

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

The screenshot shows the AWS Management Console Home page. At the top right, the account name "Ayomitope" is visible, with a dropdown menu showing "Account ID" and the value "488874248115". The main interface includes sections for "Recently visited" services like AWS Organizations, IAM Identity Center, IAM, GuardDuty, CloudWatch, EC2, S3, and Billing and Cost Management. Other sections include "Applications (0)", "AWS Health", and "Cost and usage". The "Cost and usage" section displays "Current month costs \$0.00" and "Cost breakdown No cost data available".

The screenshot shows the AWS Billing Bills page. The left sidebar is titled "Billing and Cost Management" and includes links for "Home", "Getting Started", "Billing and Payments", "Bills", "Payments", "Credits", "Purchase Orders", "Cost and Usage Analysis", "Cost Explorer", "Cost Explorer Saved Reports", "Cost Anomaly Detection", "Free Tier", "Data Exports", and "Customer Carbon Footprint Tool". The main content area is titled "Bills" and shows "AWS bill summary" with "Total charges and payment information". It displays the "Account ID" as "488874248115" and the "Billing period" as "June 1 - June 30, 2025". Below this, it shows "Estimated grand total: USD 0.00" and "Payment information". At the bottom, it shows "Highest estimated cost by service provider" with "No data to display".

The screenshot shows the AWS Billing and Cost Management console. On the left, there's a sidebar with navigation links like Home, Getting Started, Billing and Payments, Bills, Payments, Credits, Purchase Orders, Cost and Usage Analysis, Cost Explorer, Cost Explorer Saved Reports, Cost Anomaly Detection, Free Tier, Data Exports, and Customer Carbon Footprint Tool. The main area is titled 'Account Info' under 'Billing and Cost Management'. It shows 'Account details' with fields: Name (Ayomitope), Service provider (Amazon Web Services EMEA SARL), ID (488874248115), ARN (arn:aws:account:488874248115:account). Below this is 'Contact information' with fields: Full name (Ayomitope Olufemi), Company name (None), Phone number (+234 09039506046), Website URL (None), and Address (Soka, Building Ibadan, No states available 200263 NG). There's also an 'Alternate contacts' section with a note about adding contacts for Billing, Operations, and Security communications. At the bottom, there are tabs for 'Billing contact', 'Operations contact', and 'Security contact'. The footer includes links for CloudShell, Feedback, and various AWS services.

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

The screenshot shows the IAM Identity Center landing page. It features a dark header with the AWS logo and a search bar. Below the header, it says "Security, Identity, and Compliance". The main title is "IAM Identity Center" with the subtitle "Connect your existing identity provider and streamline workforce user access to AWS". A call-to-action button says "Enable IAM Identity Center". To the right, there's a box titled "Getting started" with links to "Get started with IAM Identity Center" and "IAM Identity Center prerequisites". At the bottom, there's a "How it works" section with a "Enable an instance of IAM Identity Center" button, and a "More resources" section with a "Documentation" link. The footer includes links for CloudShell, Feedback, and various AWS services.

The screenshot displays two consecutive pages from the AWS IAM Identity Center console, illustrating the management of users and groups.

Users Page:

- Left Sidebar:** Shows the navigation menu under "IAM Identity Center" with sections like "Managing instance", "Dashboard", "Users" (selected), "Groups", "Settings", "Multi-account permissions", "Application assignments", and "Related consoles".
- Top Bar:** Includes the AWS logo, search bar, and tabs for "Top", "Inv", "Int", "Am", "Uni", "fwd", "Par", "Dis", "ww", "AW", "Sim", "Cyl", "xai", "AM".
- Header:** "us-east-1.console.aws.amazon.com/singlesignon/home?region=us-east-1#instances/7223969b096d2bb4/users" and "United States (N. Virginia) Ayomitope".
- Content:** A table titled "Users (1)" showing one user entry:

Username	Display name	Status	MFA devices	Created by
emmanuel	Emmanuel CSN	Enabled	None	Manual

Groups Page:

- Left Sidebar:** Similar to the first page, with "Groups" selected in the navigation menu.
- Top Bar:** Same as the first page.
- Header:** "us-east-1.console.aws.amazon.com/singlesignon/home?region=us-east-1#instances/7223969b096d2bb4/groups" and "United States (N. Virginia) Ayomitope".
- Content:** A table titled "Groups (1)" showing one group entry:

Group name	Description	Created by
SOC Team	For Analysis purposes	Manual

Central management

- Use service control policies (SCPs) to prevent instances of IAM Identity Center from being created, or isolate the member accounts that are allowed to create account instances.
- IAM Identity Center allows member accounts of an organization to enable AWS applications that are independent of the organization instance with self-managed, account instances of IAM Identity Center.

Settings summary

- Instance name: AyomitopeCSN
- Identity source: Identity Center directory
- Region: US East (N. Virginia) | us-east-1
- Organization ID: o-vegDebo8tI
- AWS access portal URL: https://ayocsnawsapps.com/start
- Issuer URL: https://identitycenter.amazonaws.com/ssoinst-23969b096d2bb4

Monitor activities in your instances of IAM Identity Center

IAM Identity Center setup

What's new

Upcoming changes to AWS CloudTrail logs of IAM

Permission sets (1)

Permission set	Description	ARN	Provisioned
SecurityAudit	-	arn:aws:sso::permissionSet:ssoinst-7223969b096d2bb4/ps-eef1da8a4e...	Not pr...

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)

The screenshot shows the AWS EC2 Instances page. The left sidebar includes links for Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots, Lifecycle Manager). The main content area displays a table for 'Instances (1/1) Info'. The table has columns for Name (CSN-Bootcam...), Instance ID (i-07d99471dee75132b), Instance state (Running), Instance type (t2.micro), Status check (2/2 checks passed), Alarm status (View alarms +), Availability Zone (us-east-1a), and Public IPv4 IP (ec2-34-203-2-). Below the table, the instance details for 'i-07d99471dee75132b (CSN-Bootcamp-Week3)' are shown, including VPC ID (vpc-0ebb33f37ccbf0485), Subnet ID (subnet-0c1b9fa38b9d82a8c), Outpost ID (none), IP addresses (Public IPv4 address: 34.203.224.166, Private IPv4 address: 172.31.25.122, Secondary private IPv4 address: none, Carrier IP addresses (ephemeral: none)), and Hostname and DNS (Public DNS: none, Private IP DNS name (IPv4 only): none, IPv4-only IP based name: A record only). The bottom of the page includes CloudShell, Feedback, and copyright information.

This screenshot shows the same AWS EC2 Instances page as the previous one, but the instance details for 'i-07d99471dee75132b (CSN-Bootcamp-Week3)' are expanded to show its security group configuration. The 'Security groups' section lists 'sg-0a975646ff8e7a6b9 (launch-wizard-5)'. Below this, the 'Inbound rules' section is visible, showing a single rule: Name (sgr-09a40149d8ccb70df), Security group rule ID (sgr-09a40149d8ccb70df), Port range (3389), Protocol (TCP), and Source (41.203.95.6/32). The 'Outbound rules' section is also partially visible. The bottom of the page includes CloudShell, Feedback, and copyright information.

[us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instancesv3:case=tag:true%5C;client:false\\$regex=tagsfalse%5C;clientfalse](https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instancesv3:case=tag:true%5C;client:false$regex=tagsfalse%5C;clientfalse)

EC2 Instances

Instances (1/2) Info

Find Instance by attribute or tag (case-sensitive)

All states ▾

Instance state ▾ Instance type ▾ Status check ▾ Alarm status ▾ Availability Zone ▾ Public IPv4

i-024e244d87e9004cb (CSN-Bootcamp-Week3)

Networking

VPC ID: vpc-0ebb33f37ccbf0485 Subnet ID: subnet-0c1b9fa38b9d82a8c Availability zone: us-east-1a

IP addresses

Public IPv4 address: 204.236.203.206 Private IPv4 addresses: 172.31.24.63 Carrier IP addresses (ephemeral): -

Hostname and DNS

CloudShell Feedback

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstanceDetailsinstanceId=i-024e244d87e9004cb

EC2 Instances

Details

AMI ID: ami-0345f4fe05216fc4 Monitoring: disabled Platform details: Windows

AMI name: Windows_Server-2022-English-Full-Base-2025.06.11 Allowed image: - Termination protection: Disabled

Stop protection: Disabled Launch time: Tue Jul 01 2025 15:26:05 GMT+0100 (West Africa Standard Time) (5 minutes)

Instance reboot migration: Default (On) Instance auto-recovery: Default

State transition reason: - AMI Launch index: 0

State transition message: - Credit specification: standard

Owner: 488874248115 Usage operation: RunInstances:0002

Current instance boot mode: legacy-bios Enclaves Support: -

Answer RDN DNS hostname IPv4: Allow tags in instance metadata: Disabled

CloudShell Feedback

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ConnectToInstanceinstanceId=i-024e244d87e9004cb

EC2 Instances

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:

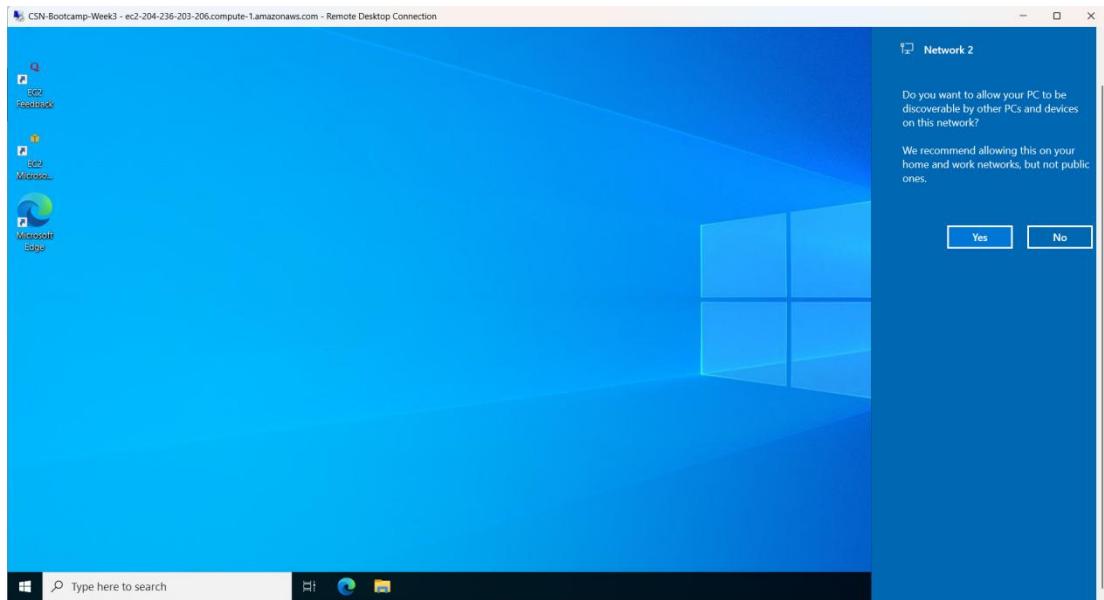
Public DNS: ec2-204-236-203-206.compute-1.amazonaws.com

Username info: Administrator

Password: OkDYRaiY9Yplo&MkqvOz|@JCuqbEDTo

If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

CloudShell Feedback



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

✓ Acceptor: 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

The screenshot shows the AWS VPC console interface. The top navigation bar includes the AWS logo, search bar, and region selection (United States (N. Virginia)). The main menu has options like S3, EC2, and VPC. The left sidebar is expanded to show the 'VPC dashboard' and 'Virtual private cloud' sections, with 'Your VPCs' selected. The main content area displays a table titled 'Your VPCs (1/3) Info' with three entries:

Name	VPC ID	State	Block Public Access	IPv4 CIDR	IPv6 CIDR
-	vpc-0ebb33f57ccbf0485	Available	Off	172.31.0.0/16	-
<input checked="" type="checkbox"/> VPC-A	vpc-0537435e2f61f40cd	Available	Off	10.10.0.0/16	-
<input type="checkbox"/> VPC-B	vpc-0a012a5bc41d14b4	Available	Off	10.20.0.0/16	-

Below the table, a specific VPC (vpc-0537435e2f61f40cd / VPC-A) is selected. The 'Subnets (2)' section shows two subnets: 'us-east-1a' with 'Public Subnet' and 'Private Subnet'. The 'Route tables (2)' section shows two route tables: 'rtb-0e7e3f0d72d8fbfb' and 'rtb-07af8a9b7b616e0b1'. The right side of the screen shows a 'Network' section with 'Connections'.

This screenshot shows the same VPC configuration as the first one, but the 'CIDRs' tab is selected for VPC-A. The 'IPv4 CIDRs' table shows a single entry: 'Address family: IPv4' with 'CIDR: 10.10.0.0/16' and 'Status: Associated'. The 'Edit CIDRs' button is visible at the top right of the table.

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

VPC > Your VPCs

Your VPCs (1/3) Info

Name	VPC ID	State	Block Public Access	IPv4 CIDR	IPv6 CIDR
-	vpc-0ebb33f57ccb0485	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 CIDs Flow logs Tags Integrations

IPv4 CIDs info Address family IPv4 CIDR Status

10.20.0.0/16 Associated

Edit CIDs

CloudShell Feedback

Last updated 3 minutes ago Actions Create VPC

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

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

VPC > Your VPCs

Your VPCs (1/3) Info

Name	VPC ID	State	Block Public Access	IPv4 CIDR	IPv6 CIDR
-	vpc-0ebb33f57ccb0485	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 CIDs Flow logs Tags Integrations

Resource map info

VPC Show details Your AWS virtual network

Subnets (2) Subnets within this VPC

- us-east-1a
 - Private Subnet rtb-03fd6c08abaa2f646
 - Public Subnet rtb-06447d3b51aa72979

Route tables (2) Route network traffic to resources

Network Connections

CloudShell Feedback

Last updated 4 minutes ago Actions Create VPC

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

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

VPC > Peering connections

Peering connections (1/2) Info

Name	Peer connection ID	Status	Requester VPC	Acceptor VPC
-	pcc-09de0f52b61e2a80	Deleted	vpc-0537435e2f61f40cd / VPC-A	vpc-0a012a5bcb41d14b4 / VPC-B
-	pcc-0ac4638b84eb51ef9	Active	vpc-0537435e2f61f40cd / VPC-A	vpc-0a012a5bcb41d14b4 / VPC-B

pcc-0ac4638b84eb51ef9

Details DNS Route tables Tags

Details

Requester owner ID 488874248115

Peer connection ID pcc-0ac4638b84eb51ef9

Status Active

Expiration time -

Requester VPC vpc-0537435e2f61f40cd / VPC-A

Requester CIDR 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/pcc-0ac4638b84eb51ef9

Acceptor VPC vpc-0a012a5bcb41d14b4 / VPC-B

Acceptor CIDR 10.20.0.0/16

Acceptor Region N. Virginia (us-east-1)

CloudShell Feedback

Actions Create peering connection

© 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:

VPC dashboard < VPC Peering connections

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

CloudShell Feedback

Peering connections (1/2) Info Actions Create peering connection

Name	Peering connection ID	Status	Requester VPC	Acceptor 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.

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

© 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:

VPC dashboard < VPC Route tables

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

CloudShell Feedback

Route tables (1/5) Info Actions Create route table

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
-	rtb-06a4a643808e63179	-	-	Yes	vpc-0ebb33f37ccbff0485
-	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)

Destination	Target	Status	Propagated
10.10.0.0/16	local	Active	No
10.20.0.0/16	pcx-0ac4638b84eb51ef9	Active	No

Both Edit routes

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

Route tables (1/5) Last updated 8 minutes ago

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
rtb-06a4a643808e63179	-	-	-	Yes	vpc-0ebb33f37ccb0485 4E
rtb-07af8a9b7b616e0h1	-	-	-	No	vpc-0537455e2f61140cd VPC-A 4E
rtb-06447d3b51aa72979	-	-	-	No	vpc-0a012a5bcb41d14b4 VPC-B 4E

rtb-06447d3b51aa72979

- Details
- Routes**
- Subnet associations
- Edge associations
- Route propagation
- Tags

Routes (2)

Destination	Target	Status	Propagated
10.10.0.0/16	pxx-0ac4638b84eb51ef9	Active	No
10.20.0.0/16	local	Active	No

WEEK 5

Created a Security Group

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

Security group (sg-016cf5f8ed653f279 | grafana-sg) was created successfully

sg-016cf5f8ed653f279 - grafana-sg

Details

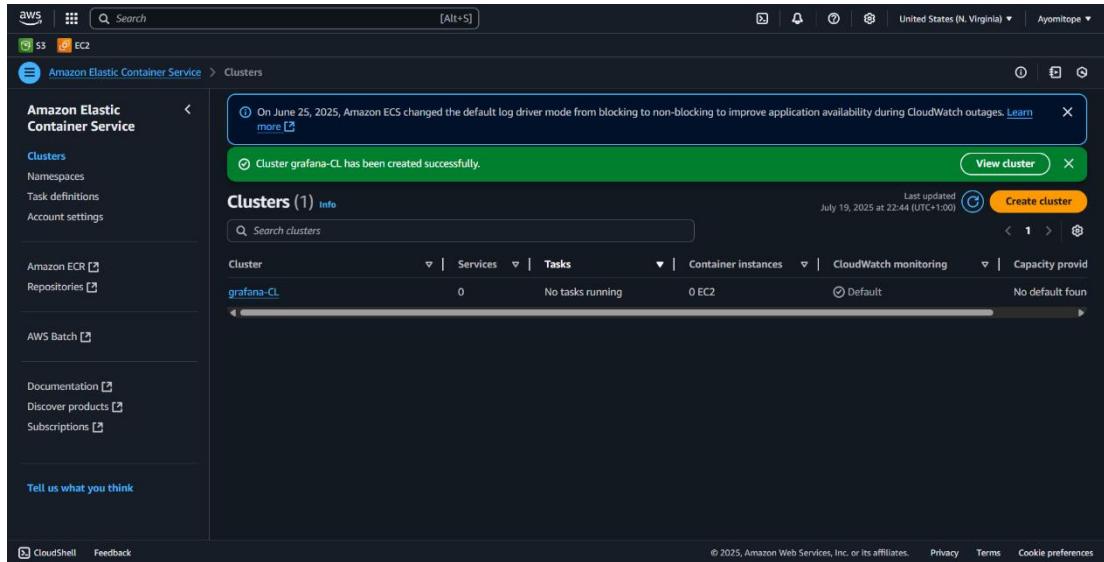
Security group name grafana-sg	Security group ID sg-016cf5f8ed653f279	Description Allow inbound HTTP on port 3000	VPC ID vpc-0ebb33f37ccb0485
Owner 488874248115	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-0916610a4e907fce1	IPv4	Custom TCP	TCP	3000

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**

The screenshot shows the AWS Elastic Container Service (ECS) Task Definitions page. The URL is <https://us-east-1.console.aws.amazon.com/ecs/v2/task-definitions/grafana-task/1/containers>. The page title is "Containers" under "Task definitions" for "grafana-task" revision 1.

Task size

- Task CPU:** 512 units (0.5 vCPU)
- Task memory:** 1,024 MiB (1 GB)
- Task CPU maximum allocation for containers:** 512 units (0.5 vCPU)
- Task memory maximum allocation for container memory reservation:** 1,024 MiB (1 GB)

Containers

Container name	Image	Private registry	Essential	CPU	Memory hard/soft	GPU
grafana	grafana/grafana:latest	-	Yes	0	-/-	-

The screenshot shows the AWS Elastic Container Service (ECS) Task Definitions page. The URL is <https://us-east-1.console.aws.amazon.com/ecs/v2/task-definitions/grafana-task/1>. A green banner at the top says "grafana-task:1 has been successfully created. You can use this task definition to deploy a service or run a task." Below it is the "grafana-task:1" task definition card.

Overview

ARN	Status	Time created	App environment
arn:aws:ecs:us-east-1:488874248115:task-definition/grafana-task:1	ACTIVE	July 19, 2025 at 23:25 (UTC+1:00)	Fargate

Task size

- Task CPU:** 512 units (0.5 vCPU)
- Task memory:** 1,024 MiB (1 GB)
- Task CPU maximum allocation for containers:** 512 units (0.5 vCPU)
- Task memory maximum allocation for container memory reservation:** 1,024 MiB (1 GB)

The screenshot shows the AWS Elastic Container Service (ECS) Task Definitions page. The URL is <https://us-east-1.console.aws.amazon.com/ecs/v2/task-definitions/grafana-task/1/json?region=us-east-1>. The page title is "JSON" under "Task definitions" for "grafana-task" revision 1.

```

1 {
2   "taskDefinitionArn": "arn:aws:ecs:us-east-1:488874248115:task-definition/grafana-task:1",
3   "containerDefinitions": [
4     {
5       "name": "grafana",
6       "image": "grafana/grafana:latest",
7       "cpu": 0,
8       "portMappings": [
9         {
10           "name": "grafana-3000-tcp",
11           "containerPort": 3000,
12           "hostPort": 3000,
13           "protocol": "tcp",
14           "appProtocol": "http"
15         }
16       ],
17       "essential": true,
18       "environment": [],
19       "environmentFiles": [],
20       "mountPoints": [],
21       "volumesFrom": [],
22       "ulimits": [],
23       "logConfiguration": {

```

Run the Service

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

The screenshot shows the AWS ECS console with the following details:

- Service overview:** Status is Active, Tasks (1 Desired) 0 Pending | 1 Running, Task definition: revision `grafana-task:1`, Deployment status Success.
- Status:** Service name is `grafana-service`, Service ARN is `arn:aws:ecs:us-east-1:488874248115:service/grafana-CL/grafana-service`, Health check grace period is 0 seconds, Deployments current state has 1 Completed task, Created at July 19, 2025 at 23:47 (UTC+1:00).
- Health:** No alarm recommendations.

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

Amazon Elastic Container Service > Clusters > grafana-CL > Tasks > 22cae7372cd14df29abd4da1e96c89de > Configuration

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-1:4888742481: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

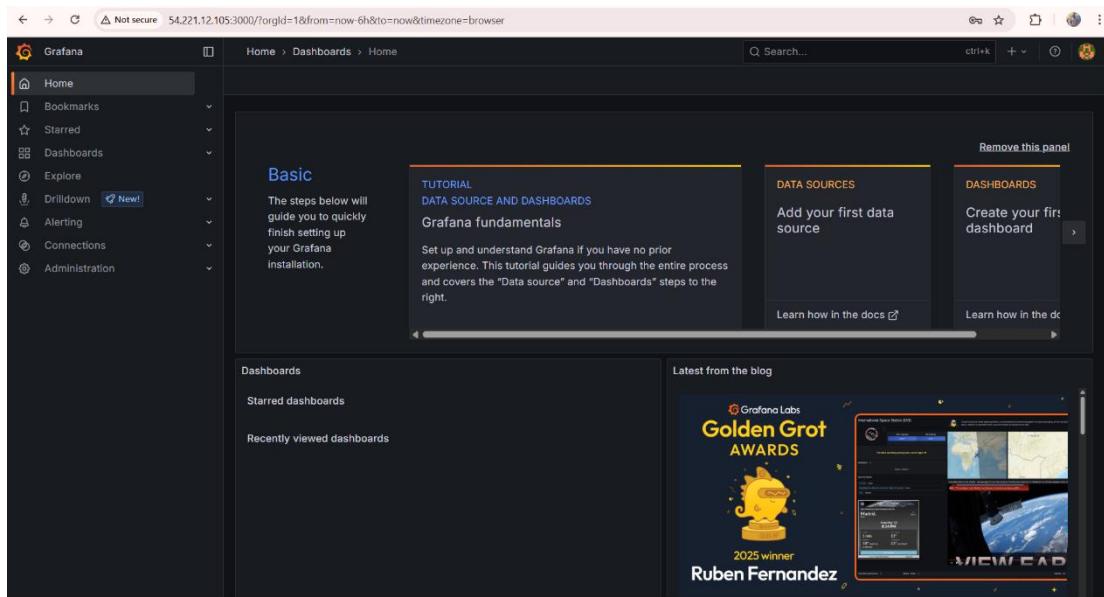
Host port	Container port	Protocol	External link
3000	3000	tcp	54.221.12.105:3000 open address

Tell us what you think

CloudShell Feedback

Not secure 54.221.12.105:3000/login

A dark-themed web page showing the Grafana logo (a yellow gear with a circle) in the center. The URL bar at the top shows "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:

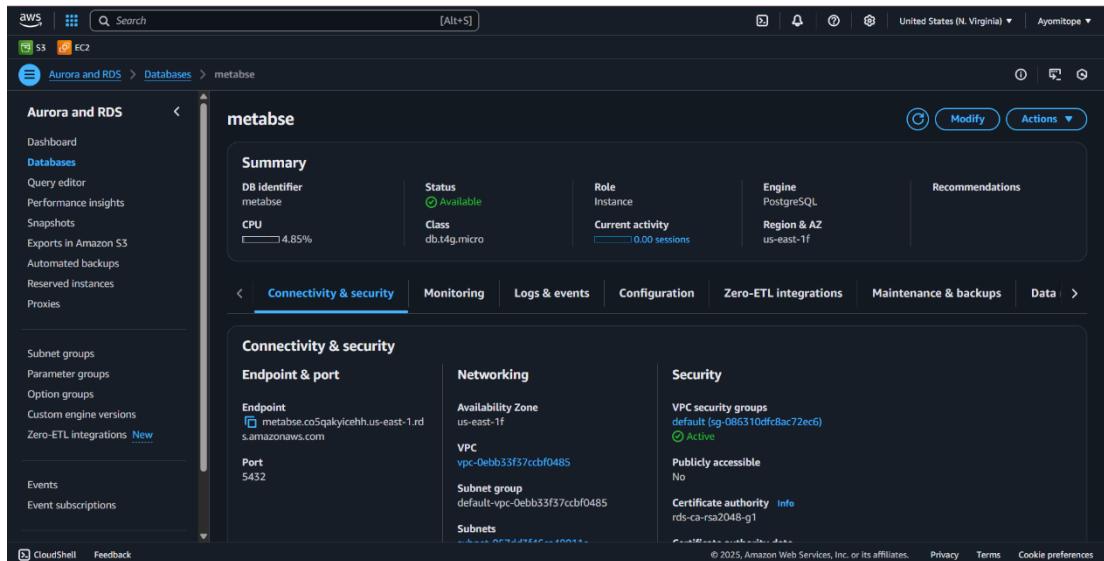
Engine: PostgreSQL

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

Cluster overview

ARN: arn:aws:ecs:us-east-1:488874248115:cluster/metabase-clus

Status: Active

CloudWatch monitoring: Default

Registered container instances:

Container instance	Task count
metabase-clus-1	1

Services:

Service name	Draining	Status	Pending	Running
database-tsk-service	-	Active	-	1

Tasks:

Task	Status
task-1	Running

Services tab selected.

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

sg-086310dfc8ac72ec6 - default

Details

Security group name	Security group ID	Description	VPC ID
default	sg-086310dfc8ac72ec6	default VPC security group	vpc-0ebb33f37cbff0485

Inbound rules (4)

Rule ID	IP version	Type	Protocol	Port range	Source	Description
537f2445f5190ec1	-	PostgreSQL	TCP	5432	sg-08cf44350257490...	-
13bb77a6e6a5591b	IPv4	HTTP	TCP	80	0.0.0.0/0	-
f7f51b14b815908	-	All traffic	All	All	sg-086310dfc8ac72ec6...	-
2c1e564f3efdf740	IPv4	HTTPS	TCP	443	0.0.0.0/0	-

Details

Security group name ecs-sg-tasks	Security group ID sg-0f8cf44350257490d	Description Allows communication with RDS	VPC ID vpc-0ebb35f37ccbf0485
Owner 488874248115	Inbound rules count 3 Permission entries	Outbound rules count 2 Permission entries	

Inbound rules (3)

IP version	Type	Protocol	Port range	Source	Description
IPv4	HTTP	TCP	80	0.0.0.0/0	-
-	Custom TCP	TCP	3000	sg-086310dfc8ac72ec6...	-
IPv4	HTTPS	TCP	443	0.0.0.0/0	-

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

Details

Target type IP	Protocol : Port HTTP: 3000	Protocol version HTTP1	VPC vpc-0ebb35f37ccbf0485
IP address type IPv4	Load balancer None associated		

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

Health check settings

Protocol HTTP	Path /	Port Traffic port	Healthy threshold 3 consecutive health check successes
Unhealthy threshold 5 consecutive health check failures	Timeout 10 seconds	Interval 30 seconds	Success codes 200

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
}
  
```

```

18 ],
19   "essential": true,
20   "environment": [
21     {
22       "name": "MB_DB_NAME",
23       "value": "metabase"
24     },
25     {
26       "name": "MB_DB_HOST",
27       "value": "metabase.cosqakyicehh.us-east-1.rds.amazonaws.com"
28     },
29     {
30       "name": "MB_DB_PASS",
31       "value": "security12!"
32     },
33     {
34       "name": "MB_DB_PORT",
35       "value": "5432"
36     },
37     {
38       "name": "MB_DB_TYPE",
39       "value": "postgres"
40     },
41     {
42       "name": "MB_DB_USER",
43       "value": "metabase"
44     }
45   ],
46   "environmentFiles": [],
47   "mountPoints": []

```

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

The screenshot shows the AWS ECS console with the path: Amazon Elastic Container Service > Clusters > `metabase-clus` > Services > `metabase-tsk-service` > Health. The service is named `metabase-tsk-service`. Status: Active. Tasks (1 Desired): 0 Pending | 1 Running. Task definition: revision `metabase-tk:1`. Deployment status: Success. Last updated: July 30, 2025 at 13:31 (UTC+1:00). The 'Health and metrics' tab is selected, showing the Load balancer health for `metabase-lb`. It lists Application Load Bal... and Target group `metabase-tg` with 1 Healthy and 0 Unhealthy targets.

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

The screenshot shows the AWS ELB console with the path: EC2 > Load balancers > `metabase-lb`. The Load balancer ARN is `arn:aws:elasticloadbalancing:us-east-1:488874248115:loadbalancer/app/metabase-lb/6848c6b2fc4e4b46`. The DNS name is `metabase-lb-143720316.us-east-1.elb.amazonaws.com`. The 'Listeners and rules' tab shows a single listener for port 80 targeting the `metabase-tg` target group. The target group stickiness is off. The 'Listeners and rules (1)' info panel shows a summary of the listener configuration.

The screenshot shows the Metabase dashboard with a sidebar containing collections like 'How to use Metabase', 'Our analytics', 'Your personal collection', 'Examples', and 'Other users' personal collections'. The main area displays various analytical insights such as 'A summary of Accounts', 'A glance at People', 'A look at Orders', 'A summary of Analytic Events', 'A glance at Products', 'Some Insights about Feedback', 'Some Insights about Reviews', 'A look at Invoices', and 'Metabase tips'. The URL in the address bar is `metabase-lb-143720316.us-east-1.elb.amazonaws.com/auto/dashboard/table/5`.

WEEK 7

Create a Task Definition

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

The screenshot shows the AWS ECS Task Definition creation interface. The task definition 'grafana-tk' has been successfully created. The task's ARN is `arn:aws:ecs:us-east-1:4888742481:task-definition/grafana-tsk:1`. The status is **ACTIVE**. The task was created on **July 31, 2025 at 13:39 (UTC+1:00)**. The app environment is **Fargate**. The task role is `ecsTaskExecutionRole`. The operating system/architecture is **Linux/X86_64**. The network mode is `awsvpc`. The task size shows **Task CPU** as 1,024 units (1 vCPU) and **Task memory** as 3,072 MB (3 GB).

Create an ECS Cluster

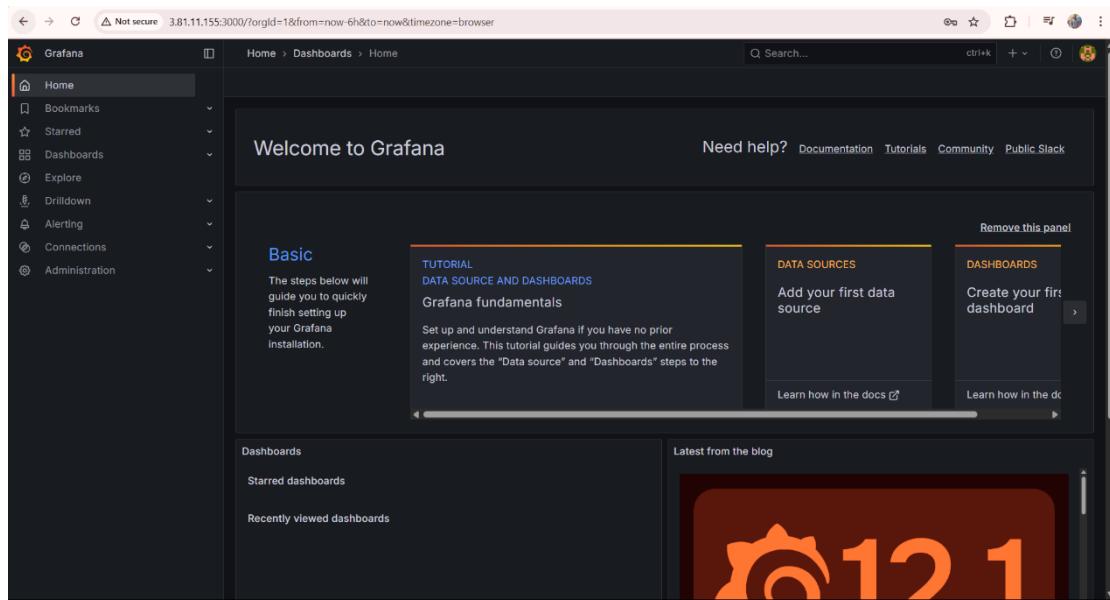
- Go to **ECS > Clusters > Create Cluster**.
- Choose “Networking only (Fargate)” type.
- Named the cluster.

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**

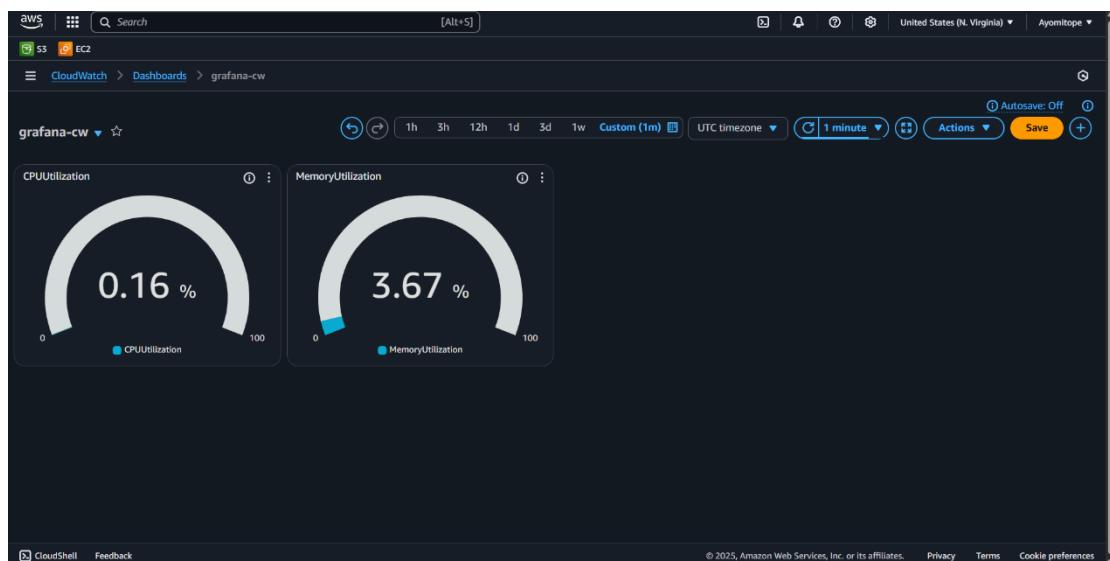
Access Grafana

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**



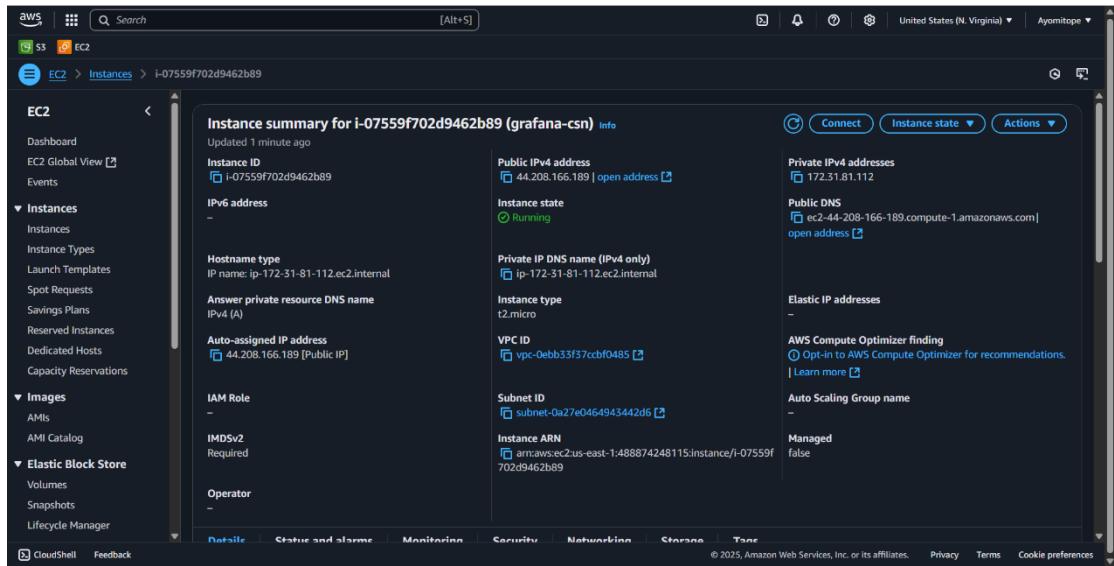
Create an EC2 Instnace

- 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
```



EC2 > Instances > i-07559f702d9462b89

Details

AMI ID	Monitoring	Platform details
ami-020cba7c5df1f615	disabled	Linux/UNIX
AMI name	Allowed image	Termination protection
ubuntu/images/hvm-ssd-gp3/ubuntu-noble-24.04-amd64-server-20250610	-	Disabled
Stop protection	Launch time	AMI location
Disabled	Thu Jul 31 2025 14:15:22 GMT+0100 (West Africa Standard Time) (about 1 hour)	amazon/ubuntu/images/hvm-ssd-gp3/ubuntu-noble-24.04-amd64-server-20250610
Instance reboot migration	Instance auto-recovery	Lifecycle
Default (On)	Default	normal
Stop-hibernate behavior	AMI Launch index	Key pair assigned at launch
Disabled	0	Ayo
State transition reason	Credit specification	Kernel ID
-	standard	-
State transition message	Usage operation	RAM disk ID
-	RunInstances	-
Owner	Enclaves Support	Boot mode
488874248115	-	uefi-preferred
Current instance boot mode	Allow tags in instance metadata	Use RBN as guest OS hostname
legacy-bios	Disabled	Disabled

```

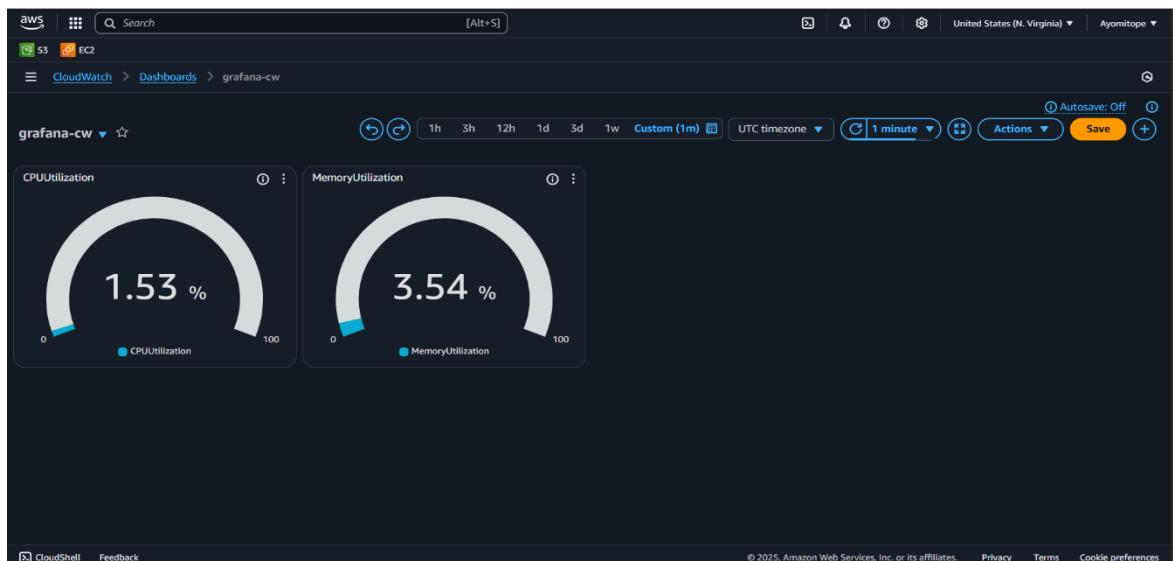
Reading package lists... Done
0 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-81-112:~$ curl http://3.81.11.155:3000/
<a href="/login">Found</a>
ubuntu@ip-172-31-81-112:~$ sudo apt install apache2-utils -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2-utils is already the newest version (2.4.58-ubuntu8.7).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-81-112:~$ ab -n 10000 -c 1000 http://3.81.11.155:3000/
This is ApacheBench, Version 2.3 <Revision: 1903610 S>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/
Benchmarking 3.81.11.155 (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests

```

i-07559f702d9462b89 (grafana-csn)

PublicIPs: 44.208.166.189 PrivateIPs: 172.31.81.112

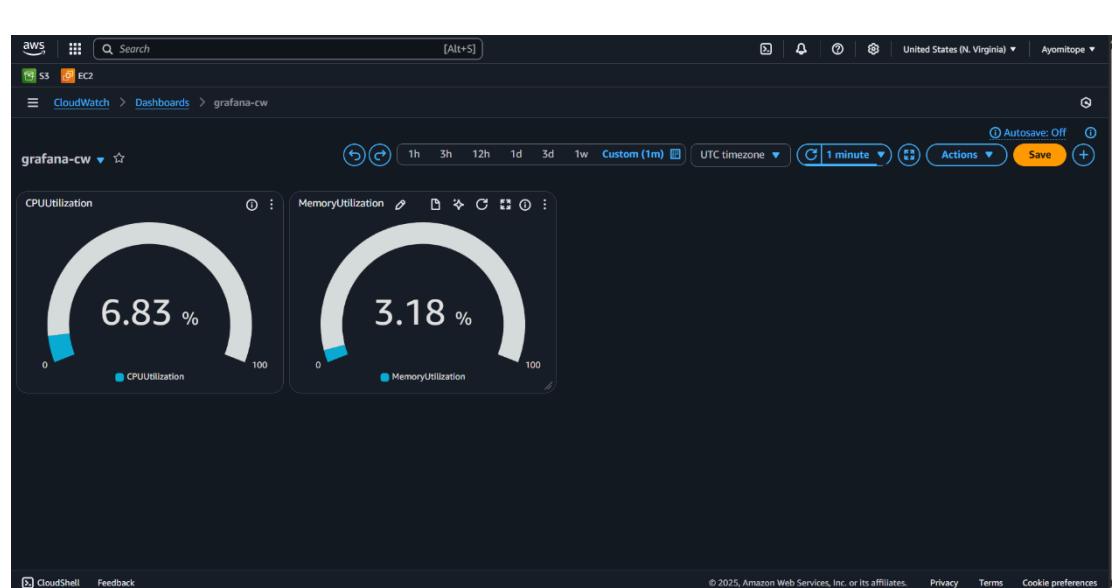
The change in the CPUUtilization and MemoryUtilization

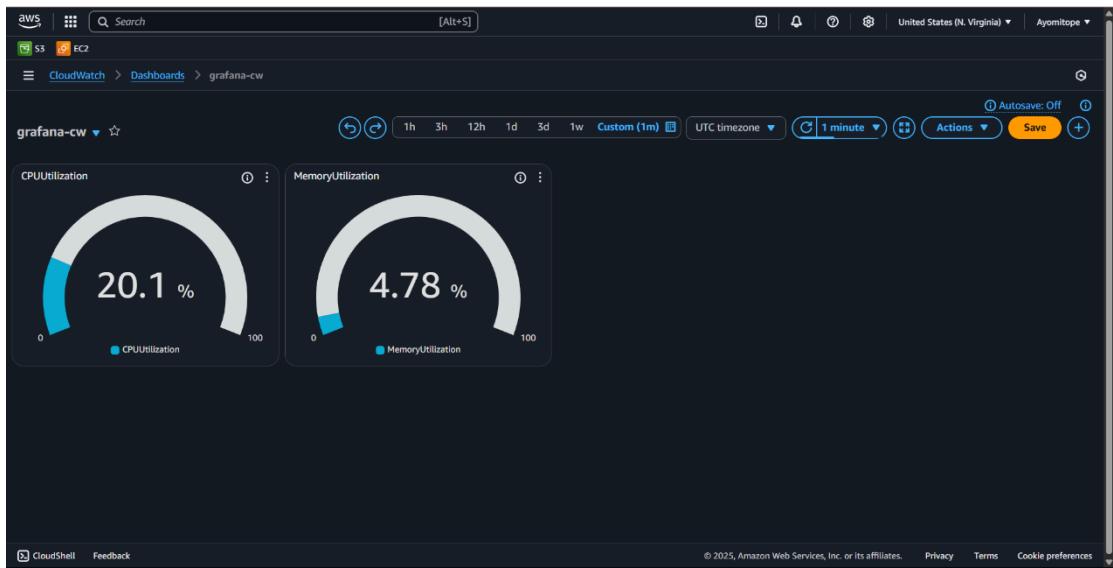


The changes in the CPUUtilization and MemoryUtilization

```
ubuntu@ip-172-31-81-112:~$ wrk -t12 -c1000 -d30s http://3.81.11.155:3000
Running 30s test @ http://3.81.11.155:3000
  12 threads and 1000 connections
    Thread Stats Avg Stdev Max +/- Stddev
      Latency 125.89ms 69.17ms 447.89ms 64.28%
      Req/Sec 666.16 243.23 1.79k 69.46%
      23849 requests in 30.09s, 63.68MB read
      Requests/sec: 7924.48
      Transfer/sec: 2.12MB
ubuntu@ip-172-31-81-112:~$ wrk -t12 -c1000 -d60s http://3.81.11.155:3000
Running 60s test @ http://3.81.11.155:3000
  12 threads and 1000 connections
    Thread Stats Avg Stdev Max +/- Stddev
      Latency 126.84ms 72.03ms 530.23ms 64.84%
      Req/Sec 666.16 251.37 1.55k 66.90%
      476448 requests in 1.00m, 127.23MB read
      Requests/sec: 7928.63
      Transfer/sec: 2.12MB
ubuntu@ip-172-31-81-112:~$
```

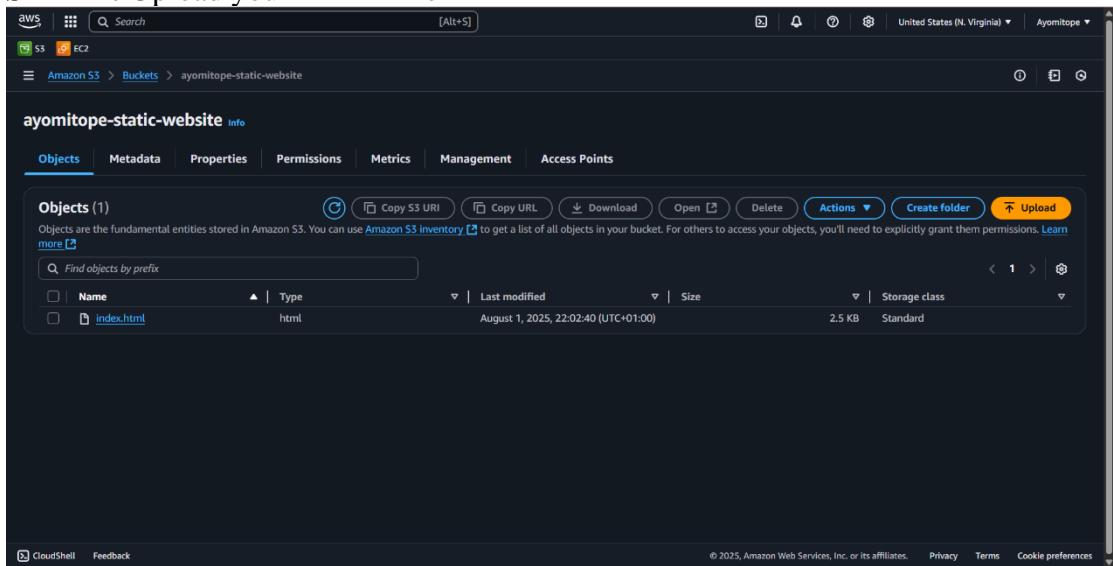
i-07559f702d9462b89 (grafana-csn)
PublicIPs: 44.208.166.189 PrivateIPs: 172.31.81.112





WEEK 8

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



- STEP 3:** Enable Static website

Successfully edited static website hosting.

Requester pays

When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. [Learn more](#)

Requester pays

Disabled

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

We recommend using AWS Amplify Hosting for static website hosting

Deploy a fast, secure, and reliable website quickly with AWS Amplify Hosting. Learn more about Amplify Hosting or View your existing Amplify apps

Create Amplify app

S3 static website hosting

Enabled

Hosting type

Bucket hosting

Bucket website endpoint

When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. [Learn more](#)

<http://ayomitope-static-website.s3-website-us-east-1.amazonaws.com>

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

STEP 4: Uncheck permission and edit the bucket policy

Successfully edited bucket policy.

Off

Individual Block Public Access settings for this bucket

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. [Learn more](#)

```
{ "Version": "2012-10-17", "Statement": [ { "Sid": "PublicReadGetObject", "Effect": "Allow", "Principal": "*", "Action": "s3:GetObject", "Resource": "arn:aws:s3:::ayomitope-static-website/*" } ] }
```

Copy

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

About Me



Hello! My name is **Ayomitope Olufemi**, and this is my very first website. I'm learning something new every day as I build this.

My background was in History and International Studies, which I studied at the Federal University Oye Ekiti, Ekiti State. Transitioning from a non-tech background into the tech world has been both challenging and rewarding. This journey has fueled my determination to succeed and has shown me that with consistency, curiosity, and the right resources, anyone can break into tech and thrive.

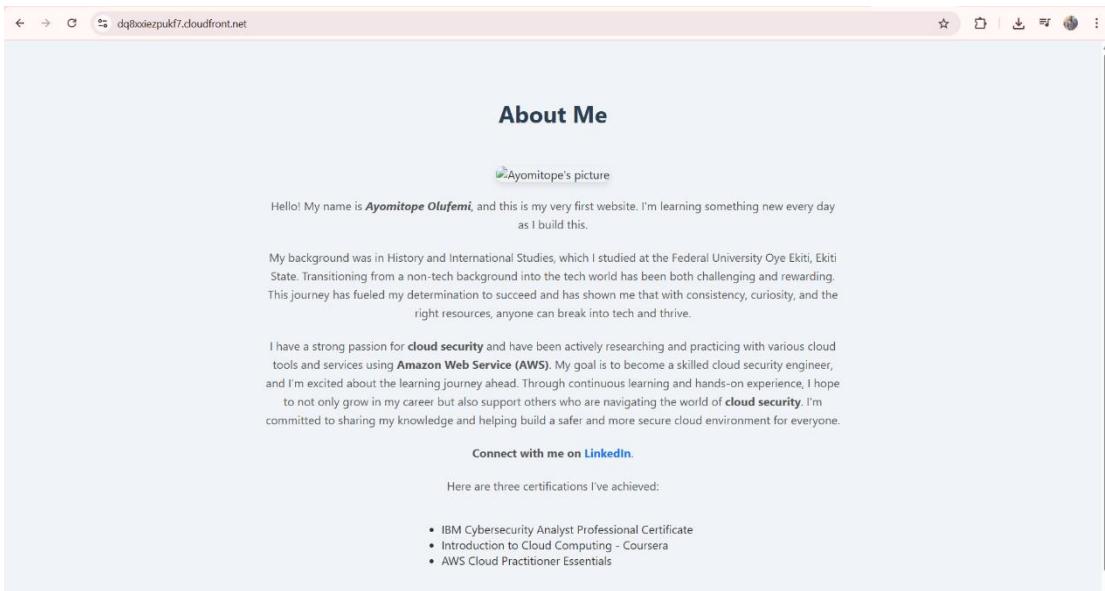
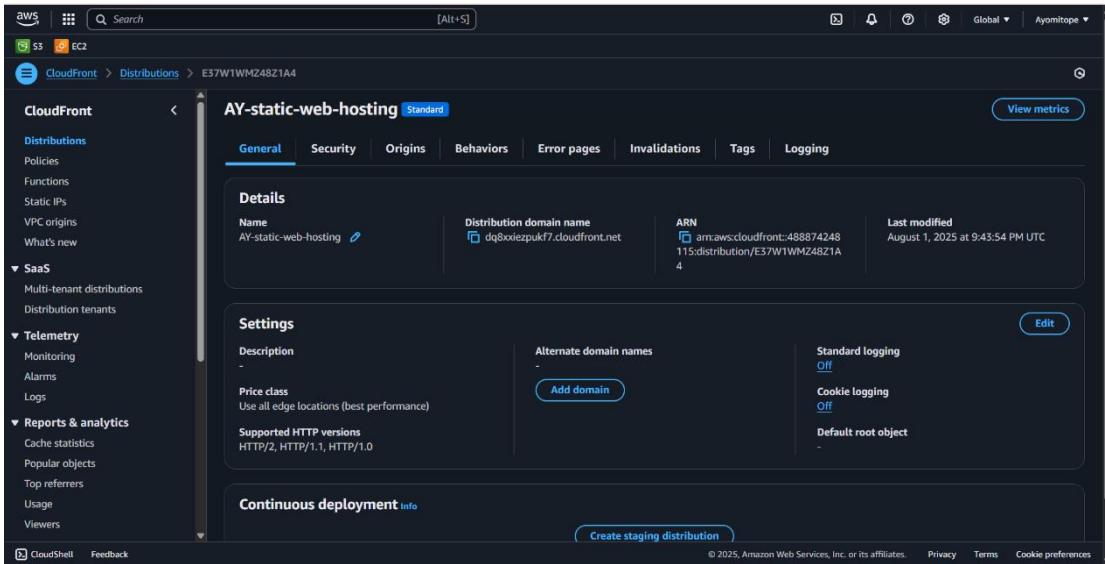
I have a strong passion for **cloud security** and have been actively researching and practicing with various cloud tools and services using **Amazon Web Service (AWS)**. My goal is to become a skilled cloud security engineer, and I'm excited about the learning journey ahead. Through continuous learning and hands-on experience, I hope to not only grow in my career but also support others who are navigating the world of **cloud security**. I'm committed to sharing my knowledge and helping build a safer and more secure cloud environment for everyone.

Connect with me on [LinkedIn](#).

Here are three certifications I've achieved:

- IBM Cybersecurity Analyst Professional Certificate
- Introduction to Cloud Computing - Coursera
- AWS Cloud Practitioner Essentials

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).

The screenshot shows the deSEC e.V. domain management interface. At the top, there are tabs for 'DOMAIN MANAGEMENT' and 'TOKEN MANAGEMENT'. The 'DOMAIN MANAGEMENT' tab is active, showing a table of DNS records for the domain 'ayoi.dedyn.io'. The table has columns for Type, Subname, Content, TTL (seconds), Last touched, and Actions. There are three entries:

- NS**: Subname '(optional)', Content 'ns1.desec.io.', 'ns2.desec.org.', TTL 3600, Last touched '6 days ago', Actions (edit, delete).
- CNAME**: Subname '_72f3f8946ba2eefaa4caac55c074', Content 'target hostname _22e876f04548d7b5c20f16983da05ad3.xlgrmvvlj.acm-validations.aws.', TTL 3600, Last touched 'less than a minute ago', Actions (edit, delete).
- CNAME**: Subname 'www', Content 'target hostname d2e4un2e48qp4j.cloudfront.net.', TTL 3600, Last touched 'less than a minute ago', Actions (edit, delete).

At the bottom of the table, there are buttons for 'Rows per page' (30) and '1-3 of 3'.

Below the table, there is a footer with links: 'Service Status', 'Source Code', 'Terms of Use', 'Privacy Policy (Datenschutzerklärung)', and 'Legal Notice (Impressum)'.

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.

The screenshot shows the AWS S3 console. The left sidebar shows navigation options like 'General purpose buckets', 'Directory buckets', 'Table buckets', 'Vector buckets', 'Access Grants', 'Access Points (General Purpose Buckets, FSx file systems)', 'Access Points (Directory Buckets)', 'Object Lambda Access Points', 'Multi-Region Access Points', 'Batch Operations', 'IAM Access Analyzer for S3', 'Block Public Access settings for this account', and 'Storage Lens' (with sub-options 'Dashboards', 'Storage Lens groups', 'AWS Organizations settings').

The main area displays 'General purpose buckets' with one entry: 'ayoi.dedyn.io'. The bucket details show it was created on 'August 14, 2025, 18:20:42 (UTC+01:00)' in 'US East (N. Virginia) us-east-1'. There are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. A 'Find buckets by name' search bar is also present.

On the right, there are two cards: 'Account snapshot' (info, updated daily, view dashboard) and 'External access summary - new' (info, updated daily). The URL at the bottom is <https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1>.

The screenshot shows the AWS S3 console interface. The left sidebar lists 'General purpose buckets' such as Directory buckets, Table buckets, Vector buckets, Access Grants, Access Points (General Purpose Buckets, FSx file systems), Access Points (Directory Buckets), Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and IAM Access Analyzer for S3. Below this is a section for 'Block Public Access settings for this account'. Under 'Storage Lens', there are options for Dashboards, Storage Lens groups, and AWS Organizations settings. The main content area shows the 'ayoi.dedyn.io' bucket. The 'Objects' tab is active, showing one object named 'index.html' which is an HTML file last modified on August 14, 2025, at 18:21:24 (UTC+01:00). There are buttons for Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload.

The screenshot shows the 'Static website hosting' configuration for the 'ayoi.dedyn.io' bucket. The sidebar remains the same. The main content area shows the 'Requester pays' section, which is currently disabled. Below it is the 'Static website hosting' section, which is enabled. The 'Hosting type' is set to 'Bucket hosting'. A note in this section recommends using AWS Amplify Hosting for static website hosting, mentioning that it allows for fast, secure, and reliable website deployment. There is a button to 'Create Amplify app'.

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 `ayoi.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"

The screenshot shows the AWS S3 console with the 'Buckets' page. A single bucket, 'ayomi-upload-trigger', is listed under 'General purpose buckets'. The bucket was created on August 10, 2025, at 18:56:05 UTC+01:00. On the right side of the page, there are two cards: 'Account snapshot' and 'External access summary - new'. The 'Account snapshot' card provides visibility into storage usage and activity trends, while the 'External access summary' card helps identify bucket permissions that allow public access or access from other AWS accounts.

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

The screenshot shows the AWS IAM console with the 'Roles' page. A role named 'Lambda-trigger-role-csn' is selected. The 'Summary' section shows the creation date as August 10, 2025, at 22:04 UTC+01:00. The ARN is arn:aws:iam::488874248115:role/Lambda-trigger-role-csn. The 'Permissions' tab is selected, showing three managed policies attached: AmazonS3ReadOnlyAccess, AmazonSESFullAccess, and AWSLambdaBasicExecutionRole. The 'Trust relationships', 'Tags', 'Last Accessed', and 'Revoke sessions' tabs are also visible.

Step 3: Verify SES Email Addresses

Go to the Amazon SES service

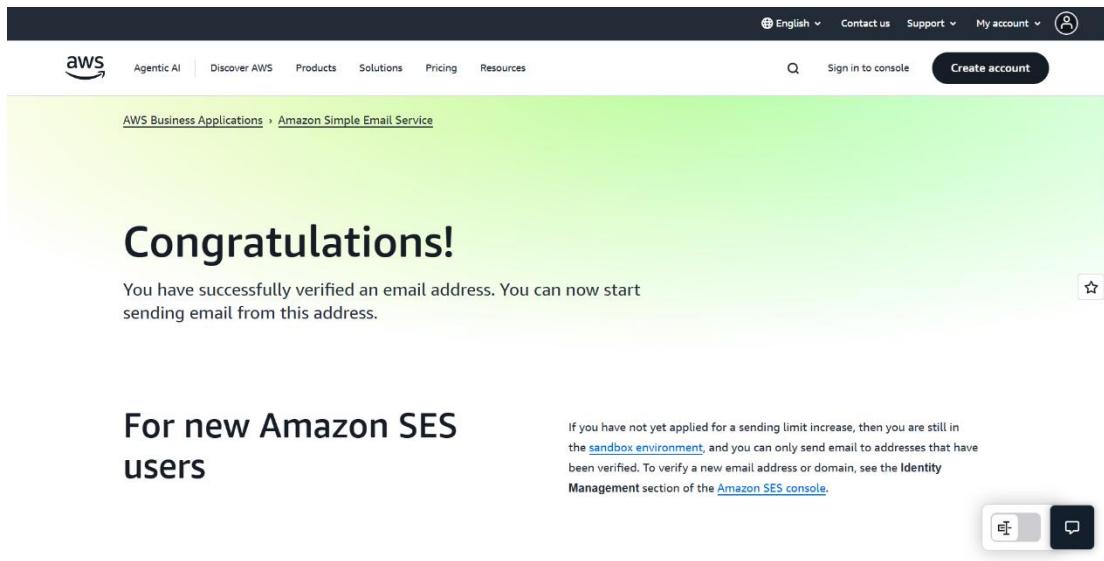
Click "Verified identities" in the left menu

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



This screenshot shows the "Identities" section of the AWS SES console. It lists one identity: "emmanuelolufemi07@gmail.com" which is marked as "Verified". There is also a "Recommendations" section showing no results. The left sidebar includes options like "Configuration sets", "Tenants", "Email templates", and "Email receiving". The top navigation bar shows "Amazon SES > Configuration: Identities".

Step 4: Create the Lambda Function

Go to the AWS Lambda service

Click "Create function"

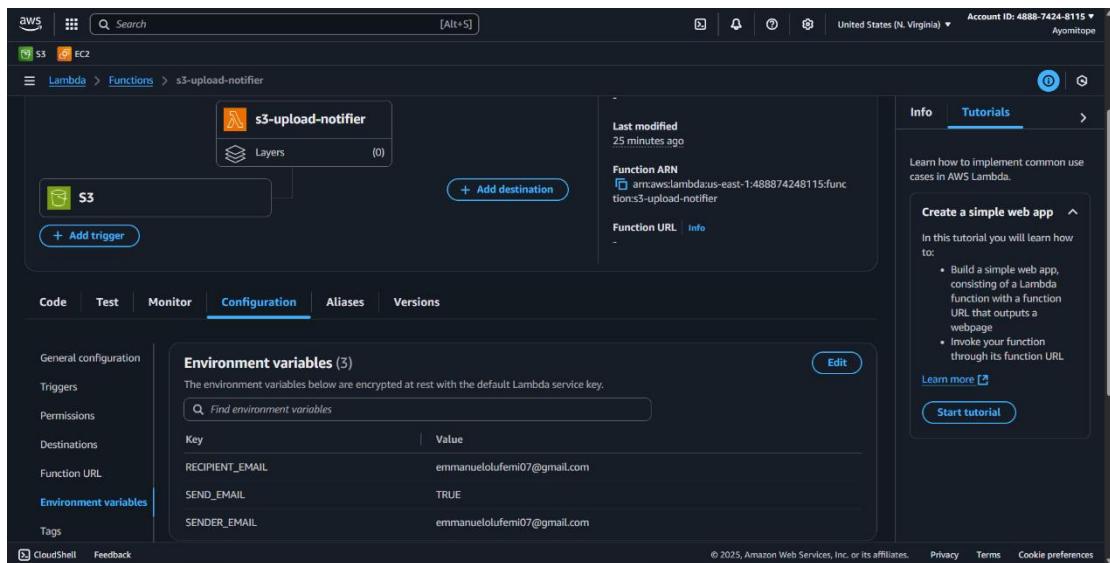
Choose "Author from scratch"

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}}
    }
}
```

```

    7 def lambda_handler(event, context):
8     for record in event['Records']:
9         bucket = record['s3']['bucket']['name']
10        key = record['s3']['object']['key']
11
12        logger.info(f"New file uploaded: {key} to bucket: {bucket}")
13
14        if os.environ.get('SEND_EMAIL', 'false').lower() == 'true':
15            send_email(bucket, key)
16
17        return {
18            'statusCode': 200,
19            'body': f"Processed upload of {key}"
20        }
21
22    def send_email(bucket, key):
23        ses = boto3.client('ses')
24        sender = os.environ['SENDER_EMAIL']
25        recipient = os.environ['RECIPIENT_EMAIL']
26
27        subject = f"New file uploaded: {key}"
28        body = f"Bucket: {bucket}\nfile: {key}"
29
30        ses.send_email(
31            Source=sender,
32            Destination={'ToAddresses': [recipient]},
33            Message={
34                'Subject': {'Data': subject},
35

```

The screenshot shows the AWS Lambda function editor with the code for the `s3-upload-notifier`. The code is a Python script named `lambda_function.py` that handles S3 events and sends an email notification if the environment variable `SEND_EMAIL` is set to true. The Lambda function is deployed to the `s3-upload-notifier` layer.

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

Key	Value
RECIPIENT_EMAIL	emmanuelolufemi07@gmail.com
SEND_EMAIL	TRUE
SENDER_EMAIL	emmanuelolufemi07@gmail.com

The screenshot shows the AWS Lambda function configuration page for the `s3-upload-notifier` function. The "Configuration" tab is selected. In the "Environment variables" section, there are three variables defined:

- `RECIPIENT_EMAIL`: emmanuelolufemi07@gmail.com
- `SEND_EMAIL`: TRUE
- `SENDER_EMAIL`: emmanuelolufemi07@gmail.com

Step 7: Set Up S3 Trigger

In your Lambda function, go to the "Configuration" tab

Click "Add trigger"

Select "S3" as the trigger type

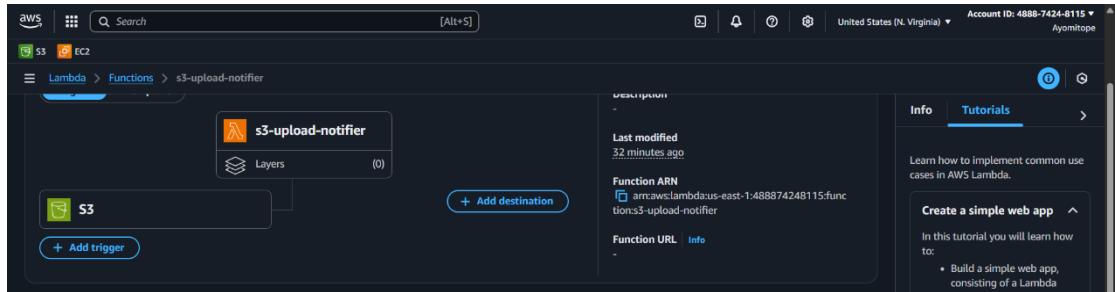
Choose your bucket from the dropdown

For "Event type", select "All object create events"

Optionally add a prefix/suffix filter if needed

Check "Recursive invocation" if you want subfolder events

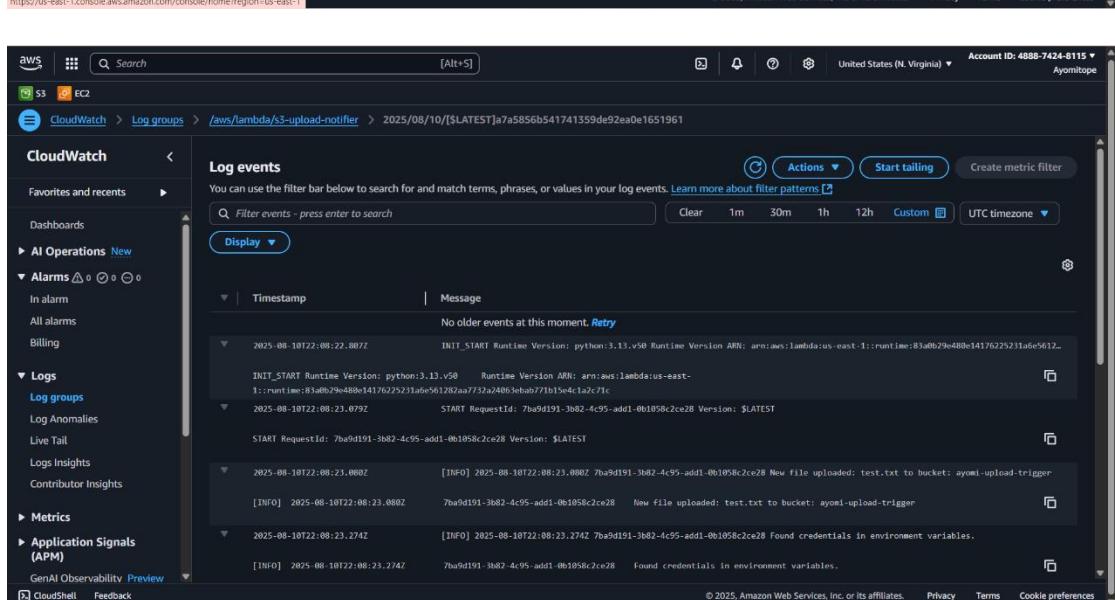
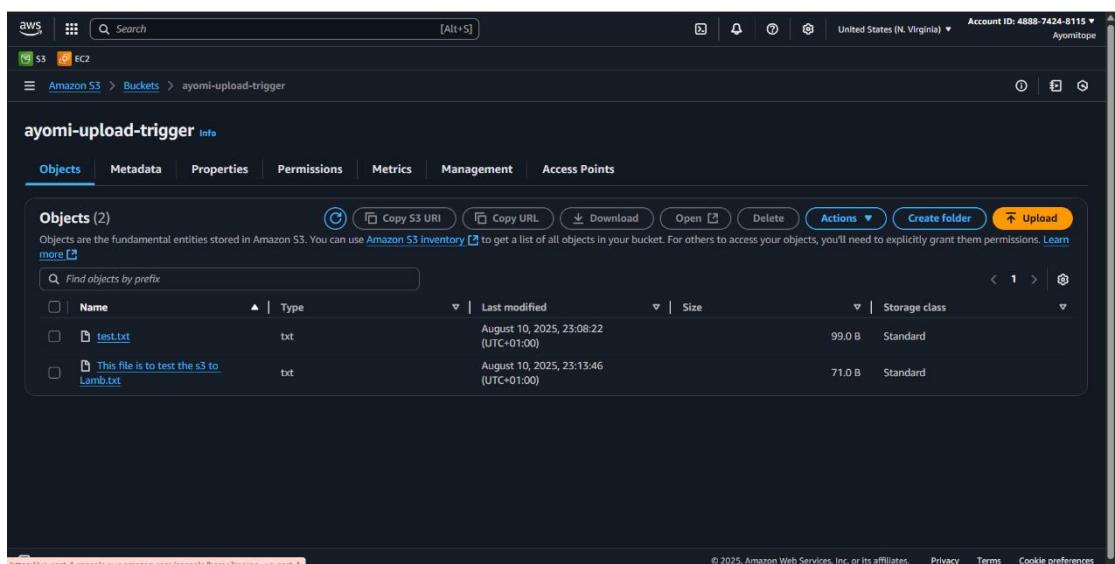
Click "Add"



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.



S3 EC2

CloudWatch > Log groups > /aws/lambda/s3-upload-notifier > 2025/08/10/[LATEST]a7a5856b541741359de92ea0e1651961

CloudWatch

Favorites and recents

CloudWatch Metrics

AI Operations New

Alarms 0

Logs

Log groups

Log Anomalies

Live Tail

Logs Insights

Contributor Insights

Metrics

Application Signals (APM)

GenAI Observability Preview

CloudShell Feedback

Search

[Alt+S]

Actions Start tailing Create metric filter

Filter events - press enter to search

Clear 1m 30m 1h 12h Custom UTC timezone

Timestamp Message

2025-08-10T22:13:46.405Z START RequestId: 94838b4b-9f1d-45e2-adef-113365eabb47 Version: \$LATEST

START RequestId: 94838b4b-9f1d-45e2-adef-113365eabb47 Version: \$LATEST

2025-08-10T22:13:46.406Z [INFO] 2025-08-10T22:13:46.406Z 94838b4b-9f1d-45e2-adef-113365eabb47 New file uploaded: This+file+is+to+test+the+s3+to+Lamb.txt t...

ayomi-upload-trigger

2025-08-10T22:13:46.406Z [INFO] 2025-08-10T22:13:46.406Z 94838b4b-9f1d-45e2-adef-113365eabb47 New file uploaded: This+file+is+to+test+the+s3+to+Lamb.txt to bucket:

2025-08-10T22:13:46.406Z END RequestId: 94838b4b-9f1d-45e2-adef-113365eabb47

END RequestId: 94838b4b-9f1d-45e2-adef-113365eabb47

2025-08-10T22:13:46.406Z REPORT RequestId: 94838b4b-9f1d-45e2-adef-113365eabb47 Duration: 481.96 ms Billed Duration: 482 ms Memory Size: 128 MB Max Memory Used: 84 MB

REPORT RequestId: 94838b4b-9f1d-45e2-adef-113365eabb47 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

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

Gmail

Compose

Inbox 8,834

Starred

Snoozed

Sent

Drafts 9

More

Labels +

Search mail

New file uploaded: test.txt

emmanuelolufemi07@gmail.com via amazonse... to me

Bucket: ayomi-upload-trigger

File: test.txt

11:08 PM (21 minutes ago)

Reply Forward

Enable desktop notifications for Gmail. OK No thanks

Gmail

Compose

Inbox 8,834

Starred

Snoozed

Sent

Drafts 9

More

Labels +

Search mail

New file uploaded: This+file+is+to+test+the+s3+to+Lamb.txt

emmanuelolufemi07@gmail.com via amazonse... to me

Bucket: ayomi-upload-trigger

File: This+file+is+to+test+the+s3+to+Lamb.txt

11:13 PM (16 minutes ago)

Reply Forward