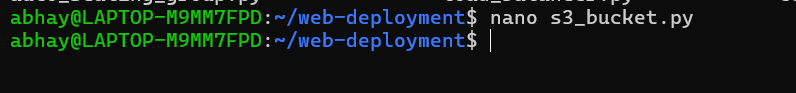
# **AWS Automation and Scaling with boto3 Documentation**

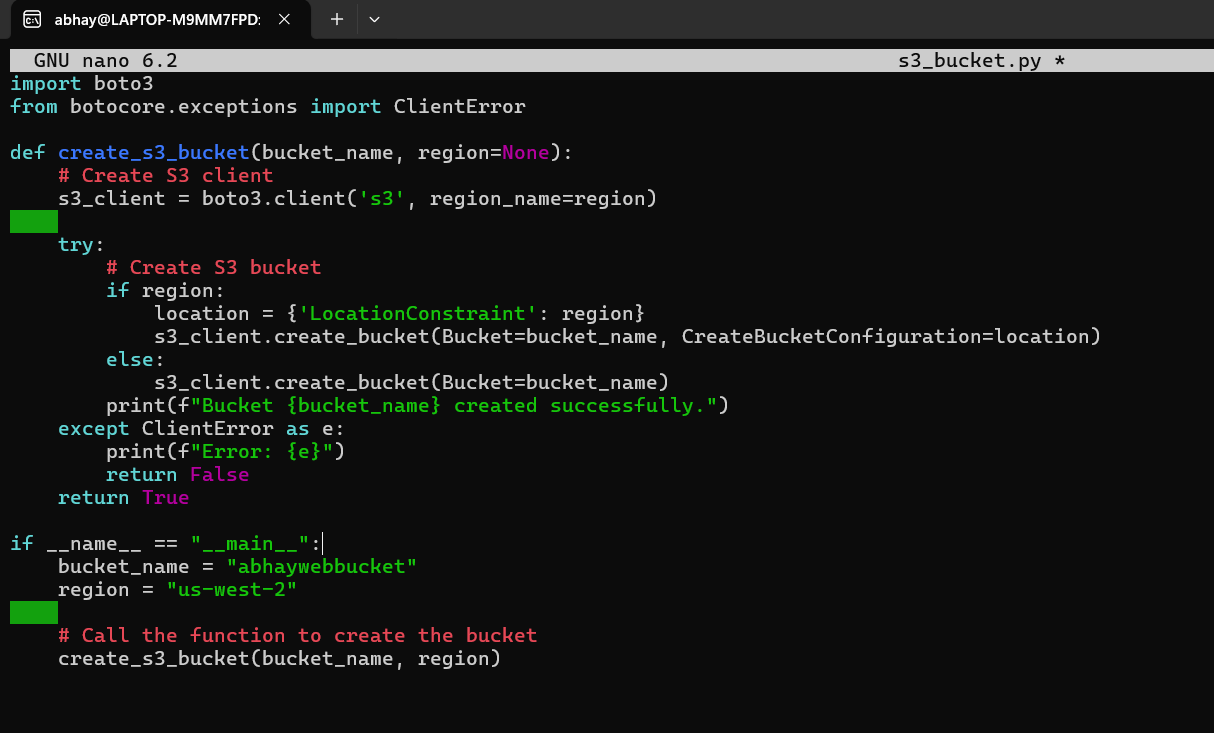
This documentation provides a step-by-step guide to the tasks required for automating and managing AWS infrastructure using Python, boto3, and AWS services.

Table of Contents

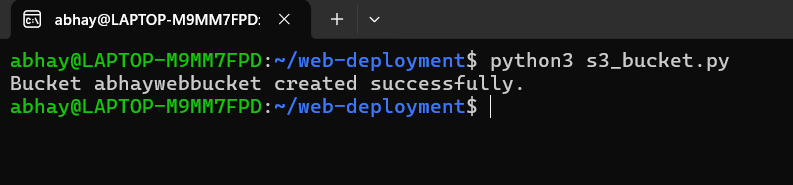
1. [Web Application Deployment on EC2](#web-application-deployment)
2. [Load Balancing with Application Load Balancer](#load-balancing-with-alb)
3. [Auto Scaling Group (ASG) Configuration](#auto-scaling-group-configuration)
4. [SNS Notifications](#sns-notifications)
5. [Infrastructure Automation with boto3](#infrastructure-automation-with-boto3)
6. [Optional Enhancement - Dynamic Content Handling](#optional-enhancement---dynamic-content-)
7. Web Application Deployment on EC2:

Create a New Python File and Write the Following Python Script.





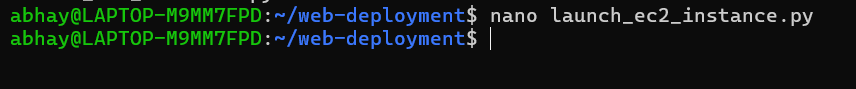
Run the Python Script: “python3 create\_s3\_bucket.py”



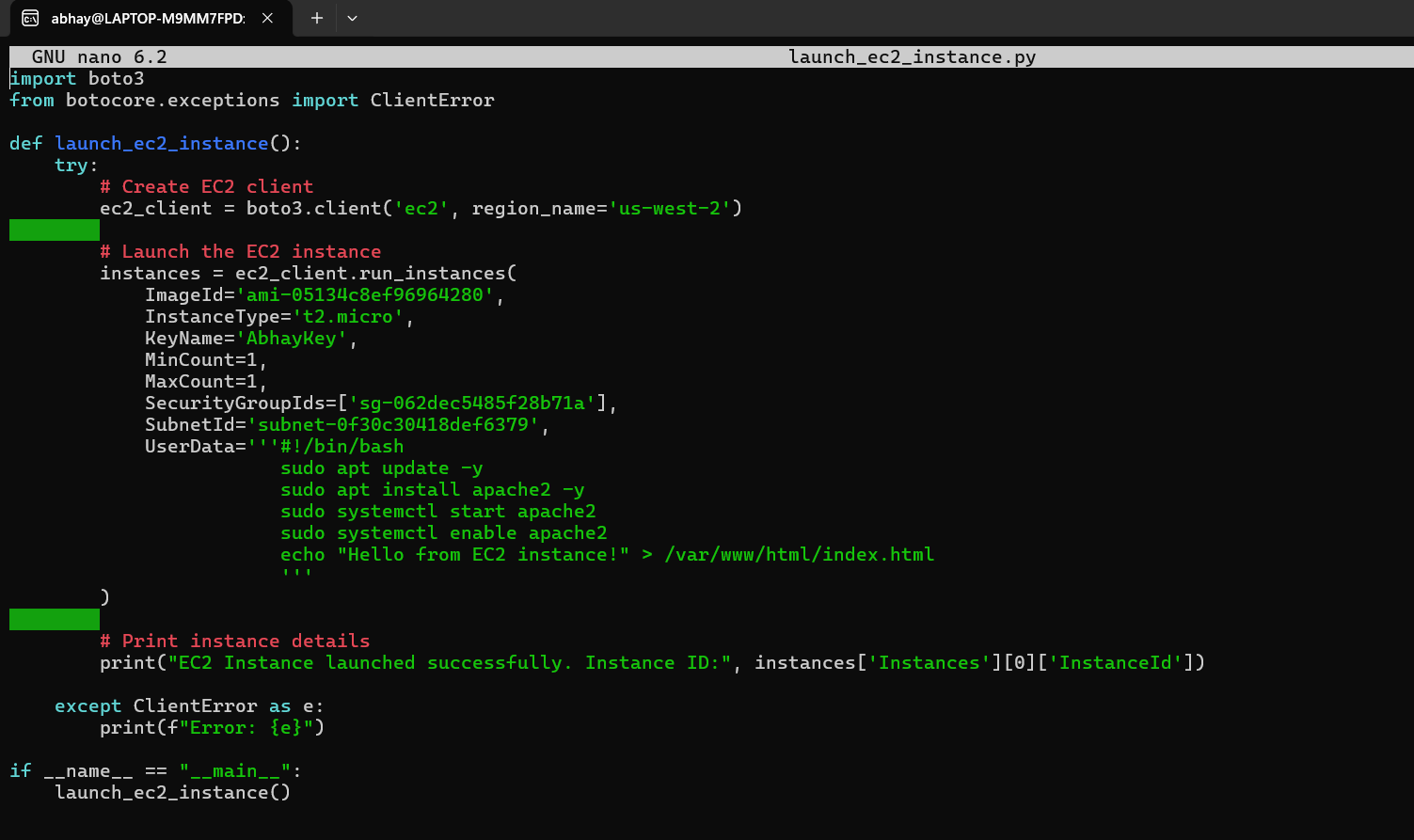


1. Launching an EC2 Instance and Configuring it as a Web Server (Apache/Nginx):
   1. Launch an EC2 Instance Using boto3:

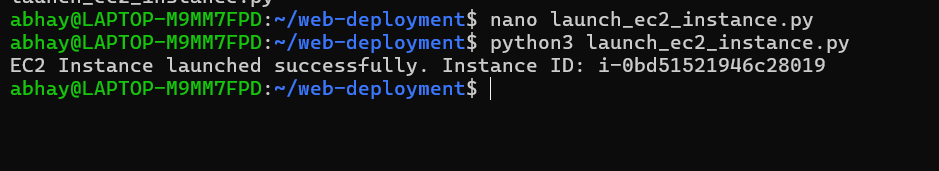
**Create a new Python file** and name it **launch\_ec2\_instance.py.**

