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: Hema S

Department: ECE

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.

Overview

This POC focuses on creating a shell script to manage AWS EC2 instances using the AWS CLI. The script allows users to:

1. Launch new EC2 instances with pre-configured settings.
2. Stop running EC2 instances to optimize costs.
3. Terminate EC2 instances when no longer needed.
4. List currently running EC2 instances for better resource tracking.

The script uses a menu-driven approach, where users can choose specific actions, making it user-friendly and flexible. It is tested using Git Bash on Windows and adheres to AWS Free Tier limitations to ensure cost-effective implementation.

Objective

The primary objective of this POC is to:

1. Automate the management of AWS EC2 instances through shell scripting.

2. Provide an easy-to-use interface for launching, stopping, terminating, and listing instances.

3. Demonstrate the capabilities of AWS CLI and shell scripting for cloud resource management.

4. Build a foundational understanding of automation practices in cloud computing.

Importance

1. Efficiency: Automating cloud resource management reduces time and effort spent on manual tasks.

2. Cost Optimization: The ability to stop or terminate unused VMs prevents unnecessary expenses, adhering to best practices in cloud cost management.

3. Scalability: Scripting provides a scalable solution for managing multiple resources simultaneously.

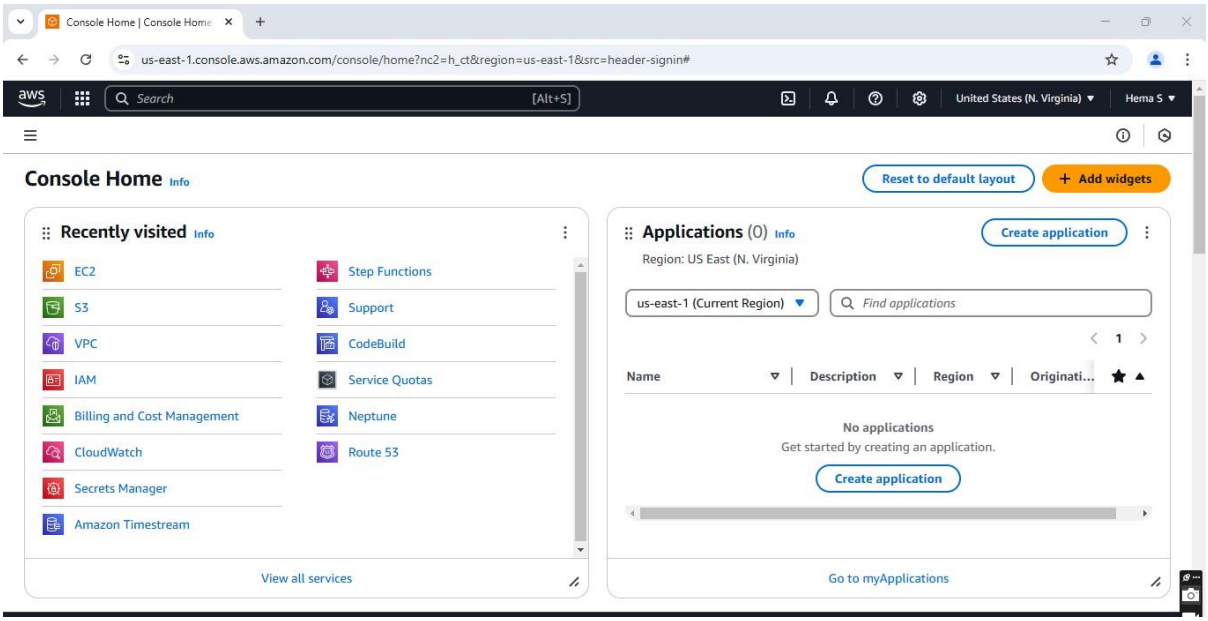
4. Skill Development: Enhances your technical expertise in AWS CLI, scripting, and cloud automation, which are in high demand in the IT industry.

