

# AWS

**Amazon Web Services (AWS)** is a comprehensive cloud computing platform provided by Amazon. It offers a wide range of on-demand computing resources such as servers, storage, databases, networking, machine learning, analytics, and more. AWS enables businesses and individuals to build, deploy, and scale applications and infrastructure without the need for physical hardware.

## Key Features of AWS

1. **Scalability**
  - Automatically scale resources up or down based on demand.
2. **Cost-Effectiveness**
  - Pay-as-you-go pricing model with no upfront costs.
3. **Global Infrastructure**
  - Operates in multiple regions and availability zones for high availability and low latency.
4. **Wide Range of Services**
  - Offers over 200 services, including compute (EC2), storage (S3), databases (RDS, DynamoDB), machine learning (SageMaker), and more.
5. **Security**
  - Provides advanced security features like encryption, compliance certifications, and access controls.
6. **Developer-Friendly**
  - Supports a variety of tools, SDKs, and APIs for integration and automation.

## Why Use AWS?

1. **Flexibility:** Supports multiple operating systems, programming languages, and platforms.
2. **Reliability:** Built for fault tolerance and disaster recovery.
3. **Innovation:** Provides cutting-edge technologies like AI, IoT, and data analytics.
4. **Ecosystem:** Large community support, extensive documentation, and third-party integrations.

## Use Cases of AWS

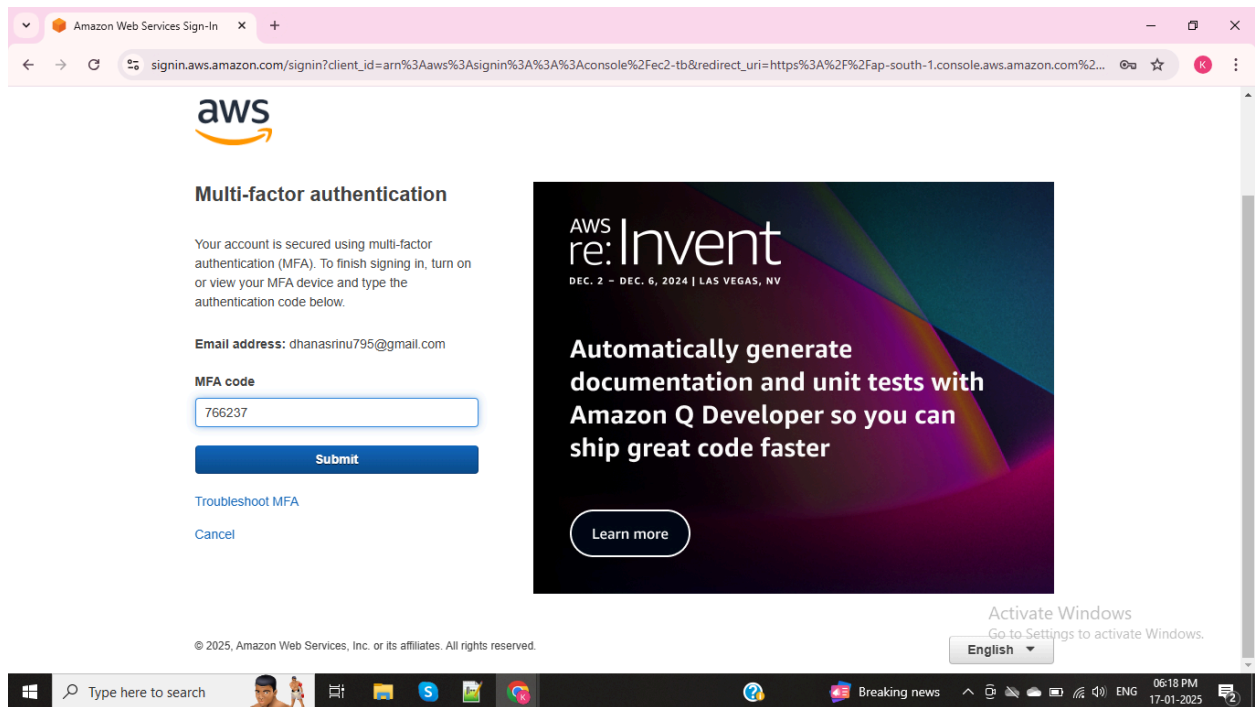
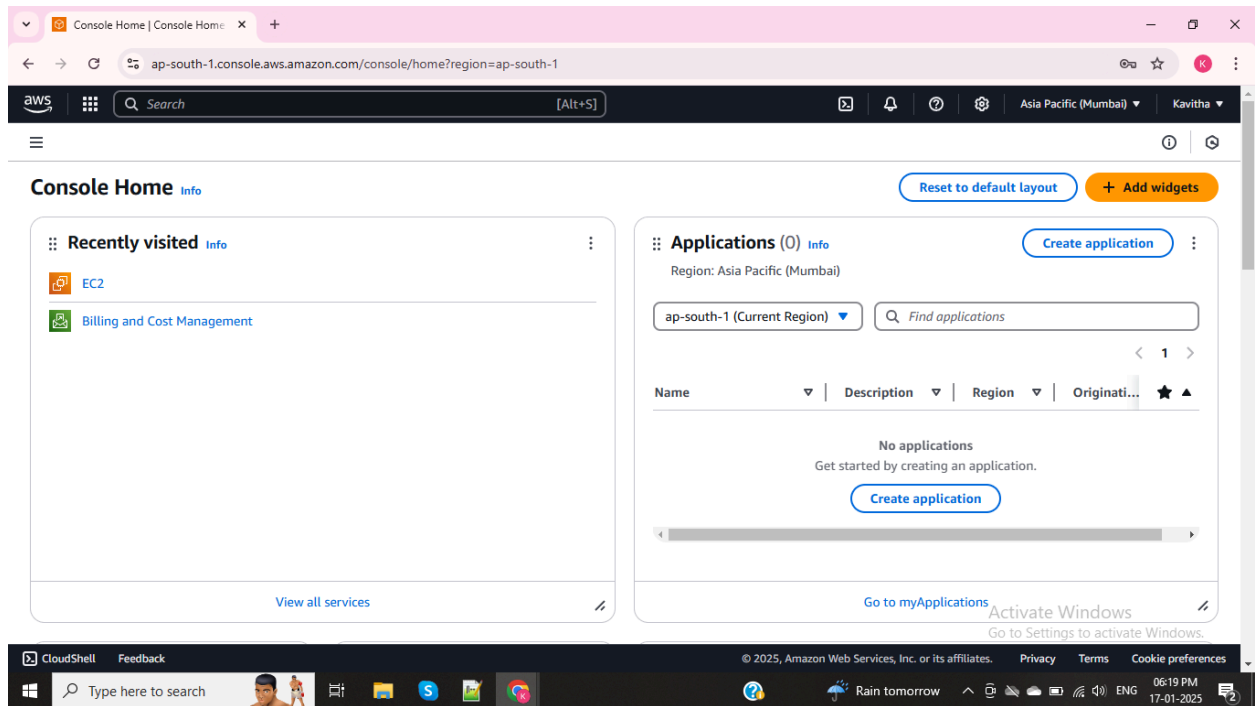
- 1. **Web Hosting:** Host websites and web applications.
- 2. **Data Analytics:** Process and analyze big data efficiently.
- 3. **Application Development:** Build and deploy scalable applications.
- 4. **Disaster Recovery:** Implement cost-effective and resilient disaster recovery solutions.
- 5. **Gaming:** Power online multiplayer games with scalable backend servers.

