

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
.....
X23288957
Student ID:
Cloud Computing 2024-25
Programme: **Year:**
Practicum Part 2
Module:
SAI EMANI
Lecturer:
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.....
12-9-
Date: 2025.....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
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.	<input type="checkbox"/>

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

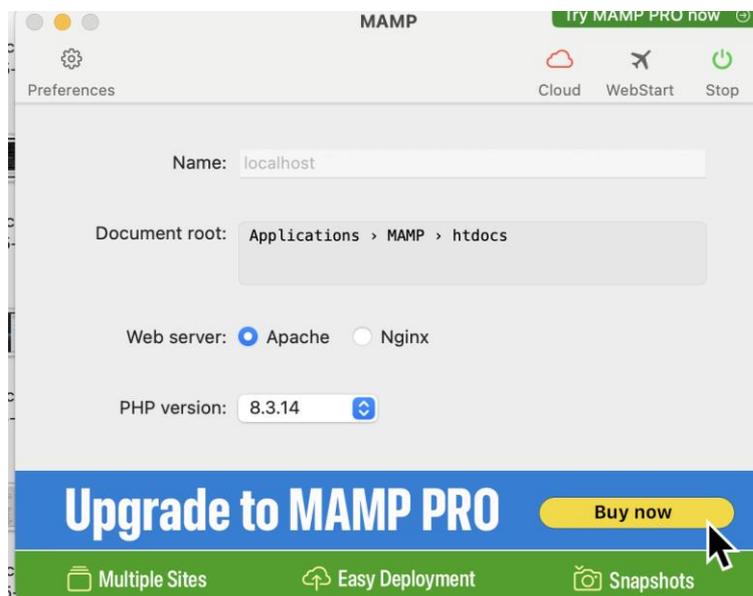
- Amazon ECR hosts the WordPress image (scan on push).
- EC2 (×2) run the WordPress container.
- MySQL on EC2 (self-managed) stores WordPress data (no RDS).

- 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.

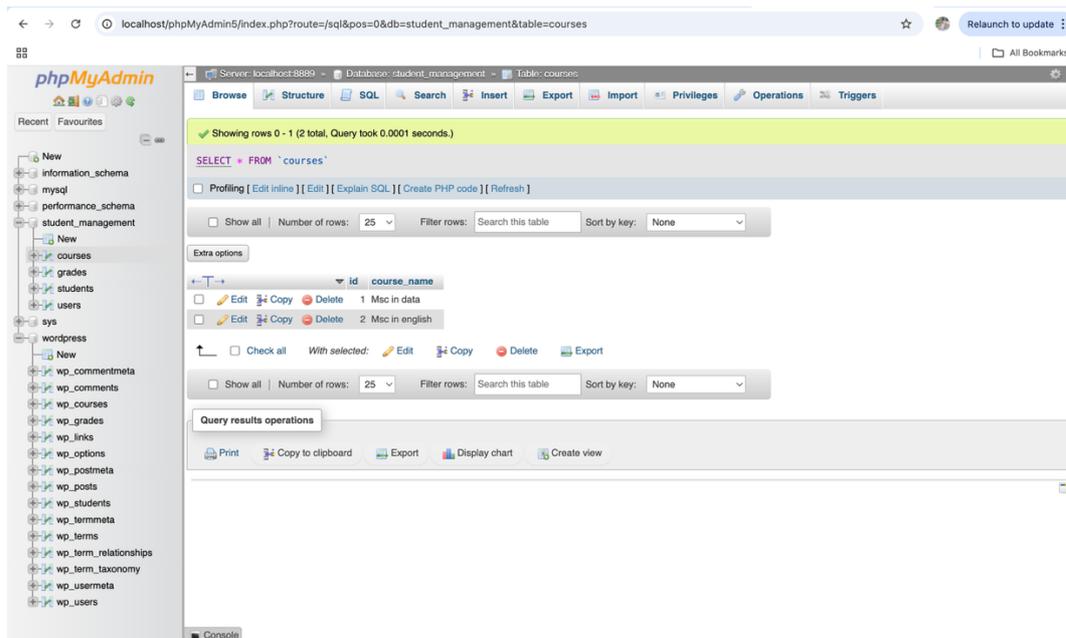
5. Local Development (MAMP + phpMyAdmin)

