WEEKLY TASK SOLUTION FOR AWS BOOTCAMP BY CLOUDSEC NETWORK
NAME: OLUFEMI AYOMITOPE

WEEK 1: Introduction to AWS Cloud

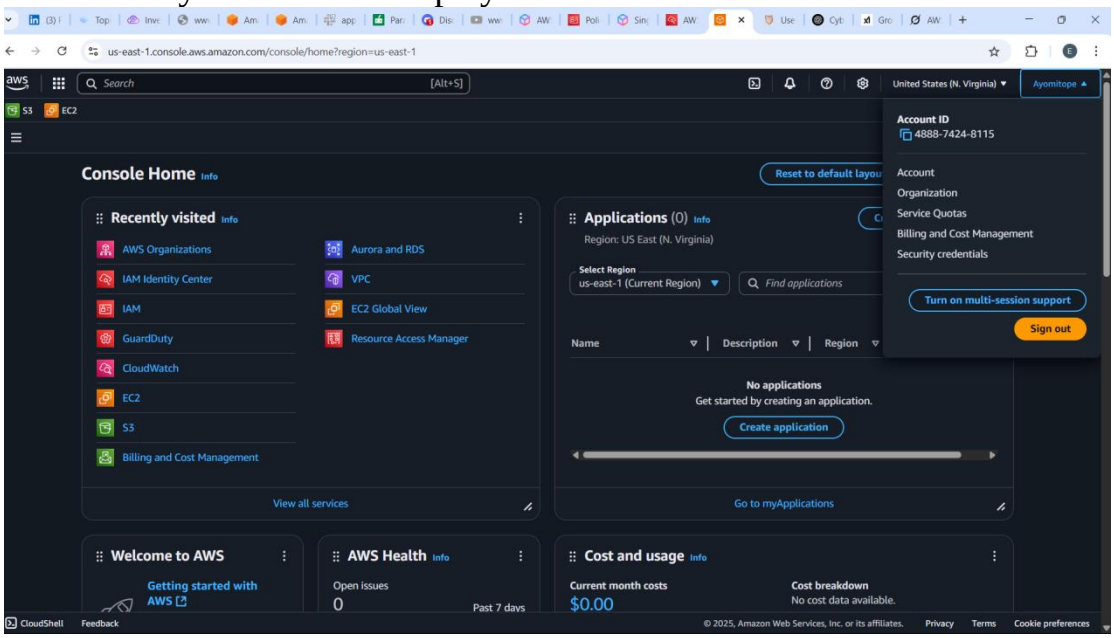
***Deliverable:**

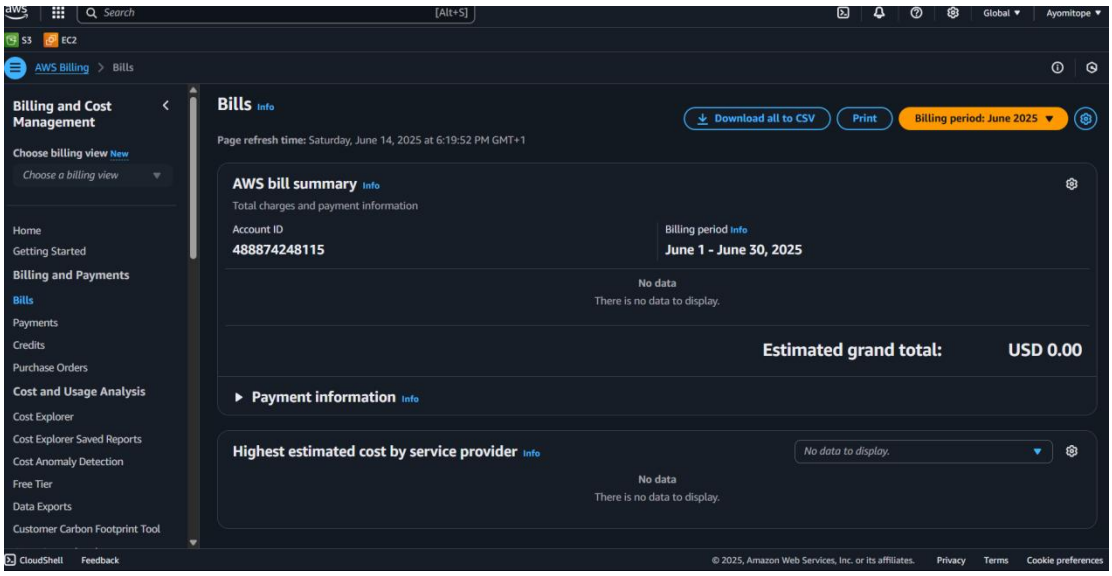
Provide evidence of an active account by submitting a screenshot of your AWS Management Console (e.g., showing your account ID or billing dashboard).

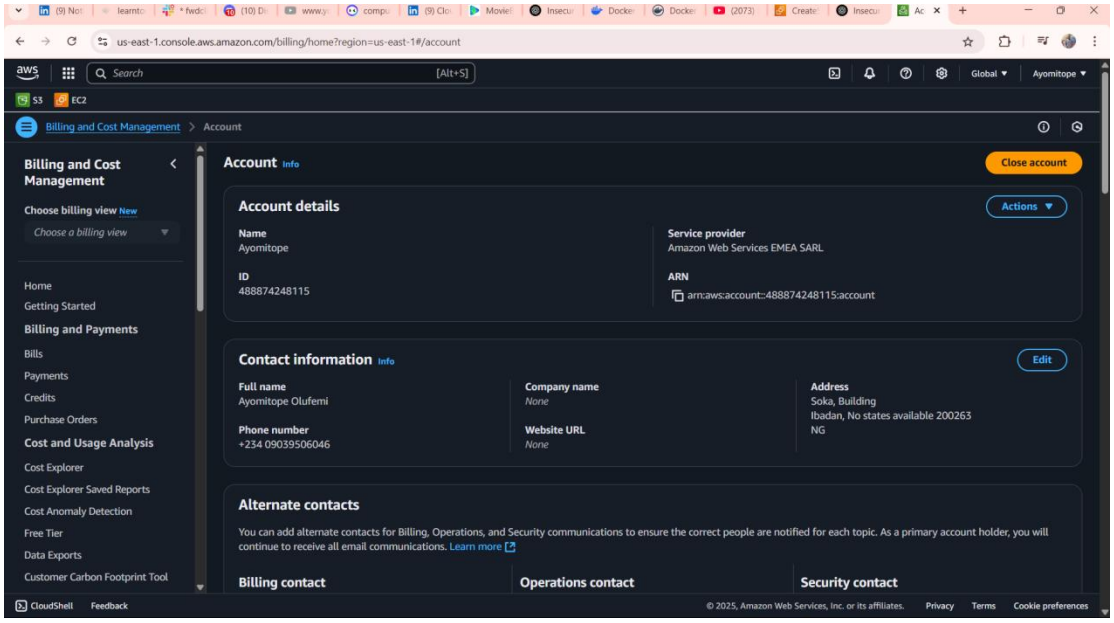
STEP 1: Logged in to AWS Management Console using my existing information

STEP 2: Clicked my account name at the top right and select My Account

STEP 3: My Account ID displayed







WEEK 2: AWS Identity Center and Permission Sets

***Deliverable:**

Submit a screenshot showing the Identity Center instance, the user created, and the assigned permission set.

STEP 1: Sign in to AWS Management Console

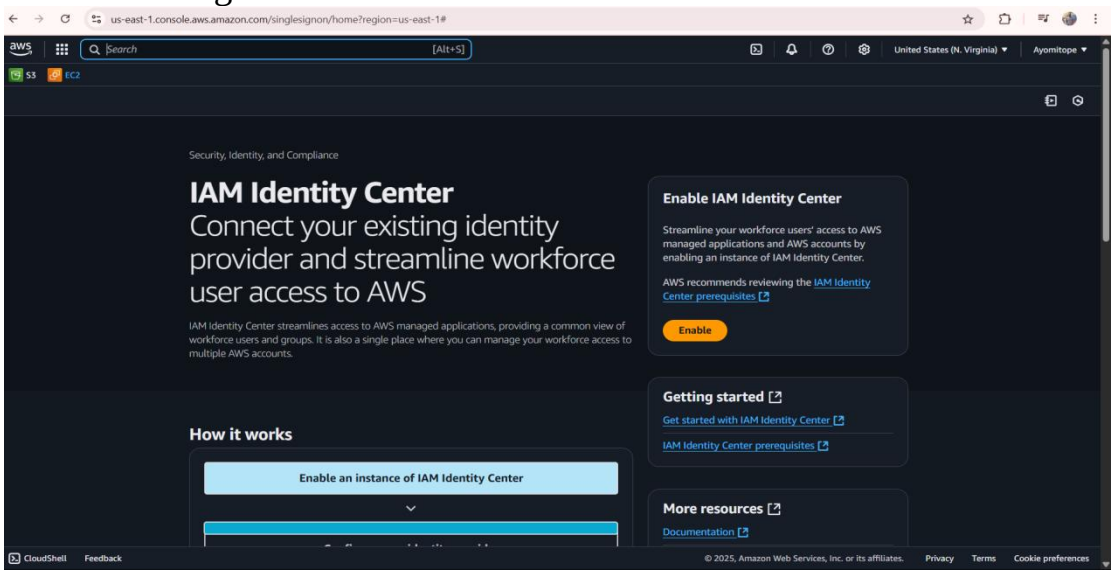
STEP 2: Navigate to AWS Identity Center

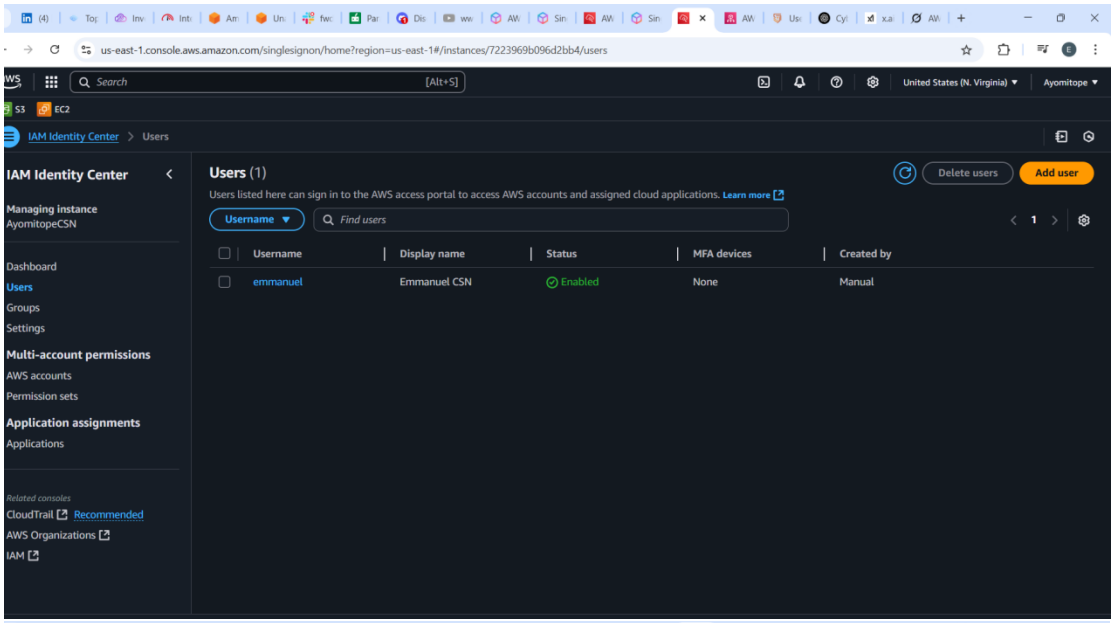
STEP 3: Enable AWS Identity Center (if not already enabled)

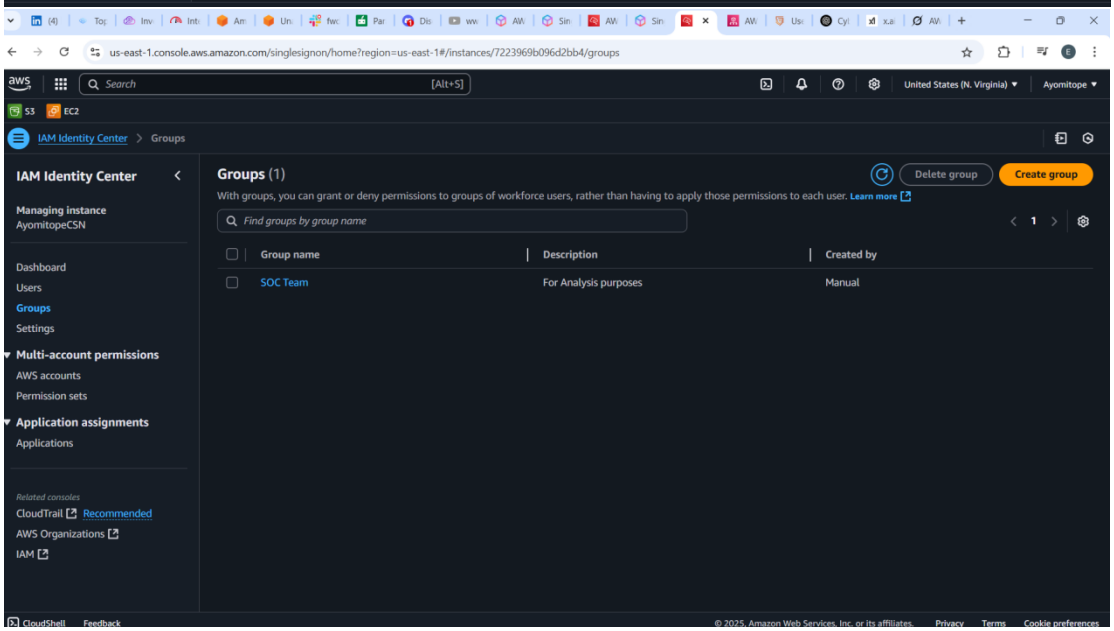
STEP 4: Create a New User

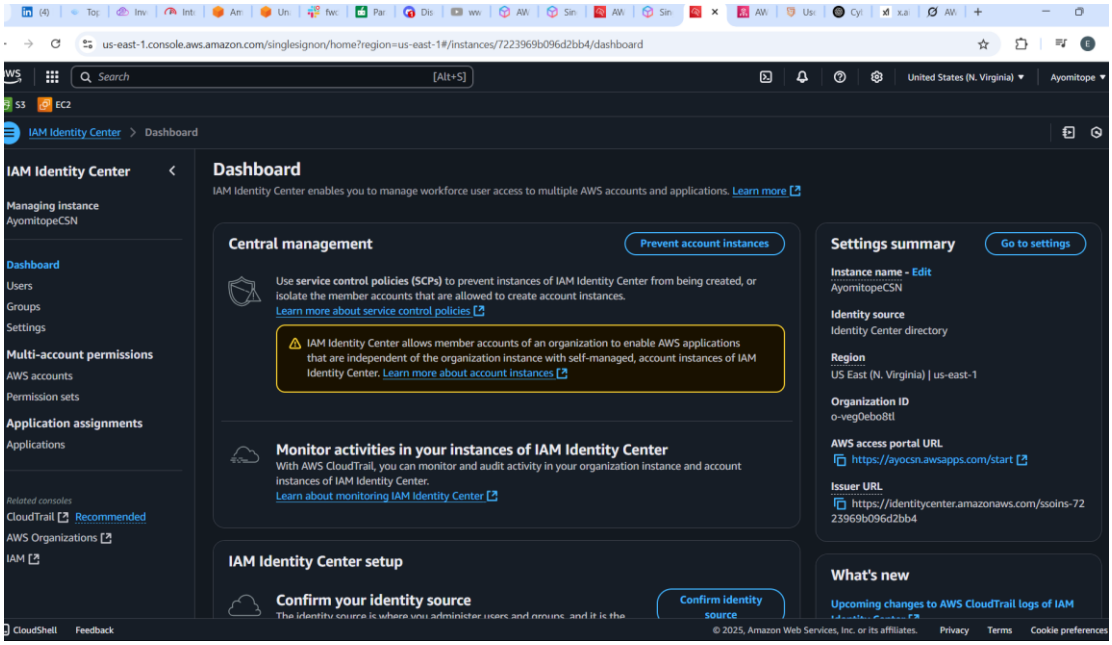
STEP 5: Create a Permission Set Using “SecurityAudit” Policy

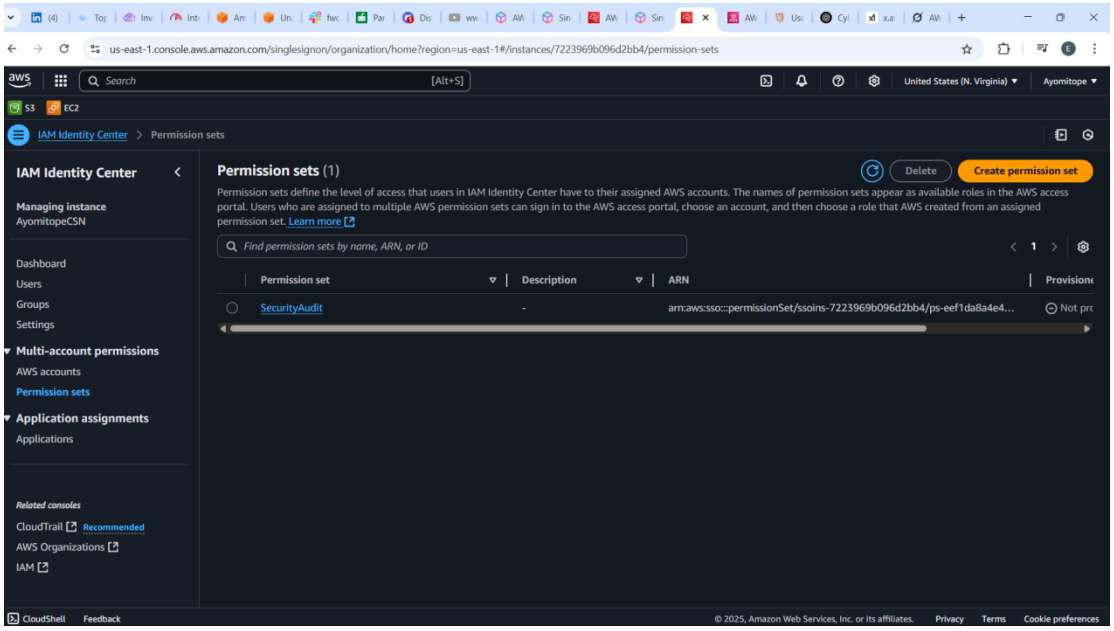
STEP 6: Assign the User to an AWS Account and Permission Set




us-east-1.console.aws.amazon.com/singlesignon/home?region=us-east-1#/instances/7223969b096d2bb4/users
Search
[Alt+S]
United States (N. Virginia)
Ayomitope
S3
EC2
IAM Identity Center
Users
IAM Identity Center
Managing instance
AyomitopeCSN
Dashboard
Users
Groups
Settings
Multi-account permissions
AWS accounts
Permission sets
Application assignments
Applications
Related consoles
CloudTrail
Recommended
AWS Organizations
IAM
Users (1)
Users listed here can sign in to the AWS access portal to access AWS accounts and assigned cloud applications.
Learn more
Delete users
Add user
Username
Find users
Username
Display name
Status
MFA devices
Created by
emmanuel
Emmanuel CSN
Enabled
None
Manual


us-east-1.console.aws.amazon.com/singlesignon/home?region=us-east-1#/instances/7223969b096d2bb4/groups
Search
[Alt+S]
United States (N. Virginia)
Ayomitope
S3
EC2
IAM Identity Center
Groups
IAM Identity Center
Managing instance
AyomitopeCSN
Dashboard
Users
Groups
Settings
Multi-account permissions
AWS accounts
Permission sets
Application assignments
Applications
Related consoles
CloudTrail
Recommended
AWS Organizations
IAM
Groups (1)
With groups, you can grant or deny permissions to groups of workforce users, rather than having to apply those permissions to each user.
Learn more
Delete group
Create group
Find groups by group name
Group name
Description
Created by
SOC Team
For Analysis purposes
Manual
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences





WEEK 3: Provisioning and Securing EC2 Instances

*Deliverable

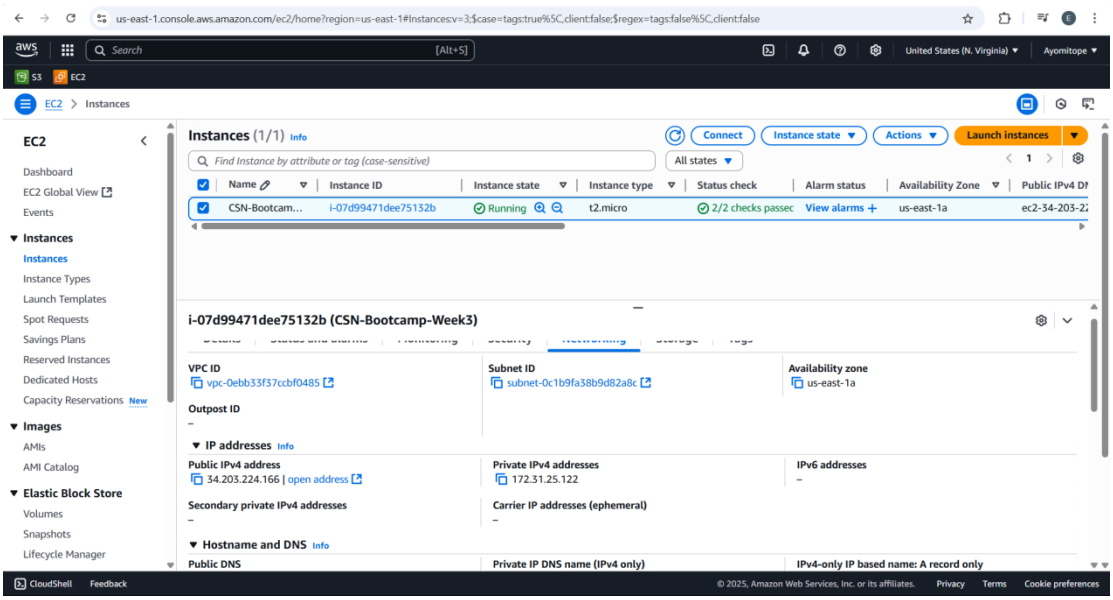
Provide screenshots of your running Windows EC2 instance in the AWS Management Console. The screenshots should clearly show:

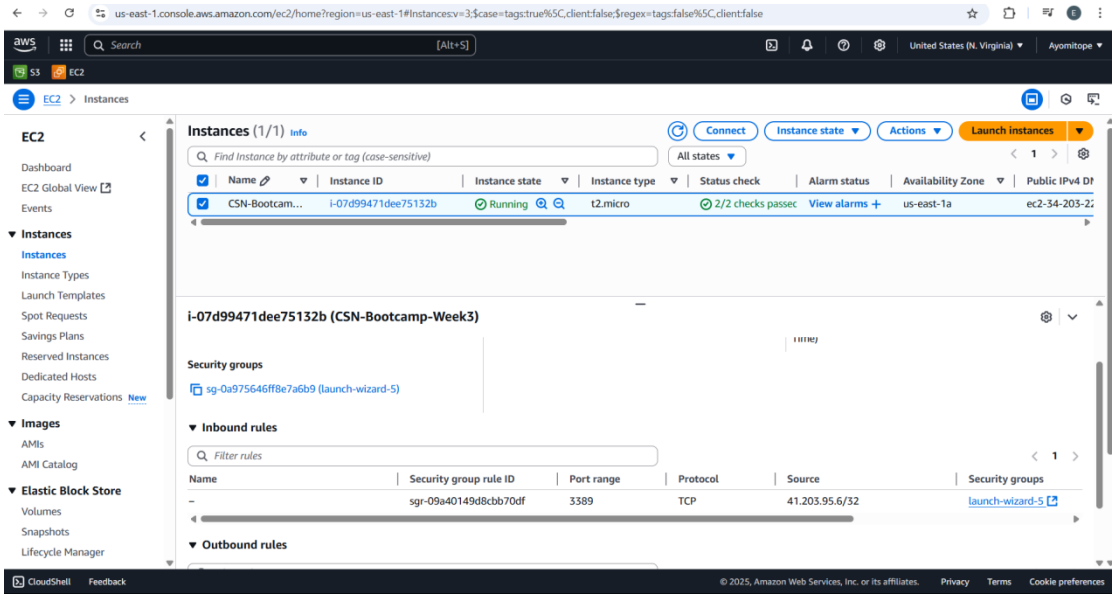
- The instance's Name tag set to CSN-Bootcamp-Week3,
- The Security Group inbound rules allowing RDP (port 3389) only from your public IP address,
- A successful Remote Desktop (RDP) connection to the instance.

