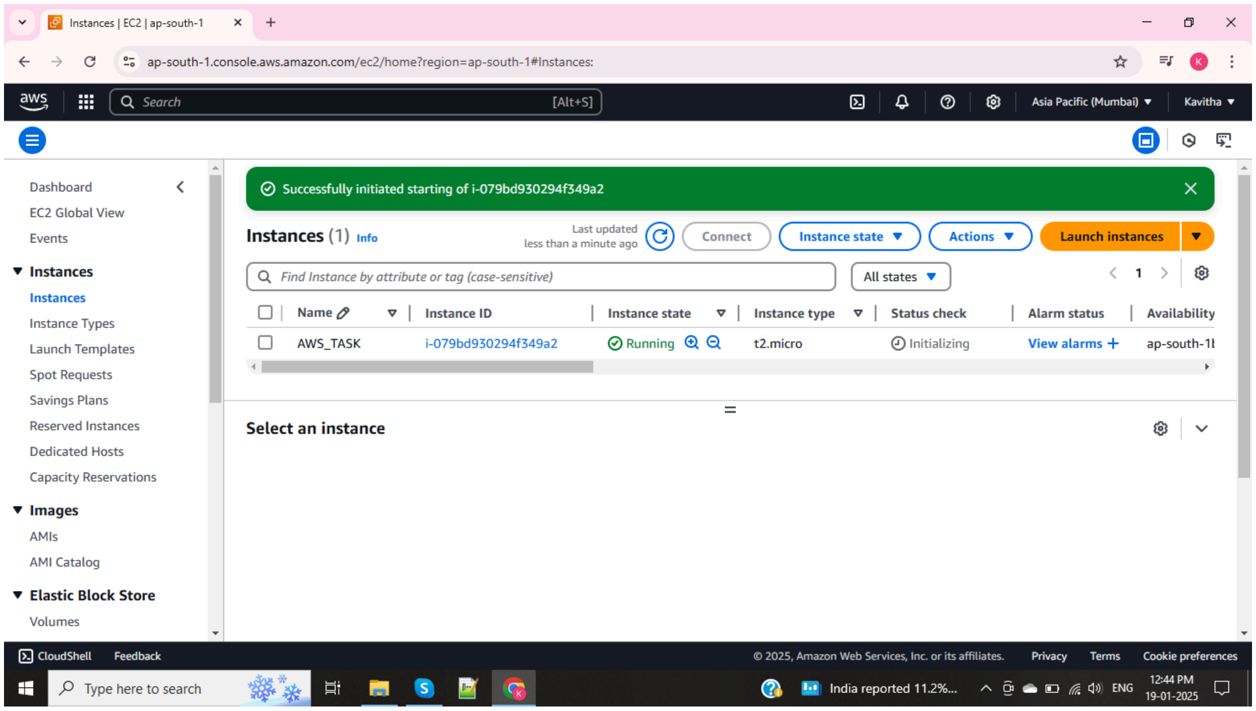
TASK 3 : Execute some more shell scripting commands and creating shell scripting file.

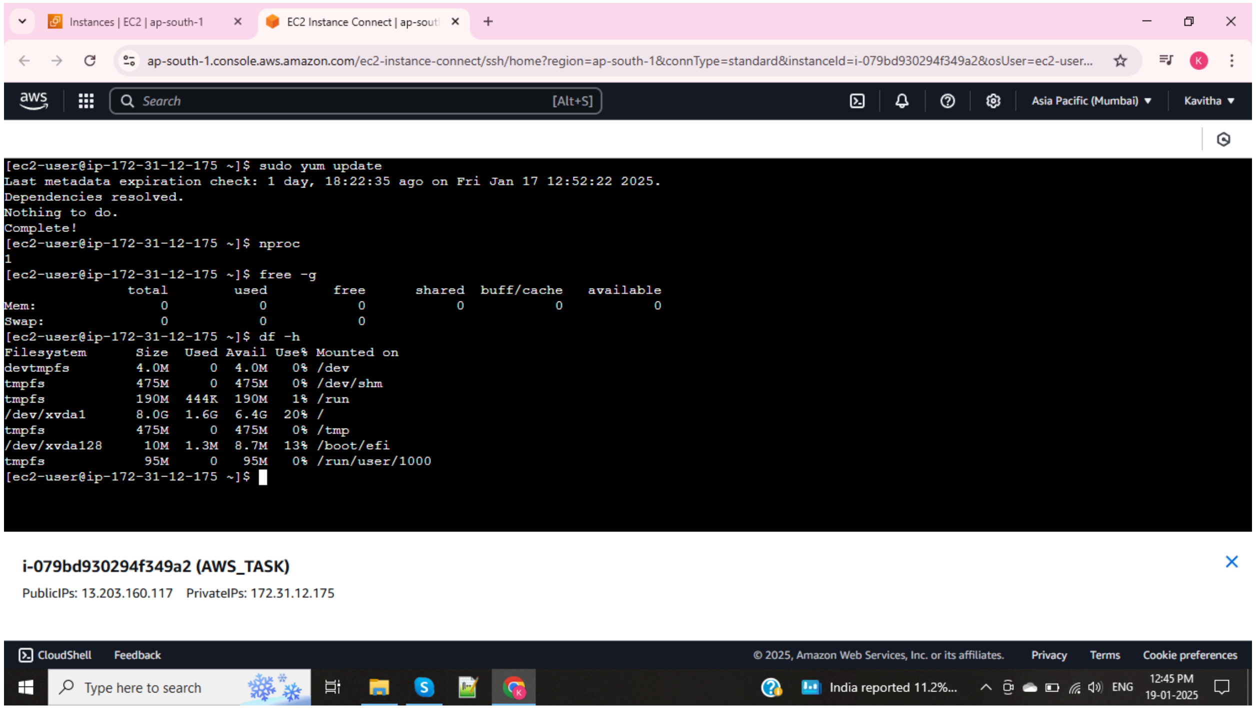
Created an EC2 instance and connected the instance



Successfully initiated starting of i-079bd930294f349a2

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
	AWS_TASK	i-079bd930294f349a2	Running	t2.micro	Initializing	View alarms +	ap-south-1

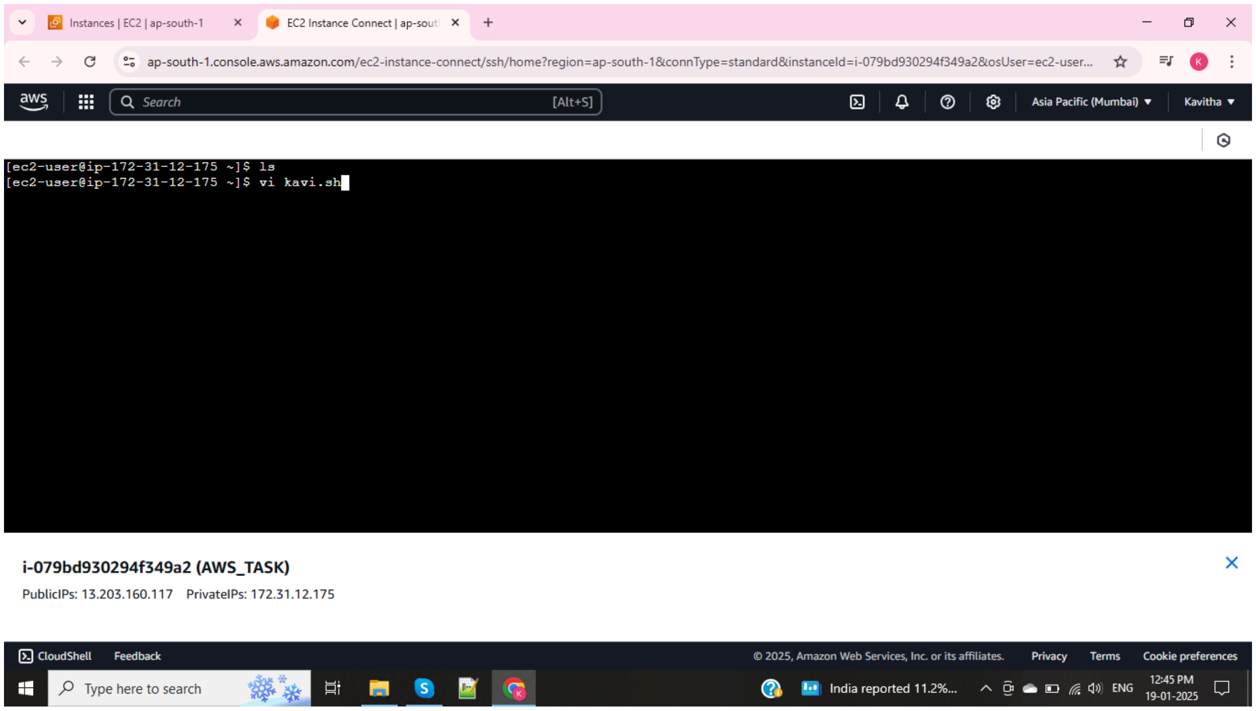
After connecting an EC2 instance --> Update the server and the command is sudo yum update
Nproc = To check the server CPUS, Free -g = To check memory usage, df -h = disk storage



```
[ec2-user@ip-172-31-12-175 ~]$ sudo yum update
Last metadata expiration check: 1 day, 18:22:35 ago on Fri Jan 17 12:52:22 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-12-175 ~]$ nproc
1
[ec2-user@ip-172-31-12-175 ~]$ free -g
               total        used        free      shared  buff/cache   available
Mem:               0           0           0           0           0           0
Swap:              0           0           0
[ec2-user@ip-172-31-12-175 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M     0  4.0M   0% /dev
tmpfs           475M     0  475M   0% /dev/shm
tmpfs           190M  444K  190M   1% /run
/dev/xvda1      8.0G  1.6G  6.4G  20% /
tmpfs           475M     0  475M   0% /tmp
/dev/xvda128     10M  1.3M  8.7M  13% /boot/efi
tmpfs            95M     0   95M   0% /run/user/1000
[ec2-user@ip-172-31-12-175 ~]$
```

