St. JOSEPH'S
GROUP OF INSTITUTIONS
OMR, CHENNAI - 119

aws

Placement Empowerment Program

Cloud Computing and DevOps Centre

WRITE A SHELL SCRIPT TO MANAGE CLOUD RESOURCES

(Create a script to launch, stop, and terminate cloud VMs using the CLI.)

NAME: NIDHISHA A DHAS

DEPARTMENT: AML

St. JOSEPH'S COLLEGE OF ENGINEERING | St. JOSEPH'S INSTITUTE OF TECHNOLOGY
AUTONOMOUS INSTITUTIONS, AFFILIATED TO ANNA UNIVERSITY

INTRODUCTION:

Managing cloud resources efficiently is critical in today's cloud driven IT landscape. AWS Command Line Interface (CLI) provides a powerful tool for interacting with AWS services programmatically. By leveraging shell scripting, we can automate repetitive tasks like launching, stopping, and terminating virtual machines (VMs). This Proof of Concept (POC) demonstrates the use of AWS CLI integrated with a shell script to simplify VM management, showcasing automation's role in reducing manual effort and increasing productivity.

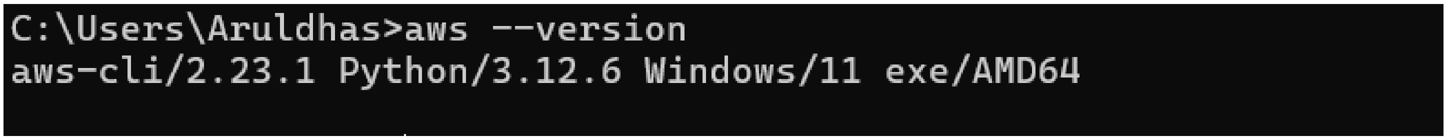
IMPORTANCE:

- Efficiency: Automating cloud resource management reduces time and effort spent on manual tasks.
- Cost Optimization: The ability to stop or terminate unused VMs prevents unnecessary expenses, adhering to best practices in cloud cost management.
- Scalability: Scripting provides a scalable solution for managing multiple resources simultaneously.
- Skill Development: Enhances your technical expertise in AWS CLI, scripting, and cloud automation, which are in high demand in the IT industry.
- Foundation for Advanced Automation: Serves as a stepping stone to more complex automation tasks, such as infrastructure as code (e.g., using tools like Terraform or CloudFormation).

STEP BY STEP OVERVIEW:

