Fatima Jinnah Women University

*Department of Software Engineering*



*---------------------------------------------------------------------------------------------------------------------------------------*

ASSIGNMENT#02

***Report***

**SUBJECT: CLOUD COMPUTING**

**SUBMITTED TO: SIR WAQAS SALEEM**

**SUBMITTED BY: UMBER QASIM**

**SUBMITTED DATE: 31 DECEMBER 2025**

**REGISTRATION NO: 2023-BSE-066**

**CLASS: BSSE V-B**

**Contents**

[Executive Summary 3](#_Toc218103356)

[Key Achievements: 3](#_Toc218103357)

[Architecture Design 3](#_Toc218103358)

[Architecture Overview 3](#_Toc218103359)

[Component Descriptions 4](#_Toc218103360)

* 1. [**Nginx Server 4**](#_Toc218103361)
  2. [**Backend Web Servers (Apache) 4**](#_Toc218103362)
  3. [**Networking Components 4**](#_Toc218103363)

[Network Topology 5](#_Toc218103364)

[Security Design 5](#_Toc218103365)

[Implementation Details 5](#_Toc218103366)

[Part 1: Infrastructure Setup 5](#_Toc218103367)

[Part 2: Webserver Module 10](#_Toc218103368)

[Part 3: Server Configuration Scripts 12](#_Toc218103369)

[Part 4: Infrastructure Deployment 14](#_Toc218103370)

[Part 5: Nginx Configuration & Testing 17](#_Toc218103371)

[Part 6: Documentation & Cleanup 29](#_Toc218103372)

[Challenges & Solutions 35](#_Toc218103373)

[Challenge 1: SSH Key Errors 35](#_Toc218103374)

[Challenge 2: Backend Not Reachable 35](#_Toc218103375)

[Challenge 3: Rate Limiting Not Triggering 35](#_Toc218103376)

[Lessons Learned 35](#_Toc218103377)

[Conclusion 35](#_Toc218103378)

[Appendices 36](#_Toc218103379)

# **Executive Summary**

This assignment focuses on designing and deploying a secure, highly available, and scalable multi‑tier web infrastructure on Amazon Web Services (AWS) using Terraform.

The infrastructure consists of:

* A Nginx reverse proxy/load balancer
* Multiple Apache backend web servers
* A secure VPC-based network
* Automated server configuration scripts
* Caching, SSL, security headers, rate limiting, and health checks

Terraform was used to provision all cloud resources in an Infrastructure as Code (IaC) approach, ensuring repeatability and consistency.

The system was tested for load balancing, fault tolerance, security, and performance.

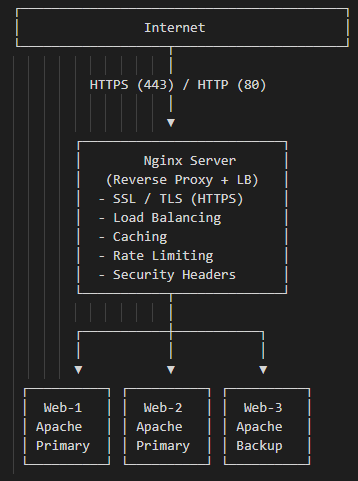
## **Key Achievements:**

* Fully automated AWS infrastructure using Terraform
* Nginx HTTPS load balancer with SSL
* Multiple Apache backend servers with dynamic metadata display
* Secure networking using security groups
* Bonus features: rate limiting, health checks, caching, and security headers

# **Architecture Design**

## **Architecture Overview**

The architecture follows a three‑tier design where traffic from the internet first reaches the Nginx load balancer, which then distributes requests to backend Apache servers.



## **Component Descriptions**

### **Nginx Server**

* Acts as a reverse proxy and load balancer
* Terminates SSL (HTTPS)
* Forwards requests to backend servers
* Implements caching and rate limiting
* Adds security headers

### **Backend Web Servers (Apache)**

* Serve dynamic HTML pages
* Display instance metadata (IP, hostname, timestamp)
* Identified as web‑1, web‑2, and web‑3
* web‑3 acts as a backup server

### **Networking Components**

* VPC
* Subnet
* Internet Gateway
* Route Table

## **Network Topology**

* All servers are deployed inside a single VPC
* Public subnet allows internet access
* Security groups restrict traffic between layers
* Backend servers only accept traffic from Nginx

## **Security Design**

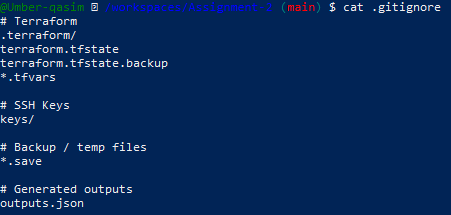
* SSH access restricted to personal IP
* Backend servers not publicly exposed
* HTTPS encryption using SSL certificates
* Security headers enabled
* Rate limiting to prevent abuse