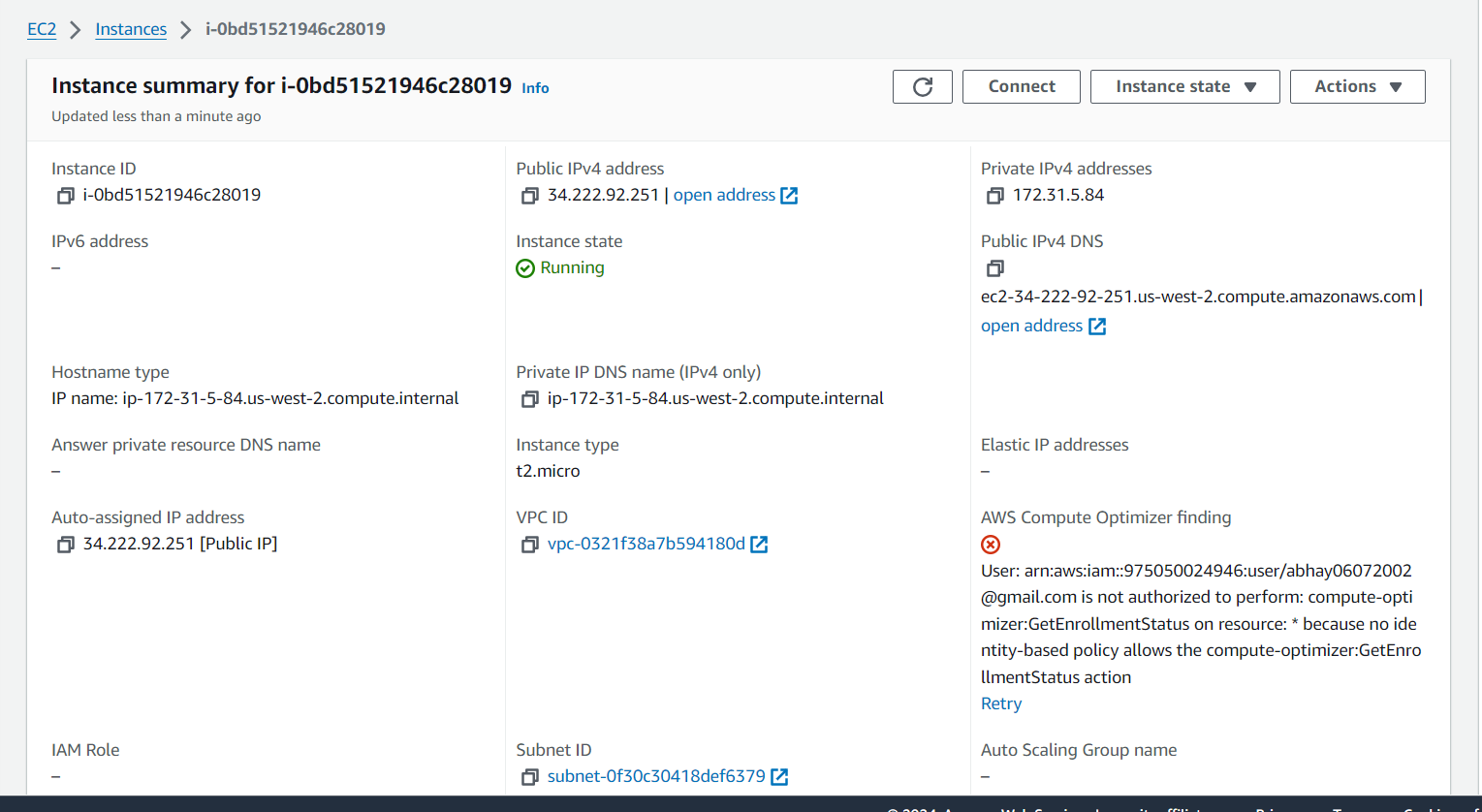
****

**Write the Python script to launch an EC2 instance:**

****

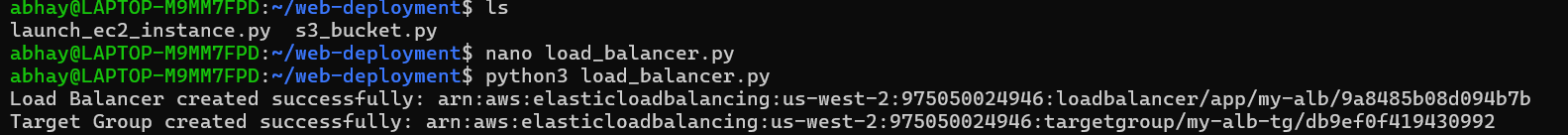
1. Run the Script: python3 launch\_ec2\_instance.py



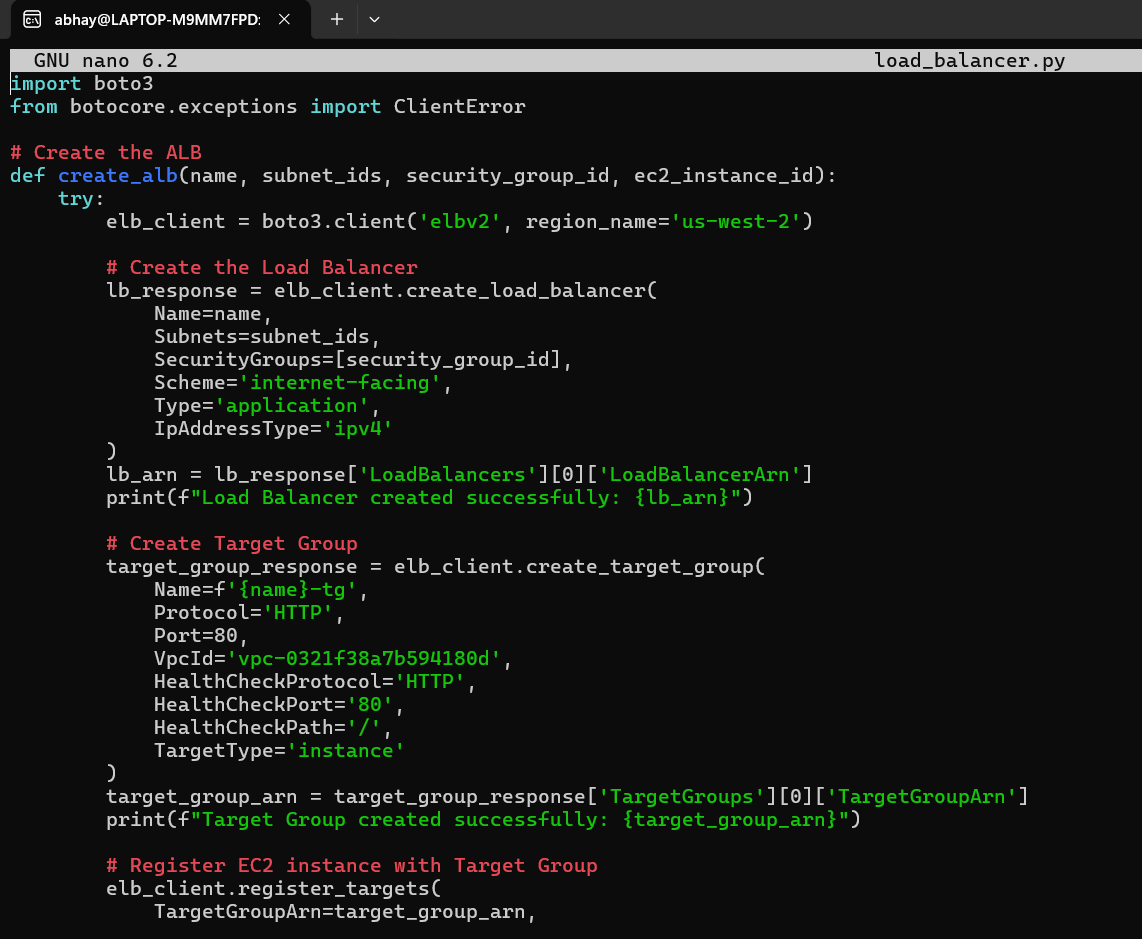


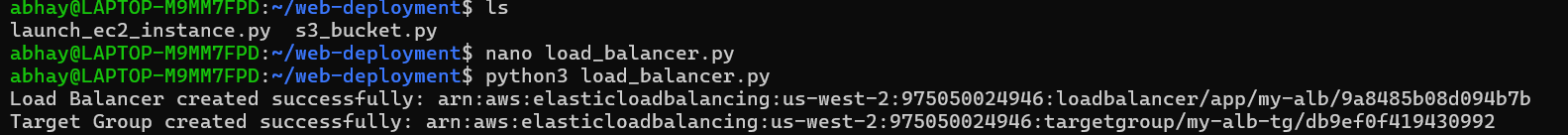
1. Setting Up Load Balancing with ALB:

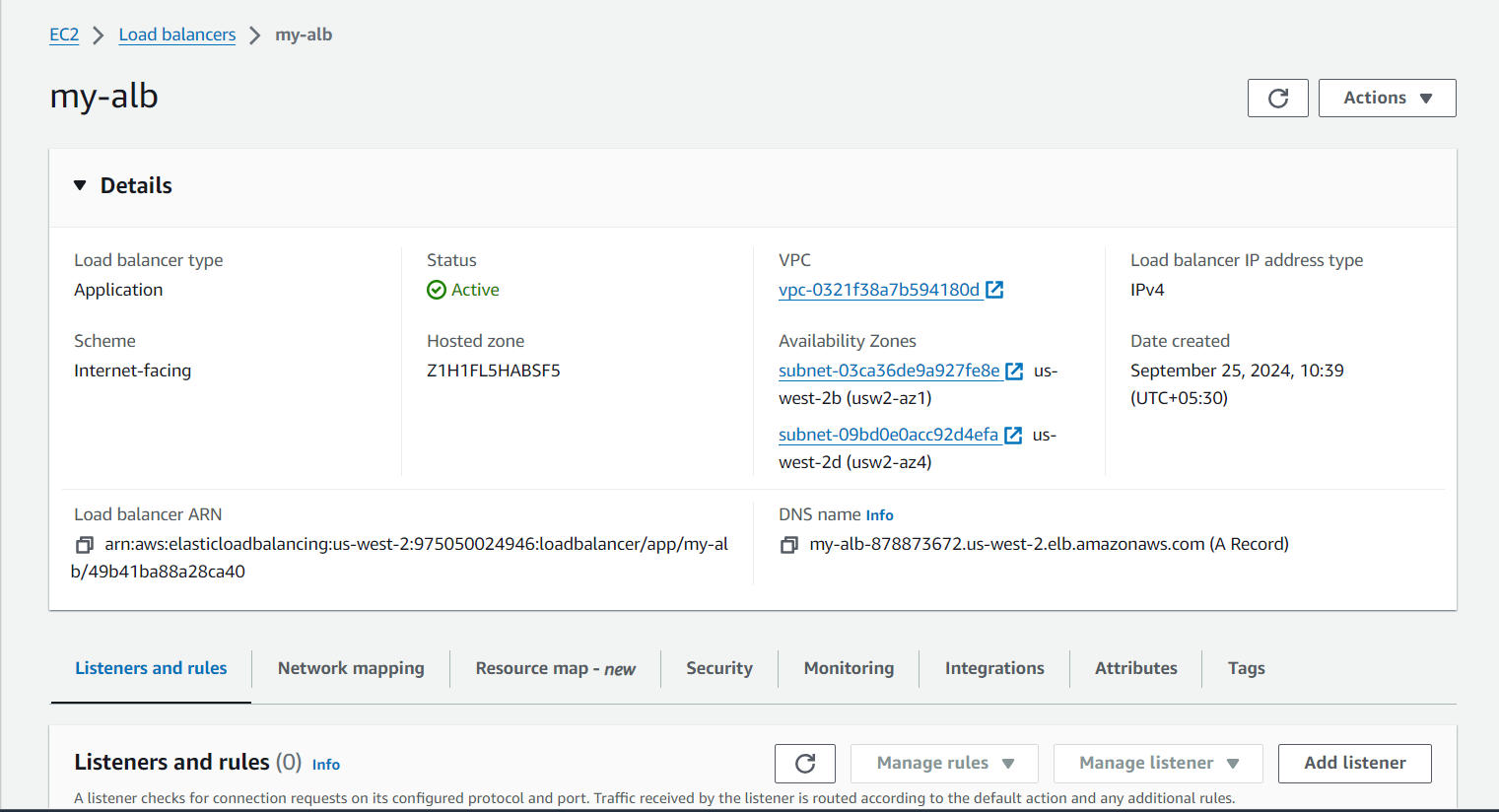
Create a New Python File: “nano load\_balancer.py”



Write the Following Script:

Run the Script: “python3 load\_balancer.py”

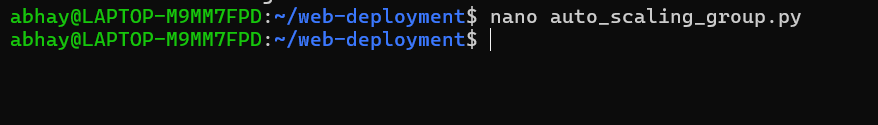




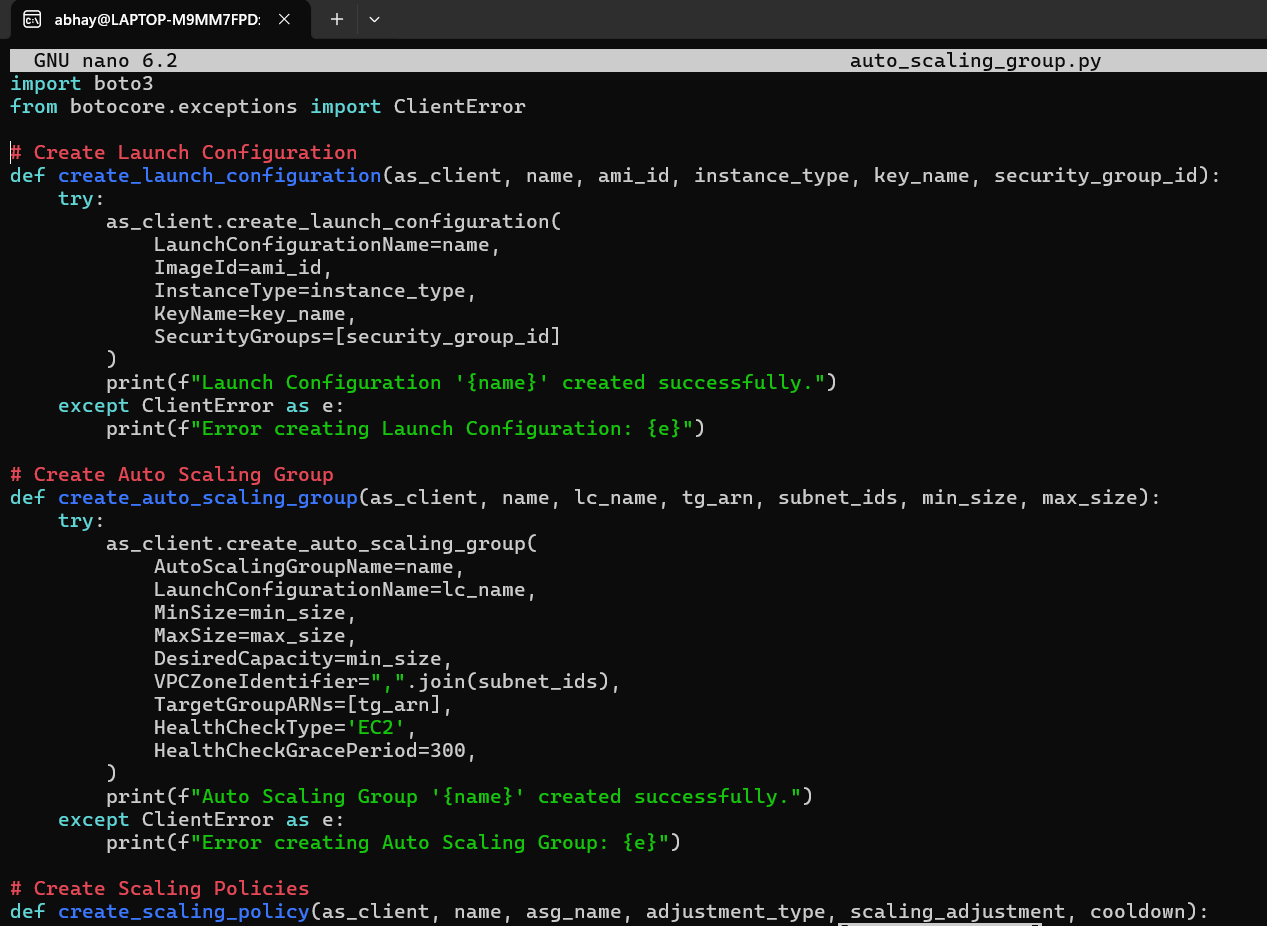
1. Setting Up Auto Scaling Group (ASG):

The Auto Scaling Group will automatically manage the scaling of your EC2 instances based on traffic and metrics like CPU utilization. It will ensure that the appropriate number of EC2 instances are running to handle the traffic.

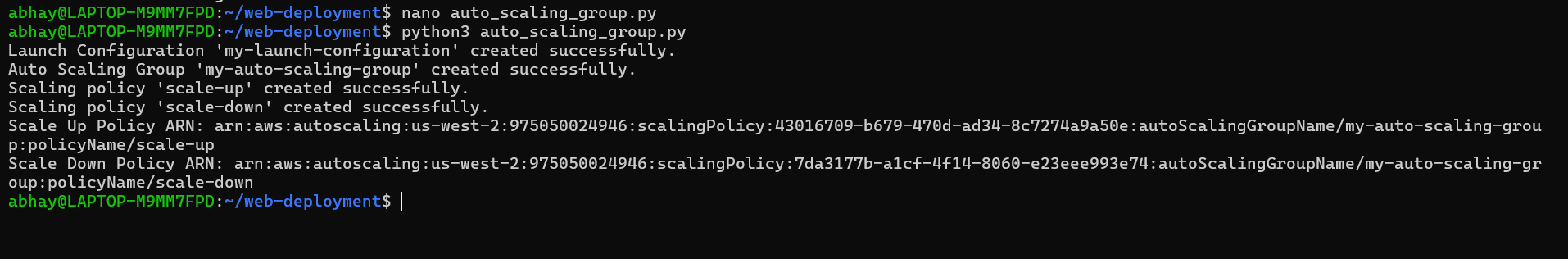
Create a New Python File: “nano auto\_scaling\_group.py”

