

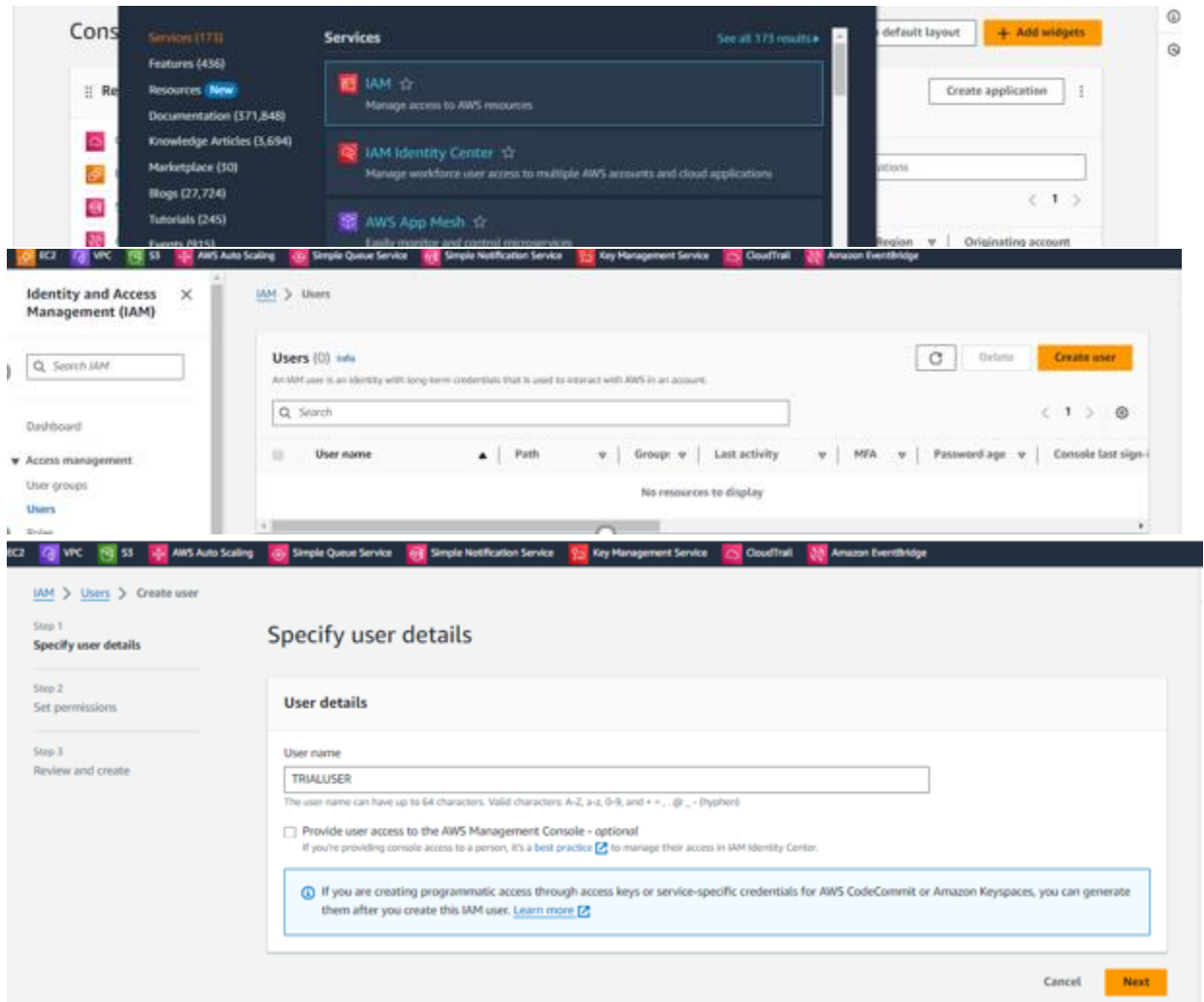
NAME –DEEPAK SIINGH

TASK -5 –CREATE EC2 USING AWS CLI

AIM- HOW TO CREATE EC2 INSTANCE USING AWS CLI

STEP1 –

CREATE A PARTICULAR USER BY USING I AM USER



User name
TRIALUSER

Console password type
None

Require password reset
No

Permissions summary

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)

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)

Actions

Upload SSH public key

User SSH public keys to authenticate access to AWS CodeCommit repositories. You can have a maximum of five SSH public keys (active or inactive) at a time.

