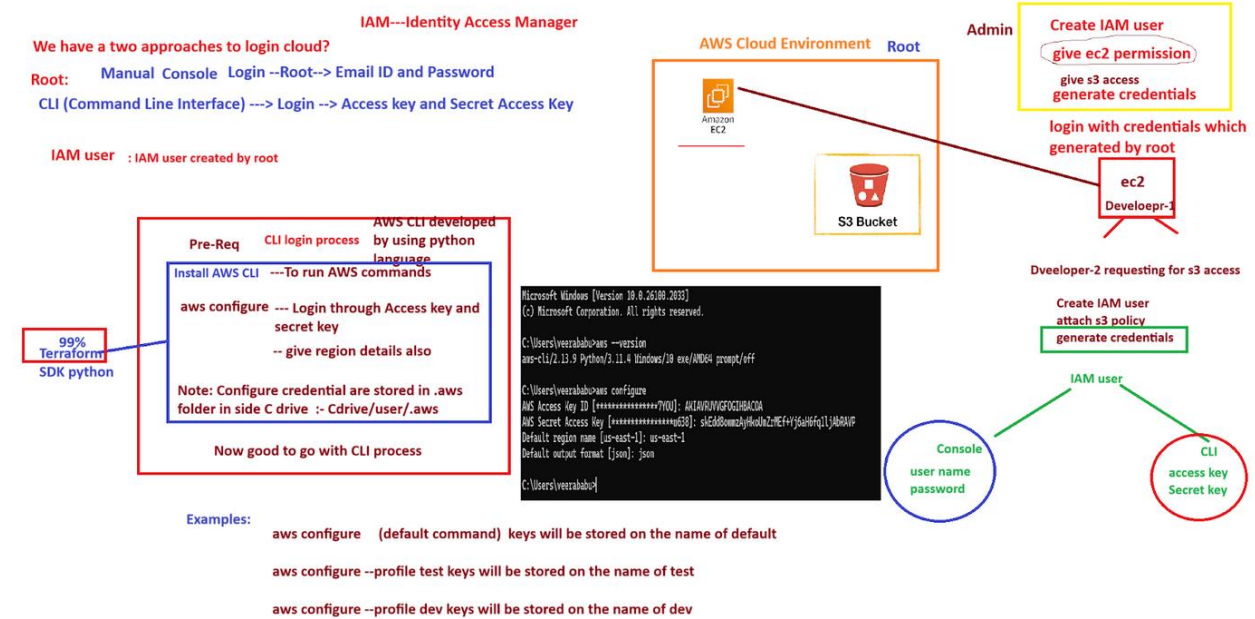


Table of Contents

TABLE OF CONTENTS.....	1
AWS CLI.....	2
<i>Class Room Discussion</i>	2
<i>Screenshots for Windows/Mac/Linux Installation</i>	3
<i>Screenshot for configuring Access Keys for AWS CLI.....</i>	5
<i>Screenshot for Configuring AWS CLI</i>	7
<i>Screenshot for Checking AWS folder on Windows.....</i>	8
<i>Screenshot for accessing another AWS account.....</i>	8
<i>Screenshot for launching a New EC2 Instance using CLI</i>	9

AWS CLI

Class Room Discussion



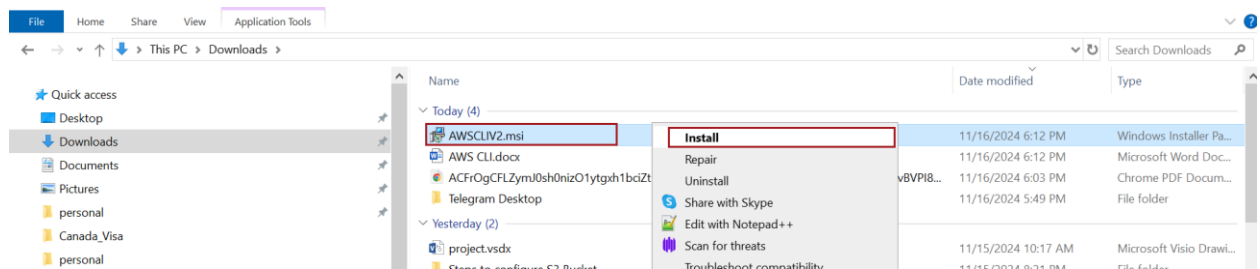
Screenshots for Windows/Mac/Linux Installation

