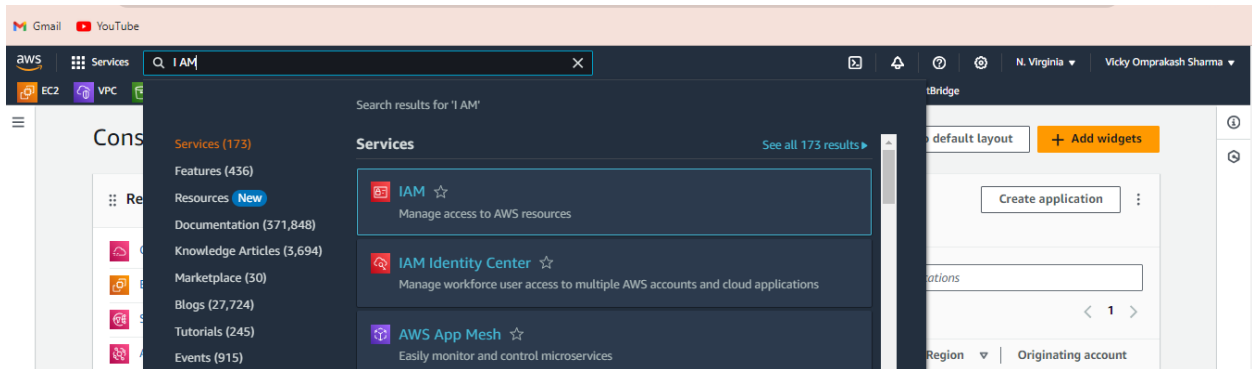


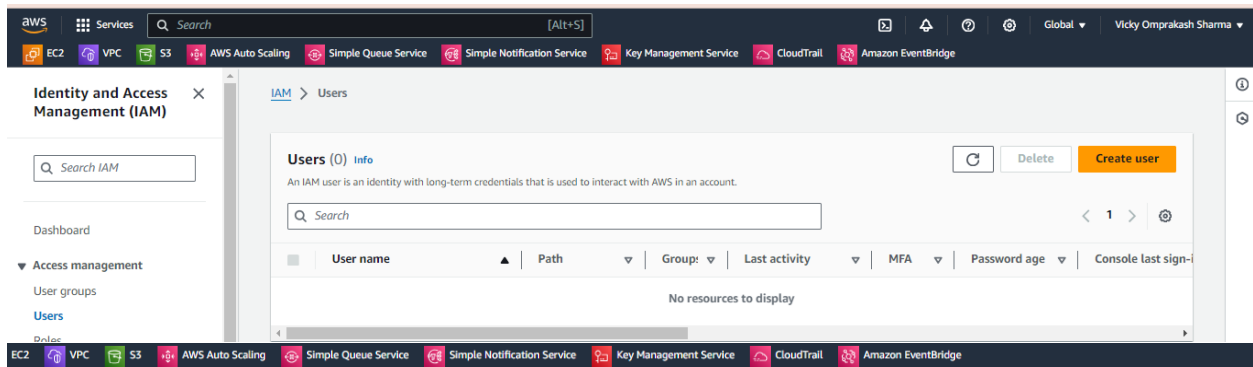
AIM- HOW TO CREATE EC2 INSTANCE USING AWS CLI

STEPS:-

STEP1 –

CREATE A PARTICULAR USER BY USING IAM USER





IAM > Users > Create user

Step 1
Specify user details

Step 2
Set permissions

Step 3
Review and create

Specify user details

User details

User name

TRIALUSER

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☐ Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

ⓘ

If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. [Learn more](#)

Cancel

Next

User name

TRIALUSER

Console password type

None

Require password reset

No

Permissions summary

< 1 >

Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel

Previous

Create user

STEP 2- MAKE A SECURITY CREDITENIAL OF THESE ACCOUNT

Created
March 13, 2024, 17:25 (UTC+05:30)

Last console sign-in
-

Permissions

Groups

Tags

Security credentials

Access Advisor

Permissions policies (1)

Permissions are defined by policies attached to the user directly or through groups.