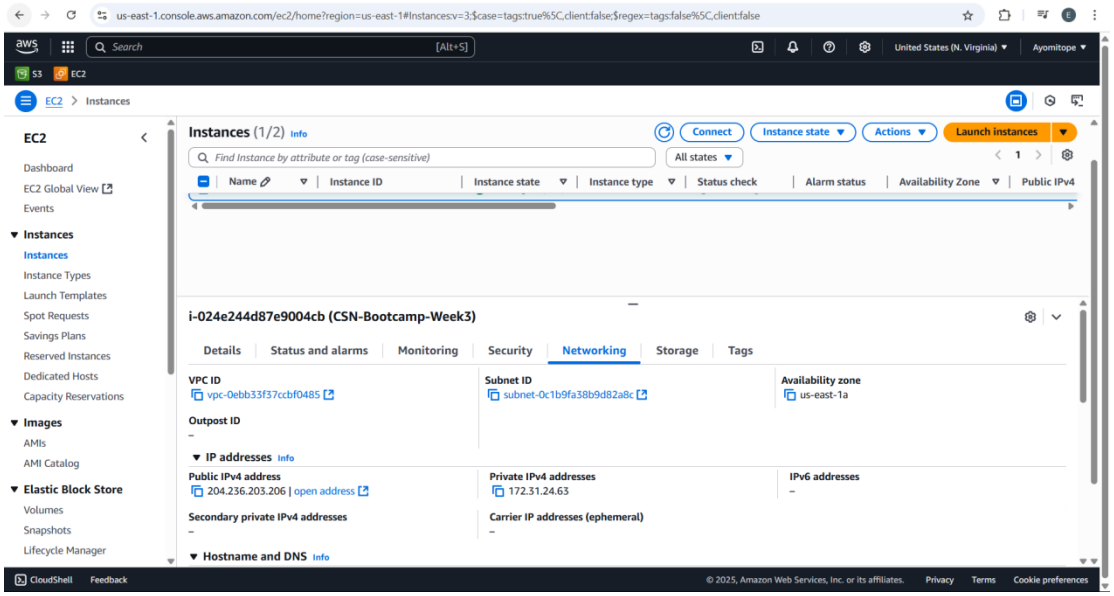
STEP 1: Sign in to AWS Management Console

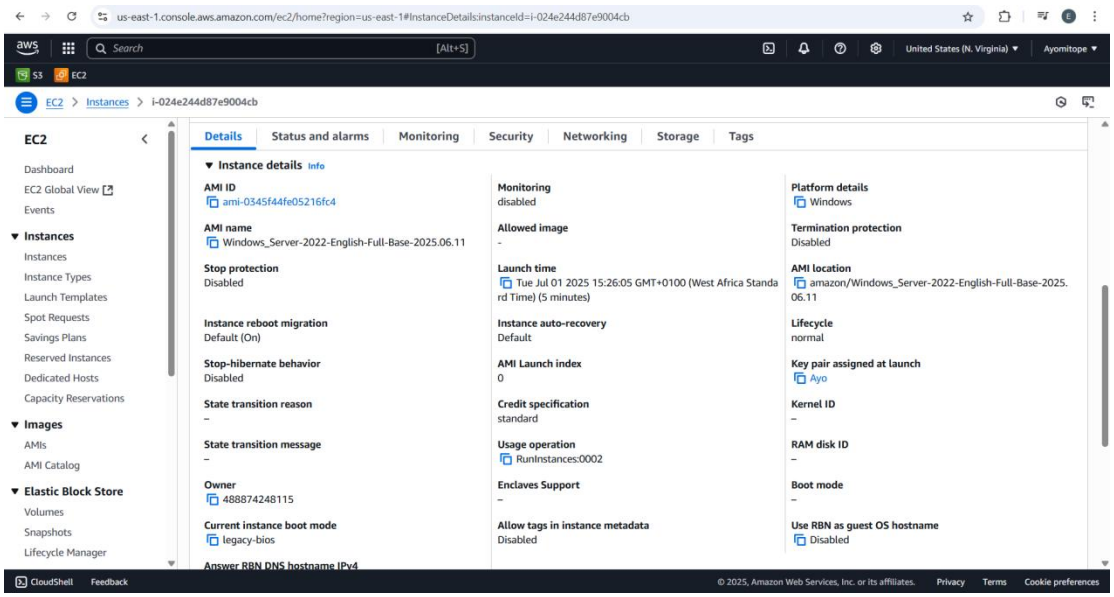
STEP 2: Navigate to EC2 Dashboard

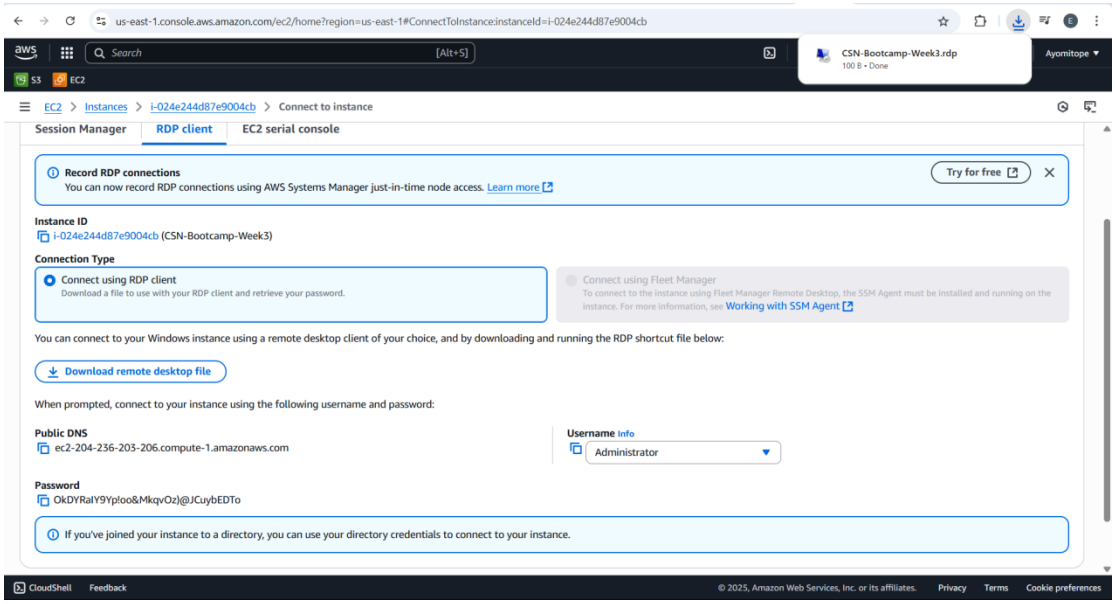
STEP 3: Launch a New Instance
· **Name** the instance: CSN-Bootcamp-Week3
· Chose **Amazon Windows Server 2022 Base AMI**
· Selected an instance type: t2.micro & created network settings
· Launched an Instance
STEP 4: When the Instance started running, it was connected via Remote Desktop (RDP)

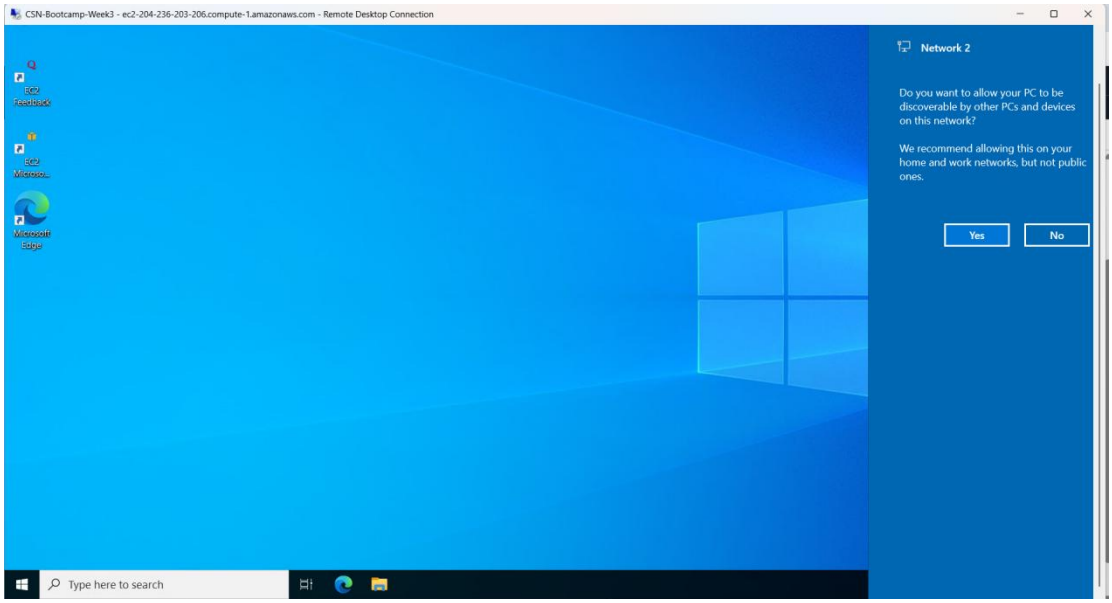





us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:false
Search [Alt+S]
United States (N. Virginia)
Ayomitope
EC2 > Instances
Instances (1/2) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4
i-024e244d87e9004cb (CSN-Bootcamp-Week3)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
VPC ID
vpc-0ebb33f37ccbf0485
Subnet ID
subnet-0c1b9fa38b9d82a8c
Availability zone
us-east-1a
Outpost ID
IP addresses Info
Public IPv4 address
204.236.203.206 | open address
Private IPv4 addresses
172.31.24.63
IPv6 addresses
Secondary private IPv4 addresses
Carrier IP addresses (ephemeral)
Hostname and DNS Info
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstanceDetails:instanceId=i-024e244d87e9004cb
Search [Alt+S]
United States (N. Virginia)
Ayomitope
EC2 > Instances > i-024e244d87e9004cb
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance details Info
AMI ID
ami-0345f44fe05216fc4
AMI name
Windows_Server-2022-English-Full-Base-2025.06.11
Stop protection
Disabled
Instance reboot migration
Default (On)
Stop-hibernate behavior
Disabled
State transition reason
State transition message
Owner
488874248115
Current instance boot mode
legacy-bios
Monitoring
disabled
Allowed image
Launch time
Tue Jul 01 2025 15:26:05 GMT+0100 (West Africa Standard Time) (5 minutes)
Instance auto-recovery
Default
AMI Launch index
0
Credit specification
standard
Usage operation
RunInstances:0002
Enclaves Support
Allow tags in instance metadata
Disabled
Platform details
Windows
Termination protection
Disabled
AMI location
amazon/Windows_Server-2022-English-Full-Base-2025.06.11
Lifecycle
normal
Key pair assigned at launch
Ayo
Kernel ID
RAM disk ID
Boot mode
Use RBN as guest OS hostname
Disabled
Answer RBN DNS hostname IPv4
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ConnectToInstance:instanceId=i-024e244d87e9004cb
Search [Alt+S]
CSN-Bootcamp-Week3.rdp
100 B • Done
Ayomitope
EC2 > Instances > i-024e244d87e9004cb > Connect to instance
Session Manager
RDP client
EC2 serial console
Record RDP connections
You can now record RDP connections using AWS Systems Manager just-in-time node access. Learn more
Try for free
Instance ID
i-024e244d87e9004cb (CSN-Bootcamp-Week3)
Connection Type
Connect using RDP client
Download a file to use with your RDP client and retrieve your password.
Connect using Fleet Manager
To connect to the instance using Fleet Manager Remote Desktop, the SSM Agent must be installed and running on the instance. For more information, see Working with SSM Agent
You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:
Download remote desktop file
When prompted, connect to your instance using the following username and password:
Public DNS
ec2-204-236-203-206.compute-1.amazonaws.com
Username Info
Administrator
Password
OkDYRaIY9Yp!oo&MkqvOz)@JCuybEDTo
If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



WEEK 4: Cloud Networking with AWS VPC and Subnets

***Deliverable**
Provide clear screenshots from the AWS Console showing:

- The two VPCs and their CIDR blocks,
- The public and private subnets in each VPC,
- The VPC peering connection in **Active** status,
- The route tables with routes pointing to the other VPC's CIDR block through the peering connection

☑ STEP 1: Created two VPCs:
✔ VPC-A with CIDR block: **10.10.0.0/16**
✔ VPC-B with CIDR block: **10.20.0.0/16**

☑ STEP 2: In each VPC, created:
✔ 1 Public Subnet (e.g., **10.10.1.0/24** in VPC-A)
✔ 1 Private Subnet (e.g., **10.10.2.0/24** in VPC-A)
Repeat for VPC-B (e.g., **10.20.1.0/24**, **10.20.2.0/24**)

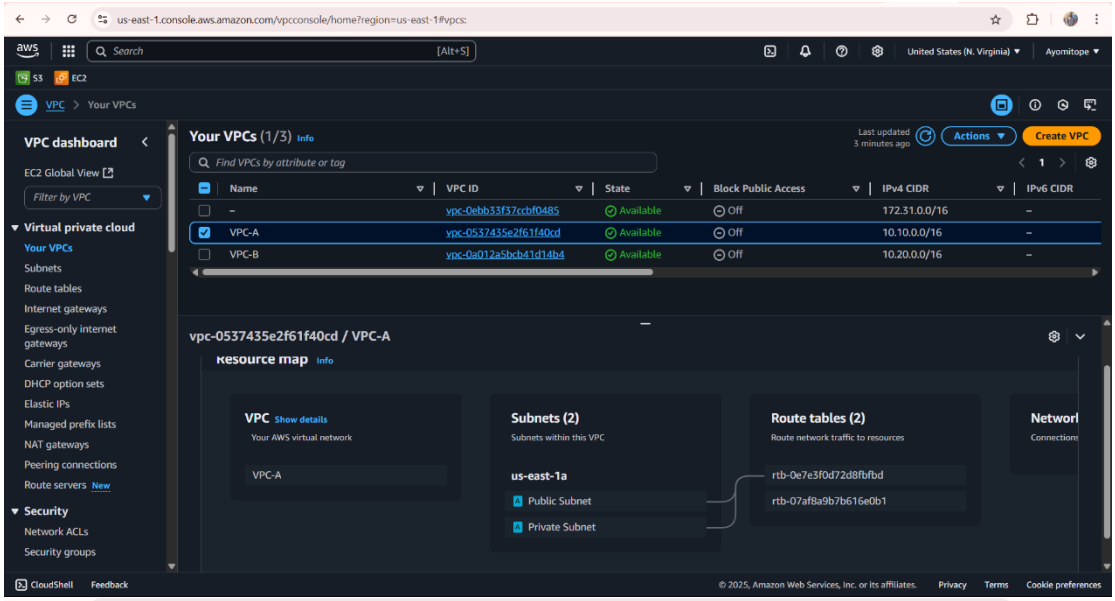
☑ STEP 3: Set up a VPC Peering connection between VPC-A and VPC-B.
✔ Requester: VPC-A

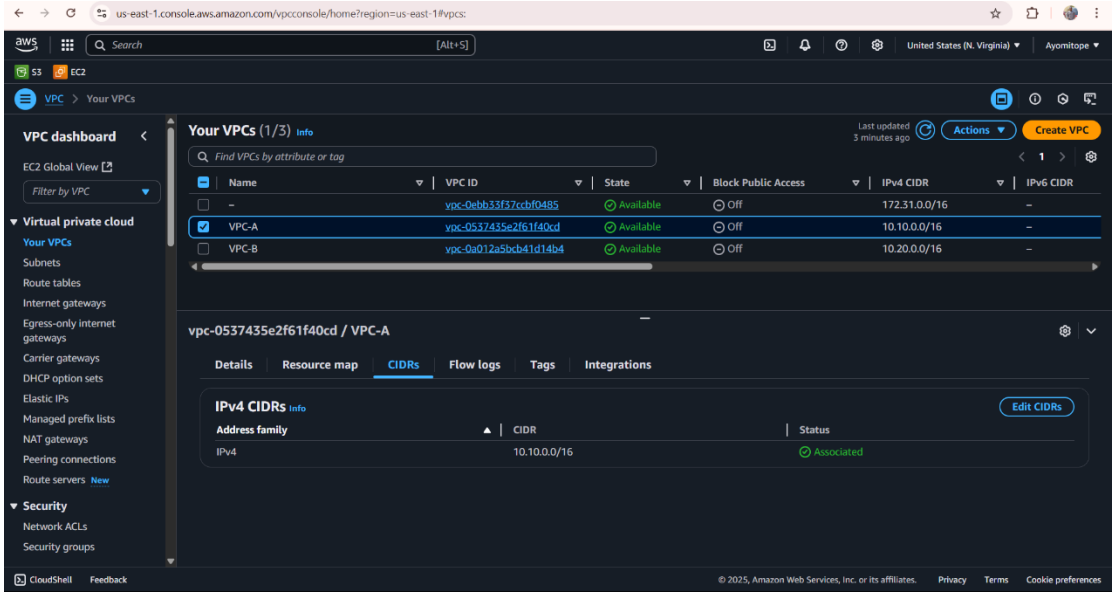
✔ Accepter: VPC-B

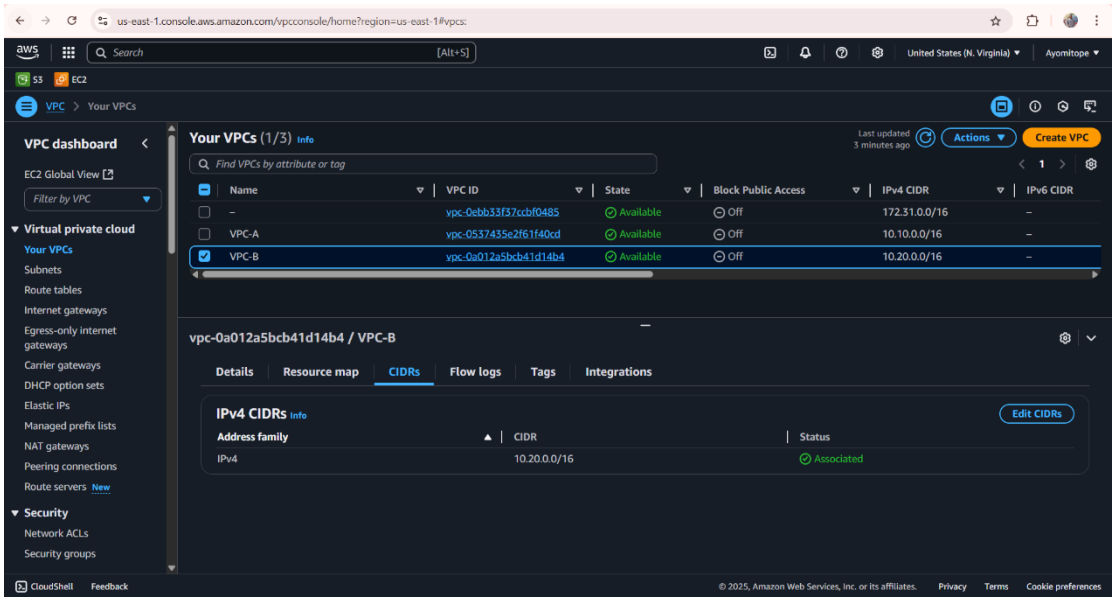
☑ STEP 4: Updated route tables in both VPCs so they can talk to each other via the peering connection.

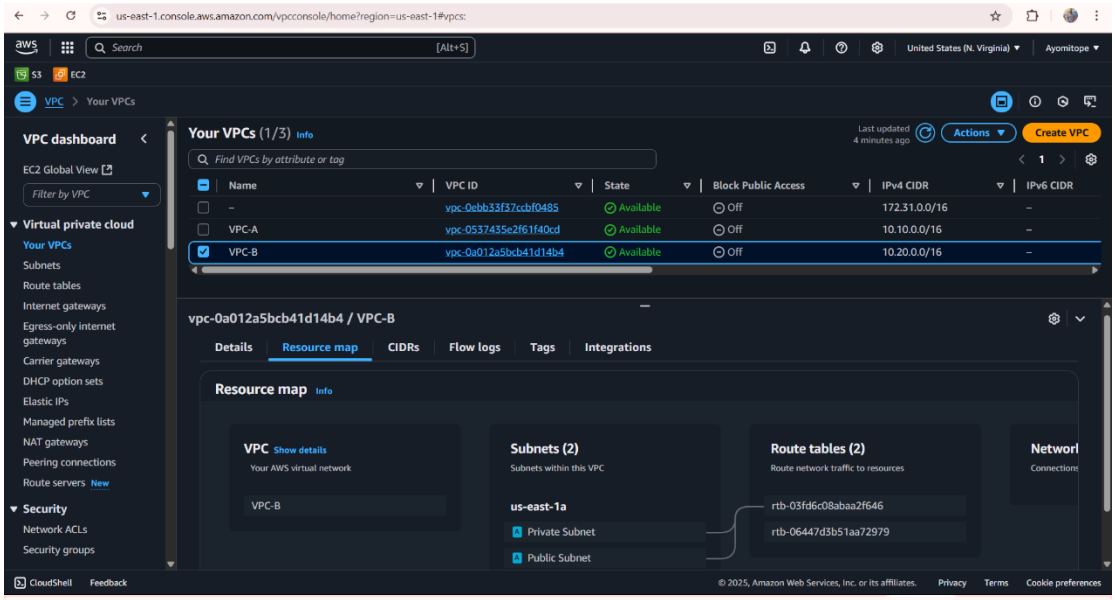
✔ In VPC-A, add a route to **10.20.0.0/16** via Peering Connection

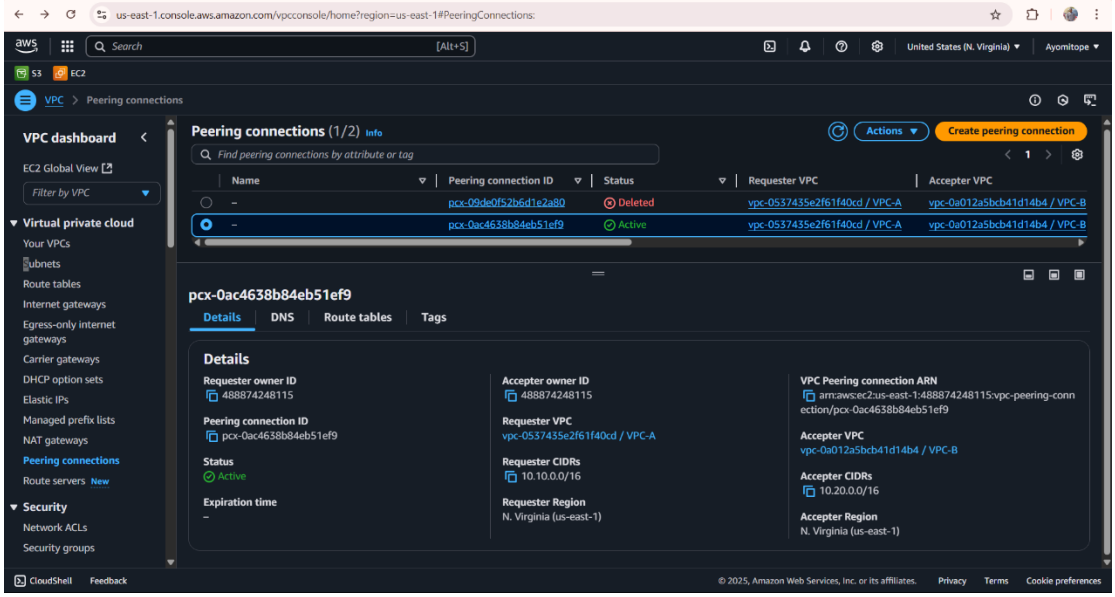
✔ In VPC-B, add a route to **10.10.0.0/16** via Peering Connection