STEP 1: INSTALL CLI

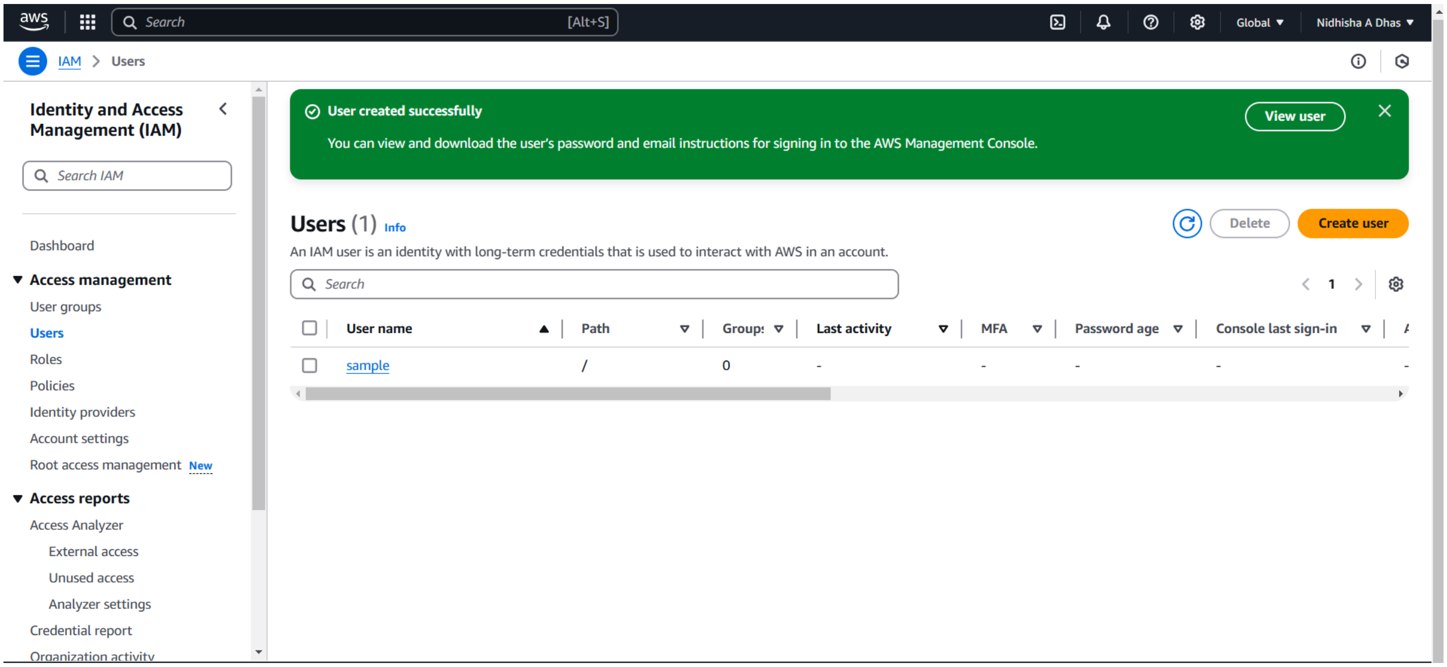
- Make sure your AWS CLI is installed and configured.



```
C:\Users\Aruldhas>aws --version
aws-cli/2.23.1 Python/3.12.6 Windows/11 exe/AMD64
```

STEP 2: CREATE AN IAM USER

- Go to the IAM Dashboard.
- Create IAM user and download the corresponding CSV file.
- Give the name for the user and create the user.