Search

Filter by Type
All types

< 1 >

Policy name	Type	Attached via
AdministratorAccess	AWS managed - job function	Directly

Permissions boundary (not set)

Access keys (IAM)

Assign MFA device

Access keys (0)

Create access key

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

SSH public keys for AWS CodeCommit (0)

Upload SSH public key

SSH Key ID	Uploaded	Status
------------	----------	--------

Retrieve access keys Info

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key

Secret access key

AKIA5FTZBAIDQEK4DTAZ

***** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.

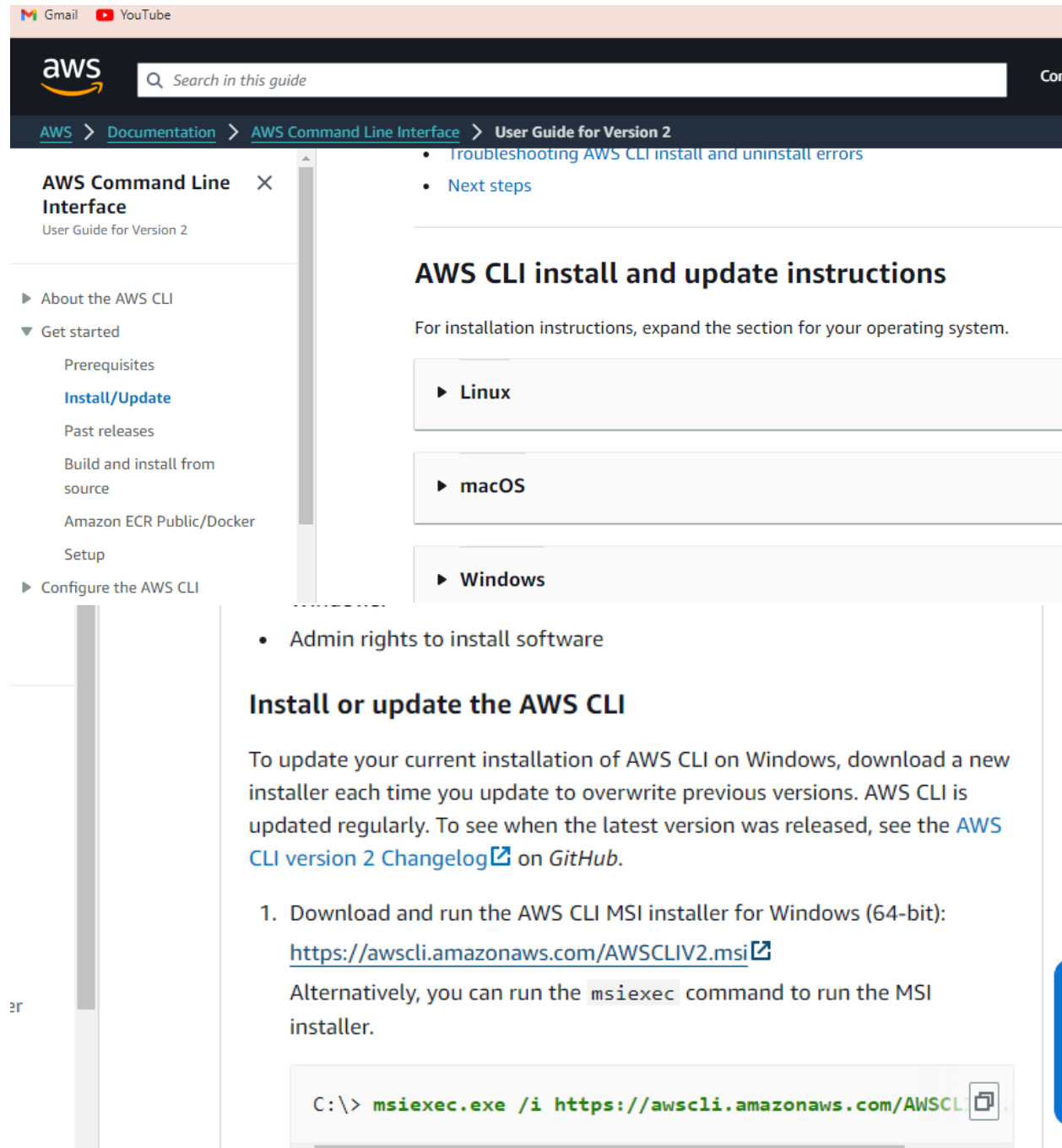
STEP-3

INSTALL AWS CLI

REFERENCE LINK –

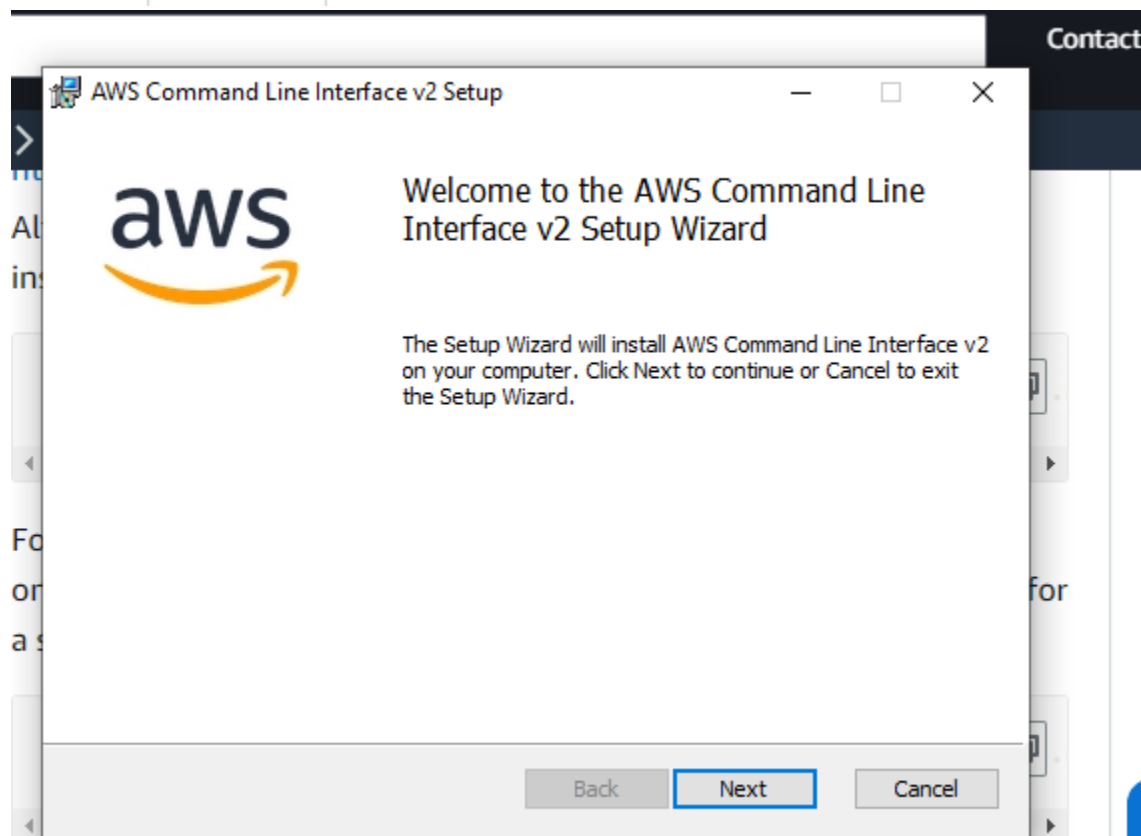
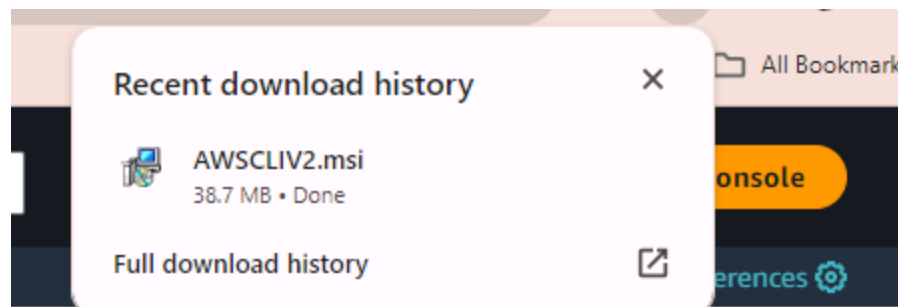
<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