Visit the official AWS CLI download page: [AWS CLI Installation](#)

This screenshot shows the 'AWS CLI install and update instructions' page. The left sidebar contains a navigation menu with links to 'About the AWS CLI', 'Get started', 'Prerequisites', 'Install/Update' (highlighted), 'Past releases', 'Build and install from source', 'Amazon ECR Public/Docker', 'Setup', 'Configure the AWS CLI', 'Authentication and access credentials', and 'Using 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 are three expandable sections for 'Linux', 'macOS', and 'Windows'. To the right of these sections are icons for feedback, a share icon, and the AWS logo. A right-hand sidebar titled 'On this page' lists links for 'AWS CLI install and update instructions', 'Troubleshooting AWS CLI install and uninstall errors', and 'Next steps'.

Download and install the appropriate version for your operating system

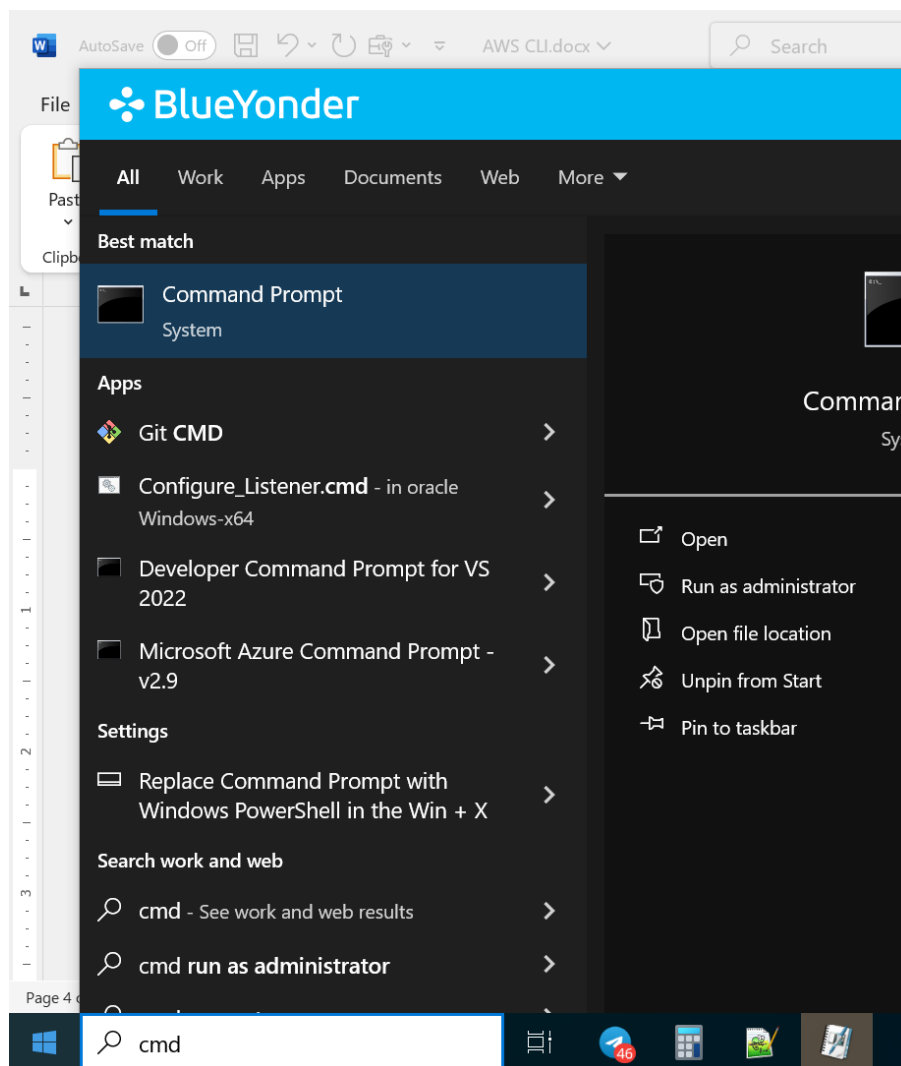
This screenshot shows the 'Install or update the AWS CLI' section of the AWS CLI documentation. The breadcrumb trail at the top reads 'AWS > Documentation > AWS Command Line Interface > User Guide for Version 2'. The left sidebar is identical to the previous screenshot, with 'Install/Update' highlighted. The main content area is titled 'Install or update the AWS CLI' and includes 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](#).' Below this is a numbered list starting with '1. Download and run the AWS CLI MSI installer for Windows (64-bit):' followed by a red-bordered box containing the URL <https://awscli.amazonaws.com/AWSCLIV2.msi>. The text continues: 'Alternatively, you can run the `msiexec` command to run the MSI installer.' The right-hand sidebar is also identical to the previous screenshot.



Proceed with the defaults by clicking "Next."

Verify Installation

Open a terminal or command prompt



Run below command

aws --version

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

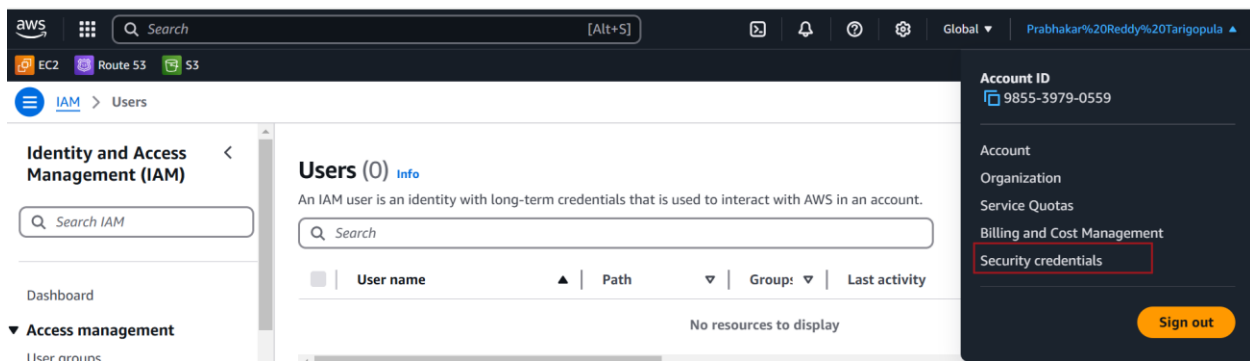
C:\Users\j1013574>aws --version
aws-cli/2.21.3 Python/3.12.6 Windows/10 exe/AMD64

C:\Users\j1013574>
```

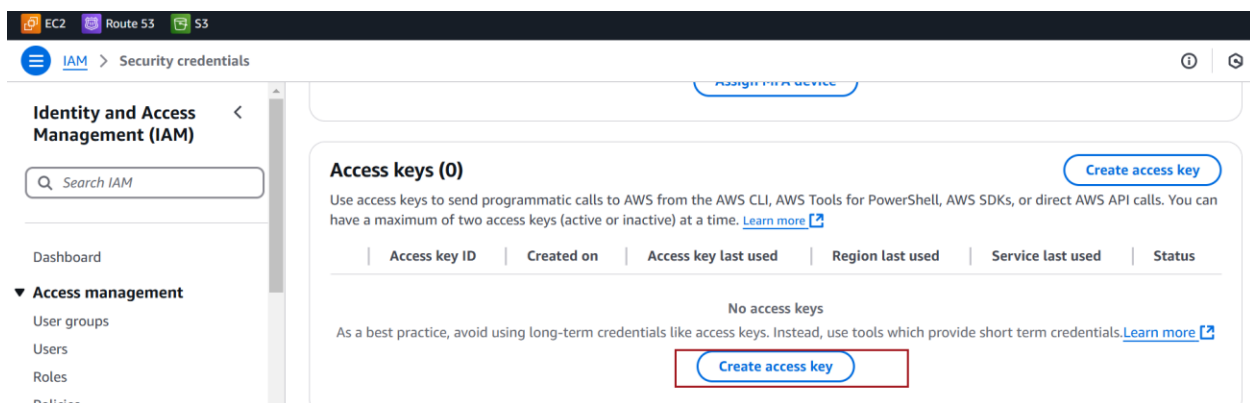
Screenshot for configuring Access Keys for AWS CLI

In the top-right corner of the AWS Management Console, click your **account name** or **account ID** (depending on your setup).

Select **Security Credentials** from the dropdown menu



Scroll down to the **Access keys** section and click **Create access key**



EC2Route S3S3

IAM > Security credentials > Create access key

Step 2

Retrieve access key

⚠

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

Save the .csv file in a secure location and click on Done

IAM > Security credentials > Create access key

🟢

Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

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.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

Download .csv file

Done

Observe that access keys created

EC2Route S3S3

IAM > Security credentials

Identity and Access Management (IAM)

Search IAM

Dashboard

Access keys (1)

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 (active or inactive) at a time. [Learn more](#)

	Access key ID	Created on	Access key last used	Region last used	Service last used	Status
<input type="radio"/>	AKIA6K5V7Y3PSZ5TUSMG	3 minutes ago	None	N/A	N/A	🟢 Active

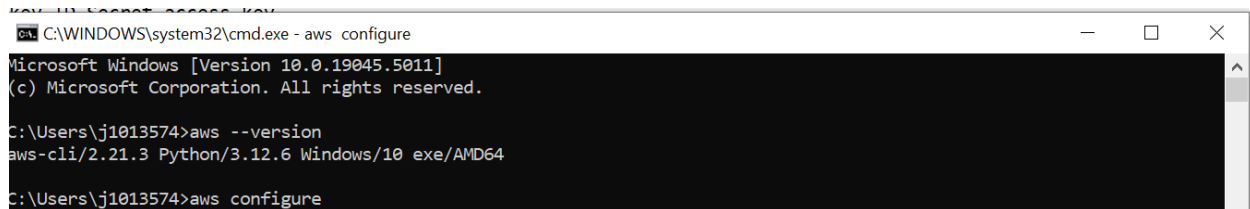
Screenshot for Configuring AWS CLI

Open a terminal or command prompt.

Run aws configure

Enter the following details:

- **AWS Access Key ID:** Enter your Access Key ID.
- **AWS Secret Access Key:** Enter your Secret Access Key.
- **Default region name:** Enter the AWS region (e.g., us-east-1 or ap-south-1).
- **Default output format:** Choose json, table, or text (default is json).

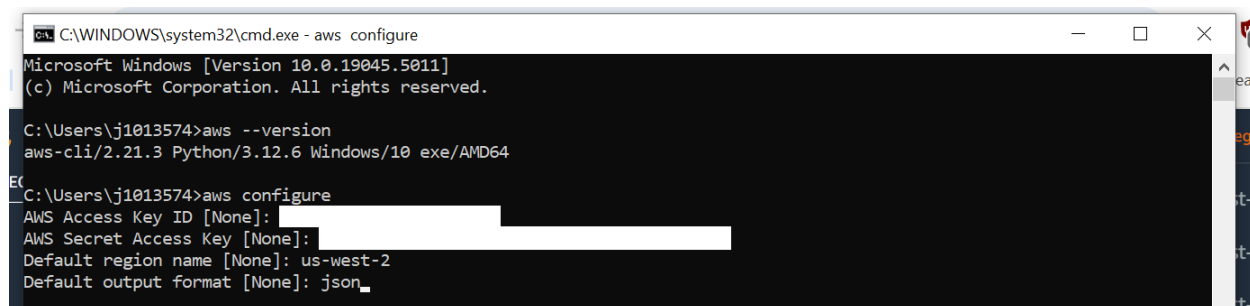
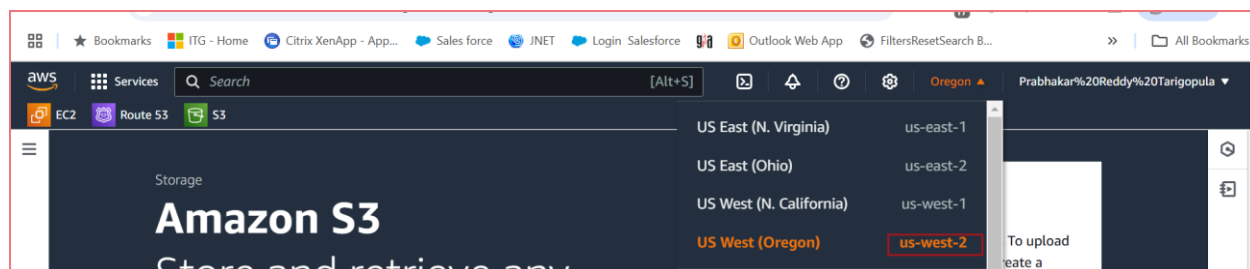