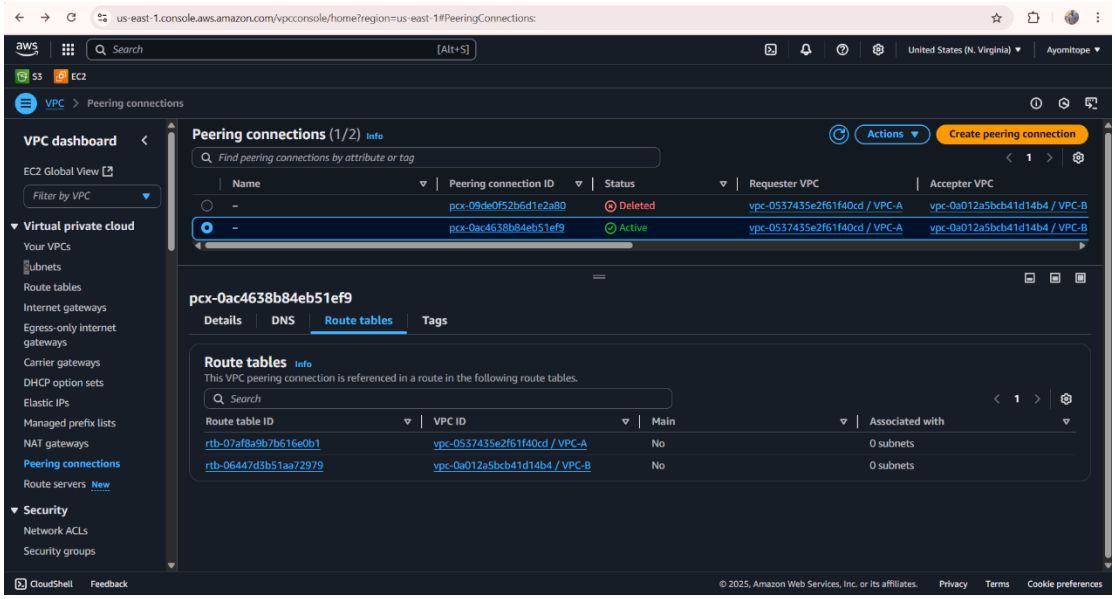


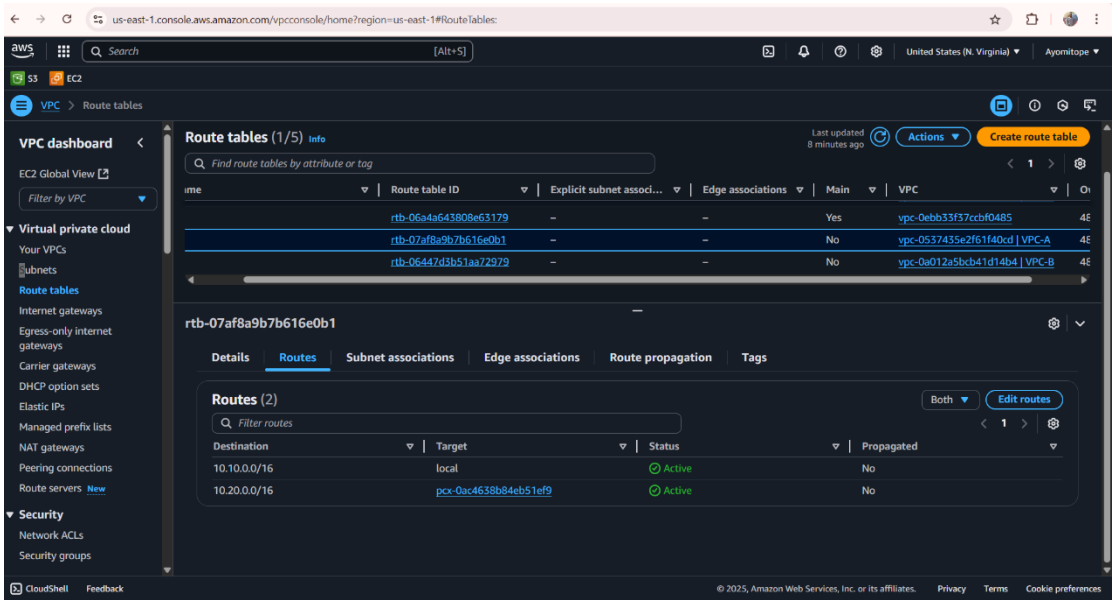


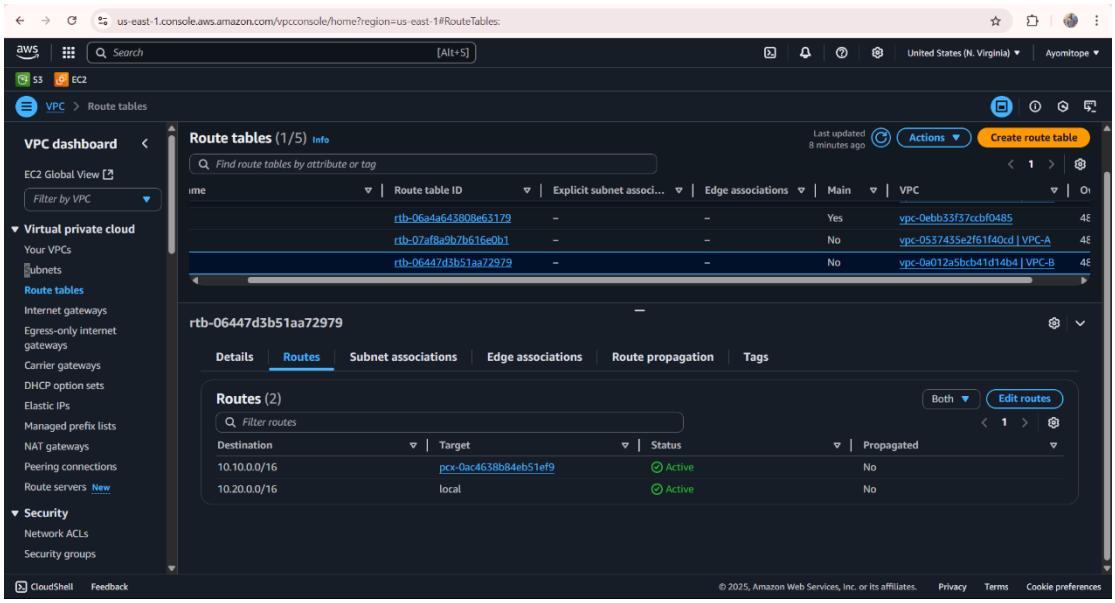

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#vpcs:
Search [Alt+S]
United States (N. Virginia)
Ayomitope
S3 EC2
VPC > Your VPCs
VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Route servers New
Security
Network ACLs
Security groups
Your VPCs (1/3) Info
Last updated 3 minutes ago
Actions
Create VPC
Find VPCs by attribute or tag
Name
VPC ID
State
Block Public Access
IPv4 CIDR
IPv6 CIDR
– vpc-0ebb33f37ccbf0485 Available Off 172.31.0.0/16 –
VPC-A vpc-0537435e2f61f40cd Available Off 10.10.0.0/16 –
VPC-B vpc-0a012a5bcb41d14b4 Available Off 10.20.0.0/16 –
vpc-0a012a5bcb41d14b4 / VPC-B
Details
Resource map
CIDRs
Flow logs
Tags
Integrations
IPv4 CIDRs Info
Edit CIDRs
Address family
CIDR
Status
IPv4 10.20.0.0/16 Associated
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#vpcs:
Search [Alt+S]
United States (N. Virginia)
Ayomitope
S3 EC2
VPC > Your VPCs
VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Route servers New
Security
Network ACLs
Security groups
Your VPCs (1/3) Info
Last updated 4 minutes ago
Actions
Create VPC
Find VPCs by attribute or tag
Name
VPC ID
State
Block Public Access
IPv4 CIDR
IPv6 CIDR
– vpc-0ebb33f37ccbf0485 Available Off 172.31.0.0/16 –
VPC-A vpc-0537435e2f61f40cd Available Off 10.10.0.0/16 –
VPC-B vpc-0a012a5bcb41d14b4 Available Off 10.20.0.0/16 –
vpc-0a012a5bcb41d14b4 / VPC-B
Details
Resource map
CIDRs
Flow logs
Tags
Integrations
Resource map Info
VPC Show details
Your AWS virtual network
VPC-B
Subnets (2)
Subnets within this VPC
us-east-1a
Private Subnet
Public Subnet
Route tables (2)
Route network traffic to resources
rtb-03fd6c08abaa2f646
rtb-06447d3b51aa72979
Networl
Connections
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#PeeringConnections:
Search [Alt+S]
United States (N. Virginia)
Ayomitope
S3 EC2
VPC > Peering connections
VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Route servers New
Security
Network ACLs
Security groups
Peering connections (1/2) Info
Actions
Create peering connection
Find peering connections by attribute or tag
Name
Peering connection ID
Status
Requester VPC
Accepter VPC
– pcx-09de0f52b6d1e2a80 Deleted vpc-0537435e2f61f40cd / VPC-A vpc-0a012a5bcb41d14b4 / VPC-B
– pcx-0ac4638b84eb51ef9 Active vpc-0537435e2f61f40cd / VPC-A vpc-0a012a5bcb41d14b4 / VPC-B
pcx-0ac4638b84eb51ef9
Details
DNS
Route tables
Tags
Details
Requester owner ID
488874248115
Peering connection ID
pcx-0ac4638b84eb51ef9
Status
Active
Expiration time
–
Accepter owner ID
488874248115
Requester VPC
vpc-0537435e2f61f40cd / VPC-A
Requester CIDRs
10.10.0.0/16
Requester Region
N. Virginia (us-east-1)
VPC Peering connection ARN
arn:aws:ec2:us-east-1:488874248115:vpc-peering-connection/pcx-0ac4638b84eb51ef9
Accepter VPC
vpc-0a012a5bcb41d14b4 / VPC-B
Accepter CIDRs
10.20.0.0/16
Accepter Region
N. Virginia (us-east-1)
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#PeeringConnections:
Search
[Alt+S]
United States (N. Virginia)
Ayomitope
S3
EC2
VPC > Peering connections
VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Route servers New
Security
Network ACLs
Security groups
Peering connections (1/2) Info
Find peering connections by attribute or tag
Actions
Create peering connection
Name
Peering connection ID
Status
Requester VPC
Accepter VPC
pcx-09de0f52b6d1e2a80
Deleted
vpc-0537435e2f61f40cd / VPC-A
vpc-0a012a5bcb41d14b4 / VPC-B
pcx-0ac4638b84eb51ef9
Active
vpc-0537435e2f61f40cd / VPC-A
vpc-0a012a5bcb41d14b4 / VPC-B
pcx-0ac4638b84eb51ef9
Details
DNS
Route tables
Tags
Route tables Info
This VPC peering connection is referenced in a route in the following route tables.
Search
Route table ID
VPC ID
Main
Associated with
rtb-07af8a9b7b616e0b1
vpc-0537435e2f61f40cd / VPC-A
No
0 subnets
rtb-06447d3b51aa72979
vpc-0a012a5bcb41d14b4 / VPC-B
No
0 subnets
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

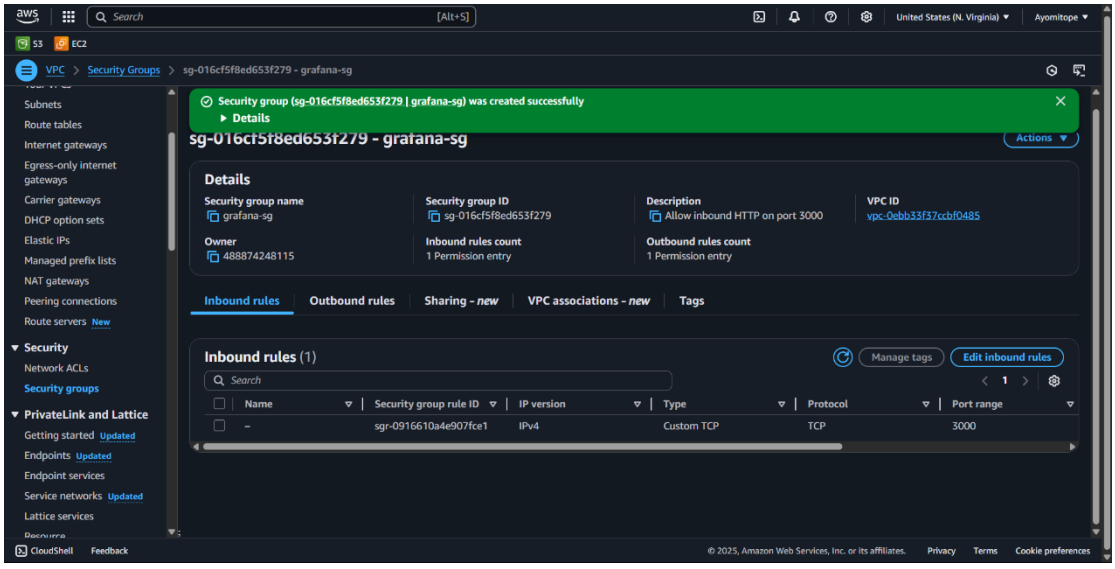

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#RouteTables:
Search
[Alt+S]
United States (N. Virginia)
Ayomitope
S3
EC2
VPC > Route tables
VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Route servers New
Security
Network ACLs
Security groups
Route tables (1/5) Info
Last updated 8 minutes ago
Actions
Create route table
Find route tables by attribute or tag
Route table ID
Explicit subnet associ...
Edge associations
Main
VPC
rtb-06a4a643808e63179
Yes
vpc-0ebb33f37ccbf0485
rtb-07af8a9b7b616e0b1
No
vpc-0537435e2f61f40cd | VPC-A
rtb-06447d3b51aa72979
No
vpc-0a012a5bcb41d14b4 | VPC-B
rtb-07af8a9b7b616e0b1
Details
Routes
Subnet associations
Edge associations
Route propagation
Tags
Routes (2)
Both
Edit routes
Filter routes
Destination
Target
Status
Propagated
10.10.0.0/16
local
Active
No
10.20.0.0/16
pcx-0ac4638b84eb51ef9
Active
No
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



WEEK 5

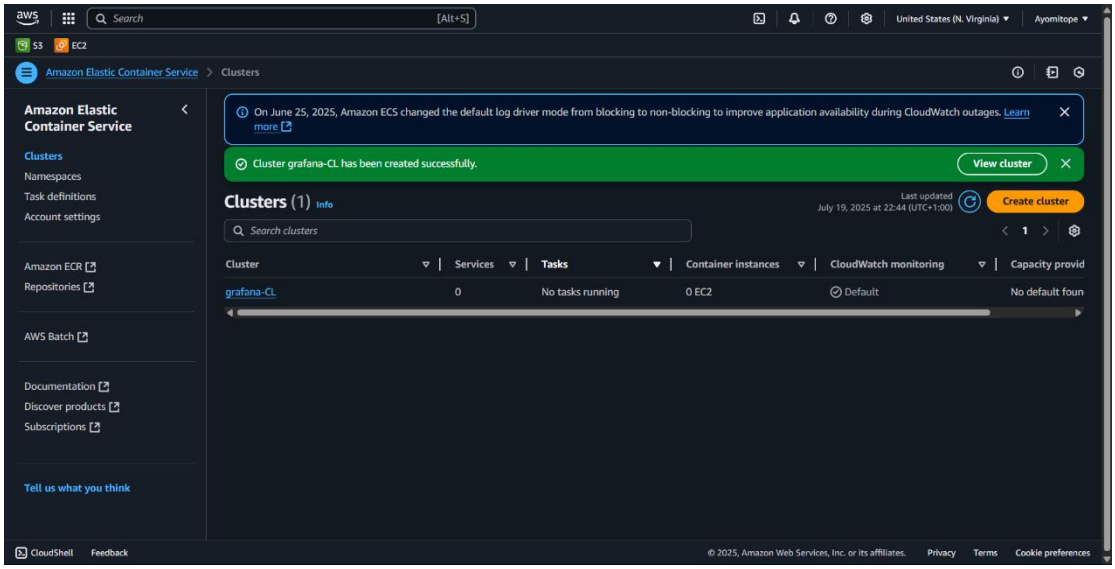
Created a Security Group

1. In the AWS Console, go to **VPC > Security Groups**.
2. Clicked **Create security group**.
    - Name: `grafana-sg`
    - Description: “Allow inbound HTTP on port 3000”
    - VPC: my *VPC*
3. Under **Inbound rules**, added:
    - Type: **Custom TCP**
    - Port range: **3000**
    - Source: **0.0.0.0/0**



Set Up an ECS Cluster

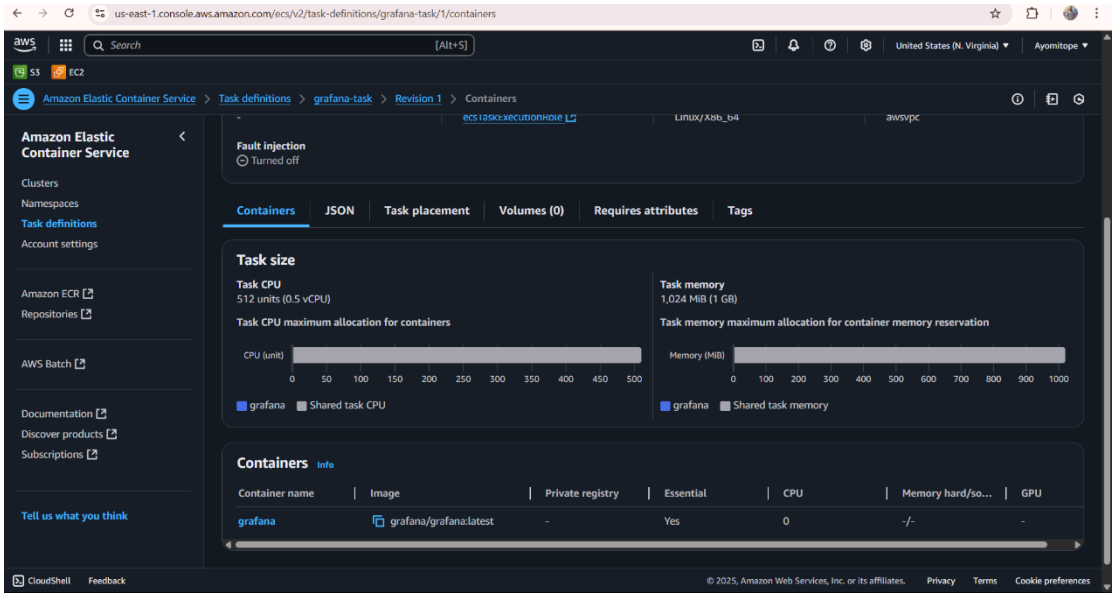
1. In the Console, navigated to **ECS > Clusters**.
2. Click **Create cluster** > **Networking only (Powered by AWS Fargate)**.
3. Named it `grafana-CL` and click **Create**.

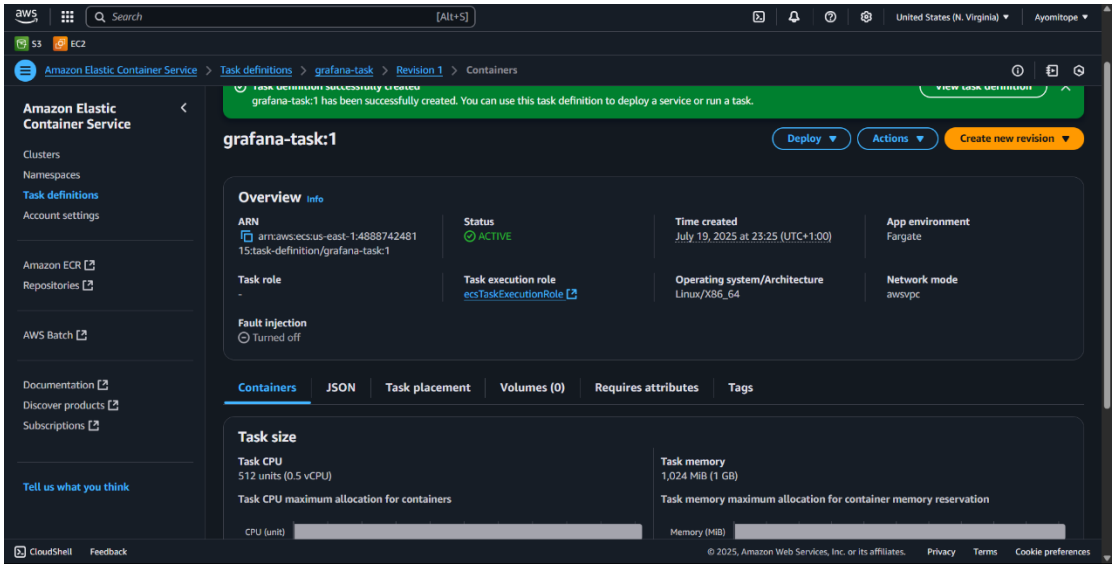


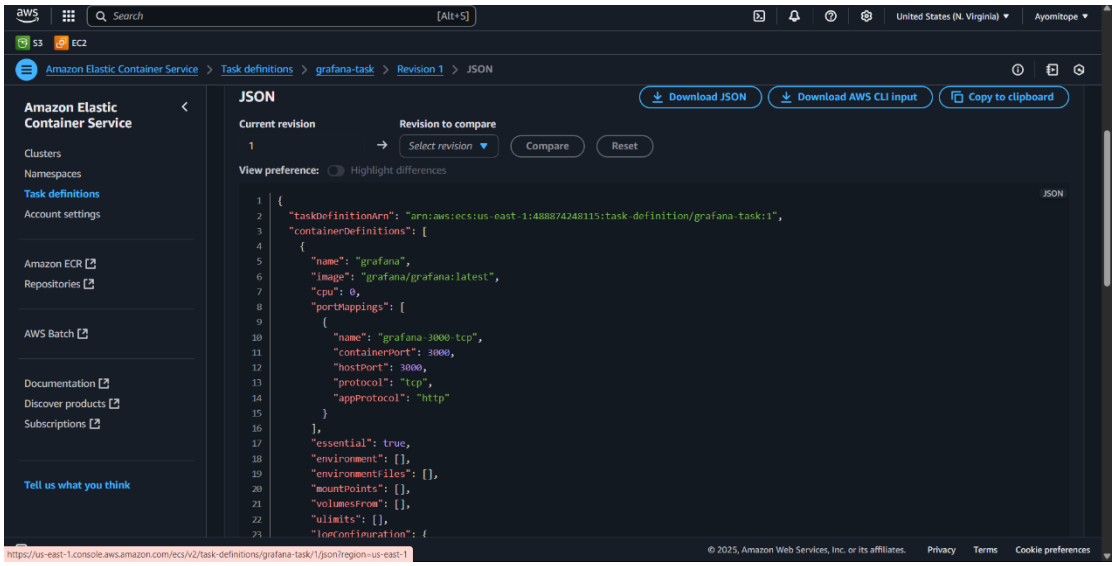
Define the Task Definition

1. Go to **ECS > Task Definitions** > **Create new Task Definition**.
2. Choose **FARGATE**, click **Next**.
3. Configure:
    - Task Definition Name: `grafana-task`
    - Task Role: *leave blank*
    - Network Mode: `awsvpc`
    - CPU & Memory: e.g., **0.5 vCPU** (512 MiB) / **1 GB**
4. Under **Container definitions**, click **Add container**:

- Name: `grafana`
- Image: `grafana/grafana:latest`
- Port mappings:
    - Container port: `3000`
    - Protocol: `tcp`
- Click **Add** and then **Create** task definition