i-079bd930294f349a2 (AWS_TASK)
PublicIPs: 13.203.160.117 PrivateIPs: 172.31.12.175

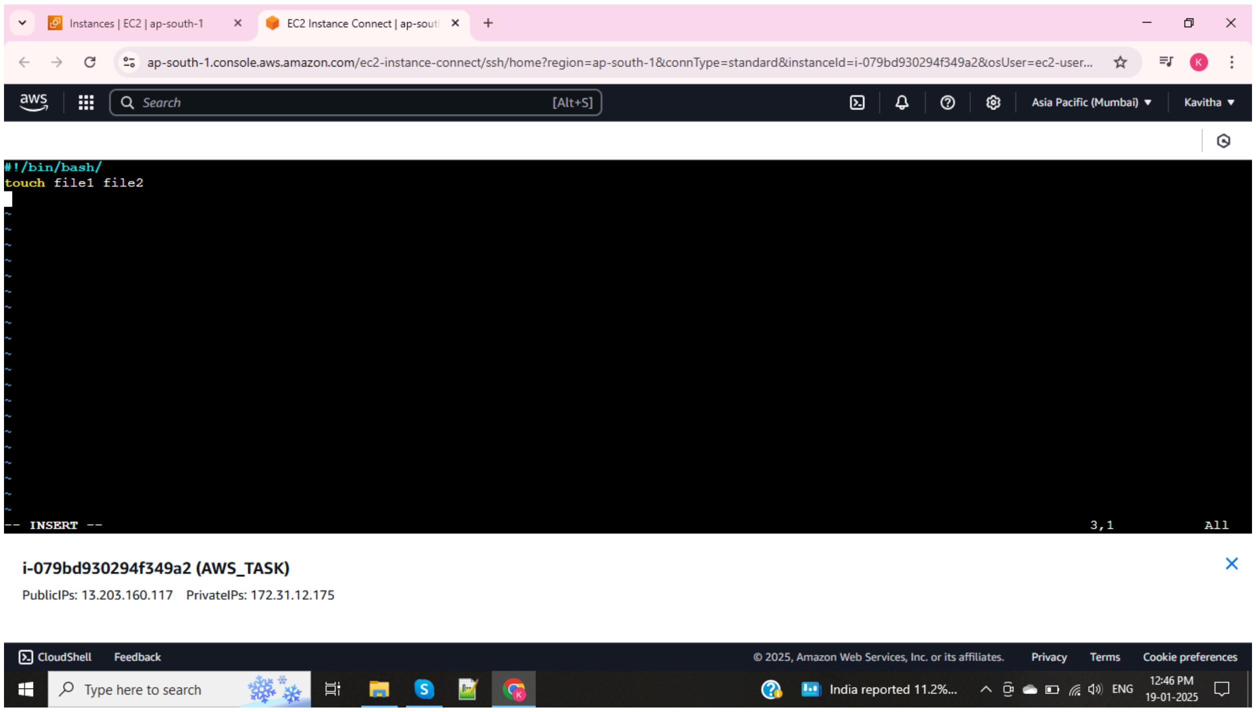
To create Shell Scripting file use filename with **.sh** and to edit the file = **vi kavi .sh**



```
[ec2-user@ip-172-31-12-175 ~]$ ls
[ec2-user@ip-172-31-12-175 ~]$ vi kavi.sh
```

i-079bd930294f349a2 (AWS_TASK)
PublicIPs: 13.203.160.117 PrivateIPs: 172.31.12.175

For shell scripting file always use **#! /bin/bash/** --> We have multiple work to do means that time we use shell scripting file to reduce the man power and do all the task in single file. Creating 2 files.

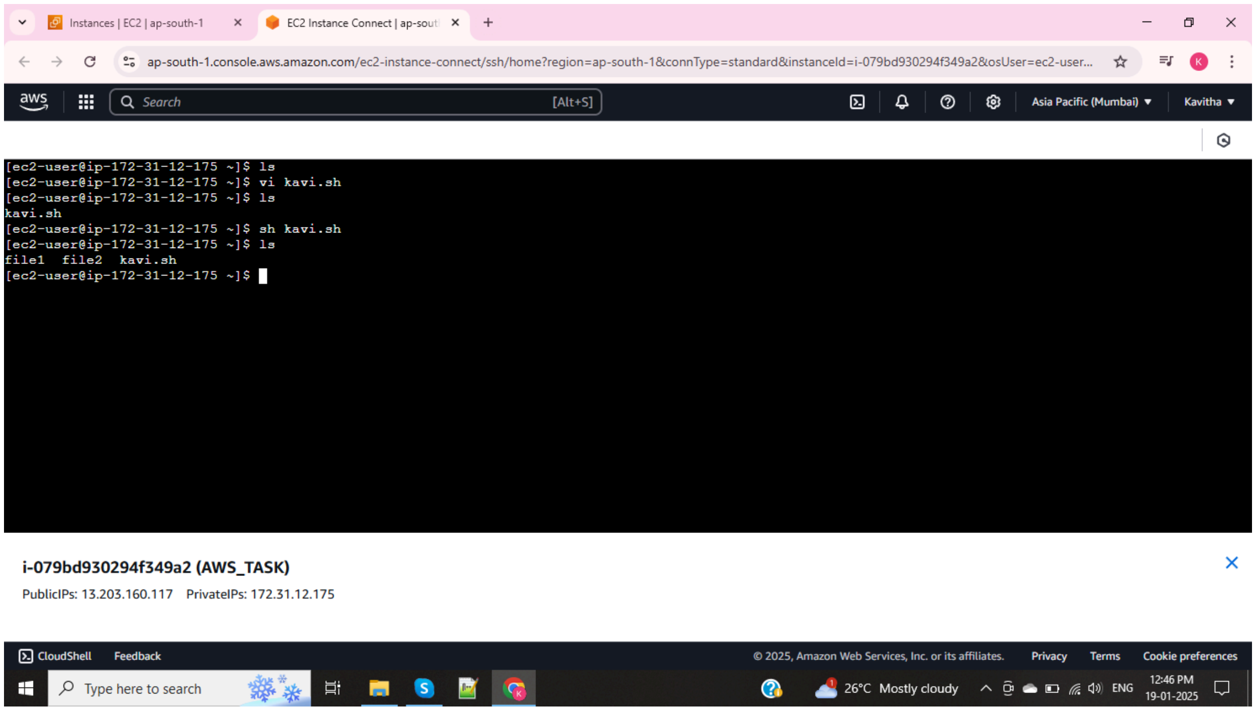


```
#!/bin/bash/
touch file1 file2

~
~
-- INSERT --                                                    3,1           All
```

i-079bd930294f349a2 (AWS_TASK)
PublicIPs: 13.203.160.117 PrivateIPs: 172.31.12.175

