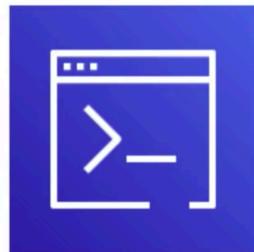


# *AWS Command Line Interface (CLI)*



**Control multiple AWS services from the command line and automate them through scripts.**



## AWS CLI

The **AWS CLI** lets you interact with AWS from anywhere by simply using a command line.

### You can from the CLI perform actions such as:

- List buckets, upload data s3
- Launch, stop, start and terminate EC2 instances
- Update security groups, create subnets
- etc...

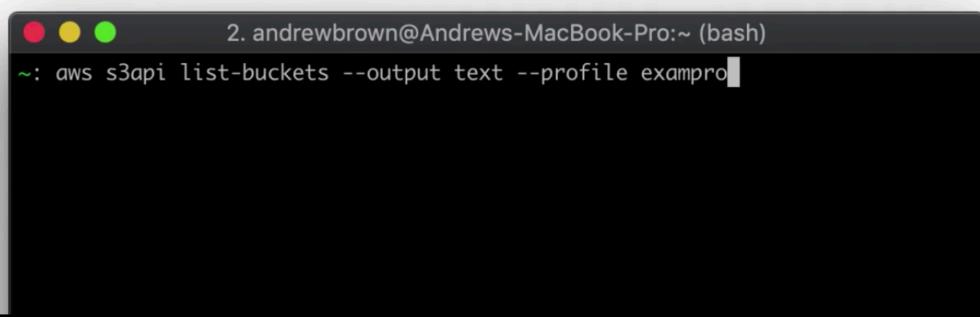
### Important AWS CLI flags to know:

- Easily switch between AWS accounts using **--profile**
- Change the **--output** between json, table and text

The CLI is installed using a



**Python** script



2. andrewbrown@Andrews-MacBook-Pro:~ (bash)  
~: aws s3api list-buckets --output text --profile exampro



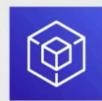
# *AWS Software Development Kit (SDK)*



**Control** multiple AWS services using popular programming languages



SUBSCRIBE



## AWS SDK



A Software Development Kit (SDK) is a set of tools and libraries that you can use to create applications for a specific software package.

The AWS SDK is a set of API libraries that let you integrate AWS services into your applications. The SDK is available for the following languages:

1. C++
2. Go
3. Java
4. Javascript
5. .NET
6. NodeJs
7. PHP
8. Python
9. Ruby

```
1 // Add to your package.json
2 // npm install aws-sdk --save-dev
3
4 const AWS = require('aws-sdk')
5
6 const bucket = 'mybucket' // the bucketname without s3://
7 const photo = 'photo.png' // the name of file
8
9 const config = new AWS.Config({
10   accessKeyId: process.env.AWS_ACCESS_KEY_ID,
11   secretAccessKey: process.env.AWS_SECRET_ACCESS_KEY,
12   region: process.env.AWS_REGION
13 })
14
15 const client = new AWS.Rekognition();
16 const params = {
17   Image: {
18     S3Object: {
19       Bucket: bucket,
20       Name: photo
21     },
22   },
23   MaxLabels: 10
24 };
25
26 client.detectLabels(params, function(err, response) {
27   if (err) {
28     console.log(err, err.stack); // an error occurred
29   } else {
30     console.log('Detected labels for: ${photo}')
31
32     response.Labels.forEach(label => {
33       console.log(`Label: ${label.Name}`)
34       console.log(`Confidence: ${label.Confidence}`)
35       console.log(`Instances:`)
36       label.Instances.forEach(instance => {
37         let box = instance.BoundingBox
38         console.log(`  Bounding box:`)
39         console.log(`    Top: ${box.Top}`)
40         console.log(`    Left: ${box.Left}`)
41         console.log(`    Width: ${box.Width}`)
42         console.log(`    Height: ${box.Height}`)
43         console.log(`    Confidence: ${instance.Confidence}`)
44       })
45       console.log(`Parents:`)
46       label.Parents.forEach(parent => {
47         console.log(`  ${parent.Name}`)
48       })
49       console.log(`-----`)
50     })
51   } // for response.labels
52 });
53
```



## Programmatic Access - Access Key and Secret

When you enable **Programmatic Access** for AWS users

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type\*  **Programmatic access**  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

**AWS Management Console access**  
Enables a **password** that allows users to sign-in to the AWS Management Console.

You'll have the ability create **Access Key ID** and **Secret Access Key**  
These are collectively known as **AWS Credentials**

Create access key

[Download .csv file](#)

Access key ID	Secret access key
AKIAZRJIQN2OLGRB6Y6V	pOOXbbbbADbMAg9UVgd9hNr+gKhG2T5ebuE2/sT/ Hide

[Close](#)

You will want to stored your credentials in your user's home eg. `~/.aws/credentials`

The credentials files allow you to manage multiple credentials (called profiles)

```
(credentials (~/.aws) - VIM13
1 [default]
2 aws_access_key_id = AZyekLr9JHxu3HJ2qfUj
3 aws_secret_access_key = S10lKPAvx8fBujiFB2197cizljkRr6EVQF7ZMbK
4 [enterprise-0]
5 aws_access_key_id = Lhb0KebuJzKILYoN2squ
6 aws_secret_access_key = eN2Yv0aywyW89gRJVadWdN9puJDzcdFe96I6Fkl
7 [deep-space-nine]
8 aws_access_key_id = AtNwPtUlaeD55TrI890
9 aws_secret_access_key = rWtRBXrZJGqJx0SZj8TGgmmh5W0Xft36skPte
~)
credentials Line:9/9[100%]Col:62Buf:#1[69][0x45]
```





AWS CLI



SDK *CheatSheet*

- **CLI** stands for Command Line Interface
- **SDK** stands for Software Development Kit
- The **AWS CLI** lets you interact with AWS from anywhere by simply using a command line
- The **AWS SDK** is a set of API libraries that let you integrate AWS services into your applications.
- **Programmatic Access** must be enabled per user via the IAM console to use CLI or SDK
- **aws configure** command used to setup your AWS credentials for the CLI
- The CLI is installed via a Python script
- Credentials get stored in a plain text file (whenever possible use roles instead of AWS credentials)
- The SDK is available for the following programming languages
  - C++
  - Go
  - Java
  - Javascript
  - .NET
  - NodeJs
  - PHP
  - Python
  - Ruby