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

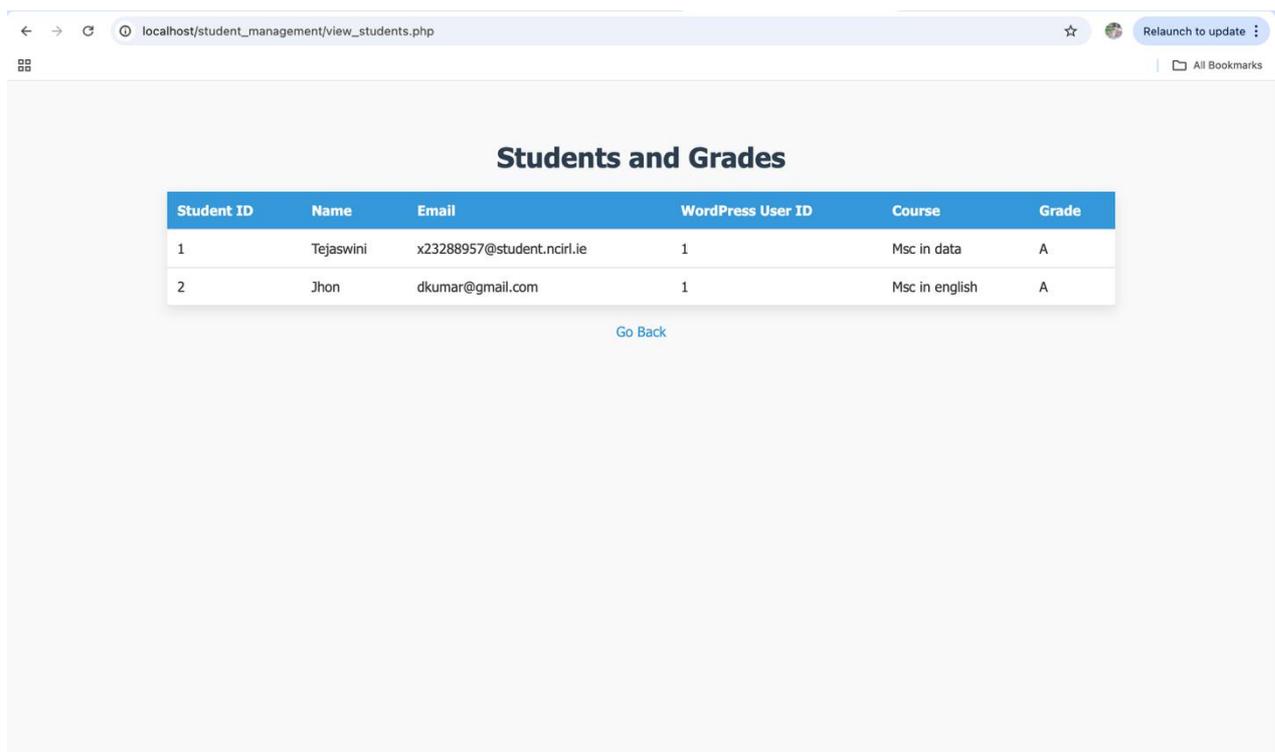
Open MAMP and run Apache and MySQL. Note the MySQL host/port (check in MAMP ► Preferences ► Ports — standard ports are 3306 or 8889).



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.

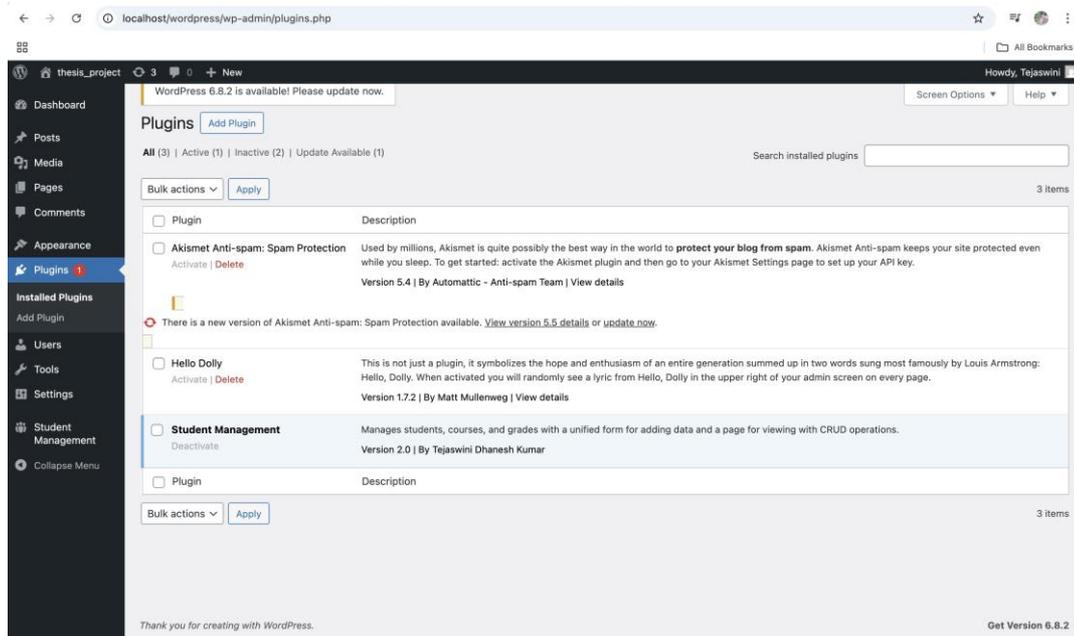
Do not set up phpMyAdmin to production. It is only for local development.

