

Weekly Tasks		
	Description	Deliverable
Week 7	<p>This week, your task is to deploy a simple containerized application (e.g., Nginx or grafana/grafana) on Amazon ECS with Fargate, using appropriate CPU and memory settings.</p> <p>Once deployed in a public subnet, create a CloudWatch dashboard with widgets for CPUUtilization and MemoryUtilization to monitor the ECS task in real time. For deeper insight, simulate load by refreshing or stress testing the app and observe how the metrics change.</p>	<p>Submit screenshots showing your ECS cluster and running service, the task definition with CPU and memory settings, and your CloudWatch dashboard with CPU and memory widgets.</p> <p>Optionally, include a screenshot showing changes in metrics under simulated load.</p>

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**:

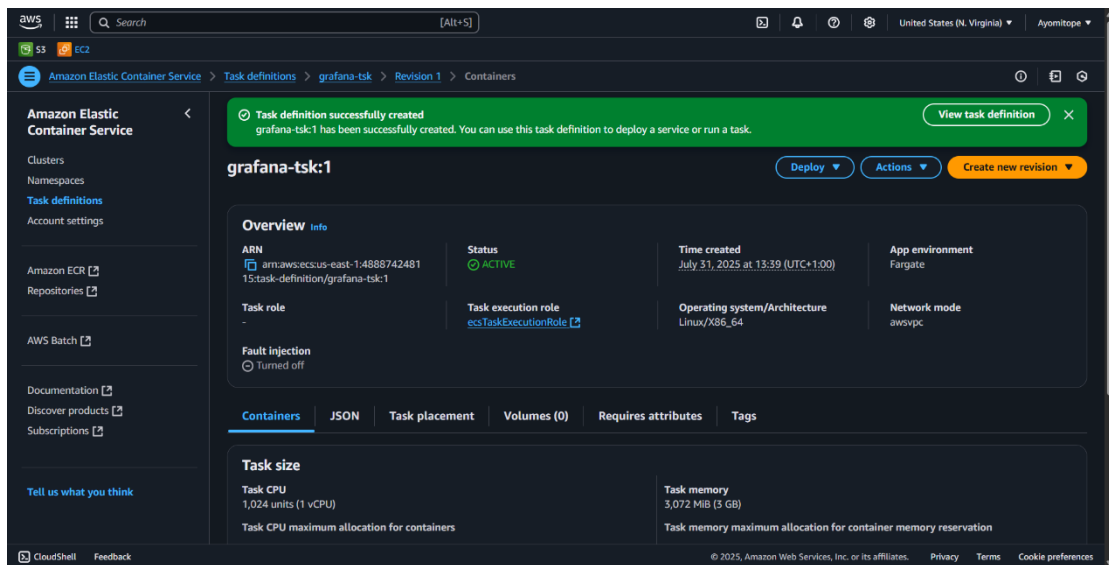
Name: Grafana

Image URI: grafana/grafana:latest

Port mappings

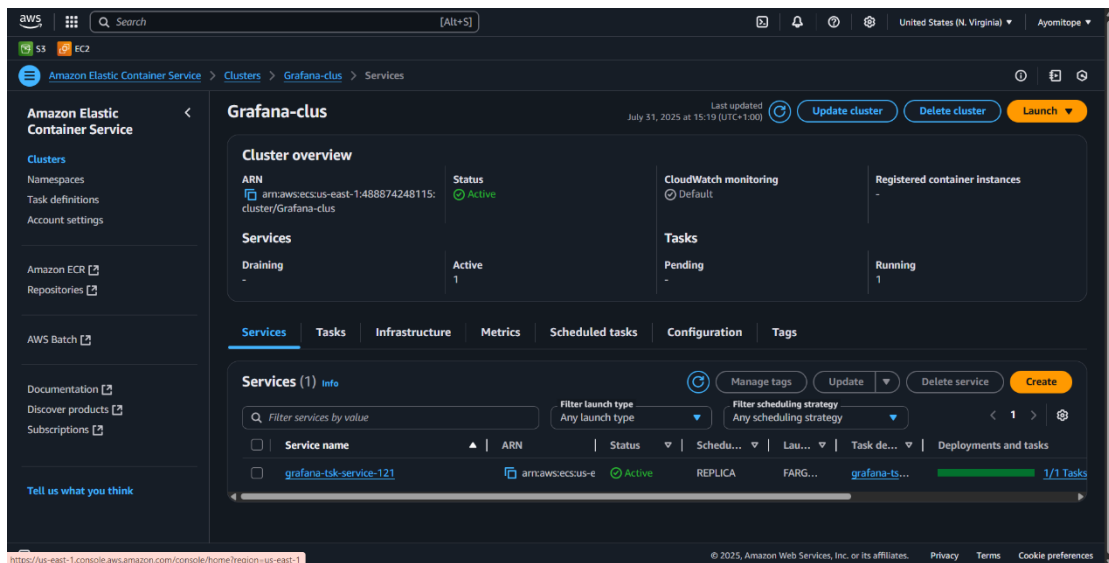
Container port: 3000

Protocol: tcp



Create an ECS Cluster

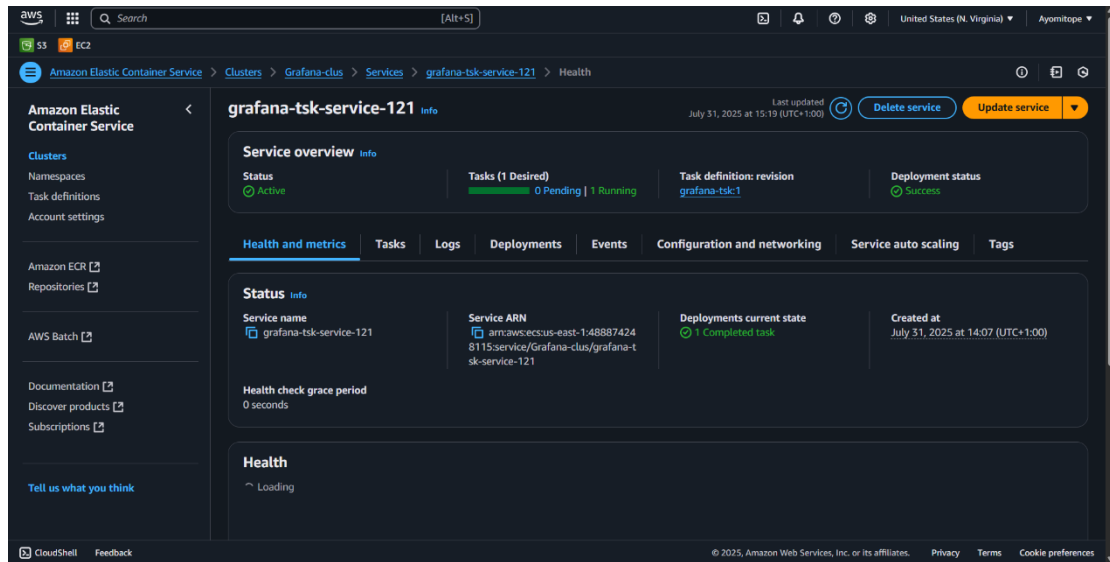
- Go to **ECS > Clusters > Create Cluster**.
- Choose “Networking only (Fargate)” type.
- Named the cluster.