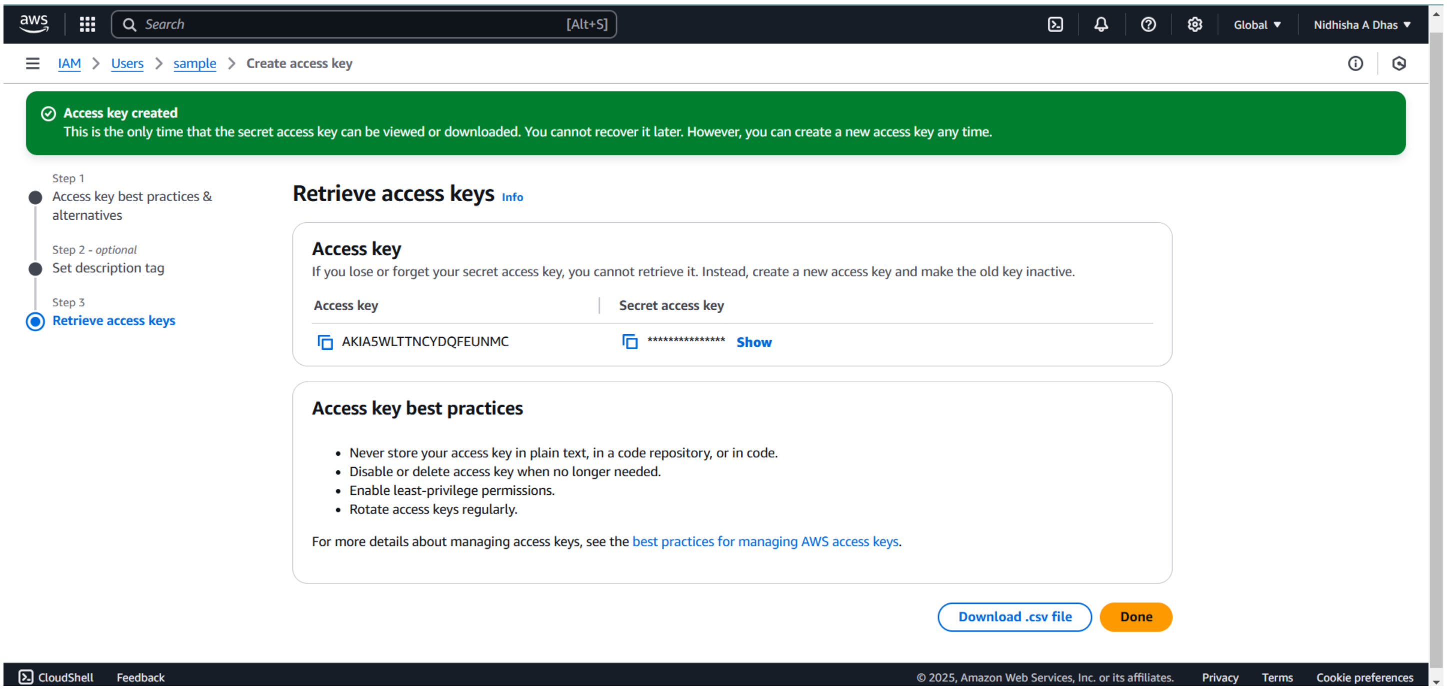
User created successfully

You can view and download the user's password and email instructions for signing in to the AWS Management Console.

Users (1) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

□	User name ▲	Path ▽	Group: ▽	Last activity ▽	MFA ▽	Password age ▽	Console last sign-in ▽	A
□	sample	/	0	-	-	-	-	-



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.

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
AKIA5WLTTNCYDQFEUNMC	*************** Show

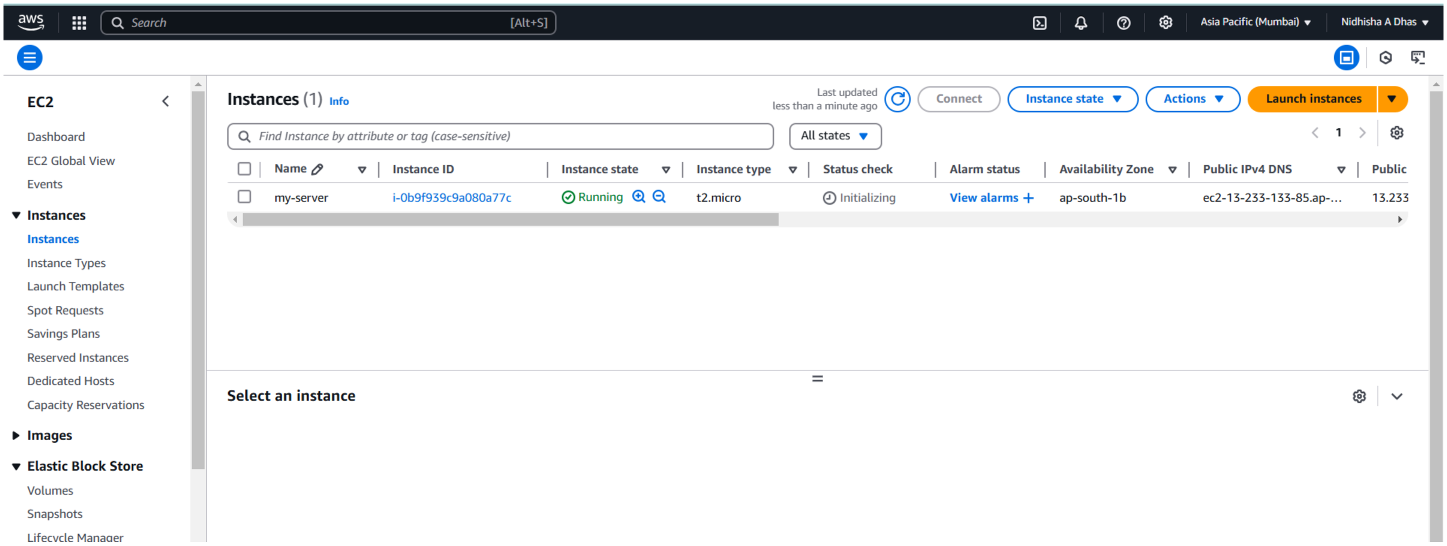
Access key best practices

- 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.
- Rotate access keys regularly.