6. WordPress Plugin Integration (connecting the local app)

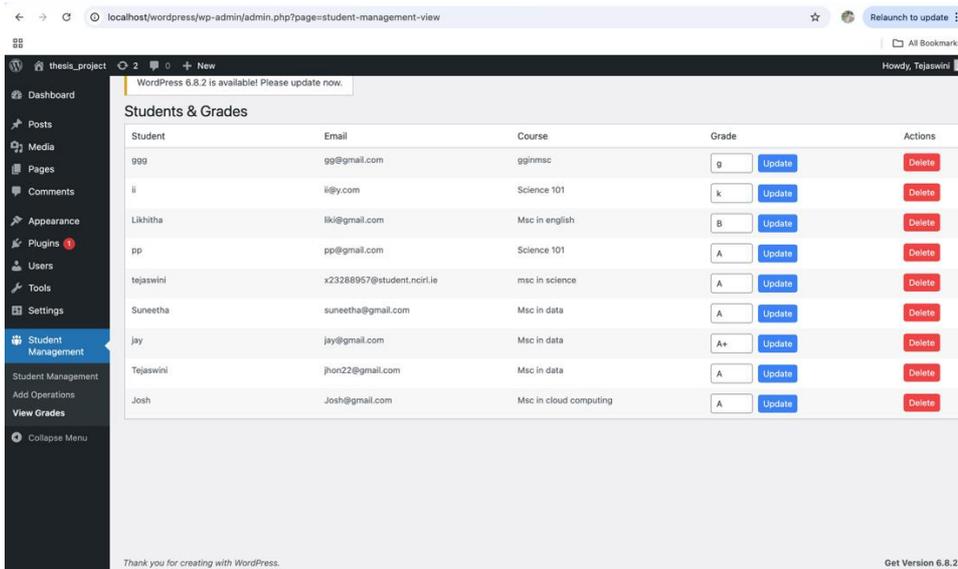
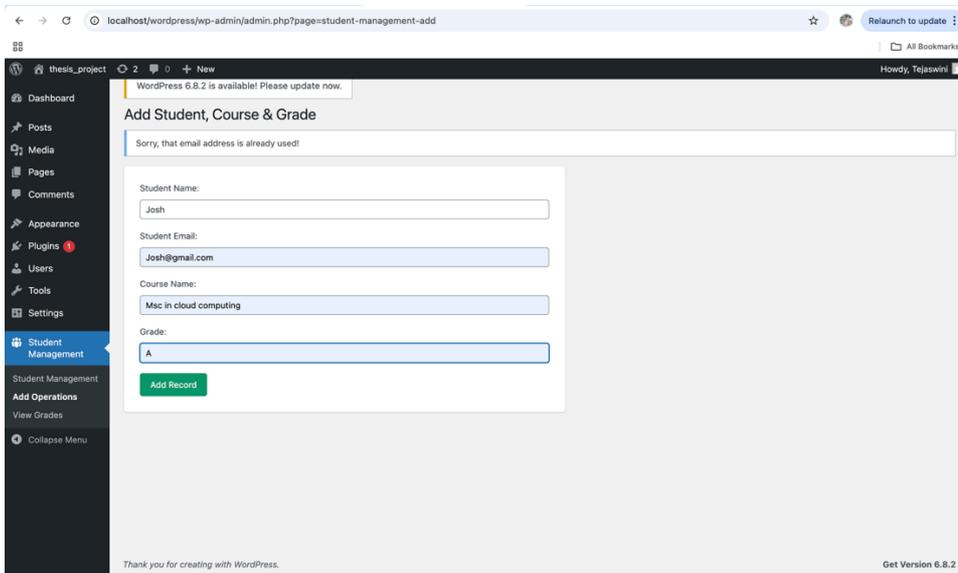
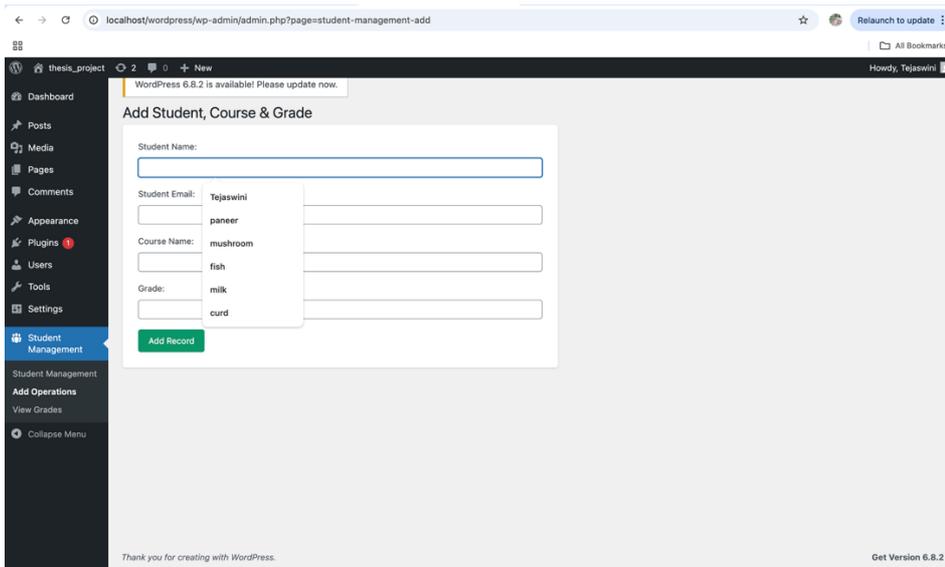
To make your locally running Student Management features available in WordPress pages, use a small custom plugin. This avoids brittle iframes and lets you reuse WordPress auth, CSRF protection (nonces), and \$wpdb securely.

