

 eBook Gratuit

APPRENEZ

aws-cli

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#aws-cli

Table des matières

À propos.....	1
Chapitre 1: Démarrer avec aws-cli.....	2
Remarques.....	2
La description.....	2
Services pris en charge.....	2
Interface de ligne de commande AWS sur GitHub.....	2
Versions.....	2
Exemples.....	2
Installation et configuration.....	2
Créer un nouveau profil.....	4
Utiliser les commandes aws cli.....	5
Liste des seaux S3.....	5
Complément AWS pour Ubuntu avec Bash.....	5
Feuille de triche AWS CLI - Liste de toutes les commandes CLI.....	6
Installer.....	6
Installer AWS CLI.....	6
Bash one-liners.....	6
Cloudtrail - Journalisation et audit.....	7
JE SUIS.....	7
Utilisateurs.....	8
Politique de mot de passe.....	8
Touches d'accès.....	9
Groupes, politiques, stratégies gérées.....	9
EC2.....	10
keypairs.....	11
Groupes de sécurité.....	11
Instances.....	12
Mots clés.....	12
Cloudwatch.....	13
Groupes de journaux.....	13

Journaux de journalisation.....	13
Chapitre 2: aws-codecommit pour git local.....	15
Remarques.....	15
Exemples.....	15
Configurer Codecommit pour la ligne de commande git.....	15
Utilisez SourceTree avec AWS Codecommit.....	15
Chapitre 3: ec2 décrit les utilisations des images.....	19
Exemples.....	19
Décrire l'image par nom AMI.....	19
Chapitre 4: Le paramètre --query.....	20
Remarques.....	20
Exemples.....	20
Liste des instances d'une manière facile à lire.....	20
Crédits.....	22

À propos

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

Chapitre 1: Démarrer avec aws-cli

Remarques

La description

L'interface de ligne de commande AWS (CLI) est un outil unifié pour gérer vos services AWS. Avec un seul outil à télécharger et à configurer, vous pouvez contrôler plusieurs services AWS à partir de la ligne de commande et les automatiser via des scripts.

AWS CLI introduit un nouvel ensemble de commandes de fichier simples pour des transferts de fichiers efficaces depuis et vers Amazon S3.

Services pris en charge

Pour obtenir la liste des services disponibles que vous pouvez utiliser avec l'interface de ligne de commande AWS, voir [Services disponibles](#) dans la référence de commande AWS CLI.

Interface de ligne de commande AWS sur GitHub

Vous pouvez afficher - et fork - le code source de l'AWS CLI sur GitHub dans le projet <https://github.com/aws/aws-cli>.

Versions

Version	Date de sortie
1.10.38	2016-06-14
1.10.35	2016-06-03
1.10.33	2016-05-25
1.10.30	2016-05-18

Exemples

Installation et configuration

Il existe différentes manières d'installer l'AWS CLI sur votre ordinateur, en fonction du système d'exploitation et de l'environnement que vous utilisez:

Sous Microsoft Windows, utilisez le programme d'installation MSI. Sous Linux, OS X ou Unix, utilisez pip (un gestionnaire de paquets pour le logiciel Python) ou installez-le manuellement avec le programme d'installation fourni.

Installez en utilisant pip:

Vous aurez besoin de python pour être installé (version 2, 2.6.5 +, 3 ou 3.3+). Vérifier avec

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

Étant donné que les deux sont installés, utilisez la commande suivante pour installer le aws cli.

```
sudo pip install awscli
```

Installer sur Windows AWS CLI est pris en charge sur Microsoft Windows XP ou version ultérieure. Pour les utilisateurs Windows, le package d'installation MSI offre un moyen familier et pratique d'installer l'AWS CLI sans installer d'autres prérequis. Les utilisateurs Windows doivent utiliser le programme d'installation MSI, sauf s'ils utilisent déjà pip pour la gestion des packages.

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

Exécutez le programme d'installation MSI téléchargé. Suivez les instructions qui apparaissent.

Pour installer l'AWS CLI à l'aide du programme d'installation fourni

Conditions préalables:

- Linux, OS X ou Unix
- Python 2 version 2.6.5+ ou Python 3 version 3.3+

1. Téléchargez le programme d'installation fourni avec AWS CLI en utilisant wget ou curl.
2. Décompressez le paquet.
3. Exécutez l'exécutable d'installation.

Sous Linux et OS X, voici les trois commandes correspondant à chaque étape:

```
$ 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
```

Installer à l'aide de HomeBrew sous OS X:

Une autre option pour OS X

```
brew install awscli
```

Test de l'installation de l'AWS CLI

Vérifiez que la CLI est installée correctement en affichant le fichier d'aide. Ouvrez un terminal, un shell ou une invite de commande, entrez l'aide de aws et appuyez sur Entrée:

```
$ aws help
```

Configuration de l'AWS CLI

Une fois l'installation terminée, vous devez la configurer. Vous aurez besoin de votre clé d'accès et de votre clé secrète lorsque vous créez votre compte sur aws. Vous pouvez également spécifier un [nom de région](#) par défaut et un type de sortie par défaut (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
```

Mise à jour de l'outil CLI

Amazon publie périodiquement de nouvelles versions de l'outil AWS. Si l'outil a été installé à l'aide de l'outil Python Pip, la commande suivante vérifie les mises à jour du référentiel distant et les applique à votre système local.

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

Créer un nouveau profil

Pour configurer un nouveau profil d'informations d'identification avec le nom `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
```

Pour l'identifiant et le secret de la clé d'accès AWS, créez un utilisateur IAM dans la console AWS et générez les clés correspondantes.

Region sera la région par défaut pour les commandes au format `eu-west-1` OU `us-east-1` .

Le format de sortie par défaut peut être `text` , `table` OU `json` .

Vous pouvez maintenant utiliser le nom du profil dans d'autres commandes en utilisant l'option `--profile` , par exemple:

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

Les bibliothèques AWS pour d'autres langages (par exemple, `aws-sdk` pour Ruby ou `boto3` pour Python) ont également des options pour utiliser le profil que vous créez avec cette méthode. Par exemple, créer une nouvelle session dans `boto3` peut se faire comme ceci,

`boto3.Session(profile_name:'myprofile')` et il utilisera les informations d'identification que vous avez créées pour le profil.

Les détails de votre configuration `aws-cli` peuvent être trouvés dans `~/.aws/config` et `~/.aws/credentials` (sur Linux et Mac-OS). Ces détails peuvent être modifiés manuellement à partir de là.

Utiliser les commandes aws cli

La syntaxe pour utiliser le `aws cli` est la suivante:

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

Quelques exemples d'utilisation de la commande 'ec2' et de la sous-commande 'describe-instances':

```
aws ec2 describe-instances
```

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

Exemple avec un faux identifiant:

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

Liste des seaux S3

```
aws s3 ls
```

Utiliser un profil nommé

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

Liste tous les objets d'un compartiment, y compris les objets dans les dossiers, avec une taille dans un format lisible par l'homme et un résumé des propriétés des compartiments à la fin -

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

Complément AWS pour Ubuntu avec Bash

L'utilitaire suivant peut être utilisé pour compléter automatiquement les commandes:


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

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

Pour les futures sessions shell, pensez à ajouter ceci à votre `~ / .bashrc`

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

Pour vérifier, tapez:

```
$ aws ec
```

Appuyez sur la touche [TAB], cela devrait ajouter `ec2` automatiquement:

```
$ aws ec2
```

Feuille de triche AWS CLI - Liste de toutes les commandes CLI

Installer

Installer AWS CLI

AWS CLI est un outil CLI commun pour la gestion des ressources AWS. Avec cet outil unique, nous pouvons gérer toutes les ressources 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 - Journalisation et audit

<http://docs.aws.amazon.com/cli/latest/reference/cloudtrail/> 5 Trails au total, avec prise en charge des autorisations au niveau des ressources

```
# 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"
```

JE SUIS

Utilisateurs

<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 Limites = 5000 utilisateurs, 100 groupes, 250 rôles, 2 clés d'accès / utilisateur

<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
```

Politique de mot de passe

<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
```

Touches d'accès

<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
```

Groupes, politiques, stratégies gérées

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

keypairs

<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>
```

Groupes de sécurité

<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

```

Instances

<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>

```

Mots clés

```

# 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

Groupes de journaux

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

créer un groupe

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

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

liste tous les groupes de journaux

<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"
```

supprimer un groupe

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

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

Journaux de journalisation

```
# 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"
```

Lire Démarrer avec aws-cli en ligne: <https://riptutorial.com/fr/aws-cli/topic/1947/demarrer-avec-aws-cli>

Chapitre 2: aws-codecommit pour git local

Remarques

Préparez-vous en configurant votre machine de développement local avec l' [outil de ligne de commande aws](#) et la commande [git](#) .

Exemples

Configurer Codecommit pour la ligne de commande git

AWS Codecommit peut être utilisé comme stockage pour les référentiels GIT privés. La configuration implique quelques étapes, en supposant que vous avez déjà un compte AWS valide.

1. Inscrivez-vous à [AWS Codecommit](#) . Actuellement, seule la région `us-east-1` est disponible.
2. Créer un [utilisateur IAM](#) qui aura accès aux référentiels, par exemple, `codecommit-user`
3. Attachez le rôle d'autorisation `AWSCodeCommitFullAccess` à cet utilisateur
4. Créer une nouvelle `Access Key` pour cet utilisateur et noter l' `key id` et le `secret code`
5. Maintenant, allez-y et créez un [profil de configuration AWS](#) sur votre ordinateur local

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

Dans l'étape suivante, nous associons la commande `aws` à `git` tant qu'assistant d'informations d'identification avec les commandes suivantes:

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

Vous pouvez vérifier ou modifier cette configuration par la suite:

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

Vous devriez noter une section:

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

Maintenant, vous pouvez utiliser `git` depuis la ligne de commande comme d'habitude.

Utilisez SourceTree avec AWS Codecommit

Atlassian [SourceTree](#) est un outil visuel pour Mac et Windows pour gérer les référentiels de code source. Cela peut être utilisé avec Codecommit en tant que référentiel distant, mais vous devez

ajouter une option de configuration supplémentaire au référentiel local dans SourceTree pour pouvoir vous connecter avec codecommit.

Commencez par configurer Codecommit pour git local.

En supposant que vous ayez un référentiel `git` local que vous souhaitez `codecommit` à `codecommit` suivez simplement les étapes suivantes:

1. Connectez-vous à AWS Codecommit à l'aide de la [console Web](#) .
2. Créer un nouveau référentiel, par exemple `my-project`
3. Copiez l'URL HTTPS, il devrait ressembler à `https://git-codecommit.us-east-1.amazonaws.com/v1/repos/my-project`
4. Maintenant, dans SourceTree, ouvrez le panneau Paramètres / Télécommandes
5. Ajouter une nouvelle télécommande avec le nom: `origin` et Url / Path: le lien que vous avez copié auparavant
6. Enfin, ouvrez l'option Edit Config File et ajoutez l'extrait suivant:

```
[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.amaz

Add

Edit

Remove

Edit Config File...

Après avoir enregistré le fichier de configuration devrait ressembler à ceci:

```
[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
```

Veuillez noter que ceci est basé sur la configuration OS-X. Faites particulièrement attention au

chemin pour aws (qui est `/usr/local/bin/aws` dans ce cas) et sera très certainement différent sous d'autres configurations Unix ou Windows.

Lire `aws-codecommit` pour git local en ligne: <https://riptutorial.com/fr/aws-cli/topic/6450/aws-codecommit-pour-git-local>

Chapitre 3: ec2 décrit les utilisations des images

Exemples

Décrire l'image par nom AMI

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

Lire ec2 décrit les utilisations des images en ligne: <https://riptutorial.com/fr/aws-cli/topic/9363/ec2-decrit-les-utilisations-des-images>

Chapitre 4: Le paramètre --query

Remarques

Le paramètre `--query` est souvent négligé, mais il est incroyablement puissant. Il utilise le langage de requête [JMESPath](#) pour filtrer précisément les réponses aux demandes .

Exemples

Liste des instances d'une manière facile à lire

Les instances ont beaucoup de métadonnées renvoyées par un appel à `describe-instances` , mais souvent, vous ne voulez que voir les bases. Vous pouvez utiliser une requête JMESPath combinée à une sortie de table pour afficher des informations d'instance concises de manière facilement lisible.

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

Maintenant, laissez cette pièce pièce par pièce. Tout d'abord, nous avons `--output table` . Cela produit une représentation de table colorisée de la réponse. Ceci est généralement plus utile avec les commandes qui renvoient de petits ensembles de données ou dans lesquelles vous avez filtré les données.

Maintenant sur le `--query` . Celui-ci a l'air long, mais c'est en fait assez simple. La première partie est `Reservations[].Instances[]` . Cela retourne une liste aplatie de toutes les instances renvoyées.

La partie suivante de la requête est encapsulée avec `.{}` . Cela consiste à créer un nouvel objet json pour chaque élément de la liste où chaque valeur est une requête JMESPath à appliquer à l'objet source (dans ce cas, une instance). La plupart sont très simples, mais `Name` est un peu plus complexe.

La requête complète pour obtenir le `Name` est `Tags[?Key == 'Name'].Value | [0]` . La première partie, `Tags[?Key == 'Name']` recherche les balises de l'instance pour une balise dont la clé est `Name` . La seconde moitié `.Value | [0]` sélectionne les valeurs de chacune de ces balises et prend ensuite le premier élément de la liste (dans ce cas, il n'y en aura qu'une).

Exactement ce que vous voulez dans cette table est entièrement à vous. Si vous souhaitez ajouter des informations DNS, par exemple, vous pouvez facilement ajouter une nouvelle clé `DNS`:

`PublicDnsName` :

```
aws ec2 describe-instances --output table --query "Reservations[].Instances[0].{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 |  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
-----+
```

Lire Le paramètre `--query` en ligne: <https://riptutorial.com/fr/aws-cli/topic/7306/le-parametre---query>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec aws-cli	chenchuk , Community , Danny , Esteban , Nithin K Anil , Paddez , richardboydii , Scroff , Yaron Idan
2	aws-codecommit pour git local	jlapoutre
3	ec2 décrit les utilisations des images	Yuki Inoue
4	Le paramètre --query	Jordon Phillips