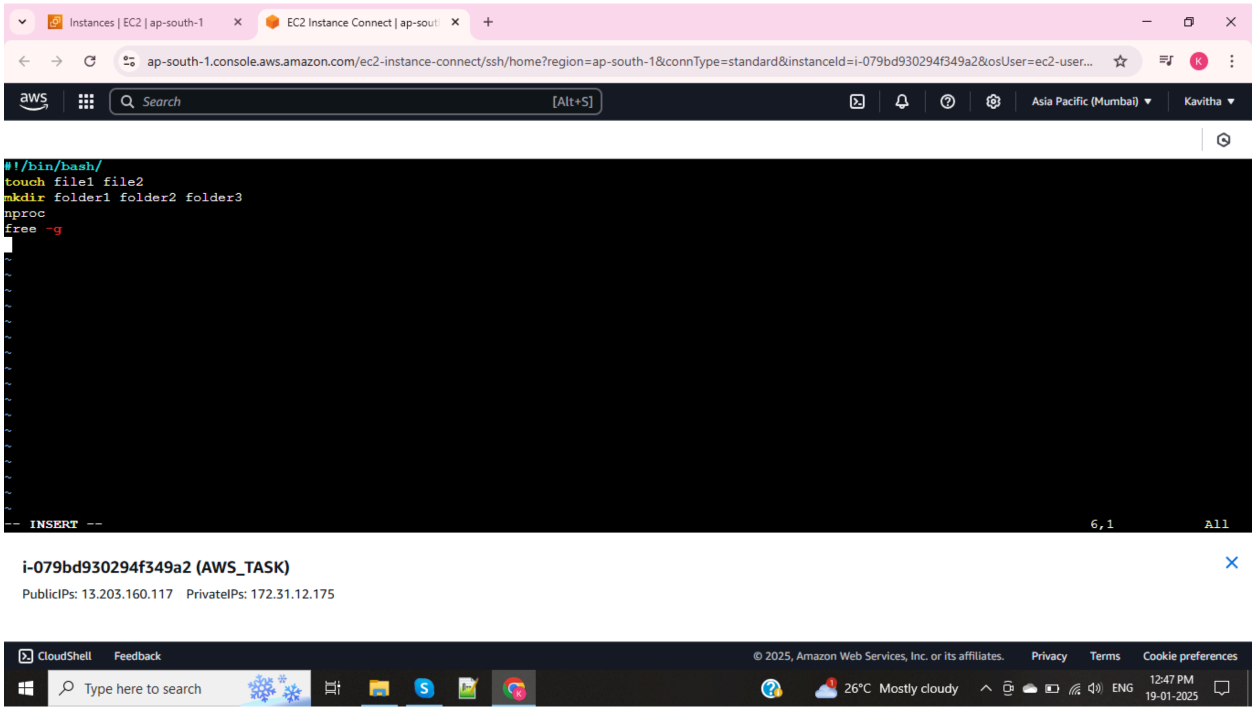
To execute the SHELL SCRIPTING file use **sh kavi .sh** --> LS to check file was created or not



```
[ec2-user@ip-172-31-12-175 ~]$ ls
[ec2-user@ip-172-31-12-175 ~]$ vi kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ ls
kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ sh kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ ls
file1  file2  kavi.sh
[ec2-user@ip-172-31-12-175 ~]$
```

i-079bd930294f349a2 (AWS_TASK)
PublicIPs: 13.203.160.117   PrivateIPs: 172.31.12.175

Go into the same file edit and add some more information whether file is working or not Giving shell commands : nproc and free -g.

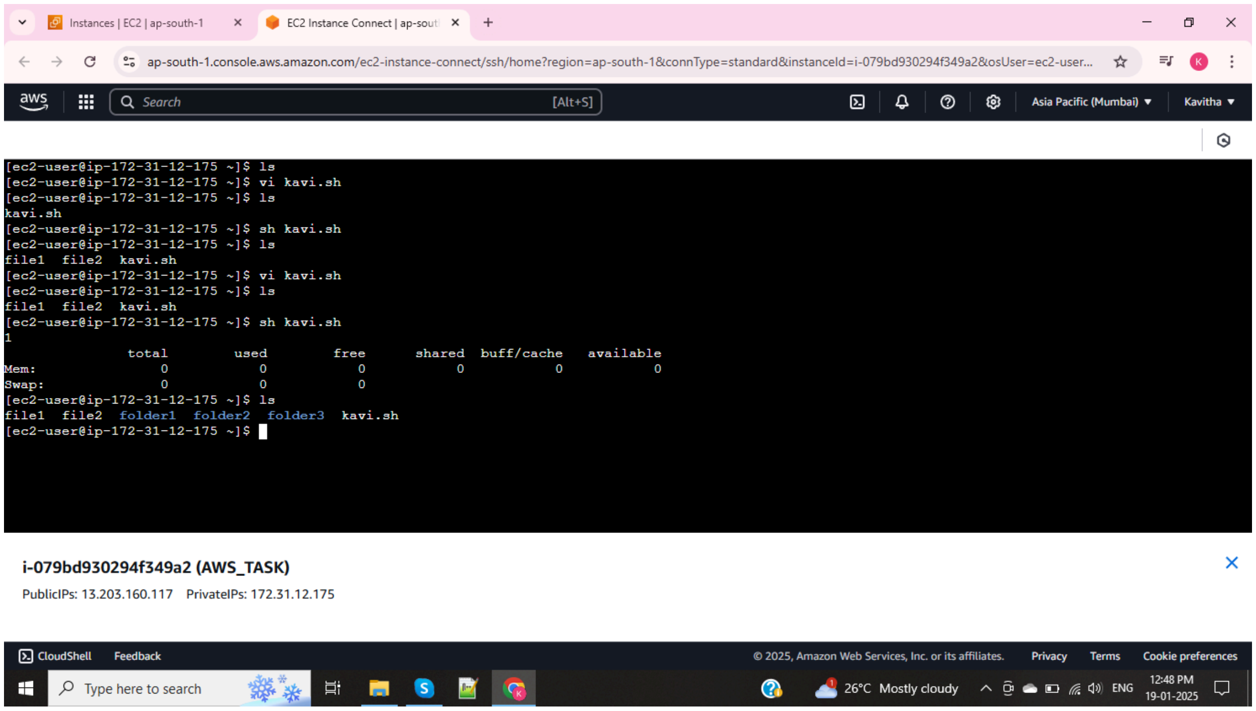


```
#!/bin/bash/
touch file1 file2
mkdir folder1 folder2 folder3
nproc
free -g

~
~
-- INSERT --                                                     6,1           All
```

i-079bd930294f349a2 (AWS_TASK)
PublicIPs: 13.203.160.117   PrivateIPs: 172.31.12.175

First execute the file and check with **ls** and the output for first command is 1 and output for second command then 3 folders are created at a time.

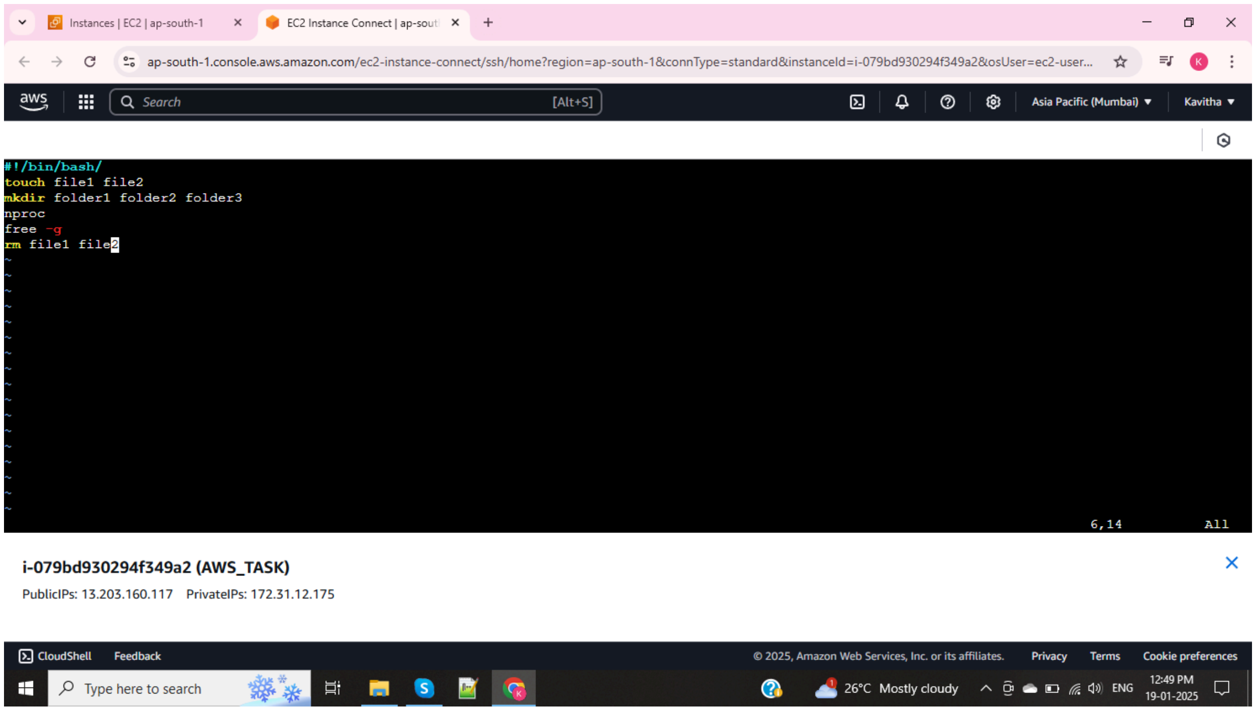


```
[ec2-user@ip-172-31-12-175 ~]$ ls
[ec2-user@ip-172-31-12-175 ~]$ vi kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ ls
kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ sh kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ ls
file1  file2  kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ vi kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ ls
file1  file2  kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ sh kavi.sh
1
               total        used        free      shared  buff/cache   available
Mem:               0           0           0           0           0           0
Swap:              0           0           0
[ec2-user@ip-172-31-12-175 ~]$ ls
file1  file2  folder1  folder2  folder3  kavi.sh
[ec2-user@ip-172-31-12-175 ~]$
```