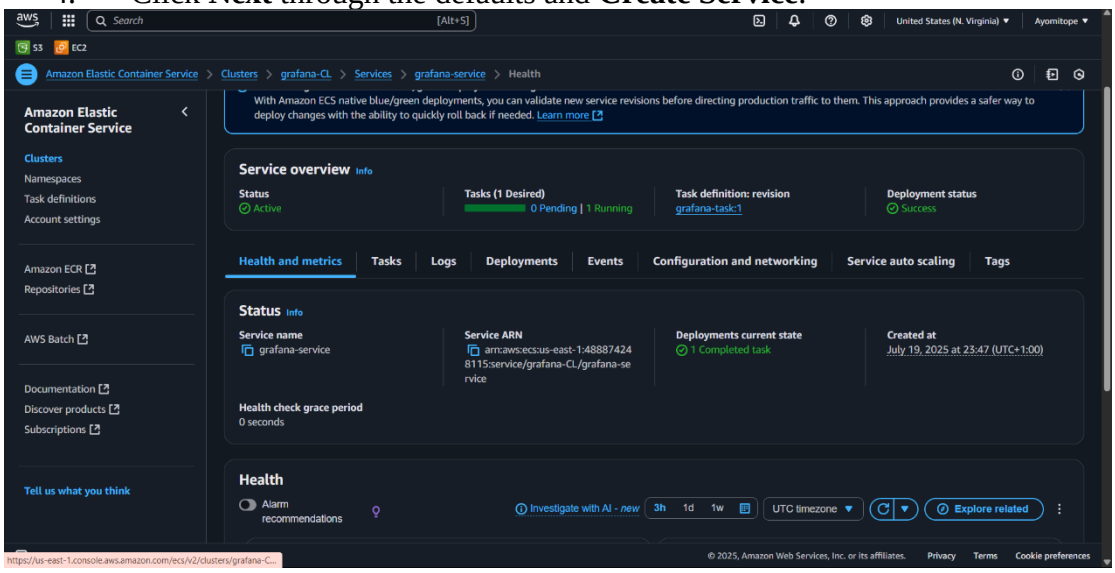






Run the Service

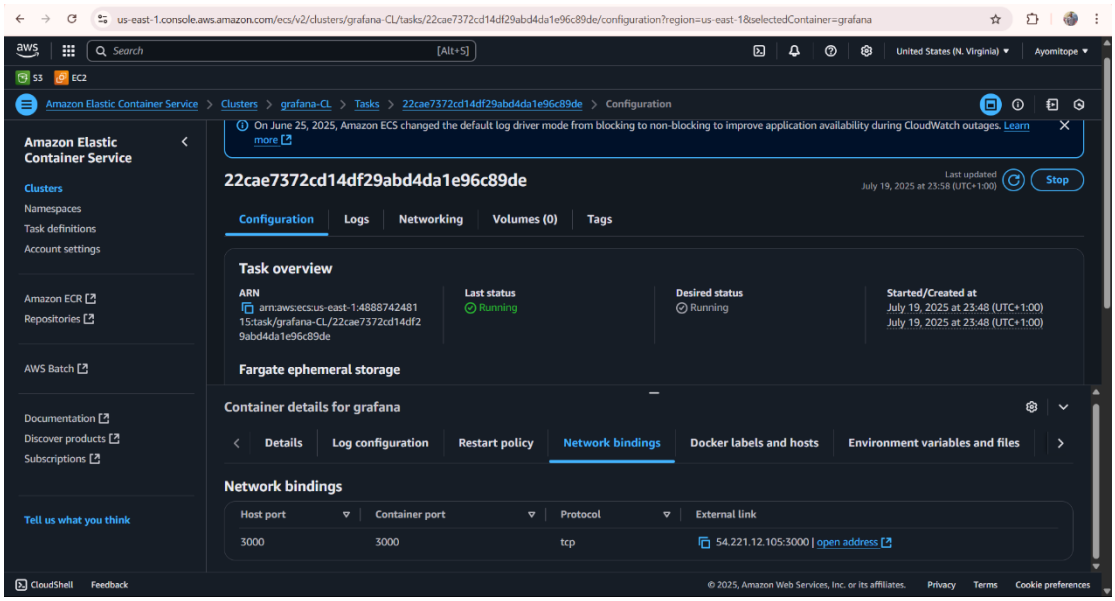
1. In your `grafana-CL` , click **Create** > **Create Service**.
2. Configure:
    - Launch type: **FARGATE**
    - Task Definition: `grafana-task`
    - Service name: `grafana-service`
    - Number of tasks: `1`
3. Under **Networking**:
    - VPC: My *VPC*
    - Subnets: select a **public subnet**
    - Security groups: select `grafana-sg`
4. • Click **Next** through the defaults and **Create Service**.

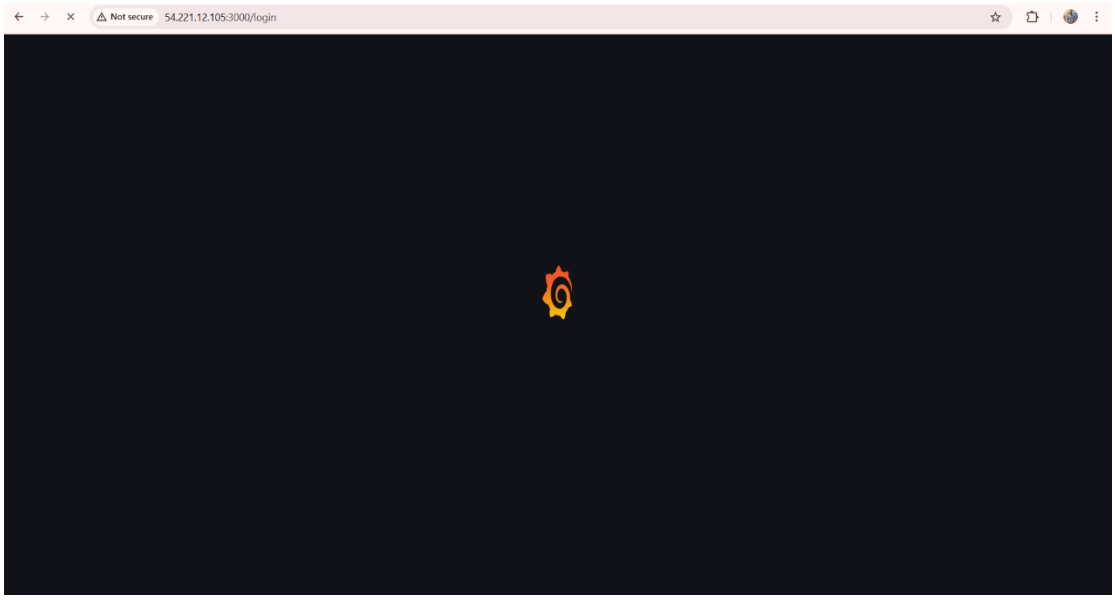


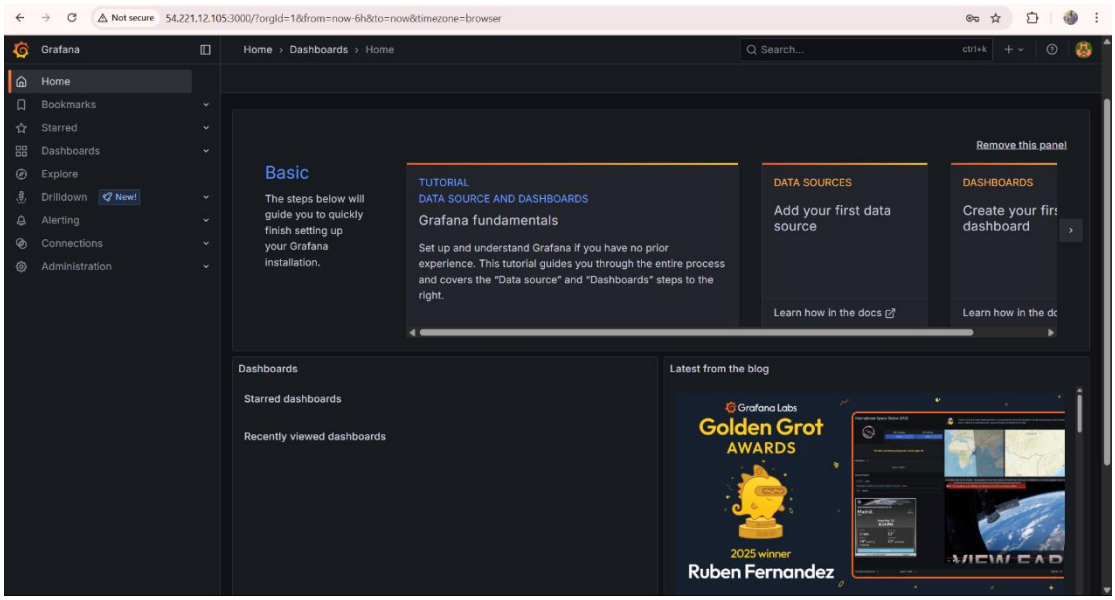
Access Grafana

1. In **EC2 > Network Interfaces**, find the ENI attached to my Grafana task.
2. Copy its **Public IPv4 address**.(54.221.12.105)
3. Open `http://` 54.221.12.105`:3000` in my browser.
4. Login with:

- Username: **admin**
- Password: **admin**


us-east-1.console.aws.amazon.com/ecs/v2/clusters/grafana-CL/tasks/22cae7372cd14df29abd4da1e96c89de/configuration?region=us-east-1&selectedContainer=grafana
Search
[Alt+S]
United States (N. Virginia)
Ayomitope
S3
EC2
Amazon Elastic Container Service
Clusters
grafana-CL
Tasks
22cae7372cd14df29abd4da1e96c89de
Configuration
Amazon Elastic Container Service
Clusters
Namespaces
Task definitions
Account settings
Amazon ECR
Repositories
AWS Batch
Documentation
Discover products
Subscriptions
Tell us what you think
On June 25, 2025, Amazon ECS changed the default log driver mode from blocking to non-blocking to improve application availability during CloudWatch outages. Learn more
22cae7372cd14df29abd4da1e96c89de
Last updated
July 19, 2025 at 23:58 (UTC+1:00)
Stop
Configuration
Logs
Networking
Volumes (0)
Tags
Task overview
ARN
arn:aws:ecs:us-east-1:488874248115:task/grafana-CL/22cae7372cd14df29abd4da1e96c89de
Last status
Running
Desired status
Running
Started/Created at
July 19, 2025 at 23:48 (UTC+1:00)
July 19, 2025 at 23:48 (UTC+1:00)
Fargate ephemeral storage
Container details for grafana
Details
Log configuration
Restart policy
Network bindings
Docker labels and hosts
Environment variables and files
Network bindings
Host port
Container port
Protocol
External link
3000
3000
tcp
54.221.12.105:3000 | open address
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Not secure
54.221.12.105:3000/login

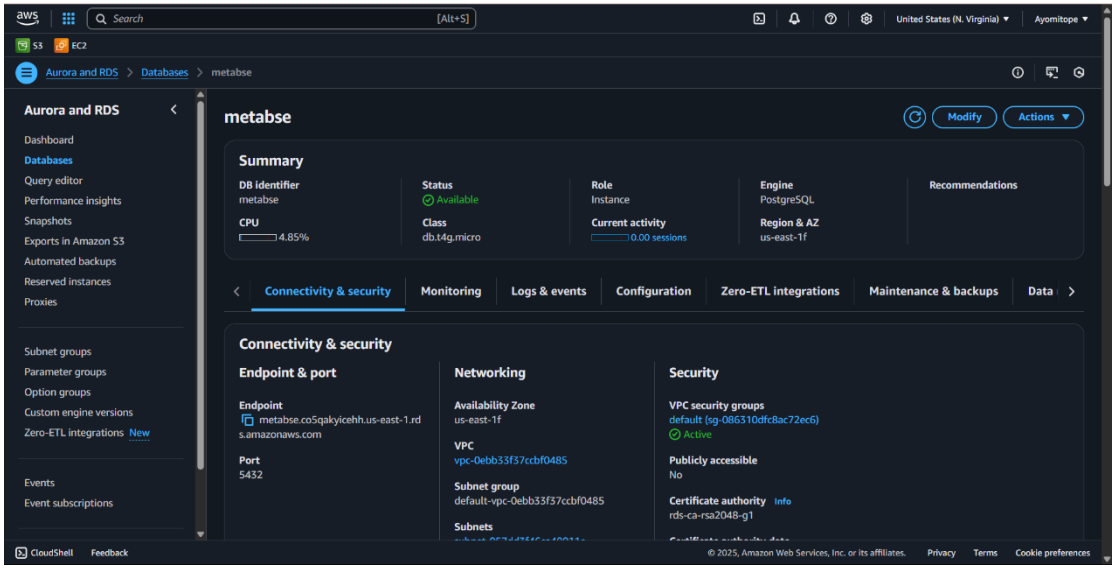


WEEK 6

STEP 1: Set Up the Database (PostgreSQL on RDS)

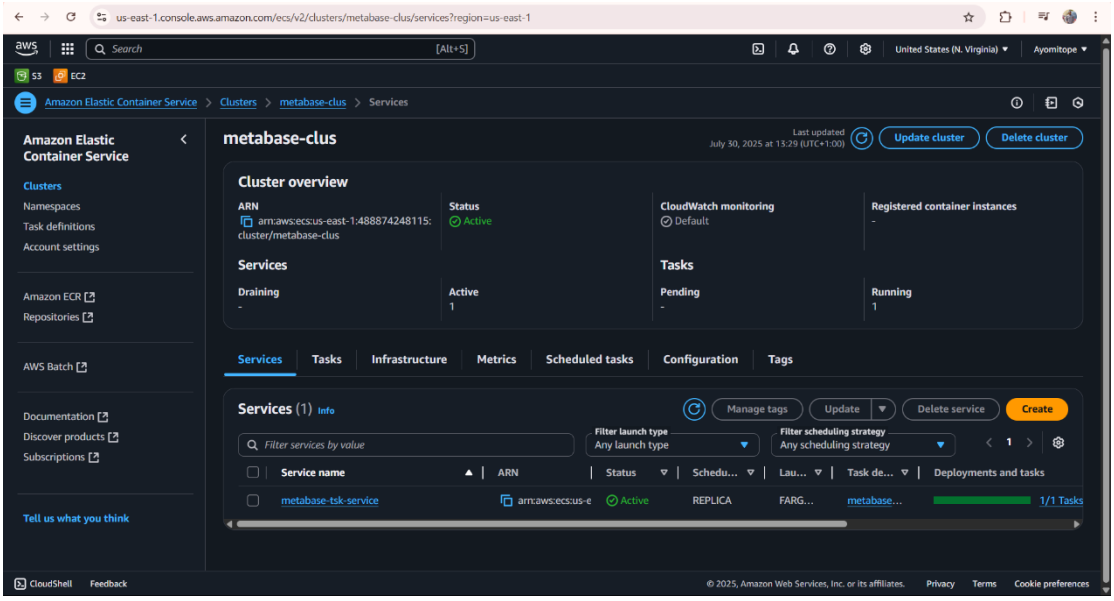
Go to: RDS > Databases > Create database

Choose:
Engine: PostgreSQL
Template: Free Tier (if eligible)
DB instance identifier: metabase-db
Username: metabase_user
Password: Choose and remember it



STEP 2: Create ECS Cluster

Go to: ECS > Clusters > Create Cluster
Choose: “Networking only (Fargate)”
Name it: metabase-cluster



STEP 3: Create Security Groups

ECS Task SG

Name: ecs-task-sg

Inbound Rule: allow TCP 3000 from the Load Balancer SG

Load Balancer SG

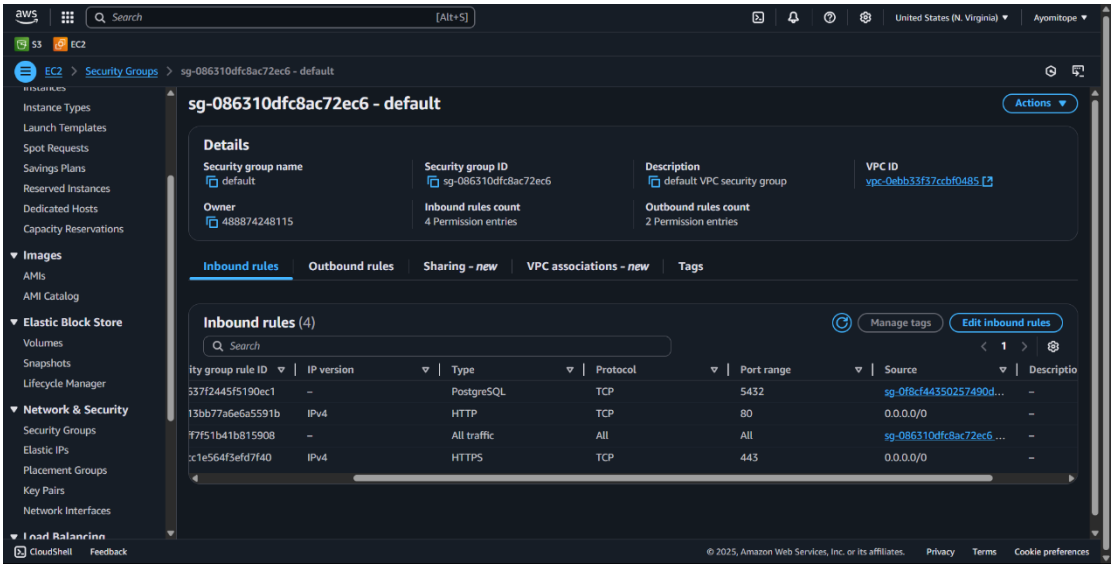
Name: alb-sg

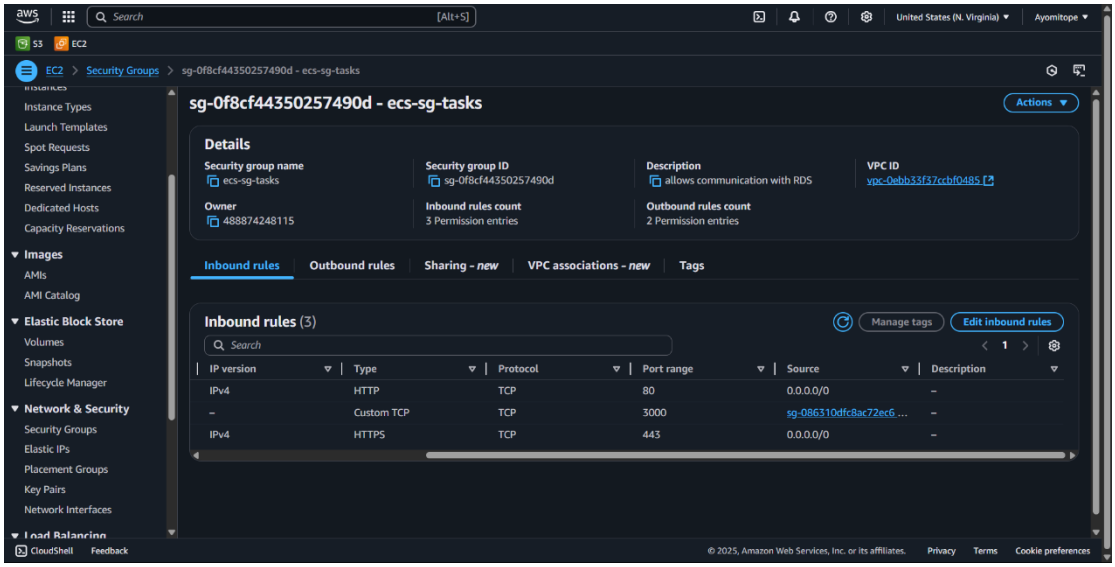
Inbound Rule: allow HTTP 80 from 0.0.0.0/0

RDS SG

Name: rds-sg

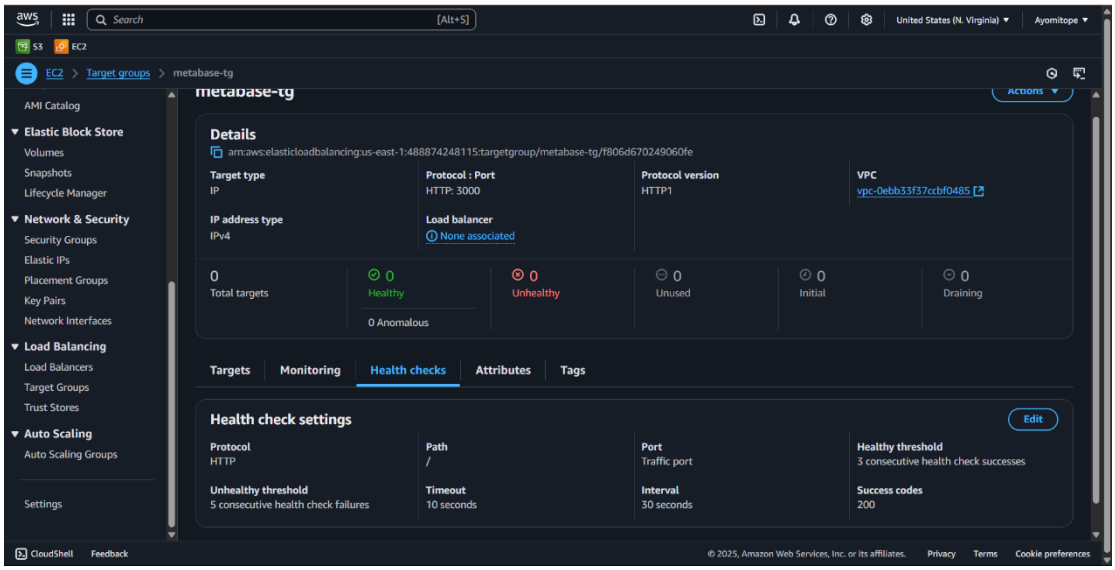
Inbound Rule: allow TCP 5432 from ECS Task SG





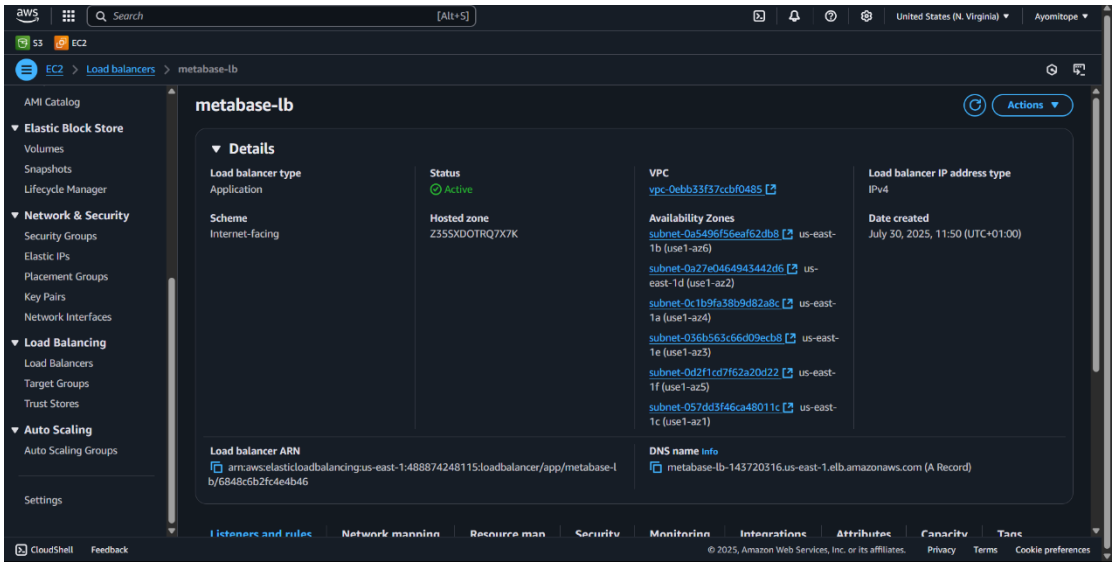
STEP 4: Create Target Group
Go to: EC2 > Target Groups > Create
Name: metabase-tg
Target type: IP
Protocol/Port: HTTP / 3000

Health check settings:
Protocol: HTTP
Path: /
Port: traffic port
Healthy threshold: 3
Unhealthy threshold: 5
Timeout: 10s
Interval: 30s



STEP 5: Create Load Balancer
Go to: EC2 > Load Balancers > Create Application Load Balancer

Name: metabase-lb
Scheme: internet-facing
Listeners: HTTP (port 80)
Select 2 public subnets
Security group: alb-sg
Forwarding rule:
Port 80 → Target group: metabase-tg

