i bucket S3.....	5
Completamento AWS per Ubuntu con Bash.....	5
AWS CLI Cheat sheet - Elenco di tutti i comandi CLI.....	6
Impostare.....	6
Installa CLI AWS.....	6
Bash one-liner.....	6
Cloudtrail - Registrazione e controllo.....	7
SONO.....	7
utenti.....	7
Politica password.....	8
Chiavi di accesso.....	9
Gruppi, politiche, politiche gestite.....	9
EC2.....	10
coppie di chiavi.....	10
Gruppi di sicurezza.....	11
istanze.....	12
tag.....	12
CloudWatch.....	13
Registra gruppi.....	13

Log Stream.....	13
<b>Capitolo 2: aws-codecommit per git locale.....</b>	<b>15</b>
Osservazioni.....	15
Examples.....	15
Codecommit di installazione per la riga di comando git.....	15
Usa SourceTree con AWS Codecommit.....	15
<b>Capitolo 3: ec2 Descrivi-immagini usi.....</b>	<b>19</b>
Examples.....	19
Descrivi l'immagine con il nome AMI.....	19
<b>Capitolo 4: Il parametro --query.....</b>	<b>20</b>
Osservazioni.....	20
Examples.....	20
Elenco delle istanze in un modo facile da leggere.....	20
<b>Titoli di coda.....</b>	<b>22</b>

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)

---

# Capitolo 1: Iniziare con aws-cli

## Osservazioni

---

### Descrizione

L'interfaccia a riga di comando (CLI) di AWS è uno strumento unificato per gestire i tuoi servizi AWS. Con un solo strumento da scaricare e configurare, puoi controllare più servizi AWS dalla riga di comando e automatizzarli tramite script.

La CLI di AWS introduce un nuovo set di semplici comandi file per trasferimenti di file efficienti da e verso Amazon S3.

---

### Servizi supportati

Per un elenco dei servizi disponibili che è possibile utilizzare con l'interfaccia della riga di comando di AWS, vedere [Servizi disponibili](#) nella Guida ai comandi CLI di AWS.

---

## Interfaccia della riga di comando AWS su GitHub

Puoi visualizzare e inserire il codice sorgente per la CLI di AWS su GitHub nel progetto <https://github.com/aws/aws-cli>.

## Versioni

Versione	Data di rilascio
<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

### Installazione e configurazione

Esistono diversi modi per installare la CLI AWS sulla tua macchina, a seconda del sistema operativo e dell'ambiente in uso:

Su Microsoft Windows: utilizzare il programma di installazione MSI. Su Linux, OS X o Unix - usa pip (un gestore di pacchetti per il software Python) o installa manualmente con il programma di installazione in bundle.

### Installa utilizzando pip:

Avrai bisogno di python per essere installato (versione 2, 2.6.5 +, 3 o 3.3+). Controllare con

```
python --version  
  
pip --help
```

Dato che entrambi sono installati, utilizzare il seguente comando per installare aws cli.

```
sudo pip install awscli
```

**Installa su Windows** La CLI AWS è supportata su Microsoft Windows XP o versioni successive. Per gli utenti Windows, il pacchetto di installazione MSI offre un modo pratico e familiare per installare la CLI di AWS senza installare altri prerequisiti. Gli utenti Windows dovrebbero utilizzare il programma di installazione MSI a meno che non stiano già utilizzando pip per la gestione dei pacchetti.

- [MSI Installer per Windows a 32 bit](#)
- [MSI Installer per Windows a 64 bit](#)

Eseguire il programma di installazione MSI scaricato. Segui le istruzioni che appaiono.

### Per installare la CLI di AWS utilizzando il programma di installazione in bundle

Prerequisiti:

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

1. Scarica il programma di installazione in bundle CLA AWS usando wget o curl.
2. Decomprimere il pacchetto.
3. Esegui l'eseguibile di installazione.

Su Linux e OS X, ecco i tre comandi che corrispondono a ciascun passo:

```
$ 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
```

### Installa utilizzando HomeBrew su OS X:

## Un'altra opzione per OS X

```
brew install awscli
```

## Testare l'installazione della CLI di AWS

Verificare che la CLI sia installata correttamente visualizzando il file della guida. Aprire un terminale, shell o prompt dei comandi, immettere `aws help` e premere Invio:

```
$ aws help
```

## Configurazione della CLI AWS

Una volta completata l'installazione, è necessario configurarla. Avrai bisogno della chiave di accesso e della chiave segreta che ottieni quando crei il tuo account su aws. È inoltre possibile specificare un [nome](#) regione predefinito e un tipo di output predefinito (`text` | `table` | `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
```

## Aggiornamento dello strumento CLI

Amazon rilascia periodicamente nuove versioni di AWS Tool. Se lo strumento è stato installato utilizzando lo strumento Python Pip, il seguente comando controllerà il repository remoto per gli aggiornamenti e lo applicherà al sistema locale.

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

## Creare un nuovo profilo

Per impostare un nuovo profilo credenziale con il nome `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
```

Per l'ID della chiave di accesso AWS e il segreto, crea un utente IAM nella console AWS e genera le chiavi per esso.

Region sarà la regione predefinita per i comandi nel formato `eu-west-1` o `us-east-1` .

Il formato di output predefinito può essere `text` , `table` o `json` .

Ora puoi usare il nome del profilo in altri comandi usando l'opzione `--profile` , ad esempio:

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

Le librerie AWS per altre lingue (ad es. `aws-sdk` per Ruby o `boto3` per Python) hanno opzioni per usare il profilo che crei anche con questo metodo. Ad esempio, la creazione di una nuova sessione in `boto3` può essere fatta in questo modo, `boto3.Session(profile_name:'myprofile')` e utilizzerà le credenziali create per il profilo.

I dettagli della tua configurazione di `aws-cli` possono essere trovati in `~/.aws/config` e `~/.aws/credentials` (su linux e mac-os). Questi dettagli possono essere modificati manualmente da lì.

## Usando i comandi cli di aws

La sintassi per l'utilizzo di `aws cli` è la seguente:

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

Alcuni esempi che utilizzano il comando 'ec2' e il sottocomando 'describe-instances':

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

Esempio con un falso id:

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

## Elenca i bucket S3

```
aws s3 ls
```

Usa un profilo con nome

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

Elenca tutti gli oggetti in un bucket, inclusi gli oggetti nelle cartelle, con le dimensioni in formato leggibile dall'utente e un riepilogo delle proprietà dei bucket alla fine -

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

## Completamento AWS per Ubuntu con Bash

La seguente utility può essere utilizzata per il completamento automatico dei comandi:

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



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

Per le future sessioni di shell, aggiungi questo al tuo `~ / .bashrc`

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

Per verificare, digitare:

```
$ aws ec
```

Premi il tasto [TAB], dovrebbe aggiungere `ec2` automaticamente:

```
$ aws ec2
```

## AWS CLI Cheat sheet - Elenco di tutti i comandi CLI

# Impostare

## Installa CLI AWS

AWS CLI è uno strumento CLI comune per la gestione delle risorse AWS. Con questo singolo strumento possiamo gestire tutte le risorse di aws

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

## Bash one-liner

```
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 - Registrazione e controllo

<http://docs.aws.amazon.com/cli/latest/reference/cloudtrail/> 5 tracce totali, con supporto per le autorizzazioni a livello di risorsa

```
# 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"
```

**SONO**

**utenti**

<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) Limits = 5000 utenti, 100 gruppi, 250 ruoli, 2 chiavi di accesso / utente

<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
```

## Politica password

<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
```

## Chiavi di accesso

<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
```

## Gruppi, politiche, politiche gestite

[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

### coppie di chiavi

<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>
```

## Gruppi di sicurezza

<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-0000000 \
  --protocol tcp \
  --port 80 \
  --cidr 0.0.0.0/24

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

## istanze

<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>
```

## tag

```
# 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

### Registra gruppi

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

crea un gruppo

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

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

elenca tutti i gruppi di log

<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"
```

cancella un gruppo

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

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

## Log Stream

```
# 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"
```

Leggi Iniziare con aws-cli online: <https://riptutorial.com/it/aws-cli/topic/1947/iniziare-con-aws-cli>

---

# Capitolo 2: aws-codecommit per git locale

## Osservazioni

Preparati impostando il tuo computer di sviluppo locale con lo [strumento da riga di comando aws](#) e il comando [git](#).

## Examples

### Codecommit di installazione per la riga di comando git

AWS Codecommit può essere utilizzato come spazio di archiviazione per repository GIT privati. L'installazione prevede alcuni passaggi, presupponendo che tu abbia già un account AWS valido.

1. Registrati per [AWS Codecommit](#). Attualmente è disponibile solo la regione `us-east-1`.
2. Creare un [utente IAM](#) che avrà accesso ai repository, ad es. `codecommit-user`
3. Allegare il ruolo di autorizzazione `AWSCodeCommitFullAccess` a questo utente
4. Crea una nuova `Access Key` per questo utente e nota l' `key id` e il `secret code`
5. Ora vai avanti e crea un [profilo di configurazione AWS](#) sul tuo computer locale

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

Nel passaggio successivo associamo il comando `aws` con `git` come l'helper delle credenziali con i seguenti comandi:

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

Puoi verificare o modificare questa configurazione in seguito:

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

Dovresti notare una sezione:

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

Ora puoi usare `git` dalla riga di comando come al solito.