SSH Key ID

Uploaded

Status

STEP-3

INSTALL AWS CLI

REFERENCE LINK -

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

NOW DOWNLOAD MSI

Gmail YouTube

[AWS](#) > [Documentation](#) > [AWS Command Line Interface](#) > [User Guide for Version 2](#)

AWS Command Line Interface

User Guide for Version 2

- ▶ About the AWS CLI
- ▼ Get started
 - Prerequisites
 - Install/Update**
 - Past releases
 - Build and install from source
 - Amazon ECR Public/Docker
 - Setup
- ▶ Configure the AWS CLI

- [Troubleshooting AWS CLI install and uninstall errors](#)
- [Next steps](#)

AWS CLI install and update instructions

For installation instructions, expand the section for your operating system.

▶ Linux

▶ macOS

▶ 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
```

Recent download history

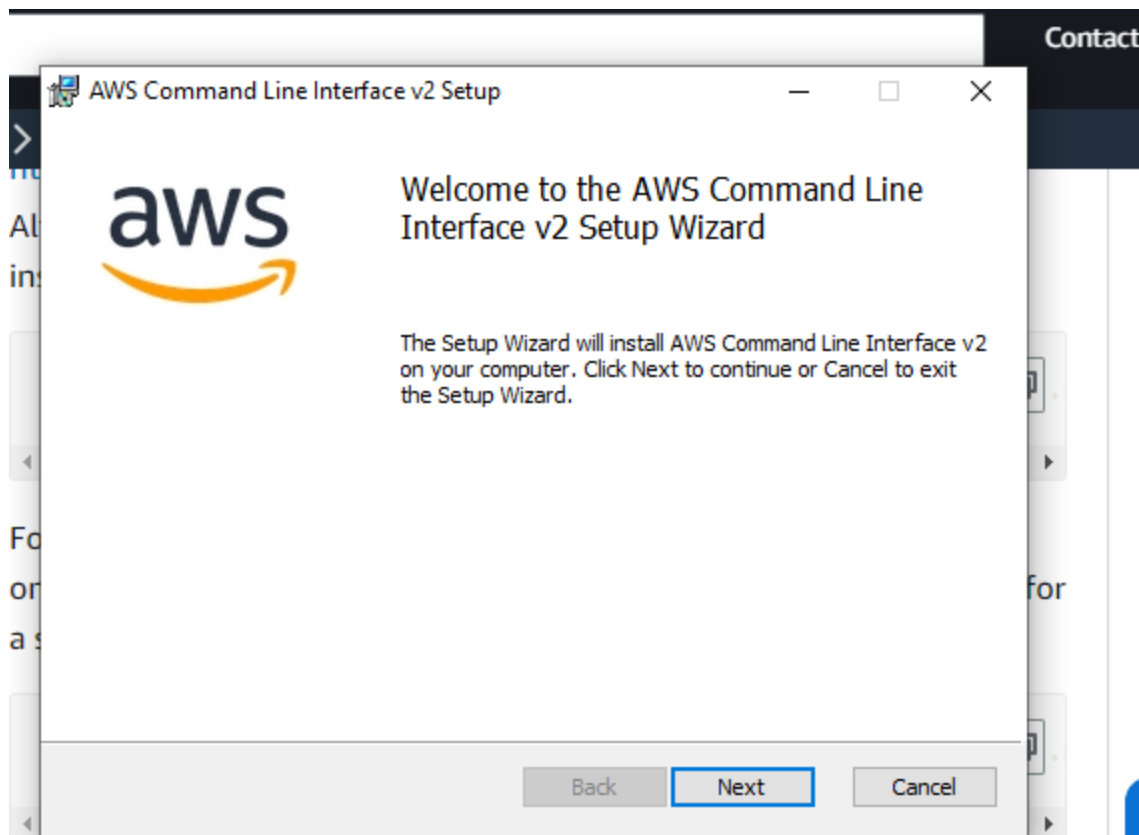
AWSCLIV2.msi
38.7 MB • Done

[Full download history](#)

All Bookmarks

Console

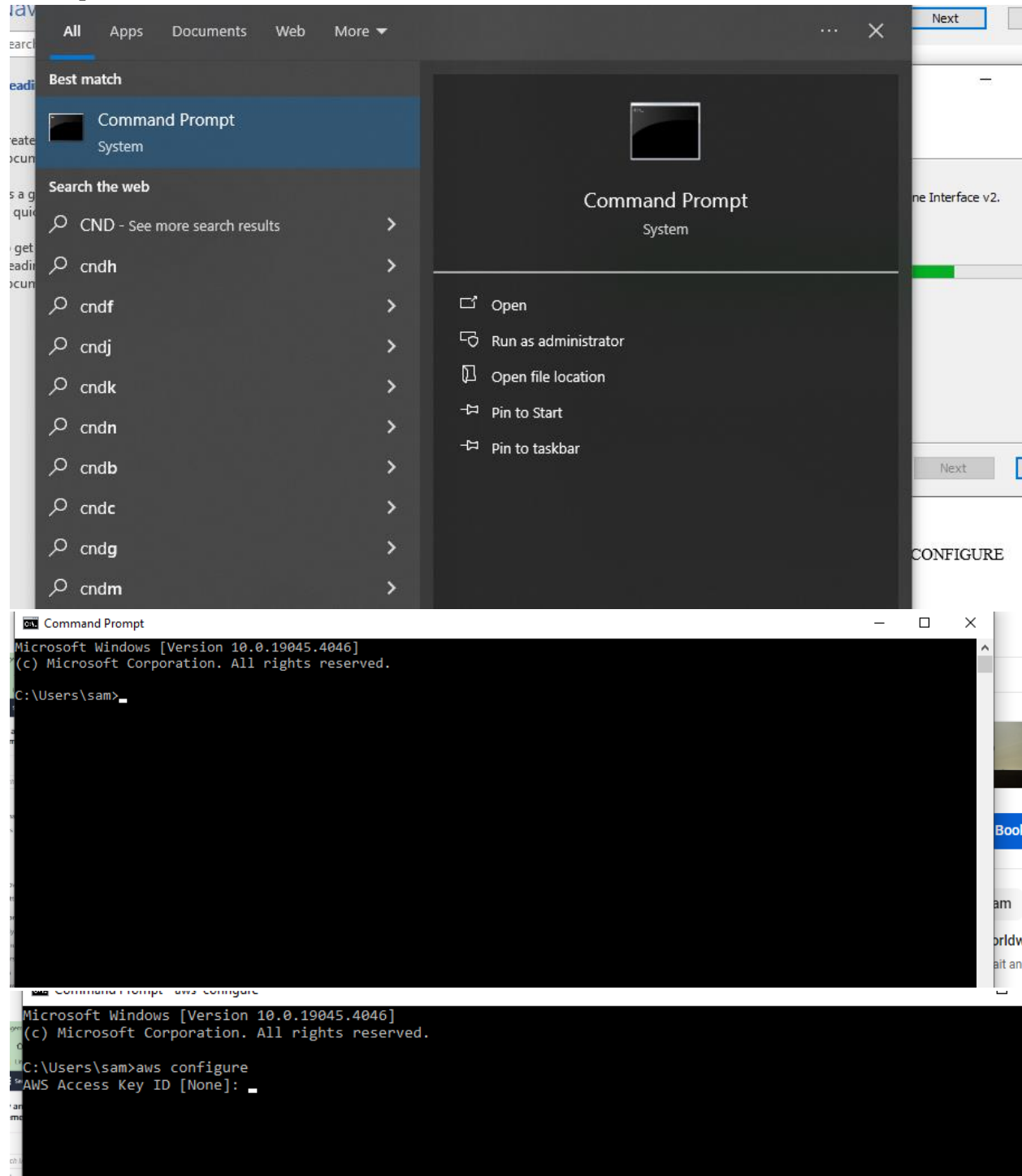
References



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 CREATER USER SECURITUY CEREDITIAL