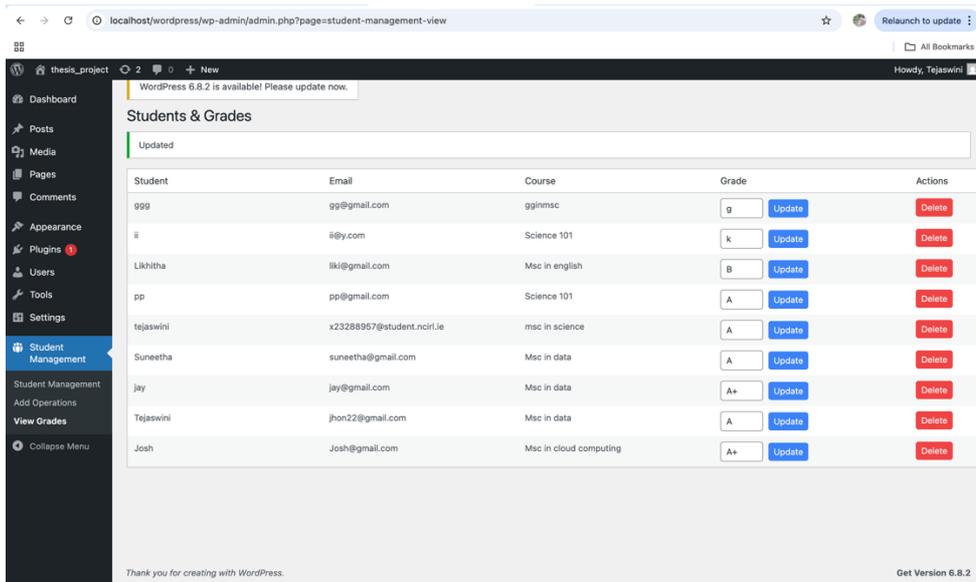
Steps

- 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.





7. Container Build & Push to ECR:

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

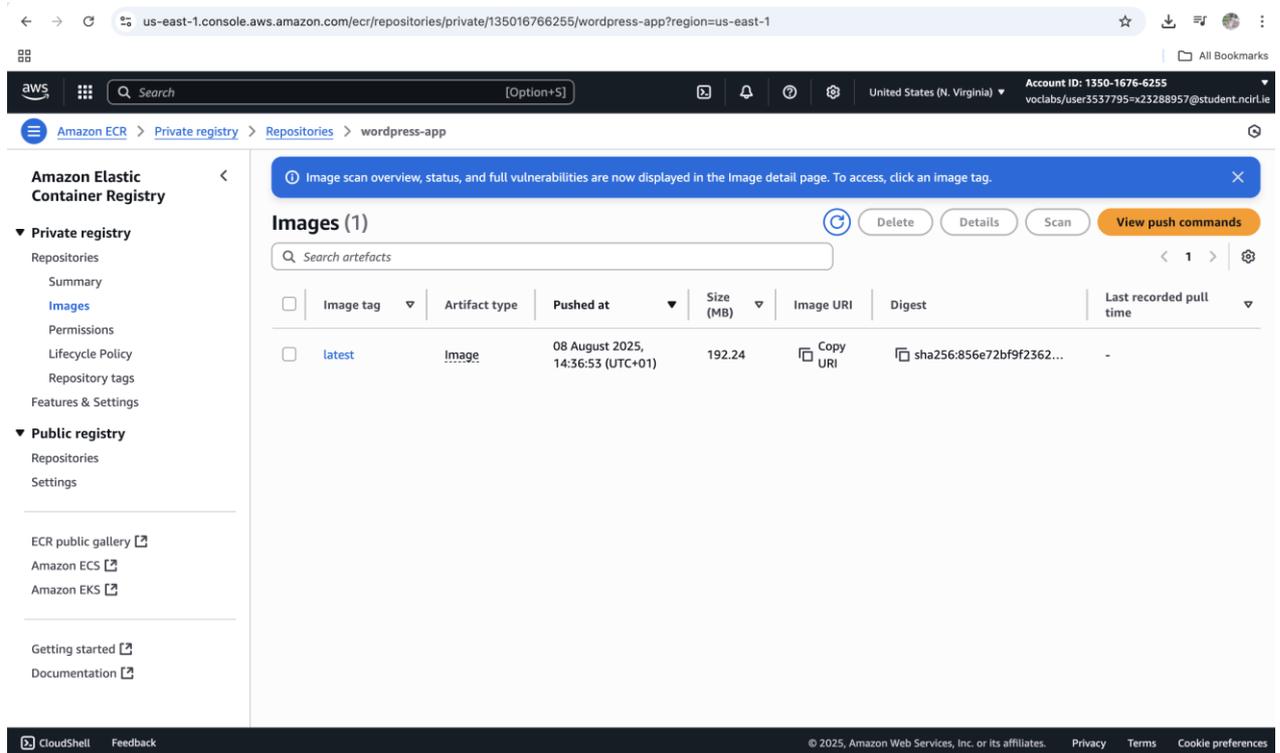
```
DOCKER_BUILDKIT=0 docker build -t wordpress-app:latest .
```

```
docker tag wordpress-app:latest 135016766255.dkr.ecr.us-east-1.amazonaws.com/wordpress-app:latest
```