### Usa SourceTree con AWS Codecommit

Atlassian [SourceTree](#) è uno strumento visivo per Mac e Windows per gestire gli archivi di codice sorgente. Può essere utilizzato con Codecommit come repository remoto, ma è necessario

aggiungere un'opzione di configurazione aggiuntiva al repository locale in SourceTree per poter connettersi con codecommit.

Innanzitutto, configura Codecommit per git locale.

Supponendo che tu abbia un repository `git` locale che vuoi `codecommit` a `codecommit` segui questi passaggi:

1. Accedi ad AWS Codecommit usando la [console web](#) .
2. Crea un nuovo repository, ad es. `my-project`
3. Copia l'URL HTTPS, dovrebbe apparire come `https://git-codecommit.us-east-1.amazonaws.com/v1/repos/my-project`
4. Ora in SourceTree apri il pannello Impostazioni / Remoti
5. Aggiungi un nuovo telecomando con nome: `origin` e Url / Percorso: il link che hai copiato prima
6. Infine apri l'opzione Modifica file di configurazione e aggiungi il seguente frammento:

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



Commit Template



Remotes



Security Ad

### Remote repository paths:

Name	Path
origin	https://git-codecommit.us-east-1.ama

Add

Edit

Remove

Edit Config File...

Dopo aver salvato il file di configurazione dovrebbe apparire qualcosa del genere:

```
[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
```

Nota: questo è basato sull'impostazione di OS-X. Presta particolare attenzione al percorso di aws

(che è `/usr/local/bin/aws` in questo caso) e sarà sicuramente diverso sotto altre configurazioni di Unix o di Windows.

Leggi `aws-codecommit per git locale` online: <https://riptutorial.com/it/aws-cli/topic/6450/aws-codecommit-per-git-locale>

---

# Capitolo 3: ec2 Descrivi-immagini usi

## Examples

### Descrivi l'immagine con il nome AMI

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

Leggi ec2 Descrivi-immagini usi online: <https://riptutorial.com/it/aws-cli/topic/9363/ec2-descrivi-immagini-usi>

# Capitolo 4: Il parametro `--query`

## Osservazioni

Il parametro `--query` viene spesso trascurato, ma è incredibilmente potente. Utilizza il [linguaggio di query JMESPath](#) per filtrare le risposte del servizio fino a esattamente ciò che si desidera.

## Examples

### Elenco delle istanze in un modo facile da leggere

Le istanze hanno un sacco di metadati che vengono restituiti da una chiamata per `describe-instances`, ma spesso le volte si vuole solo vedere le basi. È possibile utilizzare una query JMESPath combinata con l'output della tabella per mostrare le informazioni dell'istanza concise in modo facilmente leggibile.

```
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 |  
+-----+-----+-----+-----+
```

Ora lasciamoci sfondare pezzo per pezzo. Innanzitutto, abbiamo la `--output table`. Questo produce una rappresentazione tabella colorata della risposta. Questo è generalmente più utile con i comandi che restituiscono piccoli insiemi di dati o dove sono stati filtrati i dati.

Ora sul `--query`. Questo sembra lungo, ma in realtà è abbastanza semplice. La prima parte è `Reservations[].Instances[]`. Questo restituisce un elenco appiattito di tutte le istanze restituite.

La parte successiva della query è incapsulata con `.{}`. Ciò che sta facendo è creare un nuovo oggetto json per ogni elemento nell'elenco in cui ogni valore è una query JMESPath da applicare all'oggetto di origine (in questo caso, un'istanza). Molti di questi sono molto semplici, ma il `Name` è un po' più complesso.

La query completa per ottenere il `Name` è `Tags[?Key == 'Name'].Value | [0]`. La prima parte di questo, `Tags[?Key == 'Name']` sta cercando i tag dell'istanza per un tag la cui chiave è `Name`. La seconda metà `.Value | [0]` sta selezionando i valori di ciascuno di questi tag e poi prendendo il primo elemento dall'elenco (in questo caso, ci sarà sempre e solo uno).

Esattamente ciò che vuoi in quel tavolo dipende completamente da te. Se si desidera aggiungere

informazioni sul DNS, ad esempio, è possibile aggiungere facilmente una nuova chiave `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  
|  
+-----+-----+-----+-----+-----+  
-----+  
|           DNS           |   Id   |   Name   |   State   |  
Type |  
+-----+-----+-----+-----+-----+  
-----+  
|           |   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 |  
+-----+-----+-----+-----+-----+  
-----+
```

Leggi il parametro `--query` online: <https://riptutorial.com/it/aws-cli/topic/7306/il-parametro---query>



## Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare 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 per git locale	<a href="#">jlapoutre</a>
3	ec2 Descrivi-immagini usi	<a href="#">Yuki Inoue</a>
4	Il parametro --query	<a href="#">Jordon Phillips</a>