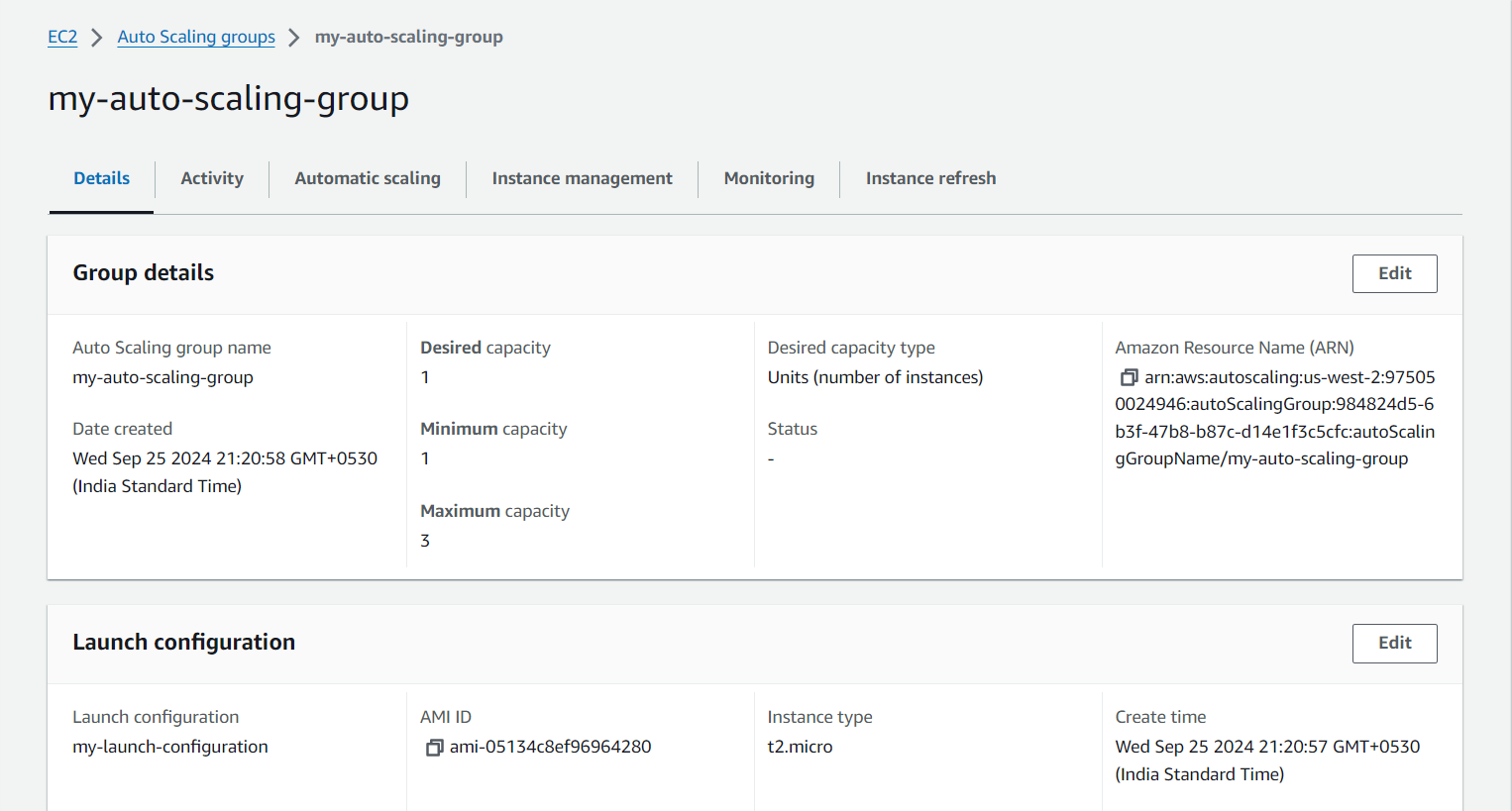


Write the Following Script:



Run the Script: “python3 auto\_scaling\_group.py”

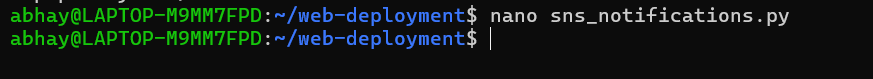




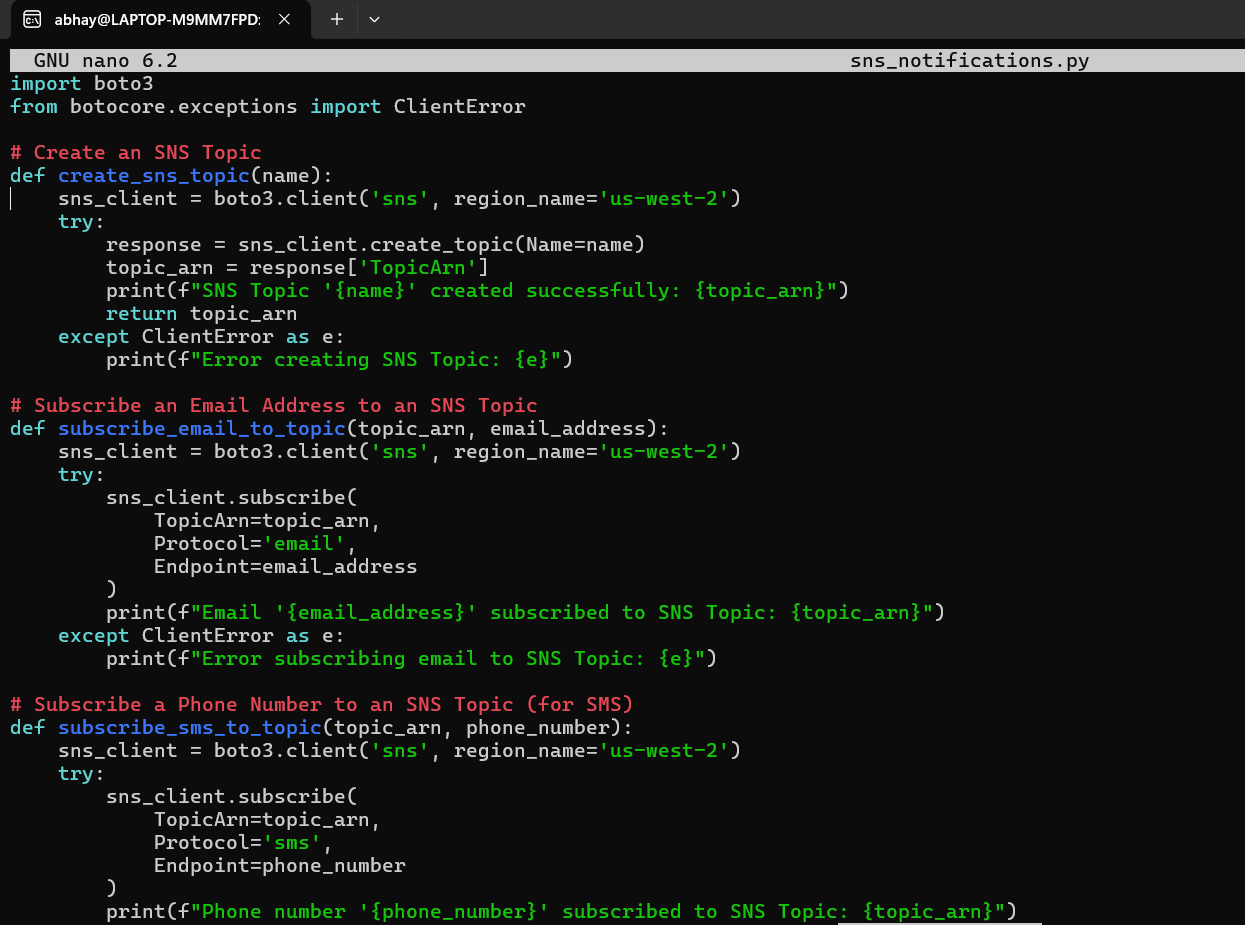
1. Setting Up SNS Notifications:

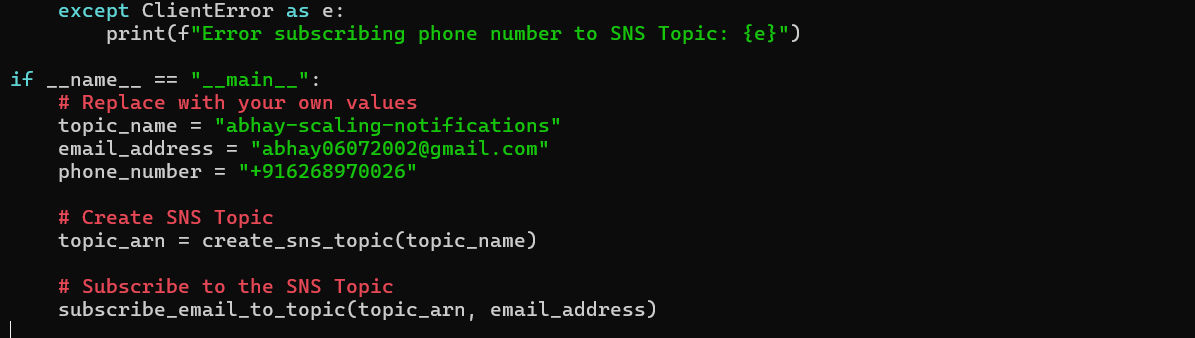
With SNS (Simple Notification Service), you can send email or SMS alerts to administrators whenever a significant event occurs, such as an instance launching, terminating, or if a health check fails.

