National College of Ireland

Configuration Manual

MSc Research Project
MSc in Cloud Computing

TEJASWINI DHANESH KUMAR
Student ID: X23288957

School of Computing
National College of Ireland

Supervisor: SAI EMANI

National College of Ireland
MSc Project Submission Sheet
School of Computing

Student Name: Tejaswini Dhanesh Kumar

Student ID: X23288957

Programme: Cloud Computing **Year:** 2024-25

Module: Practicum Part 2

Lecturer: SAI EMANI

Submission Due Date: 12-9-2025

Project Title: **Enhancing Security of WordPress Containers on AWS: A Multitool Vulnerability Analysis.**

Word Count: 947 **Page Count:** 19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Tejaswini Dhanesh Kumar

Date: 12-9-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Tejaswini Dhanesh Kumar
Student ID: X23288957

1 Introduction

This manual outlines how to replicate the thesis deployment of a secure, containerized WordPress on Amazon Web Services (AWS).It's written for engineers and evaluators who need a fast, reproducible build—from image creation and registry scanning to load balancing and web-application security.Doing this will have two WordPress sites hosted on Docker behind an Application Load Balancer (ALB), include AWS WAF with managed and custom rules, and enable alerts for high-severity findings from Amazon ECR.The emphasis is on realistic, command-oriented directions that can be executed end-to-end within a controlled laboratory setting.All prerequisites, configuration options, and test for validation are present so that outcomes can be verified independently and stamped with confidence.

Prerequisites

1. Hardware Requirements

- AWS account with admin access for initial setup
- Two EC2 instances (t3.small works for demo)

2. Software Requirements

- **MAMP** (Apache + MySQL) with **phpMyAdmin**
- WordPress (local dev copy)
- Docker & Docker CLI
- AWS CLI (configured)
- Git

3. Credentials & Info

- AWS region (example:us-east-1)
- DB host, name, user, password

4. Architecture Overview

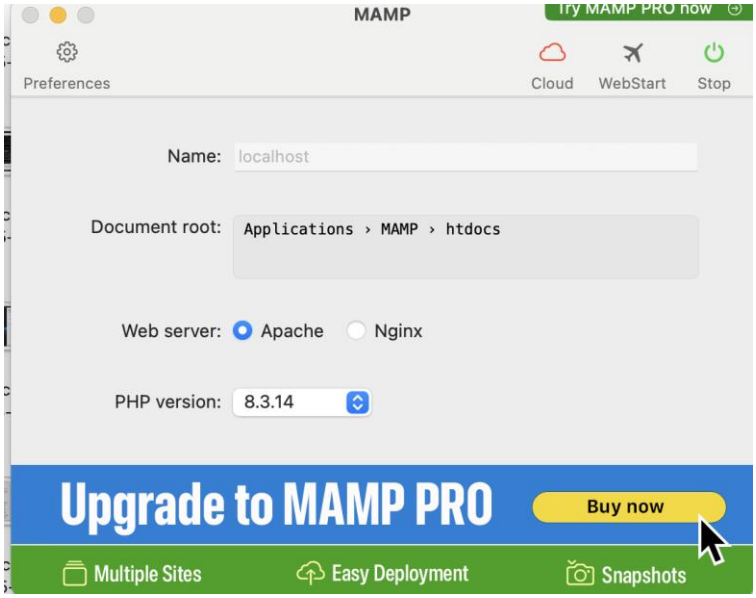
- Amazon ECR hosts the WordPress image (scan on push).
- EC2 (×2) run the WordPress container.
- MySQL on EC2 (self-managed) stores WordPress data (no RDS).

- Application Load Balancer (ALB) distributes traffic across EC2.
- AWS WAF attached to ALB with managed + custom rules.
- Amazon EventBridge watches ECR scan results → Amazon SNS email alerts (or AWS Lambda for auto actions → SNS).
- Amazon CloudWatch / AWS CloudTrail for logs and audit.

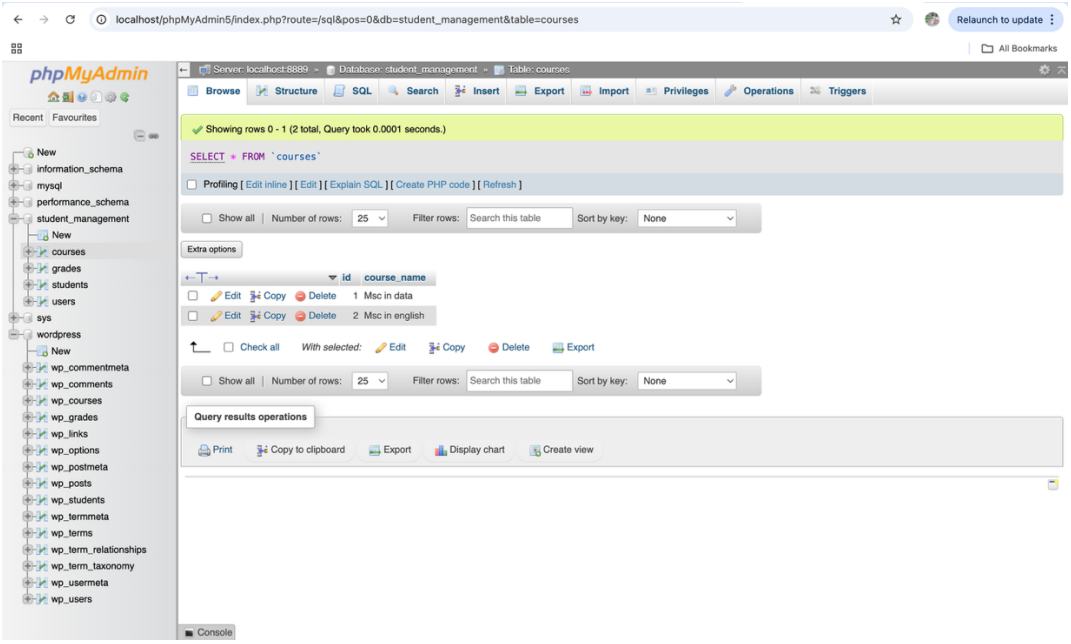
5. Local Development (MAMP + phpMyAdmin)

If you're replicating the local prototype used in the thesis before deployment to AWS, install MAMP and phpMyAdmin by following these steps:

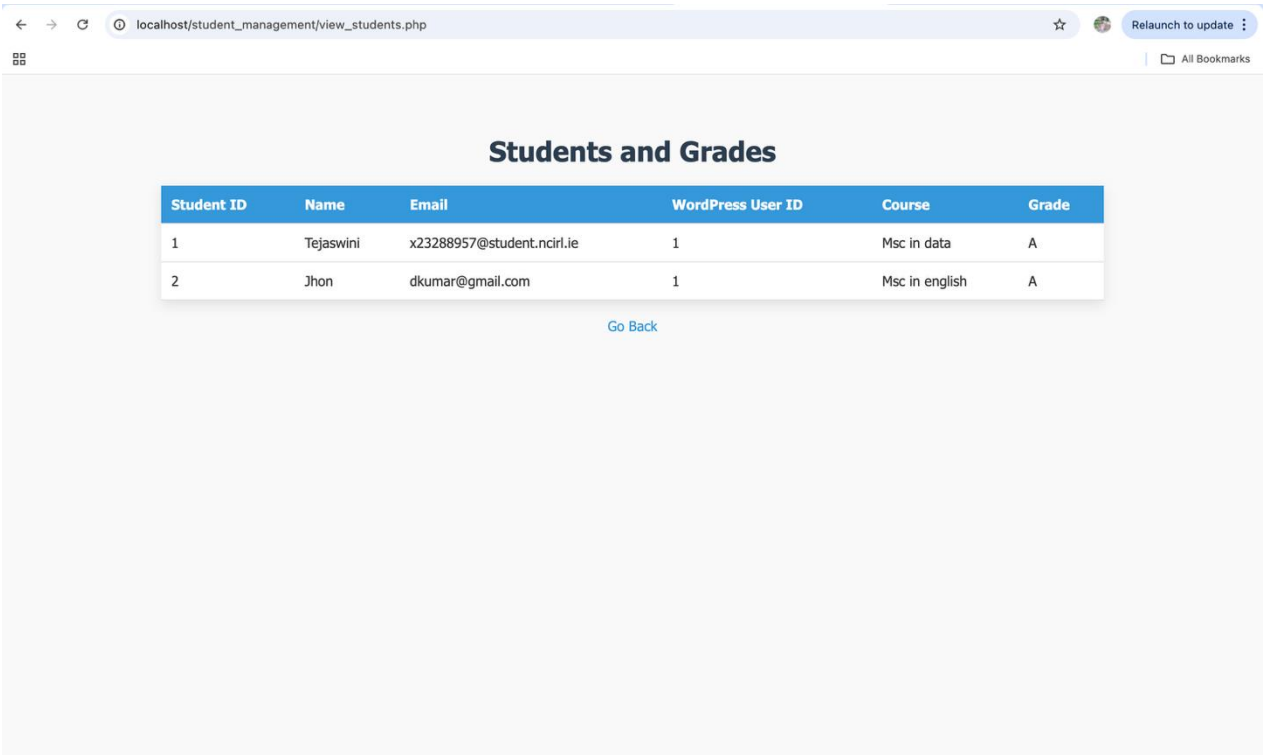
Open MAMP and run Apache and MySQL. Note the MySQL host/port (check in MAMP ▸ Preferences ▸ Ports — standard ports are 3306 or 8889).



Open phpMyAdmin at your local URL (e.g., http://localhost/phpMyAdmin5/index.php?route=/database/sql&db=wordpress) and create the wordpress database (or import your SQL dump, if it already exists). Use the SQL tab for schema/import operations.



Put your Student Management PHP files in MAMP's htdocs (e.g., ~/MAMP/htdocs/student-app/) and make sure they work locally (http://localhost/student_management/index.php).



