Capstone Project Submission

Title: Effect of Orchestration and Microsegmentation on Web Application Security **Student Name:** Oluwaseun Osunsola **Student ID:** IDEAS/24/63183 **Program:** Professional Diploma in Cybersecurity – Baze University

Abstract

This project investigates how orchestration and microsegmentation influence web application security. By deploying a WordPress web application on AWS using CloudFormation templates, the project demonstrates automated provisioning, autoscaling, and the enforcement of microsegmentation principles through security groups. The experiment verifies that these strategies improve web application availability, resilience against misconfigurations, and isolation of critical resources, thereby reducing the attack surface.

1. Introduction

Web applications are often targeted due to their exposure to the internet. Security risks arise from misconfigured servers, unpatched software, and lack of network isolation. Modern deployment strategies such as **orchestration** (automated provisioning and scaling) and **microsegmentation** (fine-grained network access control) are increasingly adopted to improve security posture.

This project focuses on implementing a fully orchestrated WordPress web application with microsegmented network components on AWS and testing its security, availability, and resilience under load.

2. Objectives

1. Deploy a fully functional WordPress web application using AWS CloudFormation orchestration.
2. Implement microsegmentation via AWS Security Groups to isolate the database and web servers.
3. Test autoscaling behavior under high CPU load to assess resilience.
4. Verify that security and connectivity remain intact after termination or recreation of resources.

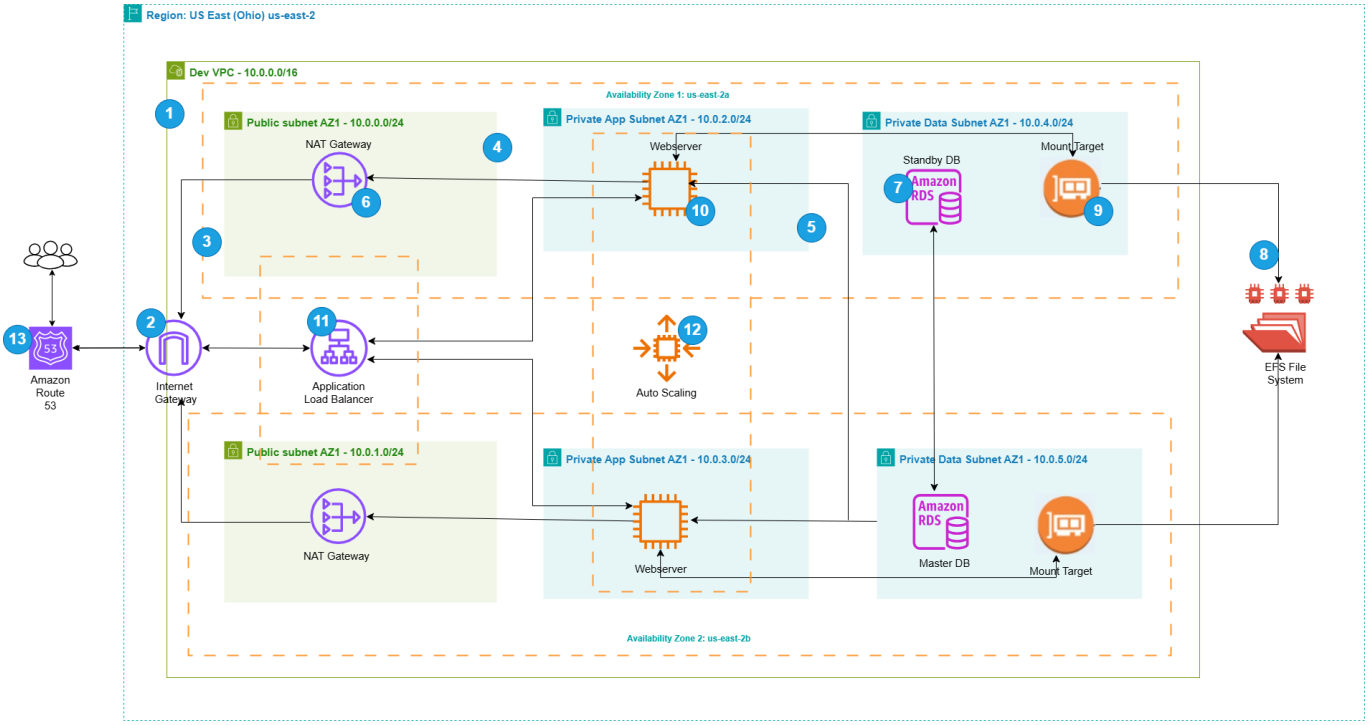
3. Methodology

3.1 Architecture Design

The architecture consists of:

- **Frontend:** EC2 instances hosting WordPress.
- **Database:** RDS instance.
- **Load Balancer:** AWS Application Load Balancer (ALB).
- **Orchestration:** AWS CloudFormation for provisioning and management.
- **Microsegmentation:** Security groups controlling inbound/outbound traffic.

Screenshot Reference:

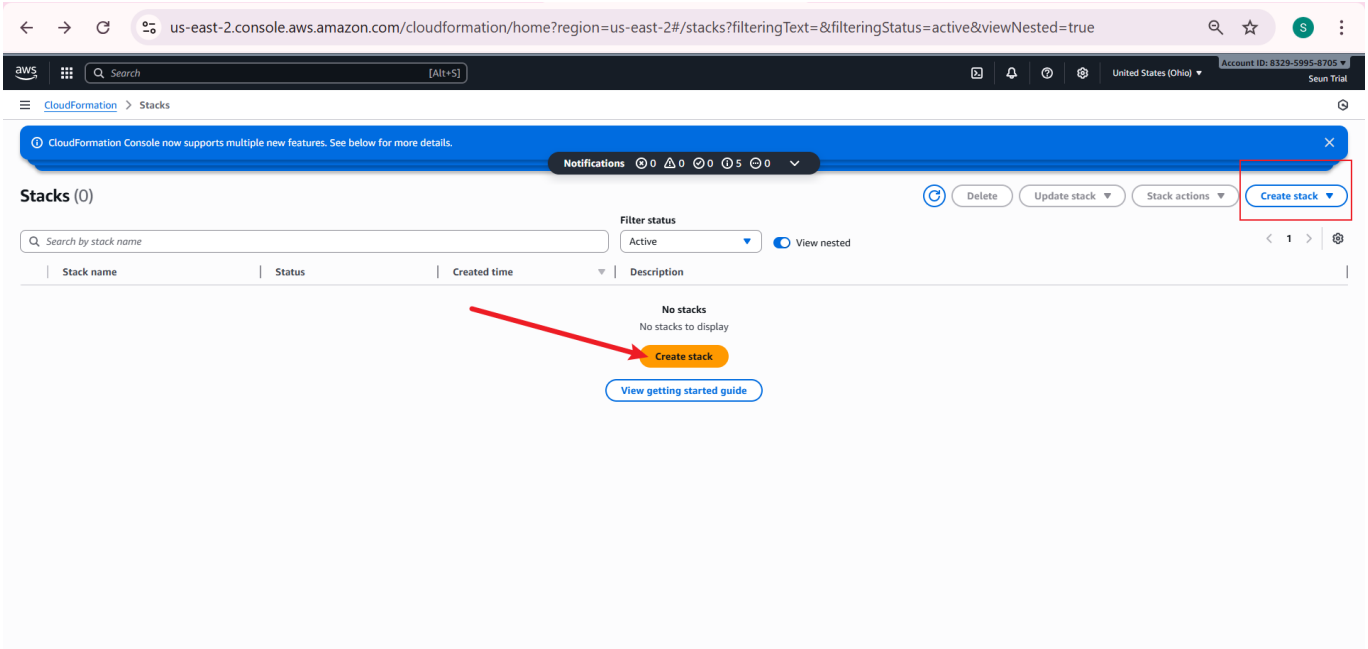


3.2 Deployment Steps

Step 1: Navigate to CloudFormation Dashboard

- Access AWS CloudFormation console.
- Click *Create Stack*.