# **Implementation Details**

## **Part 1: Infrastructure Setup**

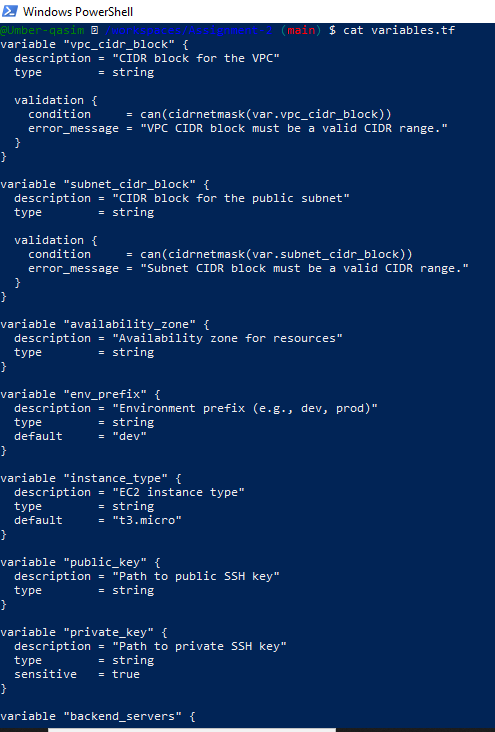
***1.1 Project Structure***



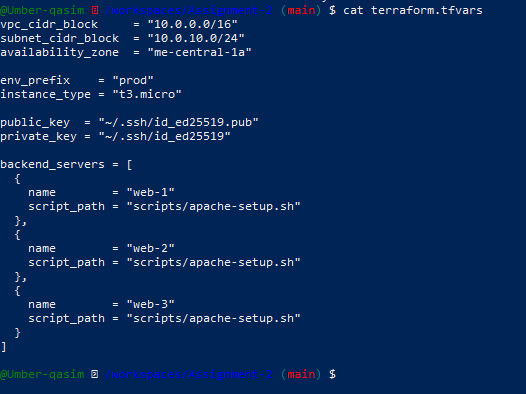


***1.2 Variable Configuration***

**File: variables.tf**

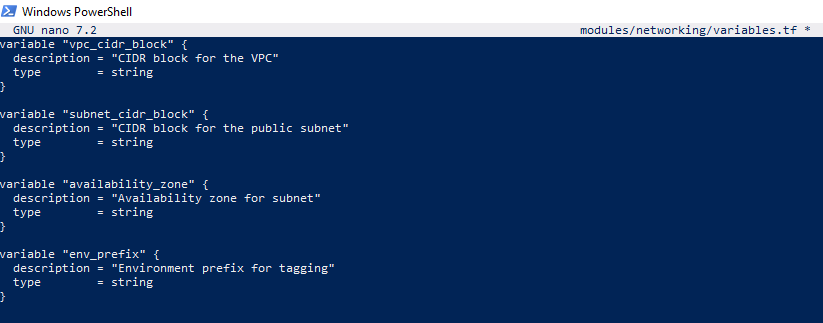


**File: terraform.tfvars**

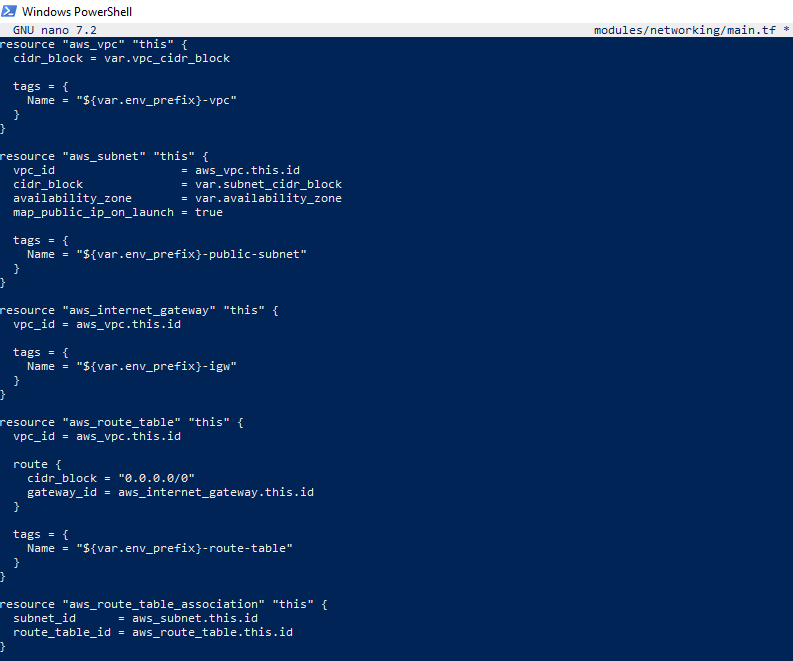


***1.3 Networking Module***

**File: modules/networking/variables.tf**



**File: modules/networking/main.tf**

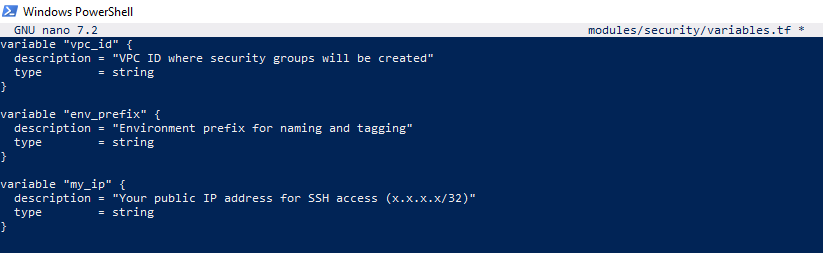


**File: modules/networking/outputs.tf**

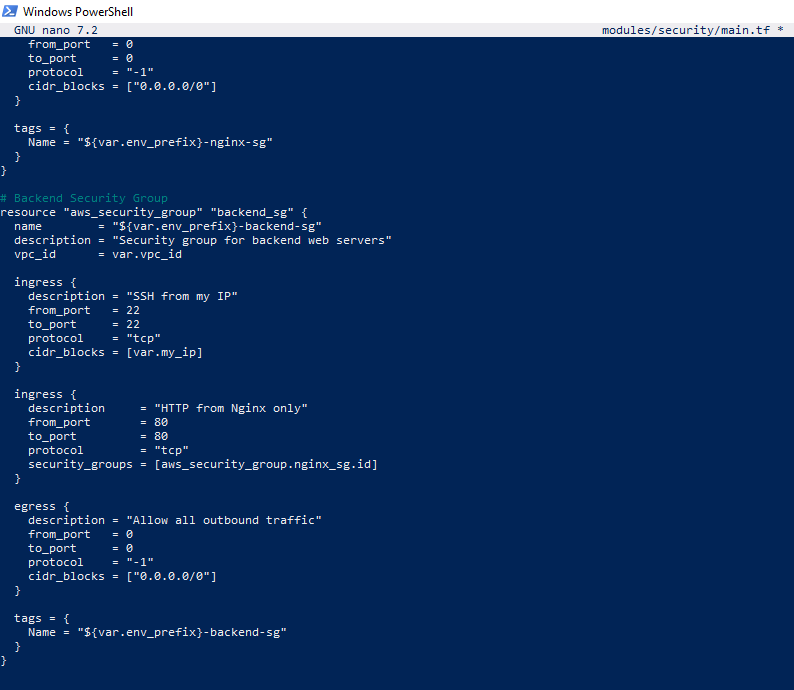


***1.4 Security Module***

**File: modules/security/variables.tf**



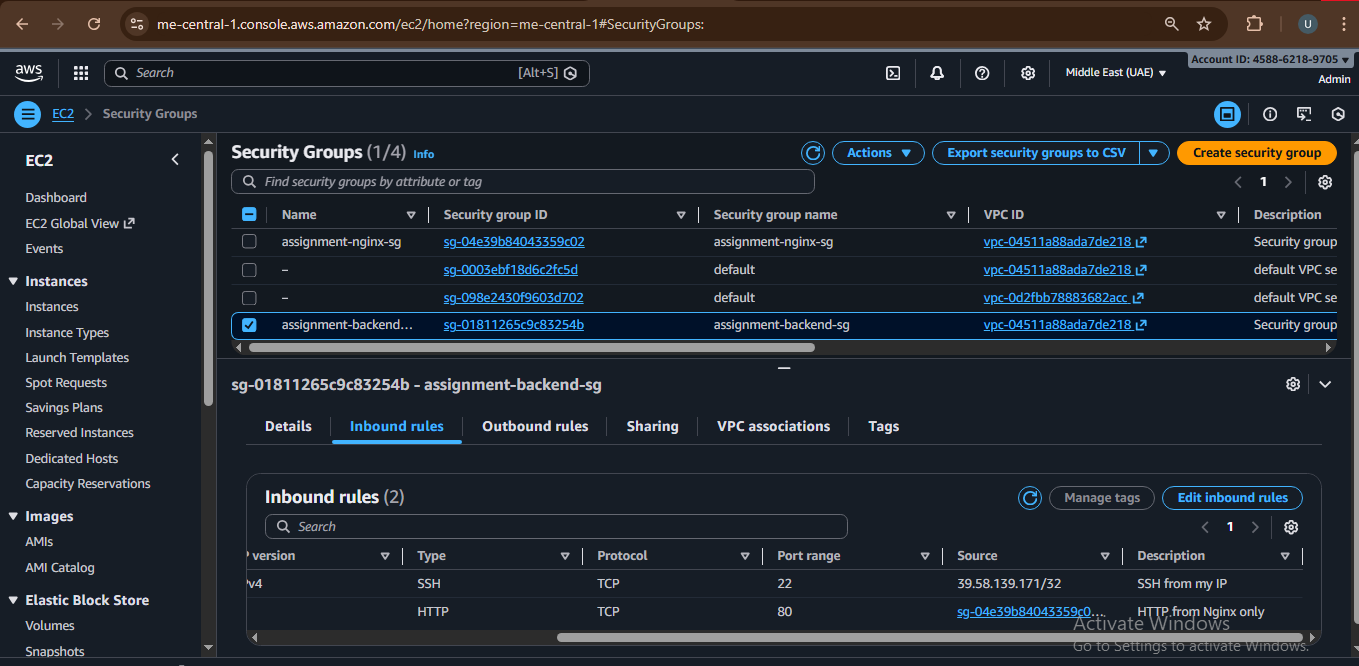
**File: modules/security/main.tf**



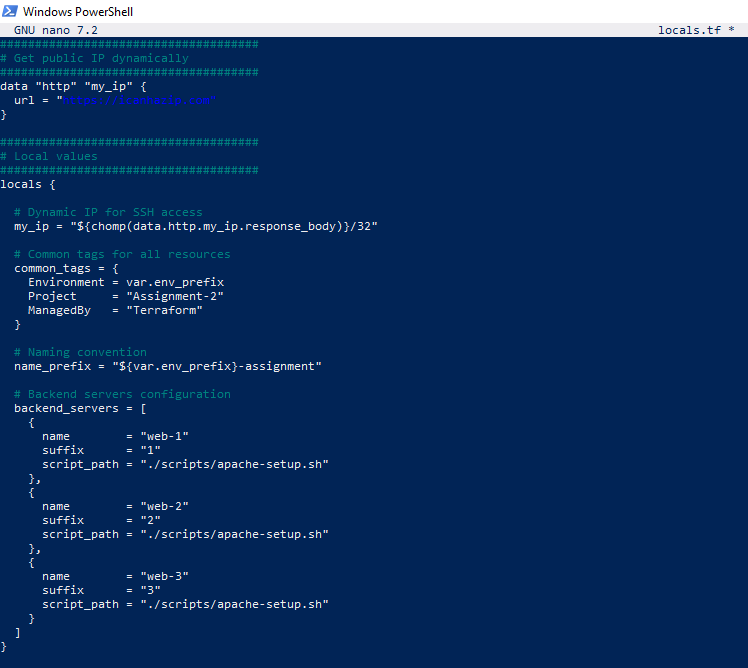
**File: modules/security/outputs.tf**



**AWS Console**



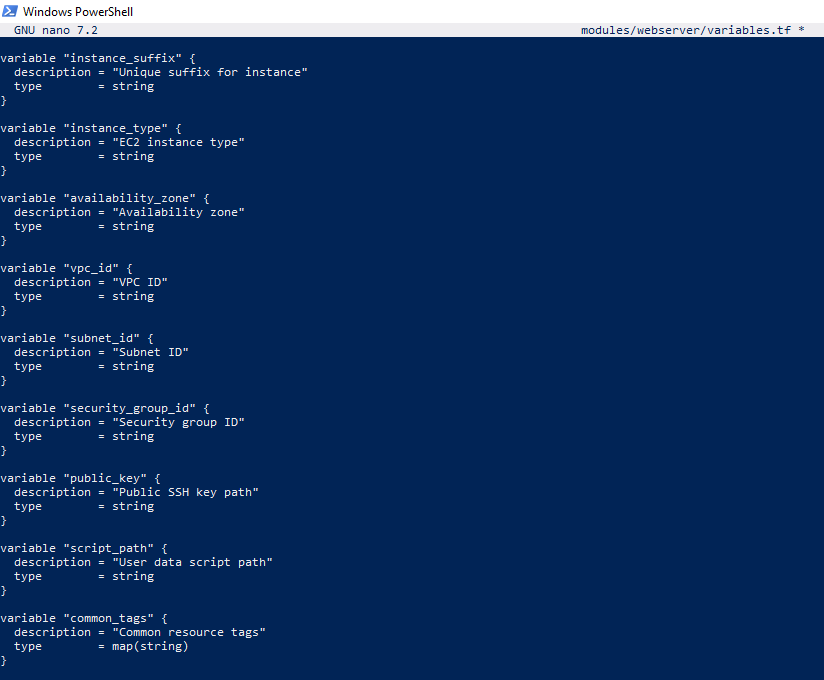