AWS is widely popular because of its ease of use, flexibility, and reliability, making it the leader in the cloud computing market.

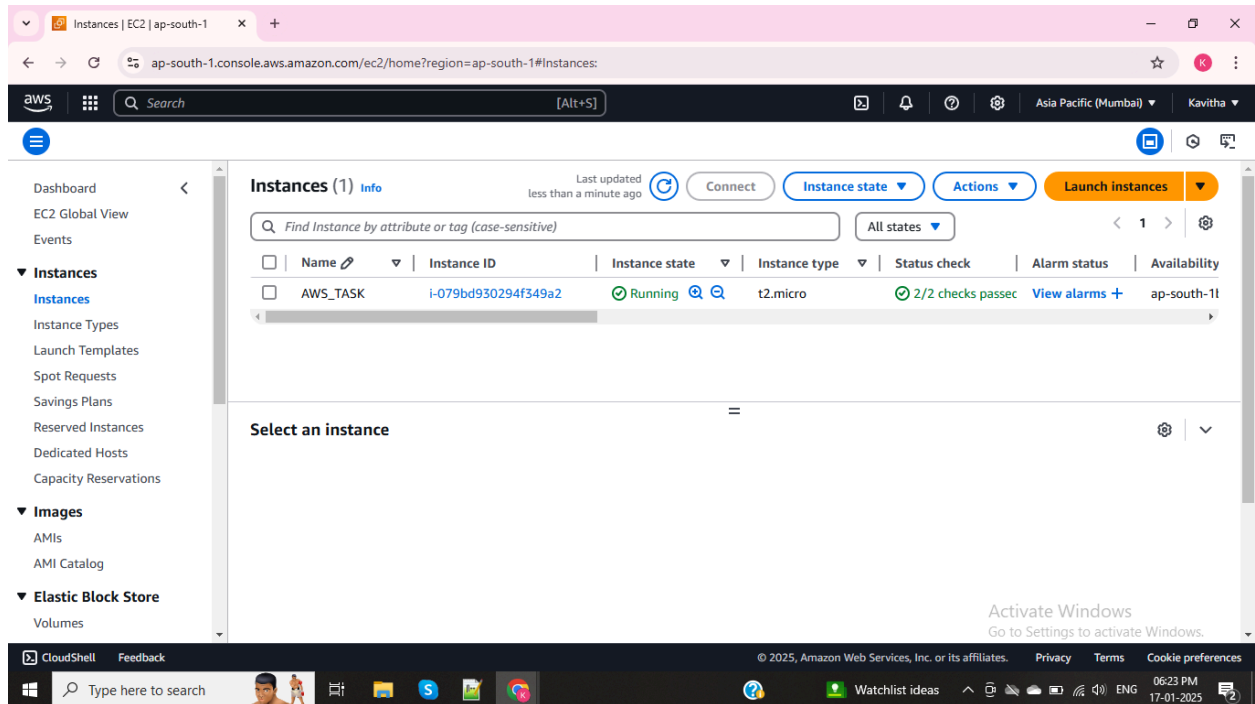
## Popular AWS Services

Category	Service Name	Purpose
Compute	EC2, Lambda	Virtual servers, serverless computing
Storage	S3, EBS, Glacier	Object storage, block storage, archiving
Databases	RDS, DynamoDB, Redshift	Relational, NoSQL, data warehousing
Networking	VPC, Route 53, CloudFront	Private networking, DNS, content delivery
AI/ML	SageMaker, Rekognition	Machine learning, image recognition
Monitoring	CloudWatch, CloudTrail	Monitoring and logging services
DevOps	CodePipeline, CodeBuild, CodeDeploy	Continuous integration and delivery

## Task 1 : AWS Account created and set Multi-factor authentication



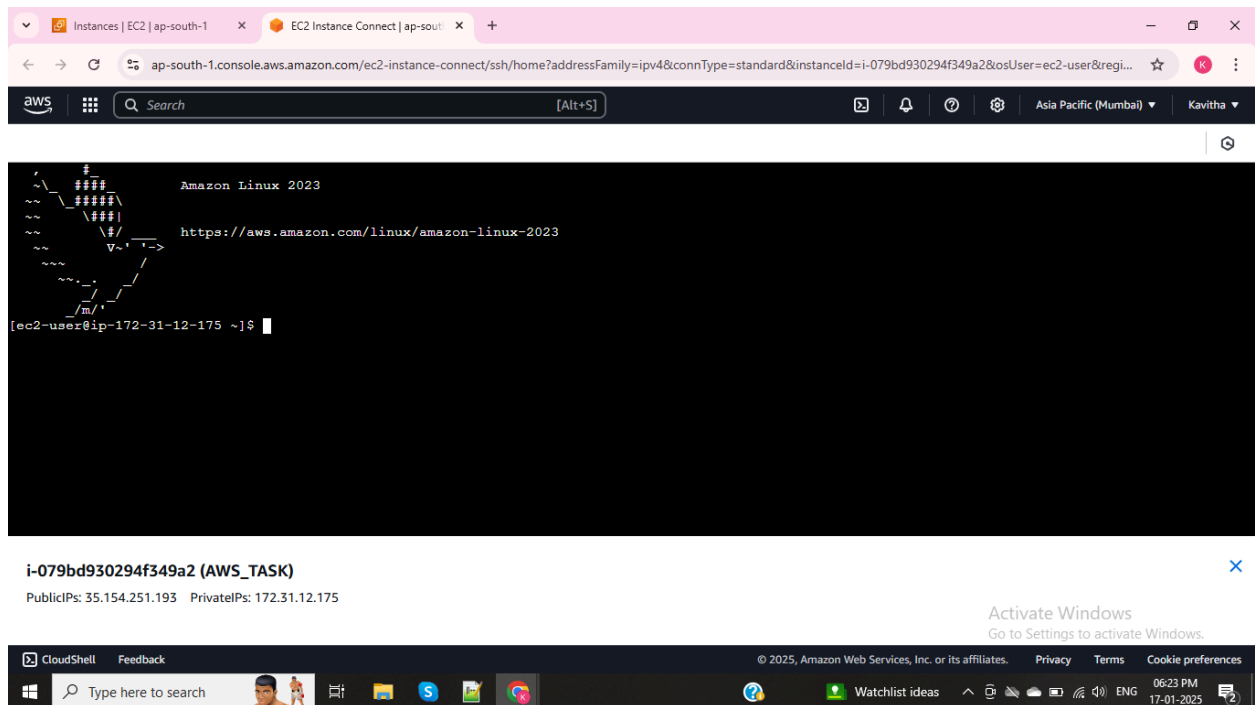
## Task 2 : Creating an EC2 instance and execute shell commands