i-079bd930294f349a2 (AWS_TASK)

PublicIPs: 13.203.160.117   PrivateIPs: 172.31.12.175

Now, To remove the file use **rm file1 file2**

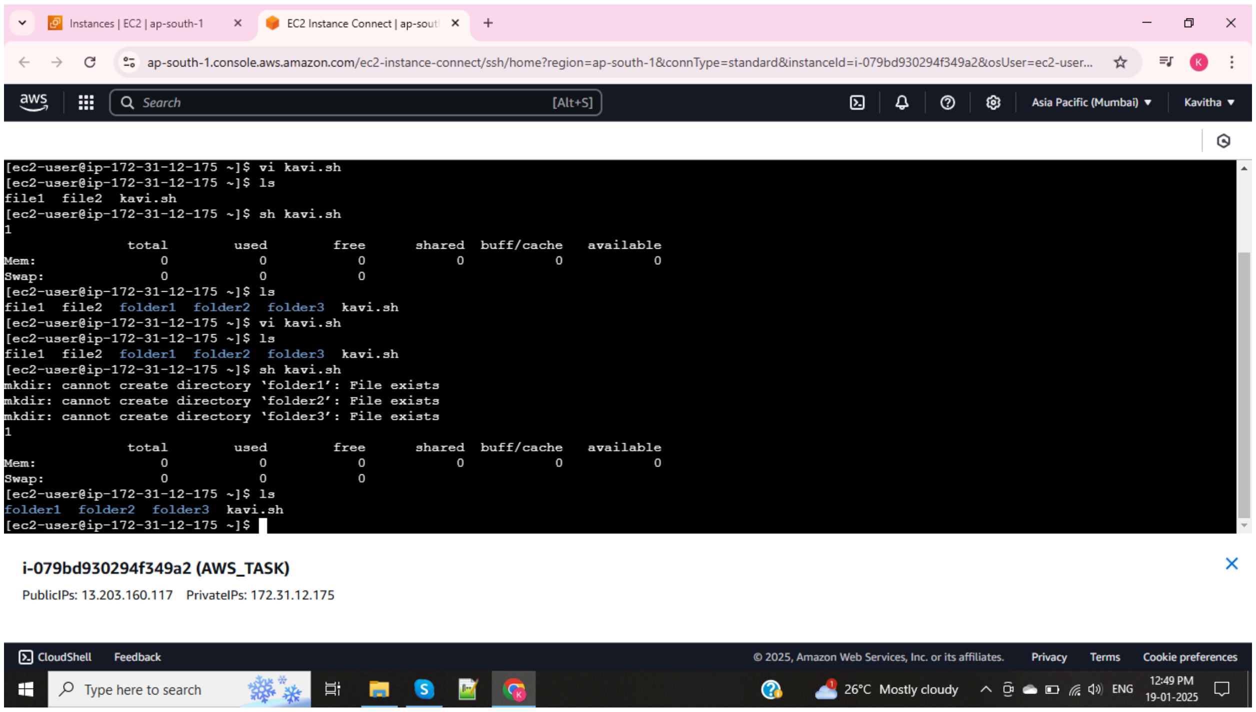


```
#!/bin/bash/
touch file1 file2
mkdir folder1 folder2 folder3
nproc
free -g
rm file1 file2
~
~
~
                                                              6,14          All
```

i-079bd930294f349a2 (AWS_TASK)

PublicIPs: 13.203.160.117   PrivateIPs: 172.31.12.175

After execute the file checking with **LS** command created files are deleted and output

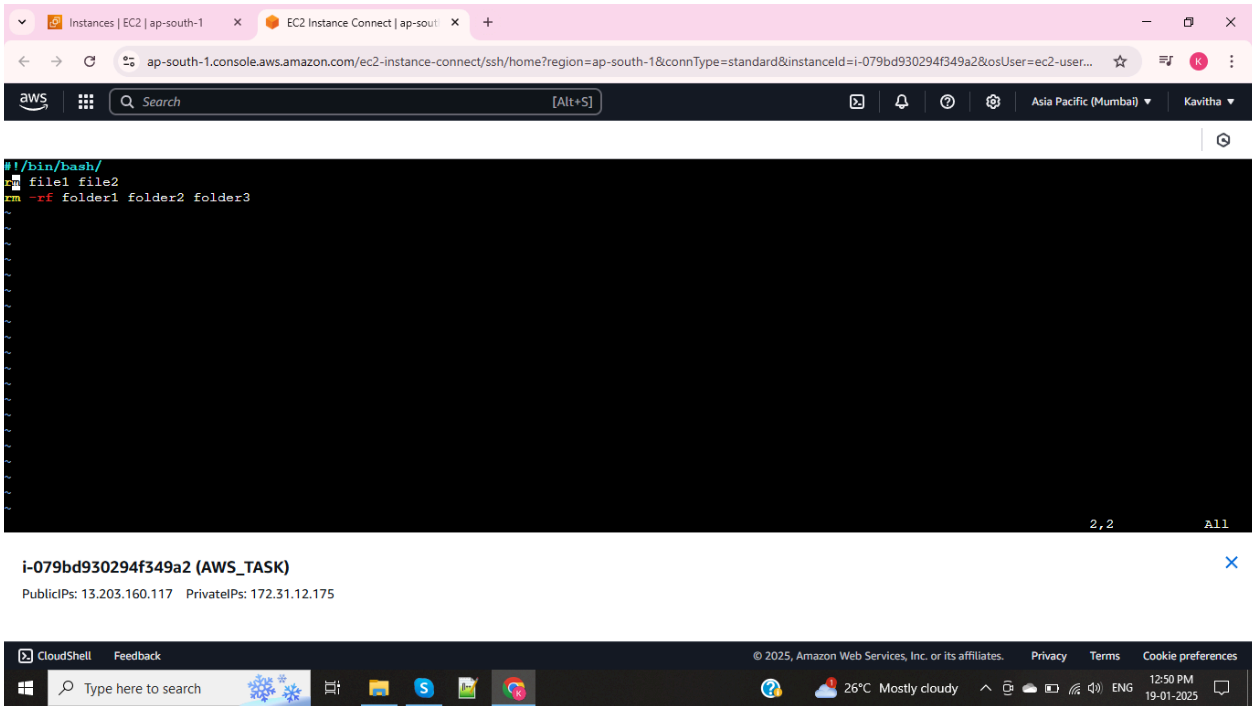


```
[ec2-user@ip-172-31-12-175 ~]$ vi kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ ls
file1  file2  kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ sh kavi.sh
1
               total        used        free      shared  buff/cache   available
Mem:               0           0           0           0           0           0
Swap:              0           0           0
[ec2-user@ip-172-31-12-175 ~]$ ls
file1  file2  folder1  folder2  folder3  kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ vi kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ ls
file1  file2  folder1  folder2  folder3  kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ sh kavi.sh
mkdir: cannot create directory 'folder1': File exists
mkdir: cannot create directory 'folder2': File exists
mkdir: cannot create directory 'folder3': File exists
1
               total        used        free      shared  buff/cache   available
Mem:               0           0           0           0           0           0
Swap:              0           0           0
[ec2-user@ip-172-31-12-175 ~]$ ls
folder1  folder2  folder3  kavi.sh
[ec2-user@ip-172-31-12-175 ~]$
```

i-079bd930294f349a2 (AWS_TASK)