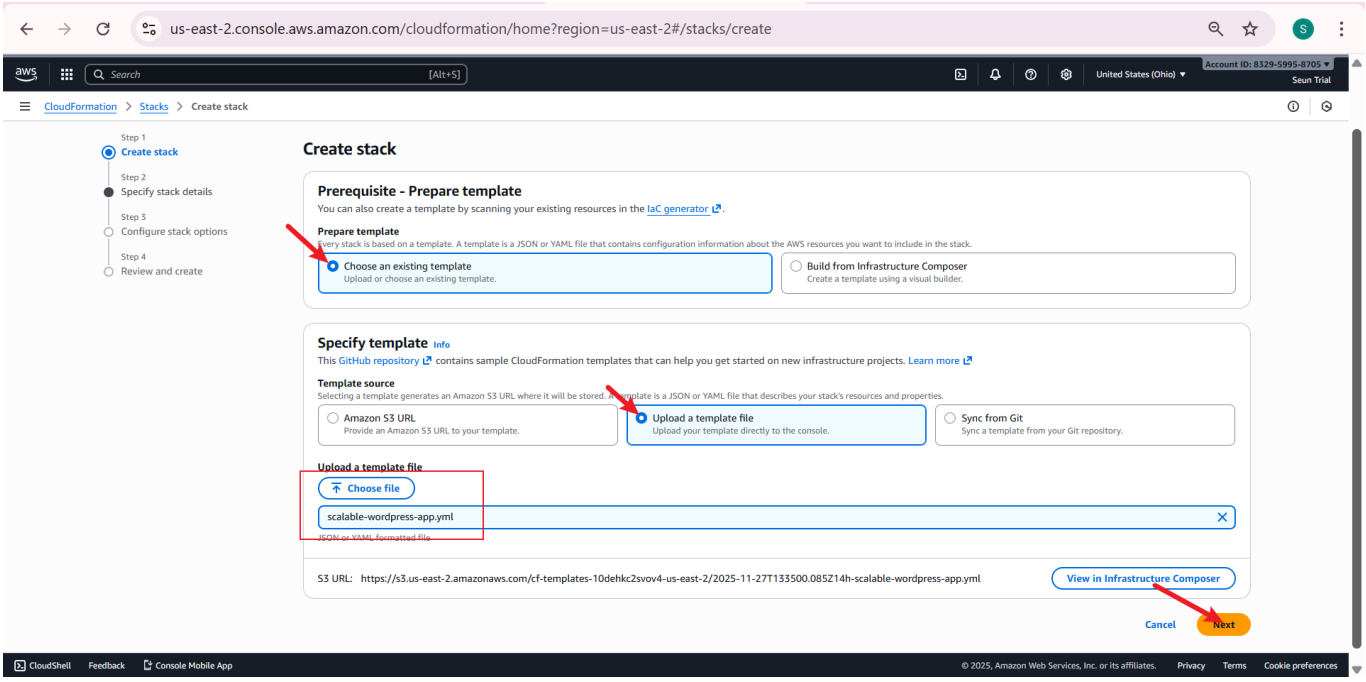
Screenshot Reference:



Step 2: Upload Template

- Choose existing template (YAML/JSON) for WordPress deployment.
- Click *Next*.

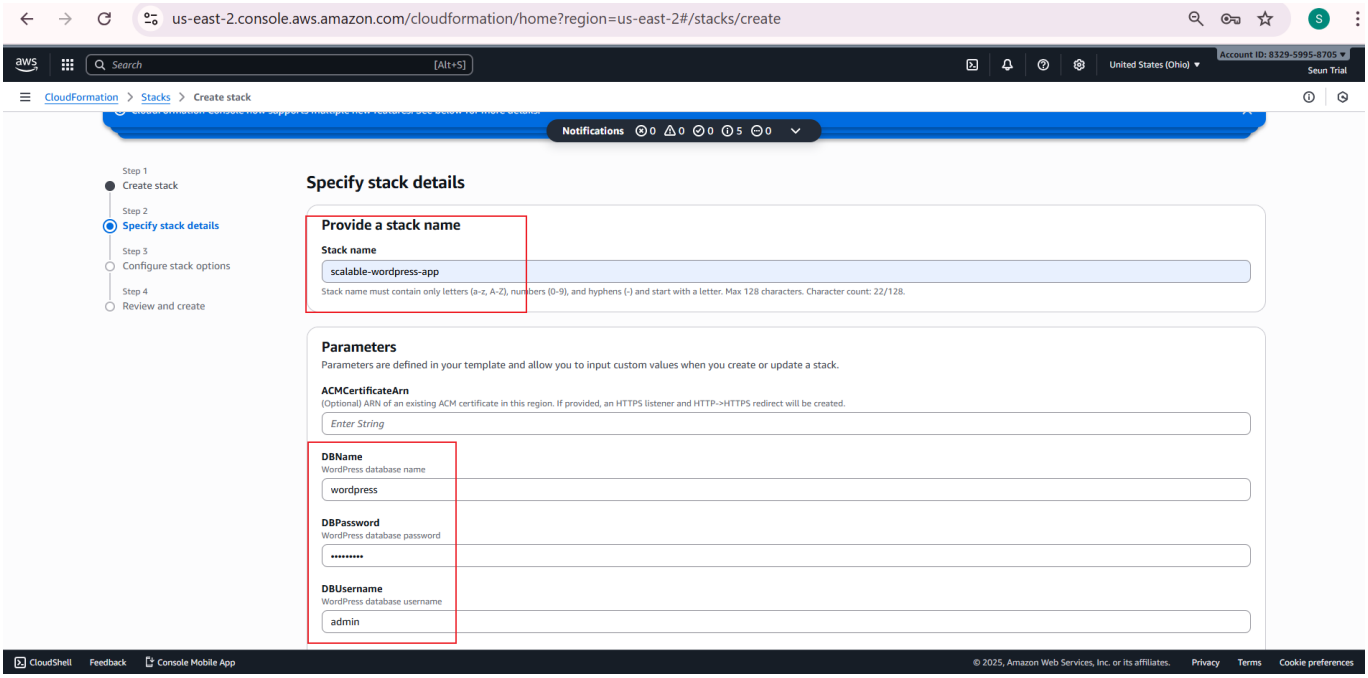
Screenshot Reference:



Step 3: Configure Stack

- Enter stack name.
- Enter database credentials.
- Scroll to next steps.

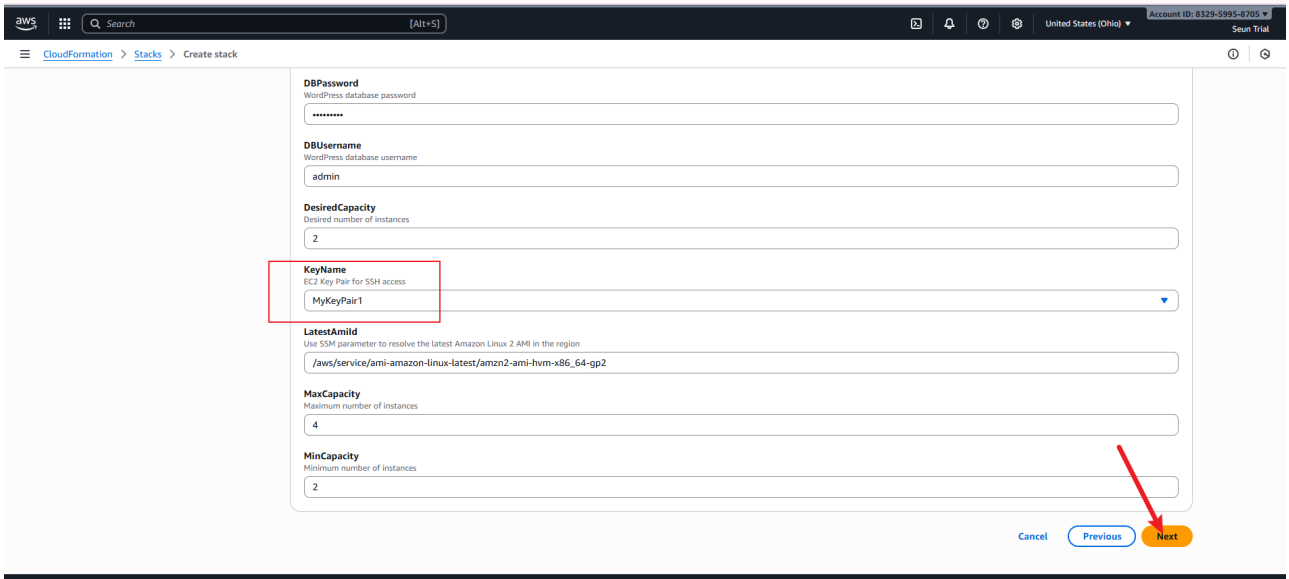
Screenshot Reference:

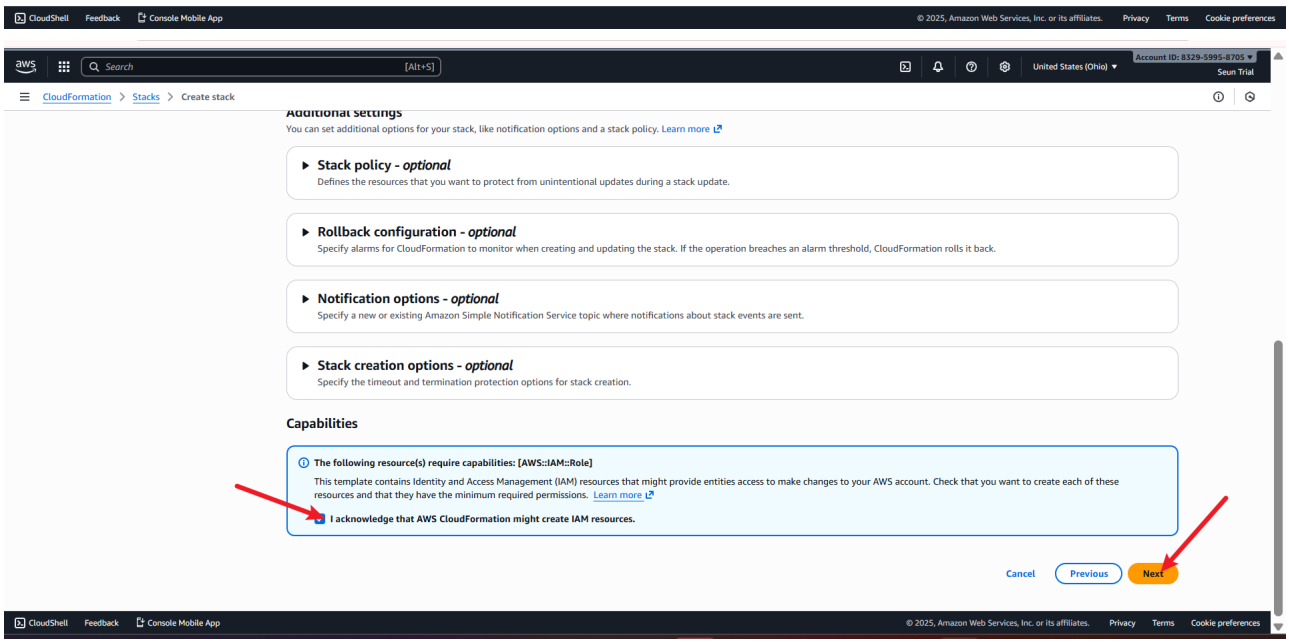


Step 4: Key Pair & IAM Acknowledgment

- Select key pair. Leave defaults.
- Acknowledge IAM resource creation warning.