Set up a local WordPress on MAMP (or utilize an existing one) and associate it with the same database (or alternate schema). For local, utilize DB values in wp-config.php depending on MAMP.

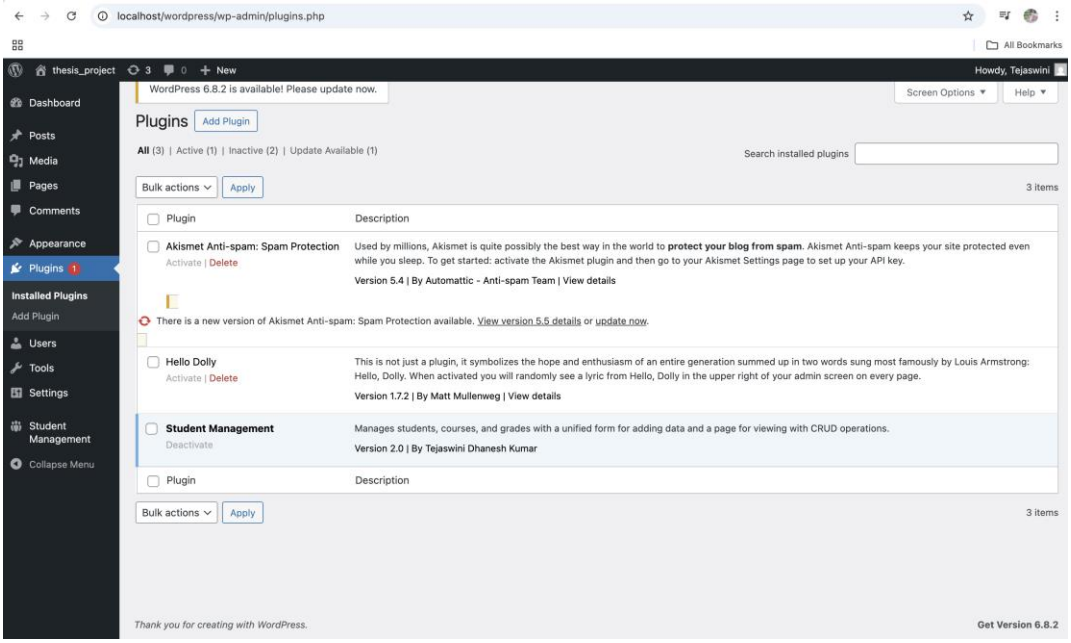
Do not set up phpMyAdmin to production. It is only for local development.

6. WordPress Plugin Integration (connecting the local app)

To make your locally running Student Management features available in WordPress pages, use a small custom plugin. This avoids brittle iframes and lets you reuse WordPress auth, CSRF protection (nonces), and $wpdb securely.

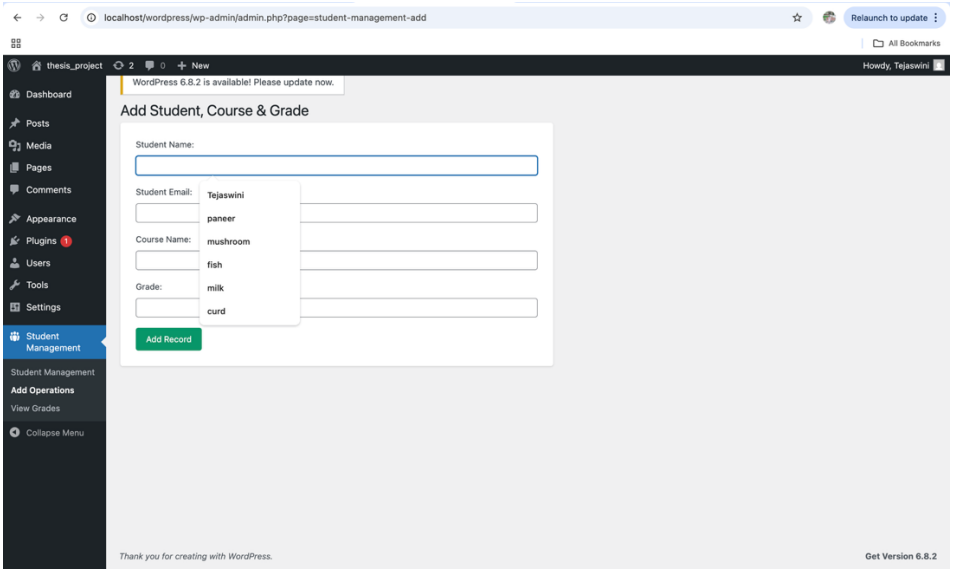
Steps

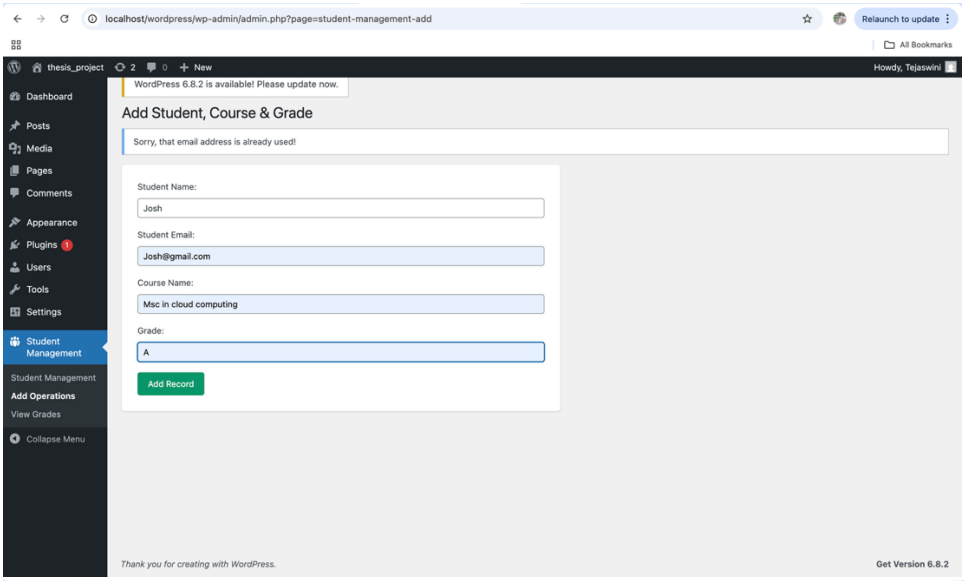
- Create folder: wp-content/plugins/student-management.

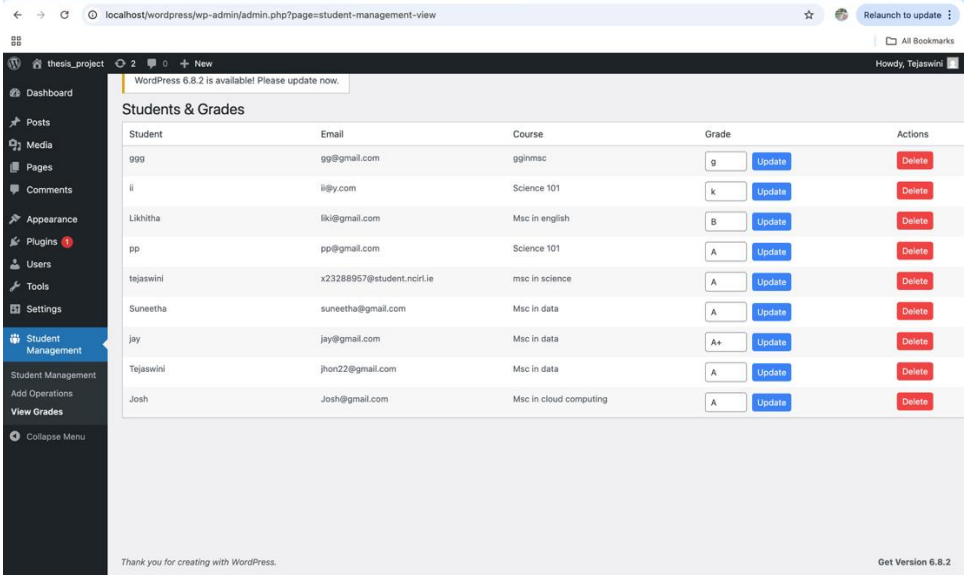


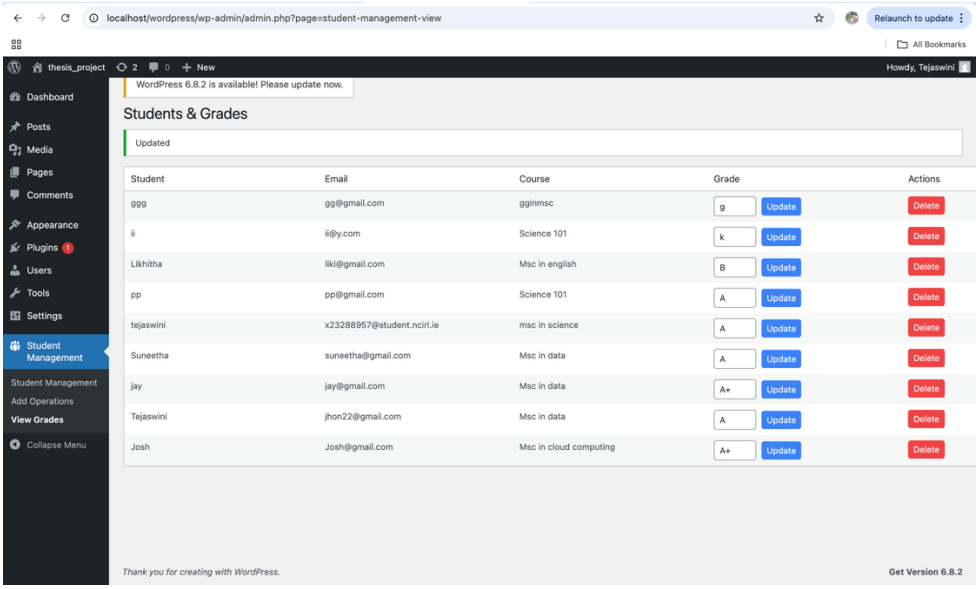
- Add file: wp-content/plugins/student-management/student-management.php with skeleton below.