For more details about managing access keys, see the best practices for managing AWS access keys.

STEP 3: CREATE AN EC2 INSTANCE

- Go to the EC2 Dashboard.
- Click on launch Instance. Specify the instance name, AMI, instance type, Key-Pair value and launch the instance.



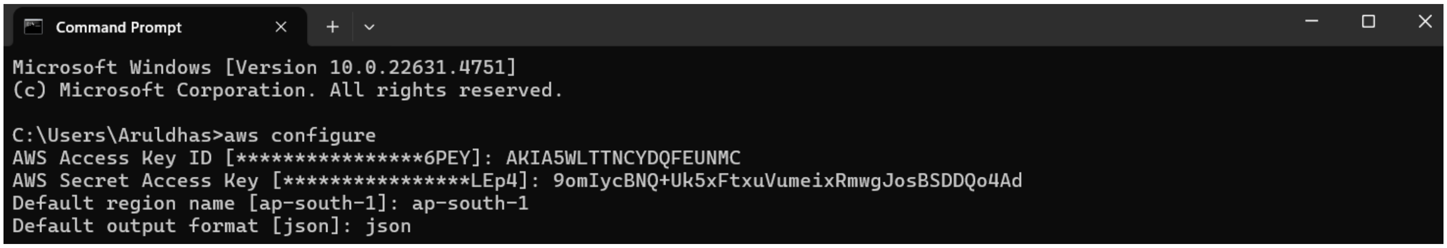
Instances (1)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
my-server	i-0b9f939c9a080a77c	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-13-233-133-85.ap-...	13.233

STEP 4: SHELL SCRIPT

(4.1)