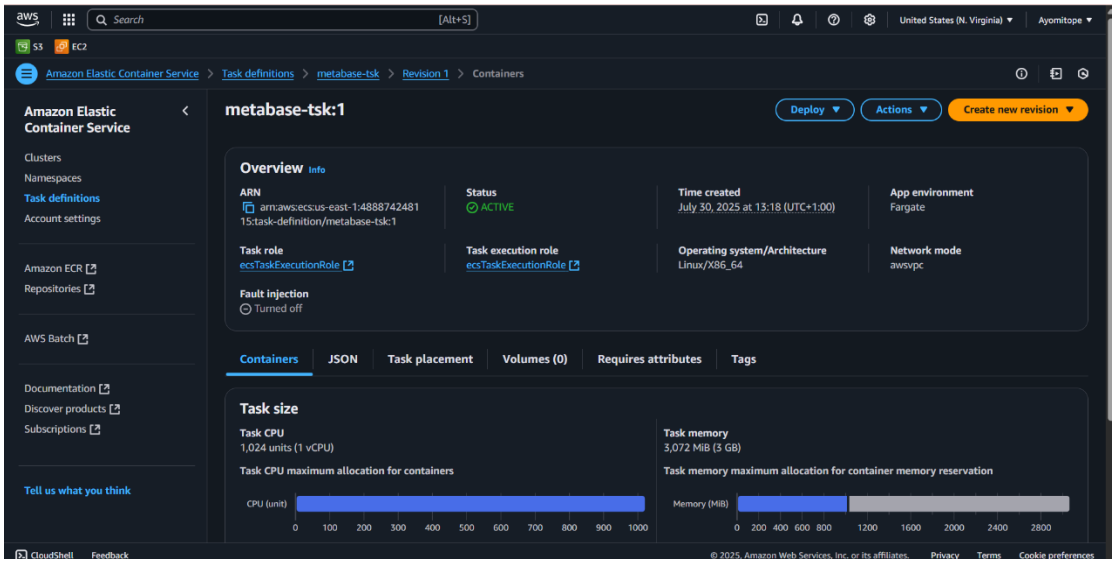


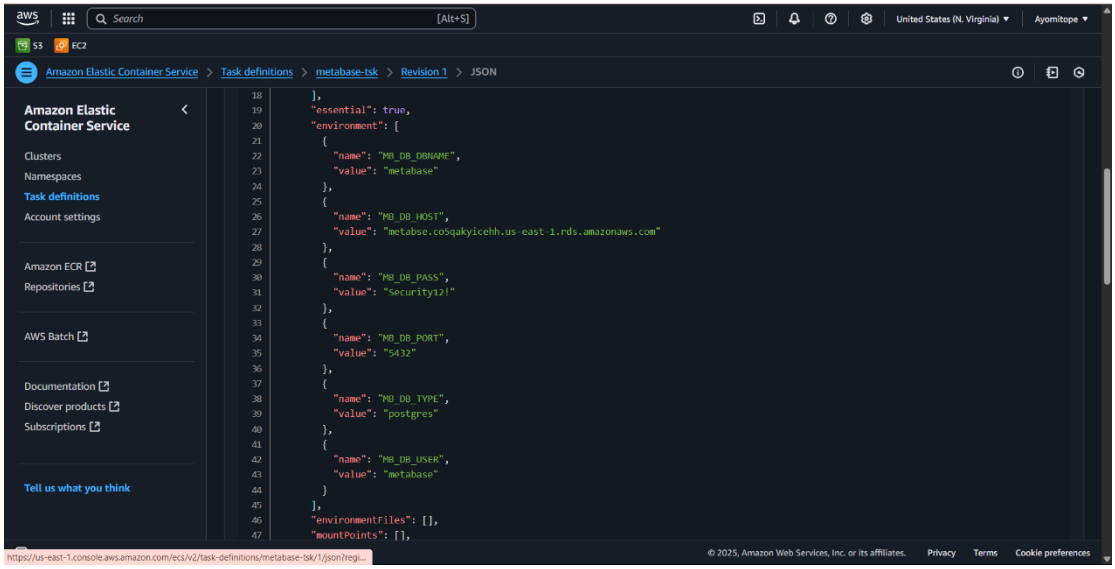
STEP 6: Create Task Definition (Fargate)

Go to ECS > Task Definitions > Create new task definition
Launch type: Fargate
Task name: metabase-tsk
CPU/Memory: 1 vCPU, 3 GB
Add container:
Name: metabase
Image: metabase/metabase:latest
Port mapping: 3000

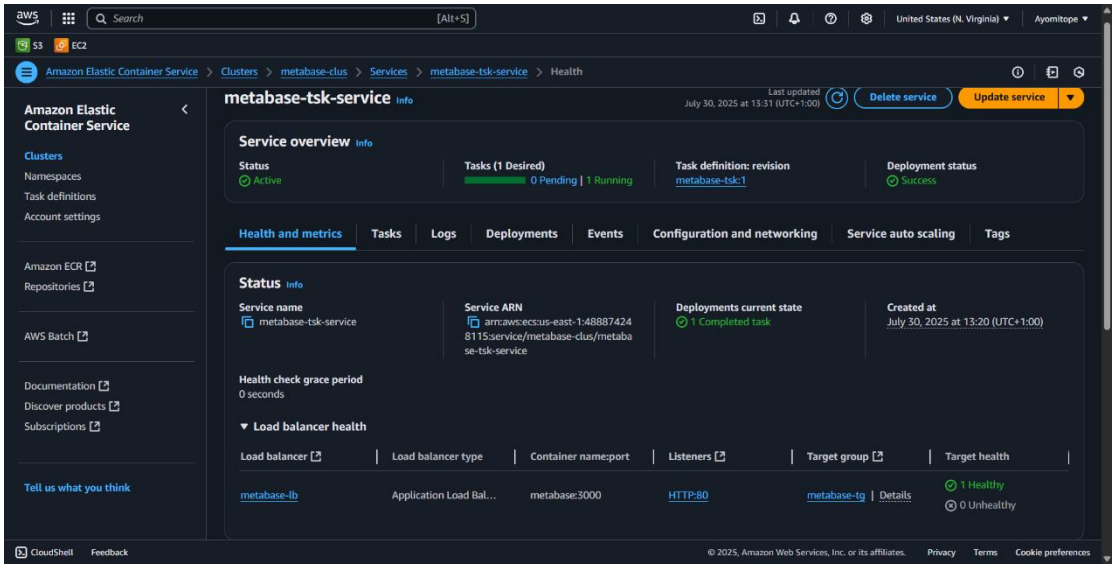
Environment Variables:
MB_DB_TYPE=postgres
MB_DB_DBNAME=metabase
MB_DB_PORT=5432
MB_DB_USER=metabase_user
MB_DB_PASS=your_password
MB_DB_HOST=your-rds-endpoint.rds.amazonaws.com

```
"healthCheck": {
  "command": ["CMD-SHELL", "curl -f http://localhost:3000 || exit 1"],
  "interval": 30,
  "timeout": 5,
  "retries": 3,
  "startPeriod": 60
}
```

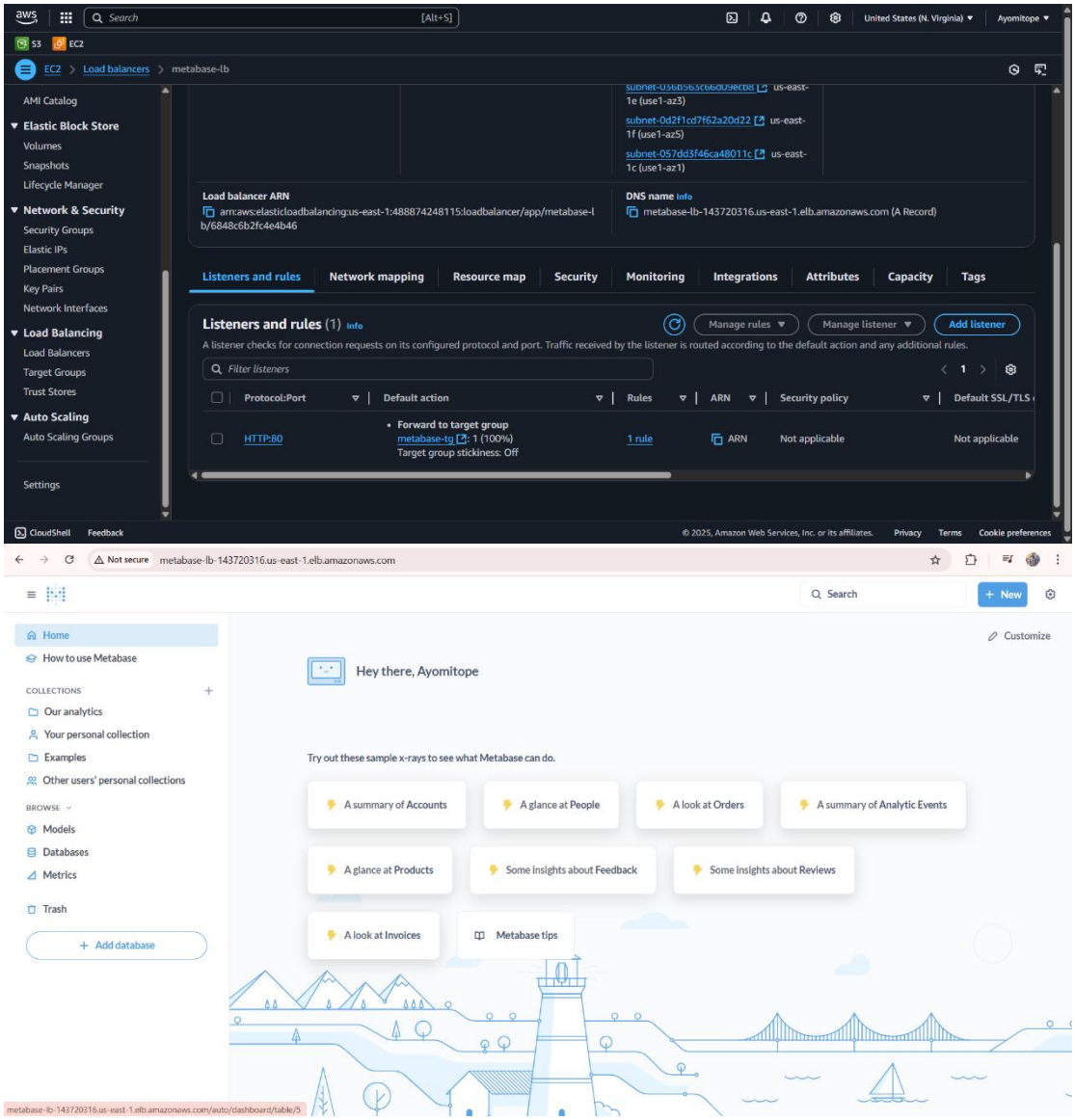




STEP 7: Create ECS Service
Go to ECS > Clusters > metabase-cluster > Create service
Launch type: Fargate
Task Definition: metabase-tsk
Service name: metabase-service
Number of tasks: 1
VPC: Same as your RDS and LB
Subnets: Select 2 public subnets
Security Group: ecs-task-sg
Enable Load Balancer integration:
Type: Application Load Balancer
Listener: HTTP 80
Target Group: metabase-tg
Click Deploy Service



Go to the Load Balancer, copy the DNS name and paste it in a new tab



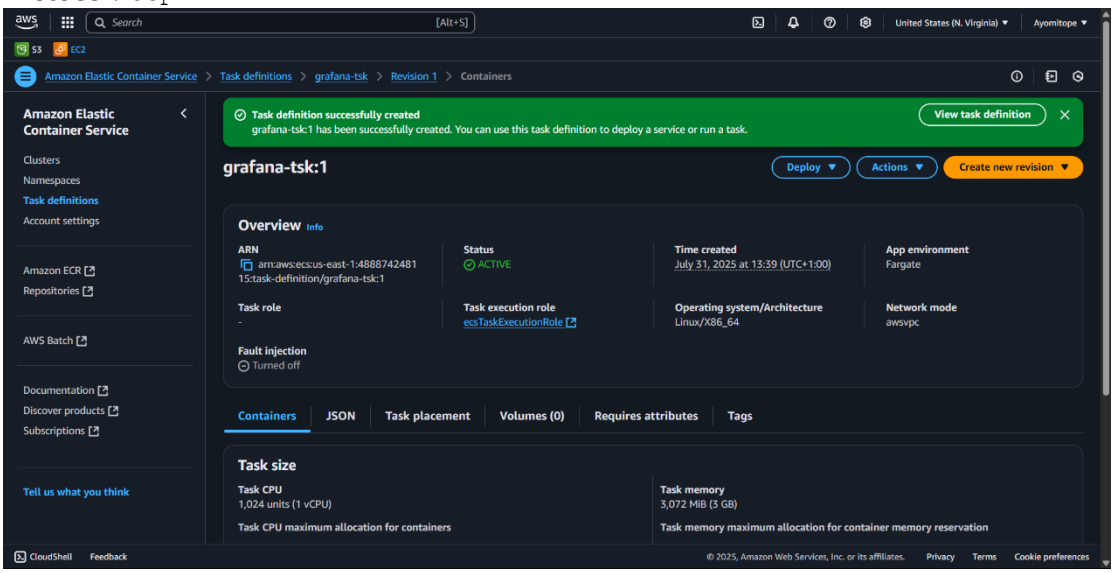
WEEK 7

Create a Task Definition

1. Go to **ECS > Task Definitions** > **Create new Task Definition**.
2. Choose **FARGATE**, click **Next**.
3. Configure:
    - Task Definition Name: `grafana-tk`
    - Task Role: *leave blank*
    - Network Mode: `awsvpc`
    - CPU & Memory: 1 vCPU/ **3 GB**

Under **Container definitions**, click **Add container**:
Name: `Grafana`
Image URI: `grafana/grafana:latest`
Port mappings

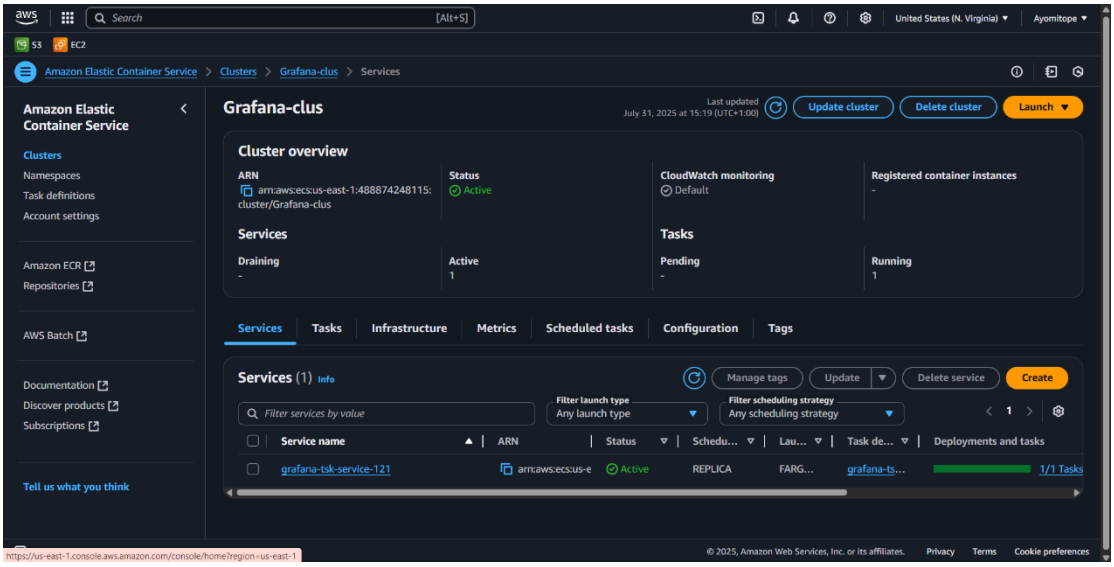
Container port: `3000`
Protocol: `tcp`



Create an ECS Cluster

- Go to **ECS > Clusters** > **Create Cluster**.

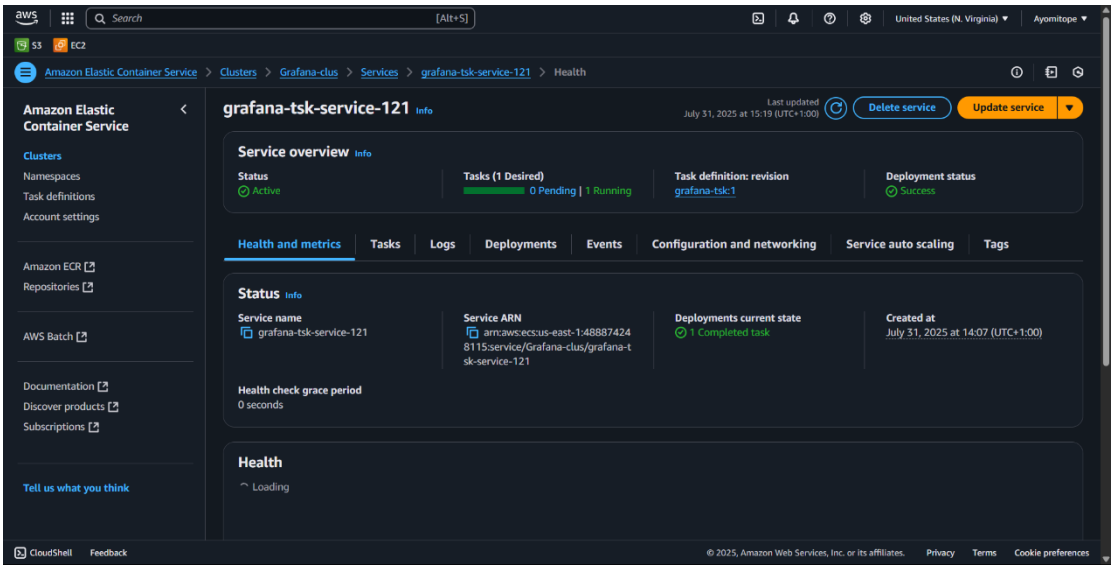
- Choose “Networking only (Fargate)” type.
- Named the cluster.



Create a Service to Run the Task

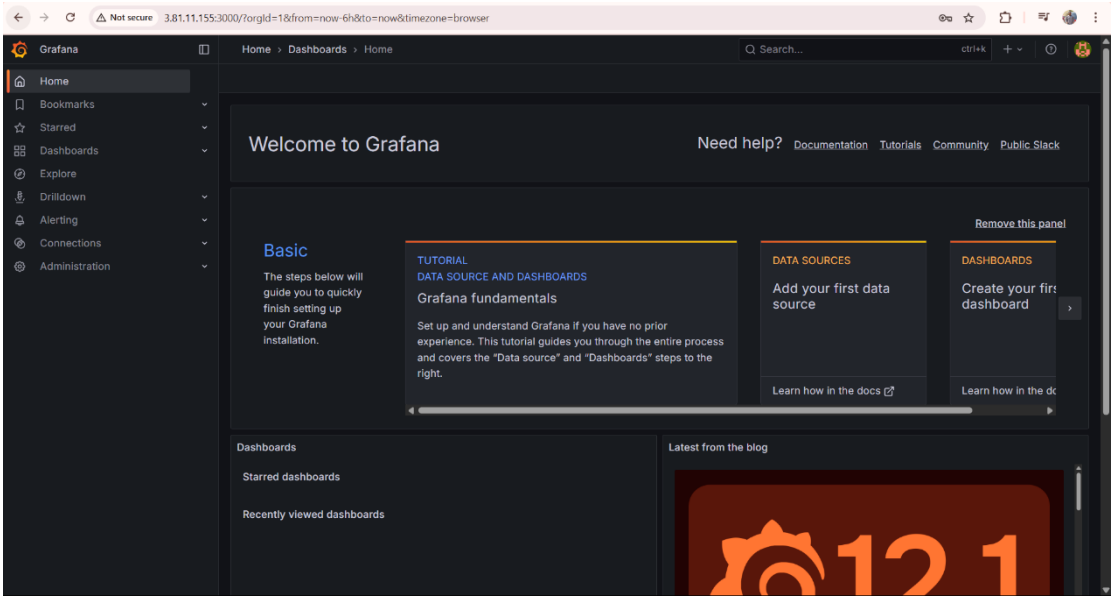
- In my cluster, create a **Service**:

- Launch type: Fargate.
- Task Definition: Select the one you created.
- Desired tasks: `1`.
- Select VPC and public subnets.
- Enable **Auto-assign Public IP**



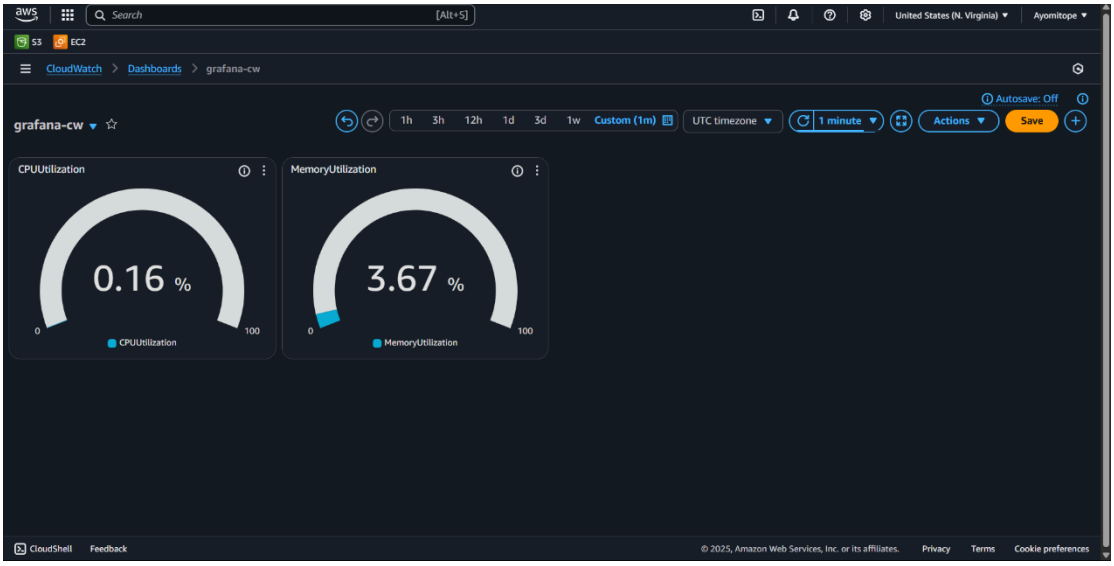
Access Grafana

1. In **ECS > Network Interfaces**, find the ENI attached to my Grafana task.
2. Copy its **Public IPv4 address**.(3.81.11.155)
3. Open `http://` 3.81.11.155`:3000` in my browser.