Create a Service to Run the Task

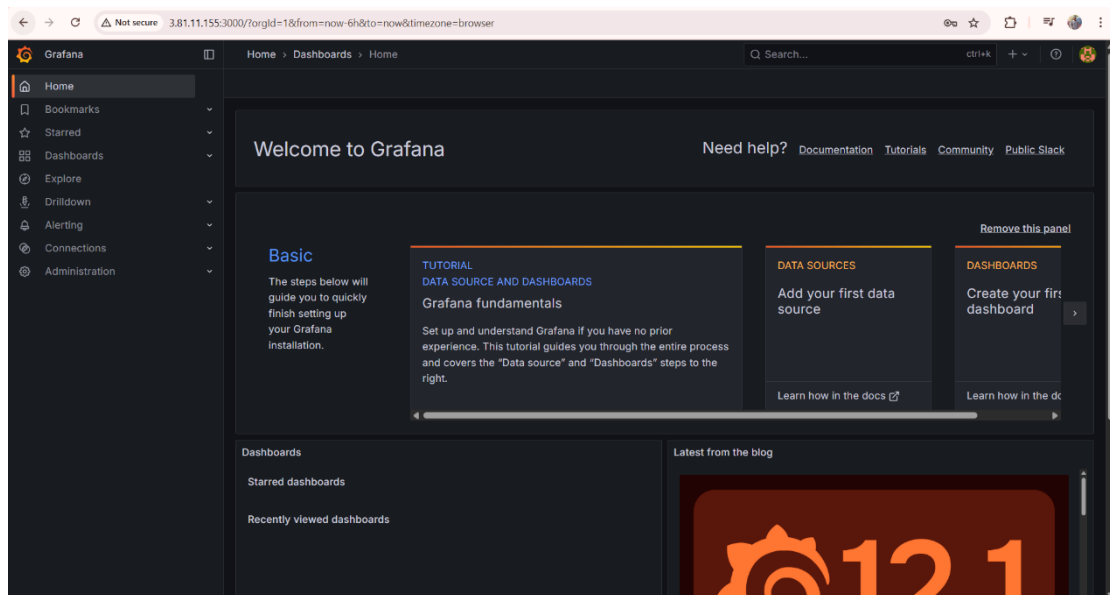
- In my cluster, create a **Service**:
- Launch type: Fargate.
- Task Definition: Select the one you created.
- Desired tasks: 1.

- Select VPC and public subnets.
- Enable **Auto-assign Public IP**



Access Grafana

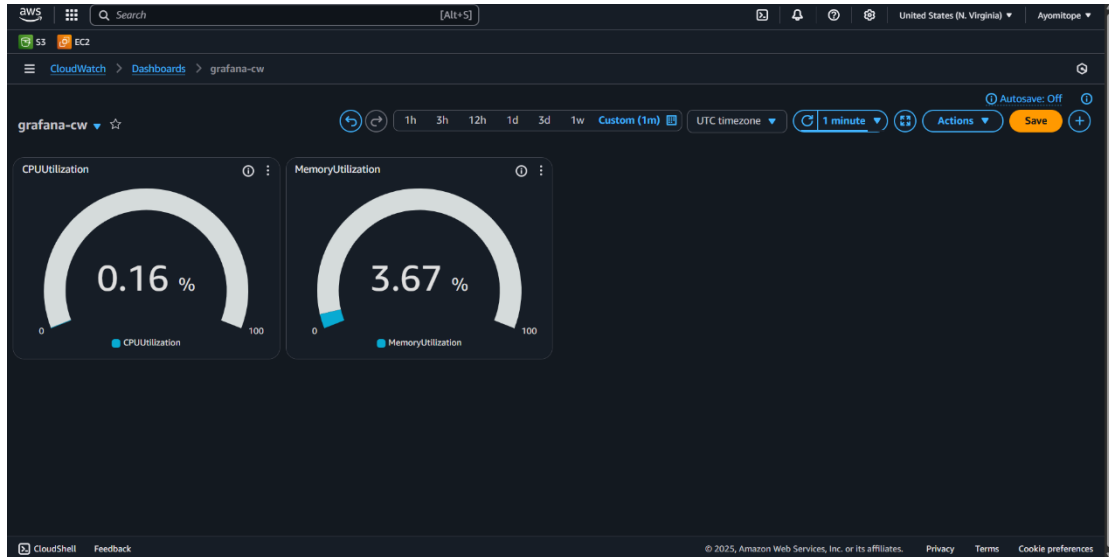
1. In **ECS > Network Interfaces**, find the ENI attached to my Grafana task.
2. Copy its **Public IPv4 address**. (3.81.11.155)
3. Open `http:// 3.81.11.155:3000` in my browser.