Create a New Python File: “nano sns\_notifications.py”

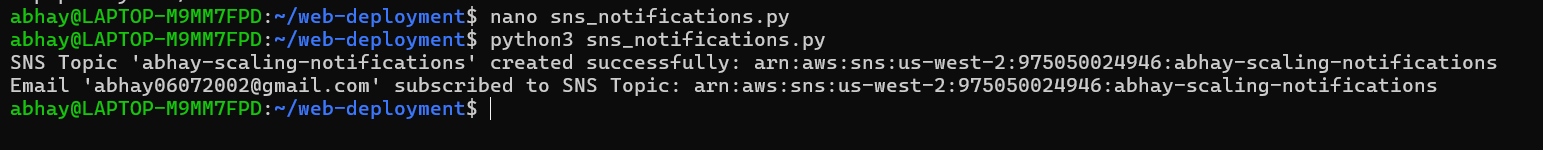


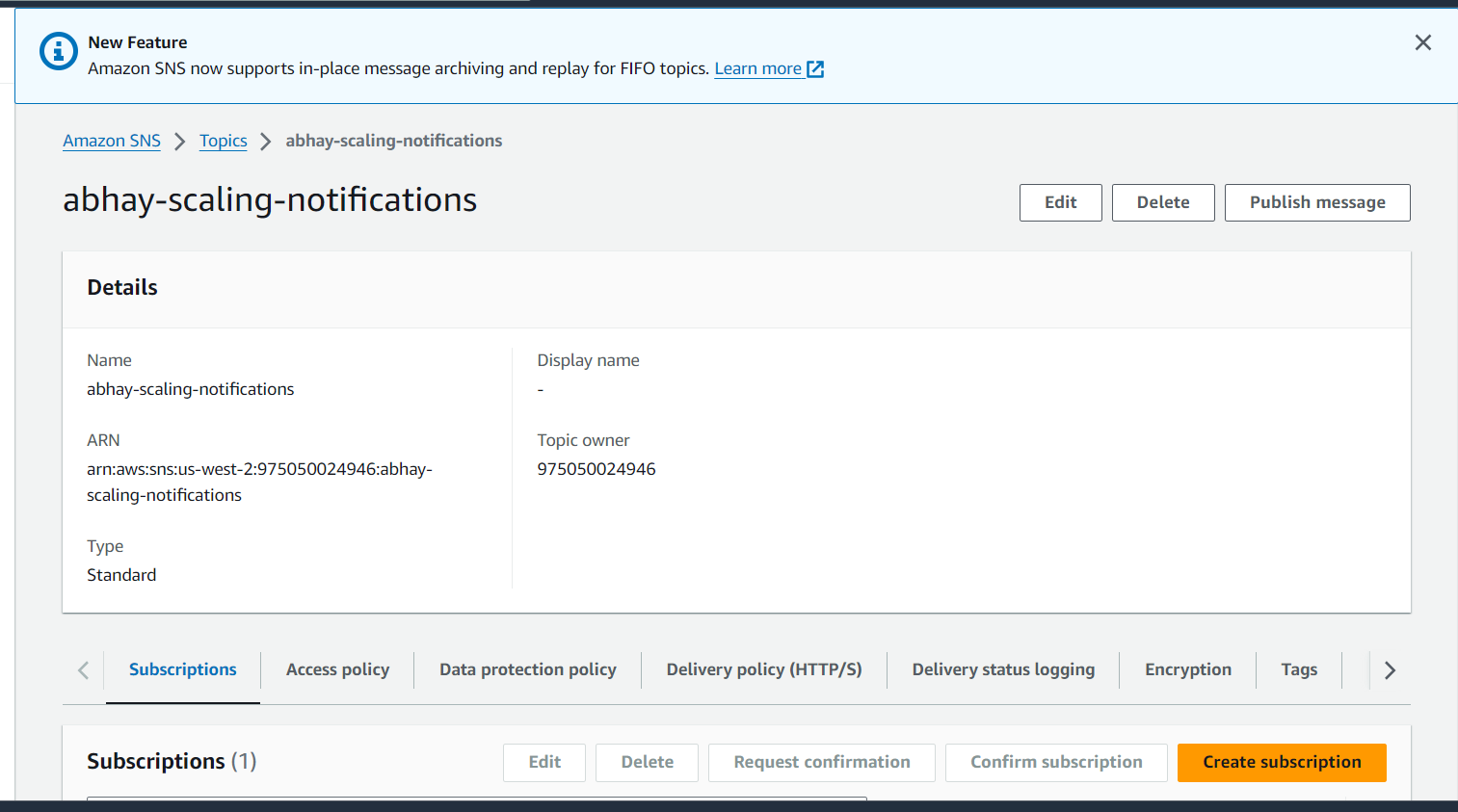
Write the Script:





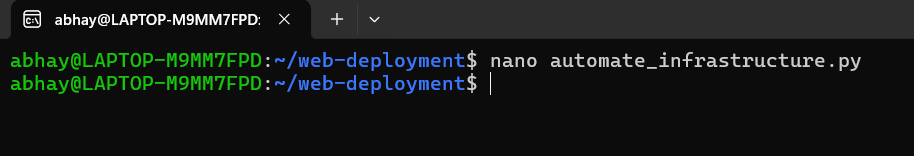
Run the Script: “python3 sns\_notifications.py”



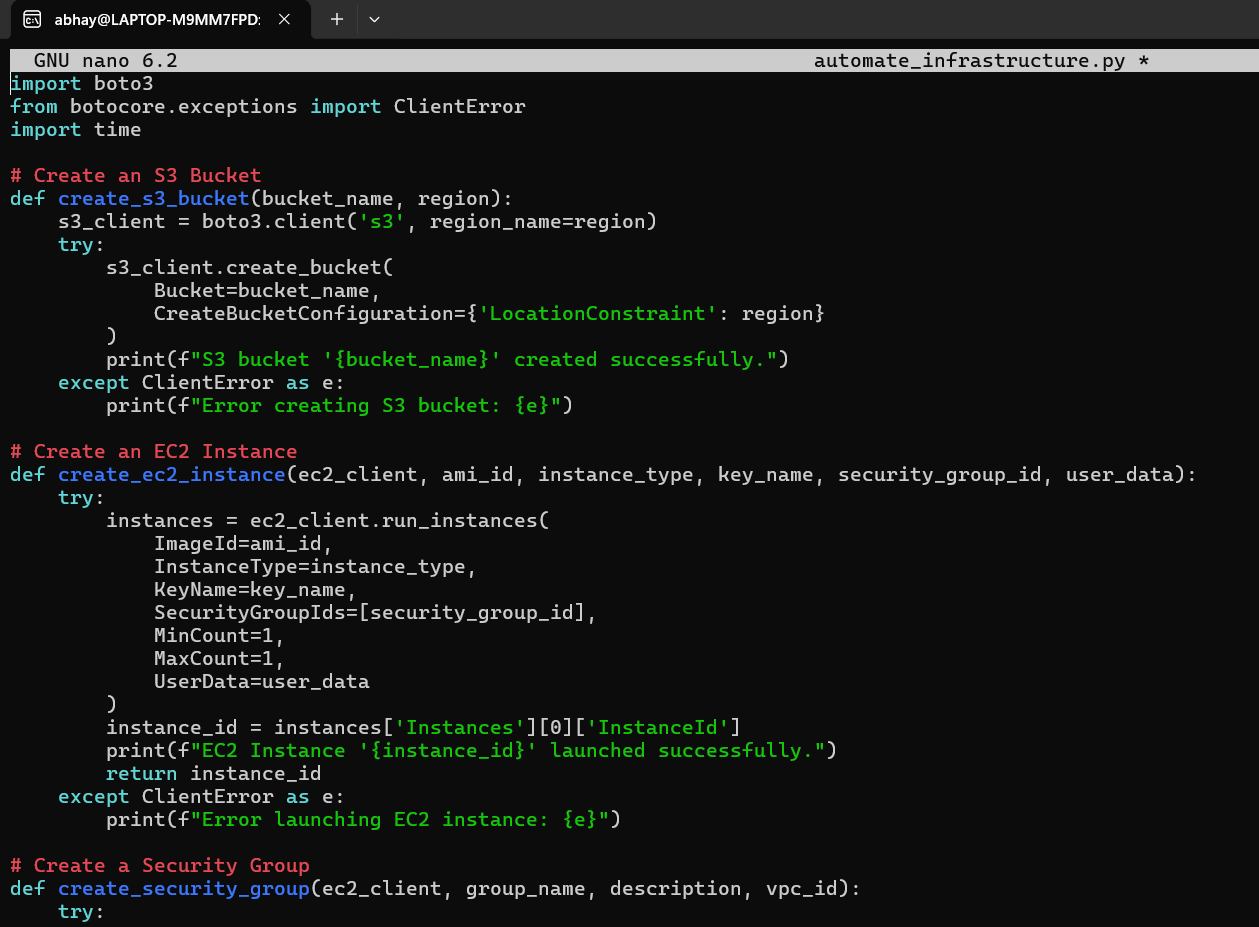


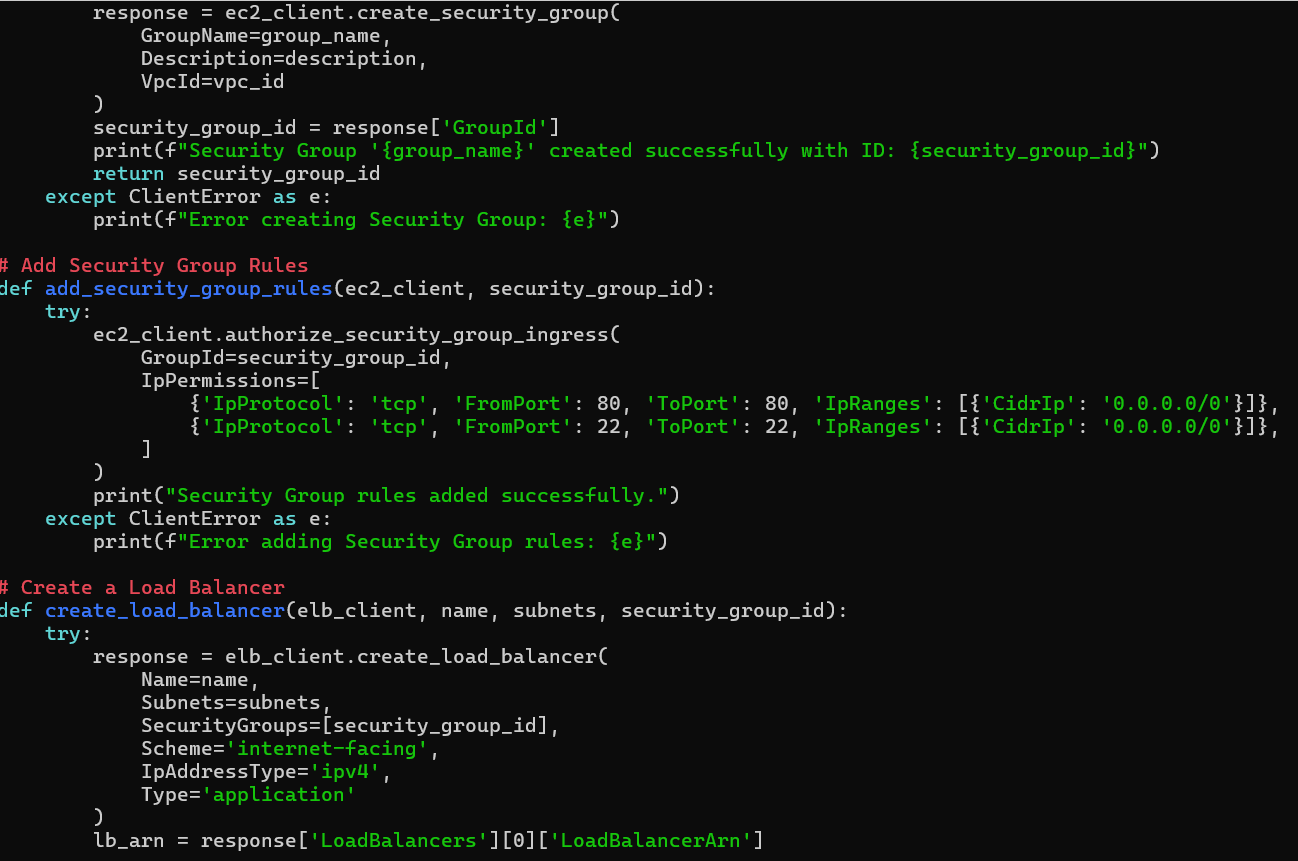
1. Automating the Entire Infrastructure with a Single Script:

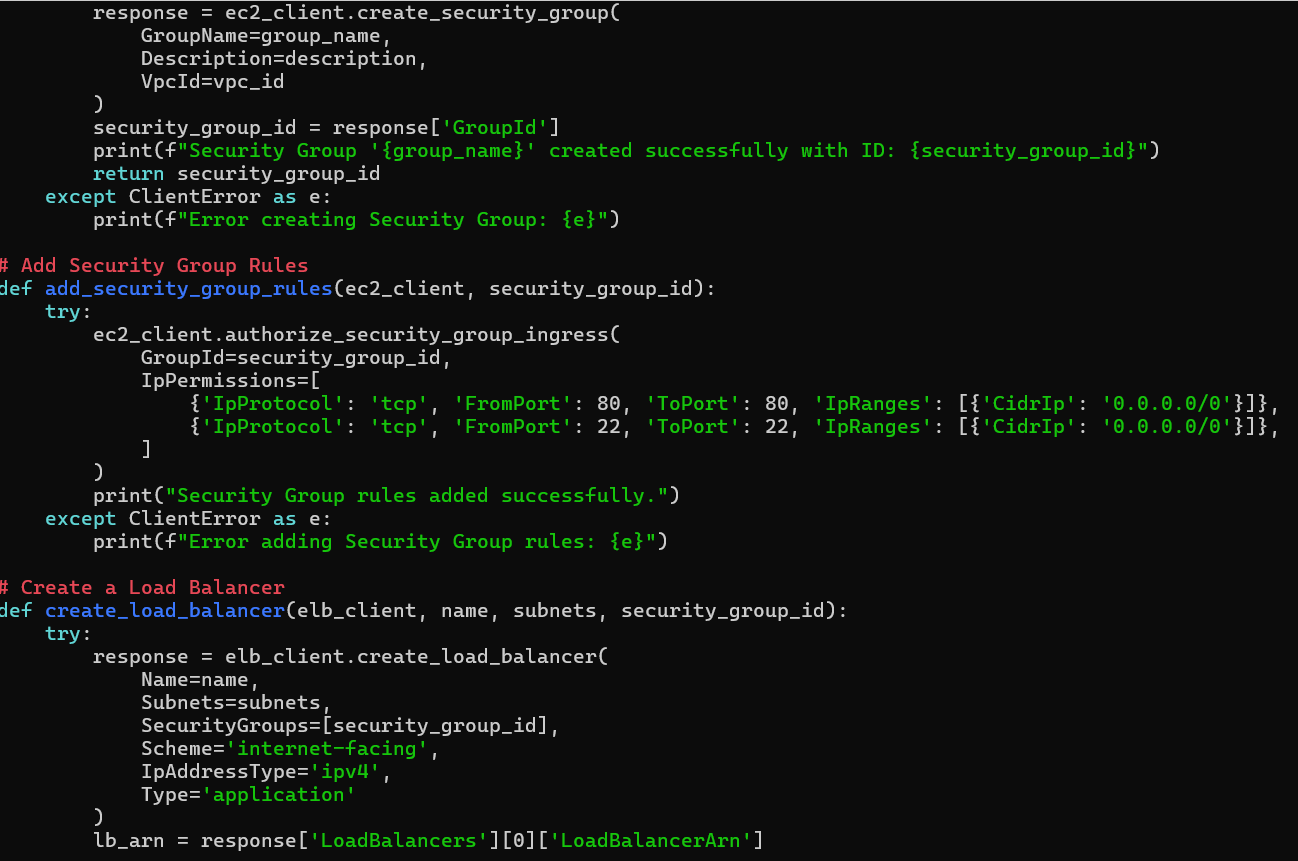
Create a New Python File: “nano automate\_infrastructure.py”