```
C:\WINDOWS\system32\cmd.exe - aws configure
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

C:\Users\j1013574>aws --version
aws-cli/2.21.3 Python/3.12.6 Windows/10 exe/AMD64

C:\Users\j1013574>aws configure
```

Region



```
C:\WINDOWS\system32\cmd.exe - aws configure
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

C:\Users\j1013574>aws --version
aws-cli/2.21.3 Python/3.12.6 Windows/10 exe/AMD64

C:\Users\j1013574>aws configure
AWS Access Key ID [None]: 
AWS Secret Access Key [None]: 
Default region name [None]: us-west-2
Default output format [None]: json_
```

Test Configuration

Try listing your **S3 buckets** to test if AWS CLI can interact with your AWS account

```
C:\WINDOWS\system32\cmd.exe
C:\Users\j1013574>aws s3 ls
2024-11-16 18:53:21 mysourcebuckets3test
C:\Users\j1013574>
```

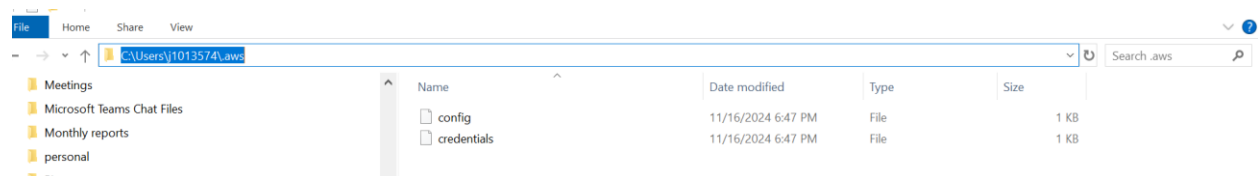
Screenshot for Checking AWS folder on Windows

Open **File Explorer**.

C:\Users\<YourName>\.aws

Inside the .aws folder, you should see two key files:

- **config**: Contains settings like the default region and output format.
- **credentials**: Contains the Access Key ID and Secret Access Key for your configured profiles.



Screenshot for accessing another AWS account

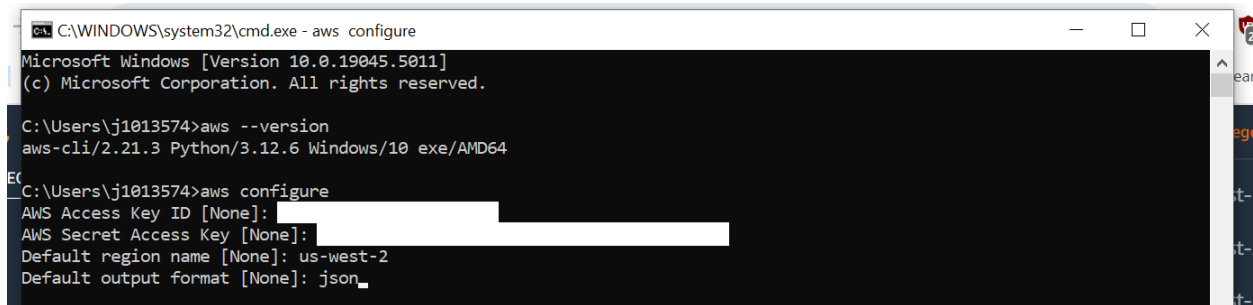
Run the **aws configure** command with a custom profile name for the other account. You can give any name to the profile e.g., another-account

```
C:\WINDOWS\system32\cmd.exe
C:\Users\j1013574>aws configure --profile another-account
```

Enter the credentials for the other account:

- **AWS Access Key ID**: Enter the access key for the other account.
- **AWS Secret Access Key**: Enter the secret key for the other account.

- **Default region name:** You can press **Enter** to accept the default region, or specify a region (e.g., us-west-2).
- **Default output format:** Press **Enter** to accept the default json, or specify your preference (e.g., text or table).