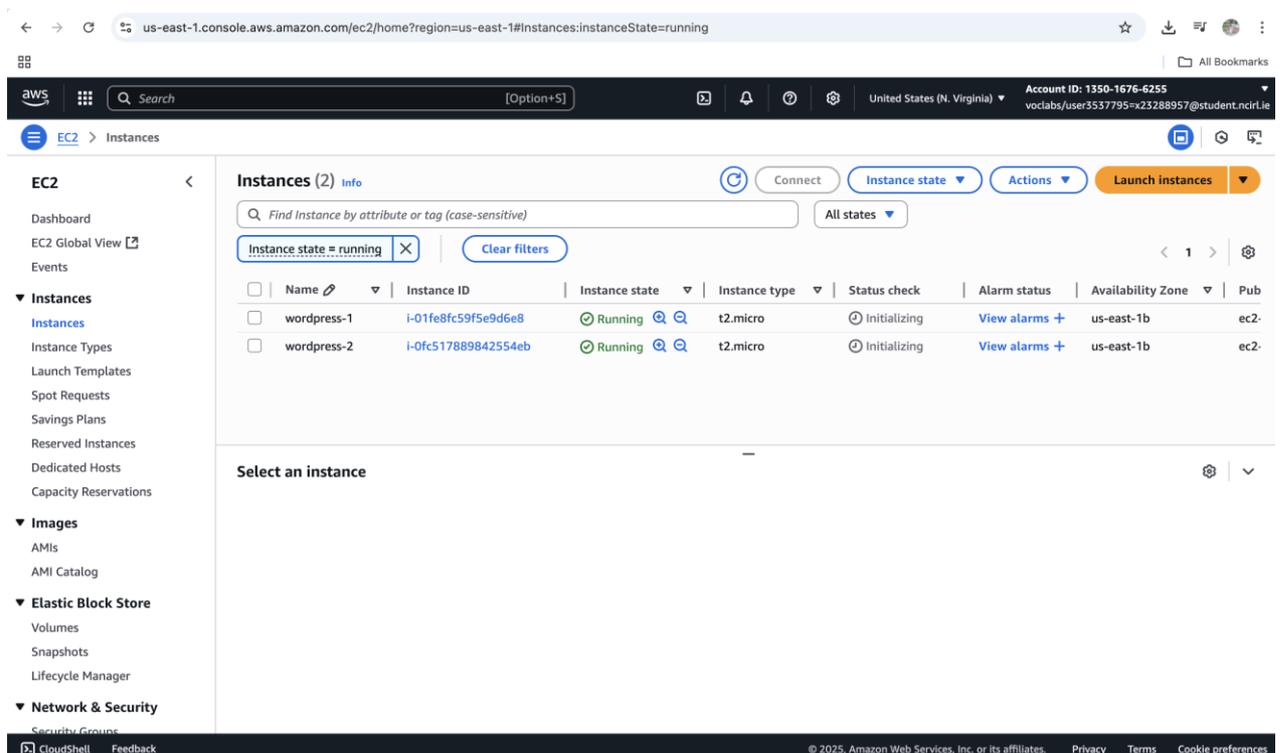
```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin
```

```
135016766255.dkr.ecr.us-east-1.amazonaws.com
```

```
docker push 135016766255.dkr.ecr.us-east-1.amazonaws.com/wordpress-app:latest
```

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



us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-01fe8fc59f5e9d6e8&osUser=ub...

Relaunch to update

Search [Option+S]

United States (N. Virginia) | voclabs/user3537795-x23288957@student.ncirlie @ 1350-1676-6255

```

* Ubuntu Pro delivers the most comprehensive open source security and
compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Fri Aug 1 15:02:25 2025 from 18.206.107.28
ubuntu@ip-172-31-27-151:~$ ls
wordpress-securedocker.zip
ubuntu@ip-172-31-27-151:~$ cd wordpress-securedocker/
ubuntu@ip-172-31-27-151:~/wordpress-securedocker$ ls
Dockerfile  docker-compose.yml
ubuntu@ip-172-31-27-151:~/wordpress-securedocker$ docker compose up -d
unable to get image 'wordpress-securedocker-web': permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://172.31.27.151:2376/v1.51/images/wordpress-securedocker-web/json": dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-27-151:~/wordpress-securedocker$ sudo su
root@ip-172-31-27-151:/home/ubuntu/wordpress-securedocker# ls
Dockerfile  docker-compose.yml
root@ip-172-31-27-151:/home/ubuntu/wordpress-securedocker# docker compose up -d
[+] Running 2/2
✔ Container wordpress-securedocker-db-1 Started      0.5s
✔ Container wordpress-securedocker-web-1 Started     0.4s
root@ip-172-31-27-151:/home/ubuntu/wordpress-securedocker#

```

i-01fe8fc59f5e9d6e8 (wordpress-1)

PublicIPs: 54.221.168.167 PrivateIPs: 172.31.27.151

CloudShell Feedback | © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=us-east-1&connType=standard&instanceId=i-0fc517889842554eb&osUser=roo...

Relaunch to update

Search [Option+S]

United States (N. Virginia) | voclabs/user3537795-x23288957@student.ncirlie @ 1350-1676-6255

```

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Fri Aug 1 15:02:40 2025 from 18.206.107.28
root@ip-172-31-24-197:~# cd home
-bash: cd: home: No such file or directory
root@ip-172-31-24-197:~# ls
snap
root@ip-172-31-24-197:~# cd ..
root@ip-172-31-24-197:~# ls
bin  boot  etc  lib  lib64  media  opt  root  sbin  snap  sys  usr
bin.usr-is-merged  dev  home  lib.usr-is-merged  lost+found  mnt  proc  run  sbin.usr-is-merged  srv  var
root@ip-172-31-24-197:~# cd home
root@ip-172-31-24-197:/home# ls
ubuntu
root@ip-172-31-24-197:/home# cd ubuntu/
root@ip-172-31-24-197:/home/ubuntu# ls
wordpress-securedocker.zip
root@ip-172-31-24-197:/home/ubuntu# cd wordpress-securedocker/
root@ip-172-31-24-197:/home/ubuntu/wordpress-securedocker# ls
Dockerfile  docker-compose.yml
root@ip-172-31-24-197:/home/ubuntu/wordpress-securedocker# docker compose up -d
[+] Running 2/2
✔ Container wordpress-securedocker-db-1 Started      0.5s
✔ Container wordpress-securedocker-web-1 Started     0.4s
root@ip-172-31-24-197:/home/ubuntu/wordpress-securedocker#