5. 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:

1. Go to AWS Management Console.
2. Enter your username and password to log in.



Step 2:

Make sure your AWS CLI is installed and configured.

```
C:\Users\sppra>aws --version
aws-cli/2.24.10 Python/3.12.9 Windows/11 exe/AMD64
```

Step 3:

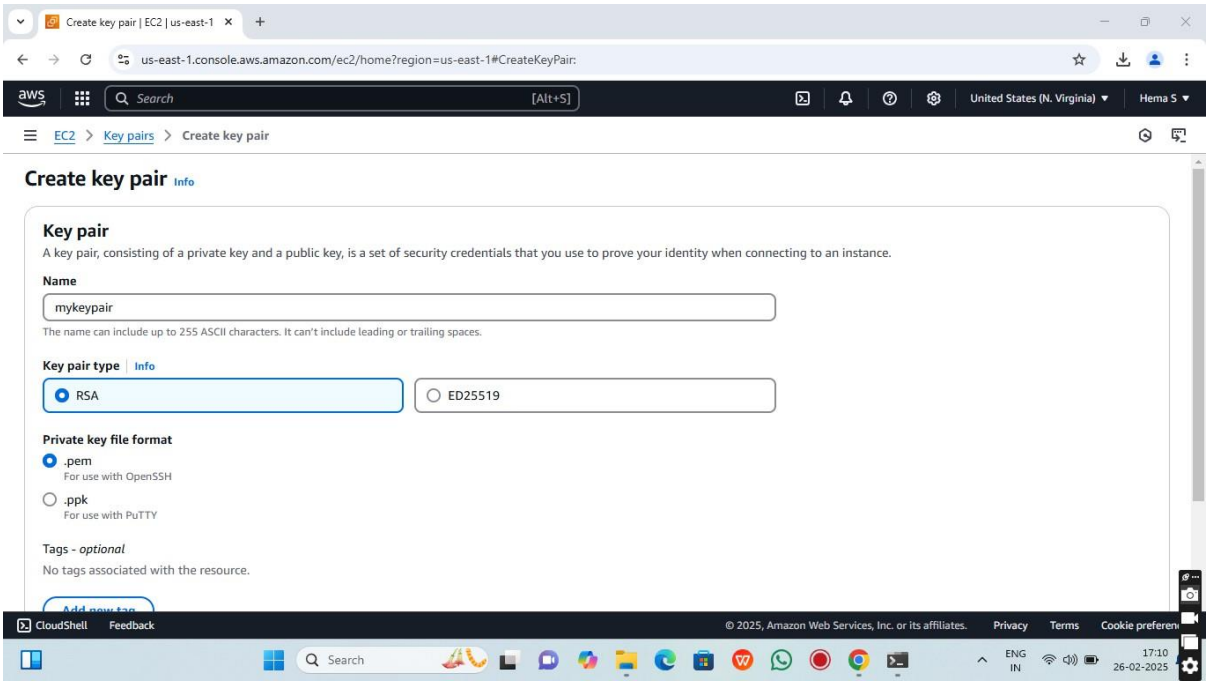
1. Go to the **EC2 Dashboard**.

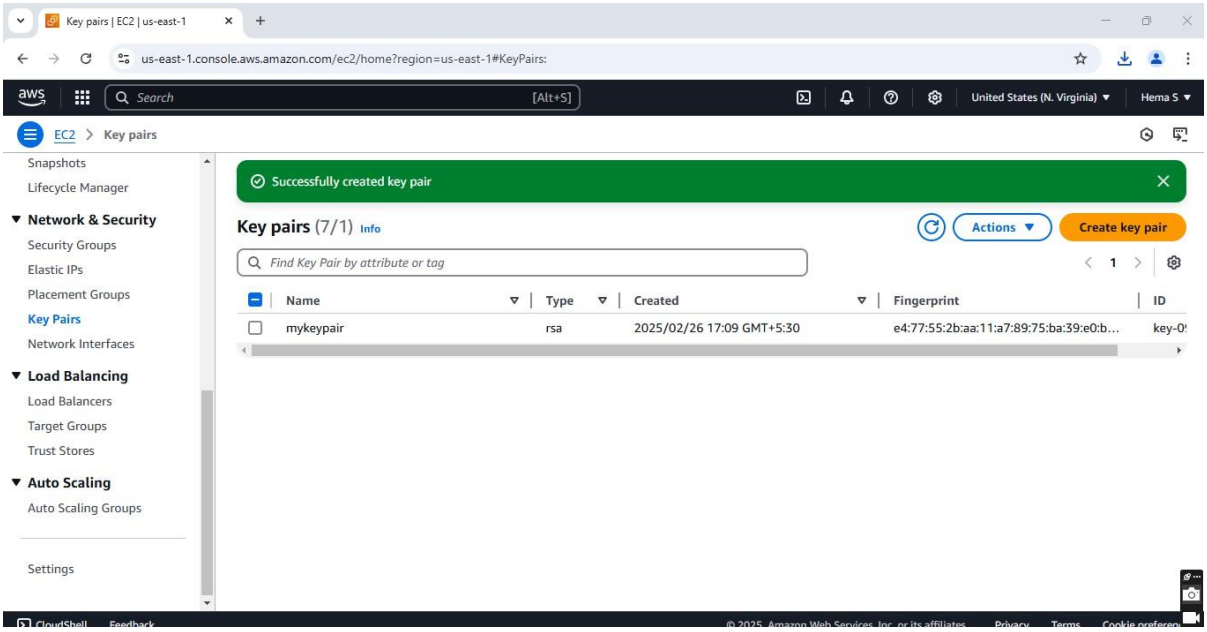
2. In the left sidebar, click **Key Pairs** under **Network & Security**.