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 Instance

- 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

Dashboard

EC2 Global View

Events

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AHMs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Instance summary for i-07559f702d9462b89 (grafana-csn) info

Updated 1 minute ago

Instance ID

i-07559f702d9462b89

IPv6 address

-

Hostname type

IP name: ip-172-31-81-112.ec2.internal

Answer private resource DNS name

IPv4 (A)

Auto-assigned IP address

44.208.166.189 [Public IP]

IAM Role

-

IMDSv2

Required

Operator

-

Public IPv4 address

44.208.166.189 | open address

Instance state

Running

Private IP DNS name (IPv4 only)

ip-172-31-81-112.ec2.internal

Instance type

t2.micro

VPC ID

vpc-0ebb33f37ccbf0485

Subnet ID

subnet-0a27e0464943442d6

Instance ARN

arn:aws:ec2:us-east-1:488874248115:instance/i-07559f702d9462b89

Private IPv4 addresses

172.31.81.112

Public DNS

ec2-44-208-166-189.compute-1.amazonaws.com | open address

Elastic IP addresses

-

AWS Compute Optimizer finding

Opt-in to AWS Compute Optimizer for recommendations. | Learn more

Auto Scaling Group name

-

Managed

false

Connect

Instance state

Actions

Details

Configure and alarms

Monitoring

Security

Networking

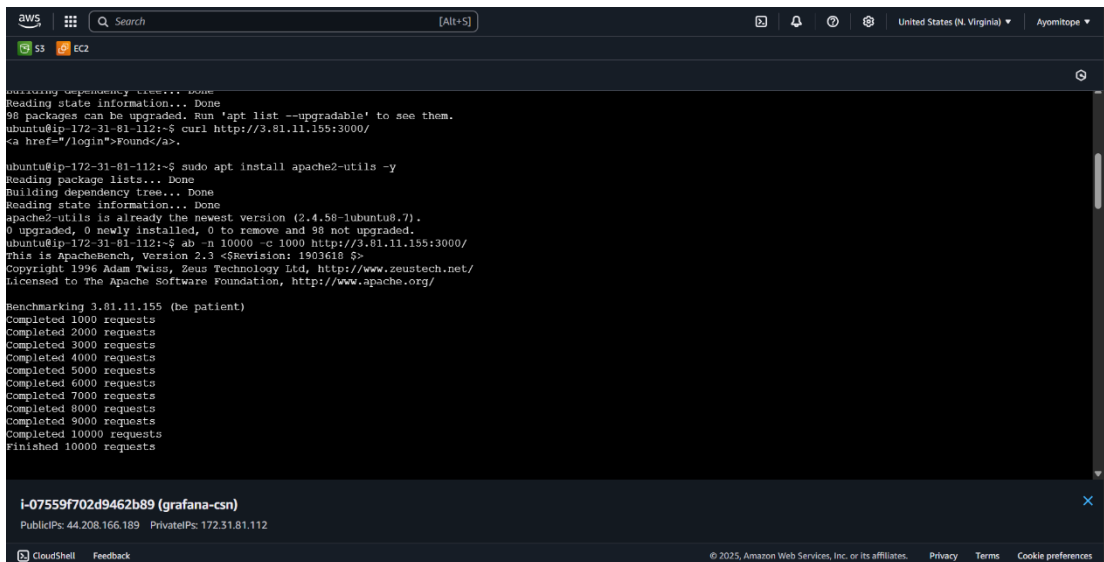
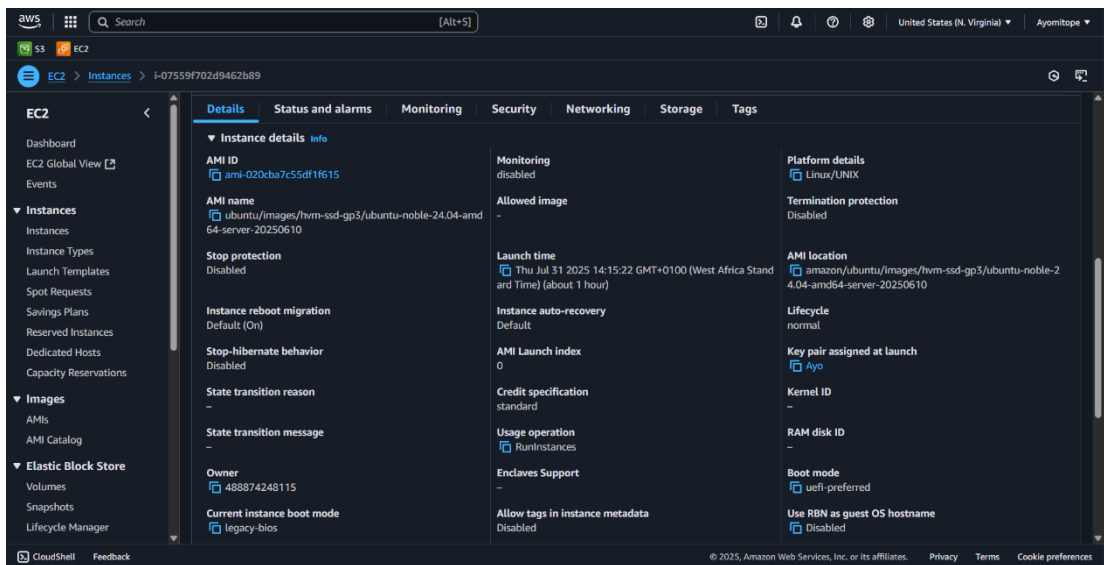
Storage