```

i-0fc517889842554eb (wordpress-2)

PublicIPs: 13.221.75.49 PrivateIPs: 172.31.24.197

CloudShell Feedback | © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

8. ALB & WAF:

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

The screenshot shows the AWS Management Console interface for configuring an HTTP:80 listener. The breadcrumb navigation is EC2 > Load balancers > wordpress-application > HTTP:80 listener. The left-hand navigation menu includes sections for EC2, Instances, Images, Elastic Block Store, Network & Security, and Auto Scaling. The main content area is titled "HTTP:80" and contains the following details:

- Details:** A listener checks for connection requests using the protocol and port that you configure. The default action and any additional rules that you create determine how the Application Load Balancer routes requests to its registered targets.
- Protocol:Port:** HTTP:80
- Load balancer:** [wordpress-application](#)
- Default actions:**
 - Forward to target group [wordpress](#): 1 (100%)
 - Target group stickiness: Off
- Listener ARN:** [arn:aws:elasticloadbalancing:us-east-1:135016766255:listener/app/wordpress-application/2b609ac82e7368f2/94405b91261c1065](#)

Below the details, there are tabs for "Rules", "Attributes", and "Tags". The "Rules" tab is active, showing "Listener rules (1)" with a search filter and a table of rules. The table has columns for Priority, Name tag, Conditions (If), Actions (Then), and Actions. One rule is listed with a priority of 1, name tag "Default", condition "If no other rule applies", and action "Forward to target group [wordpress](#): 1 (100%)".

The screenshot shows the AWS Management Console interface for configuring a Target Group. The breadcrumb navigation is EC2 > Target groups > wordpress. The left-hand navigation menu includes sections for Capacity Reservations, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area is titled "wordpress" and contains the following details:

- Details:** [arn:aws:elasticloadbalancing:us-east-1:135016766255:targetgroup/wordpress/61e81802c4662ea4](#)
- Target type:** Instance
- Protocol : Port:** HTTP: 8080
- Protocol version:** HTTP1
- VPC:** [vpc-0b0af4cbd3bc770f5](#)
- IP address type:** IPv4
- Load balancer:** [wordpress-application](#)

Below the details, there is a summary of target status:

2	0	2	0	0	0
Total targets	Healthy	Unhealthy	Unused	Initial	Draining
	0 Anomalous				

Below the summary, there is a section for "Distribution of targets by Availability Zone (AZ)" with a note: "Select values in this table to see corresponding filters applied to the Registered targets table below."

At the bottom, there are tabs for "Targets", "Monitoring", "Health checks", "Attributes", and "Tags". The "Targets" tab is active, showing "Registered targets (2)" with a search filter and buttons for "Anomaly mitigation: Not applicable", "Deregister", and "Register targets".

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

The screenshot shows the AWS Management Console interface for configuring an Elastic Load Balancing (ALB) instance. The browser address bar shows the URL: `us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LoadBalancers:`. The console header includes the AWS logo, a search bar for 'load balancers', and account information for 'United States (N. Virginia)' with account ID '1350-1676-6255'.

The main content area is titled 'Load balancers (1/1)' and provides a brief description: 'Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.' Below this is a table listing the load balancers:

Name	State	Type	Scheme	IP address type	VPC ID	Availability
wordpress-application	Active	application	Internet-facing	IPv4	vpc-0b0af4cbd3bc770f...	3 Availability

The 'Details' tab for the 'wordpress-application' load balancer is selected, showing the following information:

- Load balancer type:** Application
- Status:** Active
- VPC:** vpc-0b0af4cbd3bc770f5
- Load balancer IP address type:** IPv4
- Scheme:** Internet-facing
- Hosted zone:** Z355XD0TRQ7X7K
- Availability Zones:** subnet-008dd394f0cc7e463 (us-east-1a (use1-az2)), subnet-0f5df099d44fe92549 (us-east-1a (use1-az2))
- Date created:** July 24, 2025, 14:26 (UTC+01:00)

- Create a Web ACL and associate it to the ALB

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

The screenshot shows a web browser displaying the 'Student Management System' application. The browser address bar shows the URL: `wordpress-application-1853364263.us-east-1.elb.amazonaws.com/index.php`. The page features a header with the title 'Student Management System' and a 'Logout' link.

The main content area contains two forms:

- Add Student:** A form with input fields for 'Name:' and 'Email:', and a blue 'Add Student' button.
- Add Course:** A form with an input field for 'Course Name:' containing the text 'Msc in cloud computing', and a blue 'Add Course' button.