3. Click **Create Key Pair**.

4. Enter a name (e.g., MyKeyPair) and choose **.pem** format.

5. Download the .pem file and keep it safe—you'll need it to SSH into your instance.




Key pairs | EC2 | us-east-1
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#KeyPairs:
Search
[Alt+S]
United States (N. Virginia)
Hema S
EC2
Key pairs
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Trust Stores
Auto Scaling
Auto Scaling Groups
Settings
Successfully created key pair
Key pairs (7/1) Info
Actions
Create key pair
Find Key Pair by attribute or tag
1
Name
Type
Created
Fingerprint
ID
mykeypair
rsa
2025/02/26 17:09 GMT+5:30
e4:77:55:2b:aa:11:a7:89:75:ba:39:e0:b...
CloudShell
Feedback
Privacy
Terms

Step 4:

1. Go to the **AWS EC2 Dashboard**.

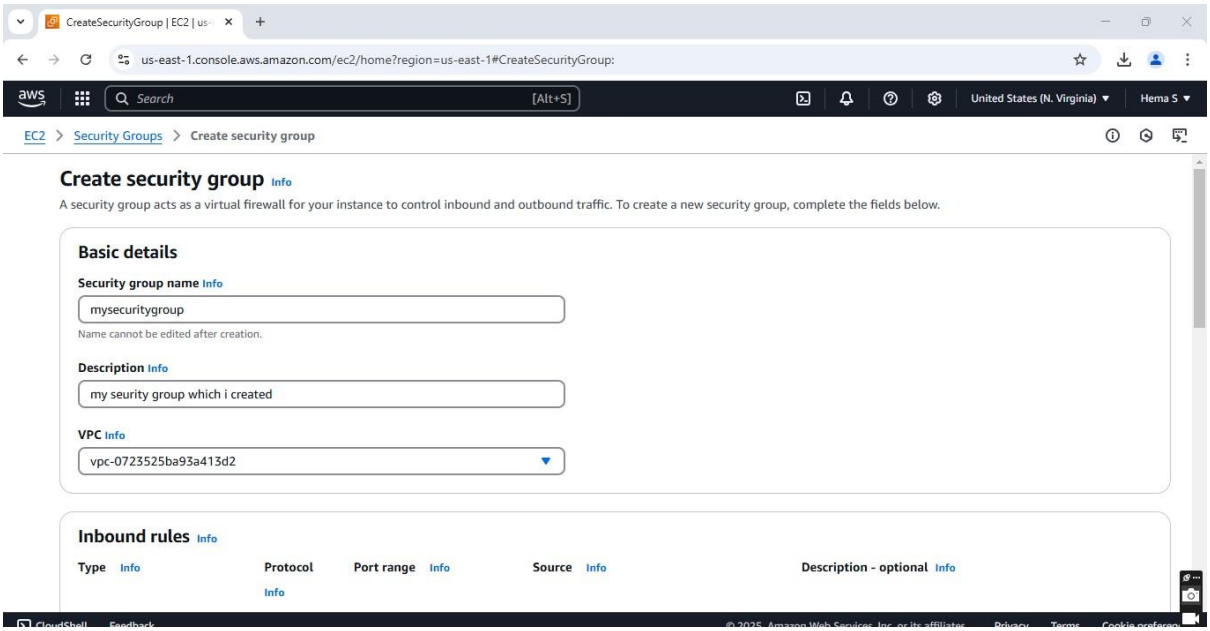
2. In the left sidebar, click **Security Groups**.

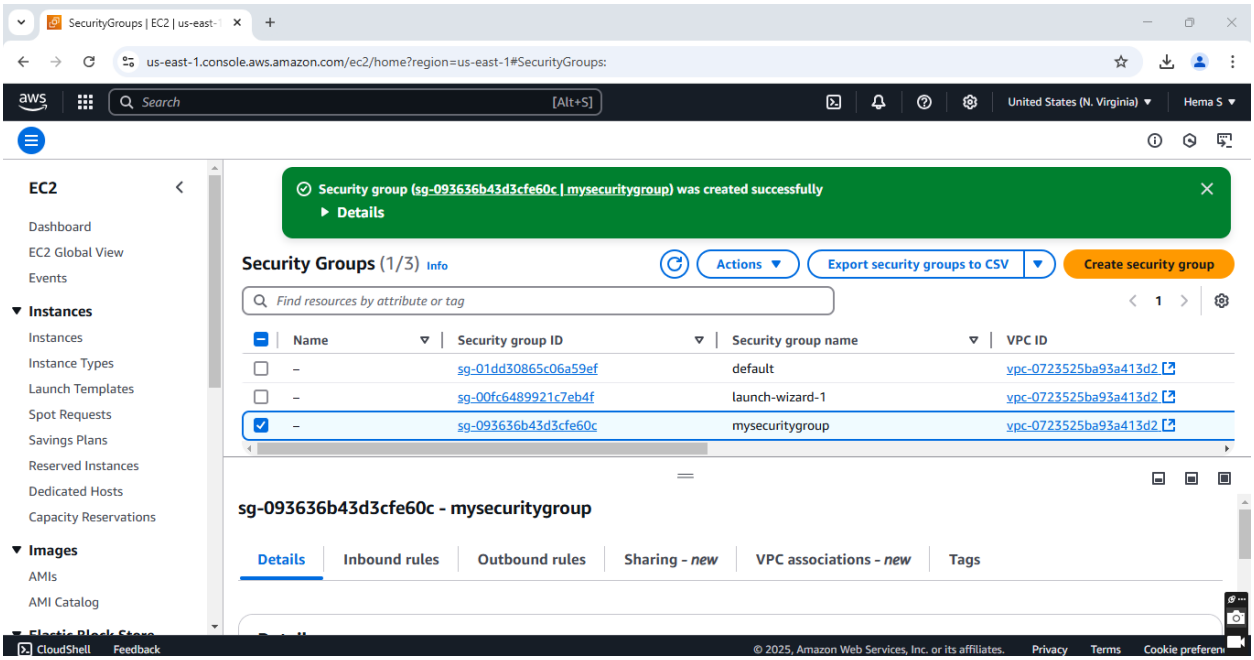
3. Click **Create Security Group**.

4. Enter a name (e.g., MySecurityGroup) and a description.

5. Add the following inbound rule:

- **Type**: SSH
- **Protocol**: TCP
- **Port Range**: 22
- **Source**: Anywhere (0.0.0.0/0) (Note the Id after created)

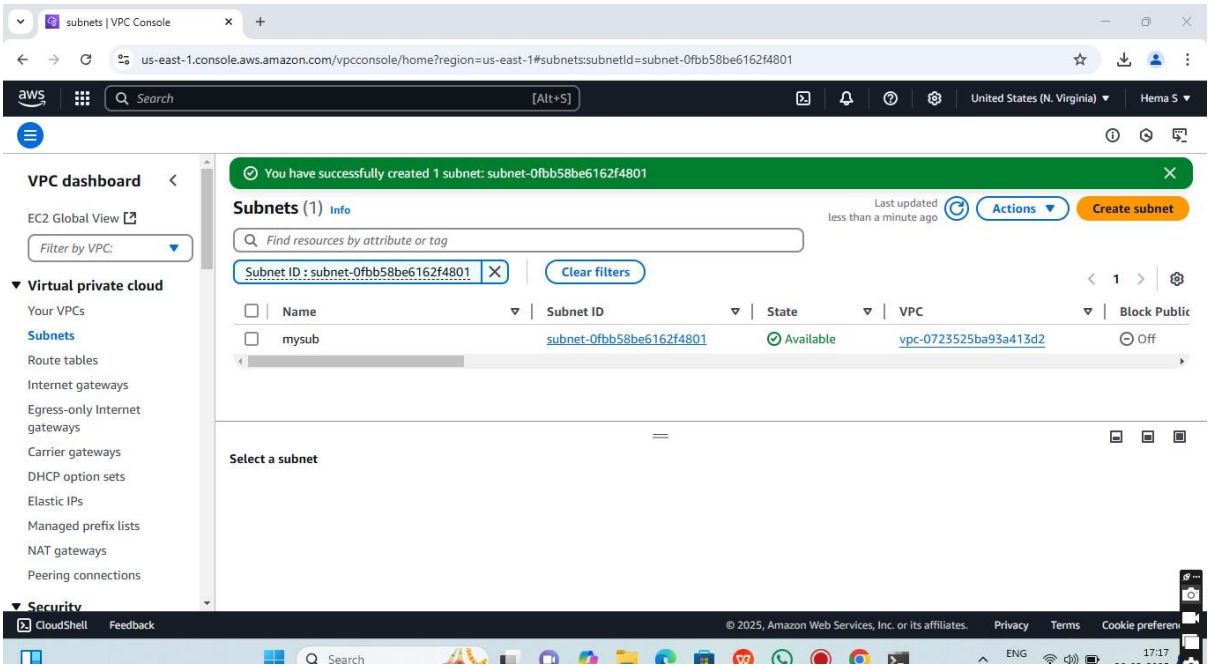



SecurityGroups | EC2 | us-east-
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroups:
Search
[Alt+S]
United States (N. Virginia)
Hema S
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Security group (sg-093636b43d3cfe60c | mysecuritygroup) was created successfully
Details
Security Groups (1/3) Info
Actions
Export security groups to CSV
Create security group
Find resources by attribute or tag
Name
Security group ID
Security group name
VPC ID
sg-01dd30865c06a59ef
default
vpc-0723525ba93a413d2
sg-00fc6489921c7eb4f
launch-wizard-1
vpc-0723525ba93a413d2
sg-093636b43d3cfe60c
mysecuritygroup
vpc-0723525ba93a413d2
sg-093636b43d3cfe60c - mysecuritygroup
Details
Inbound rules
Outbound rules
Sharing - new
VPC associations - new
Tags
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms

Step 5:

1. In the **AWS VPC Dashboard**, click **Subnets** in the left sidebar.

2. Note the **Subnet ID** of one of your subnets. Example: subnet-0abcd1234.

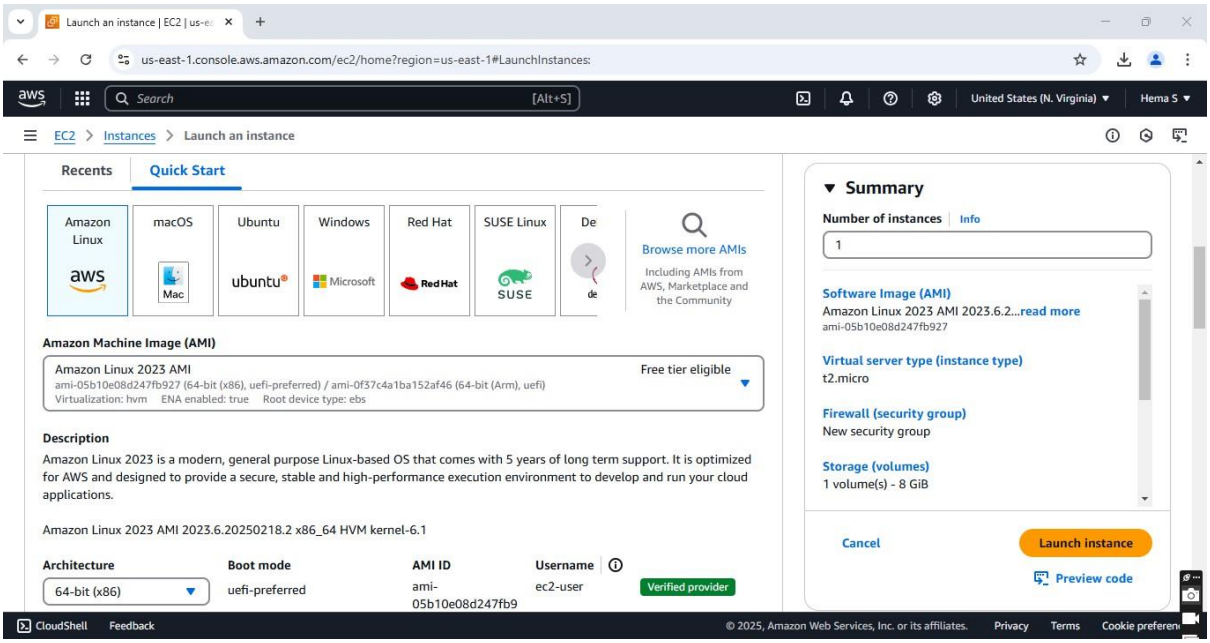


Step 6:

1. In the **AWS EC2 Dashboard**, click **Launch Instance**.

2. Search for "Amazon Linux 2" and select it.

3. Note the **AMI ID** (e.g., ami-0c02fb55956c7d316).


Launch an instance | EC2 | us-e
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Search
[Alt+S]
United States (N. Virginia)
Hema S
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)
Amazon Linux 2023 AMI
ami-05b10e08d247fb927 (64-bit (x86), uefi-preferred) / ami-0f37c4a1ba152af46 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs
Free tier eligible
Description
Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.
Amazon Linux 2023 AMI 2023.6.20250218.2 x86_64 HVM kernel-6.1
Architecture
64-bit (x86)
Boot mode
uefi-preferred
AMI ID
ami-05b10e08d247fb9
Username
ec2-user
Verified provider
Summary
Number of instances
Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-05b10e08d247fb927
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

Step 7:

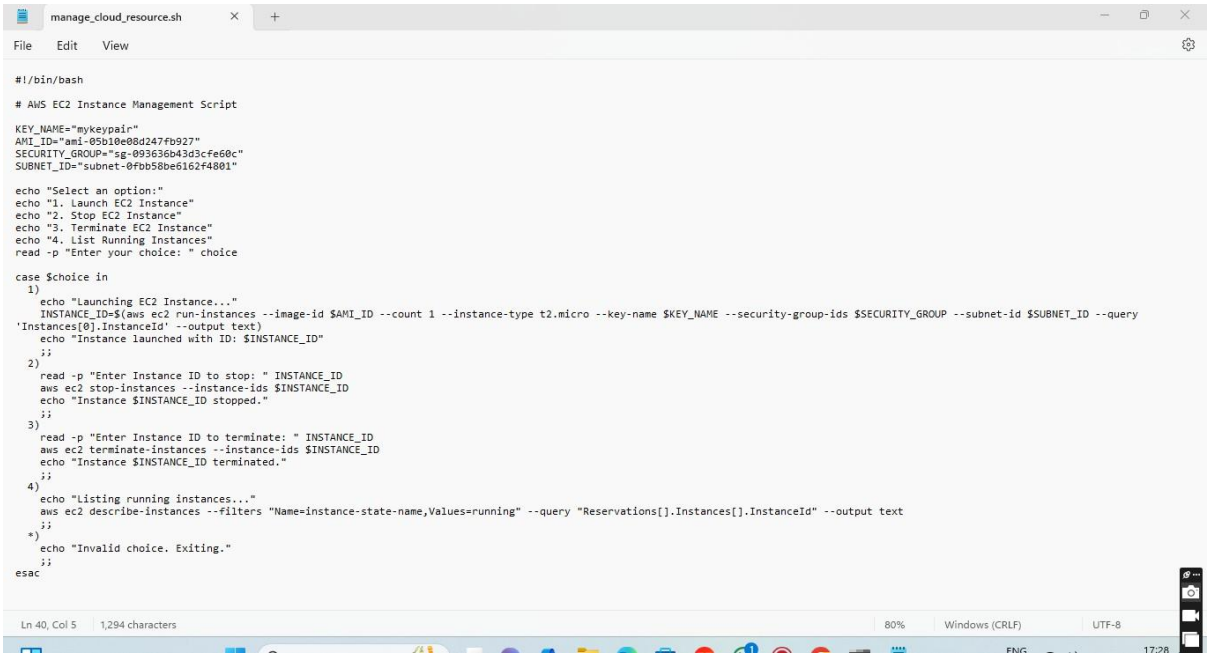
Here's a simple shell script to manage cloud resources (launch, stop, and terminate VMs) using the AWS CLI.

Open **Notepad.**

Paste the script into the Notepad.

Replace the placeholders (YourKeyPairName, YourSecurityGroupID, etc.) with your actual values:

- **Key Pair Name**: Replace with the name of your key pair.
- **Security Group ID**: Replace with your security group ID.
- **Subnet ID**: Replace with your subnet ID.
- **AMI ID**: Replace with the AMI ID.



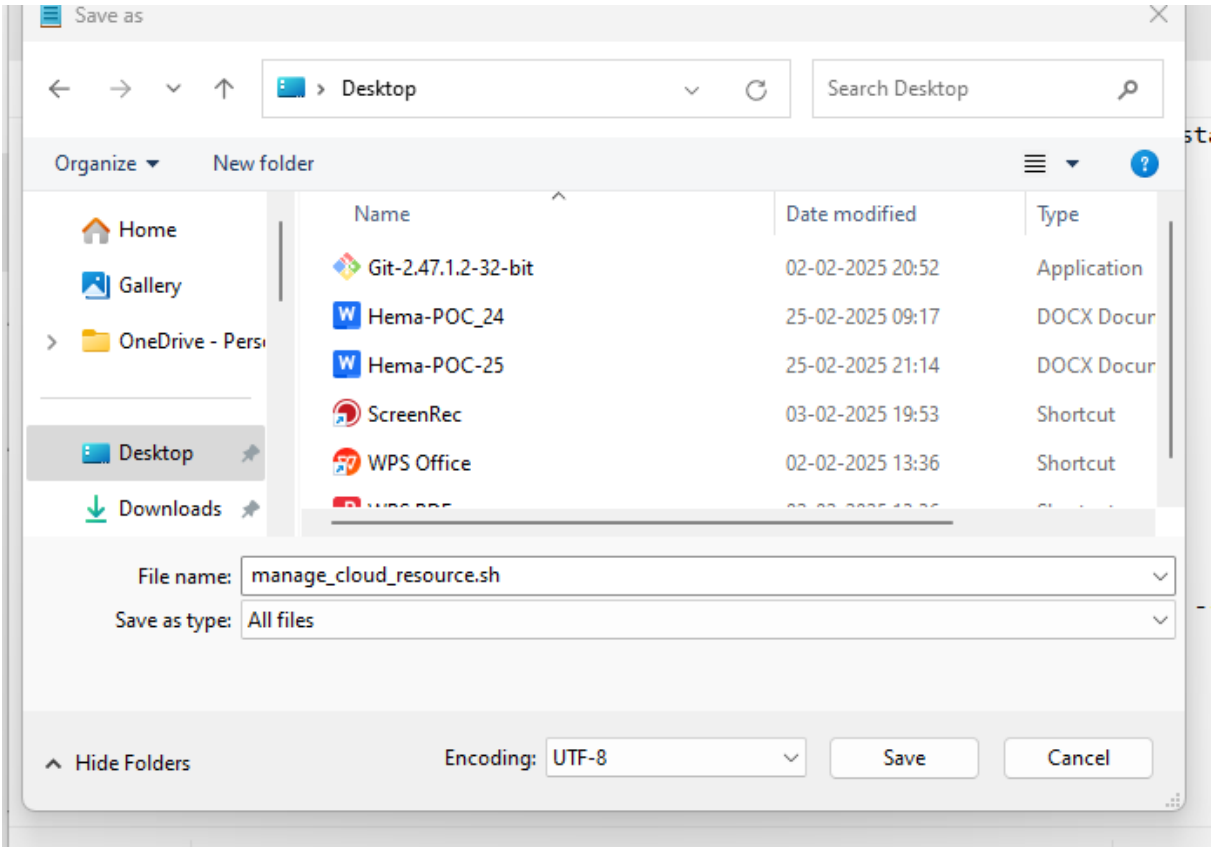
Step 8:

1. Click **File → Save As**.

2. In the **Save As** window:

   - **File Name**: Enter manage_cloud_resources.sh.
   - **Save as type**: Select **All Files** from the dropdown.
   - **Encoding**: Select **UTF-8** (if available).
   - **Location**: Save it in Desktop.

Important: Make sure the file has the .sh

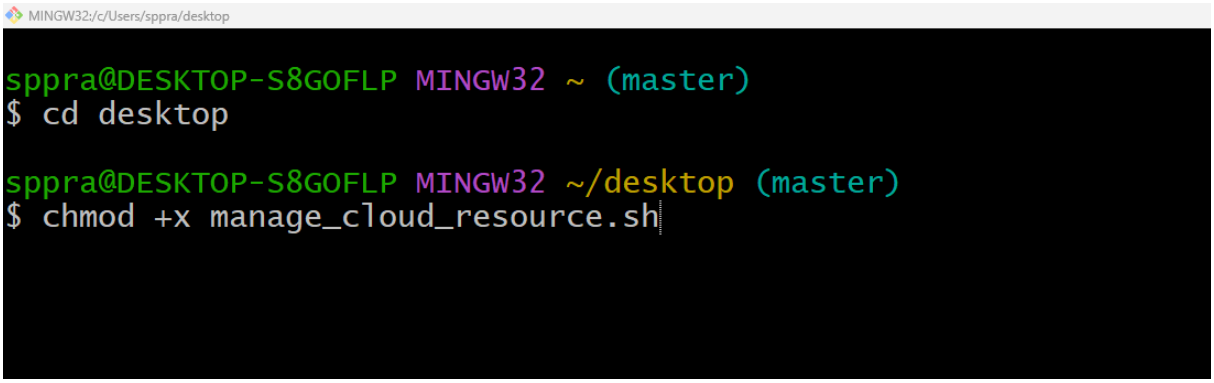


Step 9:

1. Open Git Bash

2. Run the following command in Git Bash:

chmod +x manage_cloud_resources.sh



Step 10:

Run the script using:

./manage_cloud_resources.sh

```
sppra@DESKTOP-S8GOFLP MINGW32 ~/desktop (master)
$ ./manage_cloud_resource.sh
Select an option:
1. Launch EC2 Instance
2. Stop EC2 Instance
3. Terminate EC2 Instance
4. List Running Instances
Enter your choice:
```

Step 11:

1. Select 1 to launch an instance.

2. The script will create an EC2 instance and display its **Instance ID**. Make a note of this ID for the next steps.

```
sppra@DESKTOP-S8GOFLP MINGW32 ~/desktop (master)
$ ./manage_cloud_resource.sh
Select an option:
1. Launch EC2 Instance
2. Stop EC2 Instance
3. Terminate EC2 Instance
4. List Running Instances
Enter your choice: 1
Launching EC2 Instance...
Instance launched with ID: i-0fda7a05e48314814
```

Step 12:

1. Select 2 to stop an instance.

2. Enter the **Instance ID** of the instance you launched earlier.

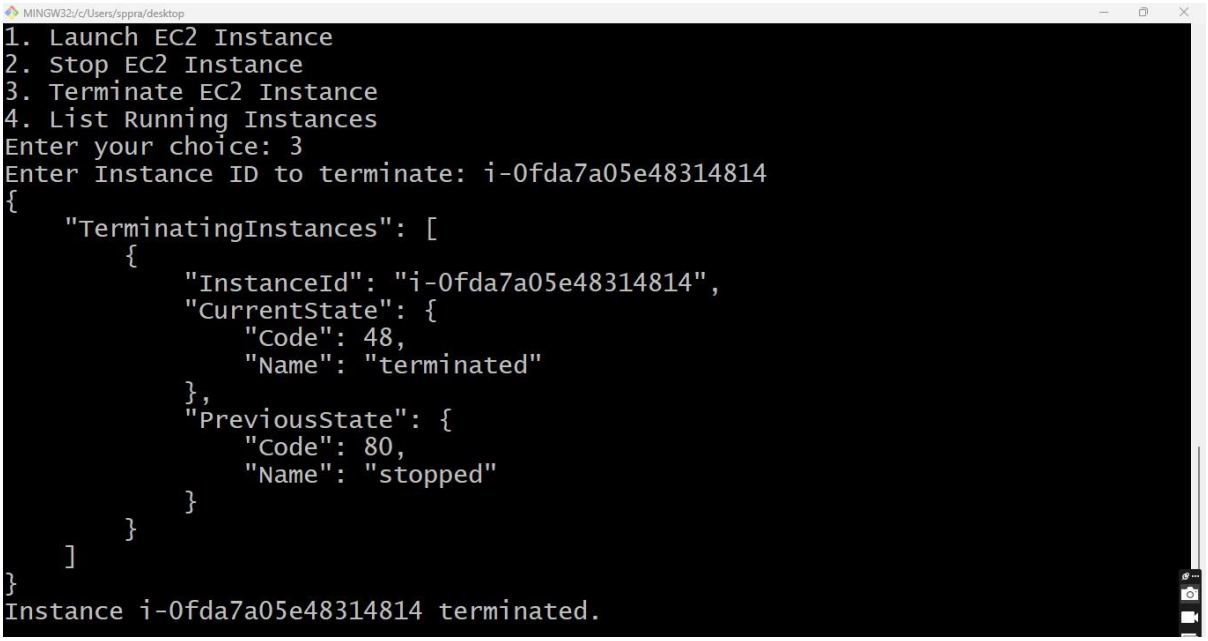
3. The script will stop the instance.

Step 13:

1. Select 3 to terminate an instance.

2. Enter the **Instance ID** of the instance you launched earlier.

3. The script will terminate the instance.



Successfully completed the PoC!

Outcome

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

1. Automate essential EC2 instance management tasks, including launching, stopping, and terminating VMs, through a menu-driven shell script.

2. Efficiently manage multiple EC2 instances using AWS CLI commands integrated with shell scripting, ensuring scalability and consistency.

3. Gain hands-on experience with AWS CLI for interacting with cloud resources programmatically, building your foundation for advanced automation.

4. Enhance your skills in shell scripting and cloud resource management, critical for DevOps and cloud engineering roles.

5. Understand key AWS services like EC2, IAM (for key pairs), and security groups, along with best practices in cloud cost optimization.

6. Validate the practical implementation of a script by successfully launching, stopping, and terminating multiple EC2 instances.