```
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. Please paste the following key ID: AKIA5FTZBAIDQEK4DTAZ
AWS Access Key ID [None]: AKIA5FTZBAIDQEK4DTAZ
to your content in Amazon S3. Please paste the following key ID: Cxk287C+bEXJDPWxGVULxh21j7yZK3w3+zg0ki3q
AWS Secret Access Key [None]: Cxk287C+bEXJDPWxGVULxh21j7yZK3w3+zg0ki3q
Default region name [None]:
to the Home folder. Please paste the following key ID: Cxk287C+bEXJDPWxGVULxh21j7yZK3w3+zg0ki3q
Default output format [None]:
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 -

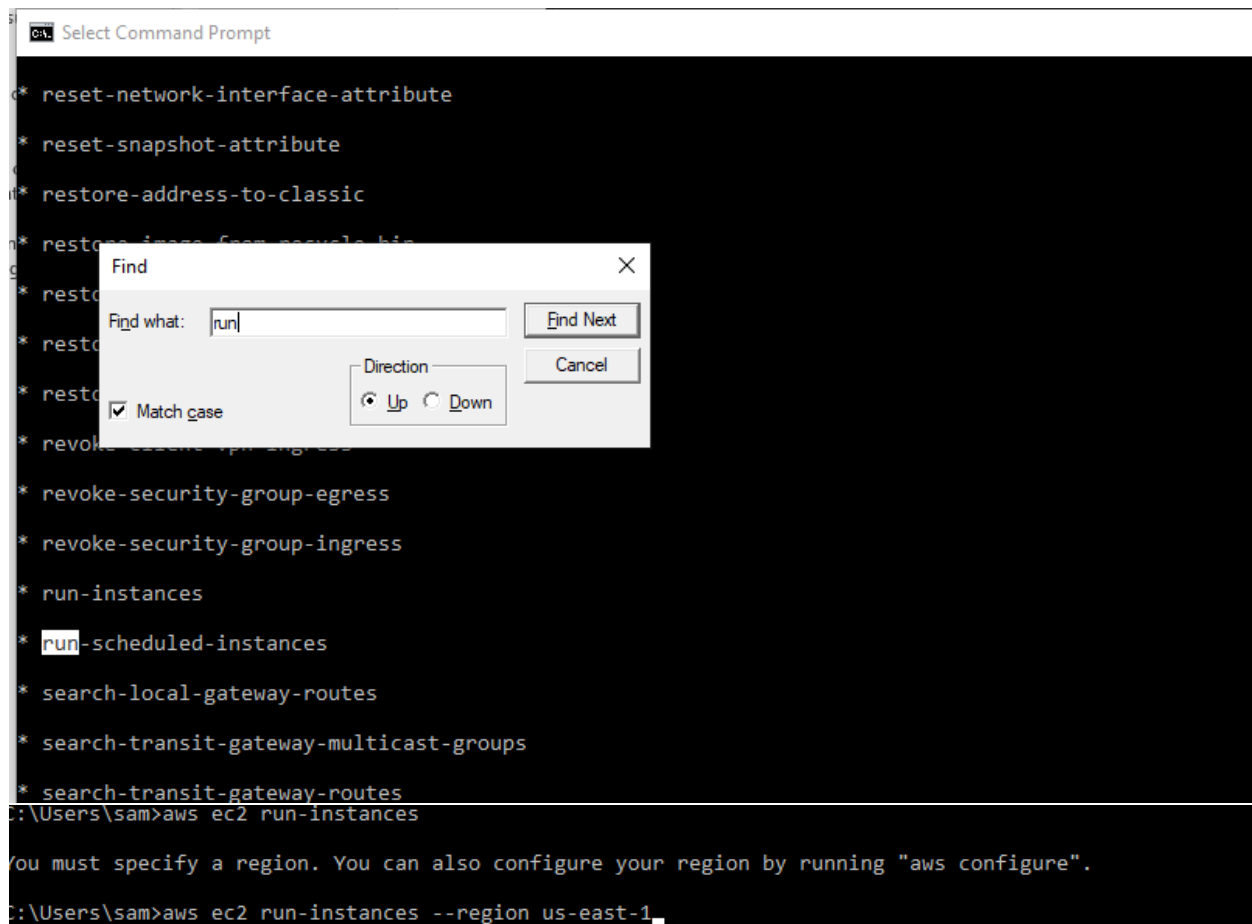
Aws ec2 help

```
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

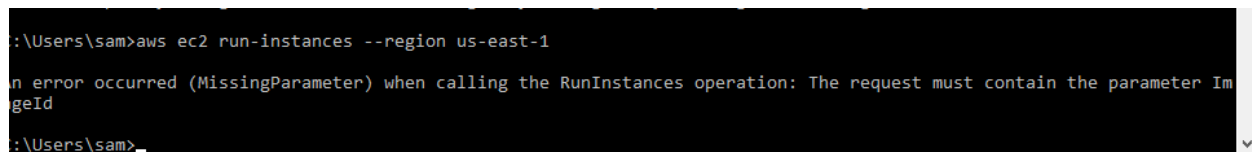
* reset-network-interface-attribute
* reset-snapshot-attribute
* restore-address-to-classic
* restore-image-from-amazon-his
* restore-image-from-snapshot
* restore-image-to-classic
* restore-snapshot-to-amazon-his