PublicIPs: 13.203.160.117   PrivateIPs: 172.31.12.175

For deleting folders use **rm -rf folder names**

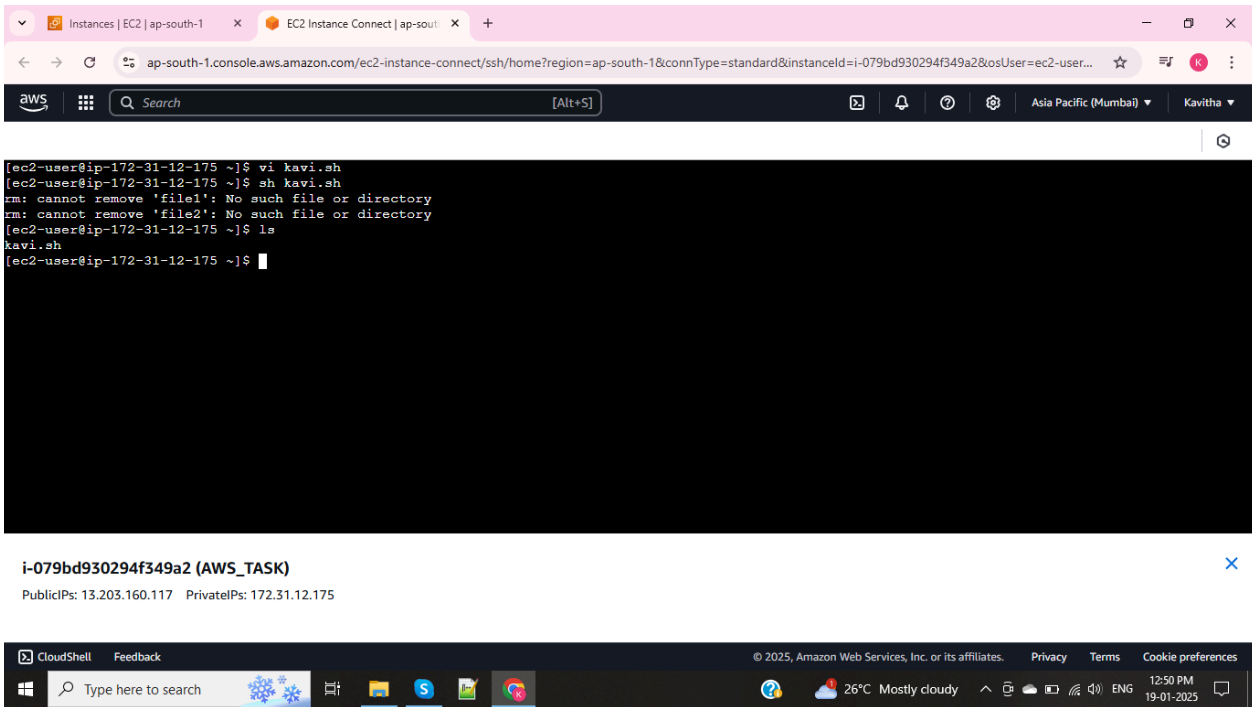


```
#!/bin/bash/
rm file1 file2
rm -rf folder1 folder2 folder3
~
~
~
                                                        2,2           All
```

i-079bd930294f349a2 (AWS_TASK)

PublicIPs: 13.203.160.117   PrivateIPs: 172.31.12.175

Output of folders removed

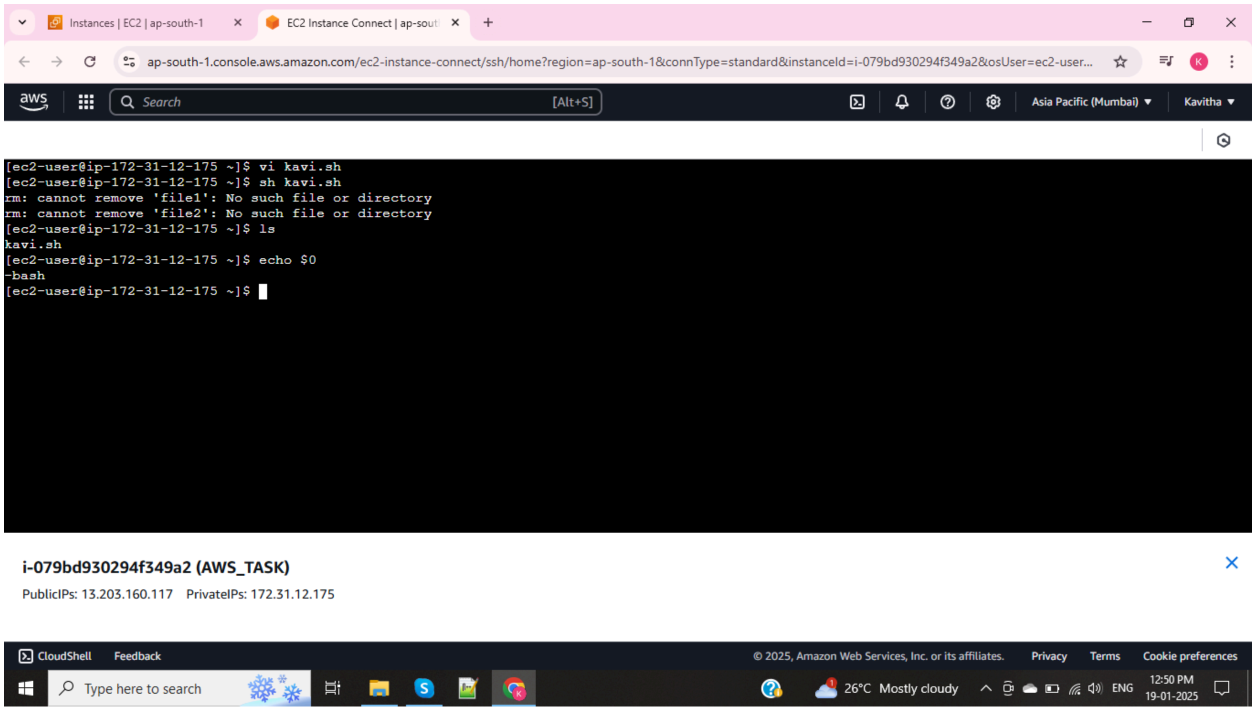


```
[ec2-user@ip-172-31-12-175 ~]$ vi kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ sh kavi.sh
rm: cannot remove 'file1': No such file or directory
rm: cannot remove 'file2': No such file or directory
[ec2-user@ip-172-31-12-175 ~]$ ls
kavi.sh
[ec2-user@ip-172-31-12-175 ~]$
```

i-079bd930294f349a2 (AWS_TASK)

PublicIPs: 13.203.160.117 PrivateIPs: 172.31.12.175

To check the execute type use **echo $0** and it shows **bash**



```
[ec2-user@ip-172-31-12-175 ~]$ vi kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ sh kavi.sh
rm: cannot remove 'file1': No such file or directory
rm: cannot remove 'file2': No such file or directory
[ec2-user@ip-172-31-12-175 ~]$ ls
kavi.sh
[ec2-user@ip-172-31-12-175 ~]$ echo $0
-bash
[ec2-user@ip-172-31-12-175 ~]$
```

i-079bd930294f349a2 (AWS_TASK)

PublicIPs: 13.203.160.117 PrivateIPs: 172.31.12.175

An **EC2 instance** (Elastic Compute Cloud instance) is a virtual server hosted on Amazon Web Services (**AWS**) that provides scalable compute capacity in the cloud. It enables users to run applications, host websites, process data, or perform other computing tasks without needing physical servers.

Key Features of EC2

1. **Elasticity**:
   - Quickly scale instances up or down.
2. **Pay-as-you-go**:
   - Pay only for the resources you consume.
3. **High Availability**:
   - Choose multiple Availability Zones for fault tolerance.
4. **Instance Types**:
   - Select from a range of instance types optimized for compute, memory, or storage.
5. **Integration**:
   - Seamless integration with other AWS services like S3, RDS, and CloudWatch.

Types of EC2 Instances

AWS offers a variety of EC2 instance types to cater to different workloads. These are categorized based on their purpose, such as compute, memory, storage, and GPU optimization. Below are the major categories:

1. General Purpose Instances

- **Use Case**: Balanced compute, memory, and networking. Ideal for web servers, application servers, and development environments.
- **Examples**:
  - **t2, t3, t4g**: Burstable performance.
  - **m5, m6g**: General-purpose workloads.