***1.5 Locals Configuration***



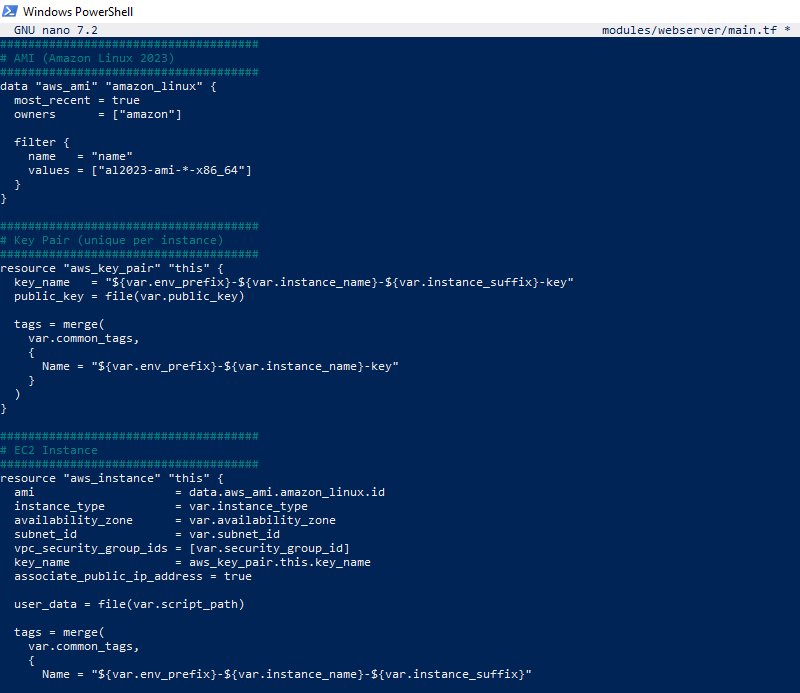
## **Part 2: Webserver Module**

***2.1 Module Design***

**File: modules/webserver/variables.tf**



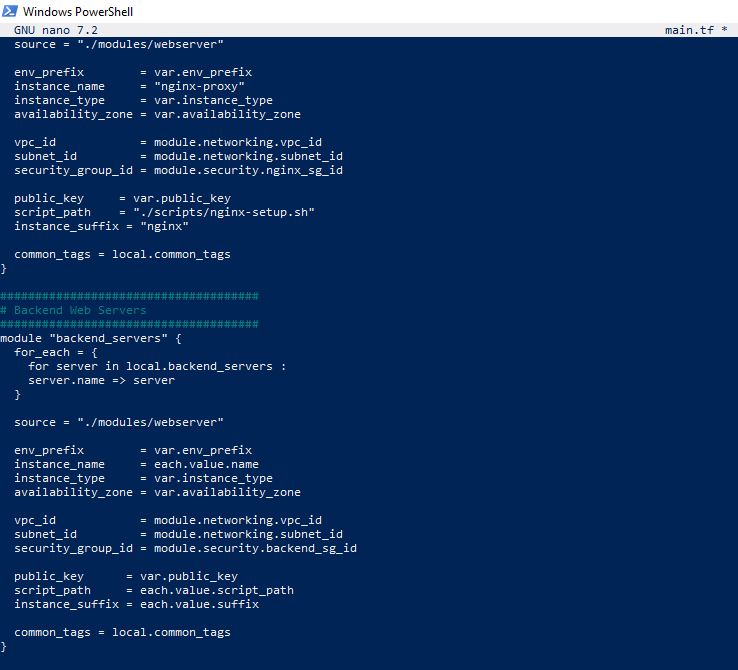
**File: modules/webserver/main.tf**



**File: modules/webserver/outputs.tf**

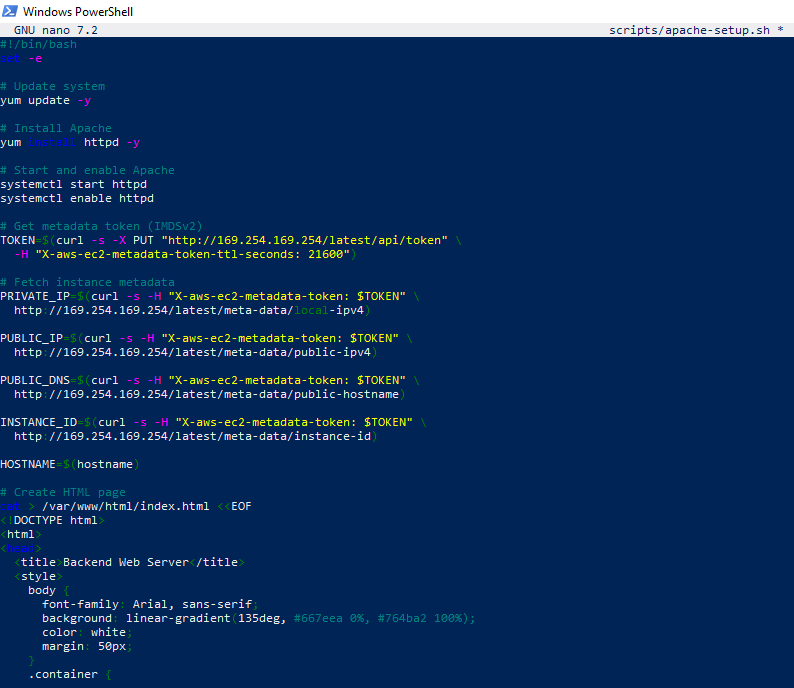


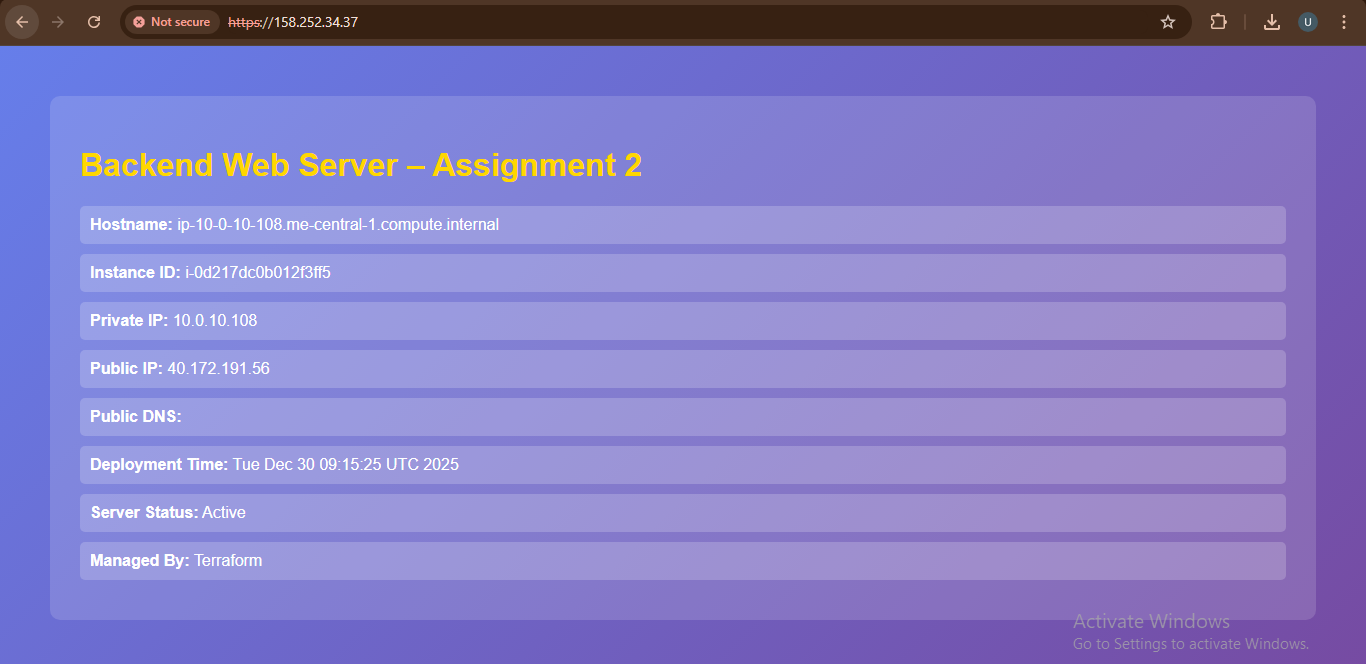
***2.2 Module Usage***

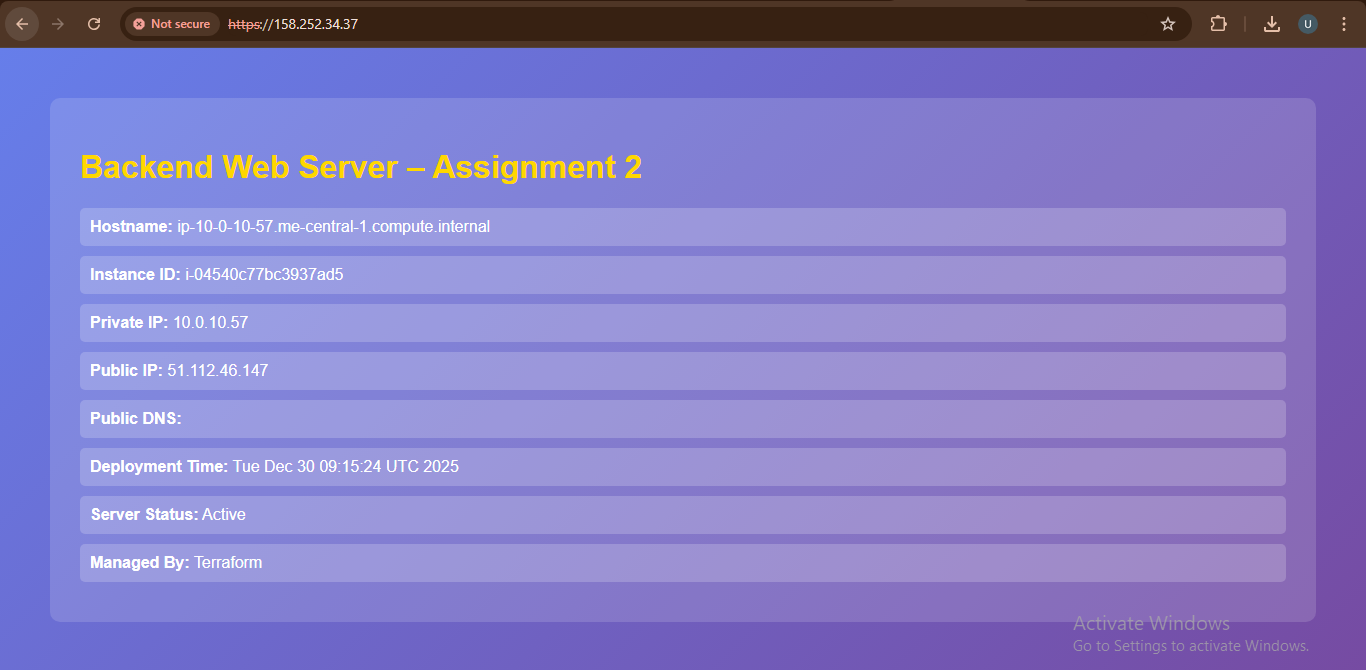


## **Part 3: Server Configuration Scripts**

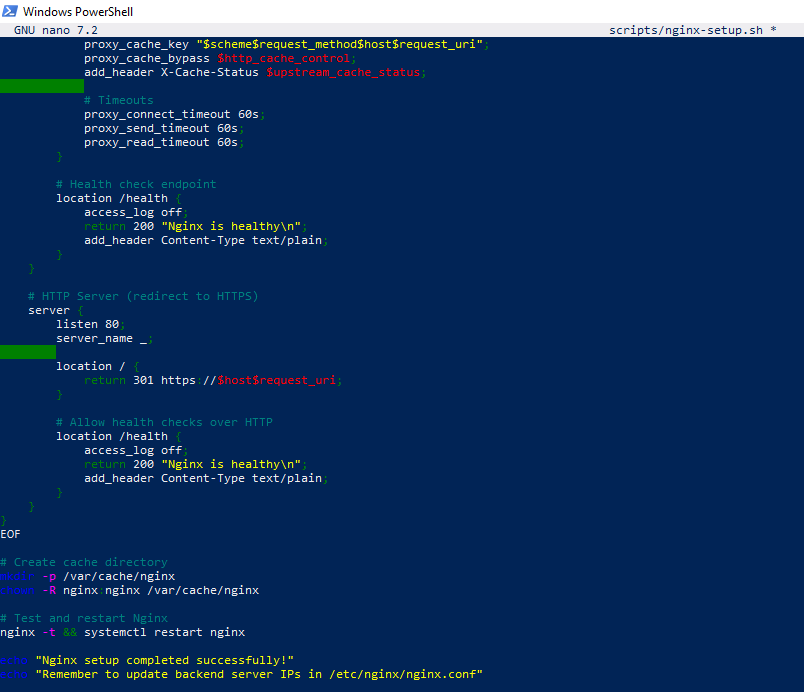
***3.1 Apache Backend Server Script***

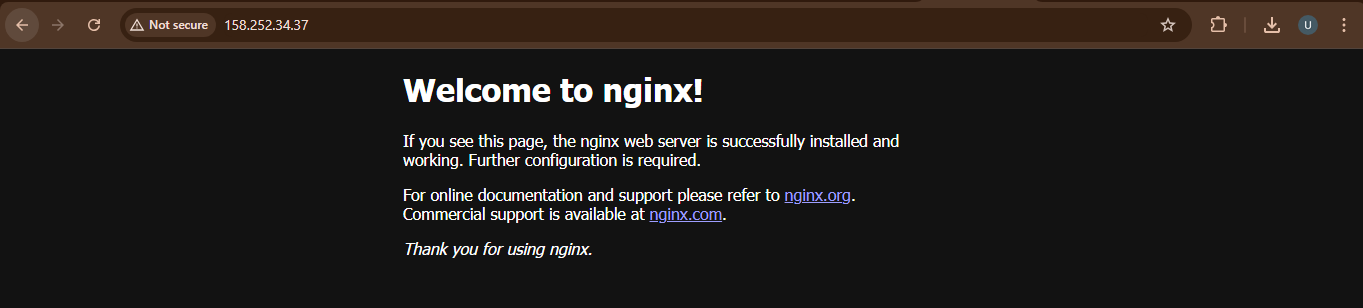






***3.2 Nginx Server Setup Script***

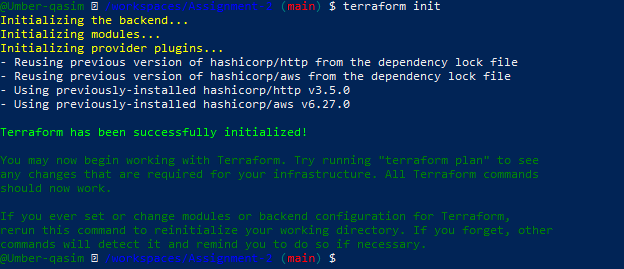




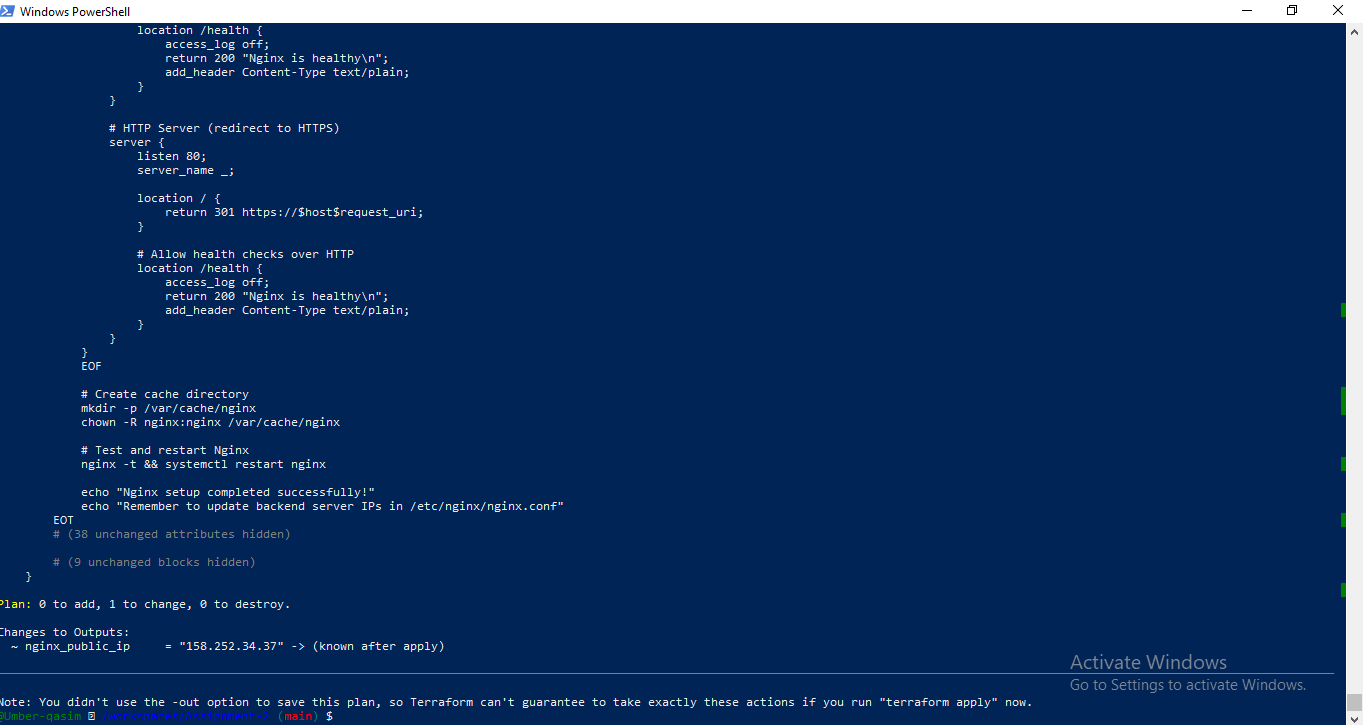
## **Part 4: Infrastructure Deployment**

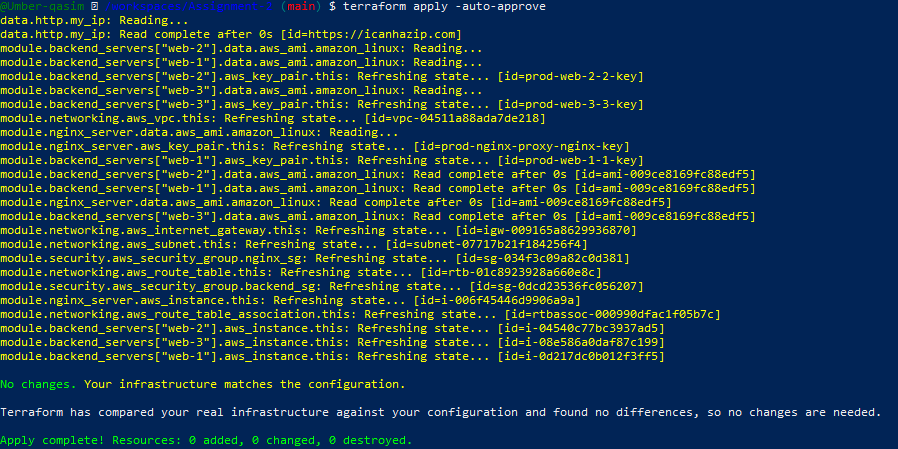
***4.1 Initial Deployment***



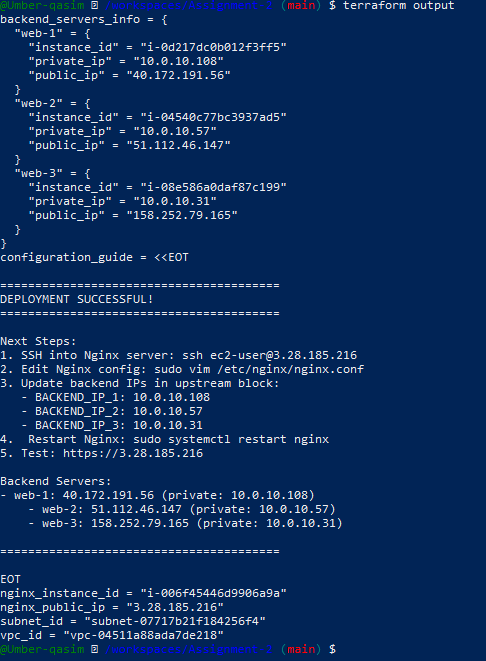






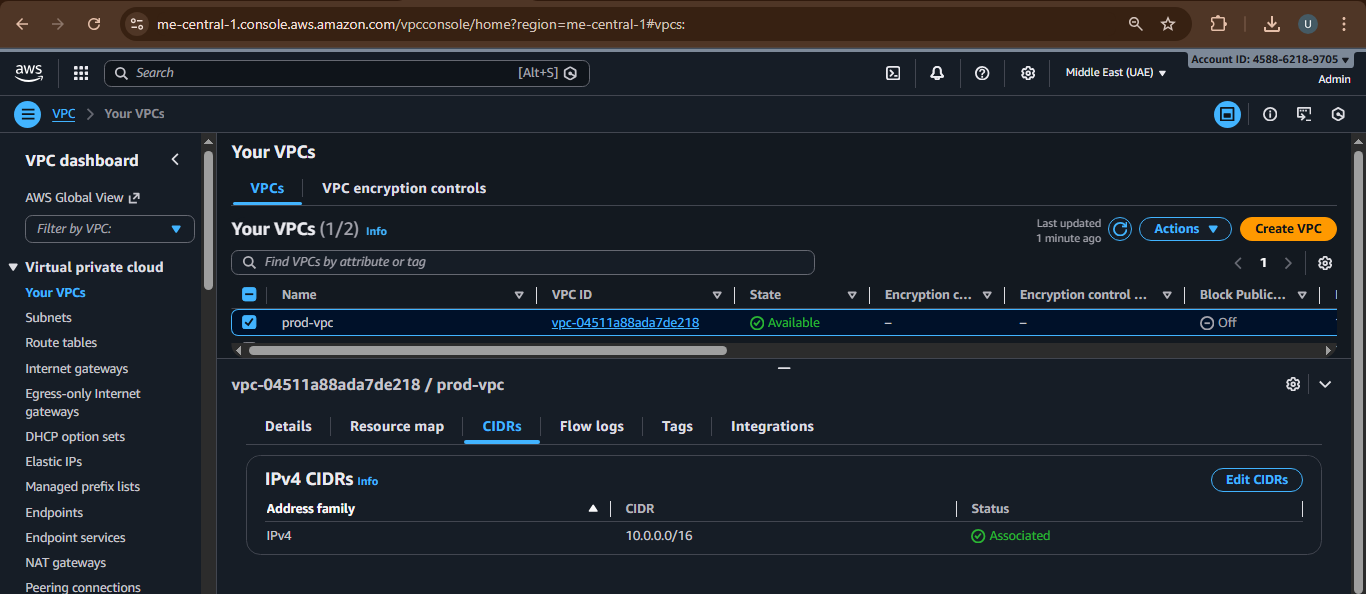


***4.2 Output Configuration***

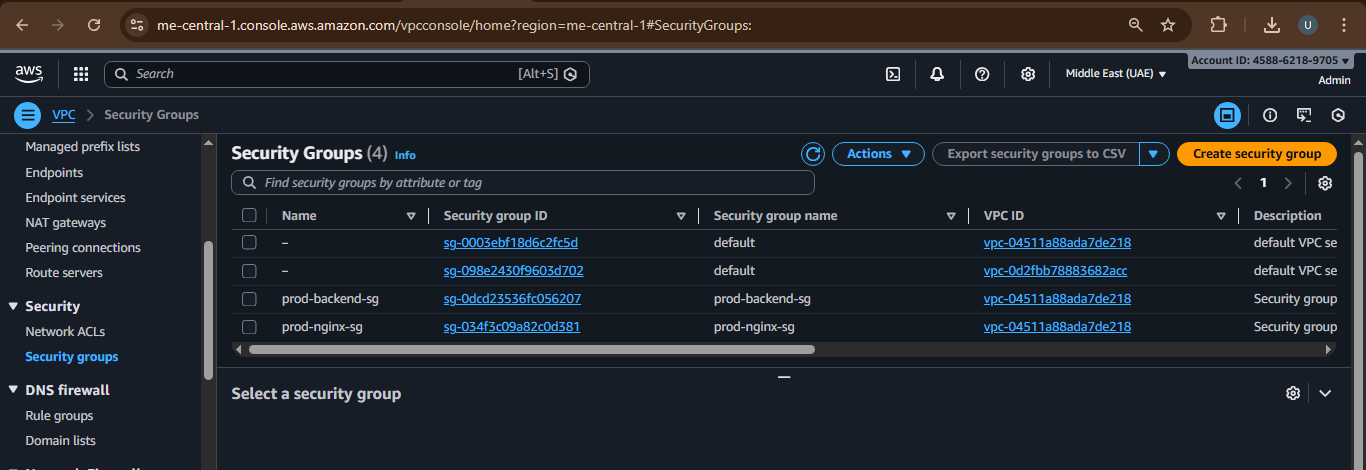


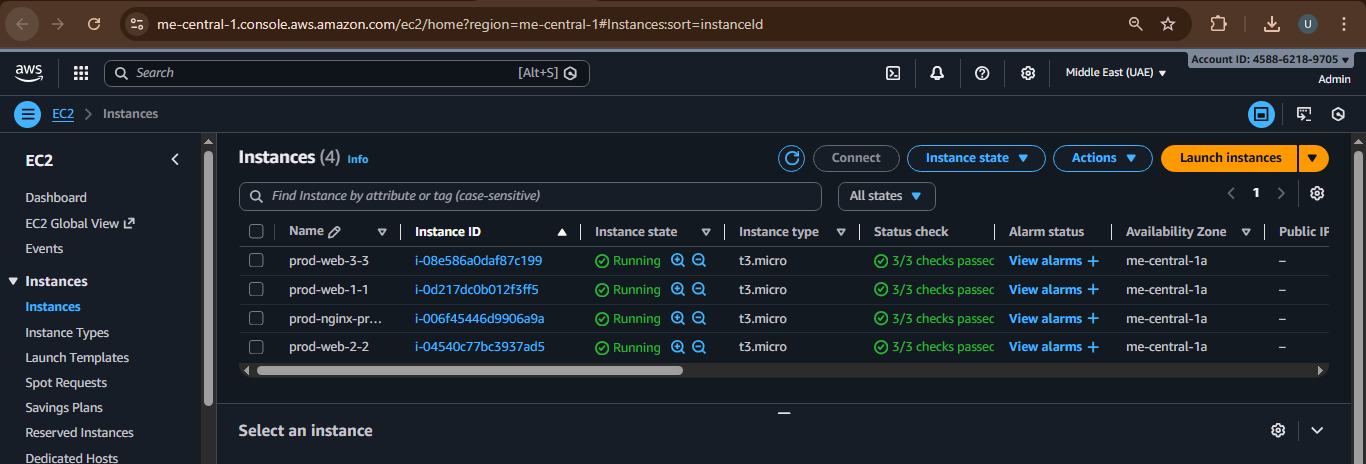


***4.3 AWS Console Verification***



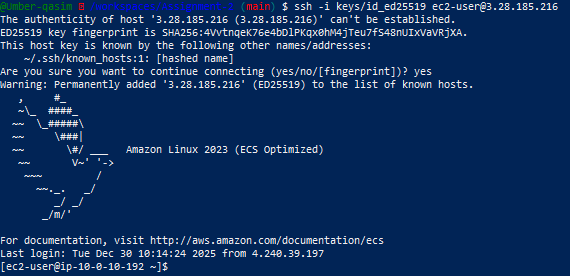


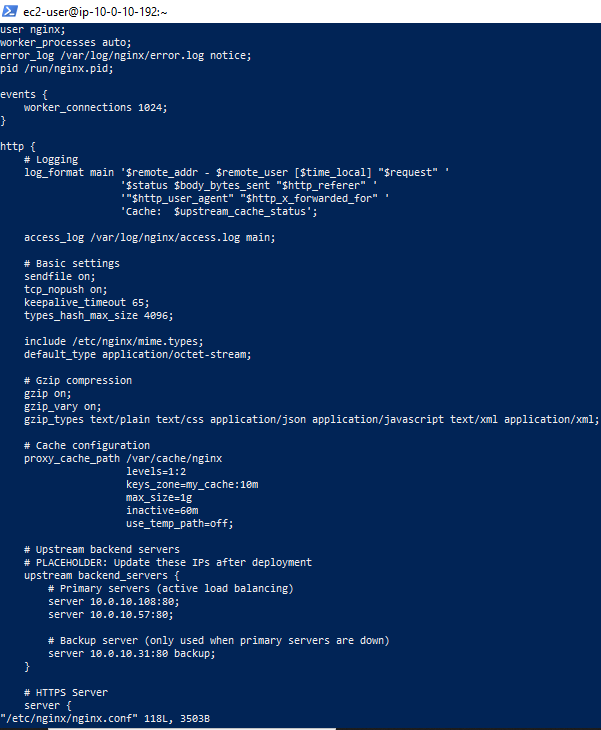