Create CloudWatch Metrics Dashboard

- Go to **CloudWatch > Dashboards** > **Create Dashboard**.
- Add 2 widgets:
- **Metric: ECS > Per-Task Metrics > CPUUtilization**
- **Metric: ECS > Per-Task Metrics > MemoryUtilization**

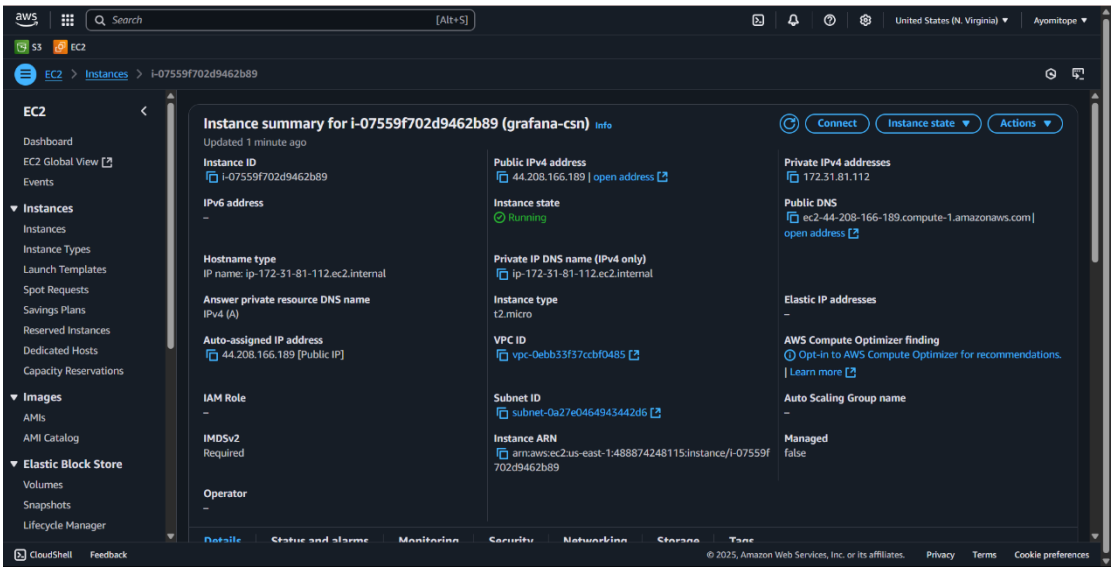


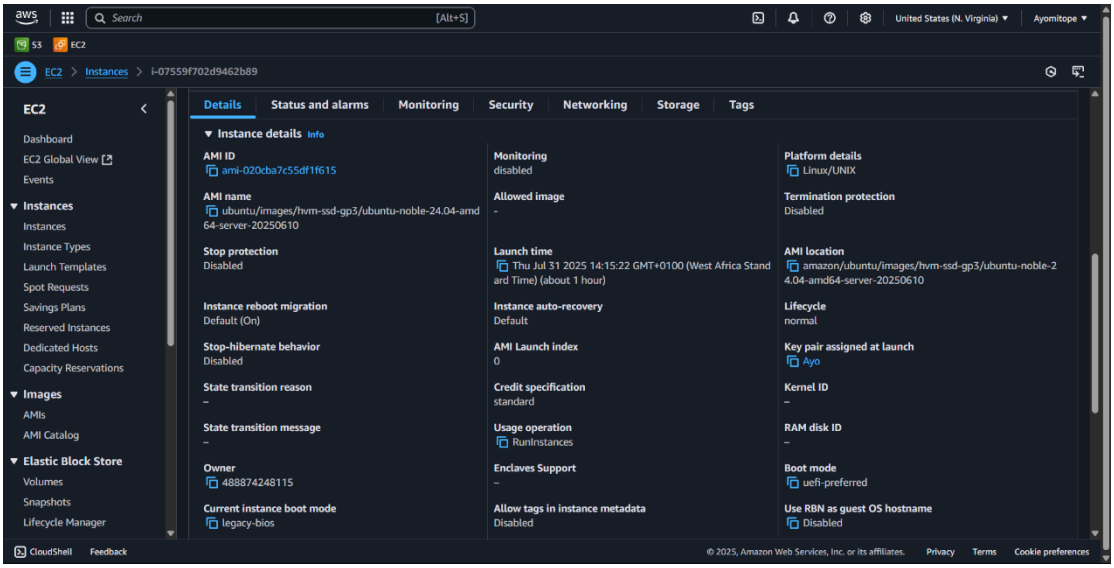
Create an EC2 Instnace

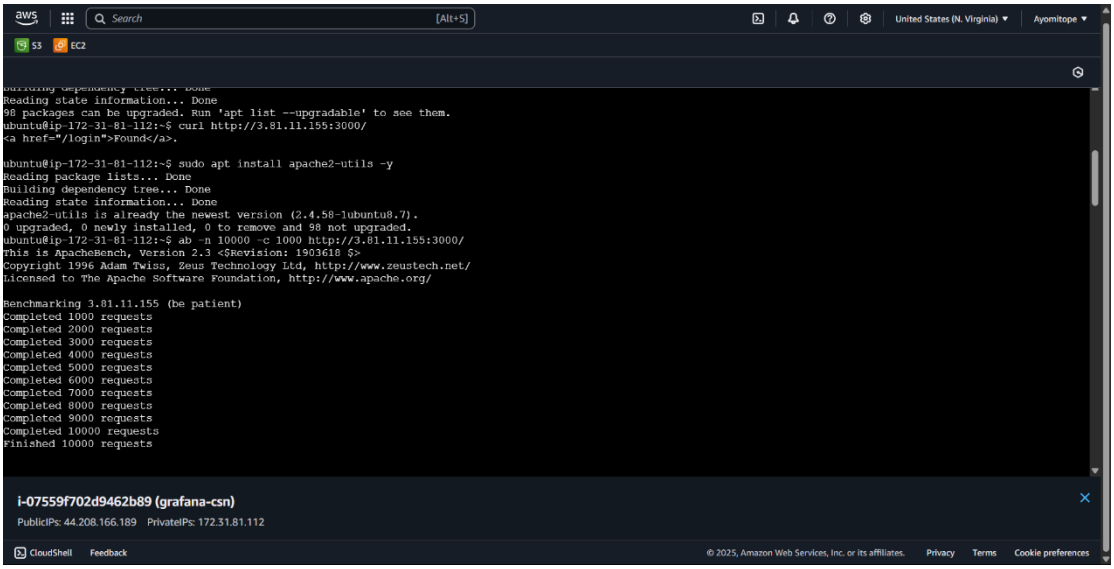
- Name: Grafana:csn
- **AMI**: Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
- **Instance Type**: t2.micro & created networking settings(ticked HTTPS, SSH, HTTP)
- Launch Instance.

Connected to the instance through HTTPS

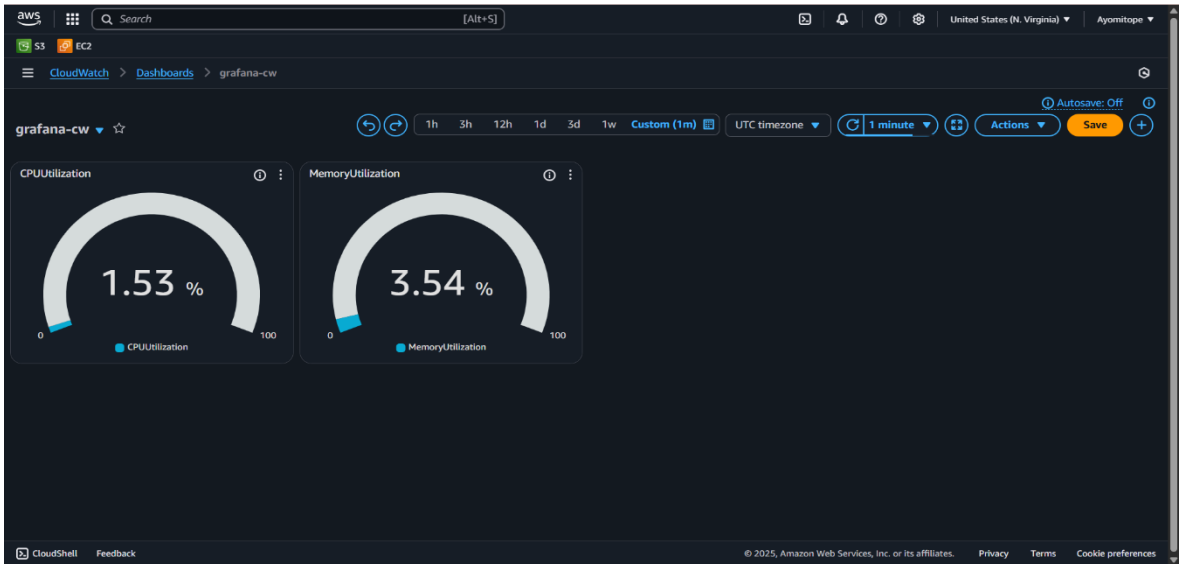
- Here are my commands for Ubuntu:
  * sudo apt update
  * sudo apt install apache2-utils -y
  * ab -n 10000 -c 1000 http://3.81.11.155:3000/
  *sudo apt install wrk -y
  *wrk -t12 -c1000 -d30s http://3.81.11.155:3000
  *wrk -t12 -c1000 -d60s http://3.81.11.155:3000



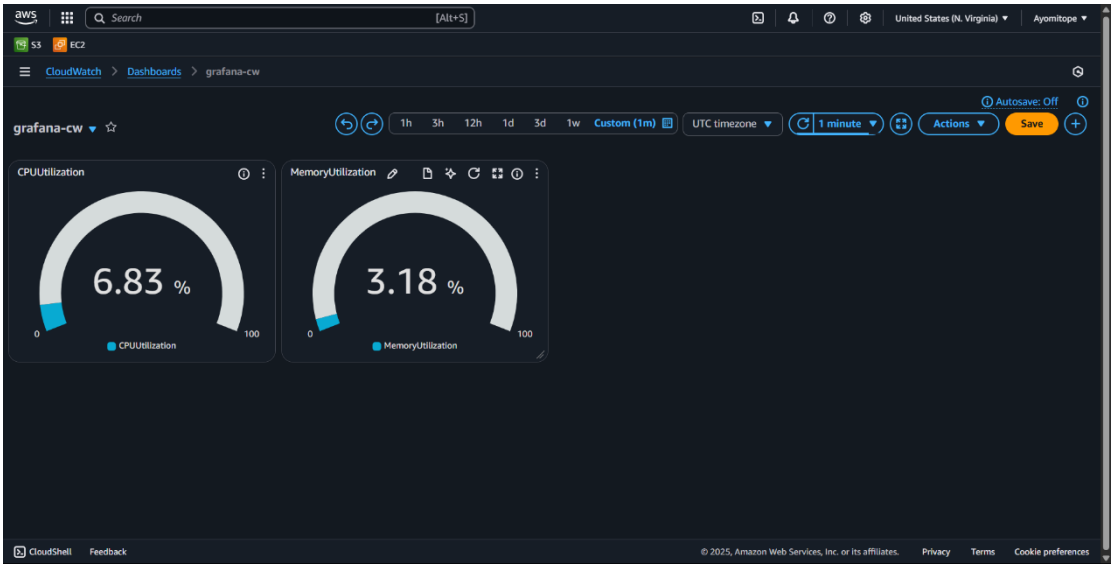


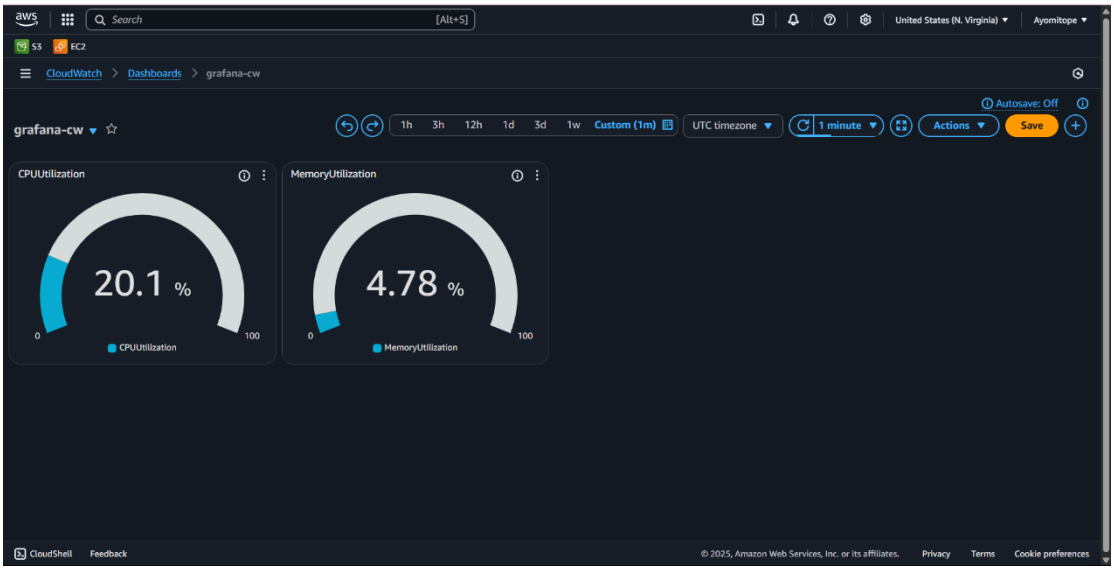


The change in the **CPUUtilization and MemoryUtilization**



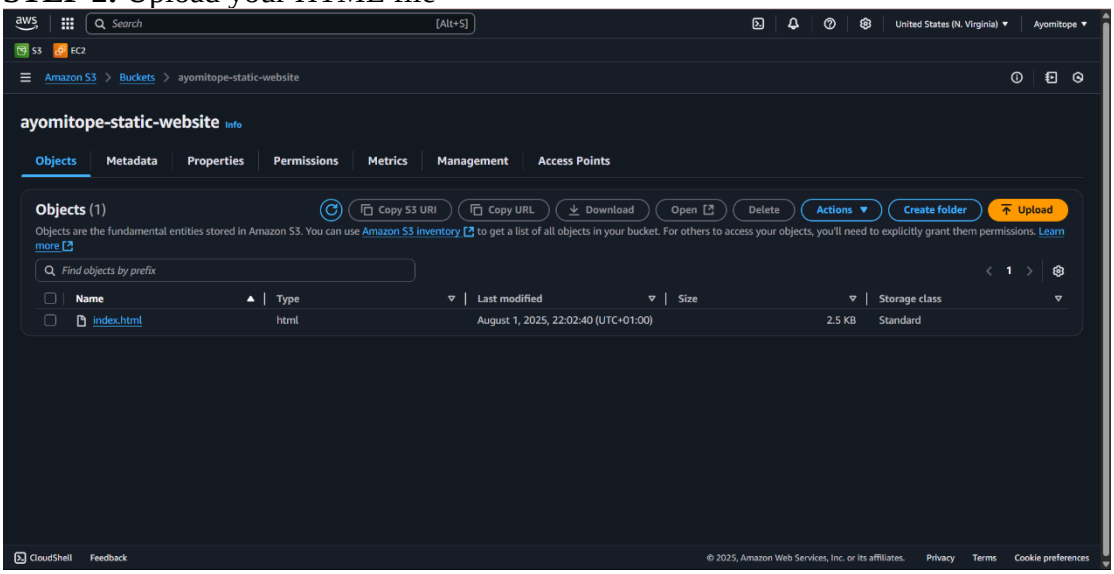
The changes in the **CPUUtilization and MemoryUtilization**



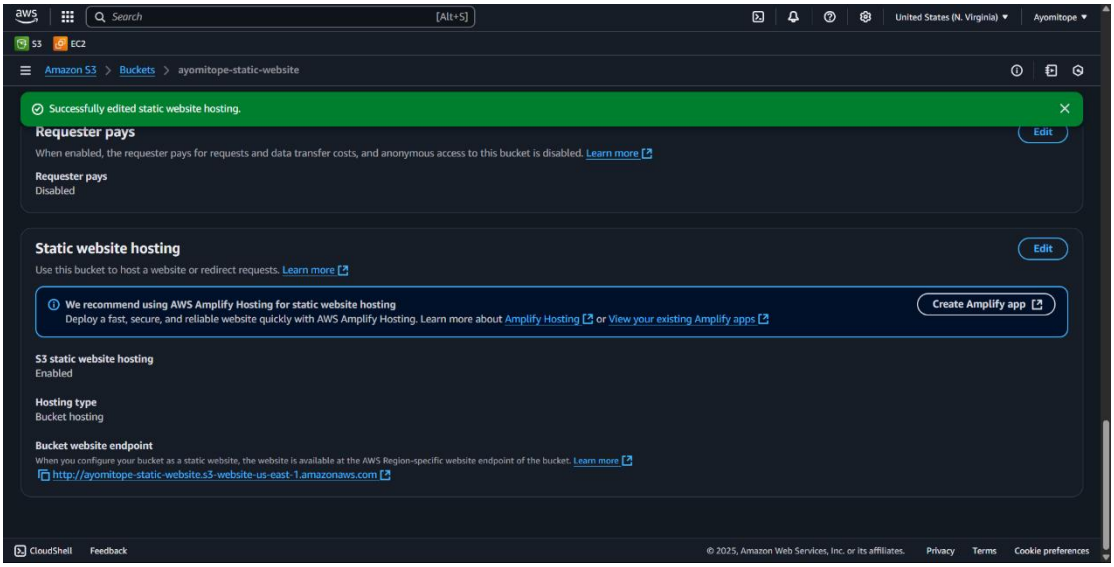


WEEK 8

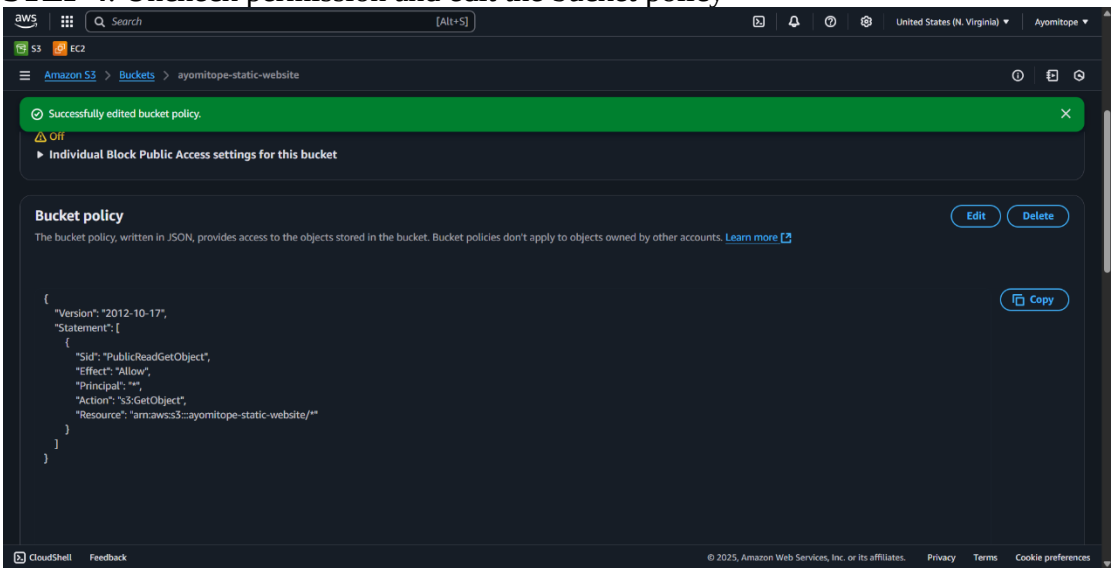
STEP 1: Create an S3 Bucket
STEP 2: Upload your HTML file

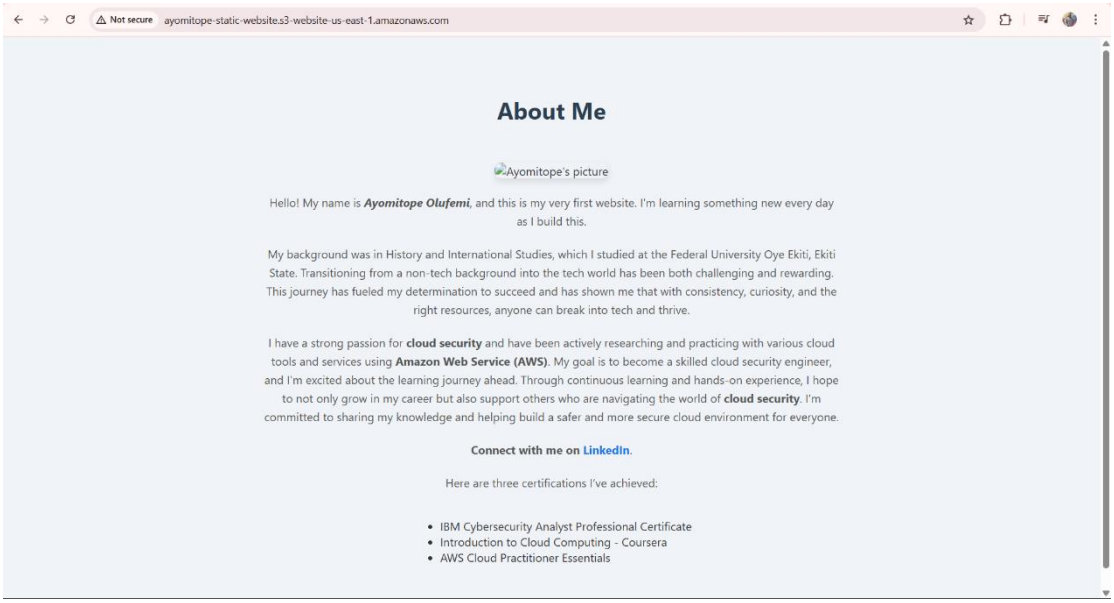


STEP 3: Enable Static website

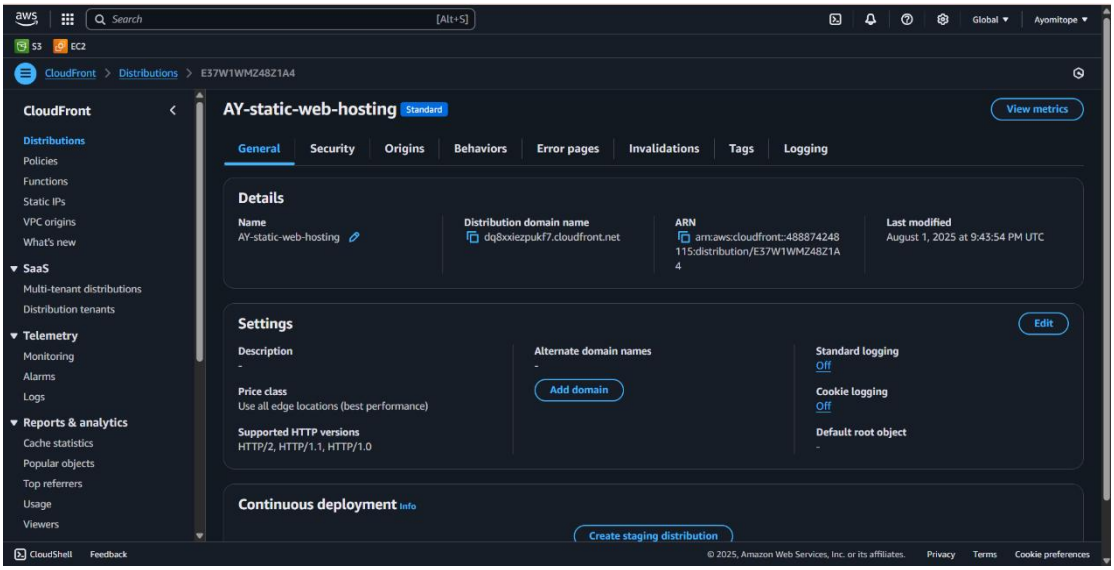


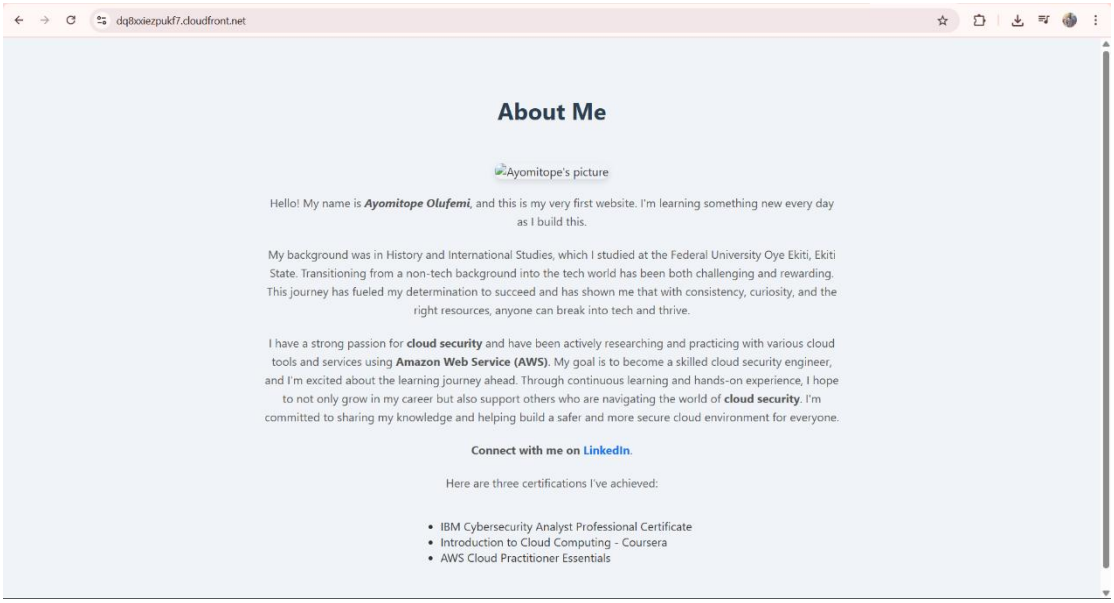
STEP 4: Uncheck permission and edit the bucket policy