The screenshot shows the AWS Management Console for an Application Load Balancer named 'wordpress-application'. The console displays various details including:

- Status:** Active
- Load balancer type:** Application
- Scheme:** Internet-facing
- Hosted zone:** Z355XDOTRQ7X7K
- VPC:** vpc-0b0af4cbd3bc770f5
- Availability Zones:** subnet-008dd594f0cc7e463 (us-east-1a), subnet-0f5df099d4fe92549 (us-east-1c), subnet-0e886fef14d3b5b95 (us-east-1b)
- Load balancer IP address type:** IPv4
- Date created:** July 24, 2025, 14:26 (UTC+01:00)
- Load balancer ARN:** arn:aws:elasticloadbalancing:us-east-1:135016766255:loadbalancer/app/wordpress-application/2b609ac82e7368f2
- DNS name:** wordpress-application-1853364263.us-east-1.elb.amazonaws.com (A Record)

The 'Listeners and rules' section shows one listener with the following configuration:

- 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"}

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"}
}
```

The screenshot shows the AWS WAF console for a web ACL named "WordPress-Container-Protection". The "Rules" tab is active, displaying a table with two rules:

Name	Action	Priority	Custom response
AWS-AWSManagedRulesSQLiRuleSet	Use rule actions	0	-
BlockBadBots	Block	5	-

Below the rules table, the "Web ACL capacity units (WCUs) used by your web ACL" section shows a progress indicator for 230/5000 WCUs. The "Default web ACL action for requests that don't match any rules" is set to "Block".

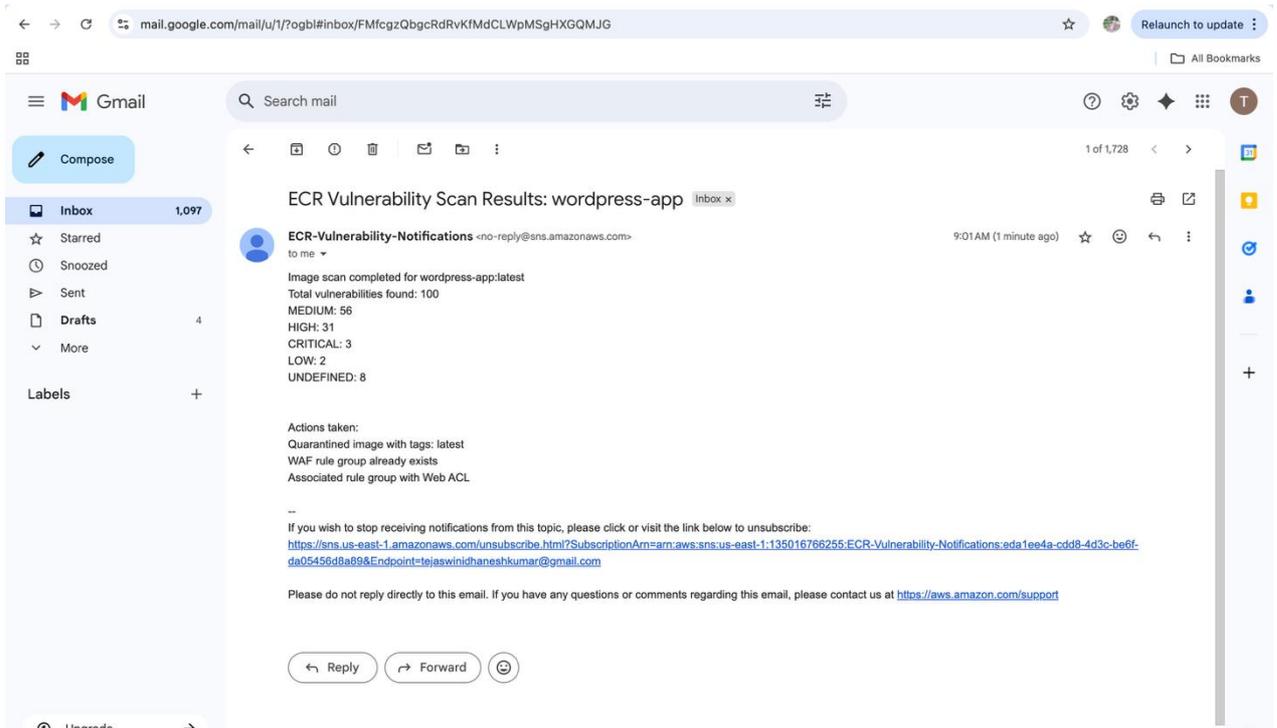
9. ECR Scan → EventBridge → Lambda → SNS (Email):

SNS topic: HighSeverityVulnAlerts (subscribe your email)

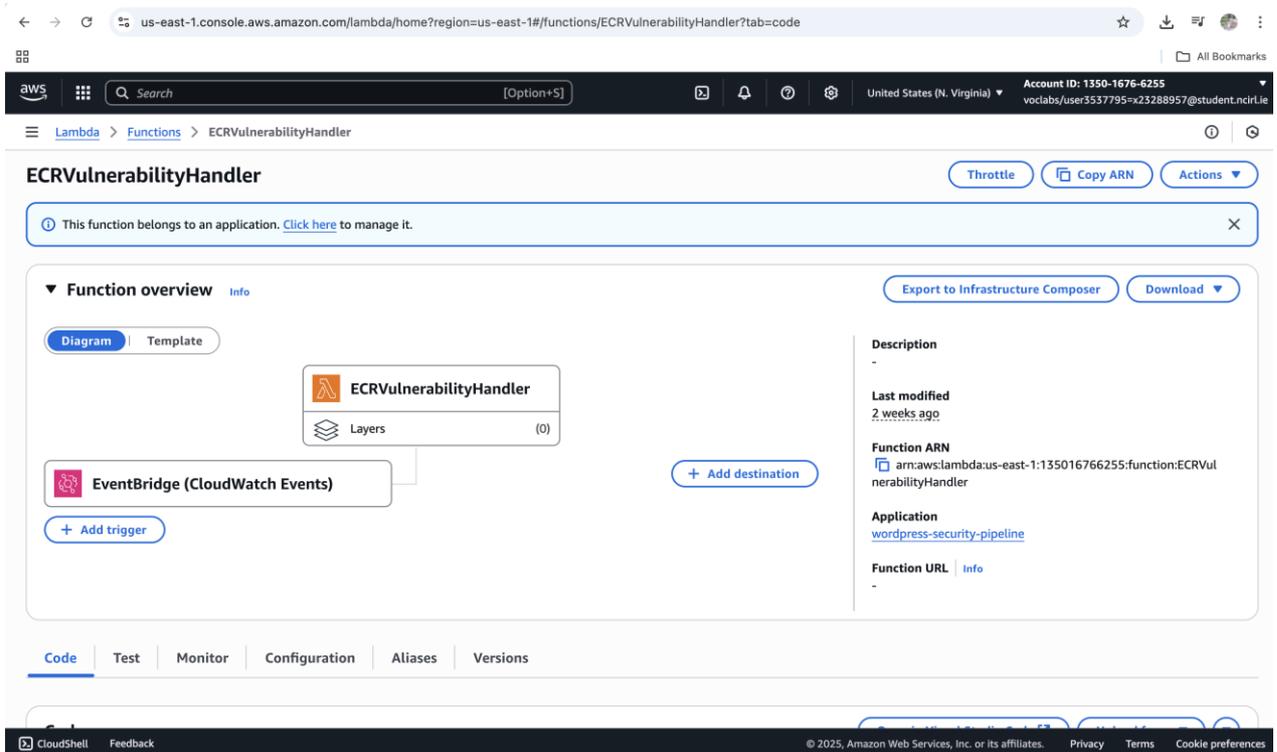
The screenshot shows the AWS SNS console for a subscription to the "HighSeverityVulnAlerts" topic. The subscription details are as follows:

Field	Value
ARN	arn:aws:sns:us-east-1:135016766255:ECR-Vulnerability-Notifications:eda1ee4a-cdd8-4d3c-be6f-da05456d8a89