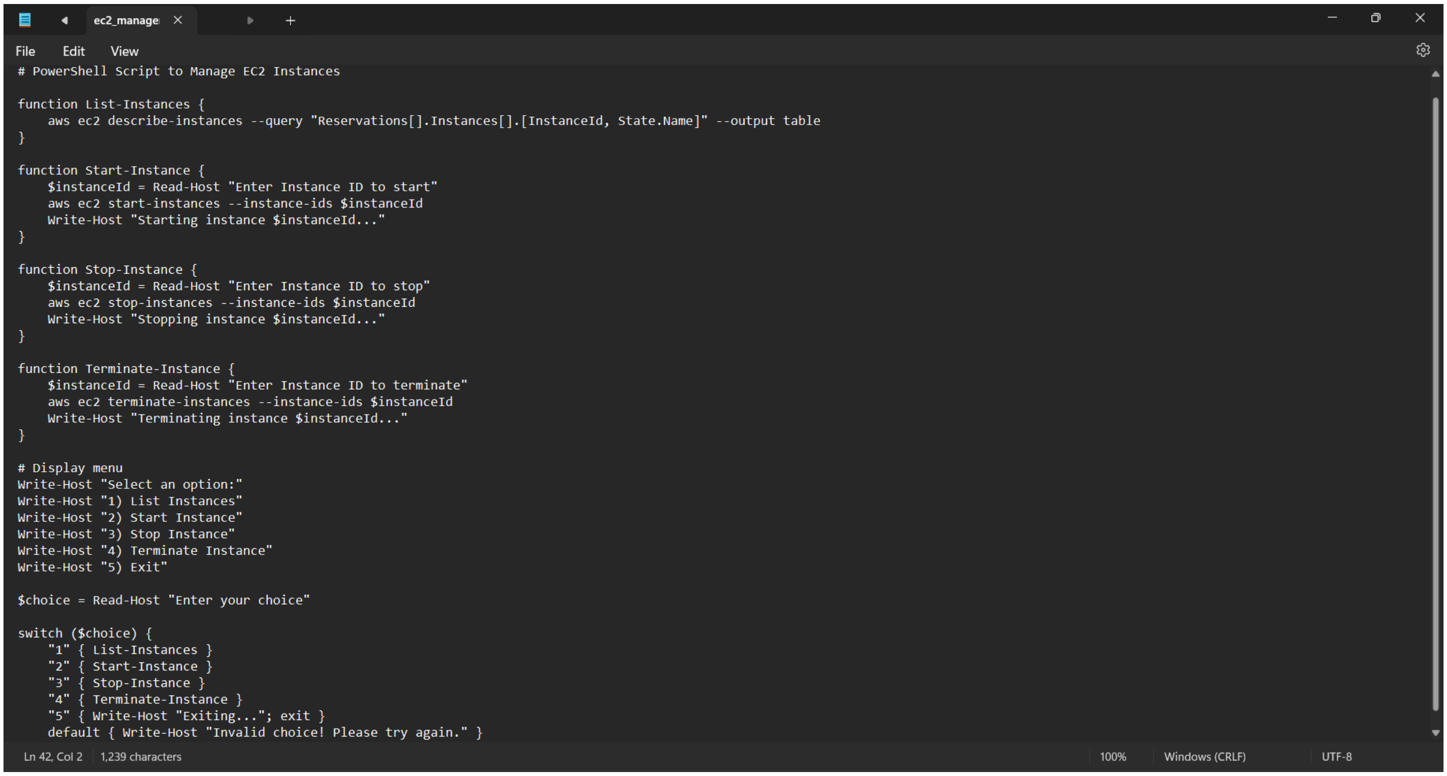
- Open your command prompt, and type ‘aws configure’.
- Give the Access key ID, secret Access key, region name and the output format.



```
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Aruldhas>aws configure
AWS Access Key ID [****************6PEY]: AKIA5WLTTNCYDQFEUNMC
AWS Secret Access Key [****************LEp4]: 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
Default region name [ap-south-1]: ap-south-1
Default output format [json]: json
```

(4.2)