STEP 5: Create a CloudFront Distribution to serve your site securely

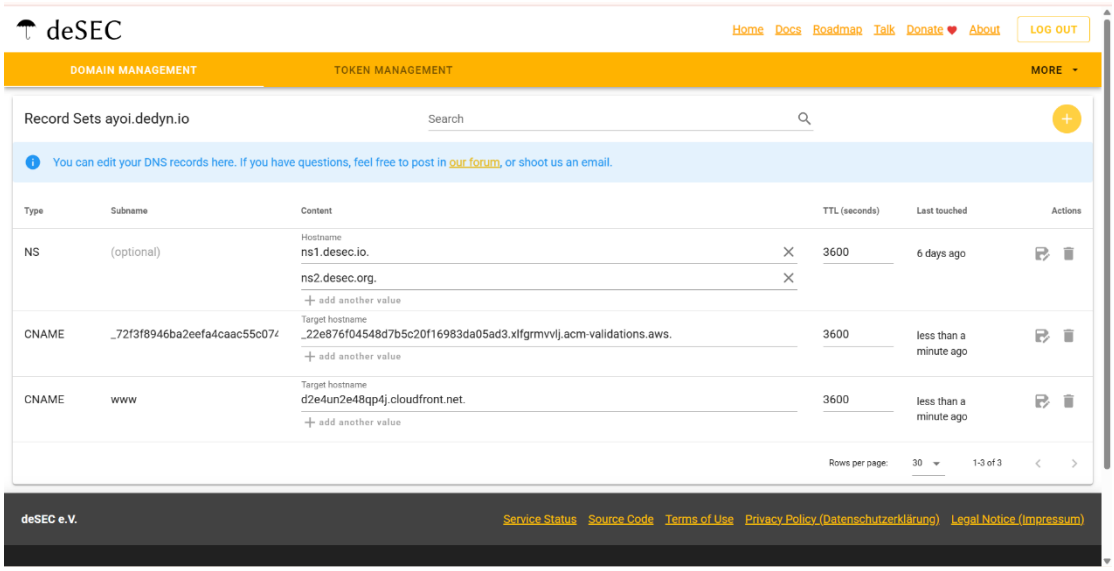




WEEK 9

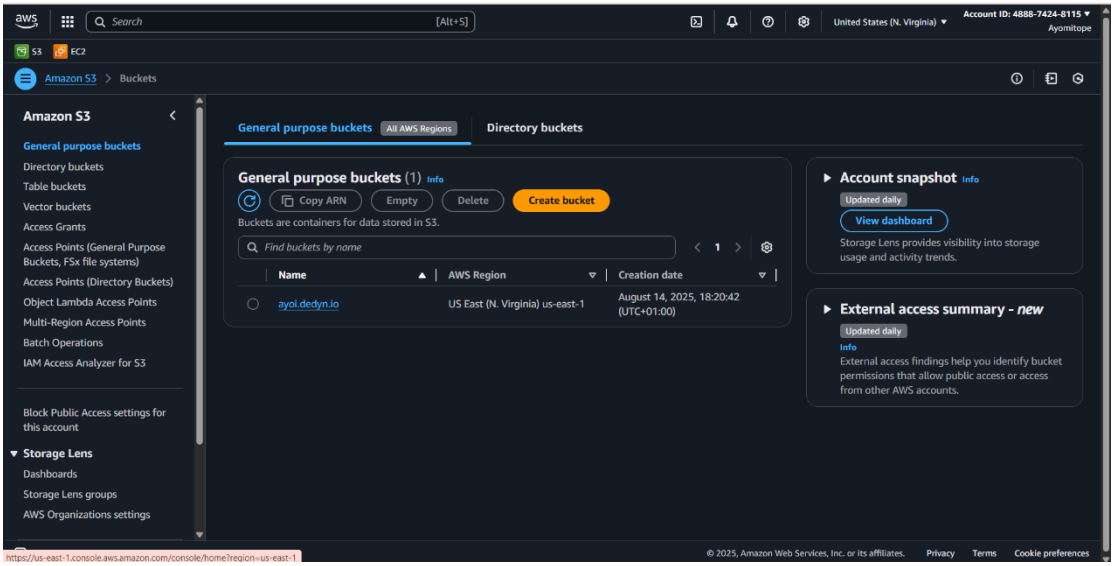
STEP 1: Set up my domain on deSEC (dedyn.io)

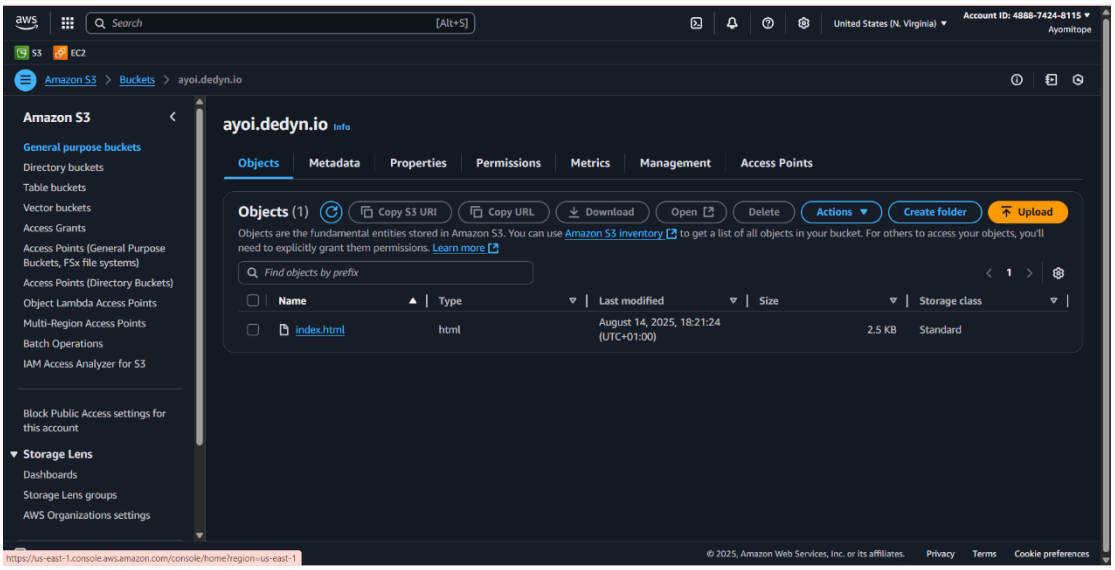
- Created a **deSEC account** and registered the subdomain **ayoi.dedyn.io**.
- Managed all DNS records for this domain in deSEC (dedyn.io).

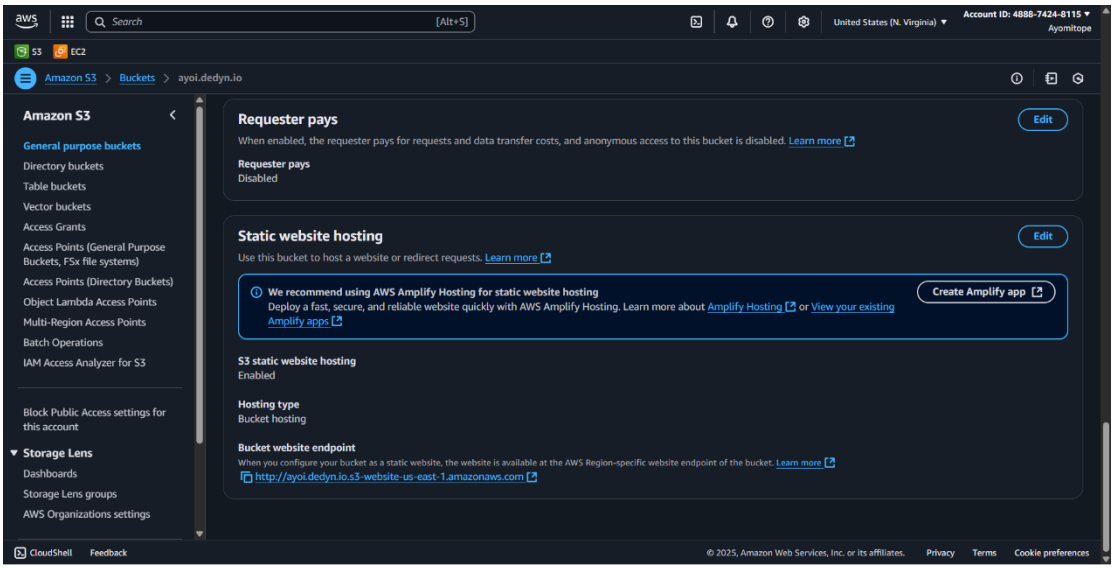


STEP 2: Create the S3 Bucket for Website Hosting

- In **S3 Console**, created a bucket named exactly as the domain: `ayoi.dedyn.io`.
- Disabled **Block Public Access** (if serving directly from S3).
- Uploaded the website files (`index.html`, `style.css`, etc.).
- Enabled **Static website hosting** in **Properties**, setting `index.html` as the root document.

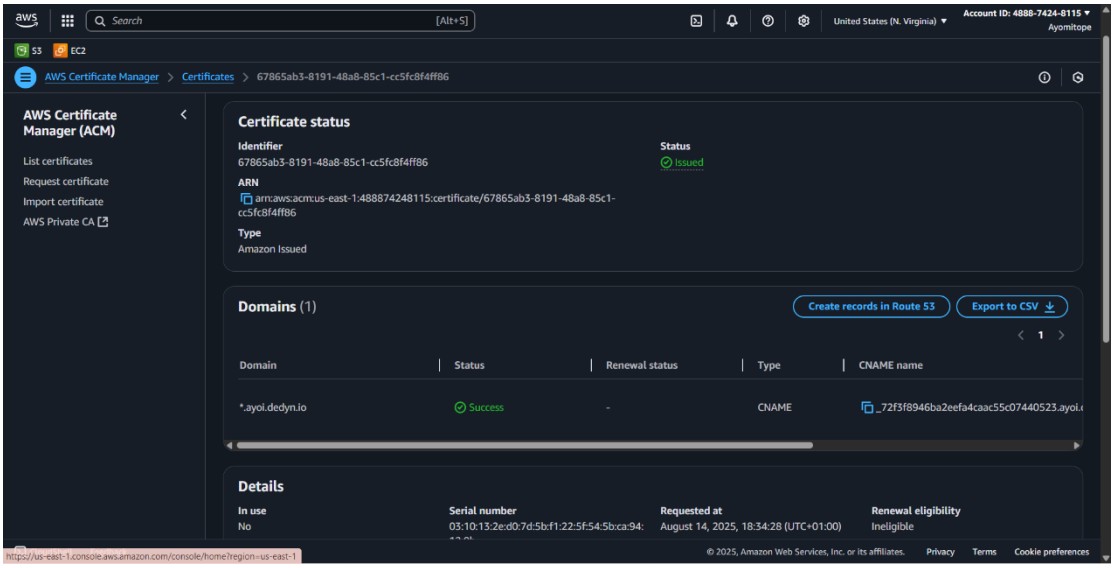






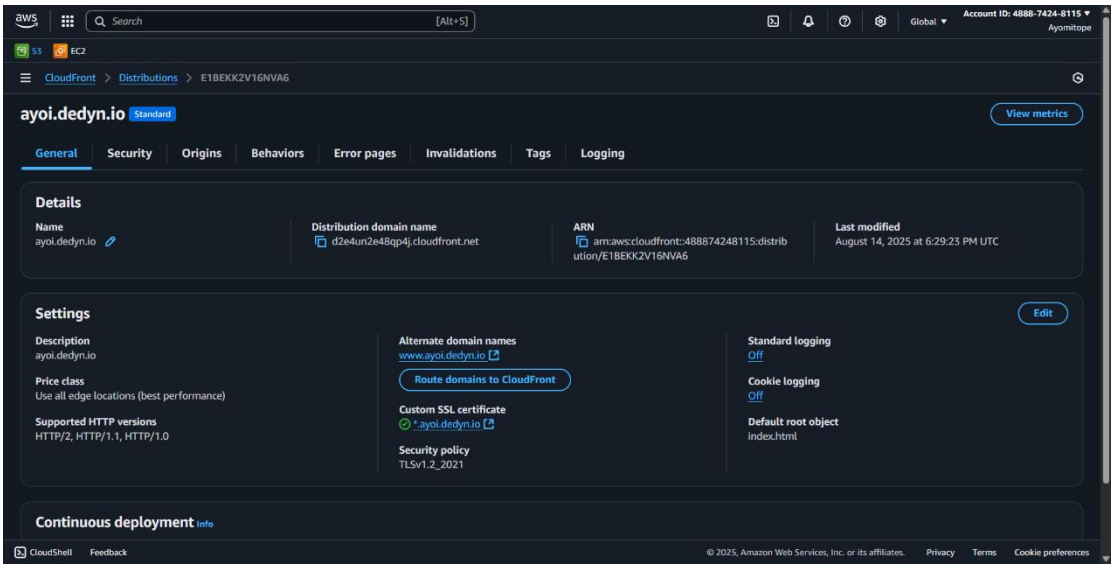
STEP 3: Request an SSL/TLS Certificate in AWS Certificate Manager

- Opened **AWS Certificate Manager (ACM)** in the **us-east-1** region (required for CloudFront).
- Requested a public certificate for:
  - `*.ayoi.dedyn.io` (wildcard for subdomains)
- Chose **DNS validation** and added the provided CNAME record in **Route 53**.
- Waited for the status to change to **Issued**.



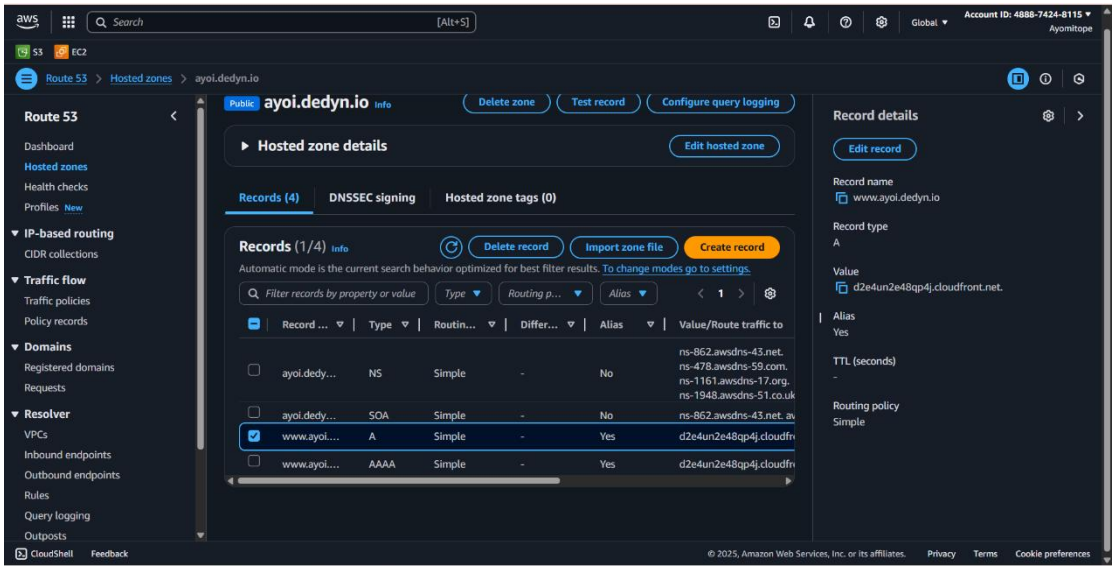
STEP 4: Create a CloudFront Distribution

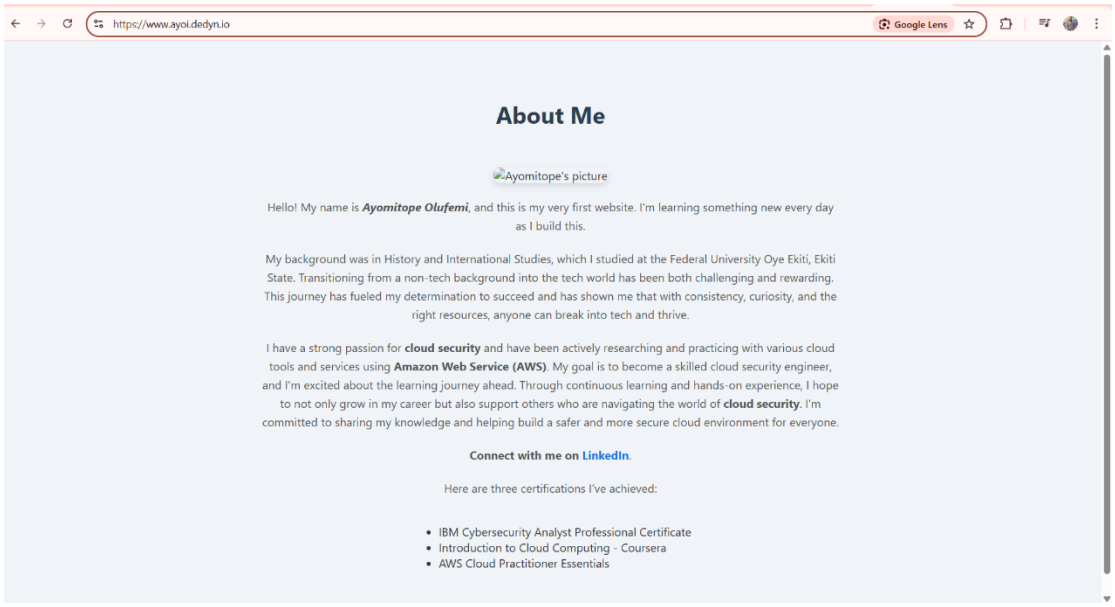
- In **CloudFront Console**, created a new distribution.
- **Origin domain**: Selected the S3 bucket's static website endpoint (or bucket name if using OAI/OAC for private content).
- **Viewer protocol policy**: Redirect HTTP to HTTPS.
- **Custom SSL certificate**: Selected the ACM certificate for `*.ayoi.dedyn.io`
- **Alternate domain names (CNAMEs)**: Added `www.ayoi.dedyn.io`
- Created the distribution and waited for deployment.



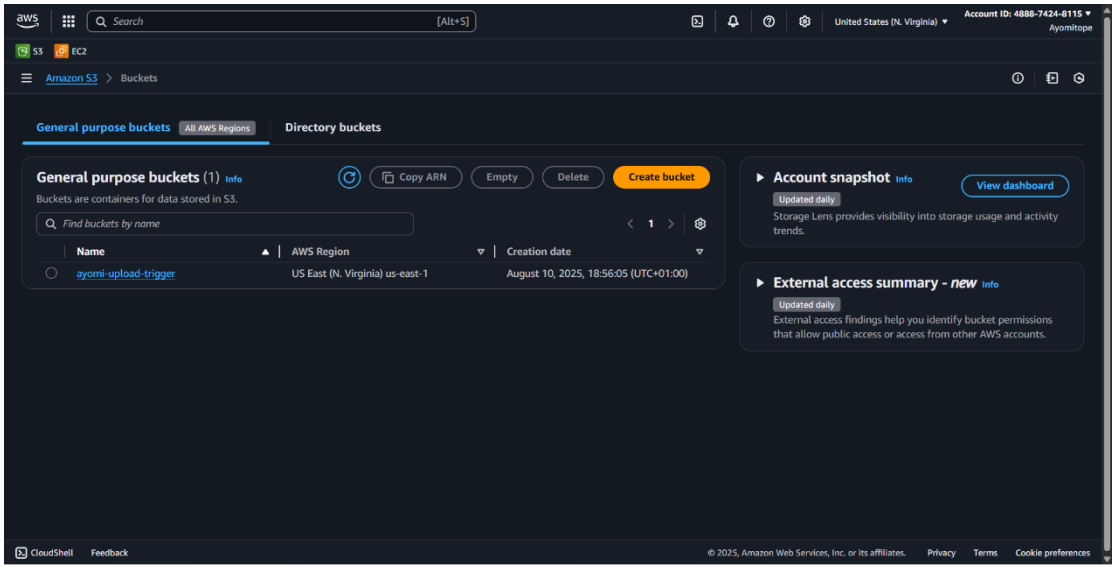
STEP 5: Create a Route 53 Hosted Zone and DNS Records

- In **Route 53**, created a hosted zone for `dedyn.io` if not already existing (or used existing if managed externally).
- Created an **A record (Alias)** pointing `ayoi.dedyn.io` to the CloudFront distribution.
- This ensures that when users type `ayoi.dedyn.io`, they are routed through CloudFront to the S3 bucket.

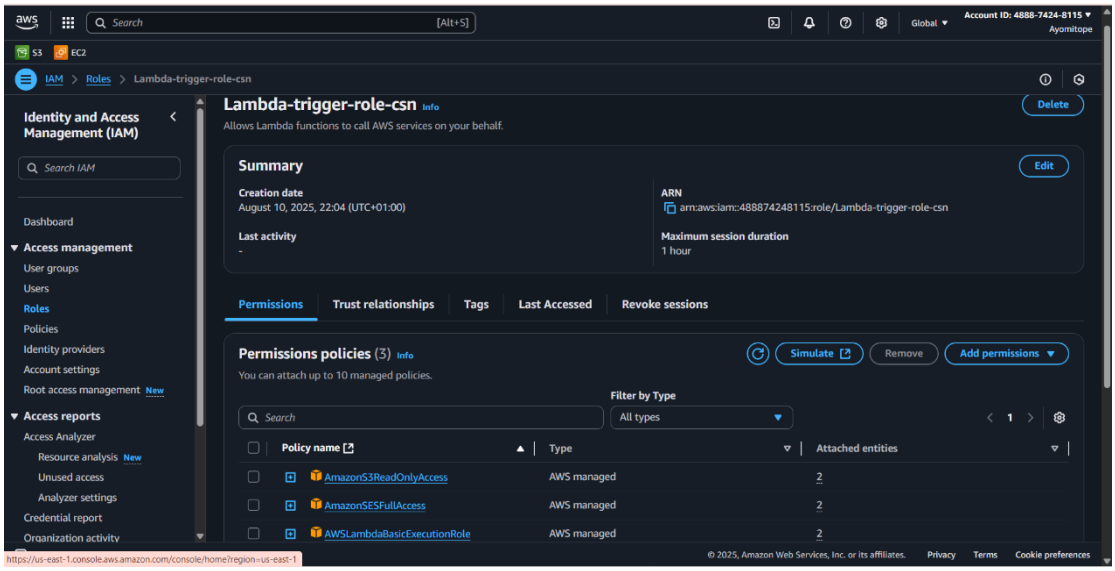




WEEK 10
Step 1: Create an S3 Bucket
Go to the AWS Management Console and open the Amazon S3 service
Click "Create bucket"
Name: ayomi-upload-trigger
Choose your preferred AWS Region
Keep all other settings as default and click "Create bucket"