NOW DOWNLOAD MSI

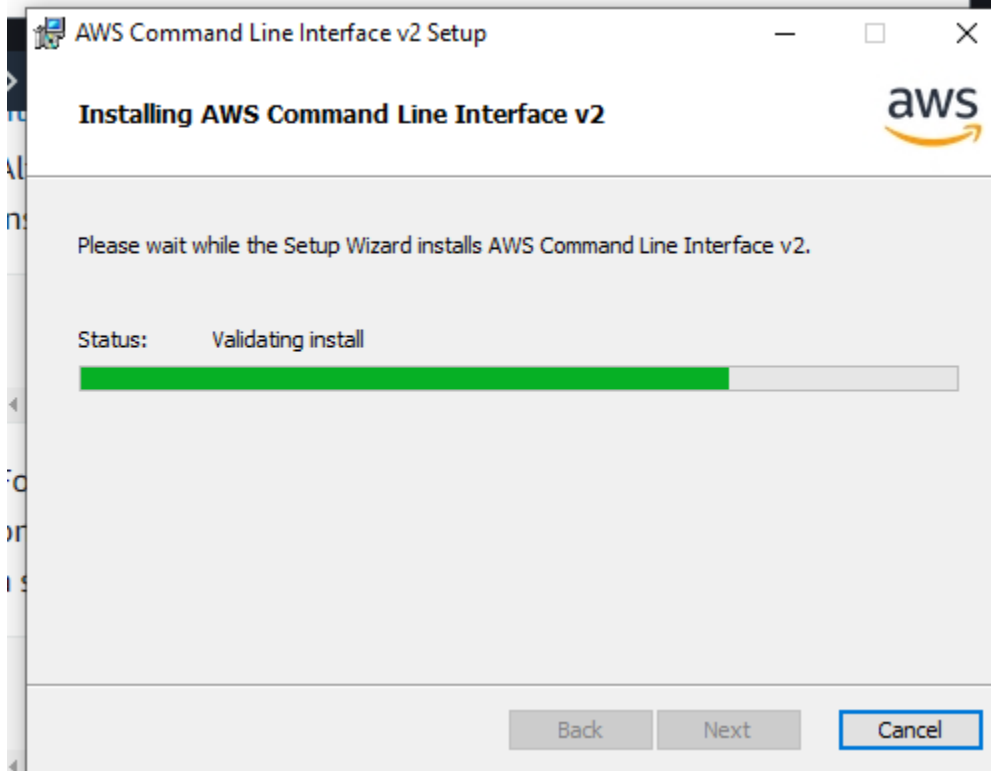
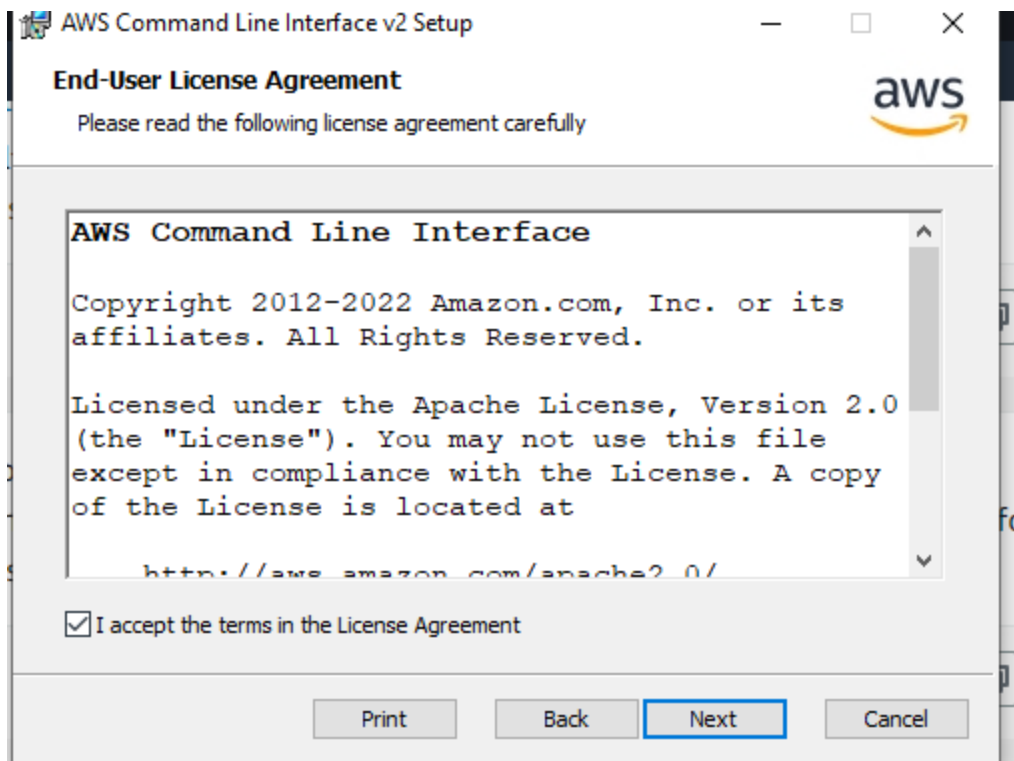


The screenshot shows the AWS CLI User Guide for Version 2. The left sidebar contains a navigation menu with the following items: About the AWS CLI, Get started (expanded), Prerequisites, **Install/Update** (selected), Past releases, Build and install from source, Amazon ECR Public/Docker, Setup, and Configure the AWS CLI. The main content area is titled 'AWS CLI install and update instructions' and includes the text: 'For installation instructions, expand the section for your operating system.' Below this, there are three expandable sections: Linux, macOS, and Windows. The Windows section is expanded, showing a list of requirements: 'Admin rights to install software'. Below the requirements, the section is titled 'Install or update the AWS CLI' and contains the text: '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](#).' The first step is: '1. Download and run the AWS CLI MSI installer for Windows (64-bit): <https://awscli.amazonaws.com/AWSCLIV2.msi>'. It also mentions: 'Alternatively, you can run the `msiexec` command to run the MSI installer.' At the bottom, there is a code block showing the command: `C:\> msiexec.exe /i https://awscli.amazonaws.com/AWSCLIV2.msi`.

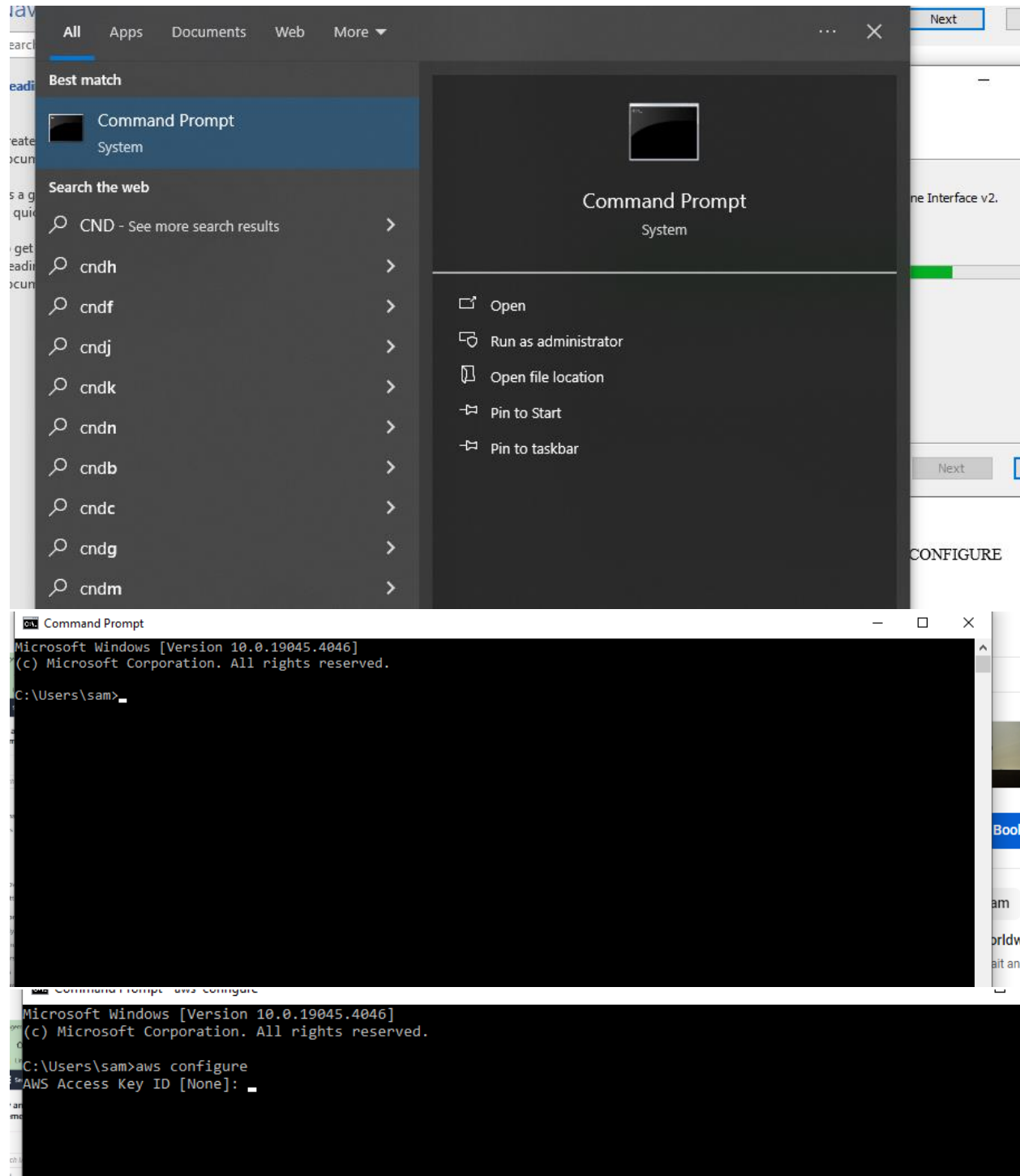
27



To confirm the installation, open the **Start** menu, search for `cmd` to



STEP-4 OPEN CMD AND GIVE COMMAND AWS CONFIGURE



Enter your I AM CREATOR USER SECURITY CREDENTIAL

```
C:\Users\sam>aws configure
AWS Access Key ID [None]: AKIA5FTZBAIDQEK4DTAZ
AWS Secret Access Key [None]: Cxk287C+bEXJDPWxGVULxh21j7yZK3w3+zg0ki3q
```

```
C:\Users\sam>aws configure
to keep track of your content in Amazon S3. Prefix logs with the following prefix:
to the Home directory.
to the heading
C:\Users\sam>
```

HERE WE WILL MAKE SURE WE DON'T USE GPT FOR ANY OUR COMMANDS:-

LET'S START

COMMAND – 1

AWS HELP

TO KNOW ABOUT AVAILABLE SERVICE JUST SCROLL DOWN YOU WILL GET EC2


```
Res Command Prompt - aws help
C:\Users\sam>aws configure
AWS Access Key ID [None]: AKIA5FTZBAIDQEK4DTAZ
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:

C:\Users\sam>aws help
aws
^^^

Description
*****

The AWS Command Line Interface is a unified tool to manage your AWS
services.

Synopsis
*****

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

Use *aws command help* for information on a specific command. Use *aws
help topics* to view a list of available help topics. The synopsis for
each command shows its parameters and their usage. Optional parameters
are shown in square brackets.
```

```
Command Prompt - aws help

Available Services
*****

* accessanalyzer
* account
* acm
* acm-pca
* alexaforbusiness
* amp
* amplify
* amplifybackend
* amplifyuibuilder
* apigateway
* apigatewaymanagementapi
* apigatewayv2
* appconfig
-- More --
```

```
Command Prompt - aws help
* dynamodb
* dynamodbstreams
* ebs
* ec2
* ec2-instance-connect
* ecr
* ecr-public
* ecs
* efs
* eks
* eks-auth
```