```

C:\WINDOWS\system32\cmd.exe - aws configure
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

C:\Users\j1013574>aws --version
aws-cli/2.21.3 Python/3.12.6 Windows/10 exe/AMD64

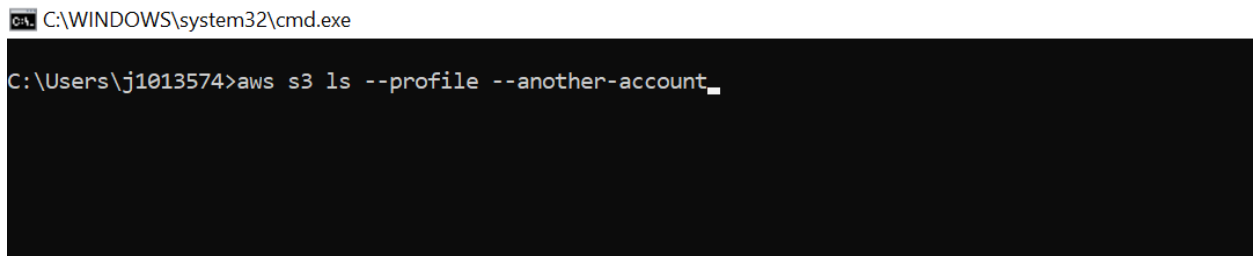
C:\Users\j1013574>aws configure
AWS Access Key ID [None]: 
AWS Secret Access Key [None]: 
Default region name [None]: us-west-2
Default output format [None]: json

```

Use the Other Account via the Named Profile

Once the profile for the other account is configured, you can run AWS CLI commands using the `--profile` option to specify the account you want to use.

`aws s3 ls --profile another-account`



```

C:\WINDOWS\system32\cmd.exe

C:\Users\j1013574>aws s3 ls --profile --another-account

```

Screenshot for launching a New EC2 Instance using CLI

Creating an **EC2 instance** using the AWS CLI involves multiple steps

Prerequisite list required for EC2 instance creation through the AWS CLI:

- **AMI ID:** Amazon Machine Image ID (e.g., ami-12345).
- **Instance Type:** Instance size/type (e.g., t2.micro).
- **Key Pair:** For SSH access to the instance.
- **Security Group:** For firewall rules.

Run the following command to create an EC2 instance:

```
aws ec2 run-instances --image-id ami-22111148 --count 1 --instance-type t1.micro --key-name stage-key --security-groups my-aws-security-group
```

Note - If you are in a default VPC and want AWS to automatically assign the default security group, omit the --security-group-ids parameter entirely:

```
aws ec2 run-instances --image-id ami-01f9f821c37661b97 --count 1 --instance-type t2.micro --key-name newkey
```

```
{
  "PrivateDnsNameOptions": {
    "HostnameType": "ip-name",
    "EnableResourceNameDnsARecord": false,
    "EnableResourceNameDnsAAAARecord": false
  },
  "MaintenanceOptions": {
    "AutoRecovery": "default"
  },
  "CurrentInstanceBootMode": "legacy-bios",
  "InstanceId": "i-0061665e47d72b272",
  "ImageId": "ami-01f9f821c37661b97",
  "State": {
    "Code": 0,
    "Name": "pending"
  },
  "PrivateDnsName": "ip-172-31-20-174.us-west-2.compute.internal",
  "PublicDnsName": "",
  "StateTransitionReason": "",
  "KeyName": "newkey",
  "AmiLaunchIndex": 0,
  "ProductCodes": [],
  "InstanceType": "t2.micro",
  "LaunchTime": "2024-11-17T16:10:56+00:00",
  "Placement": {
    "GroupName": "",
    "Tenancy": "default",
    "AvailabilityZone": "us-west-2a"
  },
  "Monitoring": {
    "State": "disabled"
  },
  "SubnetId": "subnet-06e089686ad696750",
  "VpcId": "vpc-05903633ec364f6f3",
  "PrivateIpAddress": "172.31.20.174"
}
```

C:\Users\j1013574>

To see the list of all EC2 instances in your AWS account, use the following AWS CLI command

```
aws ec2 describe-instances --query "Reservations[*].Instances[*].[InstanceId,State.Name]"
```

```
C:\Users\j1013574>
C:\Users\j1013574>aws ec2 describe-instances --query "Reservations[*].Instances[*].[InstanceId,State.Name]"
[
  [
    [
      "i-0c26062fa455b67cd",
      "terminated"
    ]
  ],
  [
    [
      "i-0061665e47d72b272",
      "terminated"
    ]
  ],
  [
    [
      "i-0a470fd3a13f69bab",
      "terminated"
    ]
  ],
  [
    [
      "i-074b4638527327c1f",
      "running"
    ]
  ]
]
```