Screenshot References:

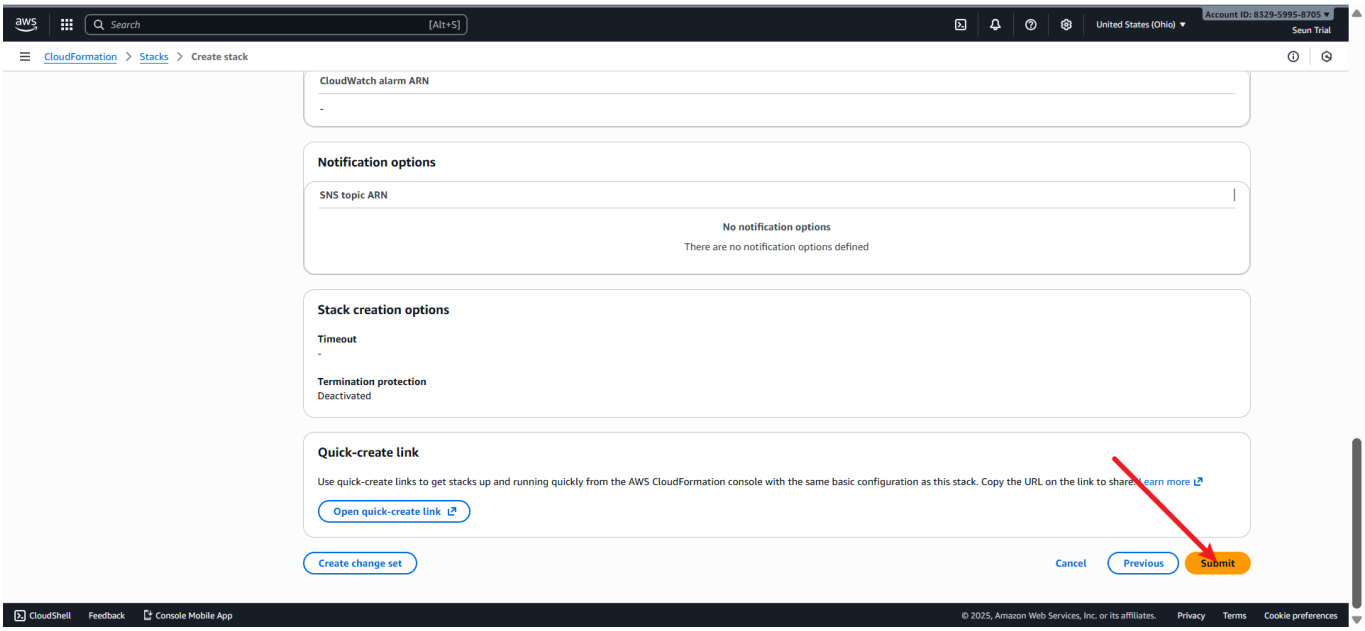
- 

- 

Step 5: Submit Stack

- Leave other options at default.
- Submit.

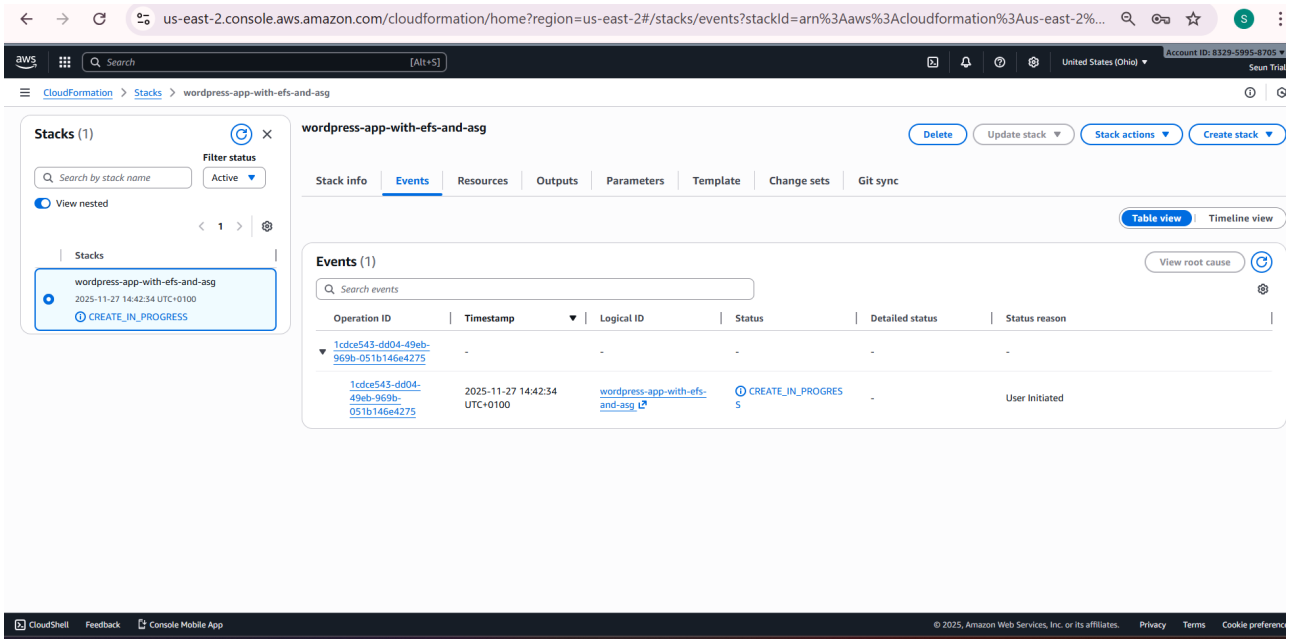
Screenshot Reference:

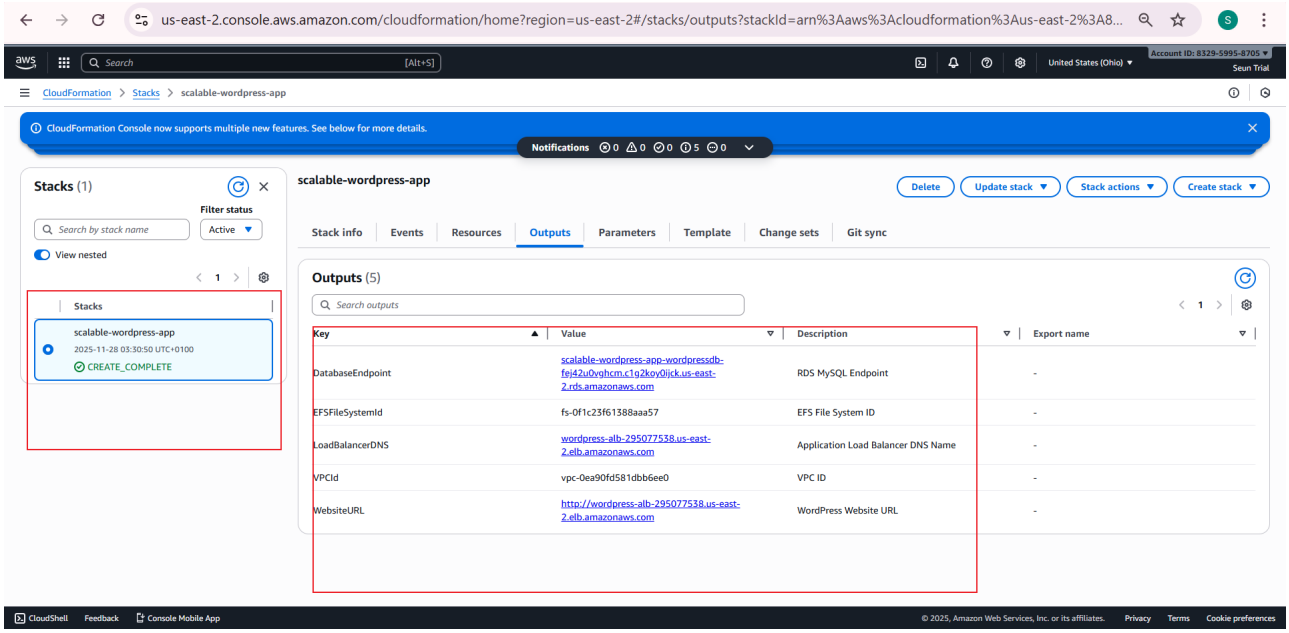


Step 6: Monitor Stack Creation

- Wait until stack creation completes.

Screenshot References:

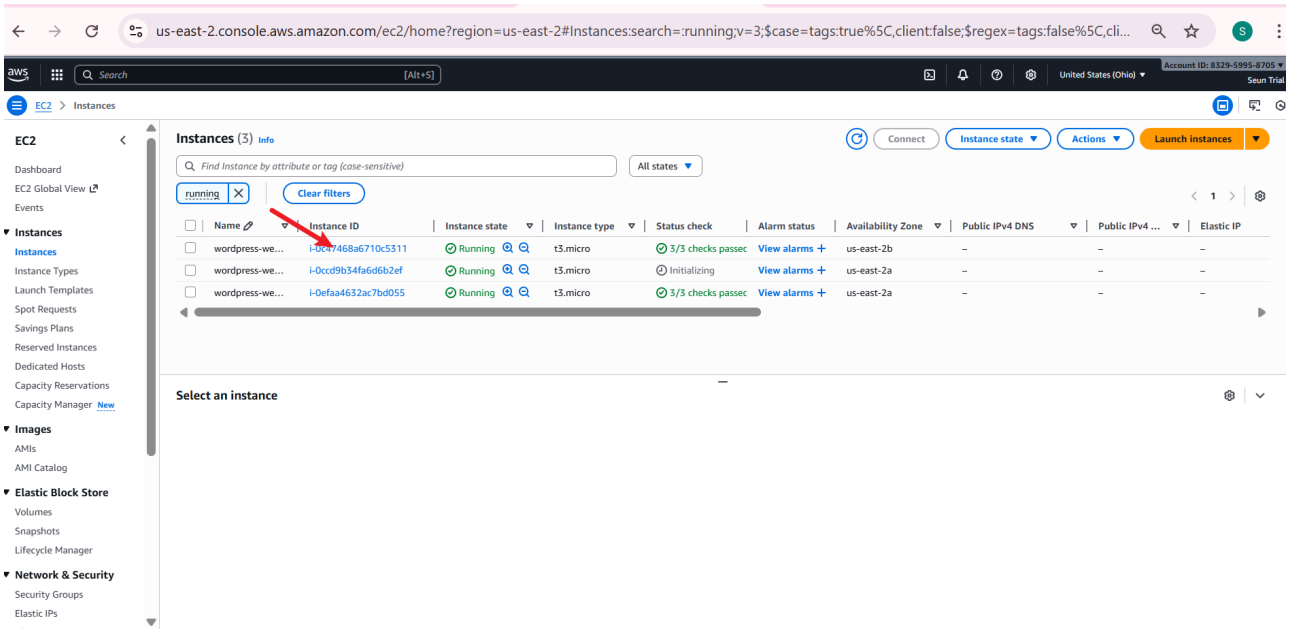
- 

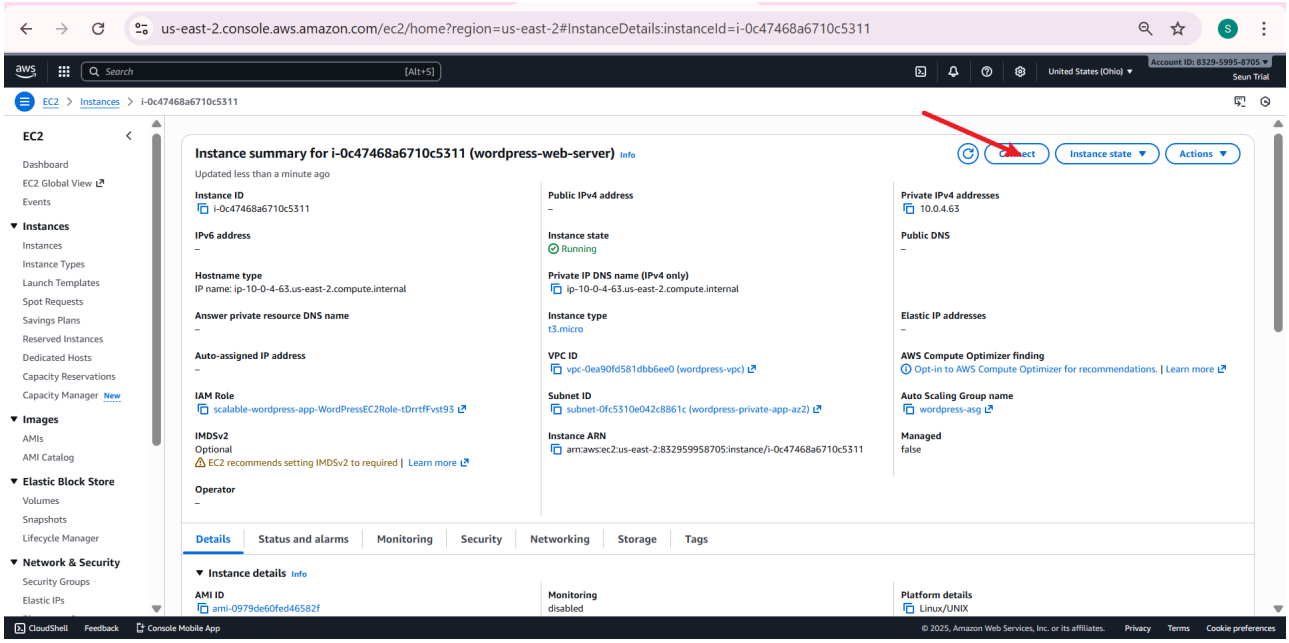


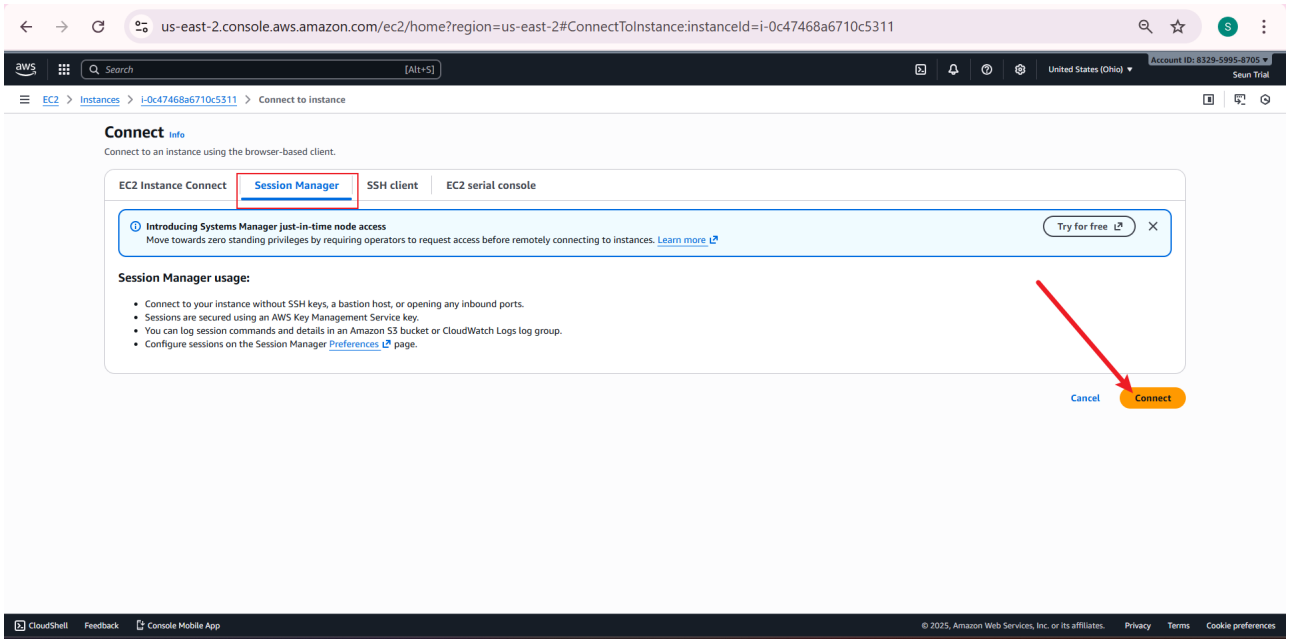
3.3 Connect to EC2 Instance

- Navigate to running EC2 instances.
- Connect using **Session Manager**.