NOW QUIT THESE BY USING Q

AND PUT COMMAND 2 –

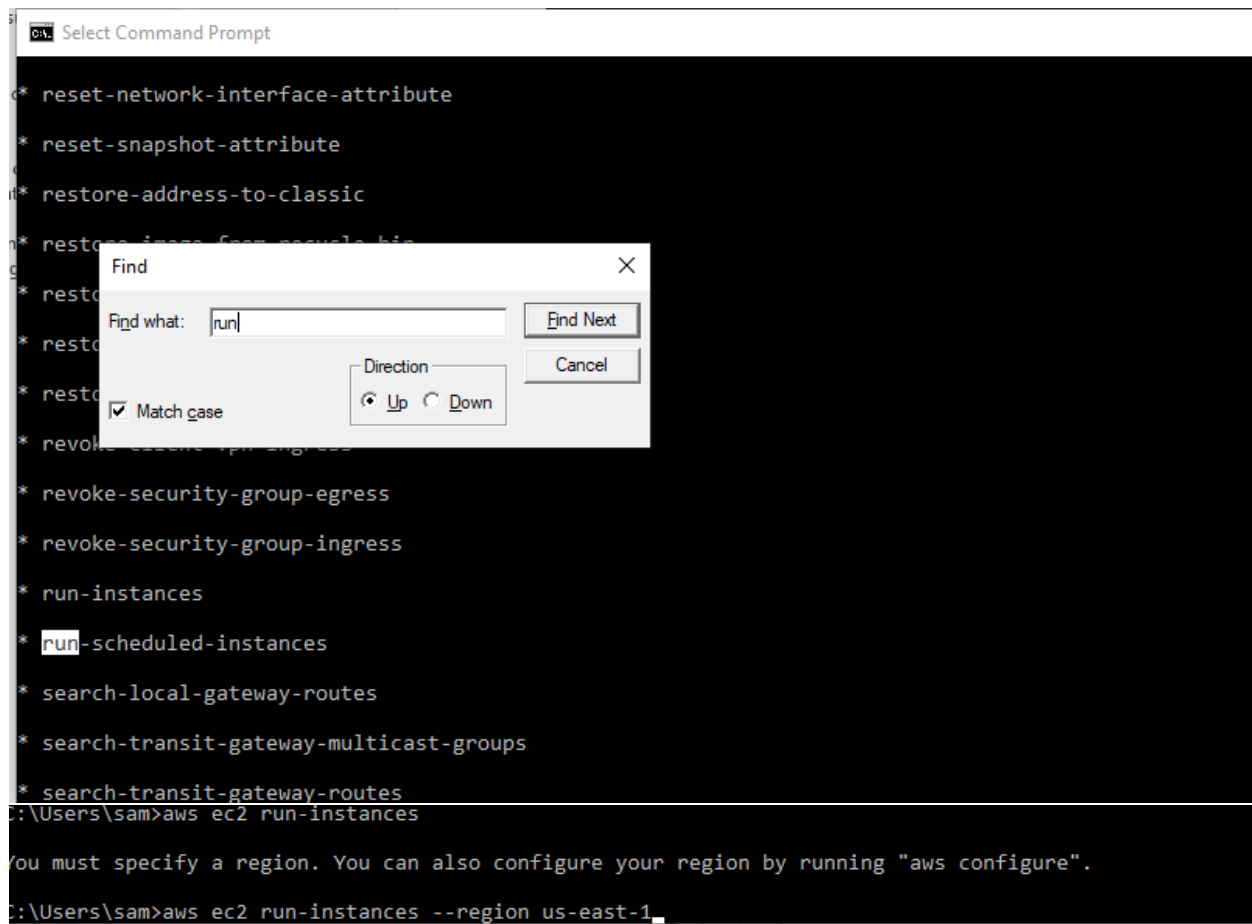
Aws ec2 help

```
C:\Users\sam>aws ec2 help
ec2
^^^

Description
*****

Amazon Elastic Compute Cloud (Amazon EC2) provides secure and
resizable computing capacity in the Amazon Web Services Cloud. Using
Amazon EC2 eliminates the need to invest in hardware up front, so you
can develop and deploy applications faster. Amazon Virtual Private
Cloud (Amazon VPC) enables you to provision a logically isolated
section of the Amazon Web Services Cloud where you can launch Amazon
Web Services resources in a virtual network that you've defined.
Amazon Elastic Block Store (Amazon EBS) provides block level storage
volumes for use with EC2 instances. EBS volumes are highly available
and reliable storage volumes that can be attached to any running
instance and used like a hard drive.
```

NOW look for instance running since we want to LAUNCH INSTANCE



```
C:\> Select Command Prompt

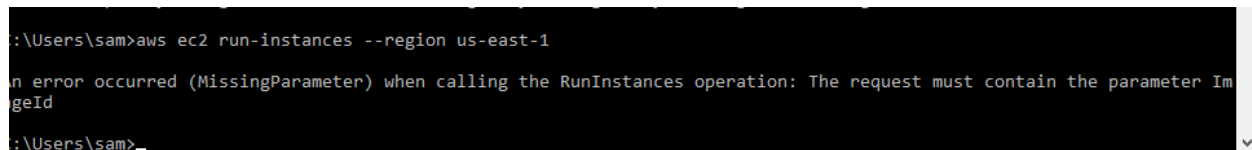
* reset-network-interface-attribute
* reset-snapshot-attribute
* restore-address-to-classic
* restore-image-from-ec2-instance
* restore-image-from-s3
* restore-image-from-snapshot
* restore-image-to-ec2-instance
* restore-image-to-s3
* revoke-security-group-egress
* revoke-security-group-ingress
* run-instances
* run-scheduled-instances
* search-local-gateway-routes
* search-transit-gateway-multicast-groups
* search-transit-gateway-routes

C:\Users\sam>aws ec2 run-instances

You must specify a region. You can also configure your region by running "aws configure".

C:\Users\sam>aws ec2 run-instances --region us-east-1
```

SPECIFIED REGION ALSO



```
C:\Users\sam>aws ec2 run-instances --region us-east-1

An error occurred (MissingParameter) when calling the RunInstances operation: The request must contain the parameter ImageId

C:\Users\sam>
```

ERROR OCCUR BECAUSE IN ORDER TO CREATE AN INSTANCE WE NEED AMI ID, INSTANCE TYPE, REGION SO TO GET THESE JUST ENTER THE COMMAND

AWS EC2 RUN-INSTANCE HELP

```
C:\Users\sam>aws ec2 run-instances help
```

```
run-instances
^^^^^^^^^^^^^^
```

```
Description
*****
```

Launches the specified number of instances using an AMI for which you have permissions.