Tags

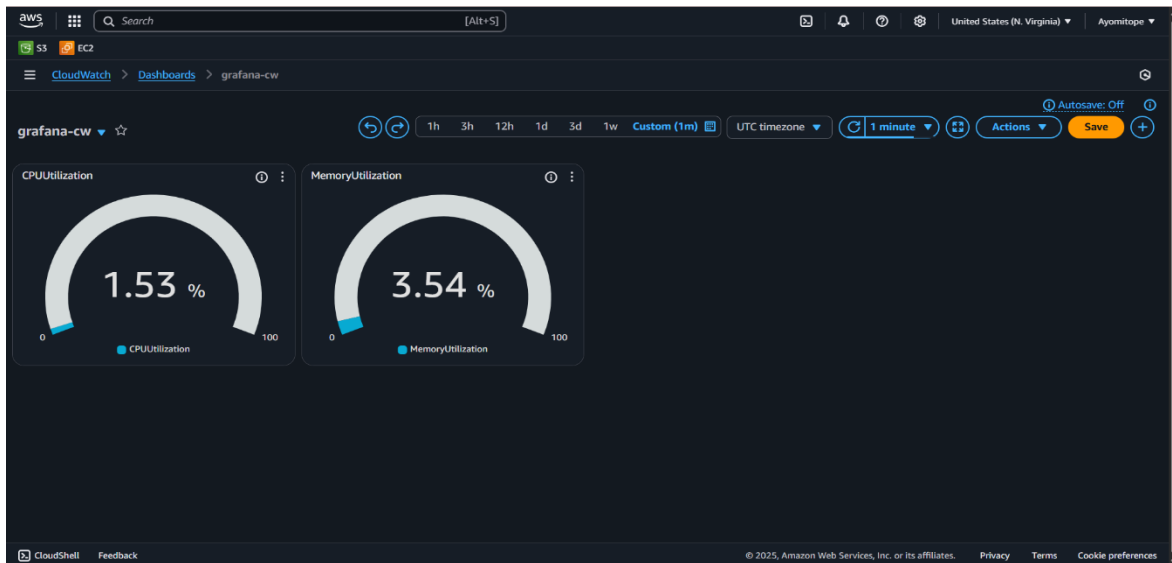
CloudShell

Feedback

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



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 +/- Stdev
Latency 125.89ms 69.17ms 447.89ms 64.28%
Req/Sec 666.37 243.23 1.79k 69.46%
238481 requests in 30.09s, 63.68MB read
Requests/sec: 7924.46
Transfer/sec: 2.12MB
ubuntu@ip-172-31-81-112:~$ wrk -t12 -c1000 -d60s http://3.81.11.155:3000
Running 1m test @ http://3.81.11.155:3000
12 threads and 1000 connections
Thread Stats Avg Stdev Max +/- Stdev
Latency 126.84ms 72.03ms 538.23ms 64.84%
Req/Sec 666.16 251.37 1.55k 66.90%
476448 requests in 1.00m, 127.23MB read
Requests/sec: 7926.63
Transfer/sec: 2.12MB
ubuntu@ip-172-31-81-112:~$
```

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