- Activate the plugin in Plugins ▸ Installed Plugins.


localhost/wordpress/wp-admin/admin.php?page=student-management-add
Relaunch to update
thesis_project
New
Howdy, Tejaswini
WordPress 6.8.2 is available! Please update now.
Add Student, Course & Grade
Student Name:
Student Email:
Tejaswini
paneer
Course Name:
mushroom
fish
Grade:
milk
curd
Add Record
Dashboard
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Student Management
Student Management
Add Operations
View Grades
Collapse Menu
Thank you for creating with WordPress.
Get Version 6.8.2


localhost/wordpress/wp-admin/admin.php?page=student-management-add
Relaunch to update
thesis_project
New
Howdy, Tejaswini
WordPress 6.8.2 is available! Please update now.
Add Student, Course & Grade
Sorry, that email address is already used!
Student Name:
Josh
Student Email:
Josh@gmail.com
Course Name:
Msc in cloud computing
Grade:
A
Add Record
Dashboard
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Student Management
Student Management
Add Operations
View Grades
Collapse Menu
Thank you for creating with WordPress.
Get Version 6.8.2

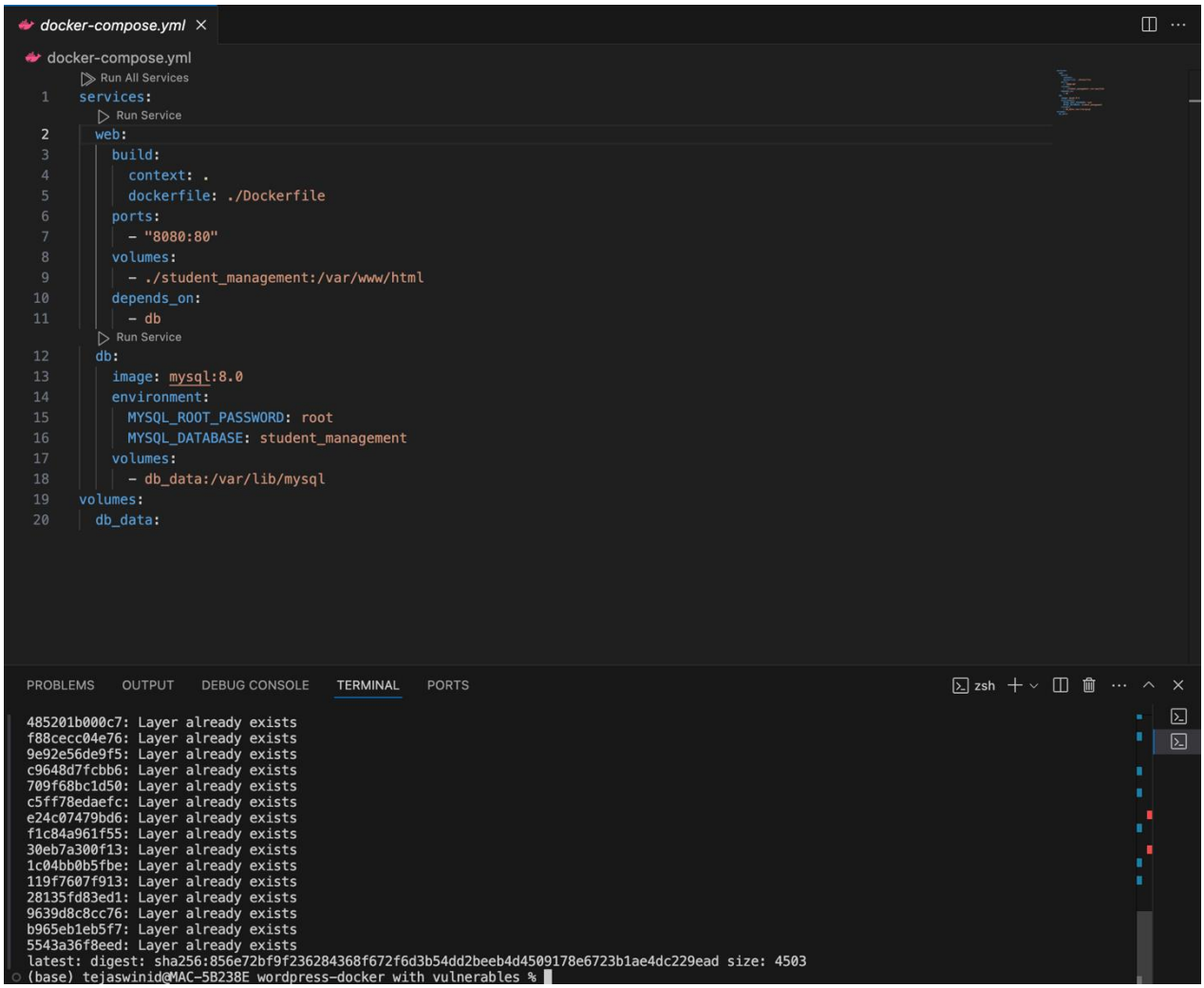

localhost/wordpress/wp-admin/admin.php?page=student-management-view
Relaunch to update
thesis_project
New
Howdy, Tejaswini
WordPress 6.8.2 is available! Please update now.
Students & Grades
Student
Email
Course
Grade
Actions
ggg
gg@gmail.com
gginmsc
g
Update
Delete
ii
ii@y.com
Science 101
k
Update
Delete
Likhitha
liki@gmail.com
Msc in english
B
Update
Delete
pp
pp@gmail.com
Science 101
A
Update
Delete
tejaswini
x23288957@student.ncirl.ie
msc in science
A
Update
Delete
Suneetha
suneetha@gmail.com
Msc in data
A
Update
Delete
jay
jay@gmail.com
Msc in data
A+
Update
Delete
Tejaswini
jhon22@gmail.com
Msc in data
A
Update
Delete
Josh
Josh@gmail.com
Msc in cloud computing
A
Update
Delete
Dashboard
Posts
Media
Pages
Comments
Appearance
Plugins
Users
Tools
Settings
Student Management
Student Management
Add Operations
View Grades
Collapse Menu
Thank you for creating with WordPress.
Get Version 6.8.2

