

The AWS Command Line Interface (AWS CLI) is a command-line tool that allows you to interact with AWS services using commands in your terminal/command prompt.

AWS CLI enables you to run commands to provision, configure, list, delete resources in the AWS cloud. Before you run any of the [aws commands](#), you need to follow three steps:

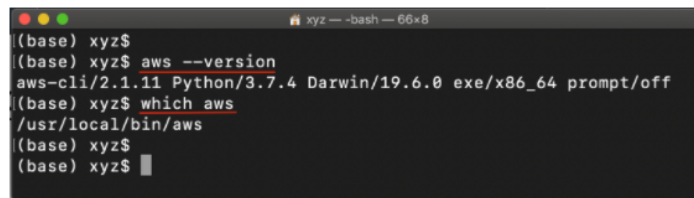
1. Install AWS CLI
2. Create an IAM user with Administrator permissions
3. Configure the AWS CLI

Step 1. Install AWS CLI v2

Refer to the official [AWS instructions to install/update AWS CLI](#) (version 2) based on your underlying OS. You can verify the installation using the following command in your terminal (macOS)/cmd (Windows).

```
# Display the folder that contains the symlink to the aws cli tool
which aws
# See the current version
aws --version
```

See the sample output below. Note that the exact version of AWS CLI and Python may vary in your system.



```
(base) xyz$
(base) xyz$ aws --version
aws-cli/2.1.11 Python/3.7.4 Darwin/19.6.0 exe/x86_64 prompt/off
(base) xyz$ which aws
/usr/local/bin/aws
(base) xyz$
(base) xyz$
```

Mac/Linux/Windows: Verify the successful installation of AWS CLI 2

Step 2. Create an IAM user

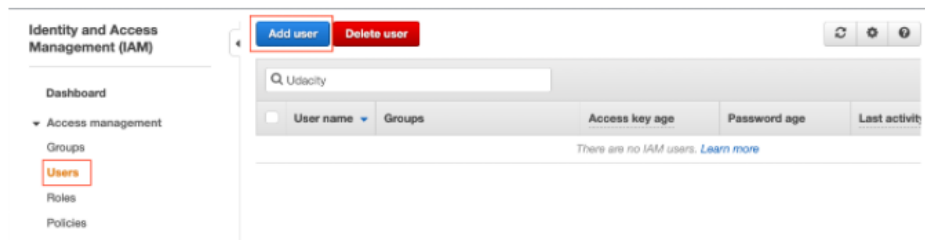
In this step, you will create an IAM user with Administrator permissions who is allowed to perform *any* action in your AWS account, only through CLI. After creating such an IAM user, we will use its **Access key** (long-term credentials) to configure the AWS CLI locally.

Let's create an [AWS IAM](#) user, and copy its Access key.

AWS Identity and Access Management (IAM) service allows you to authorize users / applications (such as AWS CLI) to access AWS resources.

The Access key is a combination of an **Access Key ID** and a **Secret Access Key**. Let's see the steps to create an IAM user, and generate its Access key.

- Navigate to the [IAM Dashboard](#), and create an IAM user.



Add a new IAM user

- Set the user details, such as the name, and access type as *Programmatic access* only.

Add user 1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name* [Add another user](#)

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.

Set the user name, and type (mode) of access

- Set the permissions to the new user by attaching the AWS Managed **AdministratorAccess** policy from the list of existing policies.

Add user 1 2 3 4 5

Set permissions

[Add user to group](#) [Copy permissions from existing user](#) [Attach existing policies directly](#)

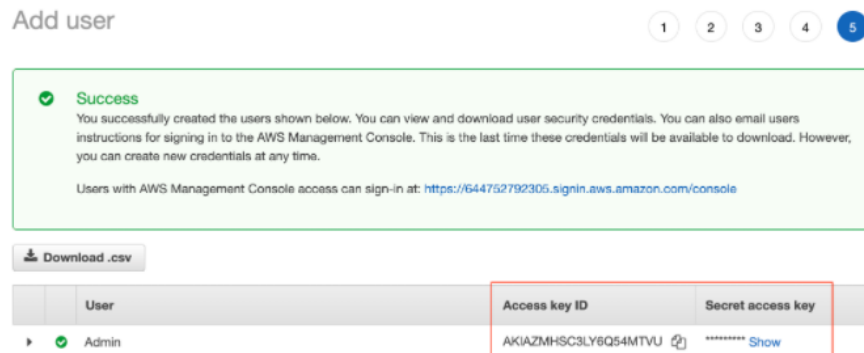
[Create policy](#) [Refresh](#)