The screenshot displays the AWS Management Console for the 'ap-south-1' region. The 'Instances' page shows a single instance named 'AWS\_TASK' with ID 'i-079bd930294f349a2'. The instance is in a 'Running' state, using the 't2.micro' instance type. The status checks show '2/2 checks passed'. The console also displays the 'Select an instance' section and the 'Activate Windows' watermark.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
AWS_TASK	i-079bd930294f349a2	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-11

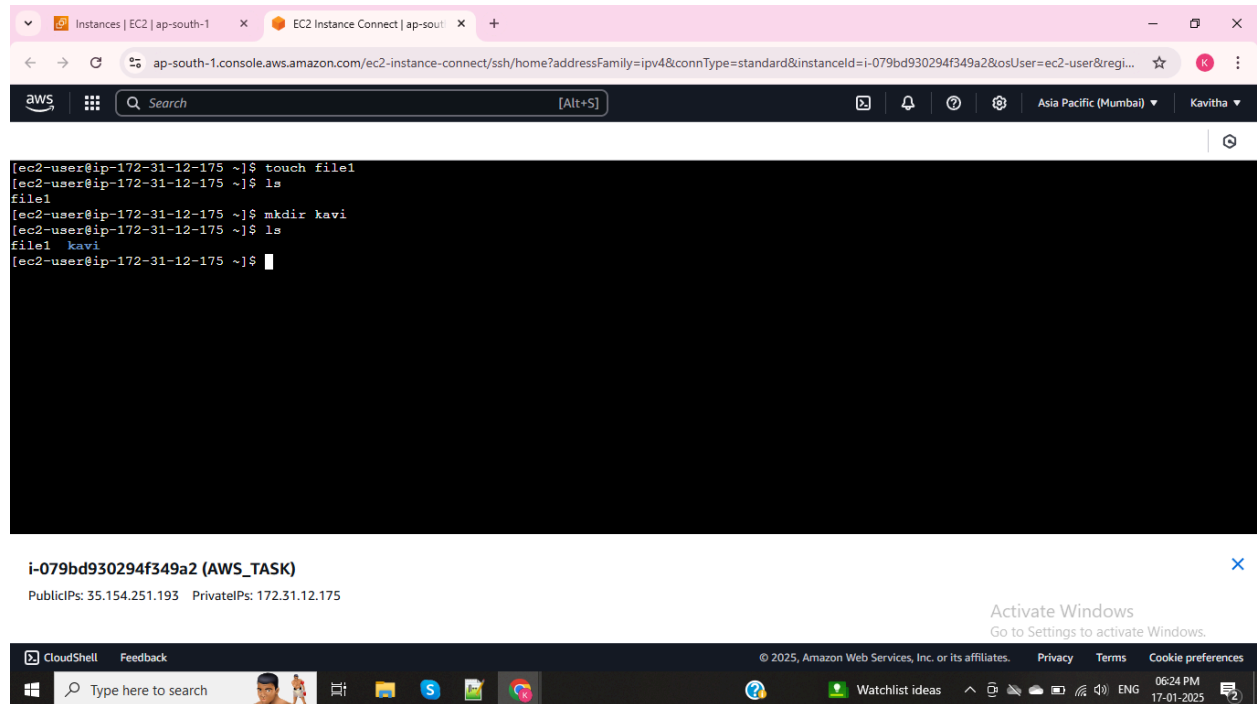
## Launch instance and connect the EC2 instance



The screenshot displays the AWS Management Console for the 'ap-south-1' region, specifically the 'EC2 Instance Connect' page. The console shows the terminal output of the 'ssh' command, indicating a successful connection to the EC2 instance. The terminal output includes the Amazon Linux 2023 logo and the public IP address 35.154.251.193.

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023
[ec2-user@ip-172-31-12-175 ~]$
```

To create file = touch filename  
To create folder = mkdir (folder name)

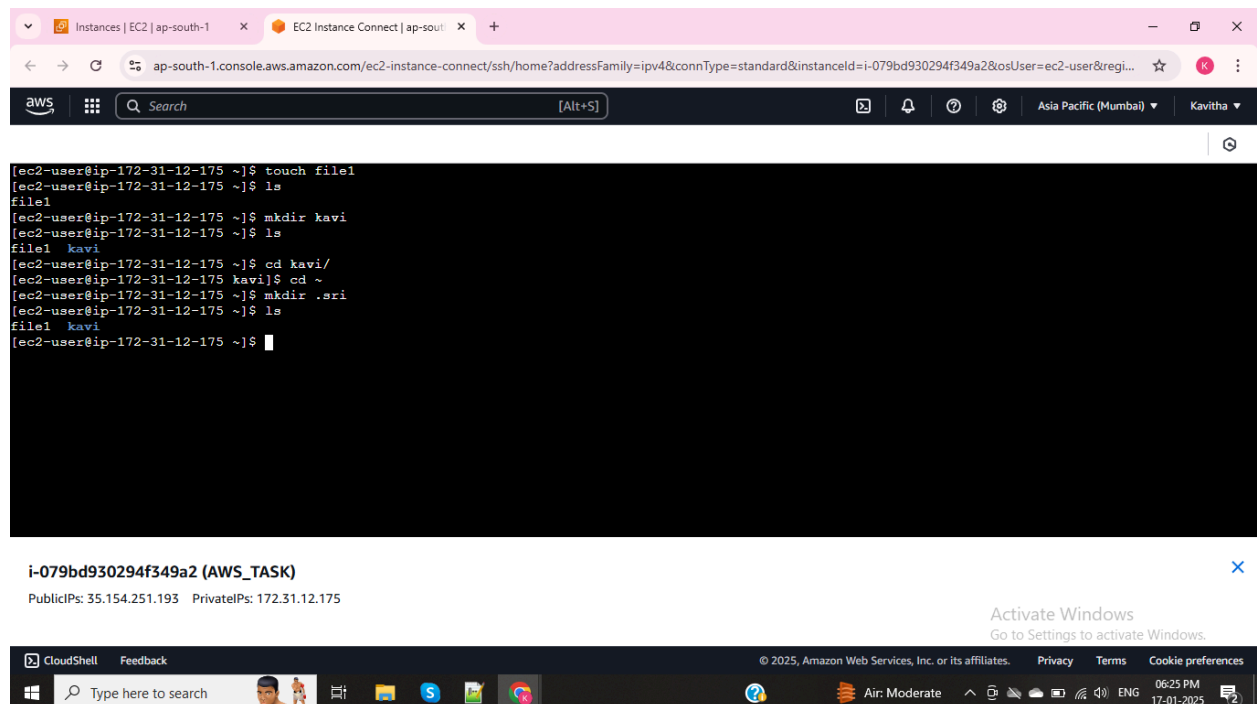


The screenshot shows the AWS Management Console interface. At the top, there are tabs for 'Instances | EC2 | ap-south-1' and 'EC2 Instance Connect | ap-south-1'. The browser address bar shows the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-079bd930294f349a2&osUser=ec2-user&regi...`. Below the console header, there is a search bar and a navigation menu. The main content area displays a terminal window for the EC2 instance `i-079bd930294f349a2 (AWS_TASK)`. The terminal output shows the following commands and their results:

```
[ec2-user@ip-172-31-12-175 ~]$ touch file1
[ec2-user@ip-172-31-12-175 ~]$ ls
file1
[ec2-user@ip-172-31-12-175 ~]$ mkdir kavi
[ec2-user@ip-172-31-12-175 ~]$ ls
file1  kavi
[ec2-user@ip-172-31-12-175 ~]$
```

Below the terminal window, the instance details for `i-079bd930294f349a2 (AWS_TASK)` are shown, including PublicIPs: 35.154.251.193 and PrivateIPs: 172.31.12.175. An 'Activate Windows' watermark is visible in the bottom right corner of the console.

To login into folder = cd (folder name)  
To come out of the folder = cd ~



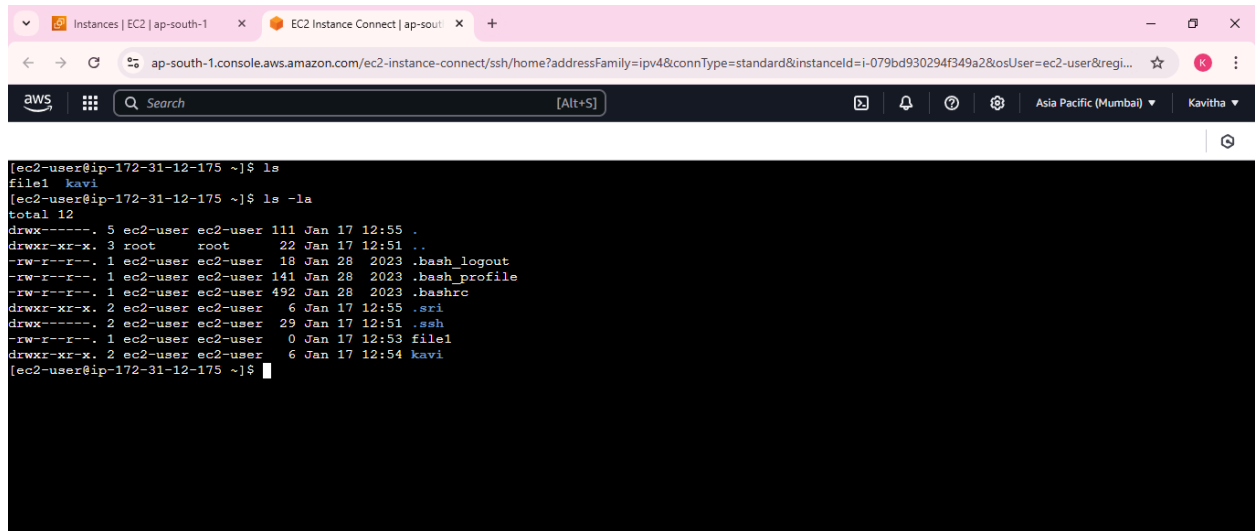
The screenshot shows the AWS Management Console interface, similar to the first one. The browser address bar shows the URL: `ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-079bd930294f349a2&osUser=ec2-user&regi...`. The main content area displays a terminal window for the EC2 instance `i-079bd930294f349a2 (AWS_TASK)`. The terminal output shows the following commands and their results:

```
[ec2-user@ip-172-31-12-175 ~]$ touch file1
[ec2-user@ip-172-31-12-175 ~]$ ls
file1
[ec2-user@ip-172-31-12-175 ~]$ mkdir kavi
[ec2-user@ip-172-31-12-175 ~]$ ls
file1  kavi
[ec2-user@ip-172-31-12-175 ~]$ cd kavi/
[ec2-user@ip-172-31-12-175 kavi]$ cd ~
[ec2-user@ip-172-31-12-175 ~]$ mkdir .sri
[ec2-user@ip-172-31-12-175 ~]$ ls
file1  kavi
[ec2-user@ip-172-31-12-175 ~]$
```

Below the terminal window, the instance details for `i-079bd930294f349a2 (AWS_TASK)` are shown, including PublicIPs: 35.154.251.193 and PrivateIPs: 172.31.12.175. An 'Activate Windows' watermark is visible in the bottom right corner of the console.

To create hidden folder = `mkdir .kavi`

To check the all & hidden files and folders = `ls -la`



The screenshot shows the AWS CloudShell interface with a terminal window. The terminal output is as follows:

```
[ec2-user@ip-172-31-12-175 ~]$ ls
file1 kavi
[ec2-user@ip-172-31-12-175 ~]$ ls -la
total 12
drwx----- 5 ec2-user ec2-user 111 Jan 17 12:55 .
drwxr-xr-x 3 root root 22 Jan 17 12:51 ..
-rw-r--r-- 1 ec2-user ec2-user 18 Jan 28 2023 .bash_logout
-rw-r--r-- 1 ec2-user ec2-user 141 Jan 28 2023 .bash_profile
-rw-r--r-- 1 ec2-user ec2-user 492 Jan 28 2023 .bashrc
drwxr-xr-x 2 ec2-user ec2-user 6 Jan 17 12:55 .sri
drwx----- 2 ec2-user ec2-user 29 Jan 17 12:51 .ssh
-rw-r--r-- 1 ec2-user ec2-user 0 Jan 17 12:53 file1
drwxr-xr-x 2 ec2-user ec2-user 6 Jan 17 12:54 kavi
[ec2-user@ip-172-31-12-175 ~]$
```

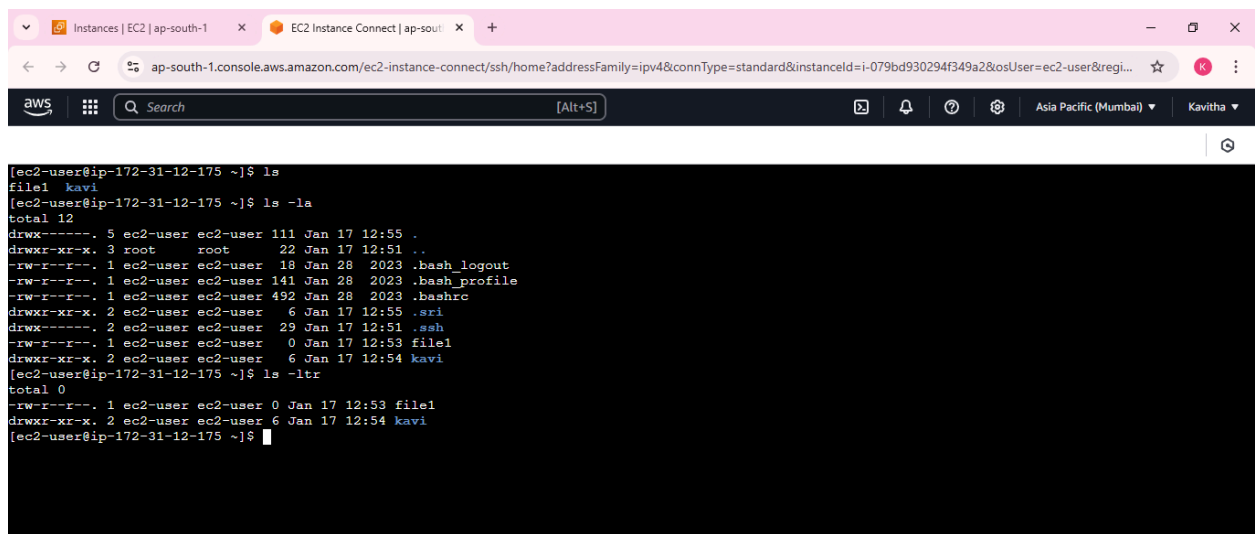
i-079bd930294f349a2 (AWS\_TASK)

PublicIPs: 35.154.251.193 PrivateIPs: 172.31.12.175

Activate Windows  
Go to Settings to activate Windows.



To check created date and time of the file or folder = `ls -ltr`



The screenshot shows the AWS CloudShell interface with a terminal window. The terminal output is as follows:

```
[ec2-user@ip-172-31-12-175 ~]$ ls
file1 kavi
[ec2-user@ip-172-31-12-175 ~]$ ls -la
total 12
drwx----- 5 ec2-user ec2-user 111 Jan 17 12:55 .
drwxr-xr-x 3 root root 22 Jan 17 12:51 ..
-rw-r--r-- 1 ec2-user ec2-user 18 Jan 28 2023 .bash_logout
-rw-r--r-- 1 ec2-user ec2-user 141 Jan 28 2023 .bash_profile
-rw-r--r-- 1 ec2-user ec2-user 492 Jan 28 2023 .bashrc
drwxr-xr-x 2 ec2-user ec2-user 6 Jan 17 12:55 .sri
drwx----- 2 ec2-user ec2-user 29 Jan 17 12:51 .ssh
-rw-r--r-- 1 ec2-user ec2-user 0 Jan 17 12:53 file1
drwxr-xr-x 2 ec2-user ec2-user 6 Jan 17 12:54 kavi
[ec2-user@ip-172-31-12-175 ~]$ ls -ltr
total 0
-rw-r--r-- 1 ec2-user ec2-user 0 Jan 17 12:53 file1
drwxr-xr-x 2 ec2-user ec2-user 6 Jan 17 12:54 kavi
[ec2-user@ip-172-31-12-175 ~]$
```

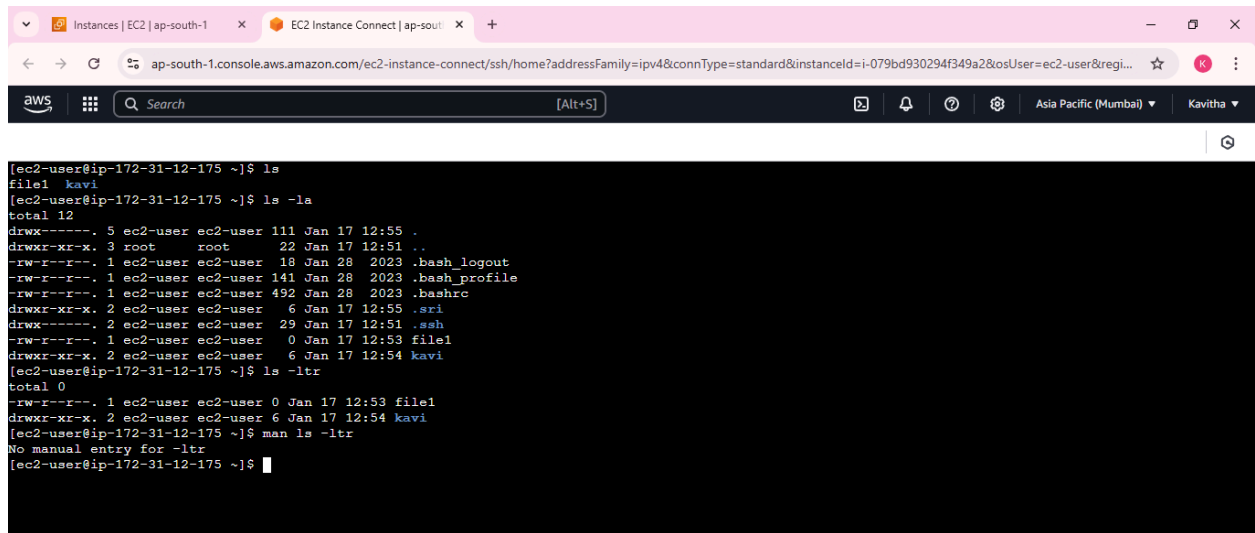
i-079bd930294f349a2 (AWS\_TASK)

PublicIPs: 35.154.251.193 PrivateIPs: 172.31.12.175

Activate Windows  
Go to Settings to activate Windows.



To check the meaning of unknow command = use man (manual)