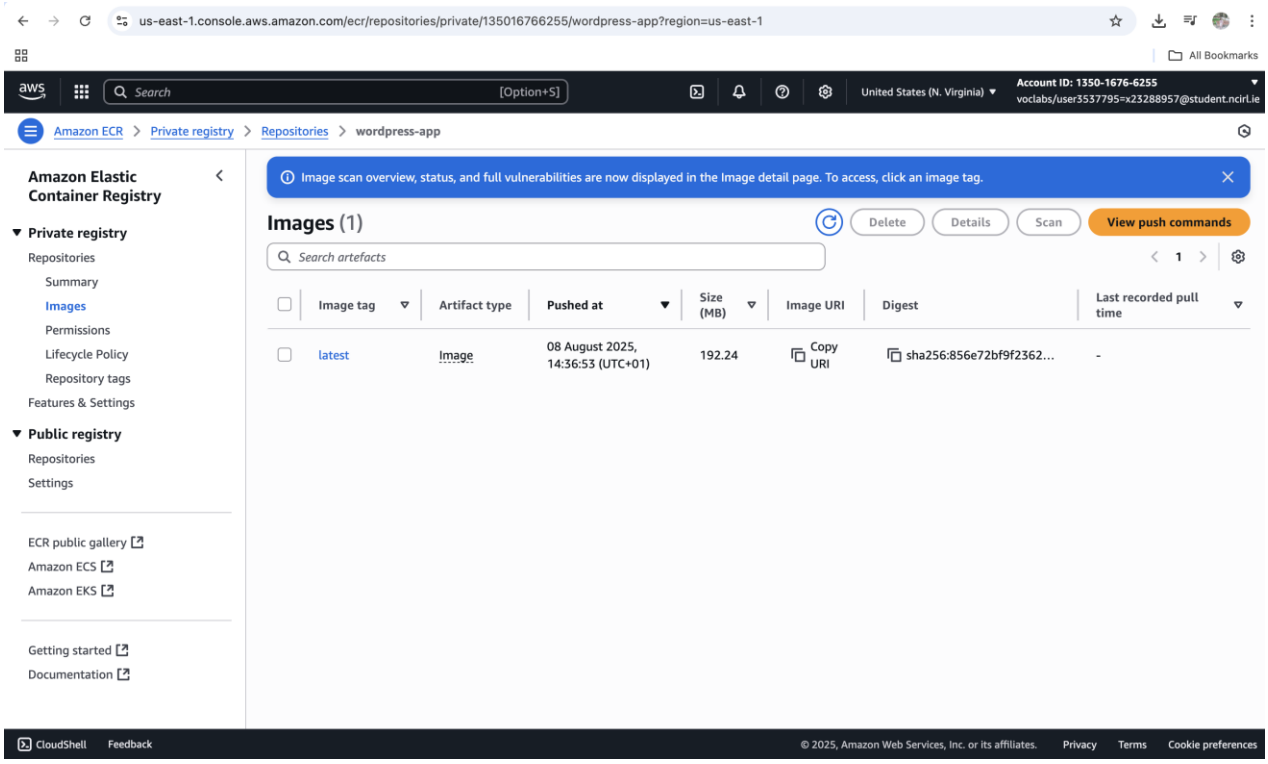


7. Container Build & Push to ECR:

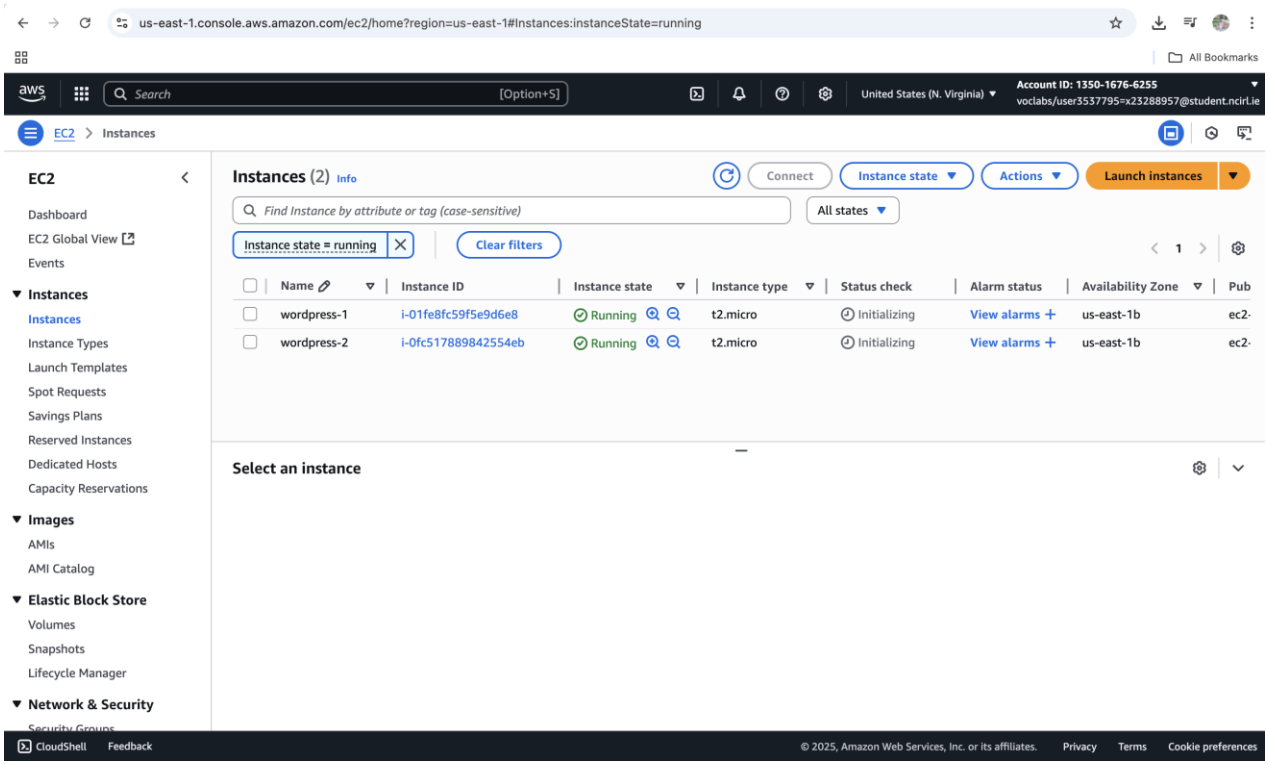
Build from your Docker file, then tag & push to ECR for a scan-on-push.

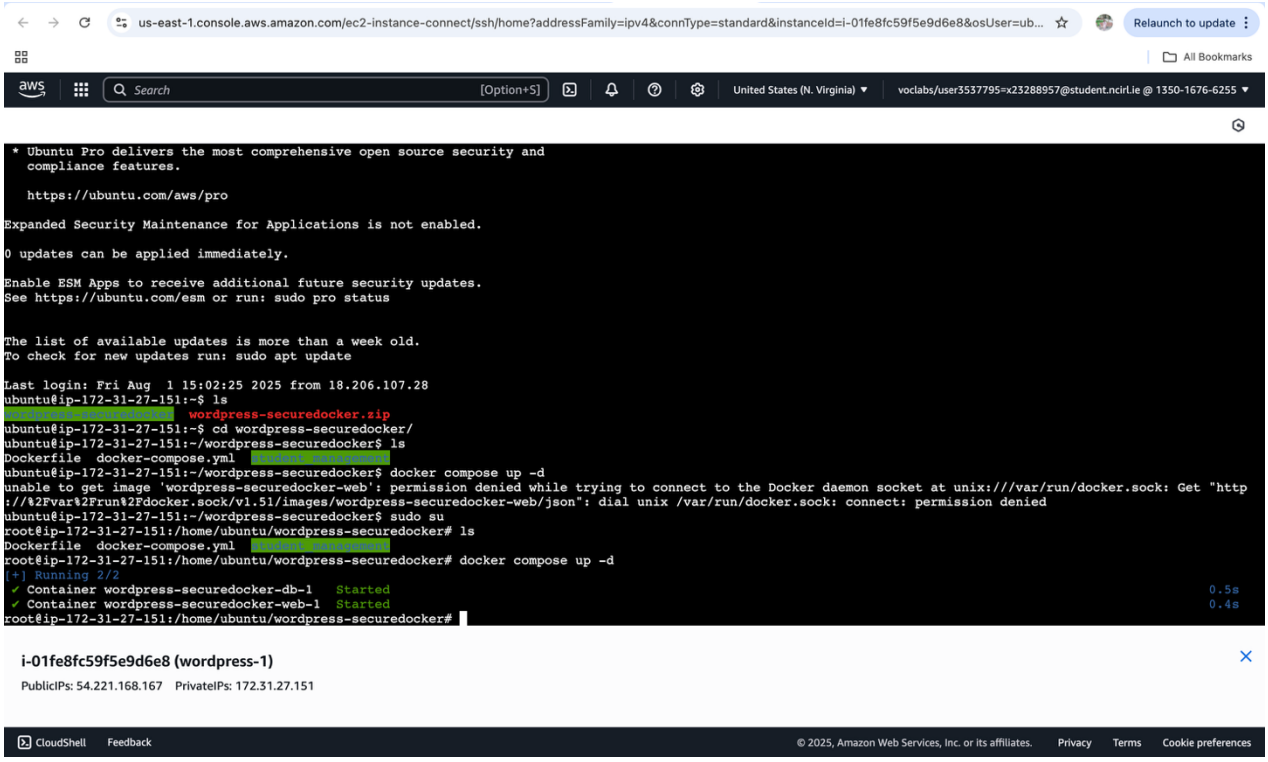
DOCKER_BUILDKIT=0 docker build -t wordpress-app:latest .
docker tag wordpress-app:latest 135016766255.dkr.ecr.us-east-1.amazonaws.com/wordpress-app:latest
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 135016766255.dkr.ecr.us-east-1.amazonaws.com
docker push 135016766255.dkr.ecr.us-east-1.amazonaws.com/wordpress-app:latest

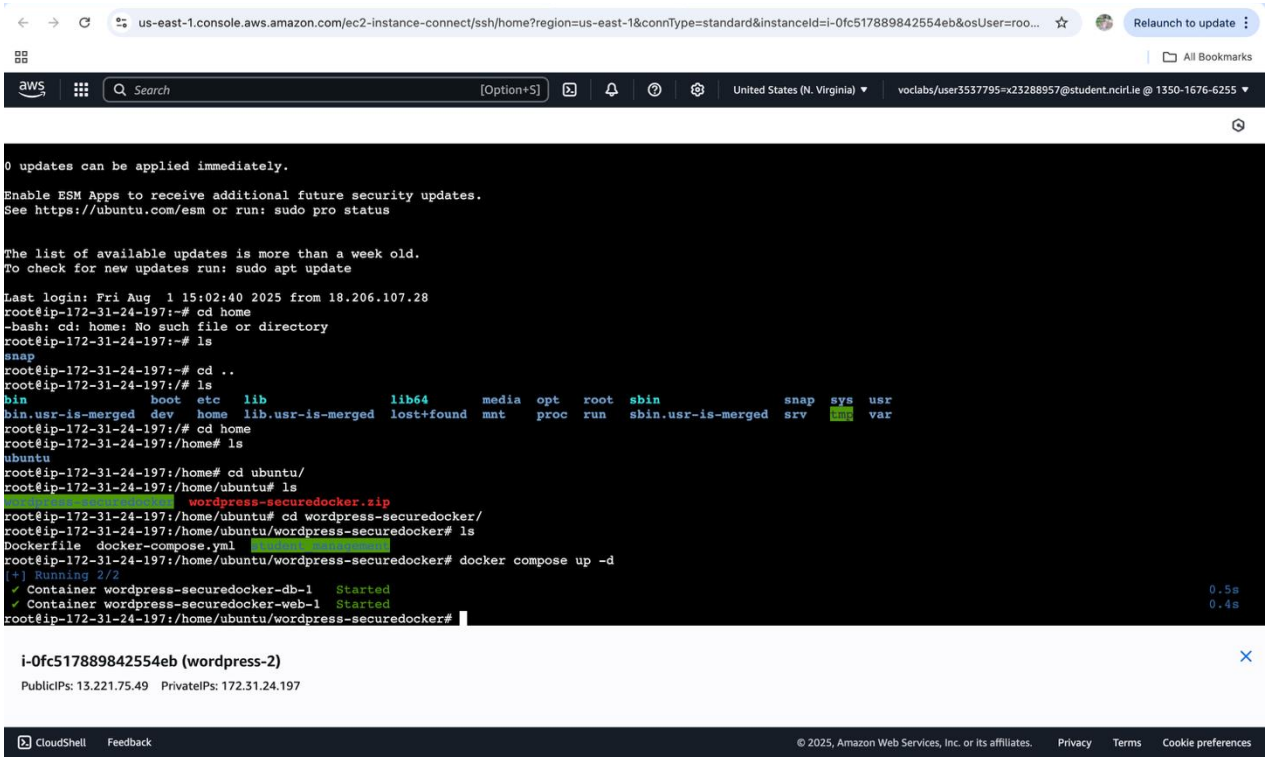

docker-compose.yml
docker-compose.yml
Run All Services
services:
Run Service
web:
build:
context: .
dockerfile: ./Dockerfile
ports:
- "8080:80"
volumes:
- ./student_management:/var/www/html
depends_on:
- db
Run Service
db:
image: mysql:8.0
environment:
MYSQL_ROOT_PASSWORD: root
MYSQL_DATABASE: student_management
volumes:
- db_data:/var/lib/mysql
volumes:
db_data:
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
zsh
485201b000c7: Layer already exists
f88cecc04e76: Layer already exists
9e92e56de9f5: Layer already exists
c9648d7fcbb6: Layer already exists
709f68bc1d50: Layer already exists
c5ff78edaefc: Layer already exists
e24c07479bd6: Layer already exists
f1c84a961f55: Layer already exists
30eb7a300f13: Layer already exists
1c04bb0b5fbe: Layer already exists
119f7607f913: Layer already exists
28135fd83ed1: Layer already exists
9639d8c8cc76: Layer already exists
b965eb1eb5f7: Layer already exists
5543a36f8eed: Layer already exists
latest: digest: sha256:856e72bf9f236284368f672f6d3b54dd2beeb4d4509178e6723b1ae4dc229ead size: 4503
(base) tejaswinid@MAC-5B238E wordpress-docker with vulnerables %