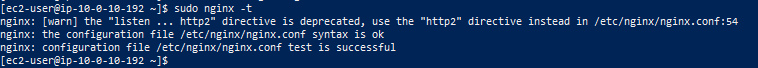


## **Part 5: Nginx Configuration & Testing**

***5.1 Update Nginx Backend Configuration***

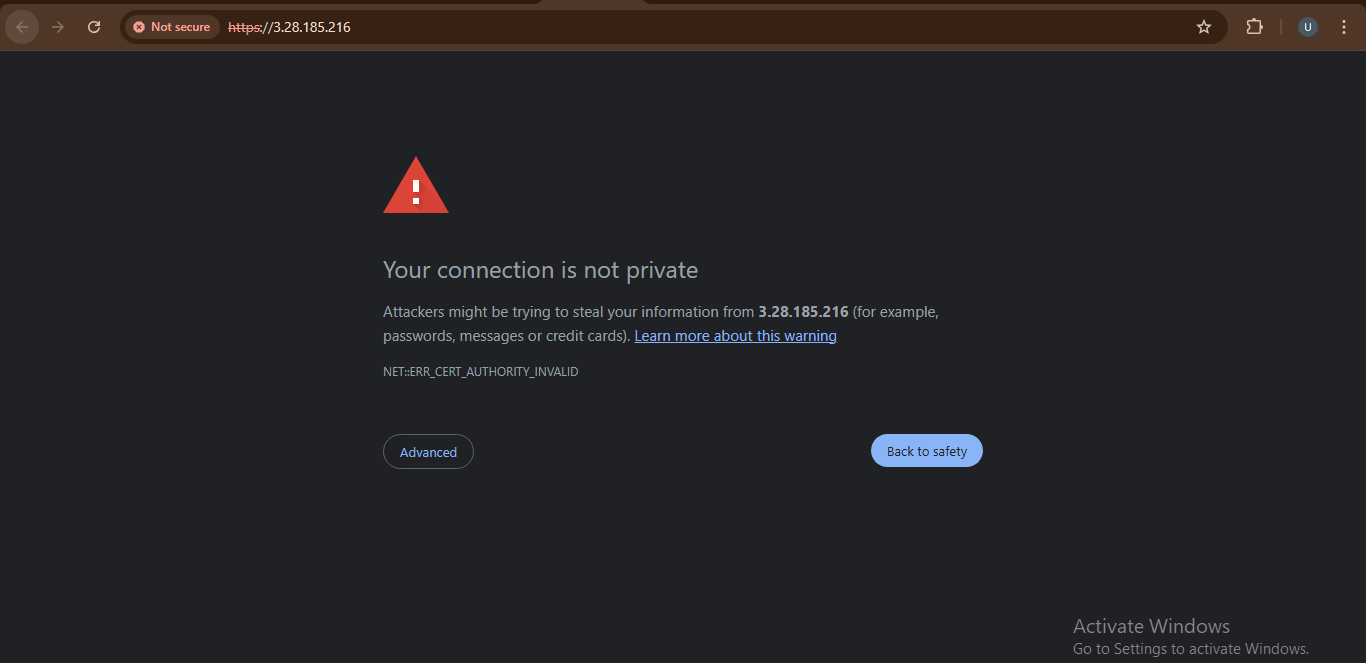


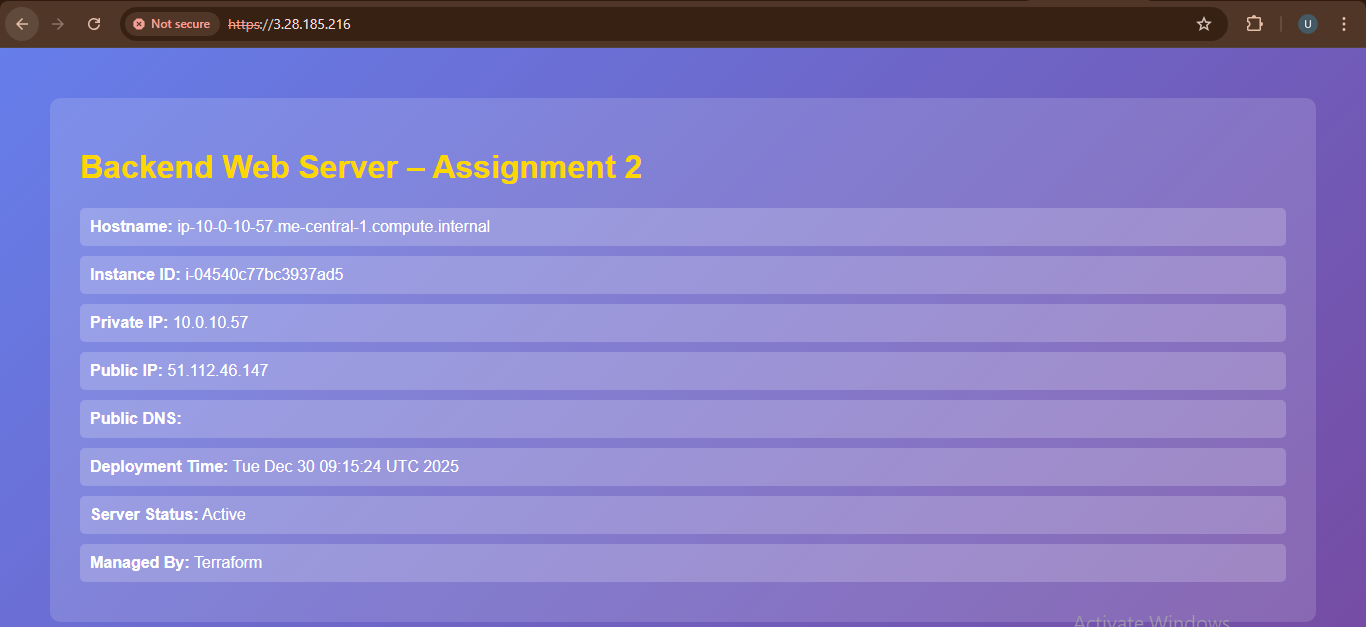


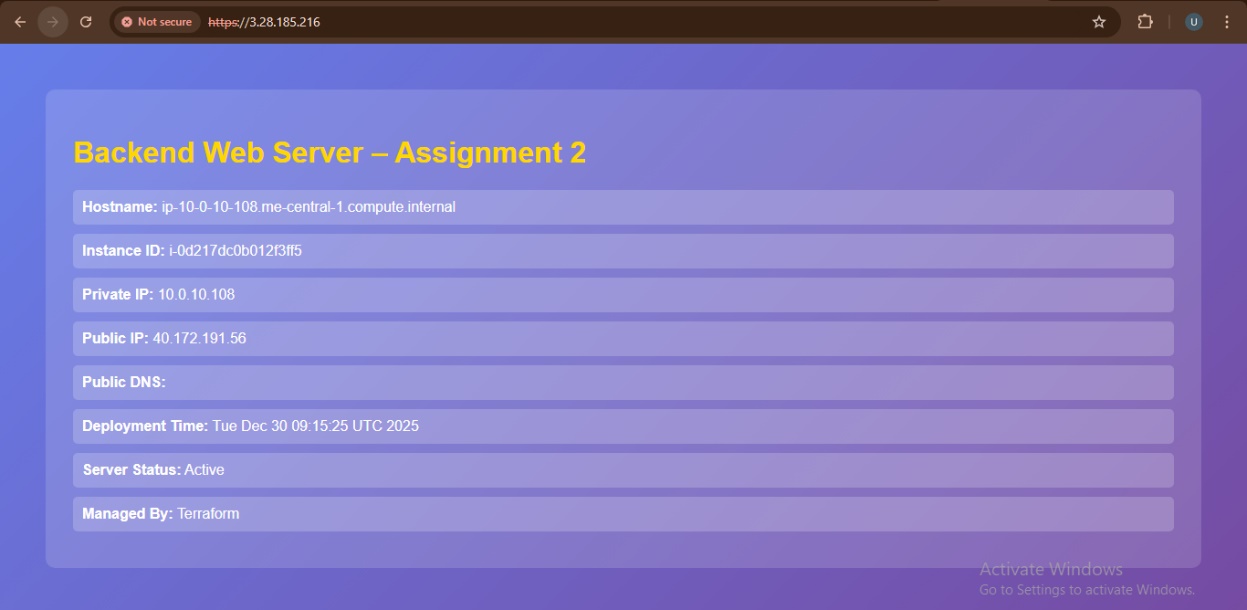


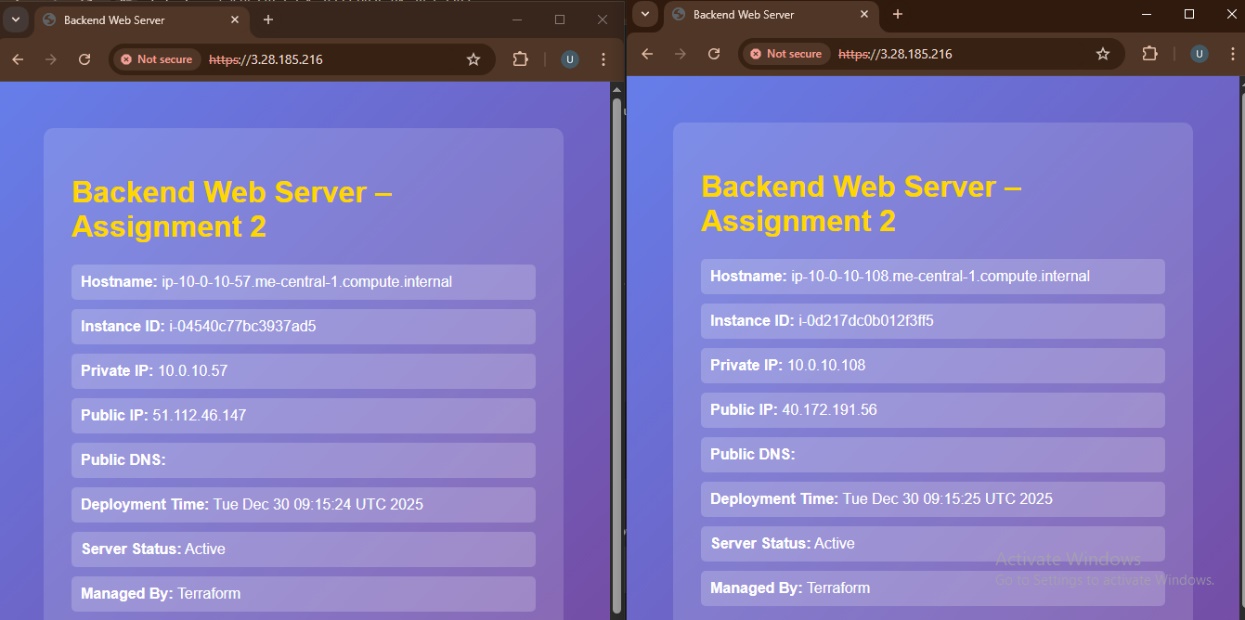


***5.2 Test Load Balancing***



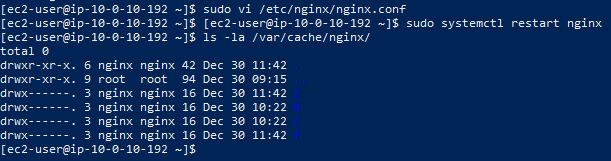


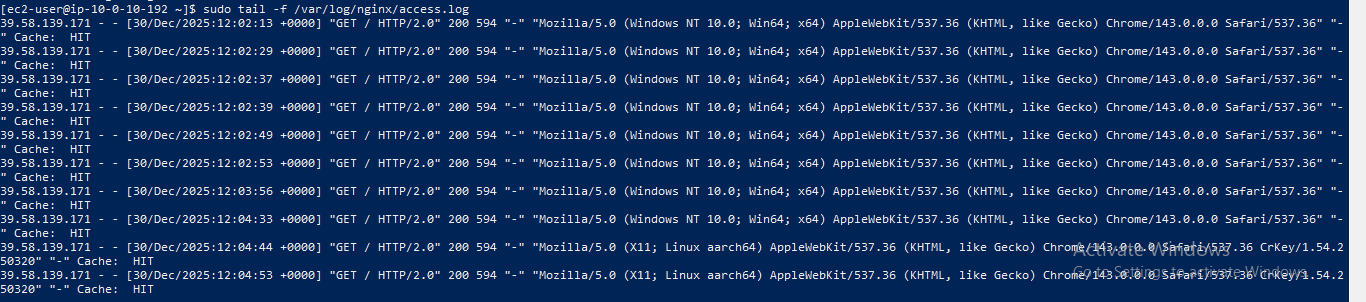


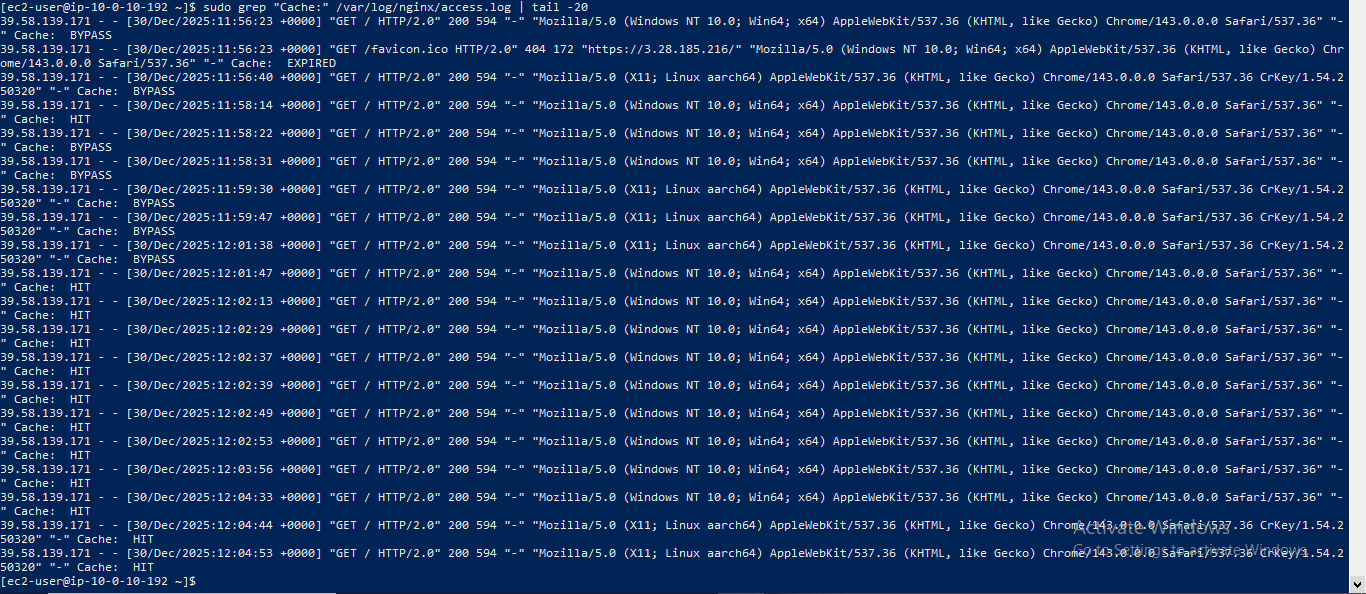


***5.3 Test Cache Functionality***

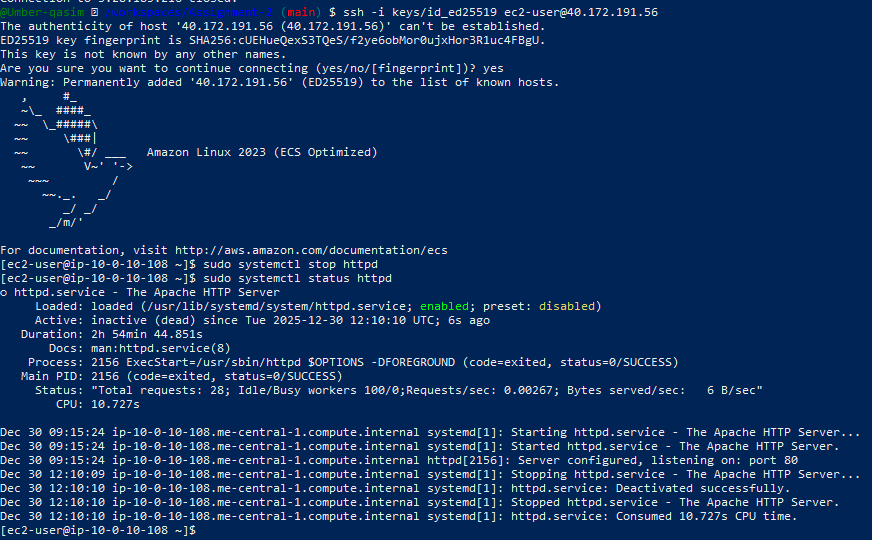


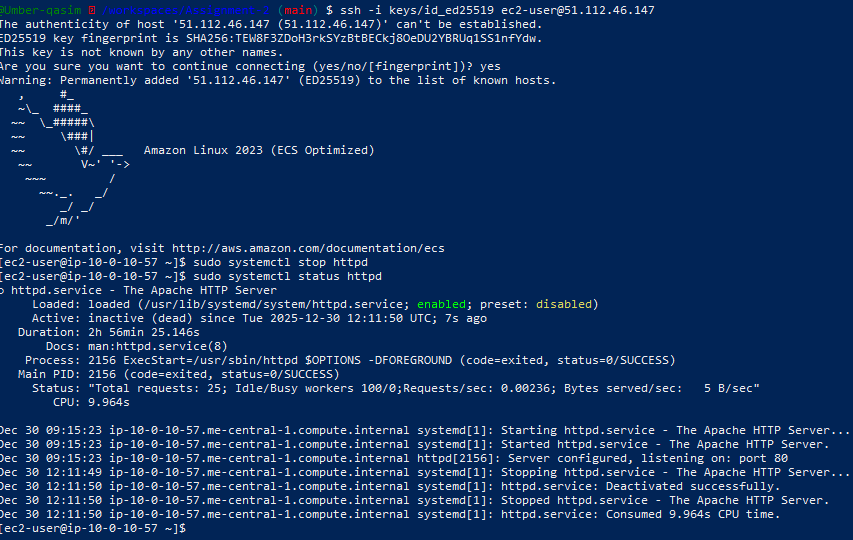


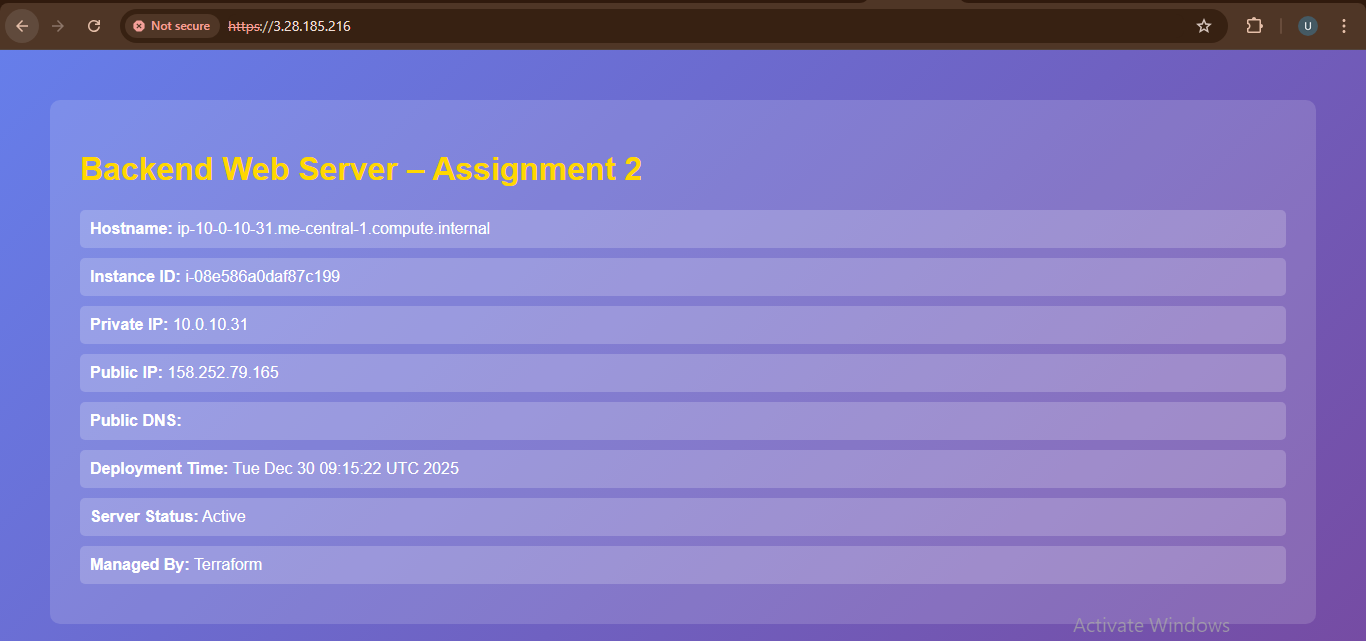


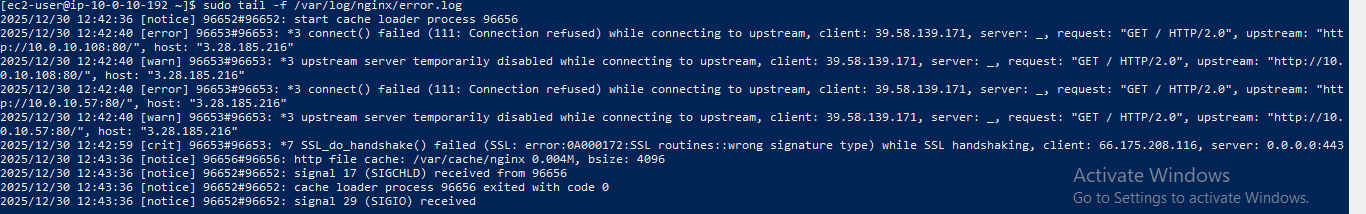


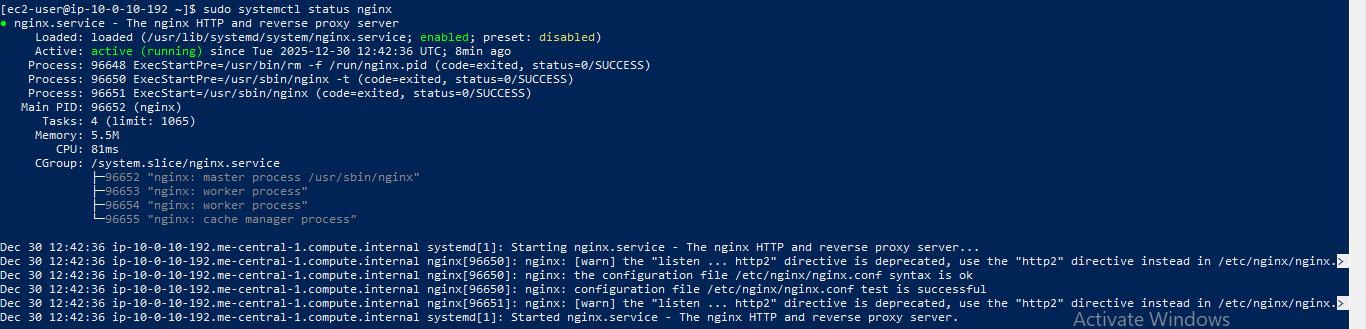
***5.4 Test High Availability (Backup Server)***

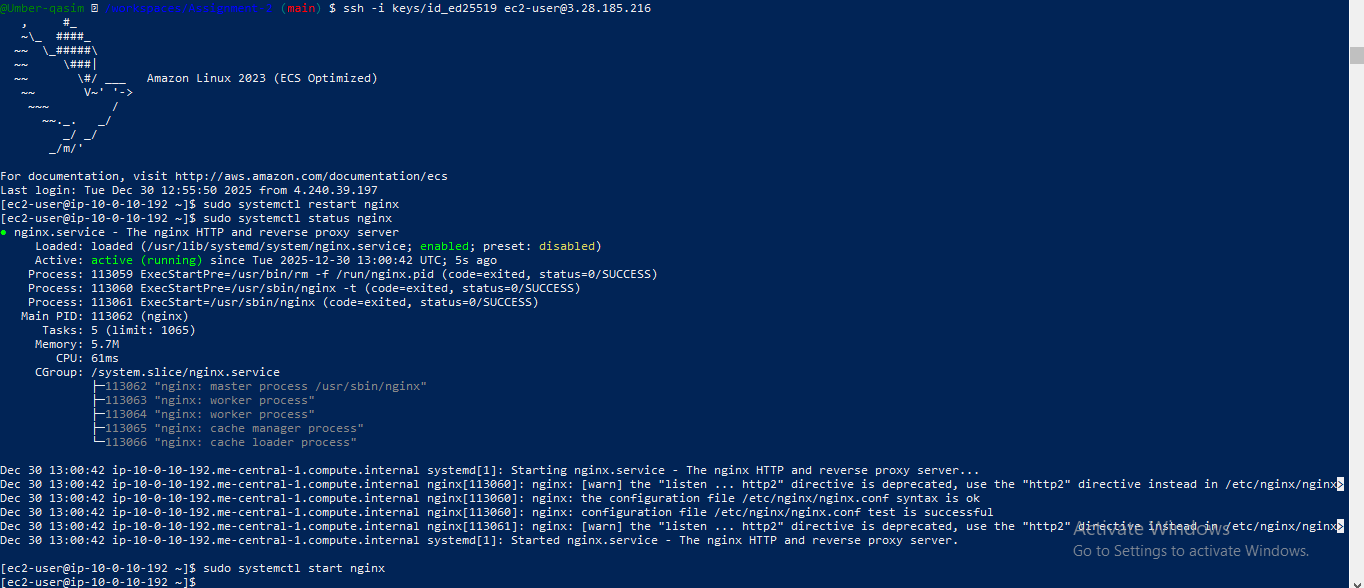




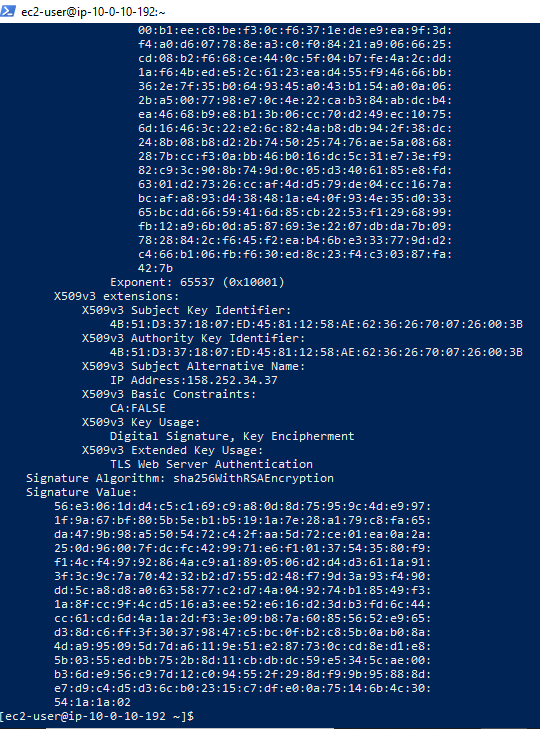


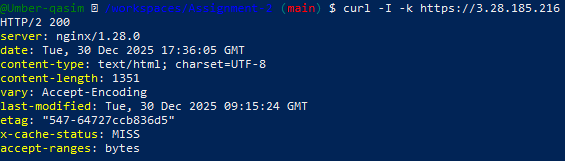


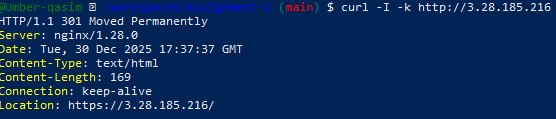


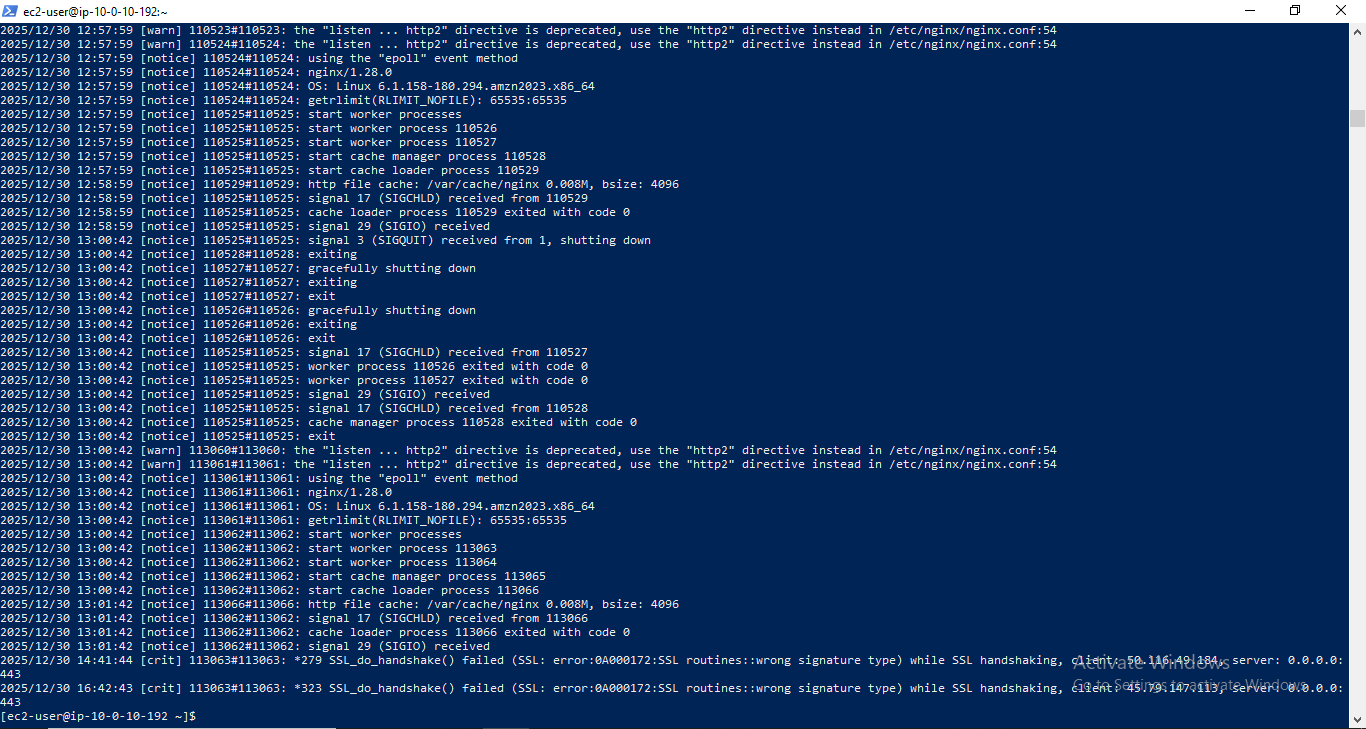


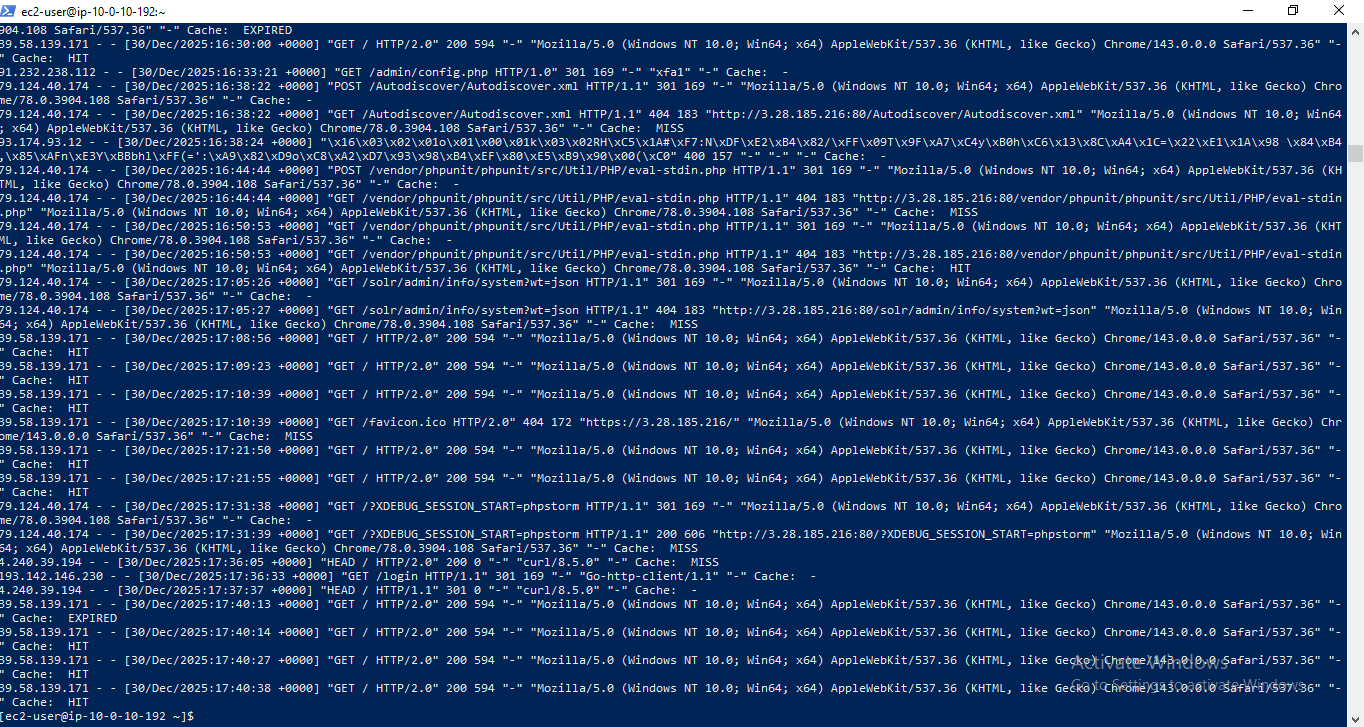
***5.5 Security & Performance Analysis***

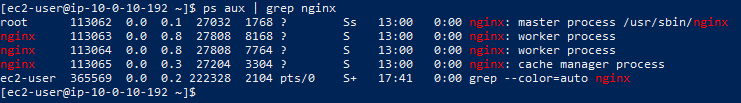












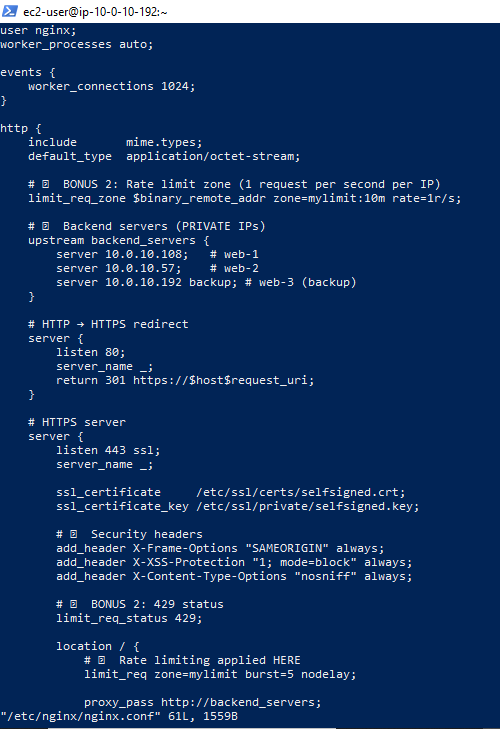
***Bonus Tasks***

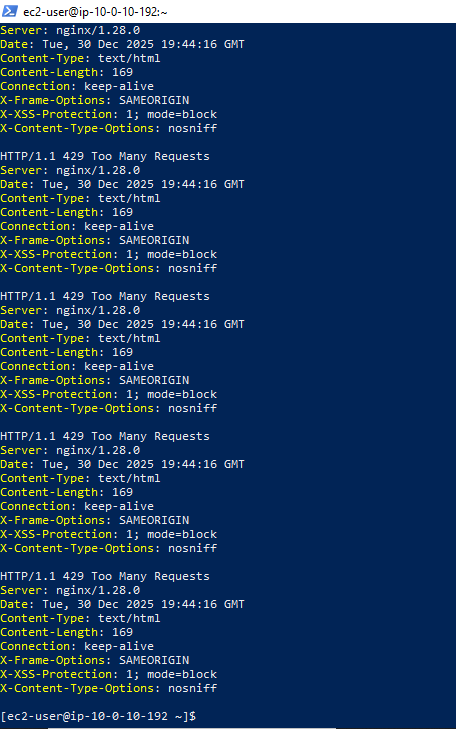
***Bonus 1: Custom Error Pages***



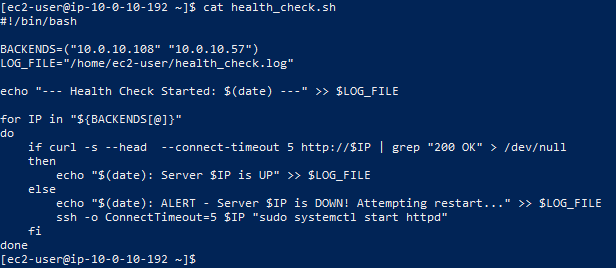


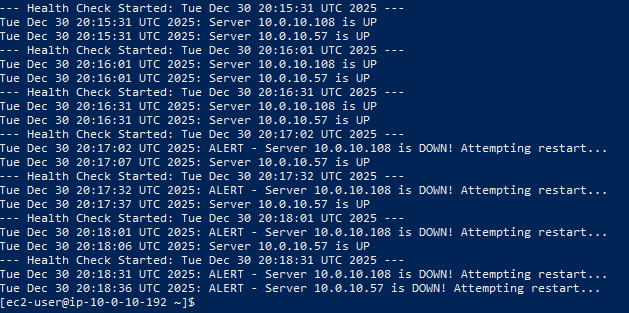
***Bonus 2: Implement Rate Limiting***





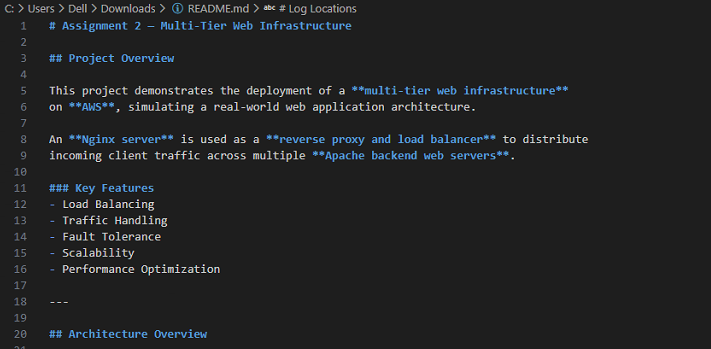
***Bonus 3: Health Check Automation***

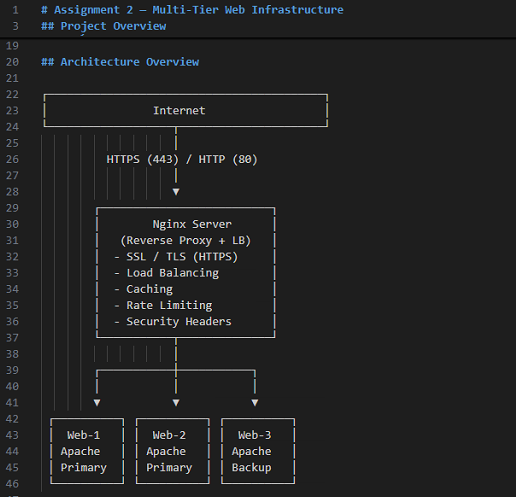


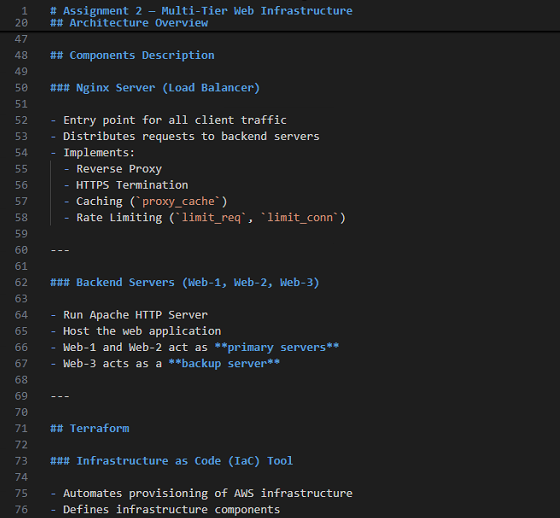
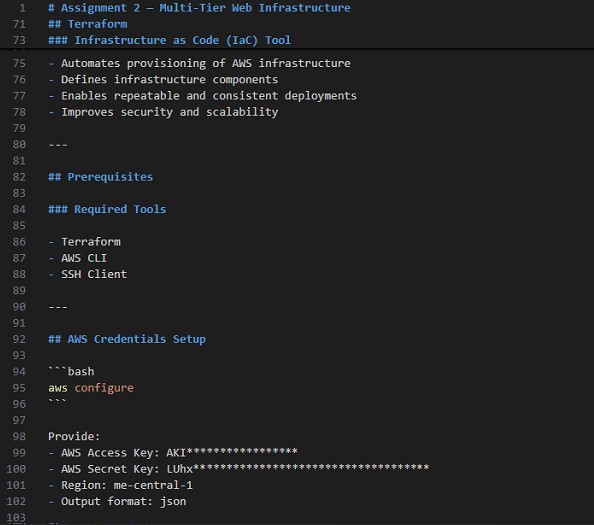


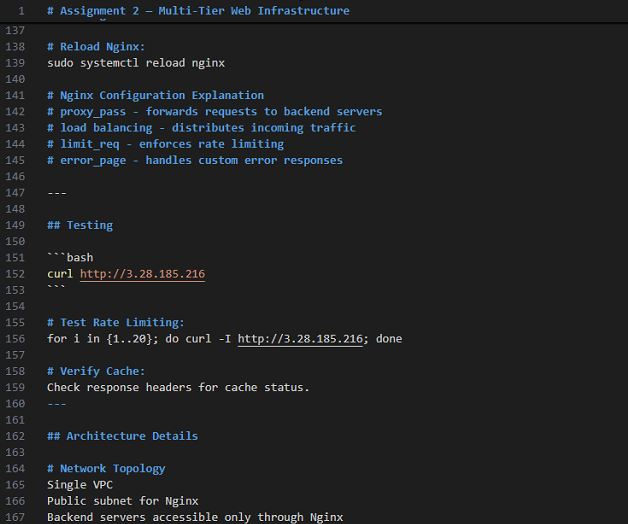
## **Part 6: Documentation & Cleanup**

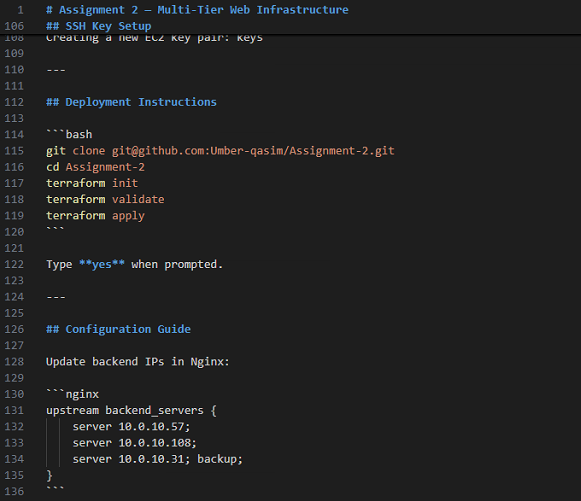
***6.1 README Documentation***

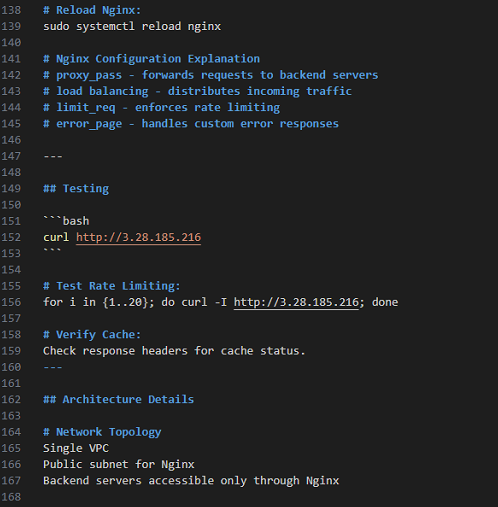


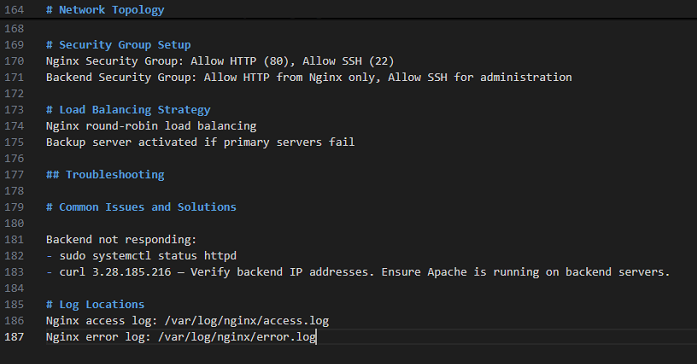




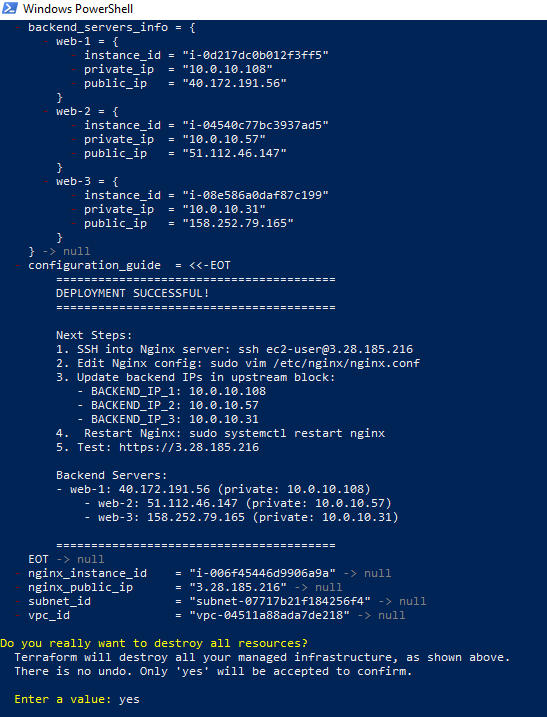


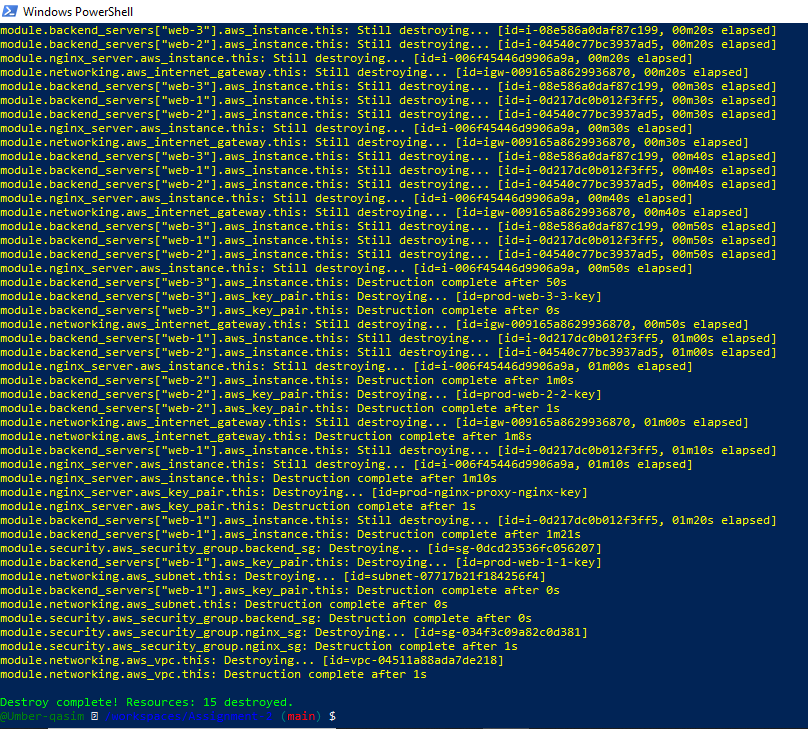




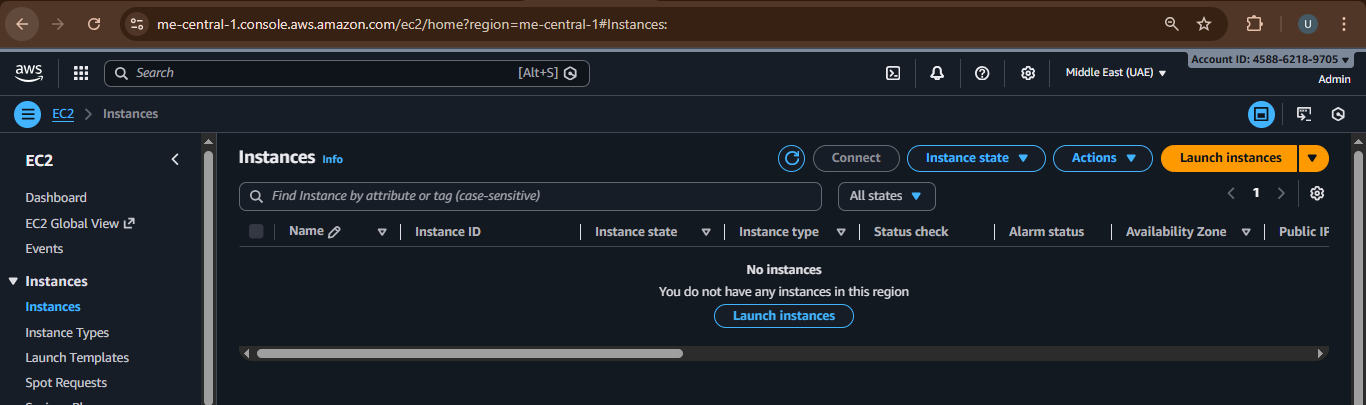


***6.2 Infrastructure Cleanup***









# **Challenges & Solutions**

## **Challenge 1: SSH Key Errors**

**Solution:** Regenerated key pairs and updated Terraform variables

## **Challenge 2: Backend Not Reachable**

**Solution:** Corrected security group rules

## **Challenge 3: Rate Limiting Not Triggering**

**Solution:** Reduced request rate threshold and tested using loops

## **Lessons Learned**

* Importance of IaC
* Security-first cloud design
* Debugging cloud networking issues

# **Conclusion**

This assignment successfully demonstrated the deployment of a production‑like multi‑tier architecture using AWS and Terraform.

It enhanced practical skills in:

* Cloud infrastructure
* Linux server automation
* Security and performance optimization
* Troubleshooting real-world issues

**Future Improvements**

* Use ACM certificates instead of self‑signed
* Auto Scaling Groups
* Monitoring with CloudWatch
* CI/CD integration

# **Appendices**

Included

* Complete Terraform code
* Apache and Nginx scripts
* Configuration files
* Screenshots

References:

* AWS Documentation
* Terraform Documentation
* Nginx Official Docs