



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



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: CHANDRU S

Department: INFORMATION TECHNOLOGY



Introduction

This Proof of Concept (POC) aims to develop a shell script for automating cloud virtual machine (VM) management via a Command-Line Interface (CLI). The script enables users to efficiently launch, stop, and terminate VMs, streamlining operations while minimizing manual effort. By automating these tasks, the solution enhances consistency, reliability, and operational efficiency in cloud infrastructure management.

Overview

- The shell script automates key cloud VM operations, allowing seamless interaction with cloud providers through the CLI.
- It simplifies routine tasks by providing a user-friendly interface for launching, stopping, and terminating VMs.
- Automation reduces human errors and saves time, ensuring reliable cloud resource management.
- Designed for adaptability, the script can be customized for different cloud platforms and integrated into broader automation workflows.

Objectives

The primary goals of this task are:

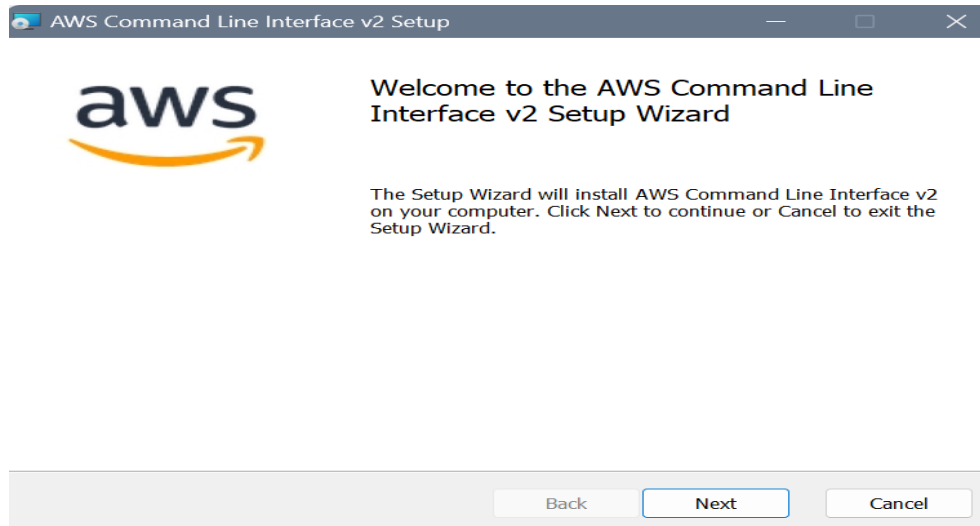
- **Automation:** Develop a shell script to manage cloud VMs efficiently using the cloud provider's CLI.
- **Functionality:** Enable users to launch, stop, and terminate VMs with minimal manual intervention.
- **Efficiency:** Reduce human errors and save time by automating repetitive cloud management tasks.
- **Scalability:** Provide a flexible and reusable solution that can be adapted to different cloud environments or integrated into complex automation workflows.

Step-by-Step Overview

Step 1:

- **Install AWS CLI:** Follow the installation guide specific to your operating system.
- **Configure AWS CLI:** Run the following command and enter your AWS credentials:
 - First, create an EC2 instance in the AWS Management Console.

- Gather your AWS Access Key ID, Secret Access Key, region, and output format for configuration.



Step 2:

Run the following command in your terminal or command prompt:

aws configure

You'll be prompted to enter:

- AWS Access Key ID
- AWS Secret Access Key
- Default region name
- Default output format (e.g., JSON)

```
PS C:\Users\chandru> aws configure
AWS Access Key ID [*****IPOG]: AKIAWX2IFIP3HFWLY7M7
AWS Secret Access Key [*****A6Wj]: GsVAMPPrfPrAe6jtyKP3pwiOU8LEI7osxPnOG4R3
Default region name [ap-south-1]: ap-south-1
Default output format [json]: json
```

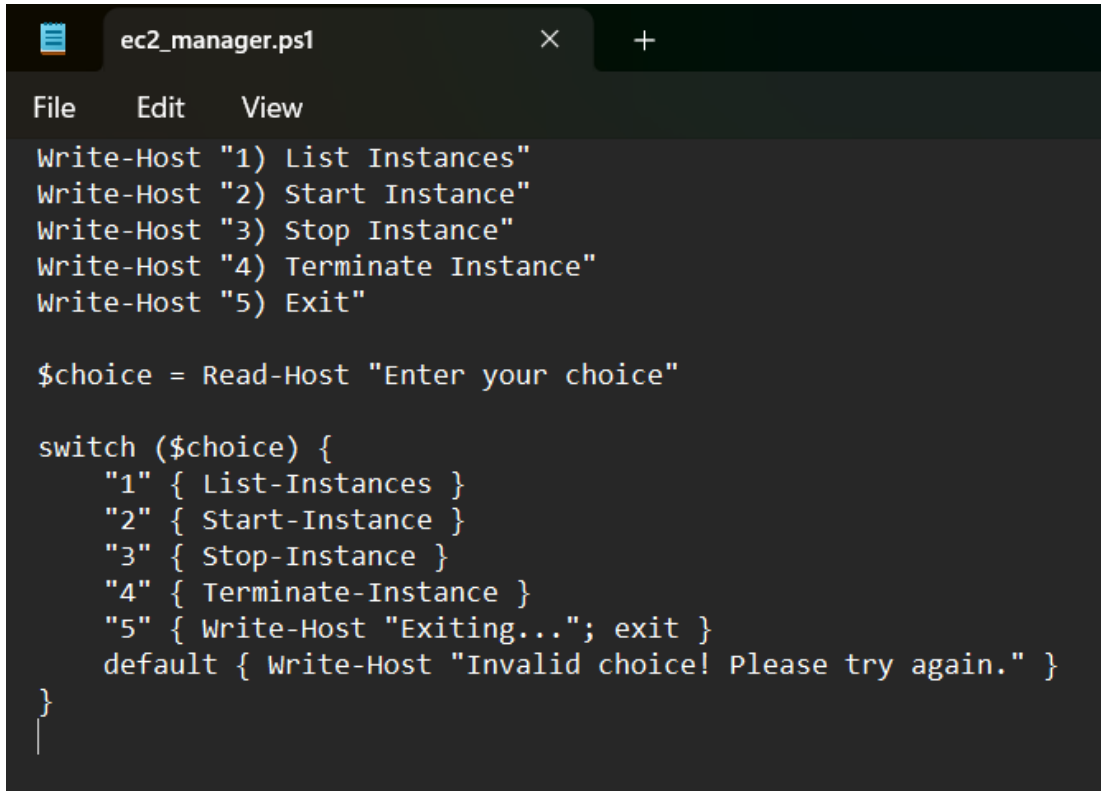
Step 3:

- Start an EC2 instance using the AWS CLI with the necessary configurations.
- Ensure you note the Instance ID, as it will be required in later steps.

```
PS C:\Users\chandru\OneDrive\Desktop\Scripts> aws ec2 run-instances --image-id ami-0c50b6f7dc3701ddd --count 1 --instance-
type t2.micro --key-name AK-key --security-group-ids sg-037133b8f3f618401 --subnet-id subnet-0d9bde7f231293042
{
  "ReservationId": "r-03fb5552defcacc60",
  "OwnerId": "713881791819",
  "Groups": [],
  "Instances": [
    {
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "61221340-c906-465b-9d01-d2497ed4bf9e",
      "EbsOptimized": false,
      "EnaSupport": true,
```

Step 4:

- Open a text editor (e.g., Notepad) and write the shell script.
- Save the script as `ec2_manager.ps1` in a dedicated folder named "script."

A screenshot of a text editor window titled 'ec2_manager.ps1'. The window has a menu bar with 'File', 'Edit', and 'View'. The script content is as follows:

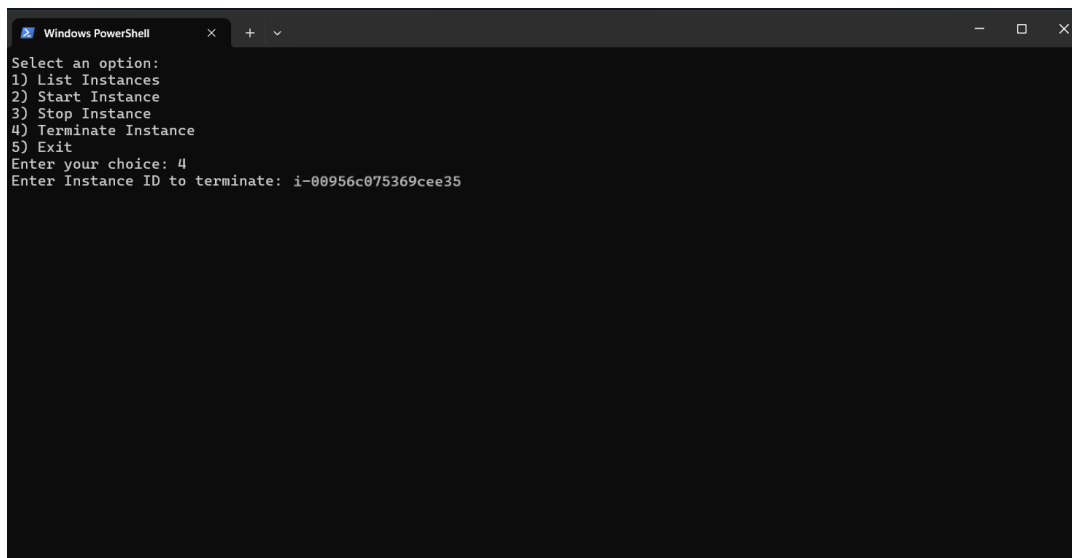
```
Write-Host "1) List Instances"
Write-Host "2) Start Instance"
Write-Host "3) Stop Instance"
Write-Host "4) Terminate Instance"
Write-Host "5) Exit"

$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." }
}
```

Step 5:

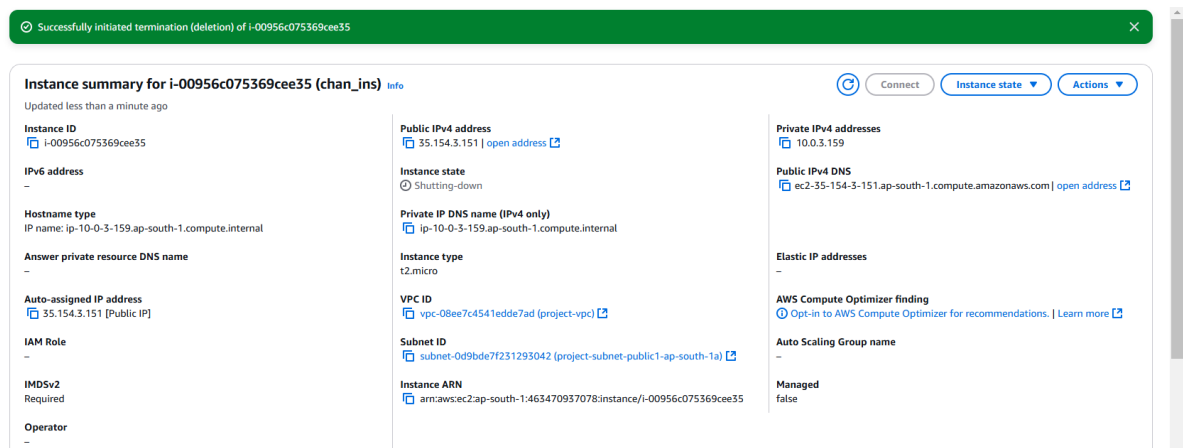
- Right-click on the file and select "Run with PowerShell."
- The script will execute in a PowerShell window.
- Choose the required operation and enter the Instance ID of the EC2 instance you created.

A screenshot of a Windows PowerShell window. The window title is 'Windows PowerShell'. The output of the script is 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-00956c075369cee35
```

Step 6:

Open the AWS Management Console and check whether the selected actions (launch, stop, terminate) have been performed successfully on your EC2 instance.



Outcome

By successfully implementing this Proof of Concept (PoC) to automate cloud resource management using a shell script, the following benefits are achieved:

1. Improved Automation

- Eliminates manual steps in VM management.
- Saves time by executing predefined commands automatically.

2. Efficient Resource Management

- Start or stop VMs based on usage patterns.
- Dynamically resize or reallocate resources.

3. Enhanced Monitoring

- Regularly check VM health, disk usage, CPU, and memory utilization.
- Send alerts when thresholds are exceeded.

4. Security and Access Control

- Implement **SSH key-based** authentication.
- Automate security updates and backups.

5. Cost Optimization

- Automatically stop **unused VMs** to reduce cloud costs.
- Schedule VM operations based on workload demands.