Filter policies Showing 30 results

	Policy name	Type	Used as
<input checked="" type="checkbox"/>	AdministratorAccess	Job function	None
<input type="checkbox"/>	AdministratorAccess-Amplify	AWS managed	None
<input type="checkbox"/>	AmazonAPIGatewayAdministrator	AWS managed	None
<input type="checkbox"/>	AmazonSageMakerAdmin-ServiceCatalogProductsServiceRolePolicy	AWS managed	None
<input type="checkbox"/>	AmazonWorkSpacesAdmin	AWS managed	None
<input type="checkbox"/>	AmazonWorkSpacesApplicationManagerAdminAccess	AWS managed	None
<input type="checkbox"/>	AWSAppSyncAdministrator	AWS managed	None
<input type="checkbox"/>	AWSAuditManagerAdministratorAccess	AWS managed	None

Attach the *AdministratorAccess* policy from the list of pre-created policies

- Provide tags [optional], review the details of the new user, and finally create the new user.
- After a user is created successfully, download the access key file (.csv) containing the *Access Key ID* and a *Secret Access Key*. You can even copy the keys and stay on the same page. **Don't skip this step as this will be your only opportunity to download the secret access key file.**



Copy the Access key of the new user OR download the .csv file containing the Access key

Step 3. Configure the AWS CLI

You will need to configure the following four items on your local machine before you can interact with any of the AWS services:

1. **Access key** - It is a combination of an *Access Key ID* and a *Secret Access Key*. Together, they are referred to as *Access key*. You can generate an Access key from the AWS IAM service, and specify the level of permissions (authorization) with the help of *IAM Roles*.
2. **Default AWS Region** - It specifies the AWS Region where you want to send your requests by default.
3. **Default output format** - It specifies how the results are formatted. It can either be a json, yaml, text, or a table.
4. **Profile** - A collection of settings is called a profile. The default profile name is `default`, however, you can create a new profile using the `aws configure --profile new_name` command. A sample command is given below.

If you have closed the web console that showed the access key, you can open the downloaded access key file (.csv) to copy the keys later. It should be something similar to:

```
AWSAccessKeyId=WANI9WATIG63GKXA89VC74A
AWSSecretKey=kMT2Jn5NPkq1GxtoUqwUbgHtPbsf10Dm/Pbsf10D
```

```
(base) xyz$ aws configure list
Name                               Value                                Type    Location
----                               -
profile                             <not set>                           None    None
access_key                           <not set>                           None    None
secret_key                           <not set>                           None    None
region                               us-east-2                           config-file /Users/xyz/.aws/config
(base) xyz$ aws configure --profile default
AWS Access Key ID [None]: AKIAZMHSC3LY6Q54MTVU
AWS Secret Access Key [None]: kMT2Jn5NPkq1GxtoUqwUbgHtPbsf10Dm/Pbsf10D
Default region name [us-east-2]: us-east-2
Default output format [json]: json
(base) xyz$
(base) xyz$
```

Mac/Linux: List your present configuration, and then configure your default aws profile

- Set the default profile credentials

```
# Navigate to the home directory
cd ~
# If you do not use the profile-name, a default profile will be created for
aws configure --profile <profile-name>
# View the current configuration
aws configure list --profile <profile-name>
# View all existing profile names
aws configure list-profiles
# In case, you want to change the region in a given profile
# aws configure set <parameter> <value> --profile <profile-name>
aws configure set region us-east-1 --profile <profile-name>
```

Moving forward, you can use `--profile <profile-name>` option with any AWS command. This will resolve the conflict if you have multiple profiles set up locally.