2. Compute-Optimized Instances

- **Use Case**: Workloads that require high-performance processors, such as high-performance computing (HPC), gaming, and batch processing.
- **Examples**:
  - **c5, c6g**: Compute-intensive applications.

3. Memory-Optimized Instances

- **Use Case**: Applications requiring high memory, such as in-memory databases, real-time big data processing, and high-performance analytics.
- **Examples**:
  - **r5, r6g**: Memory-intensive applications.
  - **x1e, x2gd**: Extreme memory needs.

4. Storage-Optimized Instances

- **Use Case**: High disk throughput or IOPS requirements, such as NoSQL databases, data warehousing, and log processing.
- **Examples**:
  - **i3, i4i**: High IOPS storage.
  - **d2, d3**: Dense storage for big data.

5. Accelerated Computing Instances

- **Use Case**: Applications requiring hardware acceleration, such as machine learning (ML), graphics rendering, and scientific simulations.
- **Examples**:
  - **p4, p5**: GPU-optimized for ML and AI.
  - **g4, g5**: Graphics-heavy applications like video encoding.
  - **f1**: FPGA for hardware acceleration.

6. High Memory Instances

- **Use Case**: Large in-memory databases like SAP HANA.
- **Examples**: **u-6tb1, u-9tb1, u-12tb1**.

7. Bare Metal Instances

- **Use Case**: Applications requiring direct access to the underlying hardware for special workloads or compliance.
- **Examples**: **m5.metal, c5.metal, r5.metal**.

8. Mac Instances

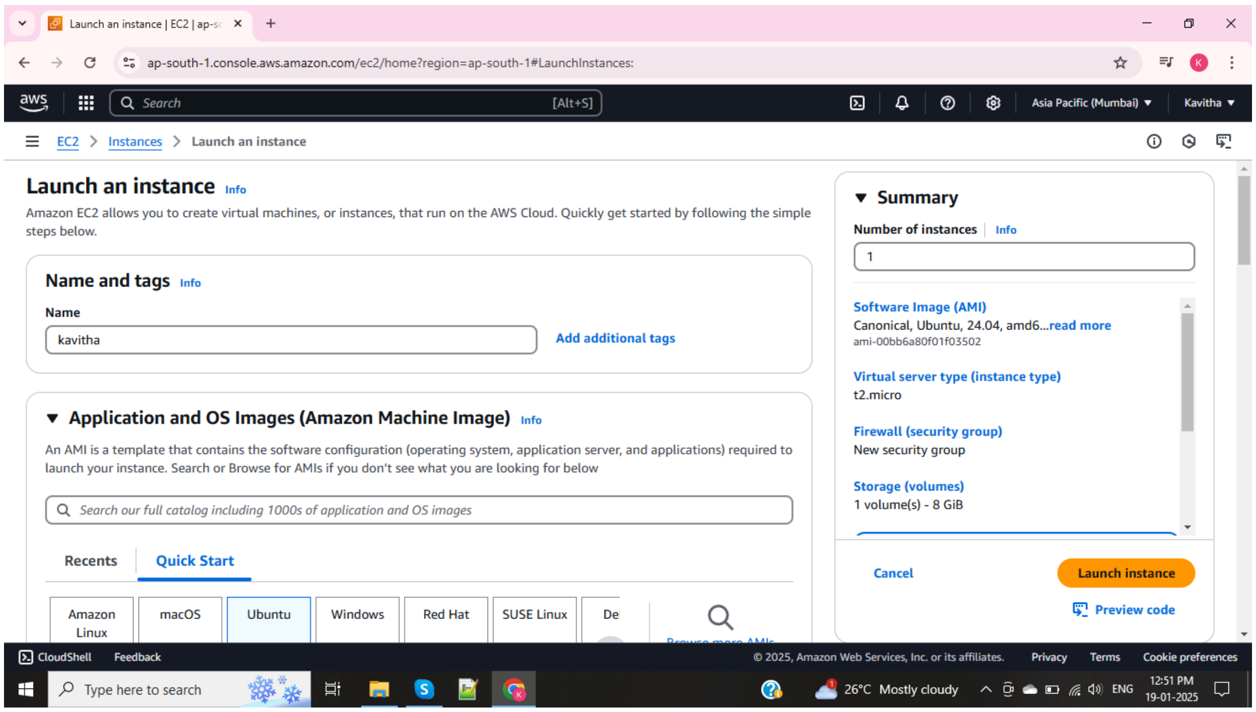
- **Use Case**: Running macOS for developing and testing Apple applications.
- **Examples**: **mac1.metal**.

TASK 4 : How to Create an EC2 Instance

1. **Login to AWS Console**:

2. **Navigate to EC2 Dashboard**: Click EC2.

3. **Launch Instance**:Click Launch Instance on the EC2 dashboard. Giving name for the EC2 instance.



Launch an instance | EC2 | ap-so

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: kavitha

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | De

Summary

Number of instances Info: 1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-00bb6a80f01f03502

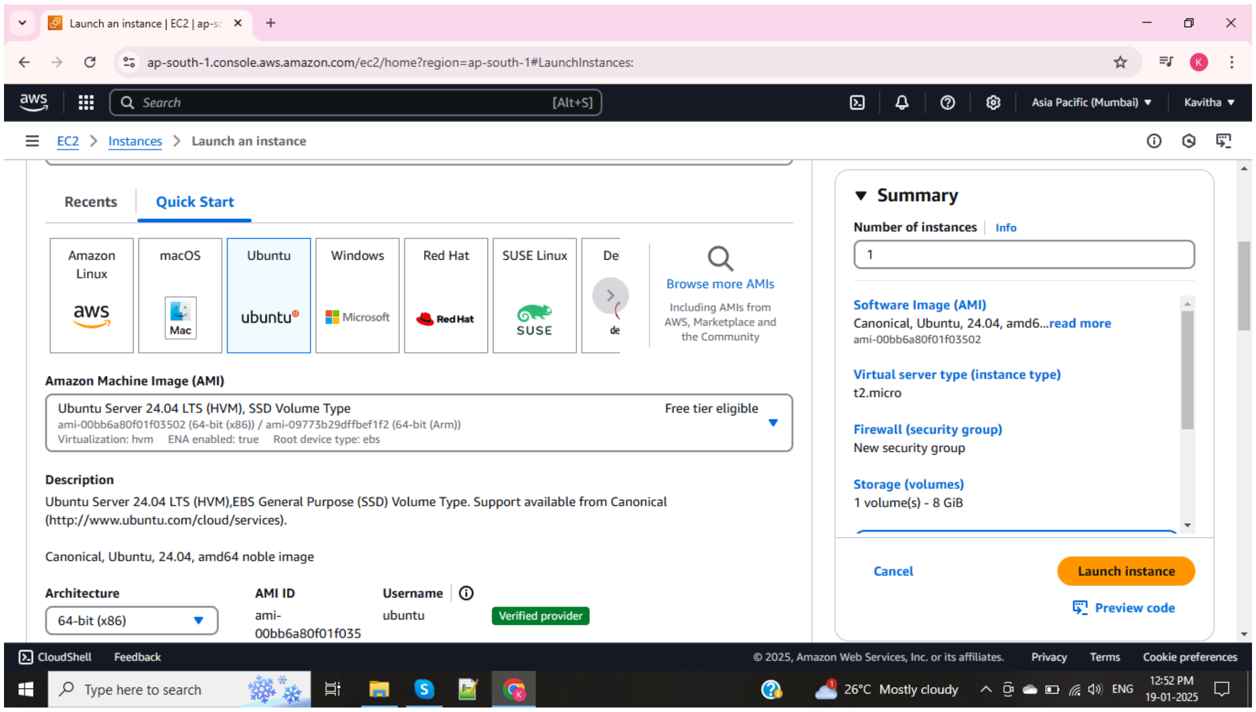
Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel | Launch instance | Preview code

4. **Select AMI**:Choose an Amazon Machine Image (**AMI**) like Amazon Linux, Ubuntu, or Windows.