* restore-snapshot-to-image
* revoke-security-group-egress
* revoke-security-group-ingress
* 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

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE L

SUS

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

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

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

Free tier eligible

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

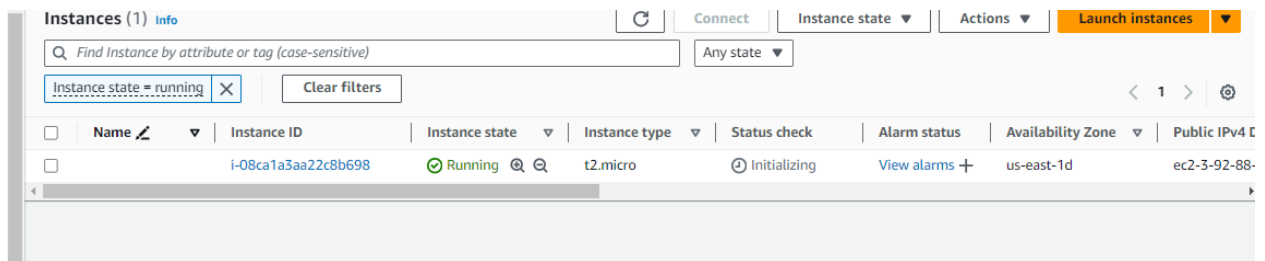
Verified provider

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 and a filter set to 'Instance state = running'. Below the search bar, a table lists the instances. One instance is shown with the ID 'i-08ca1a3aa22c8b698', which is in the 'Running' state. The instance type is 't2.micro' and it's located in the 'us-east-1d' availability zone. The status check shows 'Initializing'.

	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