Screenshot References:



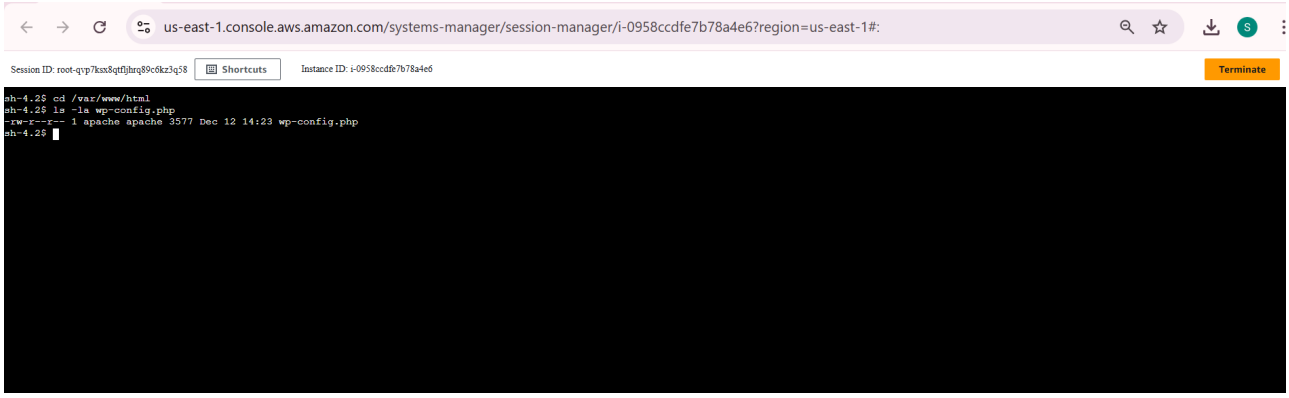


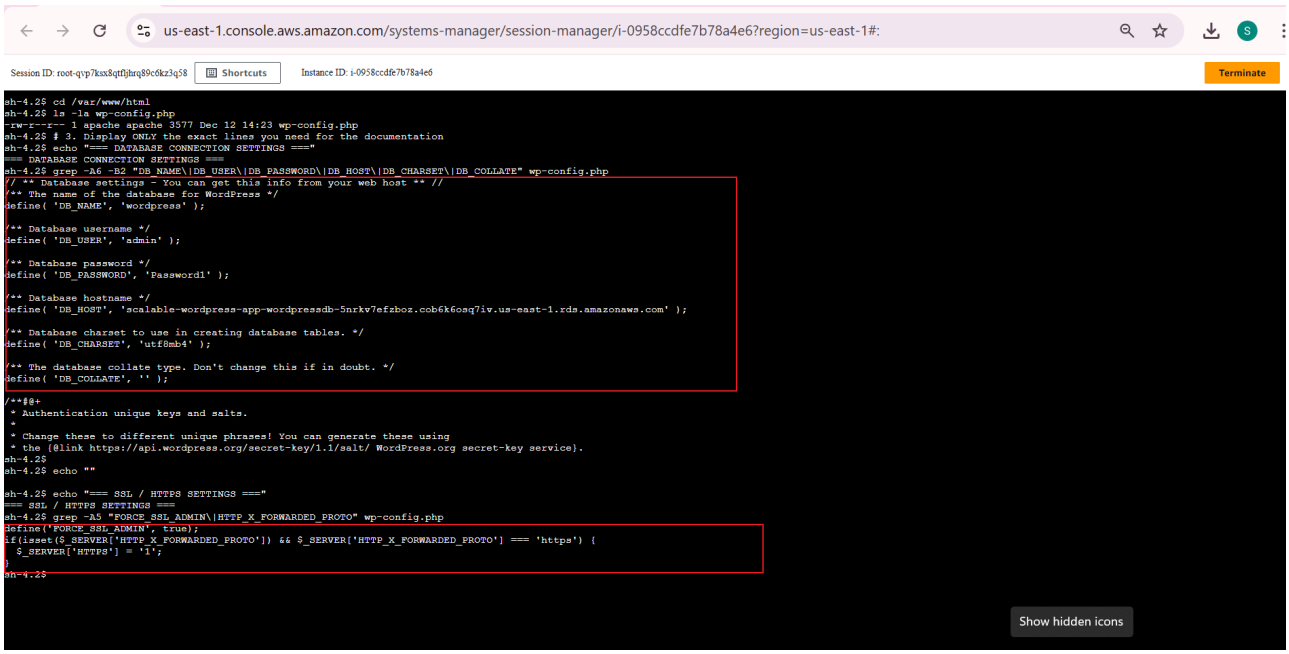


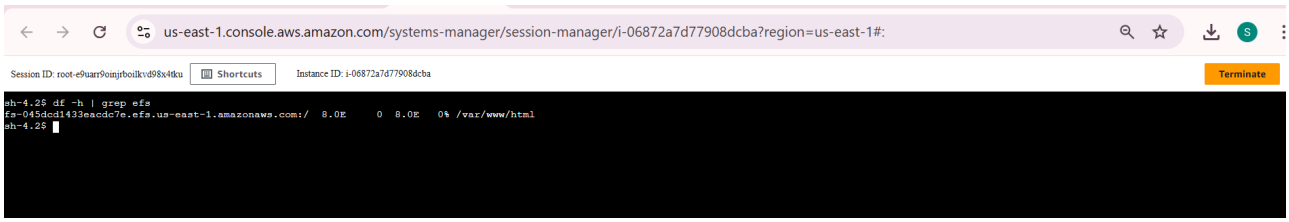
3.4 Verify WordPress Deployment

- Navigate to WordPress folder.
- List contents to verify `wp-config.php`.
- Check database connection and HTTPS settings.

Screenshot References:

- 

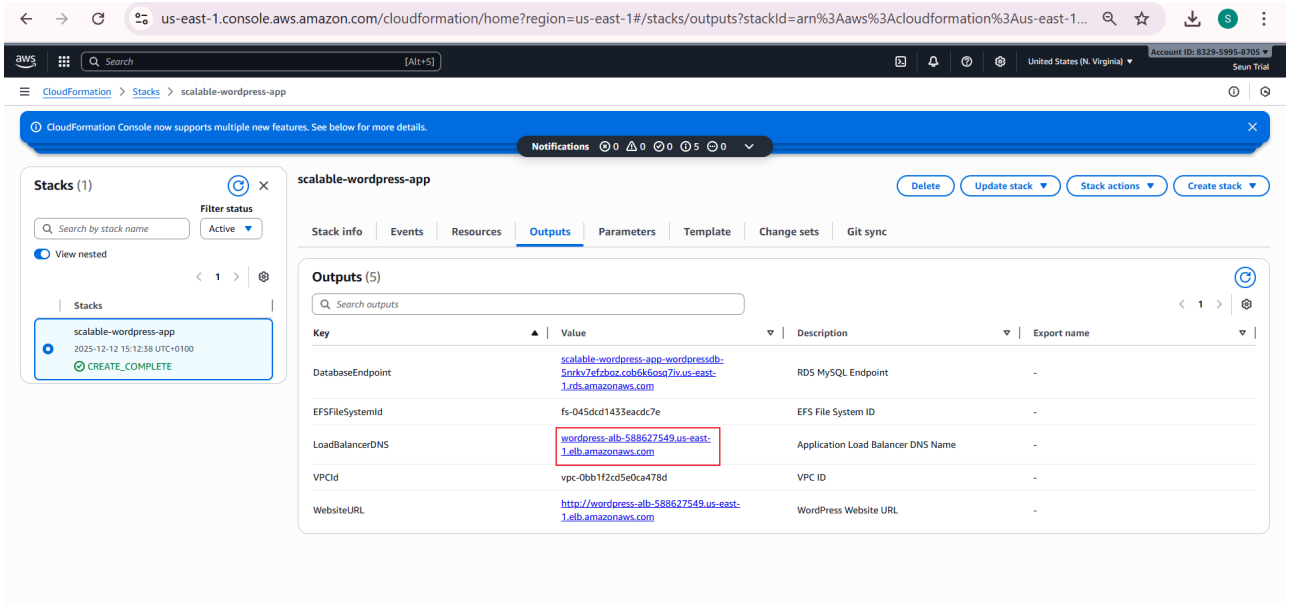
- 

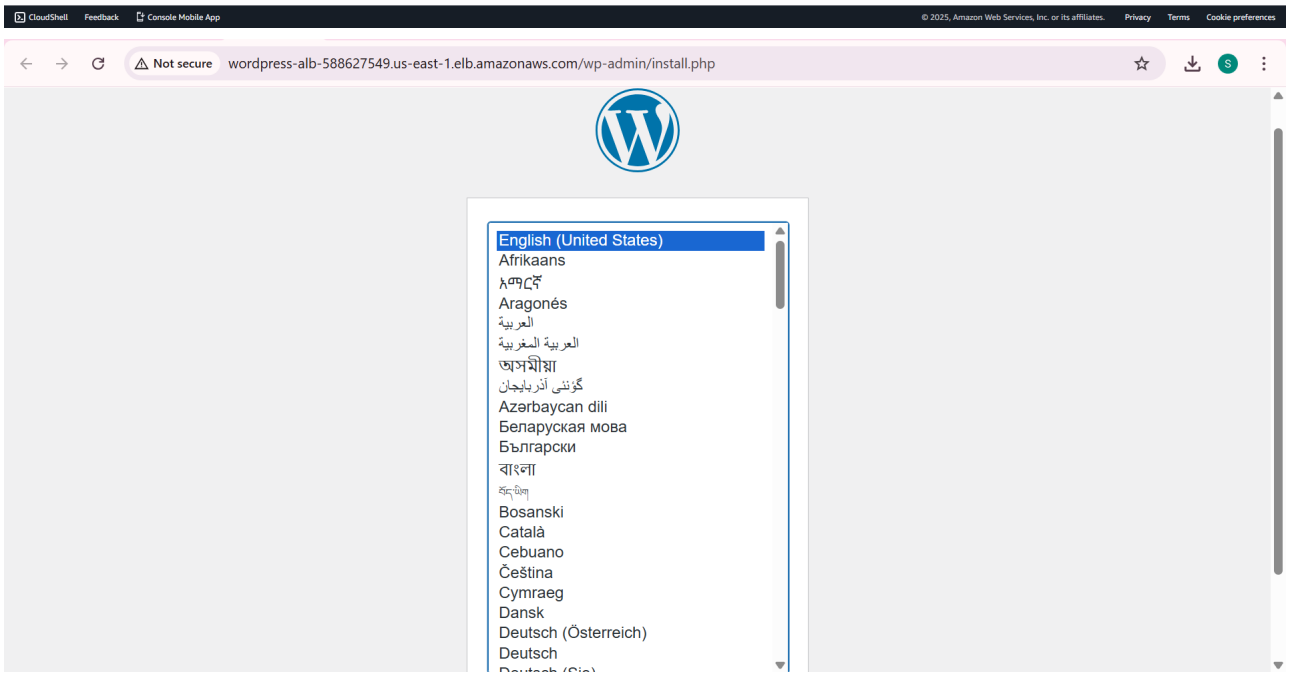
- 

3.5 Test Orchestration (Autoscaling)

Step 1: Check Load Balancer DNS output.

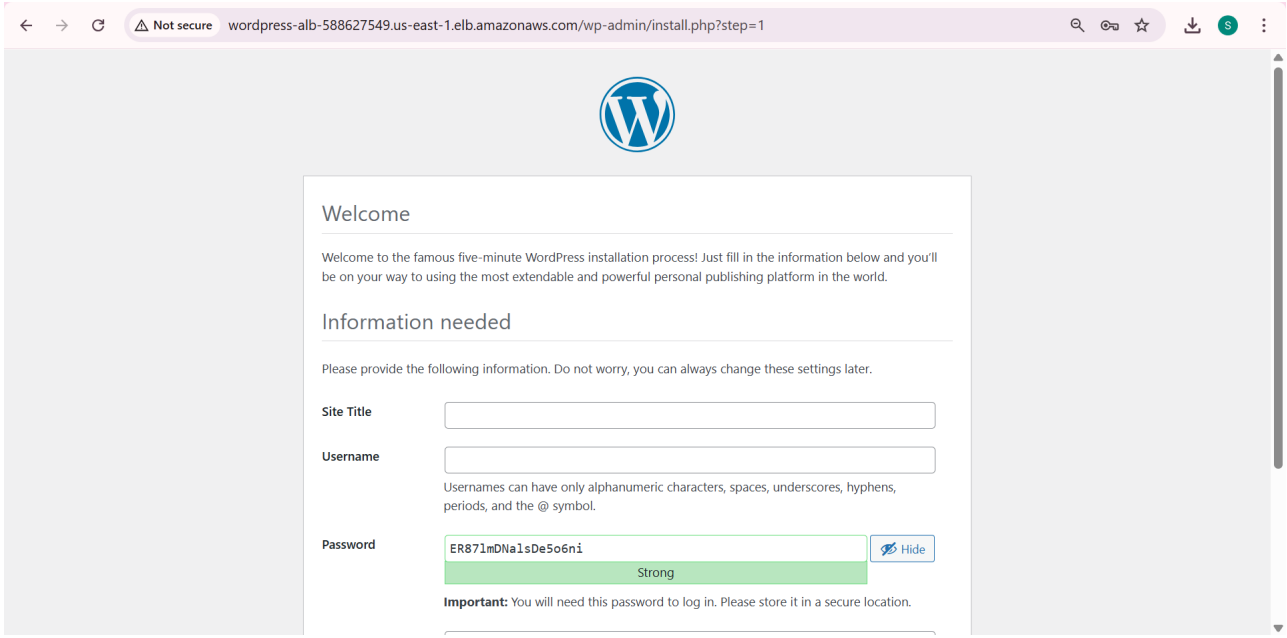
Screenshot References:





Step 2: Confirm configuration files.

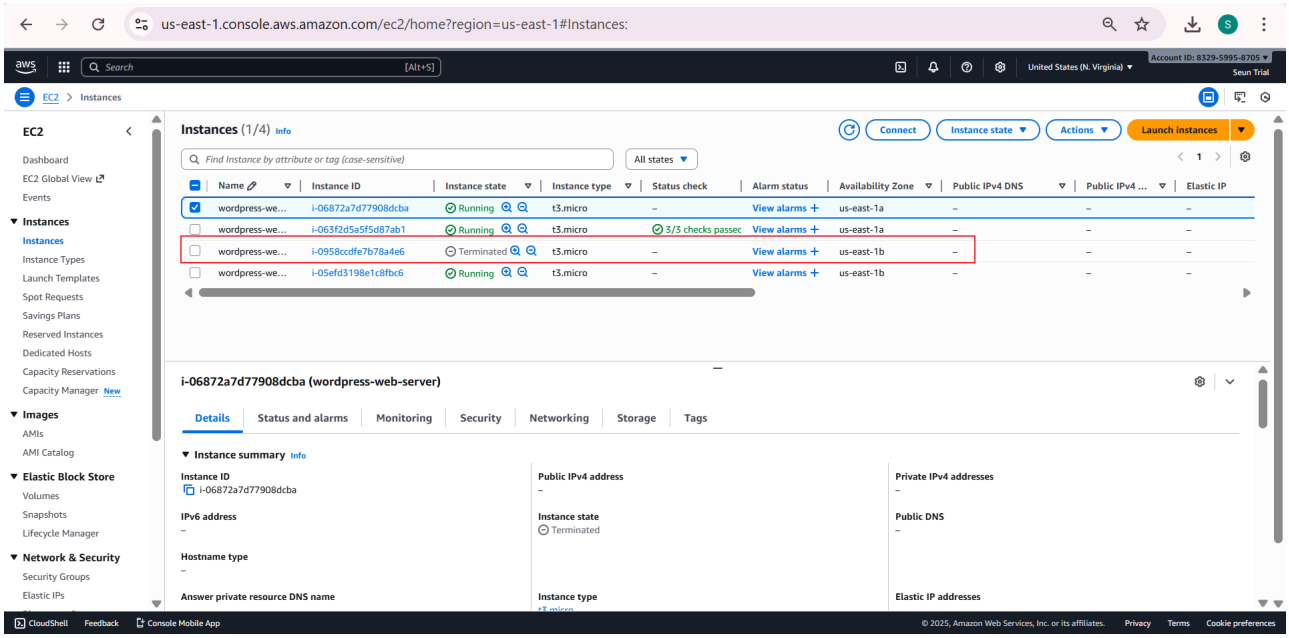
Screenshot Reference:

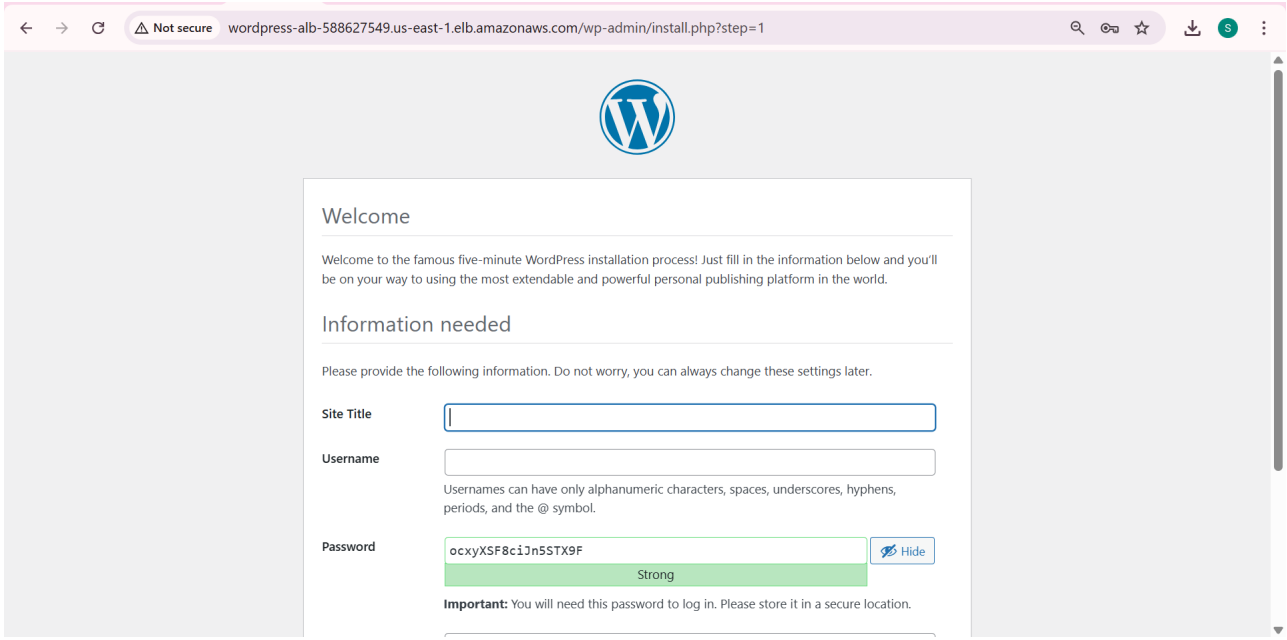
- 

Step 3: Terminate an EC2 instance to simulate failure.

- Verify website availability via Load Balancer.

Screenshot References:

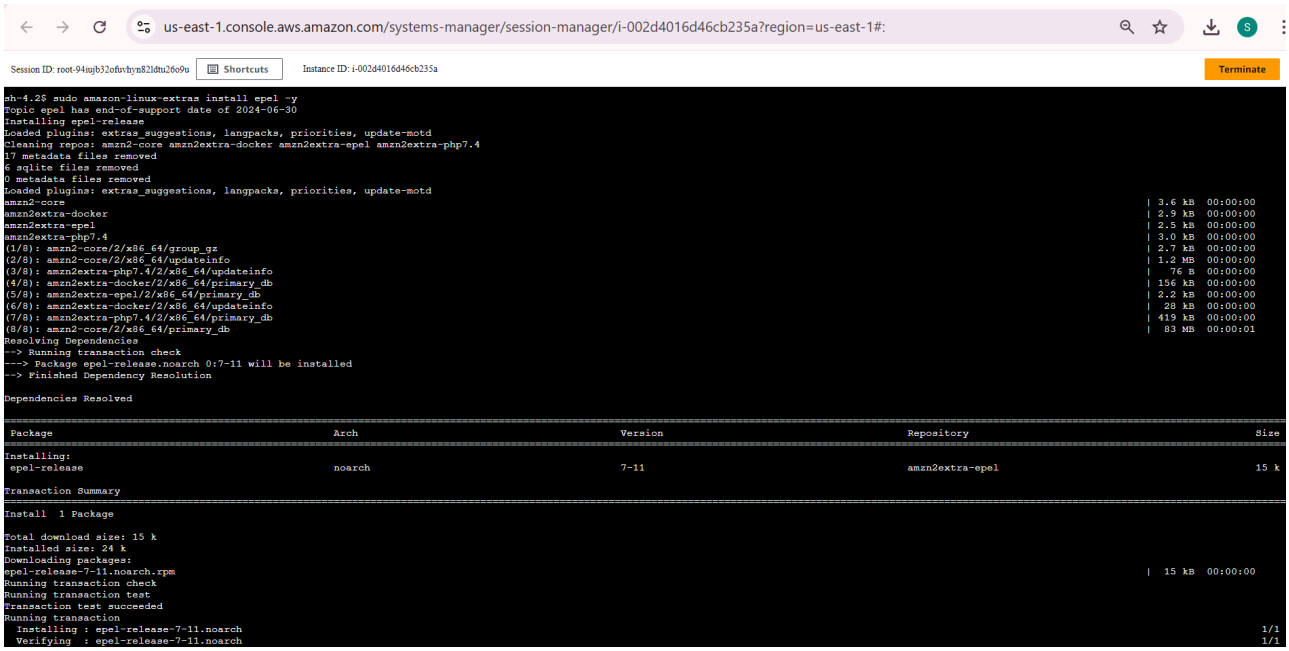
- 

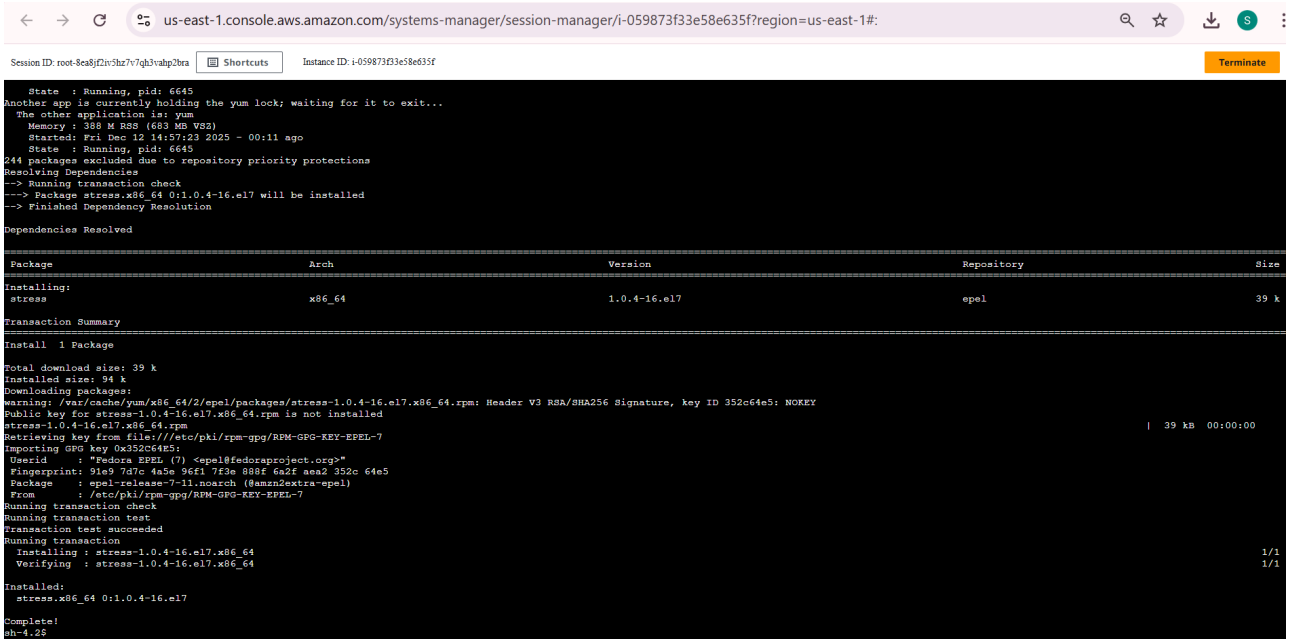
- 

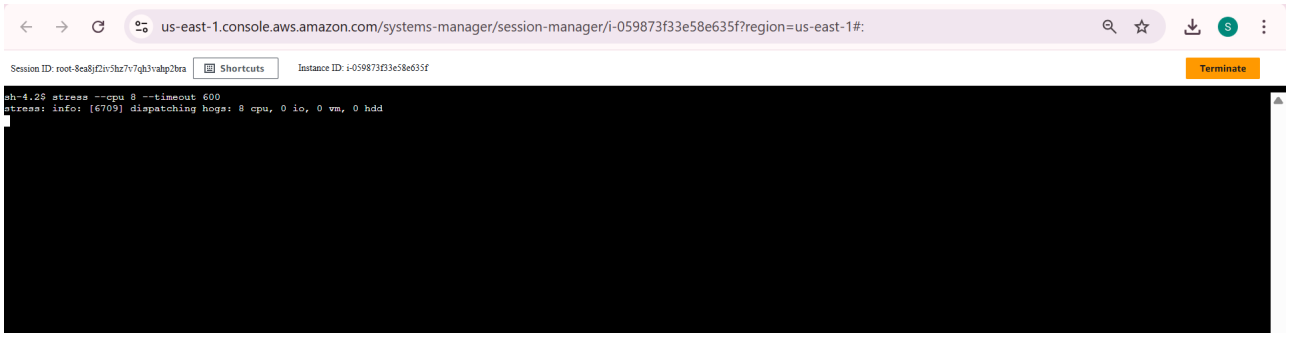
Step 4: Generate high CPU load to test autoscaling.

- Install stress package.
- Load CPU for 10 mins.

Screenshot References:

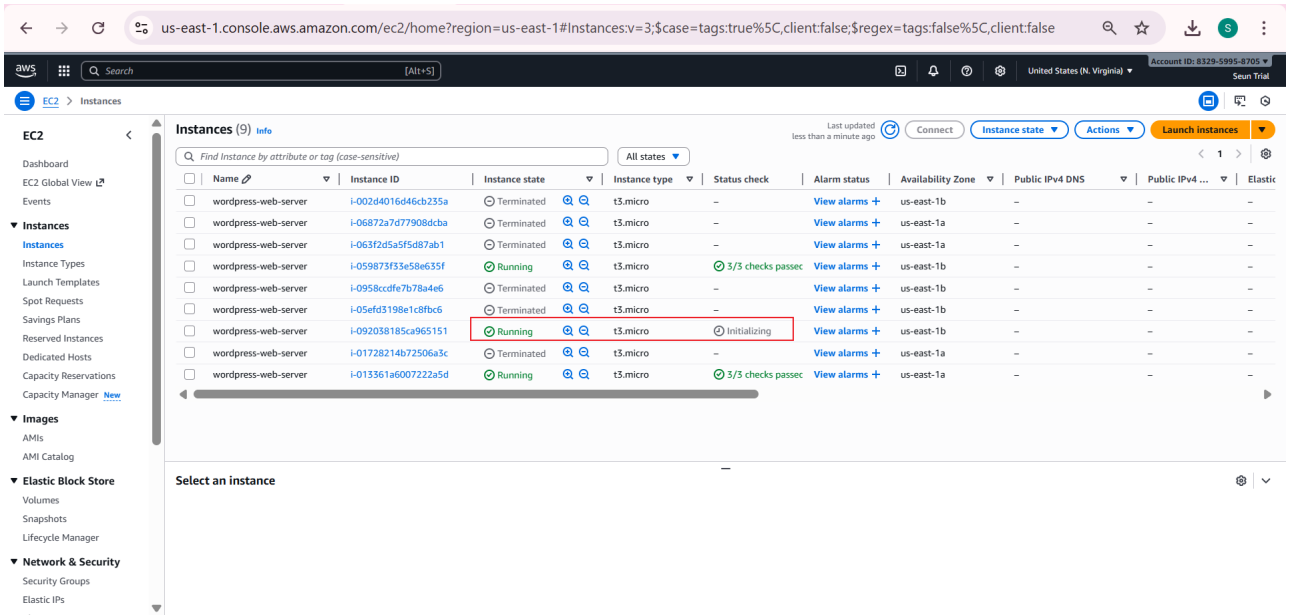
- 

- 

- 

Step 5: Verify new EC2 instance spins up under load.

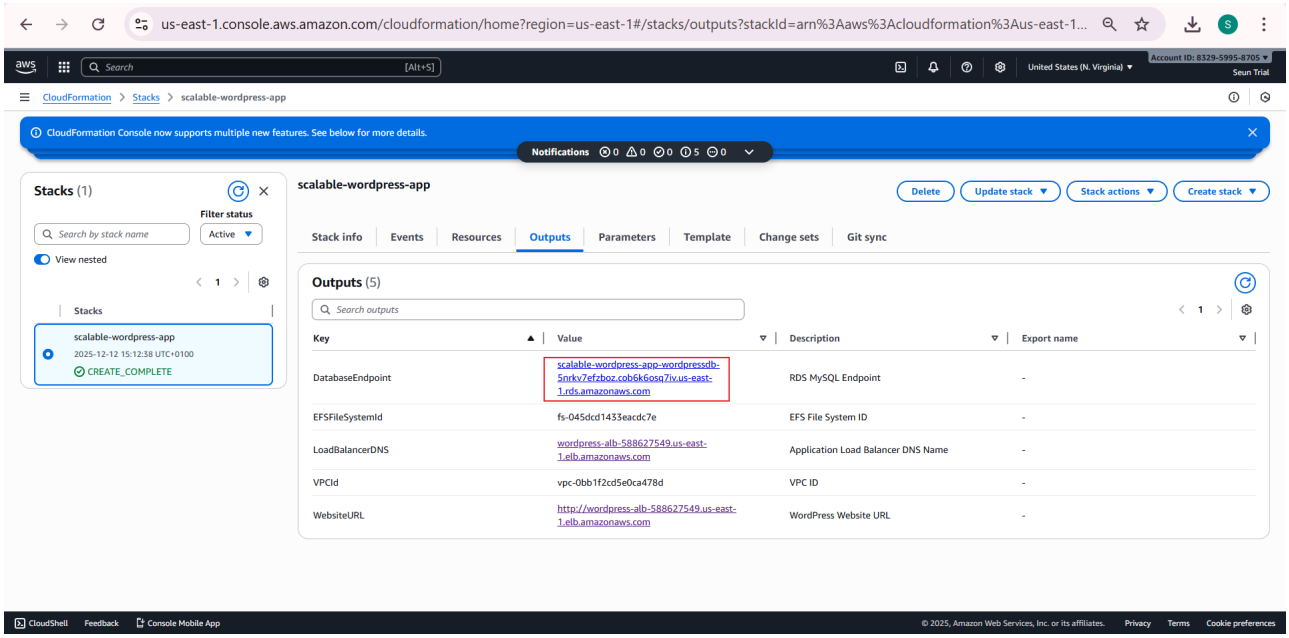
Screenshot Reference:

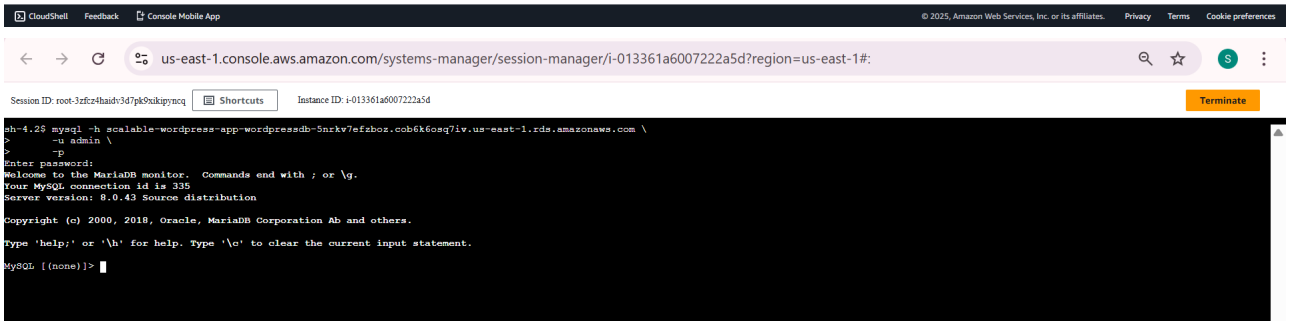
- 

3.6 Verify Database Microsegmentation

- Copy RDS Database Endpoint.
- Test connection using credentials.

Screenshot References:

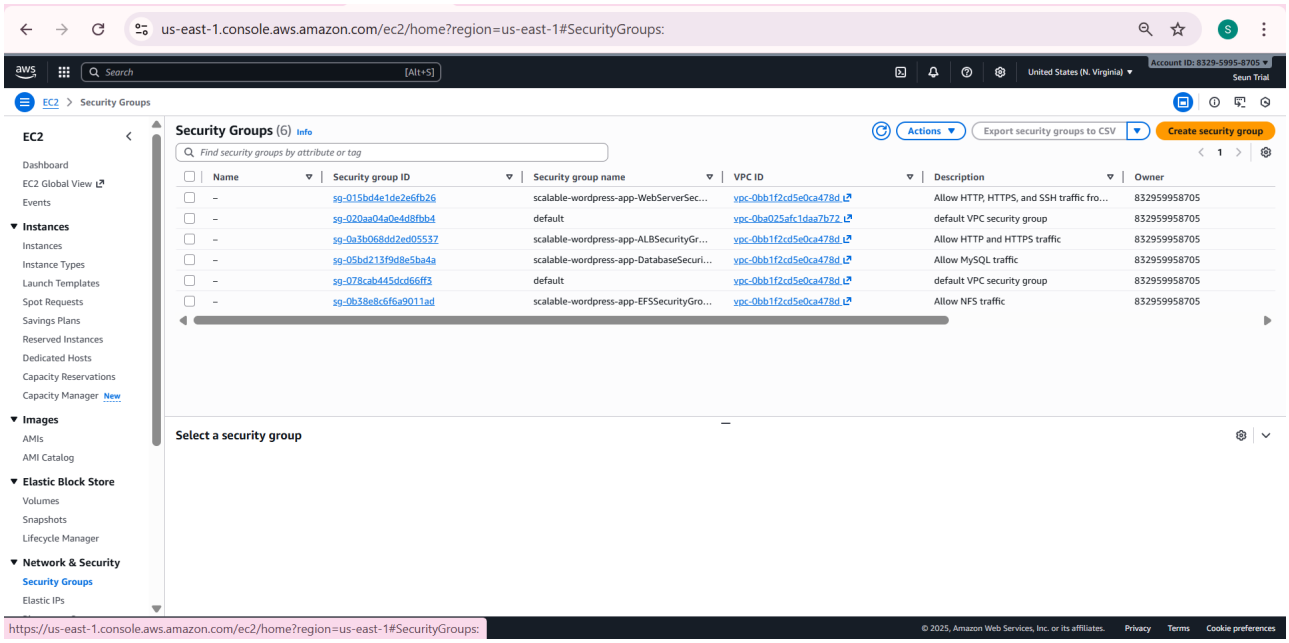
- 

- 

3.7 Verify Security Groups (Microsegmentation)

- Check security groups for EC2 and RDS instances.
- Confirm that only necessary ports are open (HTTP/HTTPS for web servers, MySQL for database).

Screenshot Reference:

- 

---

4. Results

1. CloudFormation successfully orchestrated the WordPress deployment.
2. Autoscaling worked as expected: terminating an instance did not disrupt service.
3. High CPU load triggered the creation of additional EC2 instances, maintaining availability.
4. Microsegmentation ensured database and web servers were isolated, reducing exposure.

Conclusion: Orchestration combined with microsegmentation significantly improved both security and resilience of the web application.

---

5. Discussion

- **Orchestration Benefits:** Automated provisioning reduces human errors, ensures consistent configurations, and supports rapid scaling.
- **Microsegmentation Benefits:** Limits lateral movement within the network; if one instance is compromised, others remain protected.
- **Security Posture:** The combination provides strong protection against common threats, misconfigurations, and downtime.

---

6. References

1. AWS Documentation: Amazon Web Services. (n.d.). AWS CloudFormation User Guide. Retrieved from https://docs.aws.amazon.com/cloudformation/. (Note: This also covers EC2, RDS, and Security Groups within the AWS documentation suite used in your methodology). +2

2. OWASP Top 10: OWASP Foundation. (2021). OWASP Top 10:2021 - The Ten Most Critical Web Application Security Risks. https://owasp.org/www-project-top-ten/.

3. NIST SP 800-125: Scarfone, K., Souppaya, M., & Hoffman, P. (2011). Guide to Security for Full Virtualization Technologies (NIST Special Publication 800-125). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-125.