Launch an instance | EC2 | ap-sc

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

Search [Alt+S] Asia Pacific (Mumbai) Kavitha

EC2 > Instances > Launch an instance

Recents Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | De

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)
Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-00bb6a80f01f03502 (64-bit (x86)) / ami-09773b29dffbef1f2 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture	AMI ID	Username	
64-bit (x86)	ami-00bb6a80f01f035	ubuntu	Verified provider

Summary
Number of instances Info
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-00bb6a80f01f03502

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

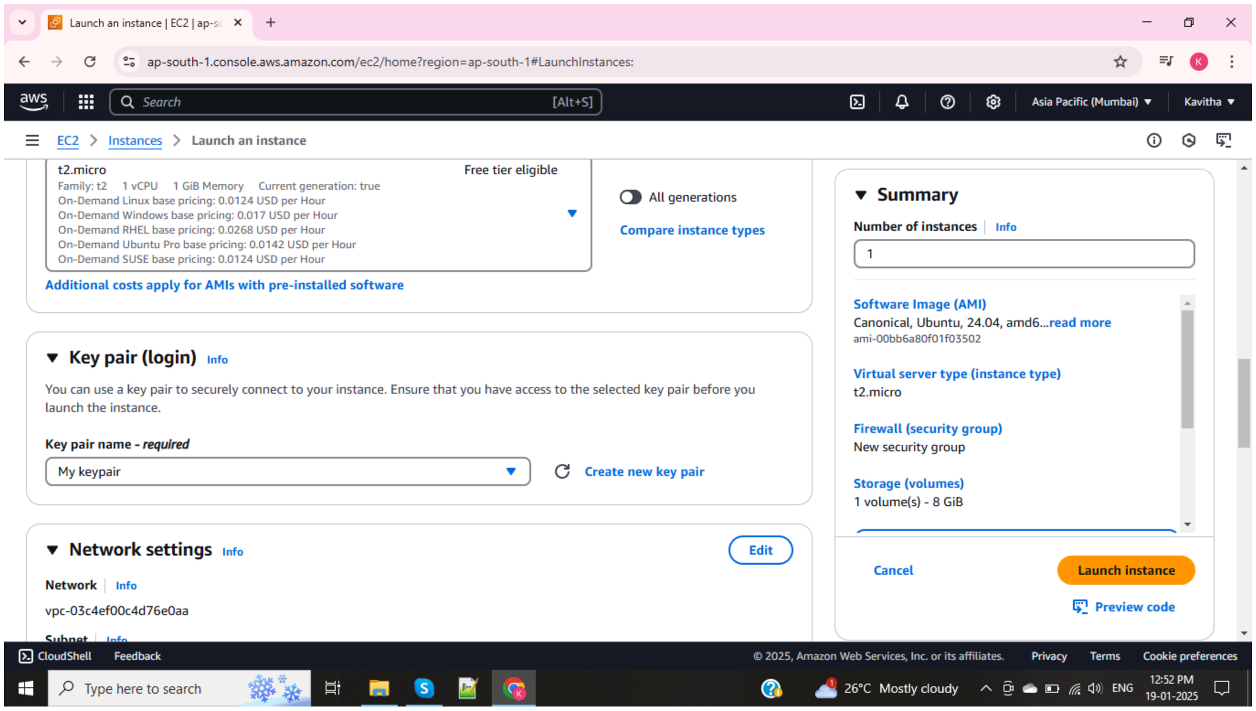
Storage (volumes)
1 volume(s) - 8 GiB

Cancel Launch instance Preview code

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search 26°C Mostly cloudy ENG 12:52 PM 19-01-2025

5. **Choose Instance Type**:Select an instance type based on your performance and cost needs (e.g., `t2.micro` for free tier).


Launch an instance | EC2 | ap-sc

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:

Search [Alt+S] Asia Pacific (Mumbai) Kavitha

EC2 > Instances > Launch an instance

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0124 USD per Hour
On-Demand Windows base pricing: 0.017 USD per Hour
On-Demand RHEL base pricing: 0.0268 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour
On-Demand SUSE base pricing: 0.0124 USD per Hour

All generations
Compare instance types

Additional costs apply for AMIs with pre-installed software

Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*
My keypair
Create new key pair

Network settings Info Edit
Network Info
vpc-03c4ef00c4d76e0aa
Subnet Info

Summary
Number of instances Info
1

Software Image (AMI)
Canonical, Ubuntu, 24.04, amd6...read more
ami-00bb6a80f01f03502

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

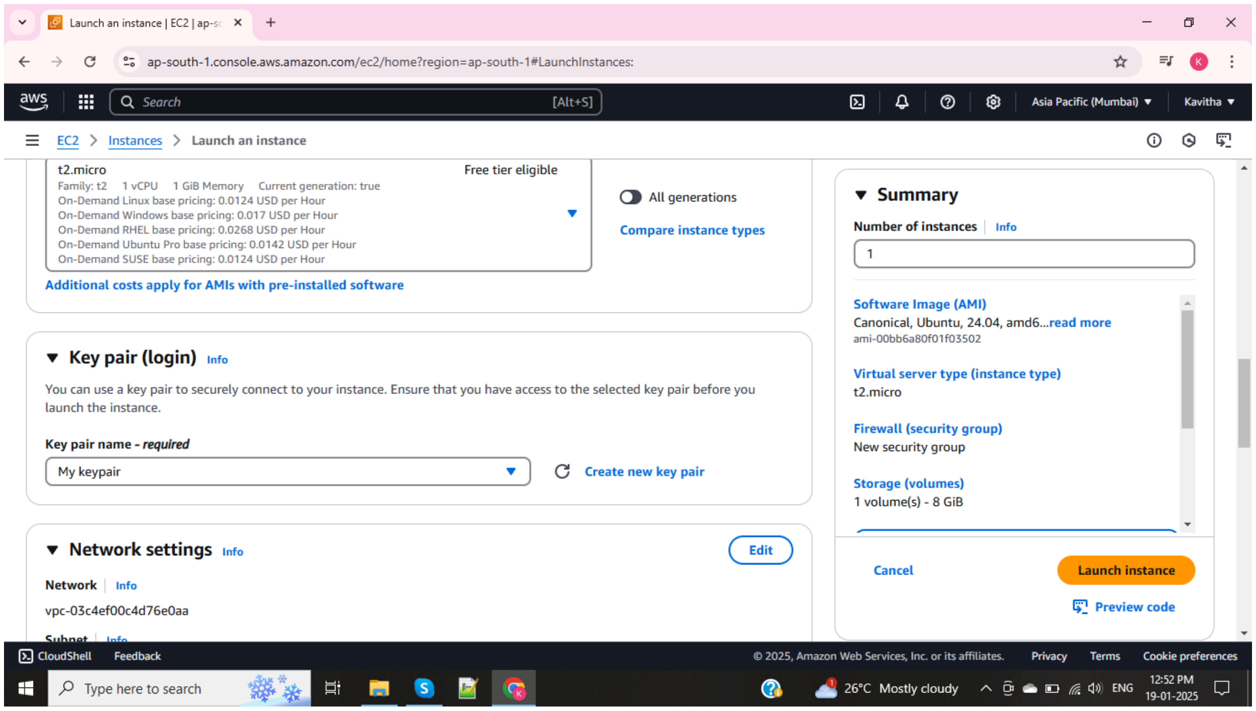
Storage (volumes)
1 volume(s) - 8 GiB

Cancel Launch instance Preview code

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Type here to search 26°C Mostly cloudy ENG 12:52 PM 19-01-2025

6. **Add keypair**: Create and select the keypair to help login into the EC2 instance.
7. **Review and Launch**:Review all settings and click **Launch**.



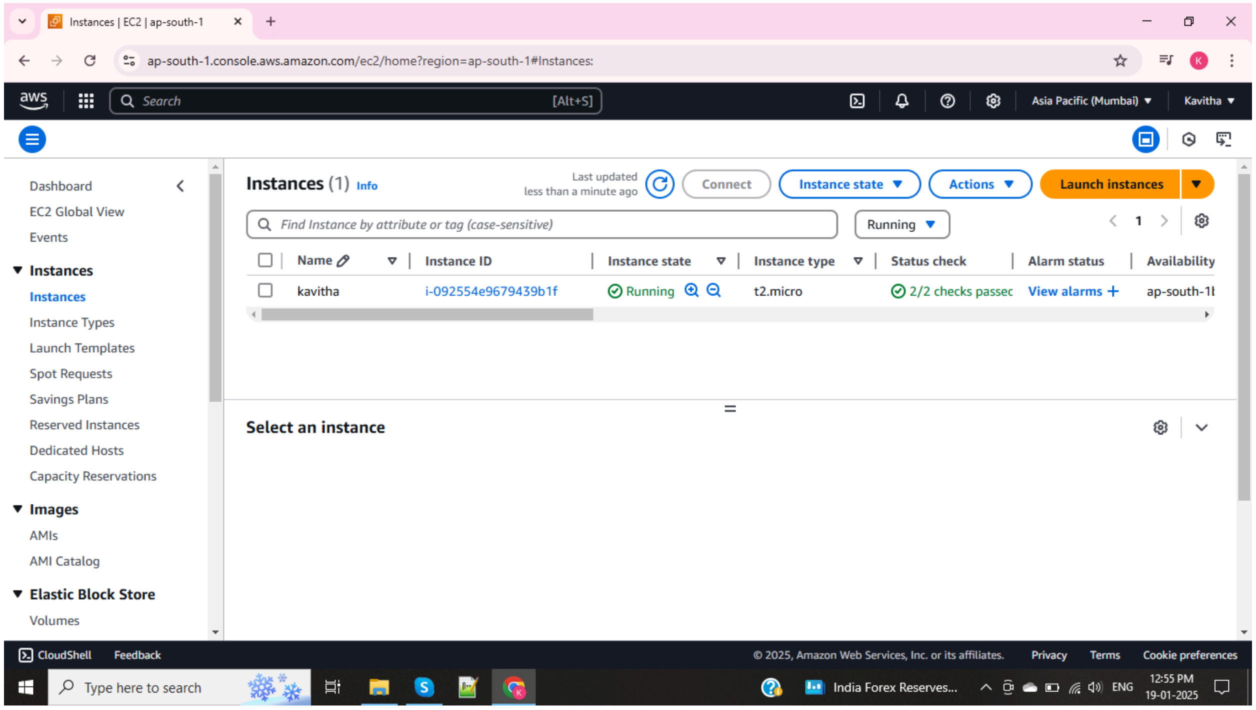
t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0124 USD per Hour
On-Demand Windows base pricing: 0.017 USD per Hour
On-Demand RHEL base pricing: 0.0268 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0142 USD per Hour
On-Demand SUSE base pricing: 0.0124 USD per Hour
Additional costs apply for AMIs with pre-installed software

Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - *required*: My keypair
Create new key pair

Network settings Info
Network: vpc-03c4ef00c4d76e0aa

Summary
Number of instances: 1
Software Image (AMI): Canonical, Ubuntu, 24.04, amd6...read more
ami-00bb6a80f01f03502
Virtual server type (instance type): t2.micro
Firewall (security group): New security group
Storage (volumes): 1 volume(s) - 8 GiB
Cancel | Launch instance | Preview code

EC2 Instance created and Output:

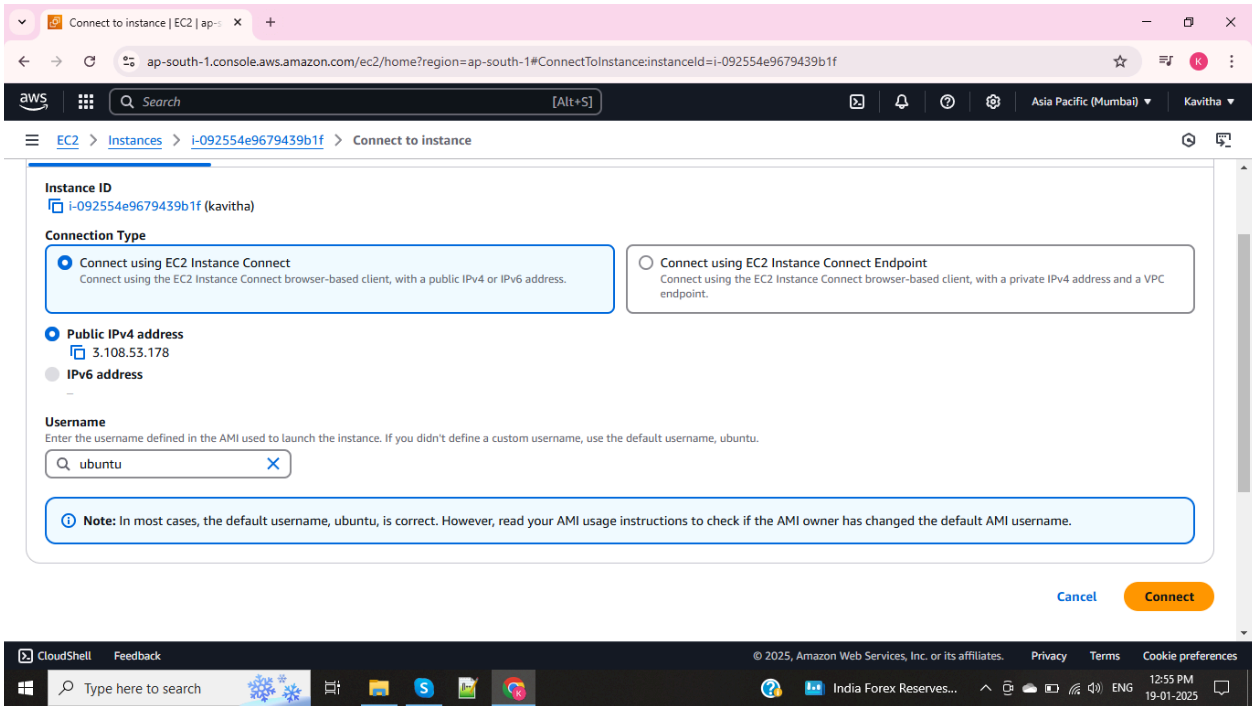


Instances (1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
kavitha	i-092554e9679439b1f	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1l

Select an instance

8. **Connect to the Instance**:Use the key pair to connect via SSH or use AWS Systems Manager Session Manager.


Instance ID
i-092554e9679439b1f (kavitha)

Connection Type

(●) Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

( ) Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

(●) Public IPv4 address
3.108.53.178

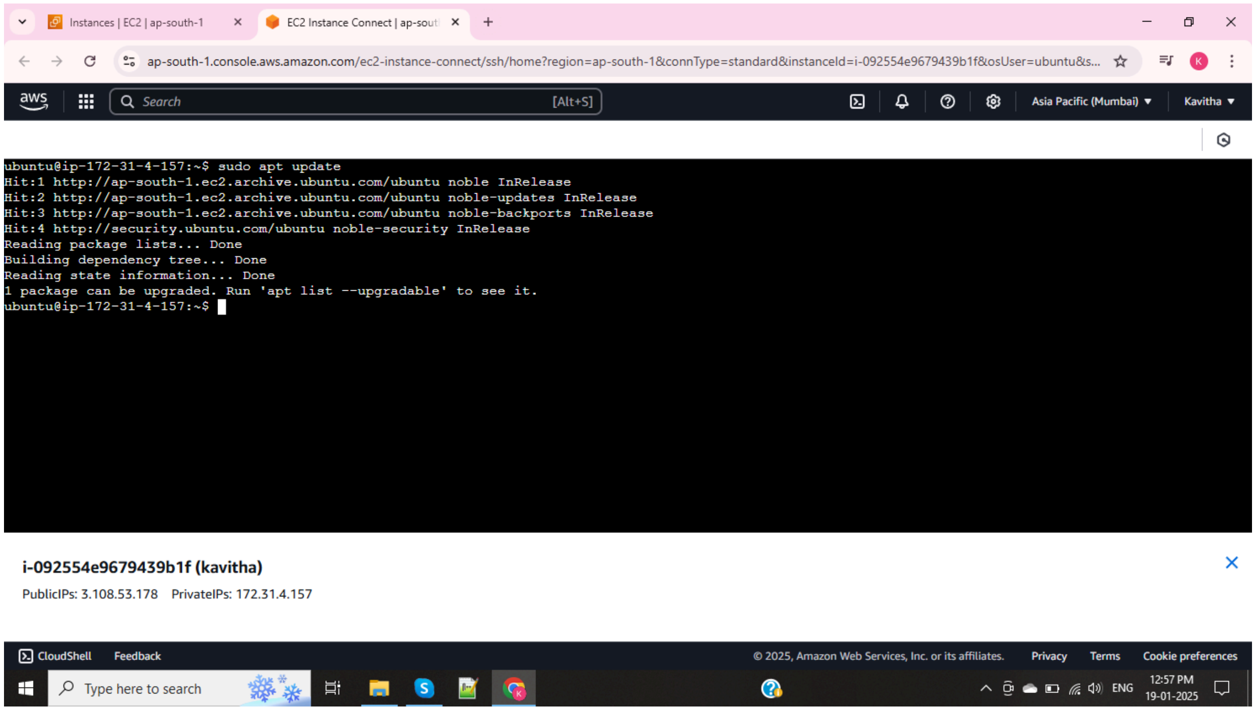
( ) IPv6 address

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

ubuntu

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

After connecting the server first we have to update the server : **sudo apt update**



```
ubuntu@ip-172-31-4-157:~$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
ubuntu@ip-172-31-4-157:~$
```

i-092554e9679439b1f (kavitha)
PublicIPs: 3.108.53.178 PrivateIPs: 172.31.4.157