The command above will store the access key in a default file `~/.aws/credentials` and store the profile in the `~/.aws/config` file. Upon prompt, paste the copied access key (access key id and secret access key). Enter the default region as `us-east-1` and output format as `json`.

- Let the system know that your sensitive information is residing in the .aws folder

```
export AWS_CONFIG_FILE=~/.aws/config
export AWS_SHARED_CREDENTIALS_FILE=~/.aws/credentials
```

```
(base) xyz$ aws configure list
      Name      Value      Type      Location
      ----      -
profile        <not set>      None      None
access_key      *****C74A  shared-credentials-file
secret_key      *****Jn5N  shared-credentials-file
region          us-east-2      config-file  /Users/xyz/.aws/config
(base) xyz$
```

Mac/Linux: A successful configuration

- After a successful credential set-up, your "credentials" file will look like:

```
(base) xyz$ cat credentials
[default]
region=us-east-2
output=json
aws_access_key_id = *****C74A*****
aws_secret_access_key = *****Jn5N*****
(base) xyz$
```

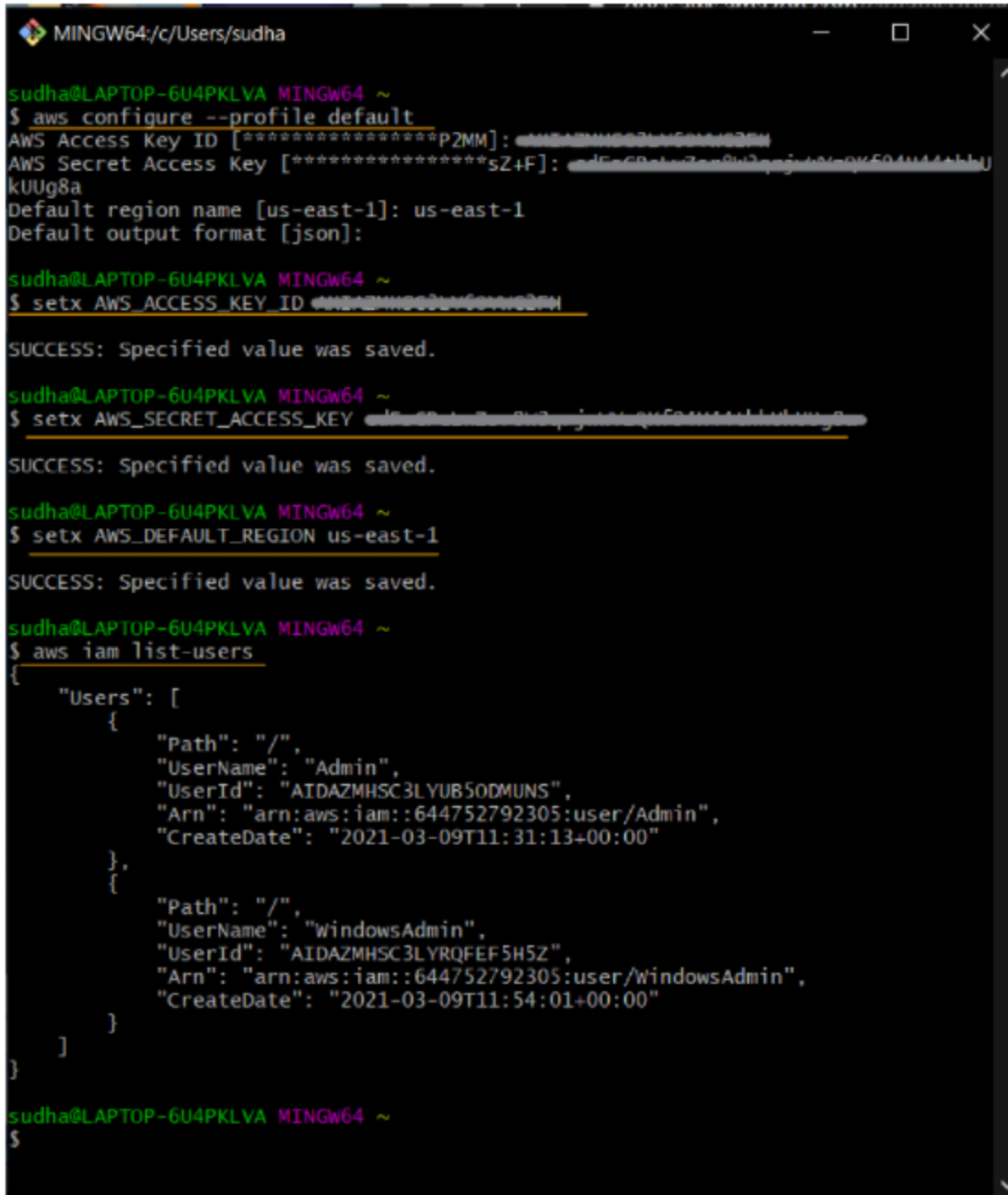
Mac/Linux: View the credentials file using `cat ~/.aws/credentials` command

- **Windows users with GitBash only**

You will have to set the environment variables. Run the following commands in your GitBash terminal:

```
setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
setx AWS_DEFAULT_REGION us-west-2
```

Replace the access key ID and secret, as applicable to you. Windows users using WSL do not need this step, they will follow all steps as if they are Linux users.



```
MINGW64:/c/Users/sudha
sudha@LAPTOP-6U4PKLVA MINGW64 ~
$ aws configure --profile default
AWS Access Key ID [*****p2MM]: 
AWS Secret Access Key [*****sZ4F]: 
Default region name [us-east-1]: us-east-1
Default output format [json]: 

sudha@LAPTOP-6U4PKLVA MINGW64 ~
$ setx AWS_ACCESS_KEY_ID 
SUCCESS: Specified value was saved.

sudha@LAPTOP-6U4PKLVA MINGW64 ~
$ setx AWS_SECRET_ACCESS_KEY 
SUCCESS: Specified value was saved.

sudha@LAPTOP-6U4PKLVA MINGW64 ~
$ setx AWS_DEFAULT_REGION us-east-1
SUCCESS: Specified value was saved.

sudha@LAPTOP-6U4PKLVA MINGW64 ~
$ aws iam list-users
{
  "Users": [
    {
      "Path": "/",
      "UserName": "Admin",
      "UserId": "AIDAZMHSC3LYUB50DMUNS",
      "Arn": "arn:aws:iam::644752792305:user/Admin",
      "CreateDate": "2021-03-09T11:31:13+00:00"
    },
    {
      "Path": "/",
      "UserName": "windowsAdmin",
      "UserId": "AIDAZMHSC3LYRQFEF5H5Z",
      "Arn": "arn:aws:iam::644752792305:user/windowsAdmin",
      "CreateDate": "2021-03-09T11:54:01+00:00"
    }
  ]
}
```

Windows: Successful configuration using the GitBash terminal

Step 4. Run your first AWS CLI command

- Check the successful configuration of the AWS CLI, by running either of the following AWS command:

```
# If you've just one profile set locally
aws iam list-users
# If you've multiple profiles set locally
aws iam list-users --profile <profile-name>
```

The output will display the details of the recently created user:

```
{
  "Users": [
    {
      "Path": "/",
      "UserName": "Admin",
      "UserId": "AIDAZMXYZ3LY2BNC5ZM5E",
      "Arn": "arn:aws:iam::388752792305:user/Admin",
      "CreateDate": "2021-01-28T13:44:15+00:00"
    }
  ]
}
```

Troubleshoot

If you are facing issues while following the commands above, refer to the detailed instructions here -

- 1.) Configuration basics
(<https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-quickstart.html>)
- 2.) Configuration and credential file settings
(<https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-files.html>)
- 3.) Environment variables to configure the AWS CLI
(<https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-envvars.html>)

Updating the specific variable in the configuration

In the future, you can set a single value, by using the command, such as:

```
# Syntax
# aws configure set <varname> <value> [--profile profile-name]
aws configure set default.region us-east-2
```

It will update only the region variable in the existing default profile.