You can specify a number of options, or leave the default options. The following rules apply:

- * If you don't specify a subnet ID, we choose a default subnet from your default VPC for you. If you don't have a default VPC, you must specify a subnet ID in the request.
- * All instances have a network interface with a primary private IPv4 address. If you don't specify this address, we choose one from the IPv4 range of your subnet.
- * Not all instance types support IPv6 addresses. For more information, see Instance types .

* If you don't specify a key name, we choose a default key name for you.

```
C:\> Select Command Prompt - aws ec2 run-instances help
```

```
See also: AWS API Documentation
```

```
Synopsis
*****
```


```
run-instances
[--block-device-mappings <value>]
[--image-id <value>]
[--instance-type <value>]
[--ipv6-address-count <value>]
[--ipv6-addresses <value>]
[--kernel-id <value>]
[--key-name <value>]
[--monitoring <value>]
[--placement <value>]
[--ramdisk-id <value>]
[--security-group-ids <value>]
[--security-groups <value>]
[--subnet-id <value>]
[--user-data <value>]
[--additional-info <value>]
```

```
*****


run-instances
[--block-device-mappings <value>]
[--image-id <value>]
[--instance-type <value>]
[--ipv6-address-count <value>]
[--ipv6-addresses <value>]
[--kernel-id <value>]
[--key-name <value>]
[--monitoring <value>]
[--placement <value>]
[--ramdisk-id <value>]
[--security-group-ids <value>]
[--security-groups <value>]
[--subnet-id <value>]
```

Command –


Amazon Linux




macOS




Ubuntu




Windows




Red Hat



SUSE L





Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

Free tier eligible

ami-0f403e3180720dd7e (64-bit (x86), uefi-preferred) / ami-0237525b5672165b3 (64-bit (Arm), uefi)

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 AMI 2023.3.20240304.0 x86_64 HVM kernel-6.1

Architecture

64-bit (x86)

Boot mode

uefi-preferred

AMI ID

ami-0f403e3180720dd7e

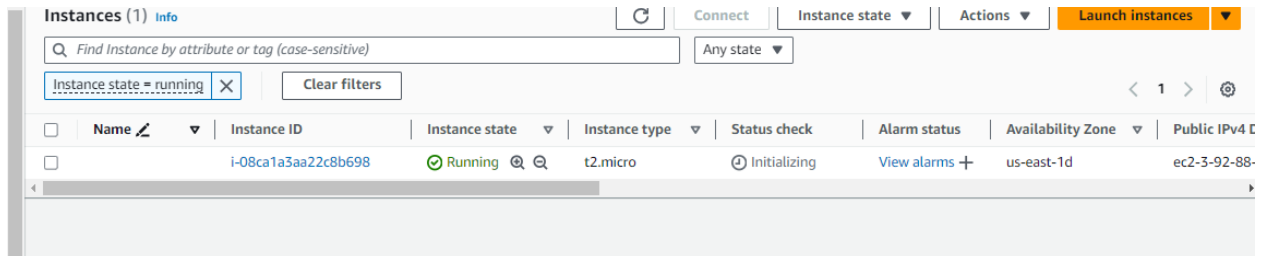
Verified provider

aws ec2 run-instances --image-id=(your any ami id) --instance-type=t2.micro --region=us-east-1

```
C:\Users\sam>aws ec2 run-instances --image-id=ami-0f403e3180720dd7e --instance-type=t2.micro --region=us-east-1
{
  "Groups": [],
  "Instances": [
    {
      "AmiLaunchIndex": 0,
      "ImageId": "ami-0f403e3180720dd7e",
      "InstanceId": "i-08ca1a3aa22c8b698",
      "InstanceType": "t2.micro",
      "LaunchTime": "2024-03-13T15:55:03+00:00",
      "Monitoring": {
        "State": "disabled"
      },
      "Placement": {
        "AvailabilityZone": "us-east-1d",
        "GroupName": "",
        "Tenancy": "default"
      },
      "PrivateDnsName": "ip-172-31-19-173.ec2.internal",

```

We create our ec2 instance now just go to console to check it



The screenshot shows the AWS Management Console 'Instances' page. At the top, there's a search bar with the text 'Find Instance by attribute or tag (case-sensitive)' and a filter dropdown set to 'Any state'. Below this, a filter box shows 'Instance state = running' with a clear filter button. The main table lists one instance with the following details:

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>		i-08ca1a3aa22c8b698	Running	t2.micro	Initializing	View alarms	us-east-1d	ec2-3-92-88-

YOU CAN CROSS VERIFY BY CHECKING THE BOTH INSTANCE ID