Endpoint	tejaswinidhaneshkumar@gmail.com
Topic	ECR-Vulnerability-Notifications
Subscription Principal	arn:aws:iam::135016766255:role/voclabs
Status	Confirmed
Protocol	EMAIL

The "Subscription filter policy" section shows a "Redrive policy (dead-letter queue)" and a note that no filter policy is currently configured for this subscription.

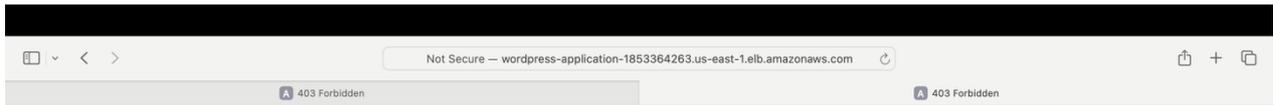


EventBridge rule: source=aws.ecr, detail-type='ECR Image Scan', filter CRITICAL>0 OR HIGH>0
 Target: Lambda → publishes to SNS (and optionally updates WAF rules)



SQL Injection ,WAF Metrics And Cloud watch logs:

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



403 Forbidden

IP Address	URL	Rule Set	Action	Standard Time
233.74.86	/?username=%27%20OR%20%27x%27=%27x	AWS#AWSManagedRulesSQLiRuleSet#SQLI_QUERYARGUMENTS	BLOCK	Mon Aug 04 2025 09:08:07 GMT+0100 (Irish Standard Time)
145.76.96	/?username=%27%20OR%20%27x%27=%27x	AWS#AWSManagedRulesSQLiRuleSet#SQLI_QUERYARGUMENTS	BLOCK	Mon Aug 04 2025 09:08:28 GMT+0100 (Irish Standard Time)
180.221.31	/?username=%27%20OR%201--	AWS#AWSManagedRulesSQLiRuleSet#SQLI_QUERYARGUMENTS	BLOCK	Mon Aug 04 2025 09:07:43 GMT+0100 (Irish Standard Time)
147.11.254	/?username=%27%20OR%20%27x%27=%27x	AWS#AWSManagedRulesSQLiRuleSet#SQLI_QUERYARGUMENTS	BLOCK	Mon Aug 04 2025 09:08:07 GMT+0100 (Irish Standard Time)

us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#logsV2:log-groups/log-group/\$252Faws\$252Flambda\$252FECRVulnerabilityHandler/log-eve... ☆

CloudWatch > Log groups > /aws/lambda/ECRVulnerabilityHandler > 2025/08/08/[\$LATEST]448b03b2bbf044ee8ba5f00c8e931b26

CloudWatch

- CloudShell
- Feedback

Account ID: 1350-1676-6255
voclabs/user3537795-x23288957@student.ncirl.ie

Log events

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	

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences