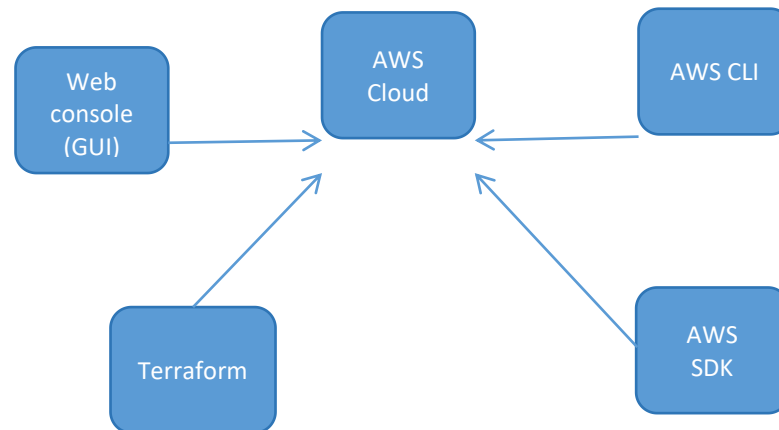


## AWS CLI:

We are able to access AWS Cloud from Web console



For DevOps Engineer, which option is trending in industry, it is Terraform

AWS CLI: Command-line interface

Ways for infrastructure configuration: AWS management web console, AWS CLI (Command-Line Interface)

AWS CLI ---> Usually the script provides you with flexibility to manage AWS resources and infrastructure

We need an AccessKey and SecretKey

Go to : <https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

Click on this link:

▼ Windows

**Install and update requirements**

- We support the AWS CLI on Microsoft-supported versions of 64-bit Windows.
- Admin rights to install software

**Install or update the AWS CLI**

To update your current installation of AWS CLI on Windows, download a new installer each time you update to overwrite previous versions. AWS CLI is updated regularly. To see when the latest version was released, see the [AWS CLI version 2 Changelog](#) on GitHub.

1. Download and run the AWS CLI MSI installer for Windows (64-bit):  
<https://awscli.amazonaws.com/AWSCLIV2.msi>

Alternatively, you can run the `msiexec` command to run the MSI installer.

```
C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi
```

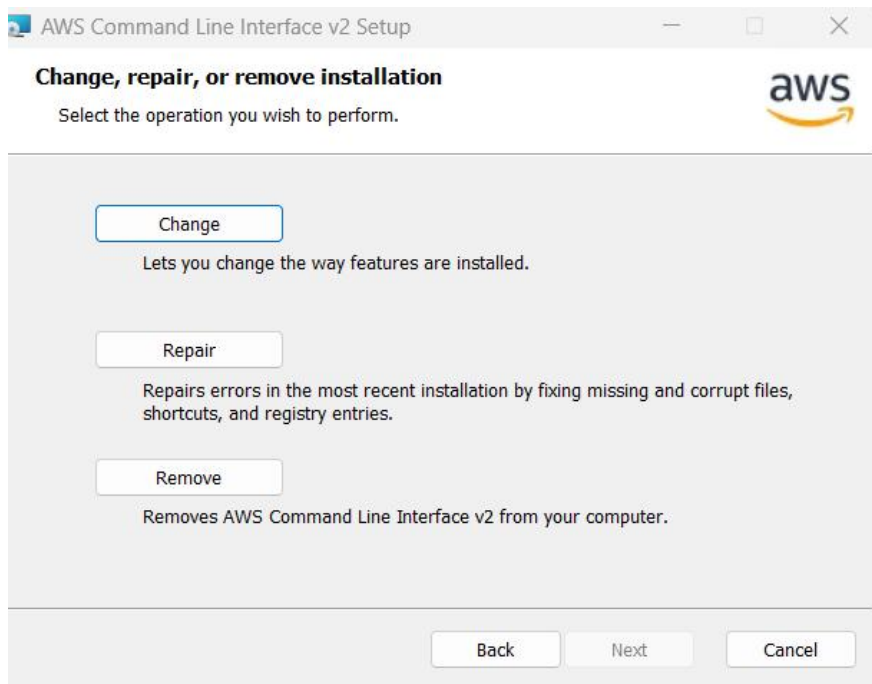
For various parameters that can be used with `msiexec`, see [msiexec](#) on the Microsoft Docs website. For example, you can use the `/q`

Install AWS CLI on Windows

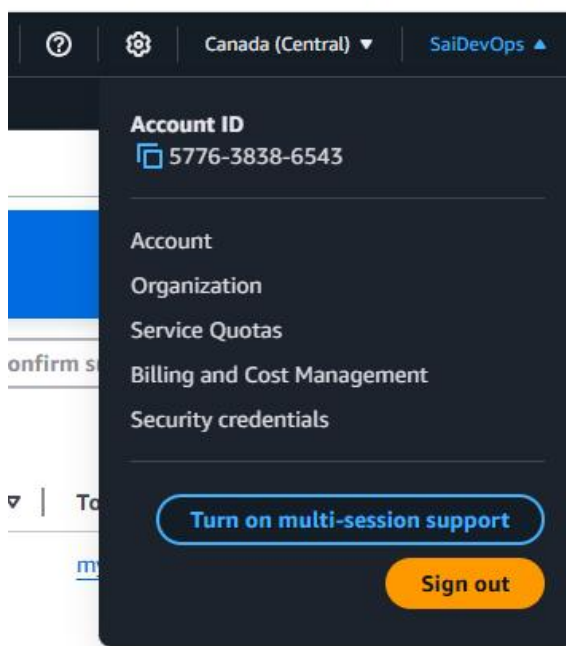
`msiexec.exe /i` <https://awscli.amazonaws.com/AWSCLIV2.msi>

Or click on <https://awscli.amazonaws.com/AWSCLIV2.msi>

Install AWS CLI



Go to AWS Console ----> Security credentials



Create access key --> What's recommended is. Create an IAM user first then create access key for that IAM user

### Access keys (0)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys per IAM user.

Access key ID	Created on	Access key last used	Region last used
---------------	------------	----------------------	------------------

No access keys

As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short-term credentials.

Create access key

### CloudFront key pairs (0)

You use key pairs in Amazon CloudFront to create signed URLs. You can have a maximum of two CloudFront key pairs (active or inactive) at a time.

I will delete access key right after using it

### Alternatives to root user access keys [Info](#)

#### ⚠ Root user access keys are not recommended

We don't recommend that you create root user access keys. Because you can't specify the root user in a permissions policy, you can't limit its permissions, which is a best practice.

Instead, use alternatives such as an IAM role or a user in IAM Identity Center, which provide temporary rather than long-term credentials. [Learn More](#)

If your use case requires an access key, create an IAM user with an access key and apply least privilege permissions for that user. [Learn More](#)

#### Continue to create access key?

☒ I understand creating a root access key is not a best practice, but I still want to create one.

Cancel

Create access key

### Configuring AWS CLI

1. Create an AWS account in order to configure AWS CLI (Use existing if you already have one)
2. Create IAM user with Security credentials (Access key and Secret key)
3. Open Command prompt and connect

```
C:\Users\saito>
C:\Users\saito>aws --version
aws-cli/2.24.22 Python/3.12.9 Windows/11 exe/AMD64
C:\Users\saito>
```

Next type --> aws configure

Enter AWS Access key and secret key

Correct the region to ca-central-1

```
C:\Users\saito>
C:\Users\saito>aws s3 ls
2025-03-22 12:32:58 elasticbeanstalk-ca-central-1-577638386543
2025-03-22 12:23:05 elasticbeanstalk-us-east-1-577638386543
```

Go to AWS CLI Command Reference  
<https://docs.aws.amazon.com/cli/latest/>

# AWS CLI Command Reference ¶

The AWS Command Line Interface is a unified tool that provides a consistent interface for interacting with all parts of AWS.

- [Command Reference](#)
  - [accessanalyzer](#)
  - [account](#)
  - [acm](#)
  - [acm-pca](#)
  - [amp](#)
  - [amplify](#)
  - [amplifybackend](#)
  - [amplifyuibuilder](#)
  - [apigateway](#)
  - [apigatewaymanagementapi](#)
  - [apigatewayv2](#)
  - [appconfig](#)

CLI Documentation: <https://docs.aws.amazon.com/cli/latest/>

Example for EC2: <https://docs.aws.amazon.com/cli/latest/reference/ec2/>

Example to attach volume to EC2

```
[ aws . ec2 ]
```

## attach-volume ¶

### Description ¶

Attaches an EBS volume to a running or stopped instance and exposes it to the instance with the specific device name.

Encrypted EBS volumes must be attached to instances that support Amazon EBS encryption. For more information, see [Amazon EBS encryption](#) in the *Amazon EBS User Guide*.

After you attach an EBS volume, you must make it available. For more information, see [Make an EBS volume available for use](#).

If a volume has an Amazon Web Services Marketplace product code:

- The volume can be attached only to a stopped instance.
- Amazon Web Services Marketplace product codes are copied from the volume to the instance.
- You must be subscribed to the product.
- The instance type and operating system of the instance must support the product. For example, you can't detach a volume from a Windows instance and attach it to a Linux instance.

For more information, see [Attach an Amazon EBS volume to an instance](#) in the *Amazon EBS User Guide*.

See also: [AWS API Documentation](#)

### Example 1: Create a bucket

The following `mb` command creates a bucket. In this example, the user makes the bucket `amzn-s3-demo-bucket`. The bucket is created in the region specified in the user's configuration file:

```
aws s3 mb s3://amzn-s3-demo-bucket
```

Output:

Create a new bucket

```
C:\Users\saito>aws s3 mb s3://devops-test
make_bucket failed: s3://devops-test An error occurred (BucketAlreadyExists) when calling the
make_bucket action: The requested bucket name is not available. The bucket namespace is shared by all users of the
aws CLI. Please choose a different name and try again.

C:\Users\saito>aws s3 mb s3://devops-test-567
make_bucket: devops-test-567
```

One more bucket is there

**General purpose buckets** | **Directory buckets**

**General purpose buckets (3)** [Info](#) [All AWS Regions](#)

Buckets are containers for data stored in S3.

	Name
<input type="radio"/>	<a href="#">devops-test-567</a>
<input type="radio"/>	<a href="#">elasticbeanstalk-ca-central-1-577638386543</a>
<input type="radio"/>	<a href="#">elasticbeanstalk-us-east-1-577638386543</a>

```
C:\Users\saito>aws s3 rb s3://devops-test-567
remove_bucket: devops-test-567
```

For every action we need to use commands and which command to run those details are available in AWS documentation

For example:

To Display bucket list: `aws s3 ls`

Create a new bucket: `aws mb s3://<new_bucket>`

Delete an empty bucket: `aws rb s3://<bucket_name>`

```
C:\Users\saito>
C:\Users\saito>aws ec2 describe-instances
```

DescribeInstances	
Reservations	
OwnerId	577638386543
ReservationId	r-0747843728a336750
Instances	
AmiLaunchIndex	0
Architecture	x86_64
BootMode	uefi-preferred
ClientToken	7721b732-ec8e-461b-a77c-3ee640ab8682
CurrentInstanceBootMode	legacy-bios
EbsOptimized	False
EnaSupport	True

```
-- More --
```