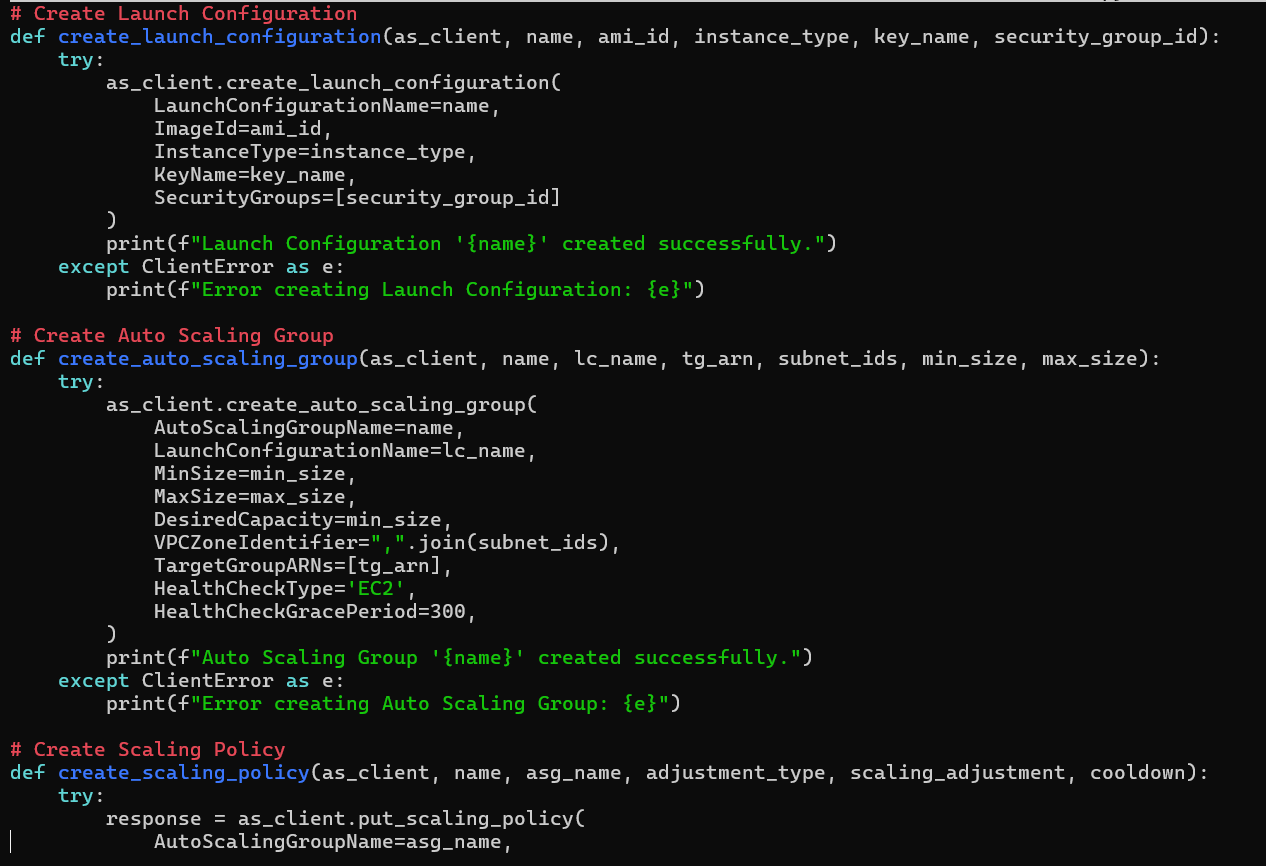


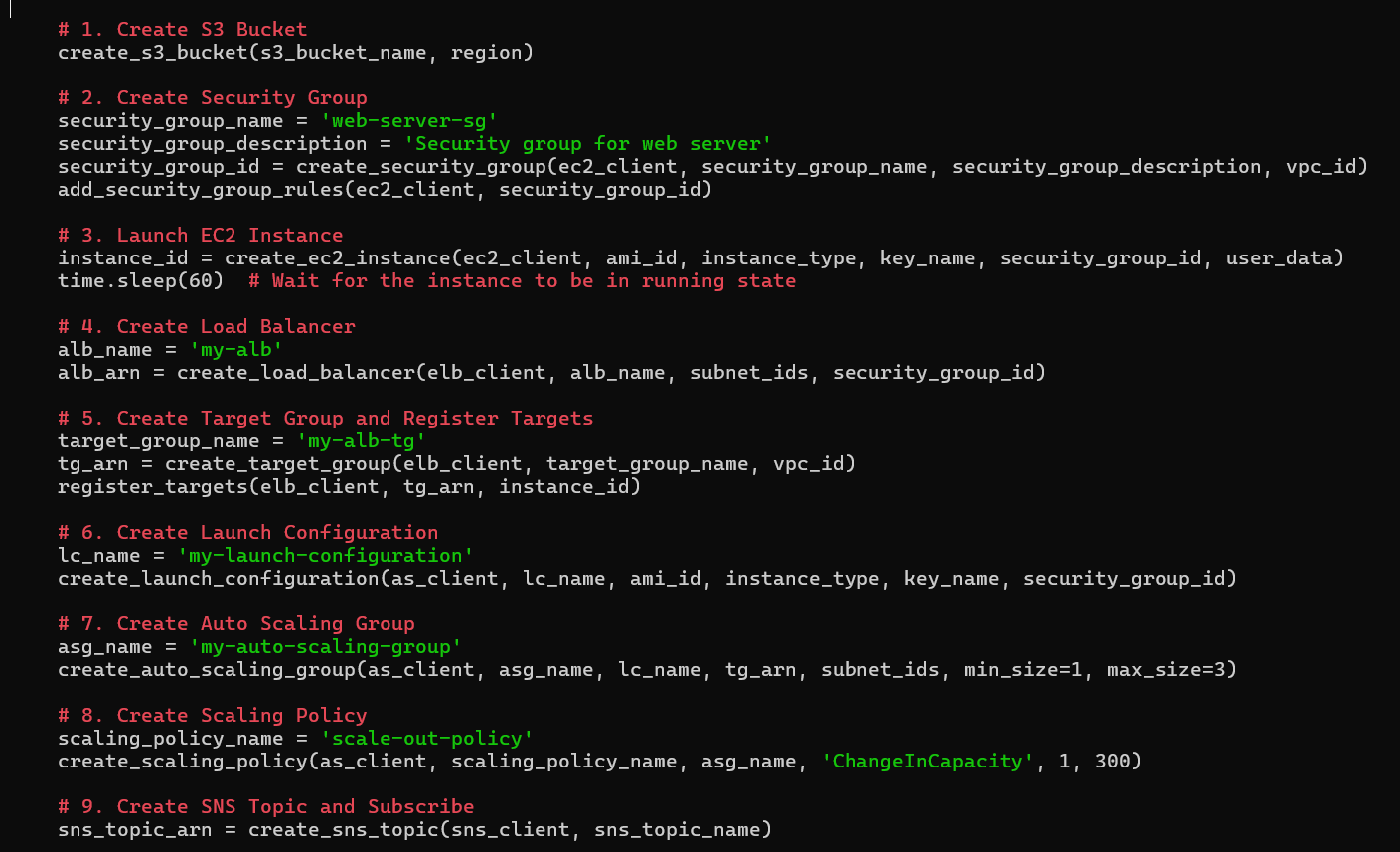
Write the Script:



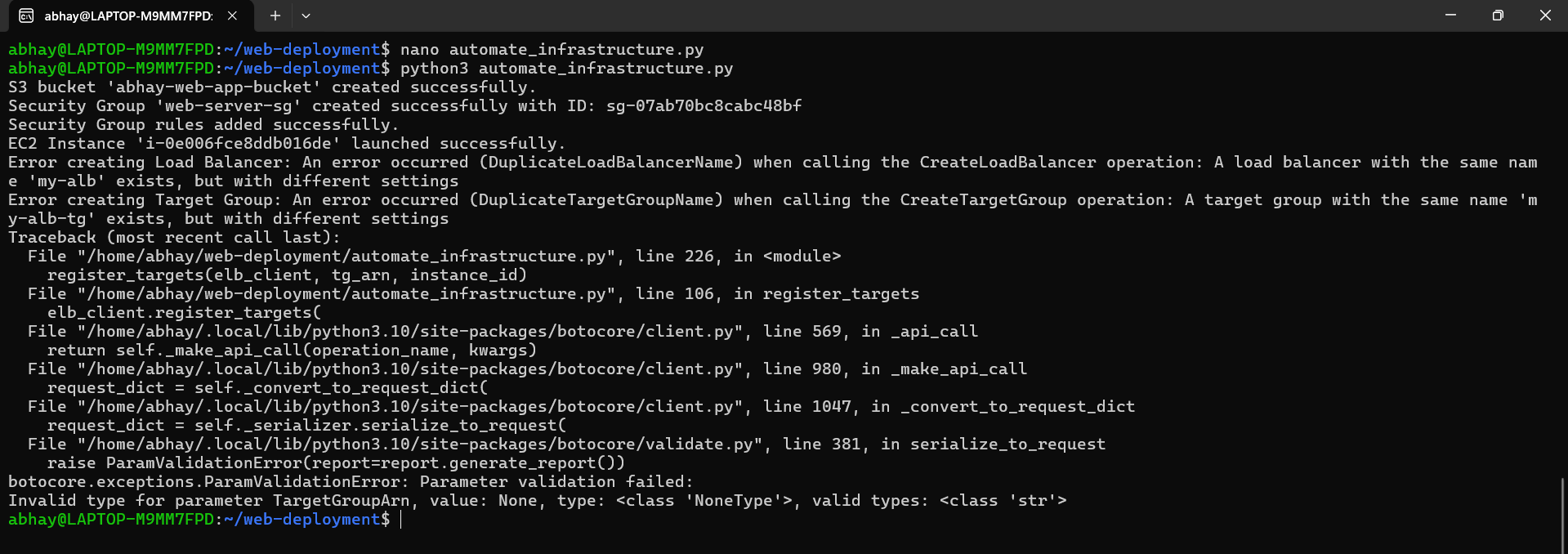








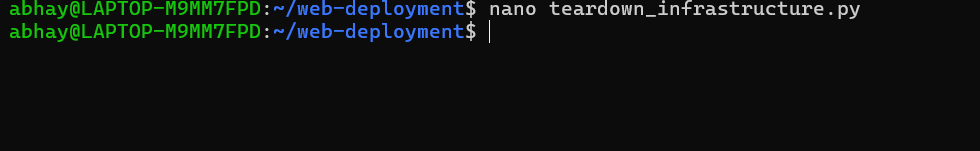
Run the Script: “python3 automate\_infrastructure.py”

****

1. Automating Teardown of the Infrastructure:

Create a New Python Script for Teardown:

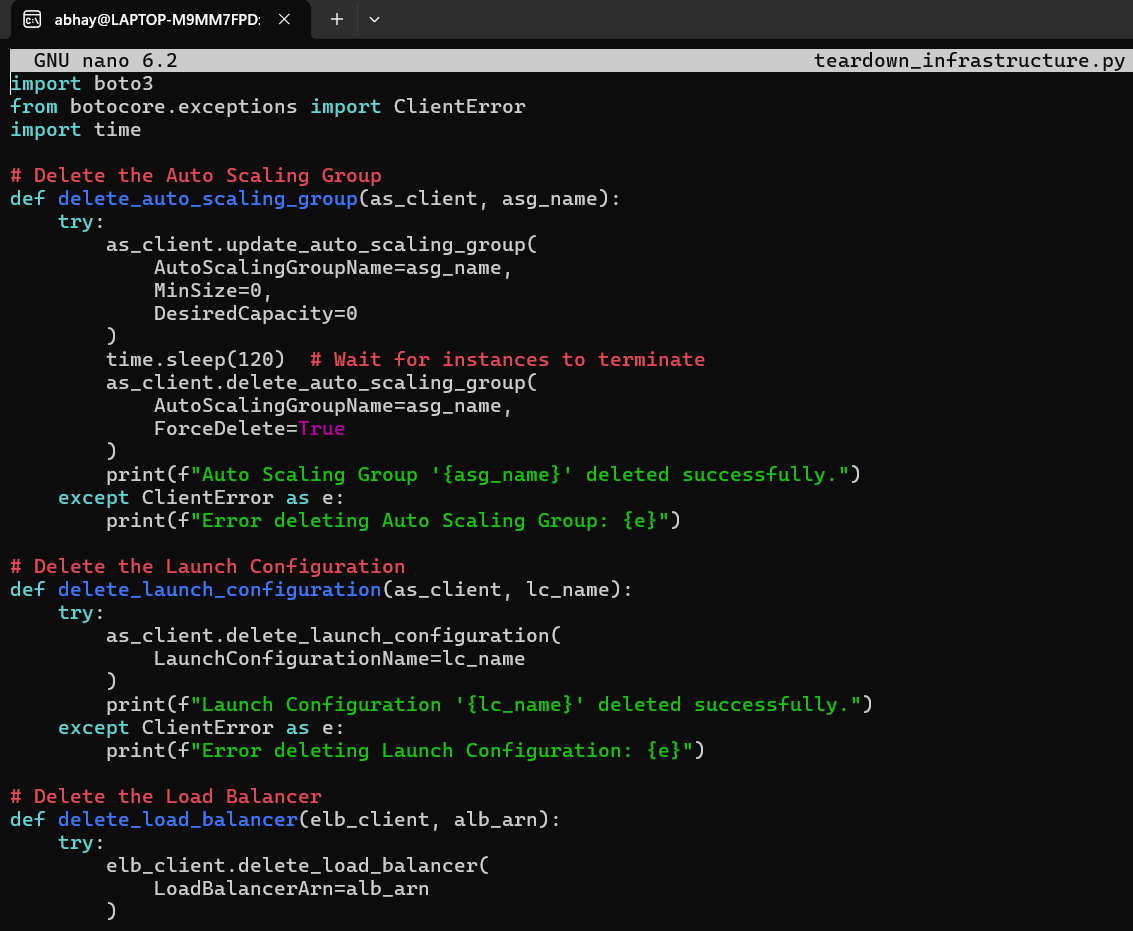
“nano teardown\_infrastructure.py”