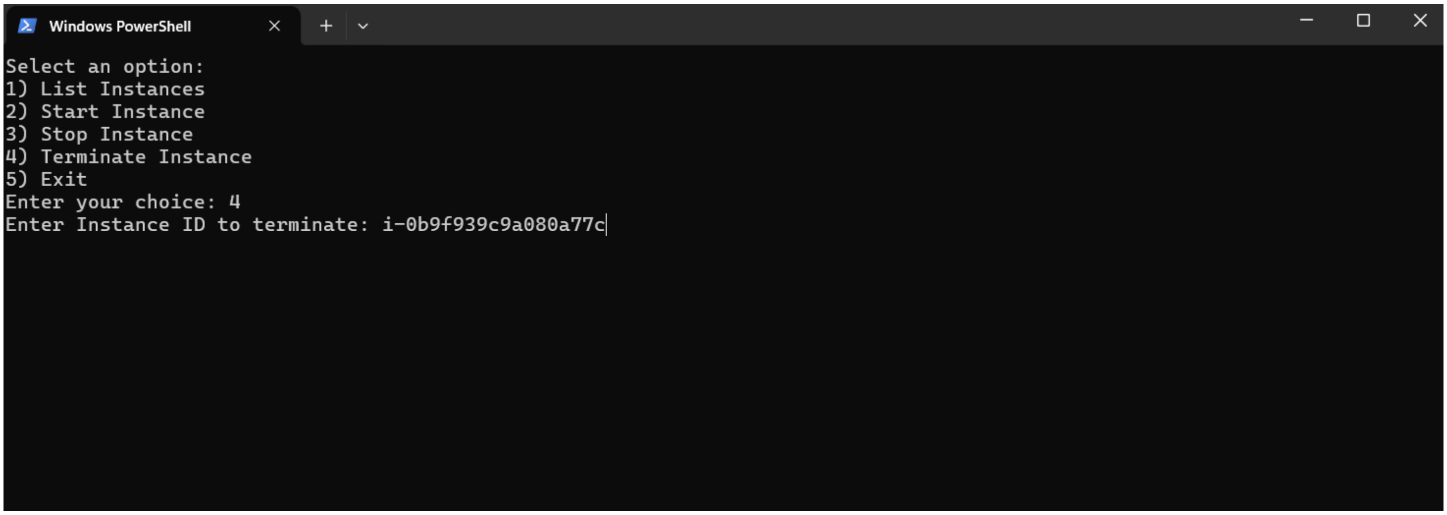
- Open any text editor and write the following script and save the file as ‘ec2-manager.ps1’



```
# PowerShell Script to Manage EC2 Instances

function List-Instances {
    aws ec2 describe-instances --query "Reservations[].Instances[].[InstanceId, State.Name]" --output table
}

function Start-Instance {
    $instanceId = Read-Host "Enter Instance ID to start"
    aws ec2 start-instances --instance-ids $instanceId
    Write-Host "Starting instance $instanceId..."
}

function Stop-Instance {
    $instanceId = Read-Host "Enter Instance ID to stop"
    aws ec2 stop-instances --instance-ids $instanceId
    Write-Host "Stopping instance $instanceId..."
}

function Terminate-Instance {
    $instanceId = Read-Host "Enter Instance ID to terminate"
    aws ec2 terminate-instances --instance-ids $instanceId
    Write-Host "Terminating instance $instanceId..."
}

# Display menu
Write-Host "Select an option:"
Write-Host "1) List Instances"
Write-Host "2) Start Instance"
Write-Host "3) Stop Instance"
Write-Host "4) Terminate Instance"
Write-Host "5) Exit"

$choice = Read-Host "Enter your choice"

switch ($choice) {
    "1" { List-Instances }
    "2" { Start-Instance }
    "3" { Stop-Instance }
    "4" { Terminate-Instance }
    "5" { Write-Host "Exiting..."; exit }
    default { Write-Host "Invalid choice! Please try again." }
```

(4.3)

- Right click on the folder where the script is saved and click on run with PowerShell.
- This will open as follows:

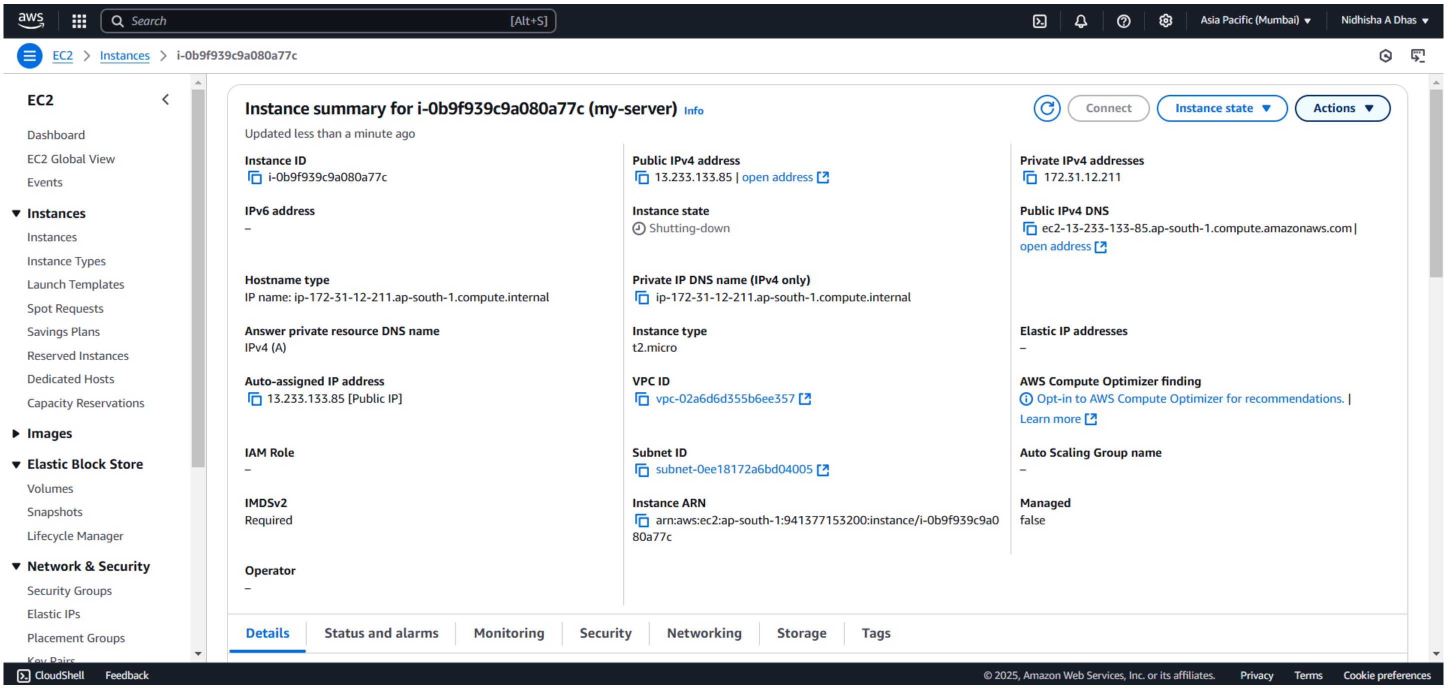


```
Select an option:
1) List Instances
2) Start Instance
3) Stop Instance
4) Terminate Instance
5) Exit
Enter your choice: 4
Enter Instance ID to terminate: i-0b9f939c9a080a77c
```

- Enter any of your choice. Here, to terminate the instance, give choice 4.
- You will see your instance getting terminated.

STEP 5: VERIFICATION

- Go to your AWS console, click on your EC2 instance and you will notice that your EC2 instance is getting terminated.



Instance summary for i-0b9f939c9a080a77c (my-server) Info

Updated less than a minute ago

Instance ID
i-0b9f939c9a080a77c

Public IPv4 address
13.233.133.85 | open address

Private IPv4 addresses
172.31.12.211

IPv6 address
–

Instance state
Shutting-down

Public IPv4 DNS
ec2-13-233-133-85.ap-south-1.compute.amazonaws.com | open address

Hostname type
IP name: ip-172-31-12-211.ap-south-1.compute.internal

Private IP DNS name (IPv4 only)
ip-172-31-12-211.ap-south-1.compute.internal

Answer private resource DNS name
IPv4 (A)

Instance type
t2.micro

Elastic IP addresses
–

Auto-assigned IP address
13.233.133.85 [Public IP]

VPC ID
vpc-02a6d6d355b6ee357

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role
–

Subnet ID
subnet-0ee18172a6bd04005

Auto Scaling Group name
–

IMDSv2
Required

Instance ARN
arn:aws:ec2:ap-south-1:941377153200:instance/i-0b9f939c9a080a77c

Managed
false

Operator
–

CONCLUSION:

By completing this POC on managing AWS cloud resources using the CLI and a shell script, you will:

- Automate essential EC2 instance management tasks, including launching, stopping, and terminating VMs, through a menu-driven shell script.
- Gain hands-on experience with AWS CLI for interacting with cloud resources programmatically, building your foundation for advanced automation.
- Enhance your skills in shell scripting and cloud resource management, critical for DevOps and cloud engineering roles.
- Understand key AWS services like EC2, IAM (for key pairs), and security groups, along with best practices in cloud cost optimization.