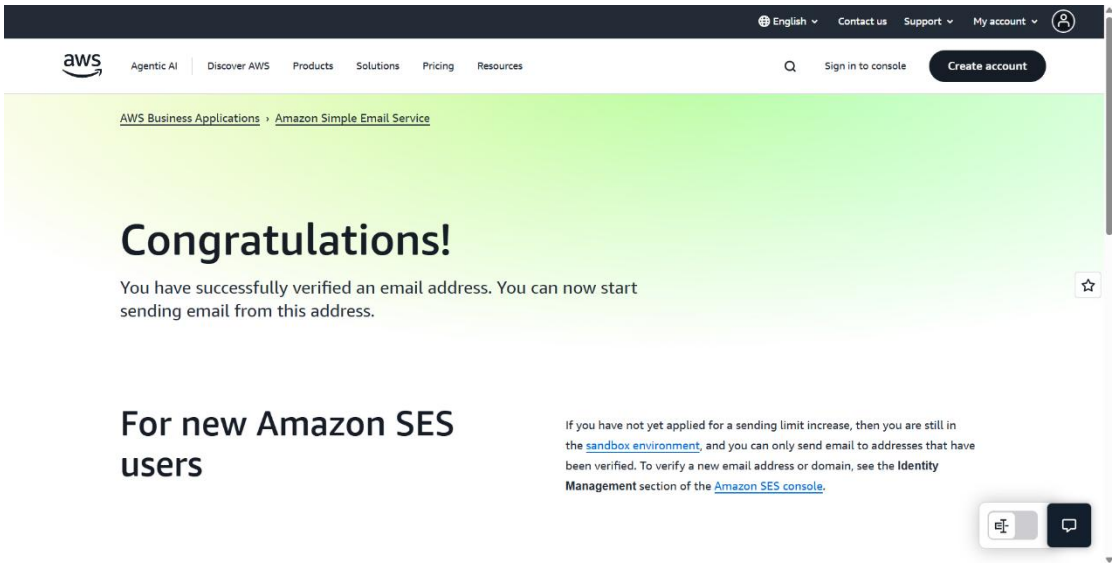


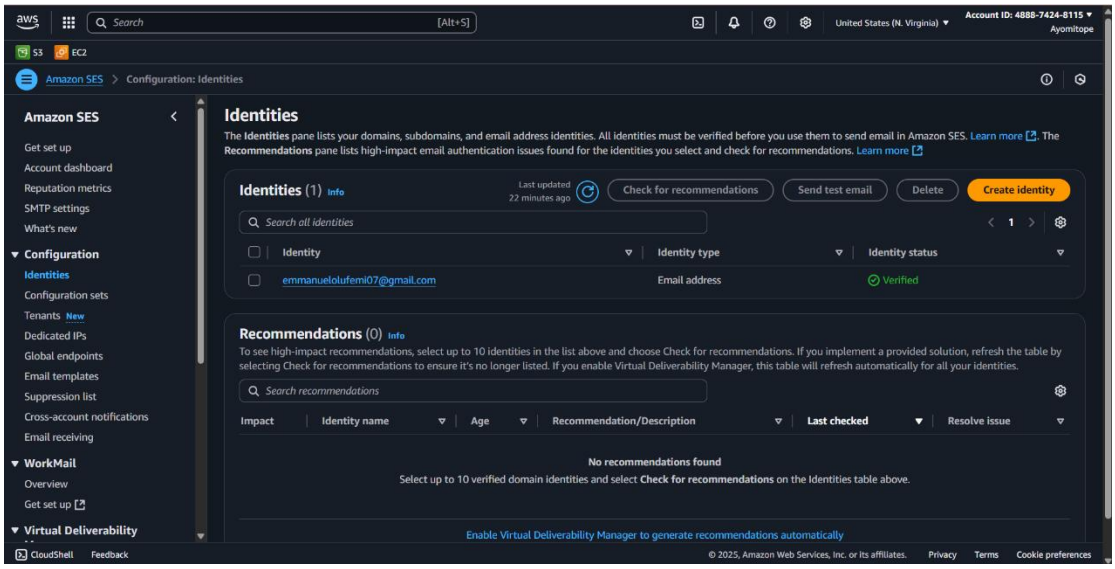
Step 2: Create an IAM Role for Lambda
Click "Roles" then "Create role"
Select "AWS service" as trusted entity type
Choose "Lambda" as the use case
Click "Next"
Attach the following policies:
AWSLambdaBasicExecutionRole (for CloudWatch logging)
AmazonS3ReadOnlyAccess (to read S3 event data)
AmazonSESFullAccess (SES for email notifications)
Click "Next"
Name the role: lambda-trigger-role-csn



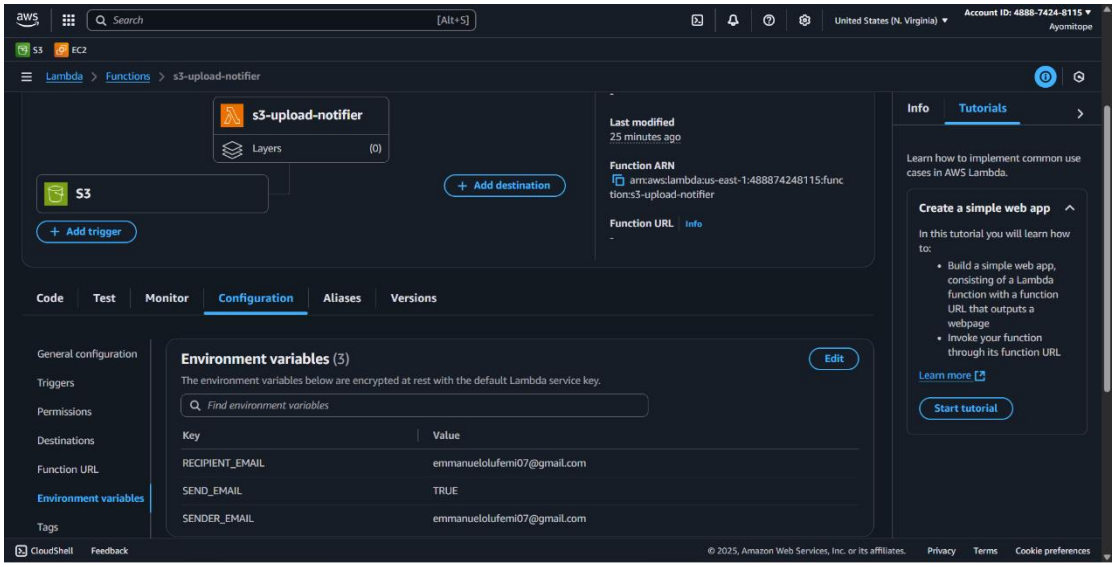
Step 3: Verify SES Email Addresses
Go to the Amazon SES service
Click "Verified identities" in the left menu
Click "Create identity"
Choose "Email address" and enter your sender email

Click "Create identity"
Check your email inbox for a verification email and click the link





Step 4: Create the Lambda Function
Go to the AWS Lambda service
Click "Create function"
Choose "Author from scratch"
Enter a function name (e.g., "s3-upload-notifier")
Select Python 3.13 as runtime
Under "Permissions", choose "lambda-trigger-role-csn" and select the role you created
Click "Create function"

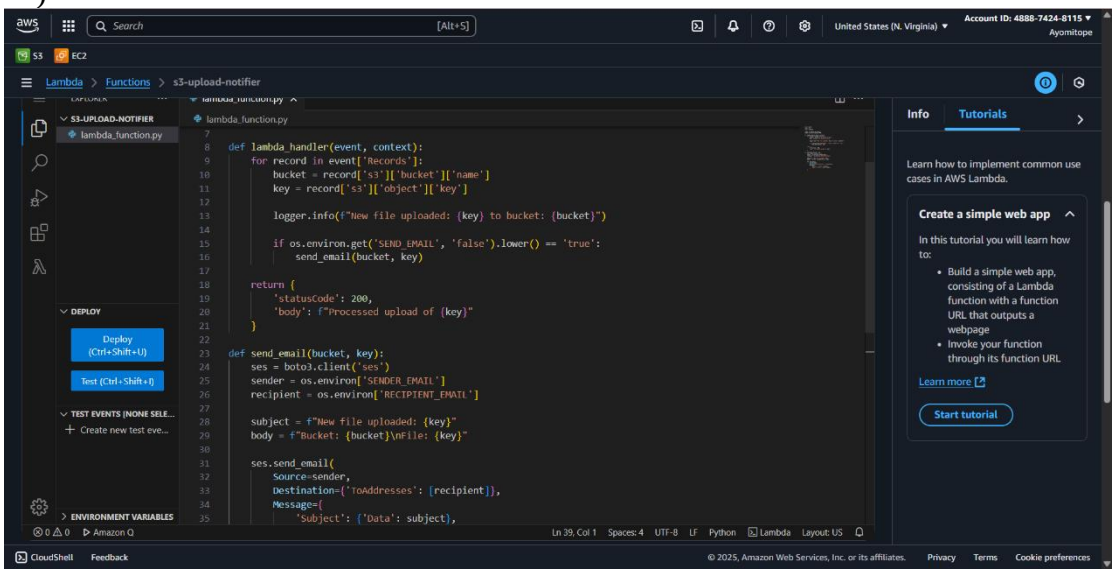


Step 5: Write the Lambda Function Code
Here's a Python Lambda function that logs the file name and sends an email:

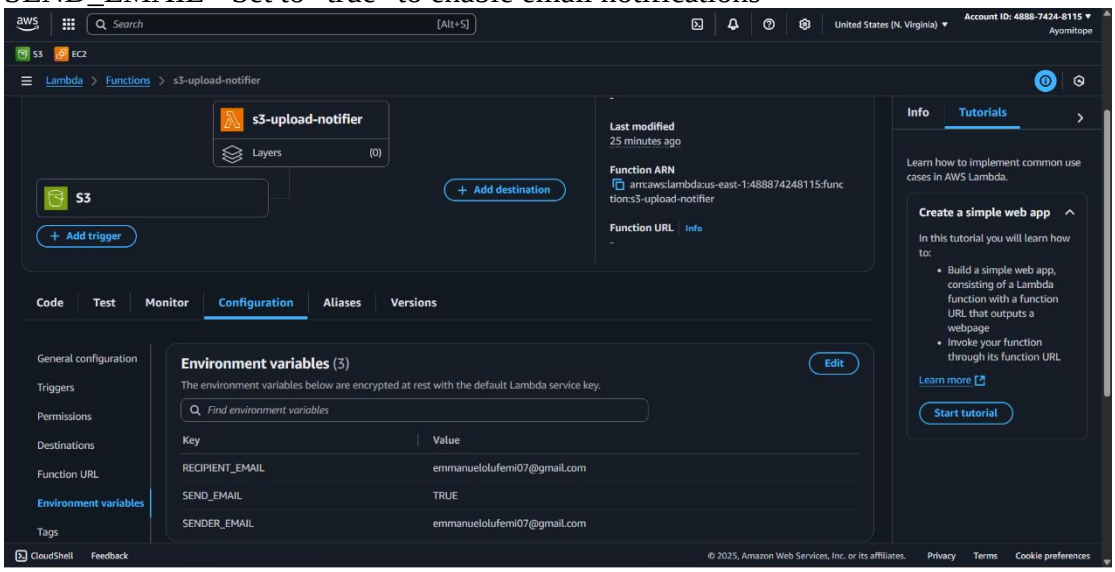
```
import boto3

import os

import logging


logger = logging.getLogger()

logger.setLevel(logging.INFO)


def lambda_handler(event, context):

    for record in event['Records']:

        bucket = record['s3']['bucket']['name']

        key = record['s3']['object']['key']


        logger.info(f"New file uploaded: {key} to bucket: {bucket}")


        if os.environ.get('SEND_EMAIL', 'false').lower() == 'true':

            send_email(bucket, key)
```

```
    return {
        'statusCode': 200,
        'body': f"Processed upload of {key}"
    }

def send_email(bucket, key):
    ses = boto3.client('ses')
    sender = os.environ['SENDER_EMAIL']
    recipient = os.environ['RECIPIENT_EMAIL']

    subject = f"New file uploaded: {key}"
    body = f"Bucket: {bucket}\nFile: {key}"

    ses.send_email(
        Source=sender,
        Destination={'ToAddresses': [recipient]},
        Message={
            'Subject': {'Data': subject},
            'Body': {'Text': {'Data': body}}
        }
```

)

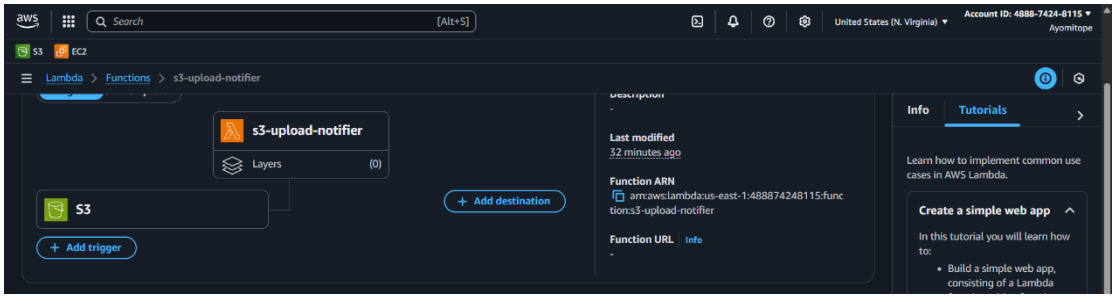


Step 6: Configure Lambda Environment Variables
In your Lambda function, go to the "Configuration" tab
Click "Environment variables"
Add the following variables:
SENDER_EMAIL - Your verified SES sender email
RECIPIENT_EMAIL - The email address to receive notifications
SEND_EMAIL - Set to "true" to enable email notifications

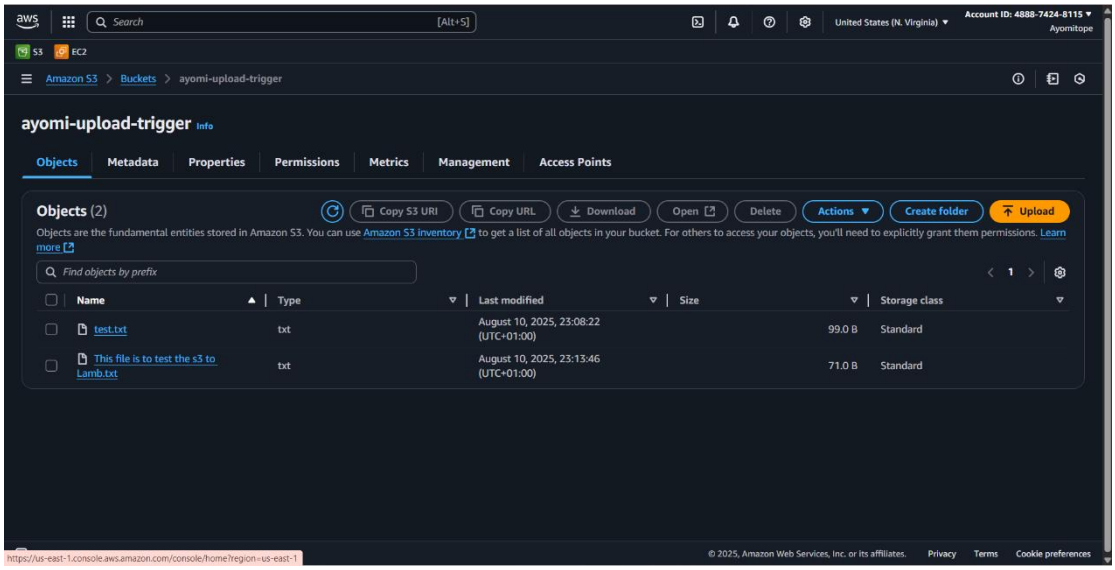


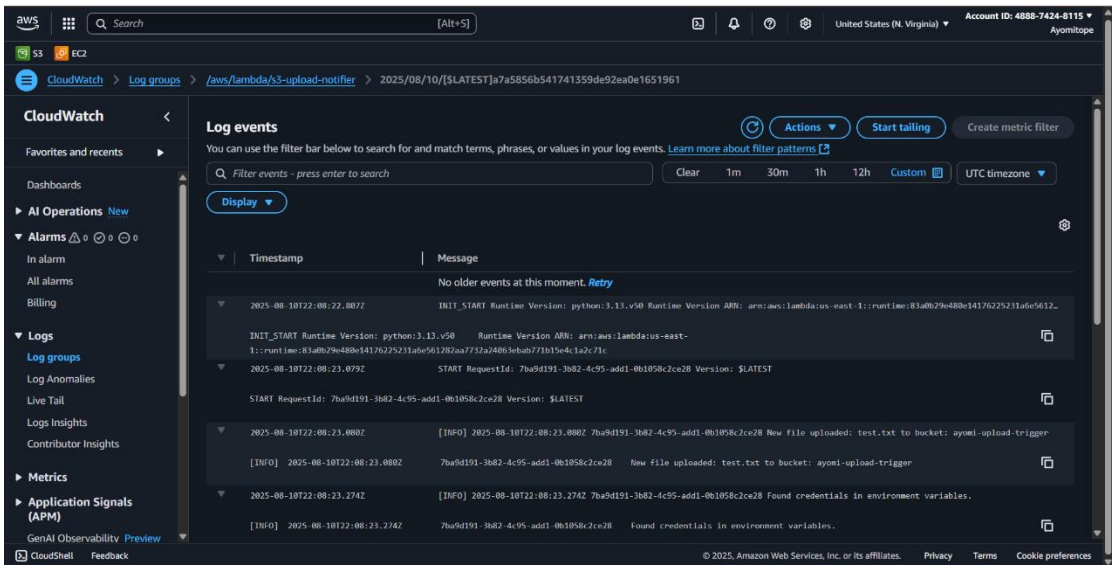
Step 7: Set Up S3 Trigger
In your Lambda function, go to the "Configuration" tab
Click "Add trigger"
Select "S3" as the trigger type
Choose your bucket from the dropdown
For "Event type", select "All object create events"
Optionally add a prefix/suffix filter if needed
Check "Recursive invocation" if you want subfolder events

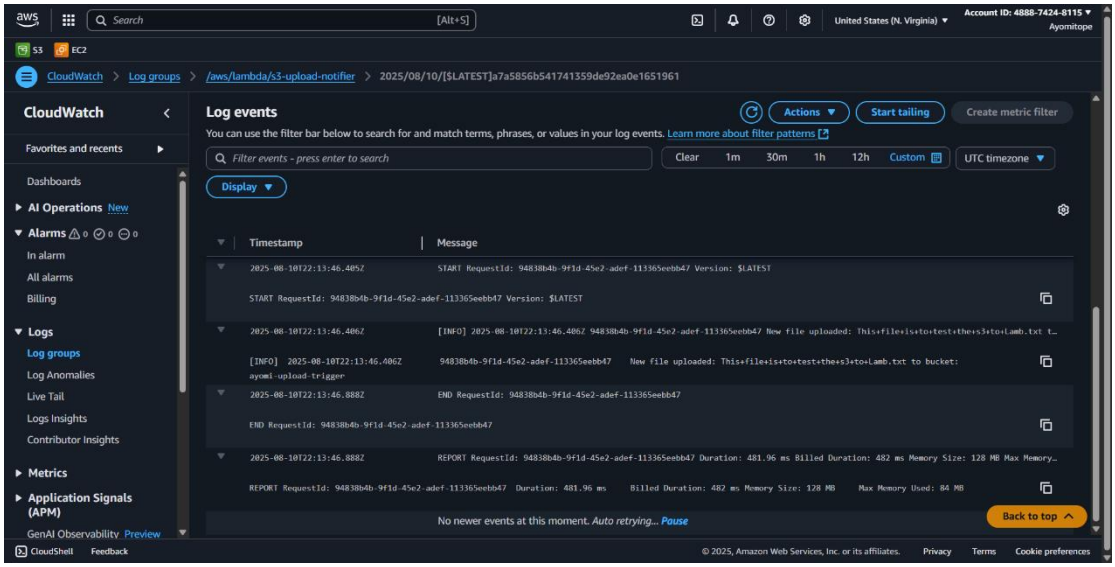
Click "Add"

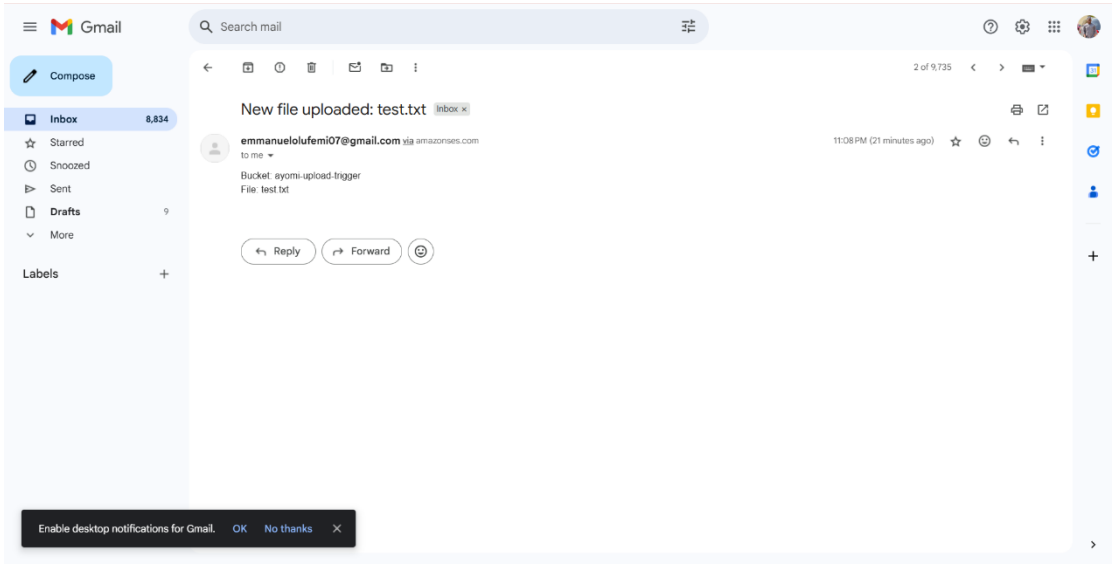


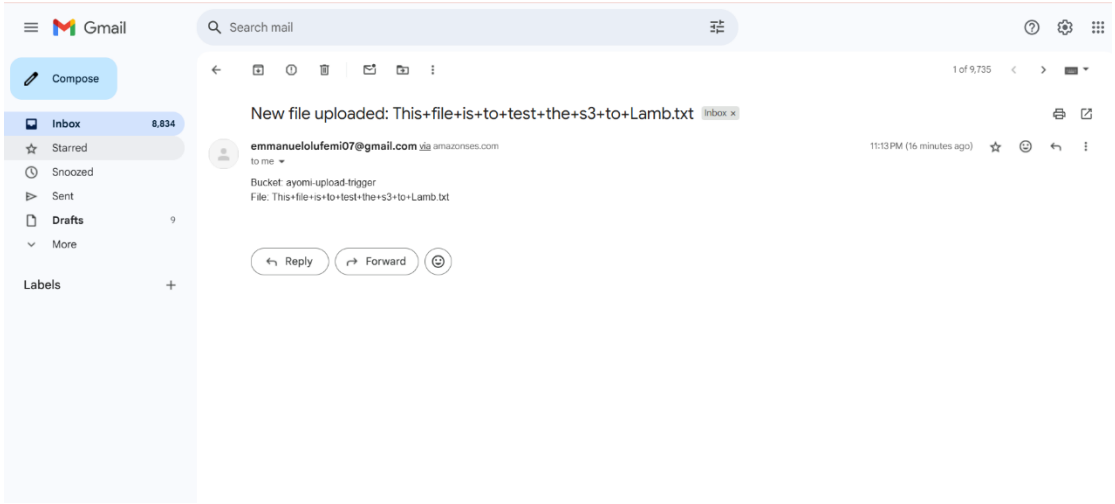
Step 8: Test the Setup
Upload files to your S3 bucket
I uploaded two files in my S3 buckets and checked the log from my CloudWatch and an email from AWS concerning the new files uploaded.






aws
Search
[Alt+S]
United States (N. Virginia)
Account ID: 4888-7424-8115
Ayomitope
S3
EC2
CloudWatch
Log groups
/aws/lambda/s3-upload-notifier
2025/08/10/[$LATEST]a7a5856b541741359de92ea0e1651961
CloudWatch
Favorites and recents
Dashboards
AI Operations New
Alarms
In alarm
All alarms
Billing
Logs
Log groups
Log Anomalies
Live Tail
Logs Insights
Contributor Insights
Metrics
Application Signals (APM)
GenAI Observability Preview
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
Actions
Start tailing
Create metric filter
Filter events - press enter to search
Clear
1m
30m
1h
12h
Custom
UTC timezone
Display
Timestamp
Message
2025-08-10T22:13:46.405Z
START RequestId: 94838b4b-9f1d-45e2-adef-113365eebb47 Version: $LATEST
2025-08-10T22:13:46.406Z
[INFO] 2025-08-10T22:13:46.406Z 94838b4b-9f1d-45e2-adef-113365eebb47 New file uploaded: This+file+is+to+test+the+s3+to+Lamb.txt to bucket: ayomi-upload-trigger
2025-08-10T22:13:46.888Z
END RequestId: 94838b4b-9f1d-45e2-adef-113365eebb47
2025-08-10T22:13:46.888Z
REPORT RequestId: 94838b4b-9f1d-45e2-adef-113365eebb47 Duration: 481.96 ms Billed Duration: 482 ms Memory Size: 128 MB Max Memory Used: 84 MB
No newer events at this moment. Auto retrying... Pause
Back to top
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Gmail
Search mail
Compose
Inbox 8,834
Starred
Snoozed
Sent
Drafts 9
More
Labels
2 of 9,735
New file uploaded: test.txt
Inbox
emmanuelolufemi07@gmail.com via amazonses.com
to me
11:08 PM (21 minutes ago)
Bucket: ayomi-upload-trigger
File: test.txt
Reply
Forward
Enable desktop notifications for Gmail.
OK
No thanks


Gmail
Search mail
Compose
Inbox 8,834
Starred
Snoozed
Sent
Drafts 9
More
Labels
1 of 9,735
New file uploaded: This+file+is+to+test+the+s3+to+Lamb.txt
Inbox
emmanuelolufemi07@gmail.com via amazonses.com
to me
11:13 PM (16 minutes ago)
Bucket: ayomi-upload-trigger
File: This+file+is+to+test+the+s3+to+Lamb.txt
Reply
Forward