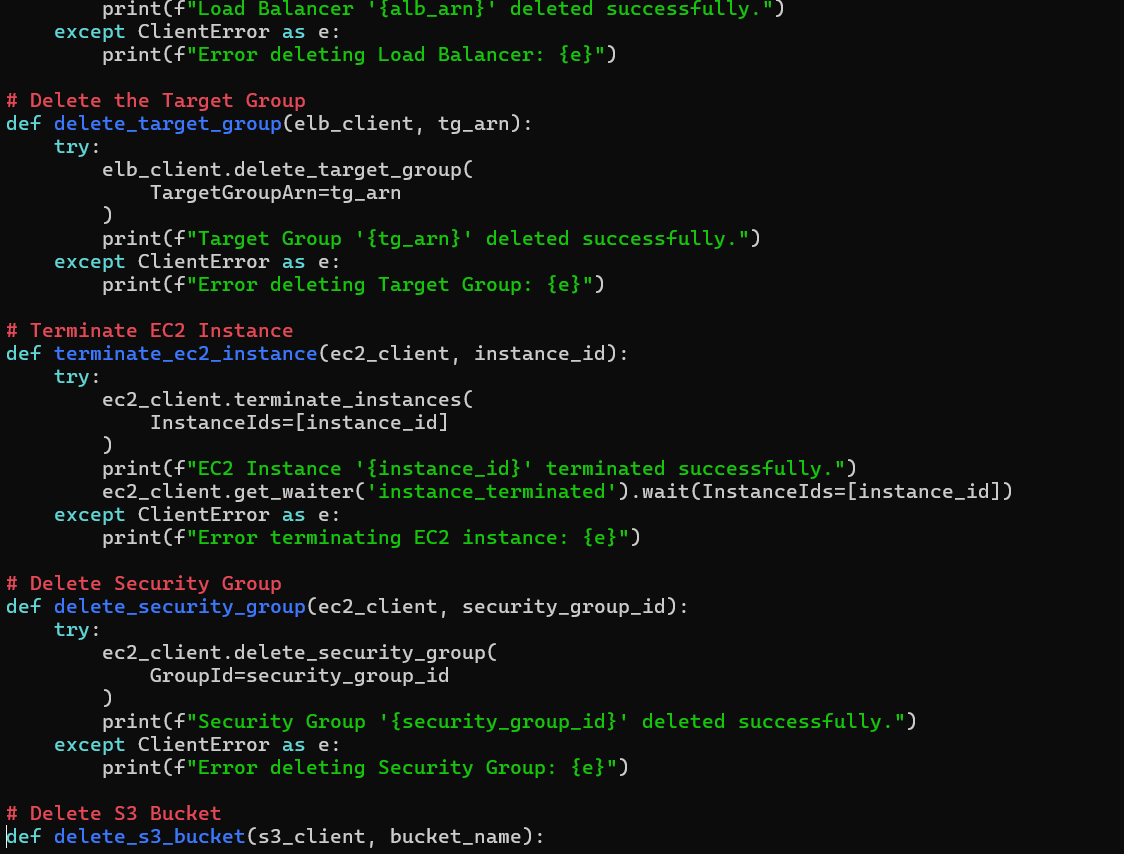


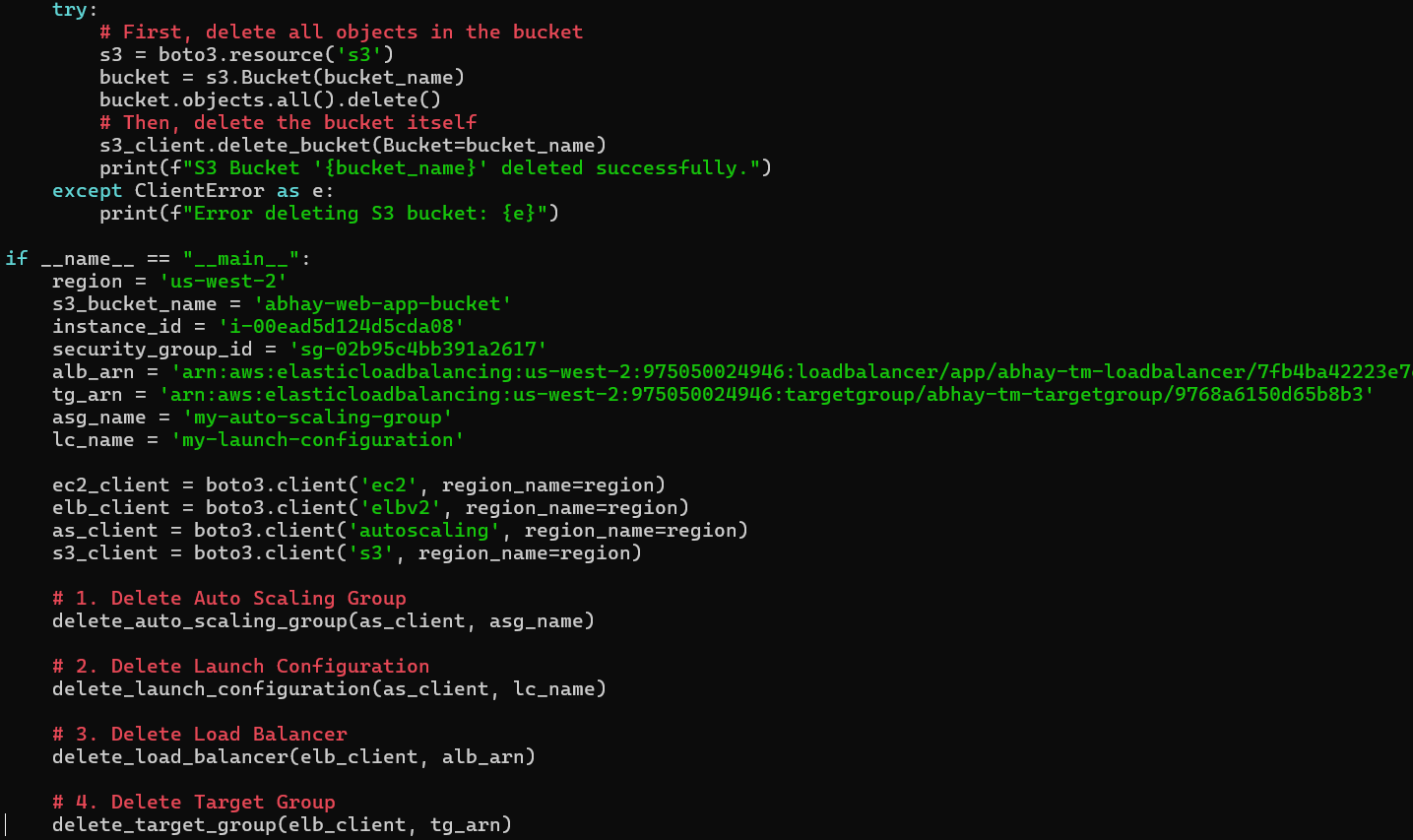
Write the Script:

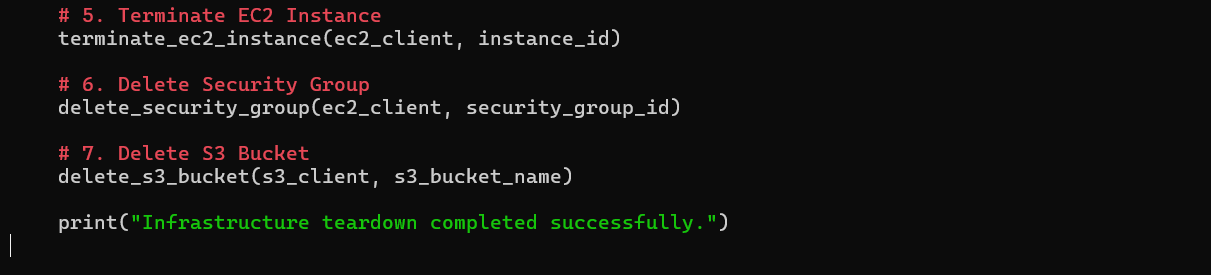
This script will:

* Delete the Auto Scaling Group (ASG).
* Delete the Launch Configuration.
* Deregister and delete the Load Balancer and Target Group.
* Terminate the EC2 instance.
* Delete the Security Group.
* Delete the S3 bucket.

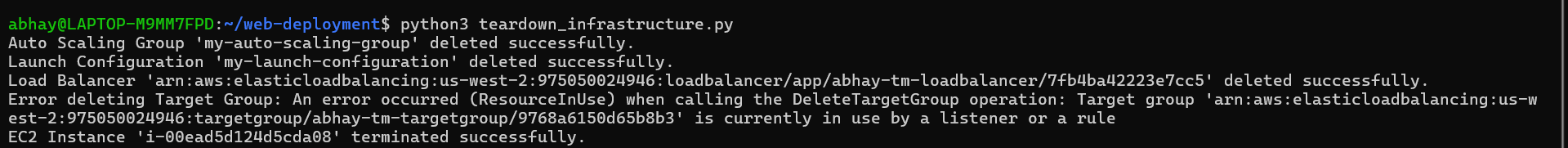


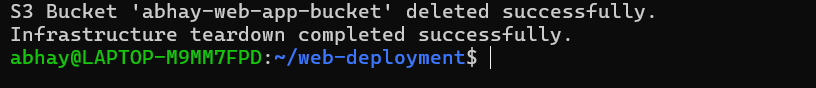






Run the Script: “python3 teardown\_infrastructure.py”





**Conclusion:**

This documentation provides a guide to automating AWS infrastructure using boto3. Each section is accompanied by Python snippets for deploying and managing EC2, ALB, Auto Scaling, SNS, and more.