I have created two EC2 one for the Deployment and Other for the load balancing:

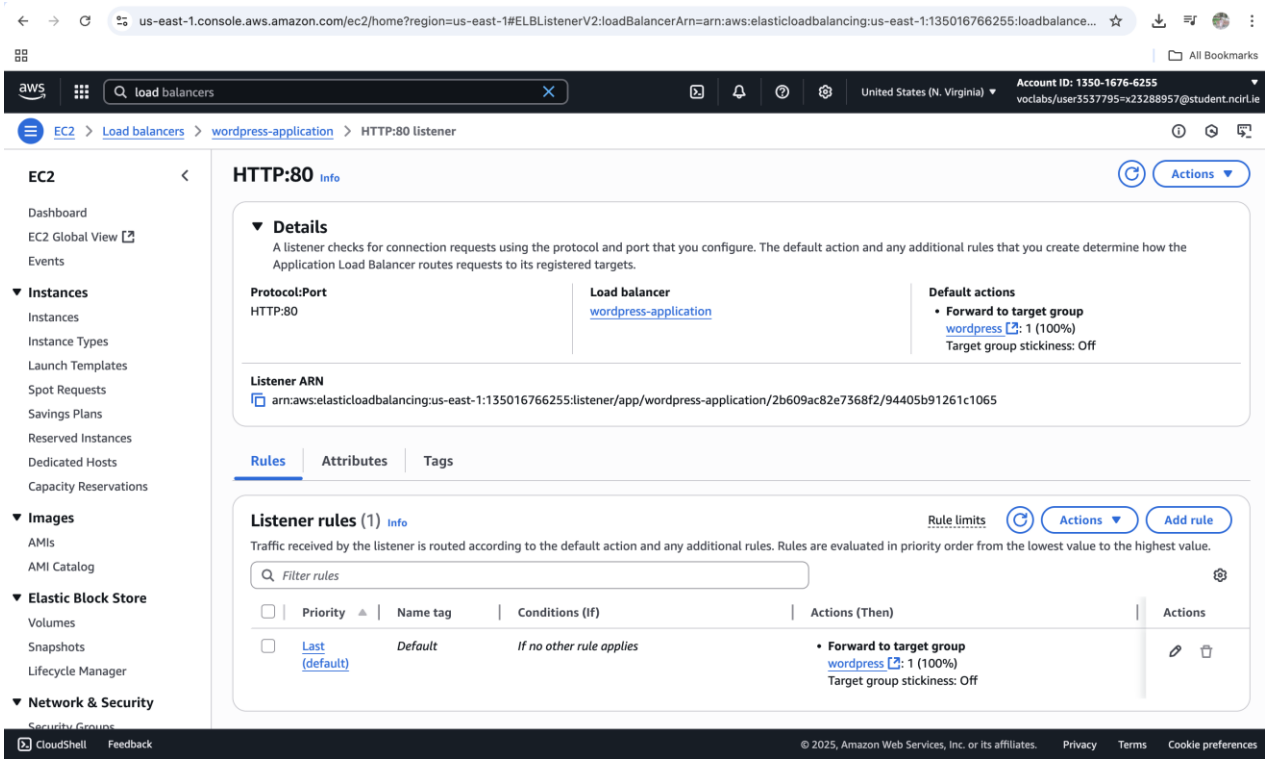


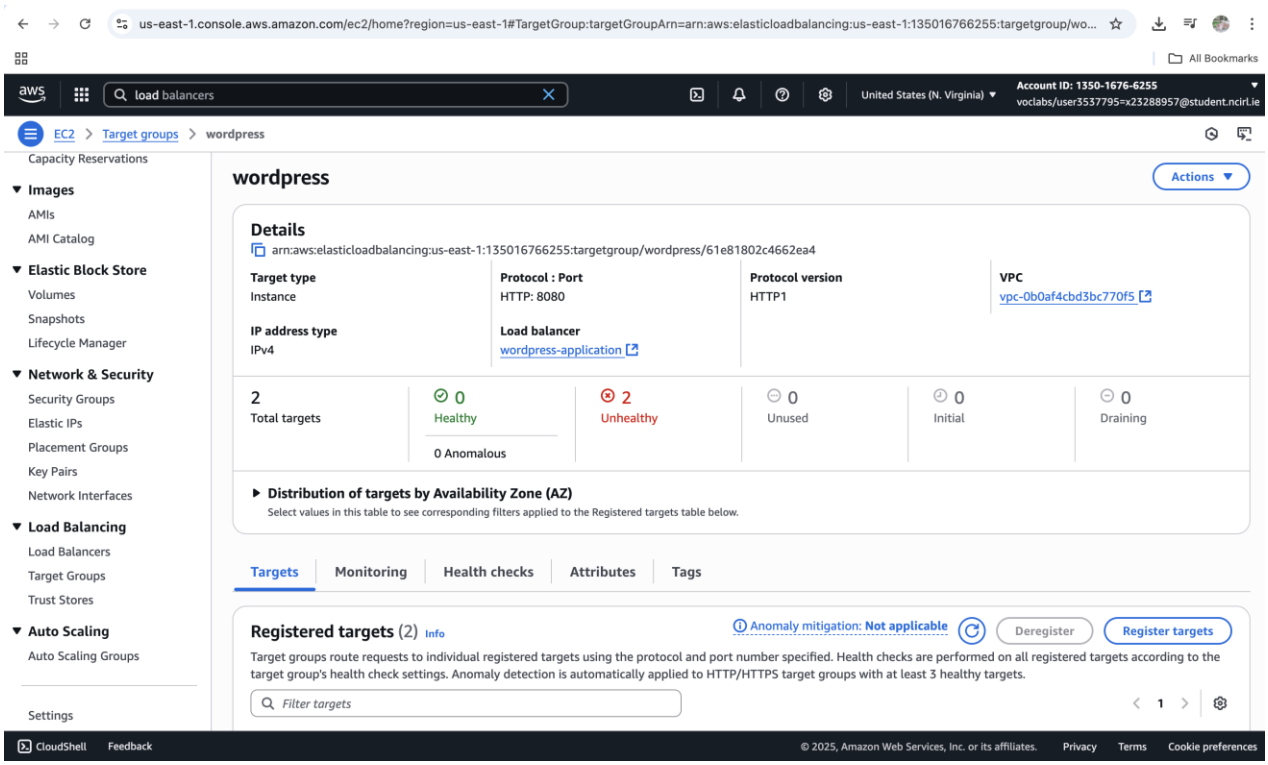




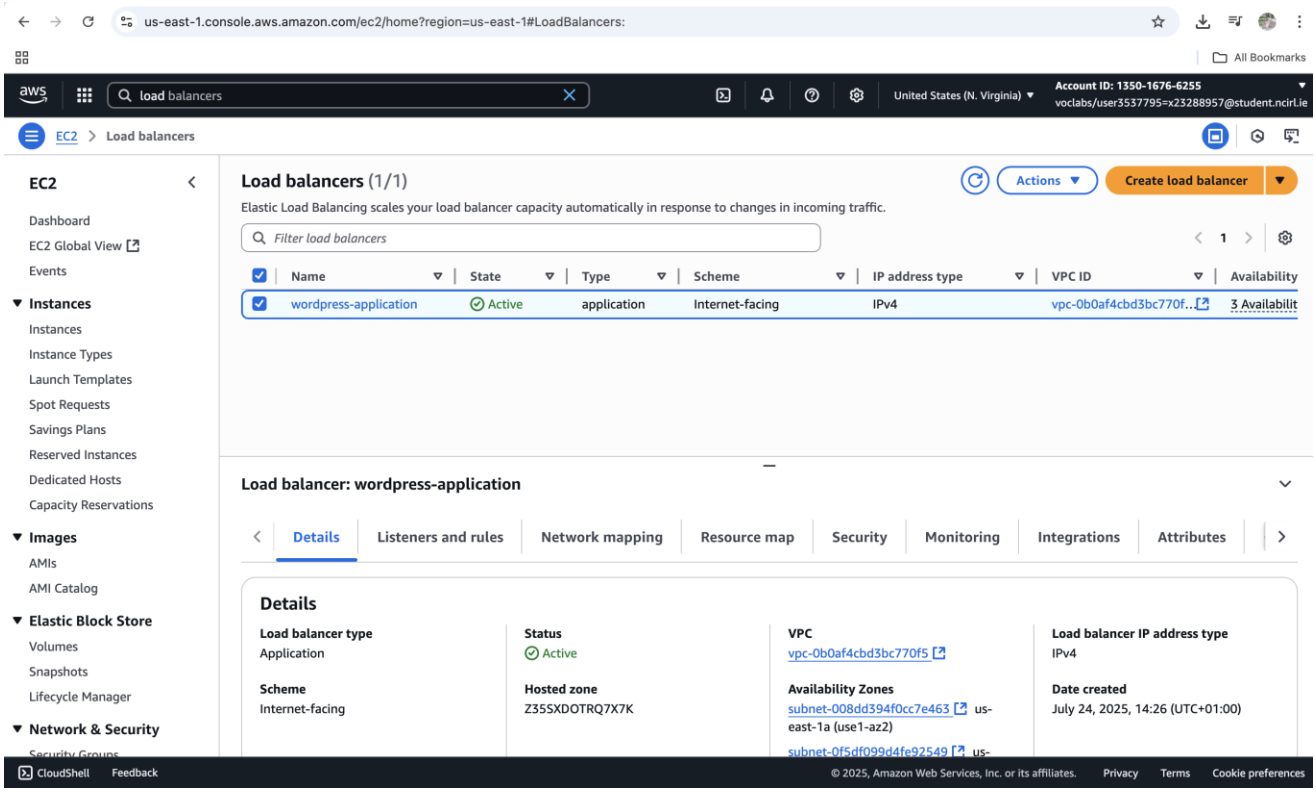
8. ALB & WAF:

- Create Target Group (instances, HTTP:80). Register both EC2s. Health check '/'



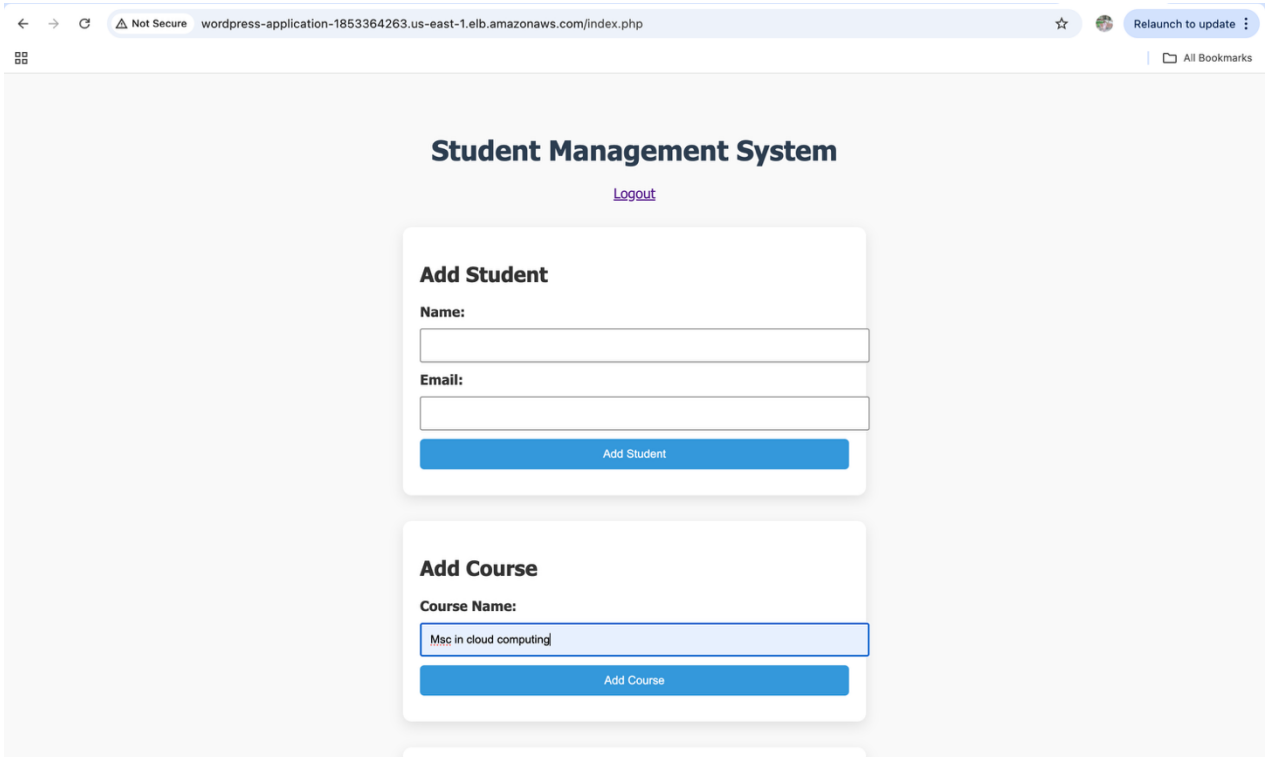


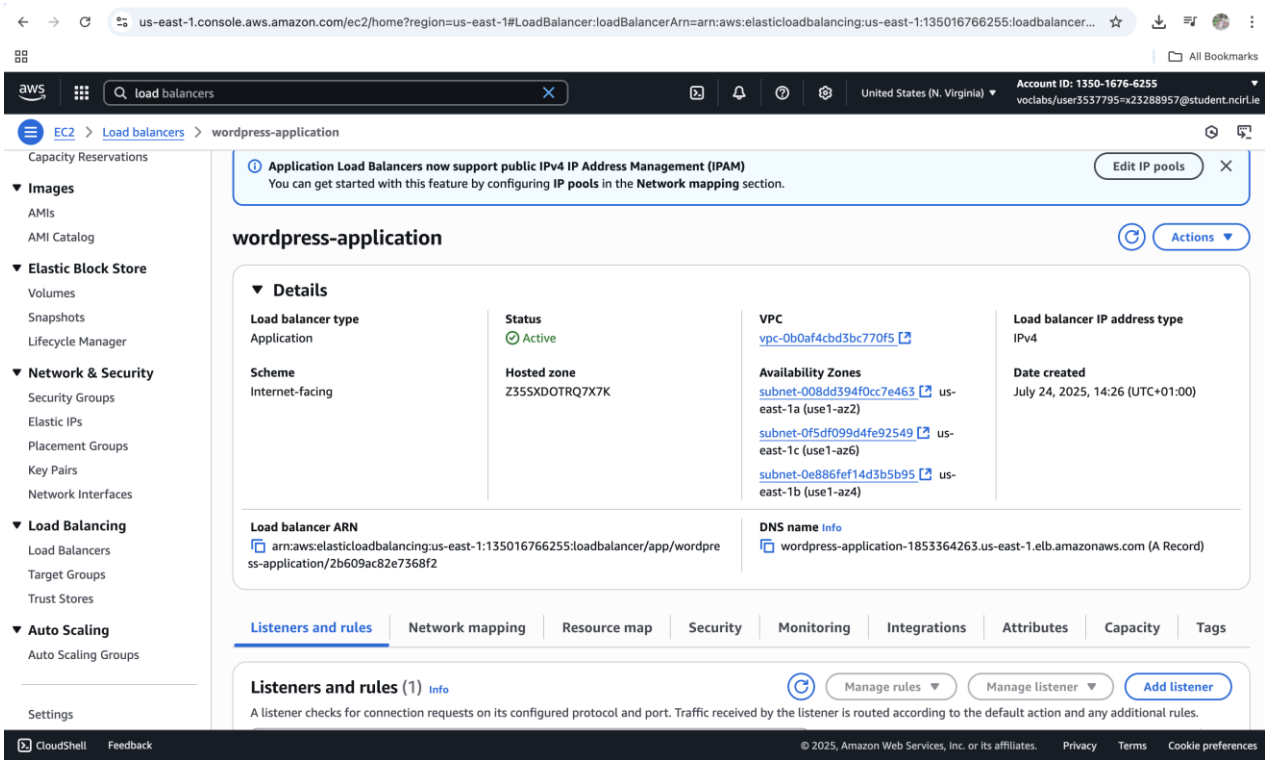
- Create ALB across public subnets; listeners 80→(optional redirect 443), 443→Target Group (ACM cert if using a domain)



- Create a Web ACL and associate it to the ALB

URL: wordpress-application-1853364263.us-east-1.elb.amazonaws.com

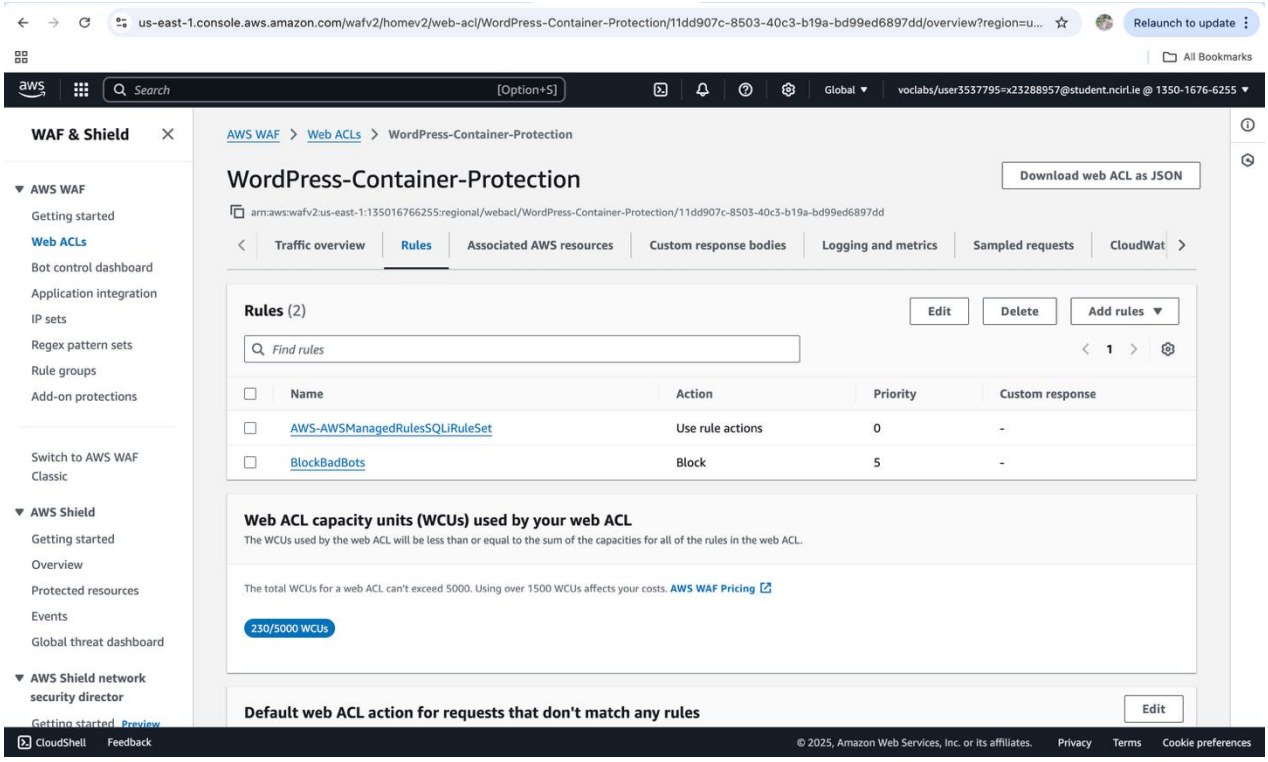




Enable AWSManagedRulesSQLiRuleSet + Common + KnownBadInputs. Turn on WAF logging to CloudWatch Logs for evidence.

