

AWS CLI

🕒 Created	@Nov 7, 2020 2:19 PM
🏷️ Tags	Training

How-to set up a basic configuration for aws-cli to ease basic operations for training courses

💡 Have a decent shell is a MUST 🚀
Linux, MacOS → Bash, ZSH
Windows → WLS → Bash, ZSH

Install AWSCLI

Installing, updating, and uninstalling the AWS CLI version 2

Install the AWS Command Line Interface version 2 (AWS CLI version 2) on your system.

📖 <https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html>

Rule of thumb:

- If you can install local software, install it locally
- If you can't and have docker, go for docker

💡 Trick for docker installations → Make an alias for the docker run command
alias awsdocker=`docker run --rm -it -v ~/.aws:/root/.aws amazon/aws-cli`

Configure AWSCLI

Log in the AWS console and go to "My Security Credentials" section and create a Acces Key. You can download the configuration as a CSV file (this file should be stored securely).

The screenshot displays the AWS IAM console interface. On the left, the navigation pane shows 'Identity and Access Management (IAM)' with options like Dashboard, Access management, Groups, Users, Roles, Policies, Identity providers, Account settings, Access reports, Access analyzer, Archive rules, and Analyzers. The main content area is titled 'My security credentials' and shows 'Account details' for the user 'imecode'. A dropdown menu is open on the right, listing various account management options, with 'My Security Credentials' highlighted by a red rectangular box. Below the account details, there are tabs for 'AWS IAM credentials', 'AWS CodeCommit credentials', and 'Amazon MCS credentials'. The 'AWS IAM credentials' tab is selected, showing a 'Password for console access' section with a 'Change password' button. The page also includes a 'Sign Out' button in the top right corner.

Access keys for CLI, SDK, & API access

Use access keys to make programmatic calls to AWS from the AWS Command Line Interface (AWS CLI), Tools for Windows PowerShell, the AWS SDKs, or direct AWS API calls. **If you lose or forget your secret key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.** [Learn more](#)

Create access key

Access key ID	Status	Created	Last used	Actions
AK 	Active	2020-11-02 10:21 UTC+0100	2020-11-07 11:19 UTC+0100	Make inactive Delete

AWSCLI works with profiles, to use different configurations, to avoid clashing with other configurations this manual will create a profile called `sre`

```
$ aws configure --profile sre
AWS Access Key ID [None]: # Access Key ID
AWS Secret Access Key [None]: # Access Secret Key
Default region name [None]: eu-west-3
Default output format [None]: json
```

Handling Key Pairs

We need a Key-Pair to access any EC2 instance. To create one

```
$ aws ec2 create-key-pair --profile sre --key-name sre-kp
{
  "KeyFingerprint": "62:65:83:13:d9:2b:bc:47:49:a9:dd:98:38:d5:bc:38:38:6a:9f:fc",
  "KeyMaterial": "-----BEGIN RSA PRIVATE KEY-----\nMIIEow...vu/93cvy0LYG3g\n-----END RSA PRIVATE KEY-----",
  "KeyName": "sre",
  "KeyPairId": "key-00e5efc6ae267d763"
}

# To create directly a PEM file
$ aws ec2 create-key-pair --profile sre \
  --key-name sre-kp \
  --query "KeyMaterial" \
  --output text > sre-kp.pem

# Even a KeyPair has no cost, to delete it
$ aws ec2 delete-key-pair --profile sre --key-name sre-kp
```

Launching EC2 instances

First we need to find the base image to launch, to find by name:

```
$ aws ec2 describe-images --profile sre \
  --filters "Name=name,Values=pinchito*" \
  --query "Images[*].[Name,ImageId]"
[
  [
    "pinchito-loadtest-2020-11-07",
    "ami-0b1caa1a26cd41f9d"
  ],
  [
    "pinchito-loadtest-2020-11-06",
    "ami-0f845d0a891816ce6"
  ]
]
```

With that ID to launch an instance:

```
# Instance types: t2.small t2.micro t3.micro t3.small ...
# Keep the InstanceId to delete the instance after use :)
$ aws ec2 run-instances --profile sre \
  --image-id ami-0b1caa1a26cd41f9d \
  --count 1 \
  --instance-type t3.micro \
  --key-name sre-kp \
  --query "Instances[*].[InstanceId]"
[
  [
    "i-029f38af0b66790b4"
  ]
]
```

To delete an instance

```
$ aws ec2 terminate-instances --profile sre \
  --instance-ids i-013b7622be52e823f
```

To check if we have any instance running

```
# Watch out any non terminaed instance
$ aws ec2 describe-instances --profile sre \
  --query "Reservations[].Instances[].[InstanceId,State.Name]"
[
  [
    "i-029f38af0b66790b4",
    "terminated"
  ],
  [
    "i-013b7622be52e823f",
    "terminated"
  ],
  [
    "i-00ba20a1ecc0fd606",
    "terminated"
  ]
]
```