```
[ec2-user@ip-172-31-12-175 ~]$ ls
file1 kavi
[ec2-user@ip-172-31-12-175 ~]$ ls -la
total 12
drwx----- 5 ec2-user ec2-user 111 Jan 17 12:55 .
drwxr-xr-x. 3 root      root    22 Jan 17 12:51 ..
-rw-r--r--. 1 ec2-user ec2-user 18 Jan 28 2023 .bash_logout
-rw-r--r--. 1 ec2-user ec2-user 141 Jan 28 2023 .bash_profile
-rw-r--r--. 1 ec2-user ec2-user 492 Jan 28 2023 .bashrc
drwxr-xr-x. 2 ec2-user ec2-user  6 Jan 17 12:55 .sri
drwx----- 2 ec2-user ec2-user 29 Jan 17 12:51 .ssh
-rw-r--r--. 1 ec2-user ec2-user  0 Jan 17 12:53 file1
drwxr-xr-x. 2 ec2-user ec2-user  6 Jan 17 12:54 kavi
[ec2-user@ip-172-31-12-175 ~]$ ls -ltr
total 0
-rw-r--r--. 1 ec2-user ec2-user 0 Jan 17 12:53 file1
drwxr-xr-x. 2 ec2-user ec2-user 6 Jan 17 12:54 kavi
[ec2-user@ip-172-31-12-175 ~]$ man ls -ltr
No manual entry for -ltr
[ec2-user@ip-172-31-12-175 ~]$
```

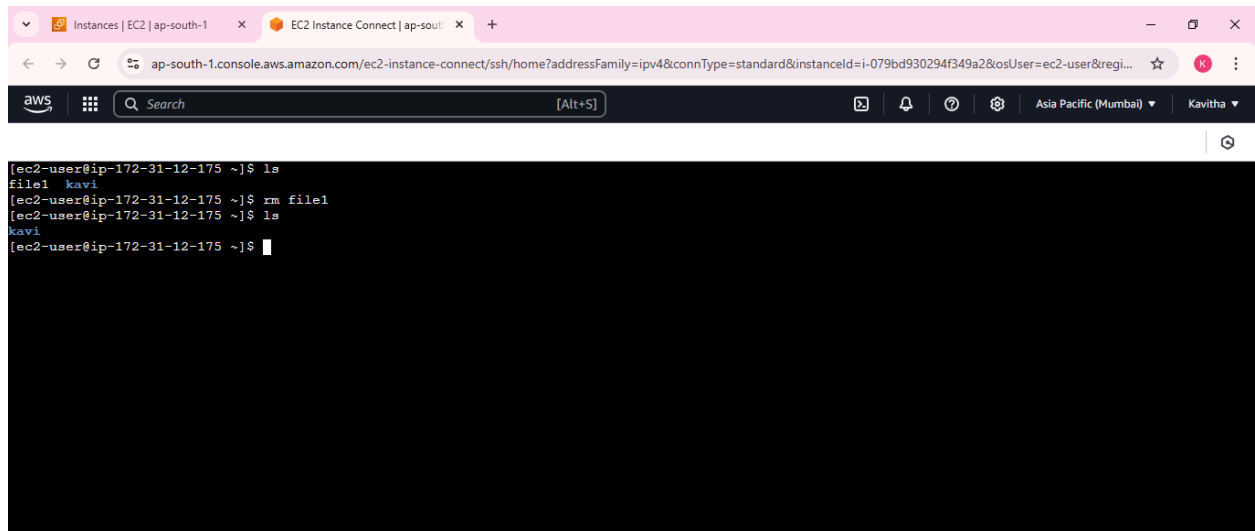
i-079bd930294f349a2 (AWS\_TASK)

PublicIPs: 35.154.251.193 PrivateIPs: 172.31.12.175

Activate Windows  
Go to Settings to activate Windows.



To remove the file = rm (file name)  
To remove overall files = rm \*(file name)



```
[ec2-user@ip-172-31-12-175 ~]$ ls
file1 kavi
[ec2-user@ip-172-31-12-175 ~]$ rm file1
[ec2-user@ip-172-31-12-175 ~]$ ls
kavi
[ec2-user@ip-172-31-12-175 ~]$
```

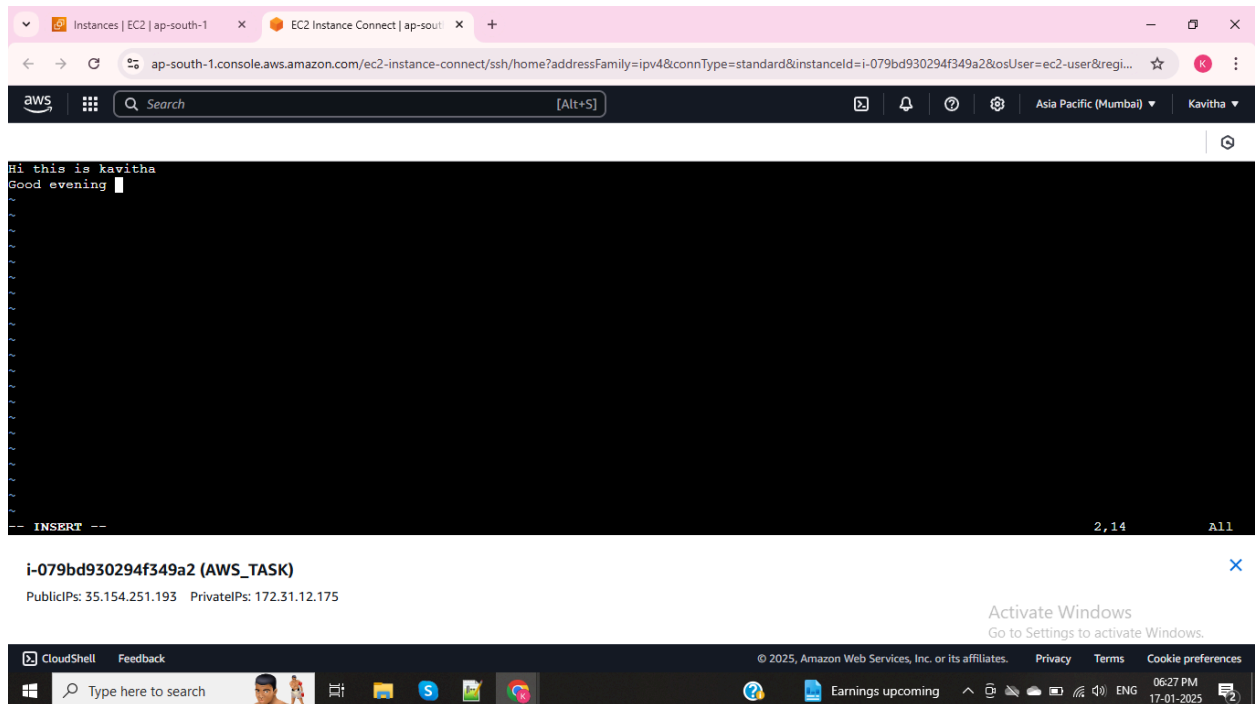
i-079bd930294f349a2 (AWS\_TASK)

PublicIPs: 35.154.251.193 PrivateIPs: 172.31.12.175

Activate Windows  
Go to Settings to activate Windows.



To edit the file or read the file = vi (file name)



The screenshot shows the AWS CloudShell interface. The terminal window displays the following text:

```
Hi this is kavitha
Good evening
```

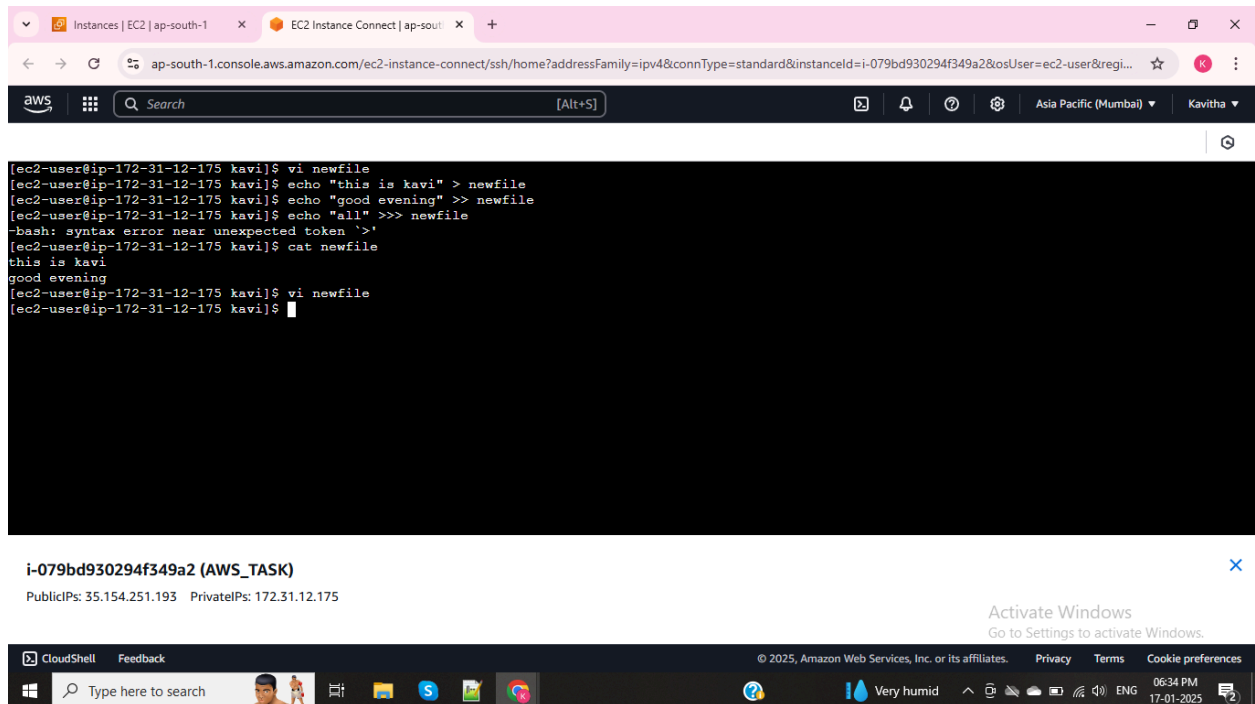
The terminal is in the `INSERT` mode of the `vi` editor. The status bar at the bottom of the terminal shows `2,14` and `All`.

Below the terminal, the instance details for `i-079bd930294f349a2 (AWS_TASK)` are shown, including Public IPs: 35.154.251.193 and Private IPs: 172.31.12.175.

The bottom of the screenshot shows the Windows taskbar with the search bar, taskbar icons, and system tray.

To view the text without opening the file means use **Cat (file name)**

Without open the file we have to add the text in outside = echo "kavitha" >newfile



The screenshot shows the AWS CloudShell interface. The terminal window displays the following text:

```
[ec2-user@ip-172-31-12-175 kavi]$ vi newfile
[ec2-user@ip-172-31-12-175 kavi]$ echo "this is kavi" > newfile
[ec2-user@ip-172-31-12-175 kavi]$ echo "good evening" >> newfile
[ec2-user@ip-172-31-12-175 kavi]$ echo "all" >>> newfile
-bash: syntax error near unexpected token `>'
[ec2-user@ip-172-31-12-175 kavi]$ cat newfile
this is kavi
good evening
[ec2-user@ip-172-31-12-175 kavi]$ vi newfile
[ec2-user@ip-172-31-12-175 kavi]$
```

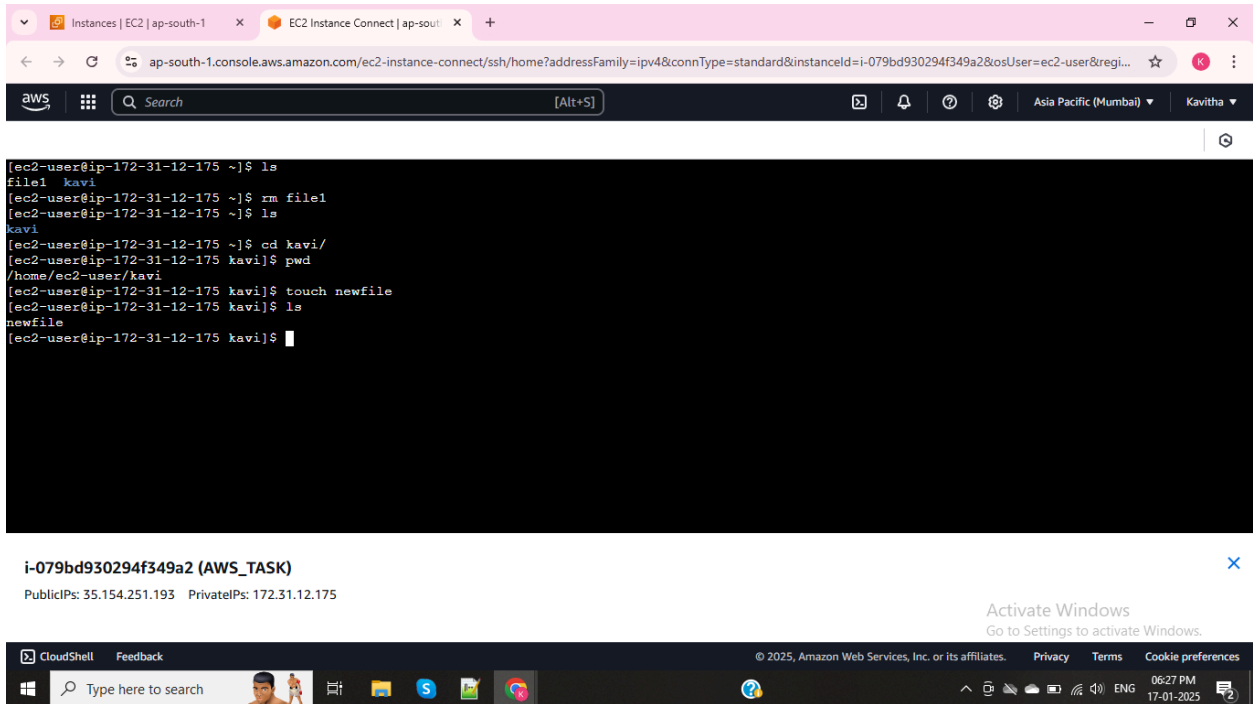
The terminal shows the user creating a file `newfile` using `vi`, then adding text to it using `echo`. Finally, the user uses `cat` to view the contents of the file.

Below the terminal, the instance details for `i-079bd930294f349a2 (AWS_TASK)` are shown, including Public IPs: 35.154.251.193 and Private IPs: 172.31.12.175.

The bottom of the screenshot shows the Windows taskbar with the search bar, taskbar icons, and system tray.



To check the location = pwd (present working directory)

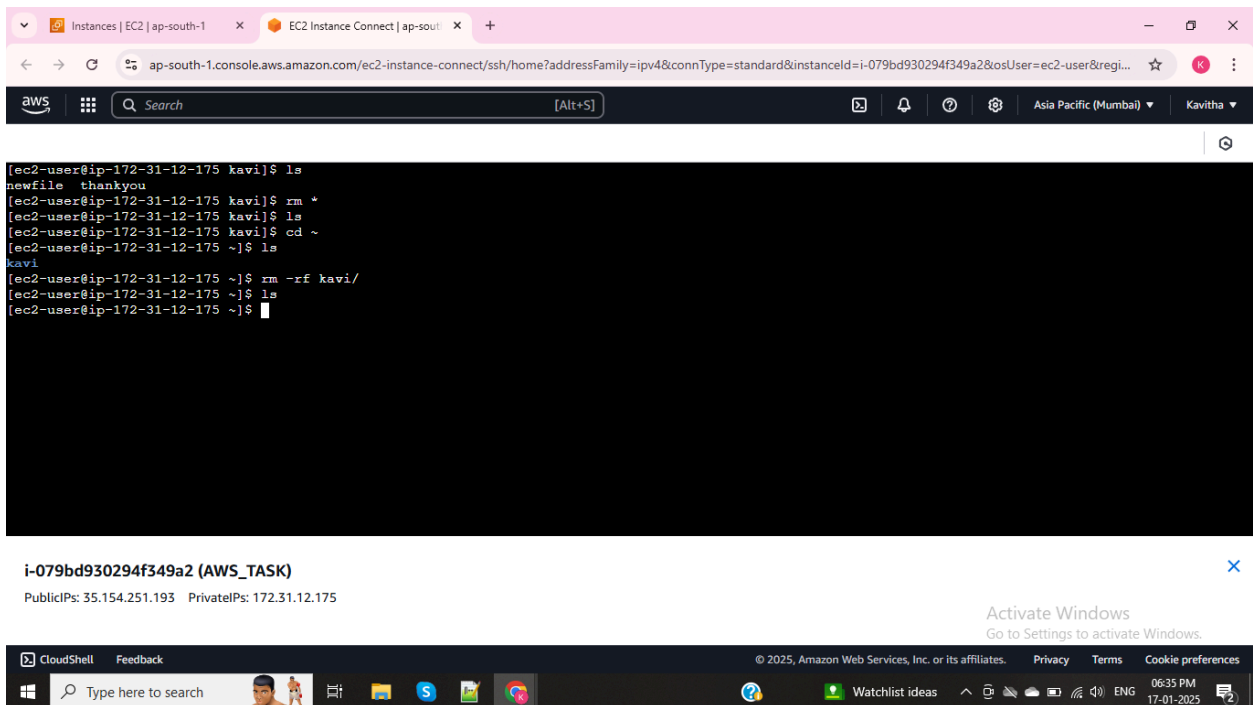


The screenshot shows the AWS CloudShell interface. The terminal window displays the following commands and output:

```
[ec2-user@ip-172-31-12-175 ~]$ ls
file1 kavi
[ec2-user@ip-172-31-12-175 ~]$ rm file1
[ec2-user@ip-172-31-12-175 ~]$ ls
kavi
[ec2-user@ip-172-31-12-175 ~]$ cd kavi/
[ec2-user@ip-172-31-12-175 kavi]$ pwd
/home/ec2-user/kavi
[ec2-user@ip-172-31-12-175 kavi]$ touch newfile
[ec2-user@ip-172-31-12-175 kavi]$ ls
newfile
[ec2-user@ip-172-31-12-175 kavi]$
```

Below the terminal, the instance details for **i-079bd930294f349a2 (AWS\_TASK)** are shown, including PublicIPs: 35.154.251.193 and PrivateIPs: 172.31.12.175. An "Activate Windows" watermark is visible in the bottom right corner.

To remove folder = rm -rf(folder name)



The screenshot shows the AWS CloudShell interface. The terminal window displays the following commands and output:

```
[ec2-user@ip-172-31-12-175 kavi]$ ls
newfile thankyou
[ec2-user@ip-172-31-12-175 kavi]$ rm *
[ec2-user@ip-172-31-12-175 kavi]$ ls
[ec2-user@ip-172-31-12-175 kavi]$ cd ~
[ec2-user@ip-172-31-12-175 ~]$ ls
kavi
[ec2-user@ip-172-31-12-175 ~]$ rm -rf kavi/
[ec2-user@ip-172-31-12-175 ~]$ ls
[ec2-user@ip-172-31-12-175 ~]$
```

Below the terminal, the instance details for **i-079bd930294f349a2 (AWS\_TASK)** are shown, including PublicIPs: 35.154.251.193 and PrivateIPs: 172.31.12.175. An "Activate Windows" watermark is visible in the bottom right corner.

## **SHELL SCRIPTING**

ls	= To list the files/folders
Mkdir	= Make directory
Pwd	= Present working directory
Cd	= Change directory
Man	= Manual
Rm	= Remove
Cat	= To view the inside text in detail
.file/folder	= hidden format
Sudo	= super user do (run as administrator)
Su	= switch user
Echo	= To print the text
History	= check used commands
Clear	= clear the screen