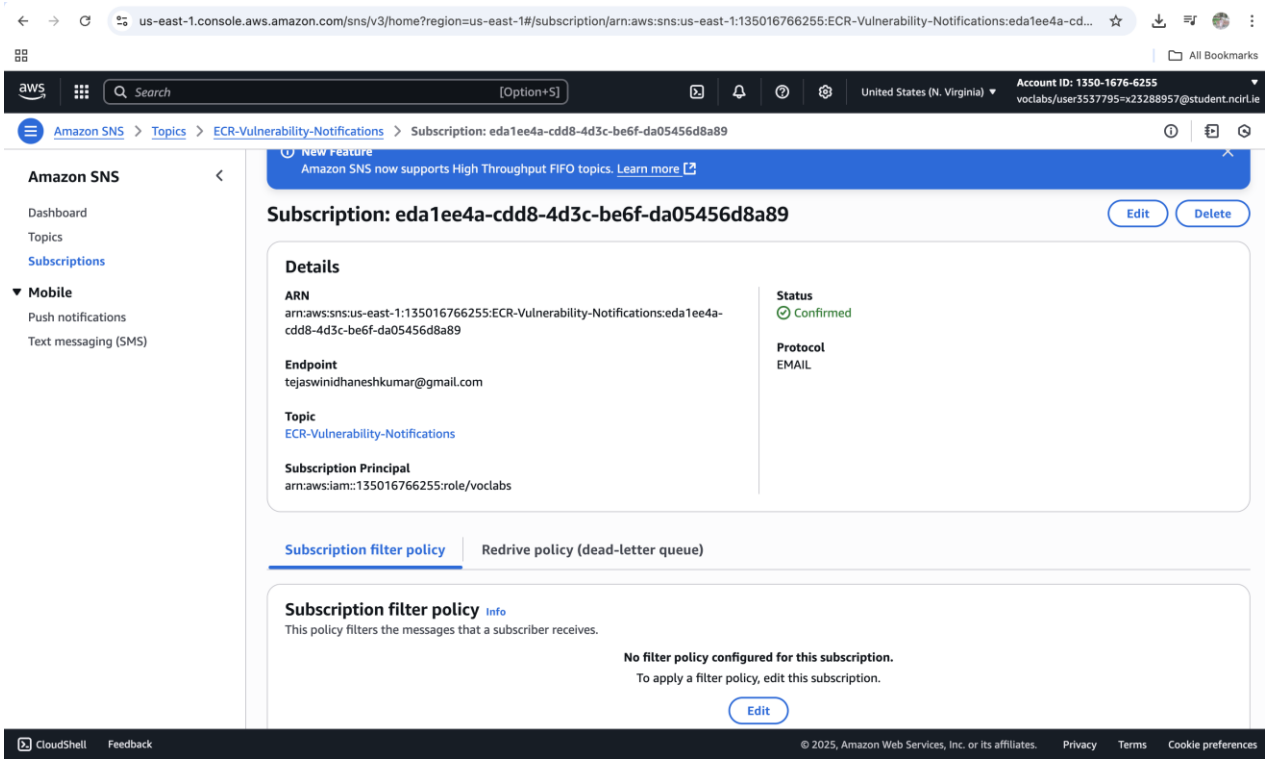
Custom digits□only param rule for 'id':

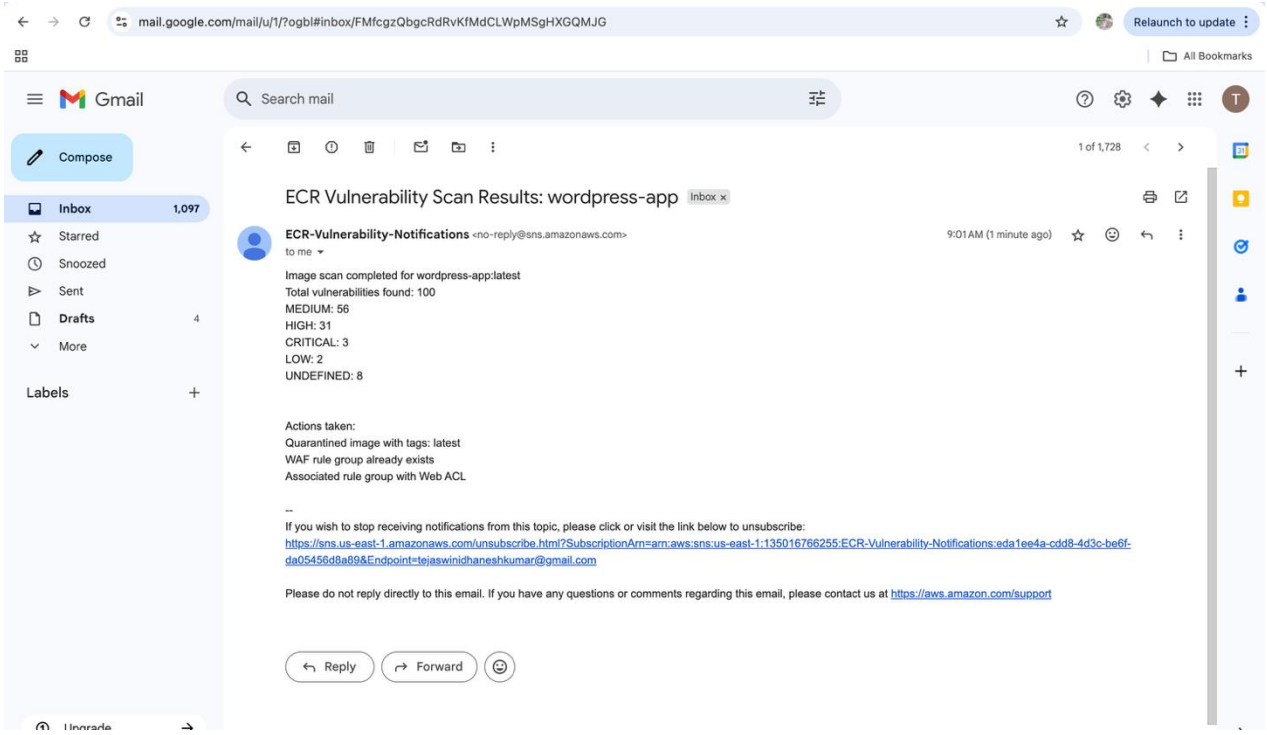
```
{
  "Name": "IdParamDigitsOnly",
  "Priority": 20,
  "Statement": {
    "RegexMatchStatement": {
      "RegexString": "^[0-9]{1,6}$",
      "FieldToMatch": { "SingleQueryArgument": { "Name": "id" } },
      "TextTransformations": [{"Priority":0,"Type":"NONE"}]
    }
  },
  "Action": {"Block": {}},
  "VisibilityConfig": {"SampledRequestsEnabled": true, "CloudWatchMetricsEnabled": true, "MetricName": "IdParamDigitsOnly"}
}
```



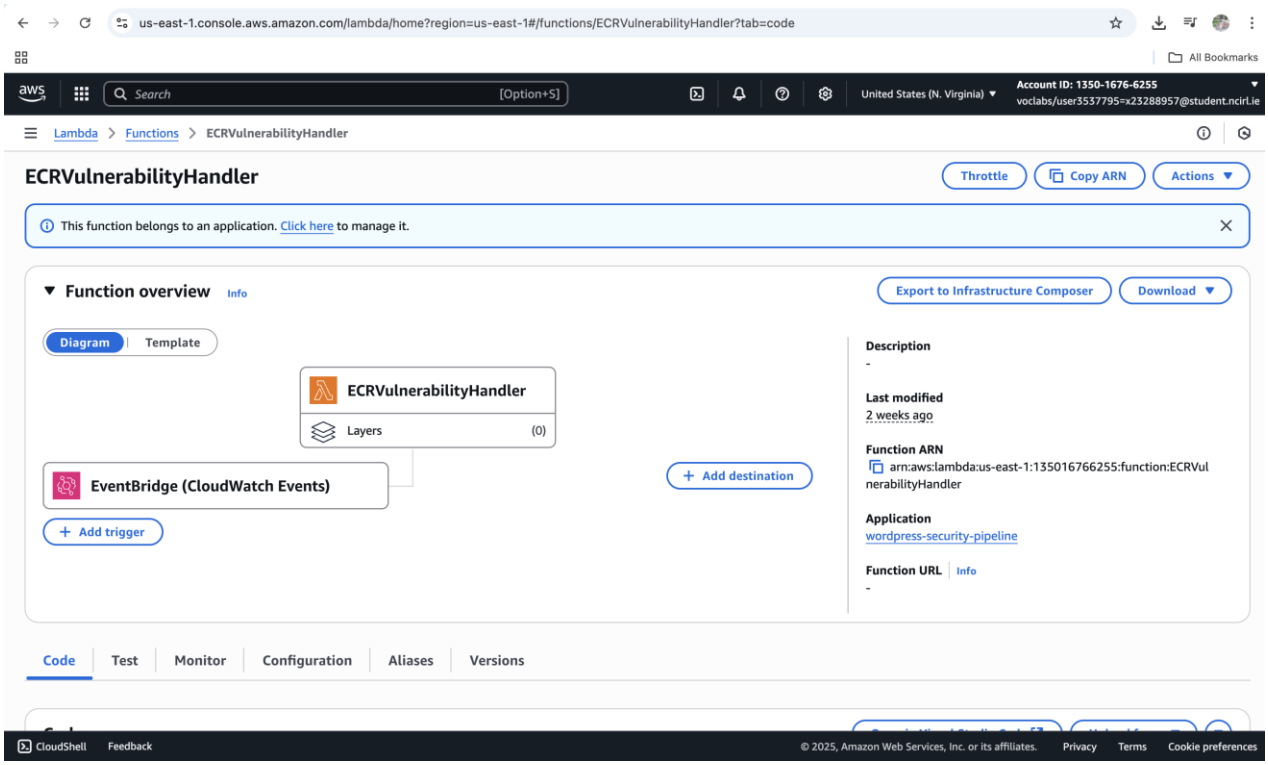
9. ECR Scan → EventBridge → Lambda → SNS (Email):

SNS topic: HighSeverityVulnAlerts (subscribe your email)





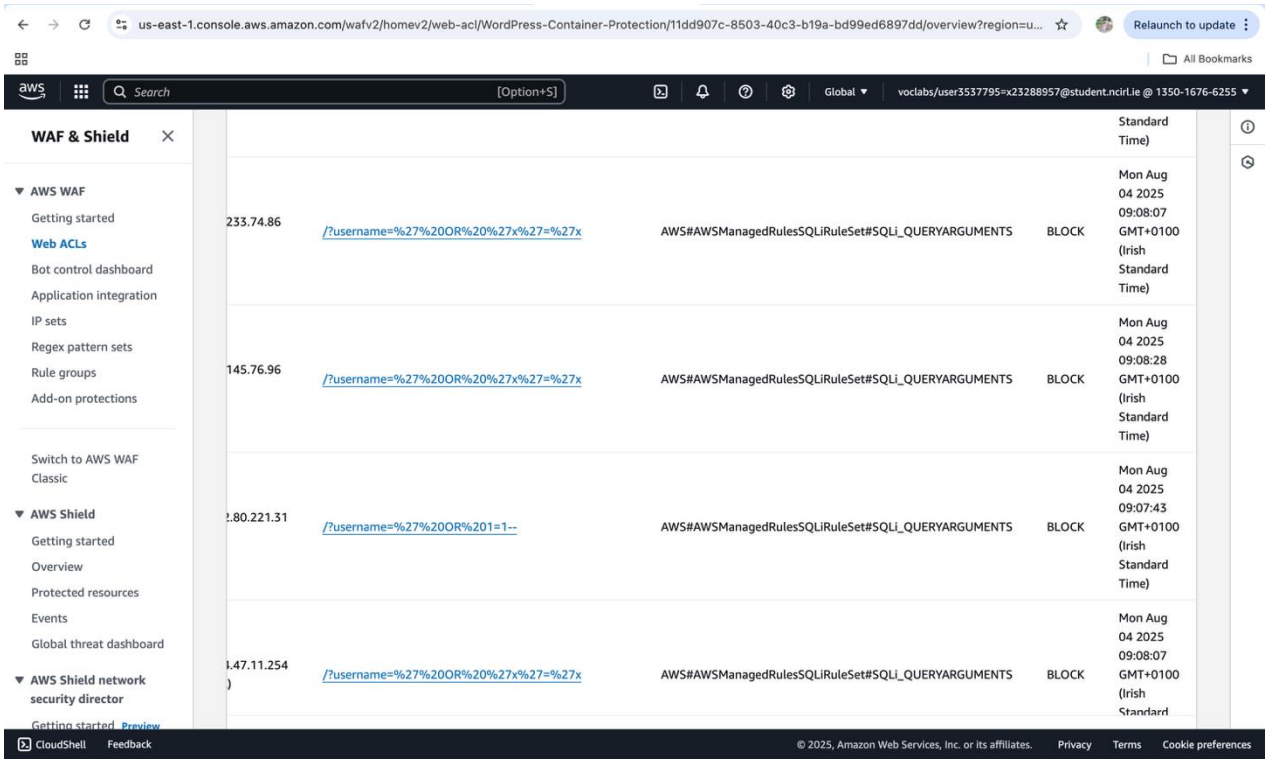
EventBridge rule: source=aws.ecr, detail- type='ECR Image Scan', filter CRITICAL>0 OR HIGH>0
Target: Lambda → publishes to SNS (and optionally updates WAF rules)

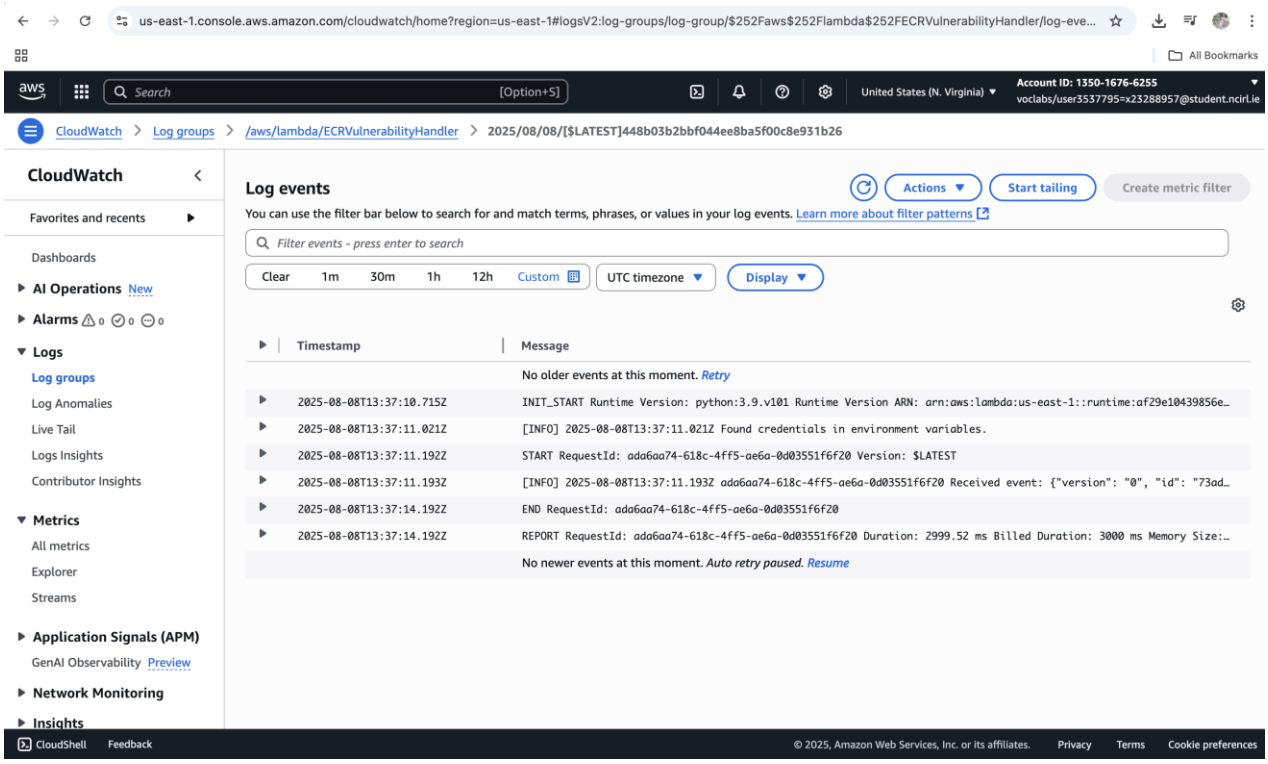


SQL Injection ,WAF Metrics And Cloud watch logs:

SQLi probes blocked (403); WAF logs show terminating rule

403 Forbidden




us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#logsV2:log-groups/log-group/$252Faws$252Flambda$252FECRVulnerabilityHandler/log-eve...
All Bookmarks
aws
Search
[Option+S]
United States (N. Virginia)
Account ID: 1350-1676-6255
voclabs/user3537795=x23288957@student.ncirl.ie
CloudWatch
Log groups
/aws/lambda/ECRVulnerabilityHandler
2025/08/08/[$LATEST]448b03b2bbf044ee8ba5f00c8e931b26
CloudWatch
Favorites and recents
Dashboards
AI Operations New
Alarms 0 0 0
Logs
Log groups
Log Anomalies
Live Tail
Logs Insights
Contributor Insights
Metrics
All metrics
Explorer
Streams
Application Signals (APM)
GenAI Observability Preview
Network Monitoring
Insights
Log events
Actions
Start tailing
Create metric filter
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns
Filter events - press enter to search
Clear
1m
30m
1h
12h
Custom
UTC timezone
Display
Timestamp
Message
No older events at this moment. Retry
2025-08-08T13:37:10.715Z
INIT_START Runtime Version: python:3.9.v101 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:af29e10439856e...
2025-08-08T13:37:11.021Z
[INFO] 2025-08-08T13:37:11.021Z Found credentials in environment variables.
2025-08-08T13:37:11.192Z
START RequestId: ada6aa74-618c-4ff5-ae6a-0d03551f6f20 Version: $LATEST
2025-08-08T13:37:11.193Z
[INFO] 2025-08-08T13:37:11.193Z ada6aa74-618c-4ff5-ae6a-0d03551f6f20 Received event: {"version": "0", "id": "73ad...
2025-08-08T13:37:14.192Z
END RequestId: ada6aa74-618c-4ff5-ae6a-0d03551f6f20
2025-08-08T13:37:14.192Z
REPORT RequestId: ada6aa74-618c-4ff5-ae6a-0d03551f6f20 Duration: 2999.52 ms Billed Duration: 3000 ms Memory Size:...
No newer events at this moment. Auto retry paused. Resume
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences