Event Sphere

Mythreya Hardur Madhukeshwara - 121307484
Kunal Gadgil - 121058843
Sai Saketh Bolla - 121393579
Shengjie Liu - 117102260
Baelul Haile - 116451246
Darpan Nagendra Karur - 120481420
Github:
https://github.com/containerization-team-cems-eks/EventSphere

Table of Contents

Introduction

EventSphere is a Campus event management system that helps manage campus events like lectures, workshops, club activities, student programs, and more. Users are students and faculty who can use the website for searching and registering for events. Organizers interact with the website to manage events. The application is deployed in the cloud with EKS on AWS, and scales automatically, allowing for high availability.

Architecture & Design

The EventSphere application handles traffic with a microservices architecture, prioritizing decoupled components to ensure scalability and maintainability. By separating the system into distinct domains, i.e, Authentication, Events, Bookings, and Notifications, we enable independent development and deployment cycles for each core feature.

Core Components

Frontend Client: A React-based Single Page Application (SPA) serves as the entry point for users. It provides a responsive interface for browsing events, managing bookings, and administration. It interacts with the backend services via RESTful HTTP APIs.

Backend Services:

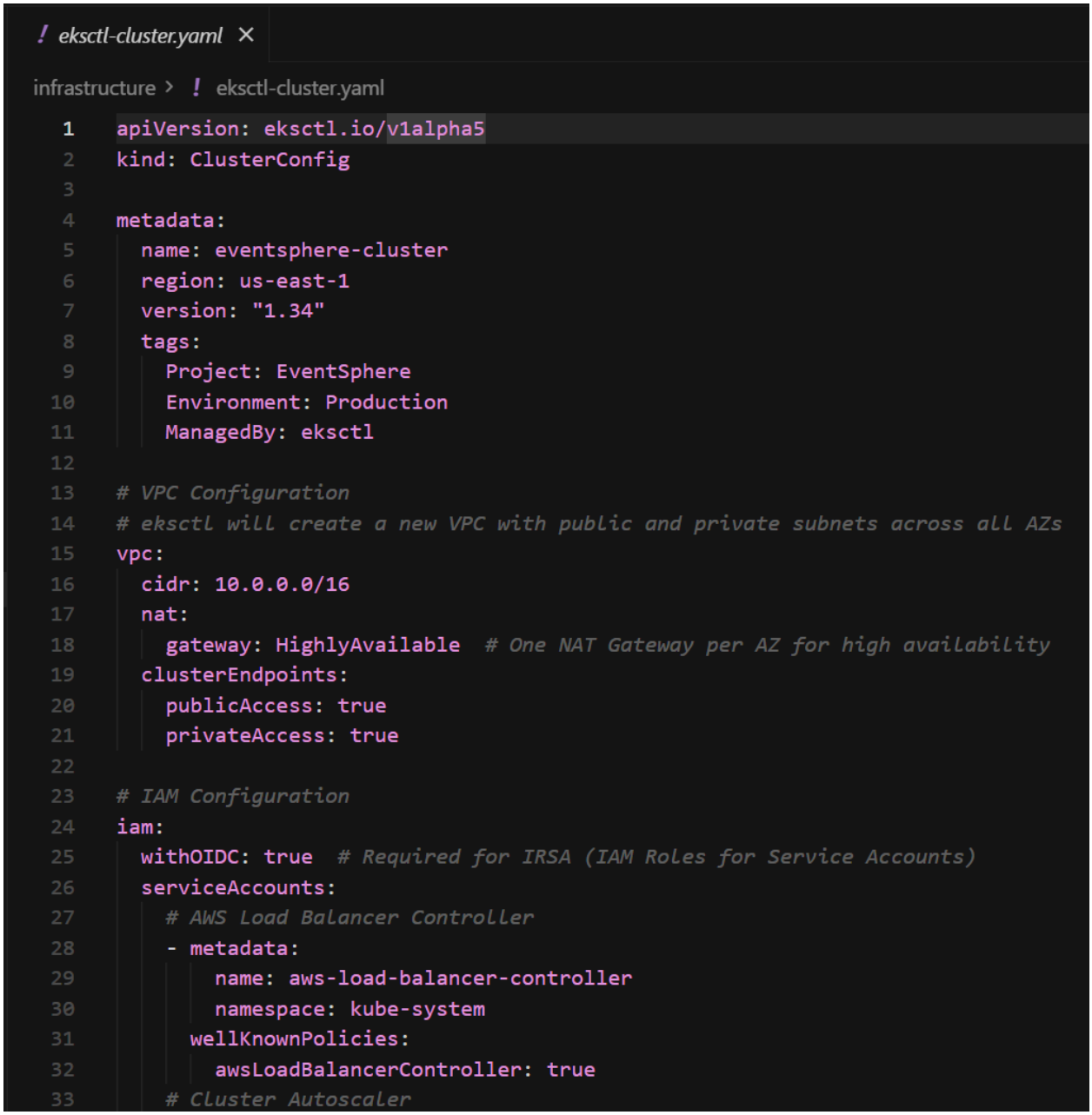
- **Auth Service**: Handles user registration and login, issuing JSON Web Tokens (JWT) to secure subsequent requests across the platform. It enforces Role-Based Access Control (RBAC) to distinguish between attendees and administrators.
- **Event Service**: Serves as the catalog source of truth, managing event metadata (locations, dates, prices) and inventory.
- **Booking Service**: Manages the transaction logic for reservations, ensuring that bookings are linked correctly to users and events while validating availability with the Event Service.
- **Notification Service**: A lightweight service designed to decouple email operations from the critical path. Instead of sending emails directly, it publishes messages to an AWS SNS topic, offloading the actual delivery logic to the cloud.

Data Layer: We utilize MongoDB as our persistent storage. While the services share a database cluster for development simplicity, the architecture supports logical separation, where each microservice owns its specific collections (e.g., auth, events, bookings), preventing tight data coupling.

Infrastructure & Orchestration: The application is containerized using Docker and orchestrated via Amazon EKS (Elastic Kubernetes Service). This cloud-native approach allows for features like Horizontal Pod Autoscaling (HPA) and automated health checks. Traffic is routed into the cluster via an AWS Application Load Balancer (ALB).

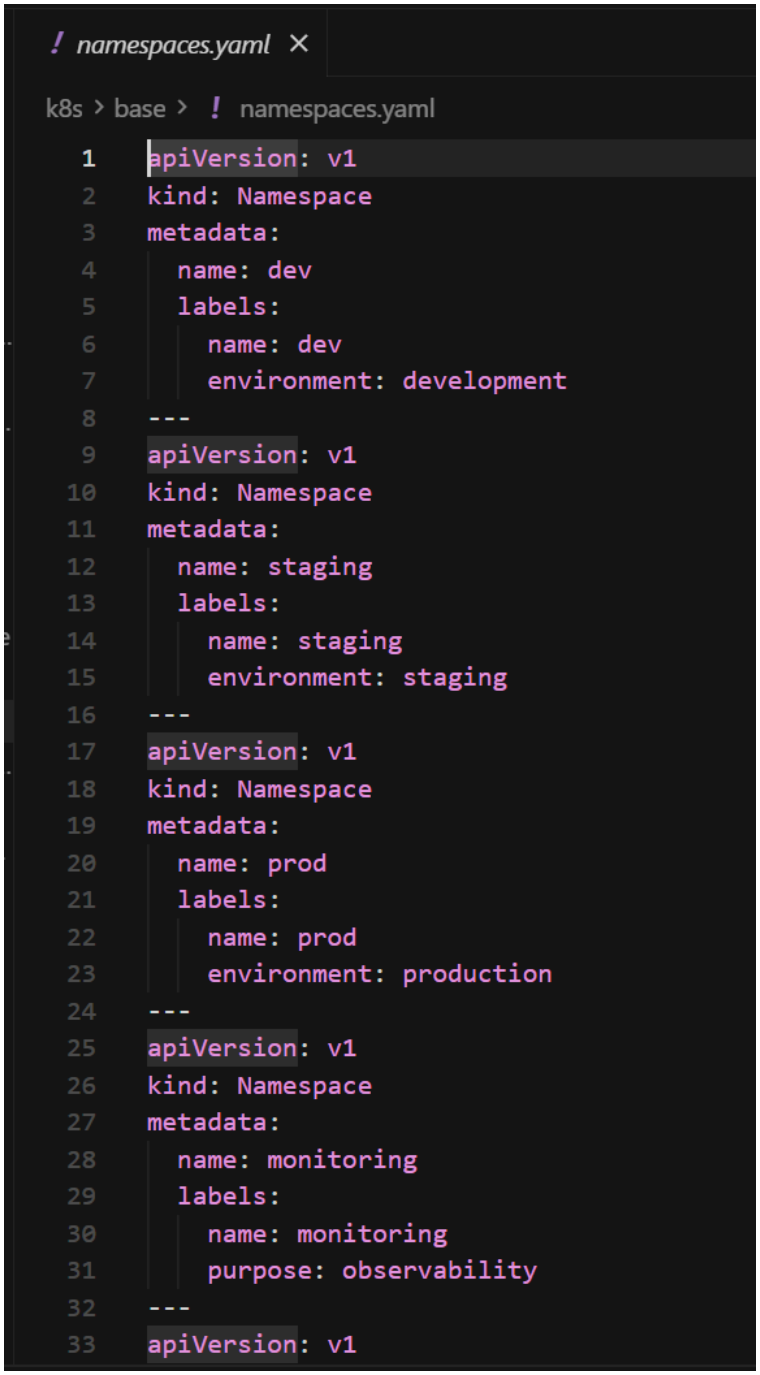
Infrastructure-as-Code

- eksctl IaC template VPC, subnets, node groups, and the EKS control plane: [eks-cluster.yaml]



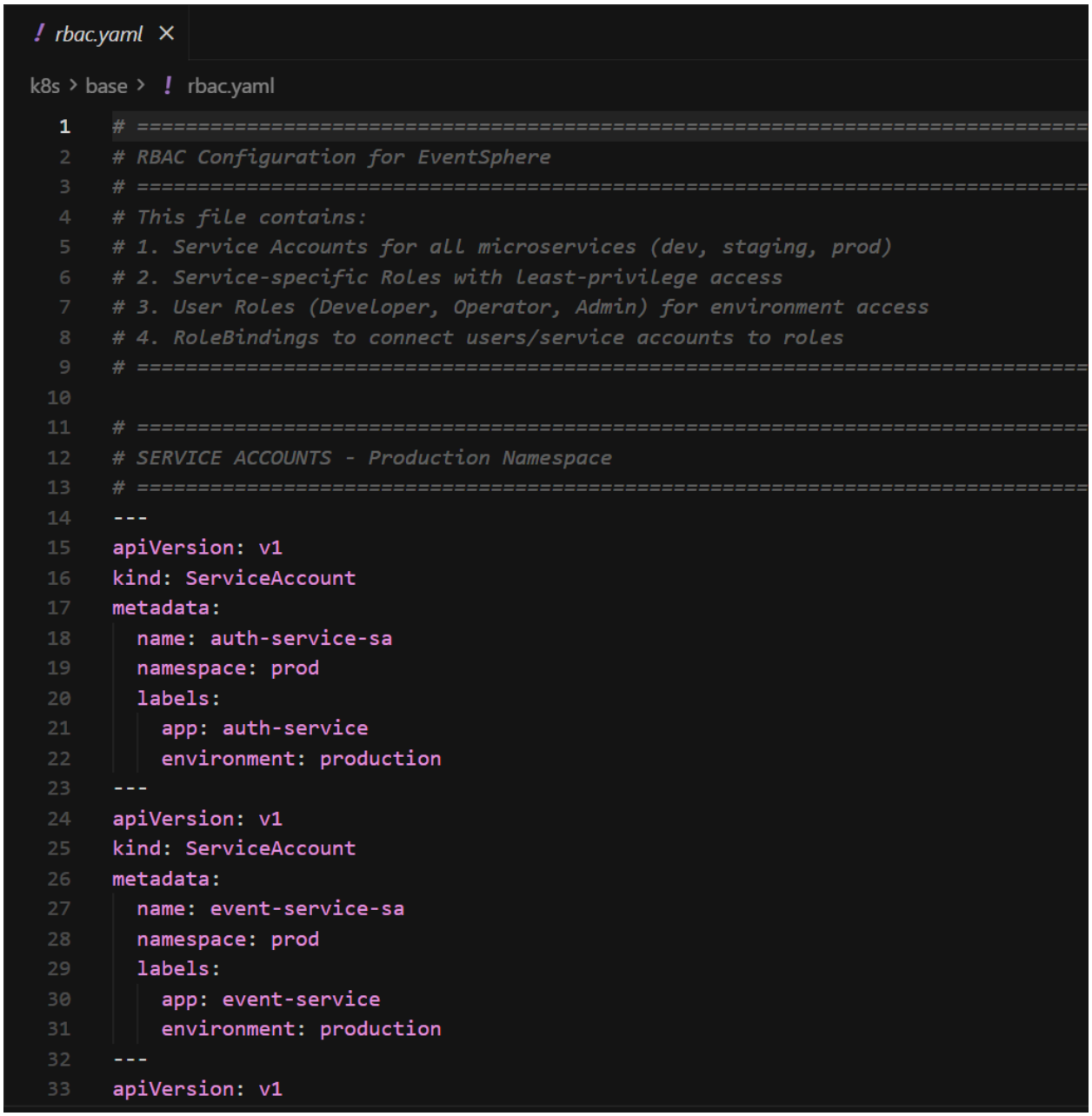
- RBAC manifests for namespaces and service accounts (including IRSA):
    - Namespaces:

namespaces.yaml



- Service Accounts:

rbac.yaml

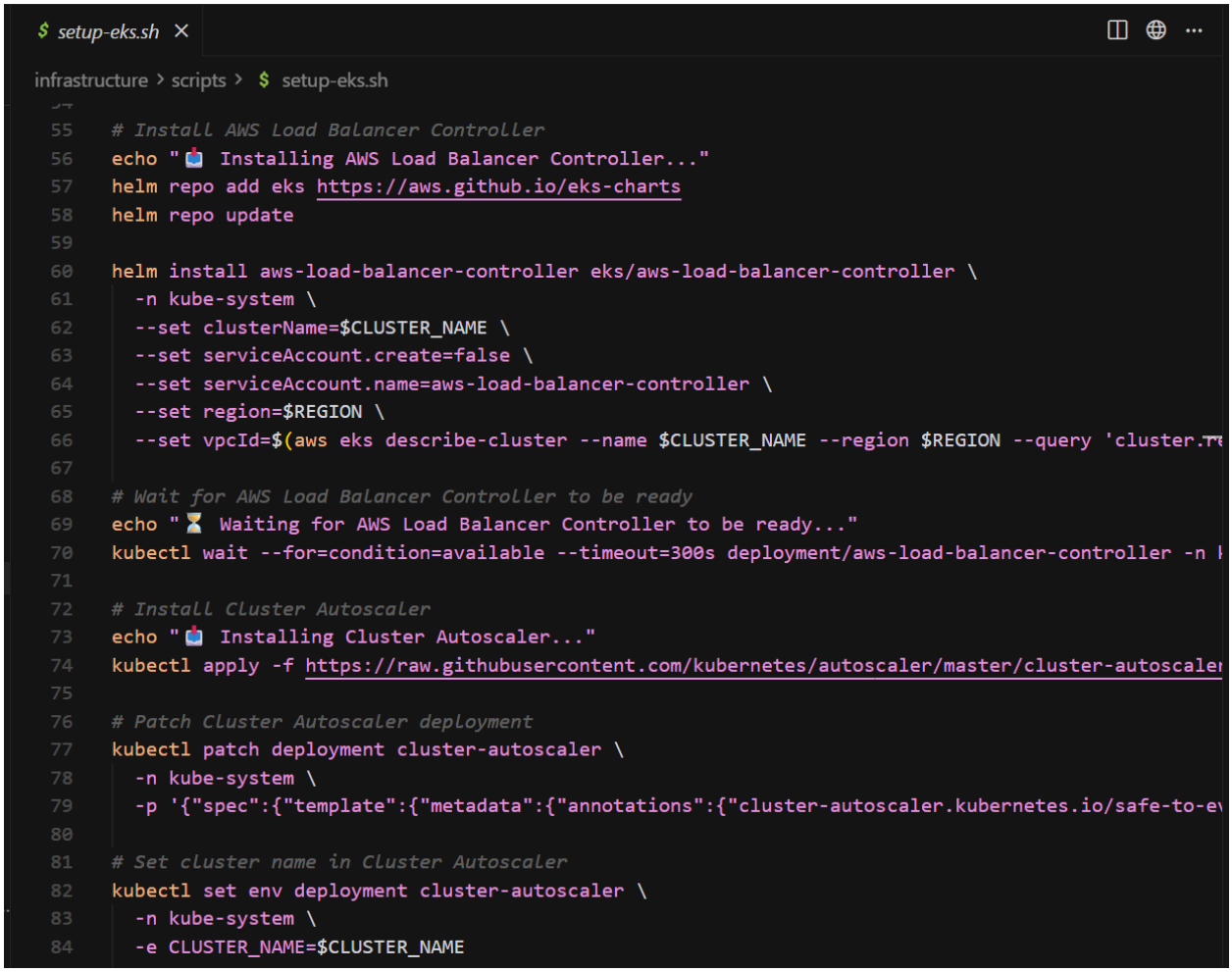


Docker and Kubernetes Configuration

Deploy and Manage a Kubernetes Cluster on AWS EKS:

- Installed add-ons (ALB Controller, Cluster Autoscaler) with configs:

[setup-eks.sh]



- kubectl get nodes -o wide shows nodes in multiple AZs:

```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventS
phere$ kubectl get nodes -o wide
NAME                         STATUS   ROLES    AGE   VERSION               INTERNAL-IP   EXTERNAL-IP      OS-IMAGE
             KERNEL-VERSION                  CONTAINER-RUNTIME
ip-10-0-39-3.ec2.internal    Ready    <none>   15h   v1.34.2-eks-ecaa3a6   10.0.39.3     184.72.214.172   Amazon Linux 2023
.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-4-244.ec2.internal   Ready    <none>   15h   v1.34.2-eks-ecaa3a6   10.0.4.244    34.204.168.21    Amazon Linux 2023
.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-40-82.ec2.internal   Ready    <none>   15h   v1.34.2-eks-ecaa3a6   10.0.40.82    107.20.100.184   Amazon Linux 2023
.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-41-129.ec2.internal  Ready    <none>   15h   v1.34.2-eks-ecaa3a6   10.0.41.129   54.81.207.250    Amazon Linux 2023
.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-6-193.ec2.internal   Ready    <none>   15h   v1.34.2-eks-ecaa3a6   10.0.6.193    98.92.155.102    Amazon Linux 2023
.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventS
phere$ kubectl get nodes -o custom-columns=NAME:.metadata.name,ZONE:.metadata.labels."topology\.kubernetes\.io/zone",INSTANC
E-TYPE:.metadata.labels."node\.kubernetes\.io/instance-type",STATUS:.status.conditions[-1].type
NAME                         ZONE         INSTANCE-TYPE   STATUS
ip-10-0-39-3.ec2.internal    us-east-1b   t3.small        Ready
ip-10-0-4-244.ec2.internal   us-east-1f   t3.small        Ready
ip-10-0-40-82.ec2.internal   us-east-1b   t3.small        Ready
ip-10-0-41-129.ec2.internal  us-east-1b   t3.small        Ready
ip-10-0-6-193.ec2.internal   us-east-1f   t3.small        Ready
```

- Autoscaler responds to load changes and logs scale events:
    1. Total node count 5 before load test:

```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventS
phere$ kubectl get nodes -o wide
NAME                         STATUS   ROLES    AGE   VERSION               INTERNAL-IP   EXTERNAL-IP      OS-IMAGE
             KERNEL-VERSION                  CONTAINER-RUNTIME
ip-10-0-39-3.ec2.internal    Ready    <none>   15h   v1.34.2-eks-ecaa3a6   10.0.39.3     184.72.214.172   Amazon Linux 2023
.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-4-244.ec2.internal   Ready    <none>   15h   v1.34.2-eks-ecaa3a6   10.0.4.244    34.204.168.21    Amazon Linux 2023
.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-40-82.ec2.internal   Ready    <none>   15h   v1.34.2-eks-ecaa3a6   10.0.40.82    107.20.100.184   Amazon Linux 2023
.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-41-129.ec2.internal  Ready    <none>   15h   v1.34.2-eks-ecaa3a6   10.0.41.129   54.81.207.250    Amazon Linux 2023
.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-6-193.ec2.internal   Ready    <none>   15h   v1.34.2-eks-ecaa3a6   10.0.6.193    98.92.155.102    Amazon Linux 2023
.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
```

    2. Node group 1 node count before load test (desiredSize=2):

```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/C
yber/Sem 3/ENPM818R - Virtualization/Final project 2/EventSphere$
 aws eks describe-nodegroup --cluster-name eventsphere-cluster --
nodegroup-name eventsphere-mng-1 --region us-east-1 --query 'node
group.scalingConfig'
{
    "minSize": 2,
    "maxSize": 5,
    "desiredSize": 2
}
```

    3. Starting load test:

```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventSphere$ kubectl apply -f ./k8s/load-test.yaml
deployment.apps/load-test created
```

4. Node group 1 node count during load test (desiredSize=5):

```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventSphere$ aws eks describe-nodegroup --cluster-name eventsphere-cluster --nodegroup-name eventsphere-mng-1 --region us-east-1 --query 'nodegroup.scalingConfig'
{
    "minSize": 2,
    "maxSize": 5,
    "desiredSize": 5
}
```

5. New nodes created and total node count is now 8:

```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventSphere$ kubectl get nodes -o wide
NAME                          STATUS   ROLES    AGE     VERSION               INTERNAL-IP   EXTERNAL-IP      OS-IMAGE
KERNEL-VERSION                  CONTAINER-RUNTIME
ip-10-0-11-113.ec2.internal   Ready    <none>   20h     v1.34.2-eks-ecaa3a6   10.0.11.113   44.203.89.35     Amazon Linux 20
23.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-31-26.ec2.internal    Ready    <none>   21h     v1.34.2-eks-ecaa3a6   10.0.31.26    44.203.197.239   Amazon Linux 20
23.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-4-42.ec2.internal     Ready    <none>   3m22s   v1.34.2-eks-ecaa3a6   10.0.4.42     18.205.189.23    Amazon Linux 20
23.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-40-129.ec2.internal   Ready    <none>   21h     v1.34.2-eks-ecaa3a6   10.0.40.129   18.207.240.232   Amazon Linux 20
23.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-43-2.ec2.internal     Ready    <none>   9h      v1.34.2-eks-ecaa3a6   10.0.43.2     44.192.19.69     Amazon Linux 20
23.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-50-151.ec2.internal   Ready    <none>   3m25s   v1.34.2-eks-ecaa3a6   10.0.50.151   3.230.158.223    Amazon Linux 20
23.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-55-2.ec2.internal     Ready    <none>   21h     v1.34.2-eks-ecaa3a6   10.0.55.2     3.238.116.104    Amazon Linux 20
23.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-60-236.ec2.internal   Ready    <none>   3m24s   v1.34.2-eks-ecaa3a6   10.0.60.236   98.93.42.178     Amazon Linux 20
23.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
```

6. Autoscaler logs showing scale up event:

```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Mary
land/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/
EventSphere$ kubectl get events -n kube-system --sort-by='.
lastTimestamp' | grep -i scale-up
9m8s        Normal    ScaledUpGroup      configmap/cluster-
autoscaler-status   Scale-up: setting group eks-eventsphere
-mng-1-dacd7a87-de3e-36a2-dbd2-c697cda37bfe size to 5 inste
ad of 2 (max: 5)
9m7s        Normal    ScaledUpGroup      configmap/cluster-
autoscaler-status   Scale-up: group eks-eventsphere-mng-1-d
acd7a87-de3e-36a2-dbd2-c697cda37bfe size set to 5 instead o
f 2 (max: 5)
```

7. Deleting load-test deployment to trigger scale-down:

```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/
Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventSpher
e$ kubectl delete deployment load-test -n prod
deployment.apps "load-test" deleted from prod namespace
```

8. Scale-down logs:

```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/
Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventSpher
e$ kubectl logs -n kube-system -l app=cluster-autoscaler --tail=
100 | grep -i "scale.*down\|removed\|drain"
I1207 16:22:40.716653       1 eligibility.go:163] Node ip-10-0-1
1-113.ec2.internal unremovable: memory requested (52.0856% of al
locatable) is above the scale-down utilization threshold
I1207 16:22:40.716695       1 eligibility.go:163] Node ip-10-0-3
1-26.ec2.internal unremovable: memory requested (55.9954% of all
ocatable) is above the scale-down utilization threshold
I1207 16:22:40.716728       1 eligibility.go:163] Node ip-10-0-4
0-129.ec2.internal unremovable: memory requested (77.7791% of al
locatable) is above the scale-down utilization threshold
I1207 16:22:40.716759       1 eligibility.go:163] Node ip-10-0-5
5-2.ec2.internal unremovable: memory requested (86.437% of alloc
atable) is above the scale-down utilization threshold
I1207 16:22:40.716771       1 eligibility.go:104] Scale-down cal
culation: ignoring 1 nodes unremovable in the last 5m0s
I1207 16:22:40.716819       1 cluster.go:173] node ip-10-0-50-15
1.ec2.internal may be removed
I1207 16:22:40.716856       1 cluster.go:173] node ip-10-0-60-23
6.ec2.internal may be removed
I1207 16:22:40.716907       1 cluster.go:173] node ip-10-0-4-42.
ec2.internal may be removed
I1207 16:22:40.717160       1 static_autoscaler.go:598] Scale do
wn status: lastScaleUpTime=2025-12-07 16:10:35.936363273 +0000 U
TC m=+75397.498723648 lastScaleDownDeleteTime=2025-12-06 18:14:2
0.939505455 +0000 UTC m=-3577.498134162 lastScaleDownFailTime=20
25-12-06 18:14:20.939505455 +0000 UTC m=-3577.498134162 scaleDow
nForbidden=false scaleDownInCooldown=false
I1207 16:22:40.717211       1 static_autoscaler.go:619] Starting
 scale down
```

9. Node group 1 node count after scale-down (desiredSize=2):

```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/
Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventSpher
e$ kubectl delete deployment load-test -n prod
deployment.apps "load-test" deleted from prod namespace
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/
Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventSpher
e$ aws eks describe-nodegroup --cluster-name eventsphere-cluster
 --nodegroup-name eventsphere-mng-1 --region us-east-1 --query '
nodegroup.scalingConfig'
{
    "minSize": 2,
    "maxSize": 5,
    "desiredSize": 2
}
```

10. Total node count 5 after scale-down:

```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventS
phere$ kubectl get nodes -o wide
NAME                          STATUS   ROLES    AGE   VERSION               INTERNAL-IP    EXTERNAL-IP      OS-IMAGE
          KERNEL-VERSION                  CONTAINER-RUNTIME
ip-10-0-39-3.ec2.internal     Ready    <none>   15h   v1.34.2-eks-ecaa3a6   10.0.39.3      184.72.214.172   Amazon Linux 2023
.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-4-244.ec2.internal    Ready    <none>   15h   v1.34.2-eks-ecaa3a6   10.0.4.244     34.204.168.21    Amazon Linux 2023
.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-40-82.ec2.internal    Ready    <none>   15h   v1.34.2-eks-ecaa3a6   10.0.40.82     107.20.100.184   Amazon Linux 2023
.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-41-129.ec2.internal   Ready    <none>   15h   v1.34.2-eks-ecaa3a6   10.0.41.129    54.81.207.250    Amazon Linux 2023
.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
ip-10-0-6-193.ec2.internal    Ready    <none>   15h   v1.34.2-eks-ecaa3a6   10.0.6.193     98.92.155.102    Amazon Linux 2023
.9.20251117   6.12.55-74.119.amzn2023.x86_64   containerd://2.1.4
```

- At least one workload uses IRSA for AWS access:
    1. External Secrets Operator Uses IRSA:

```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Marylan
d/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventS
phere$ kubectl get serviceaccount external-secrets -n external
-secrets-system -o yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  annotations:
    eks.amazonaws.com/role-arn: arn:aws:iam::277707118728:role
/external-secrets-role
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","kind":"ServiceAccount","metadata":{"
annotations":{},"creationTimestamp":null,"name":"external-secr
ets","namespace":"external-secrets-system"}}
  creationTimestamp: "2025-12-06T19:05:19Z"
  labels:
    app.kubernetes.io/managed-by: eksctl
  name: external-secrets
  namespace: external-secrets-system
  resourceVersion: "6037"
  uid: d13b643b-fdbc-4f35-8086-30b780237d6c
```

- ALB or NGINX Ingress Controller routes public traffic correctly:
    1. Curl to ALB DNS responds with 301 status and location as https://www.enpm818rgroup7.work.gd:443/ to enforce HTTPS:

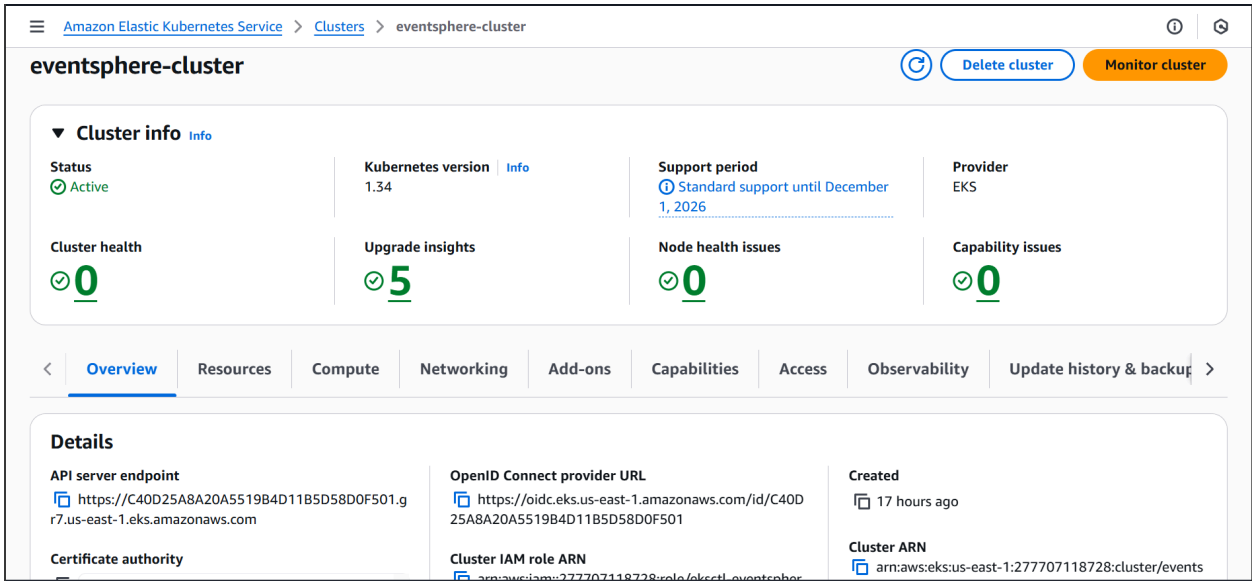
```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Marylan
d/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventS
phere$ curl -v -H "Host: www.enpm818rgroup7.work.gd" http://$A
LB_DNS/
* Host k8s-prod-eventsph-395841f6f1-1344915046.us-east-1.elb.a
mazonaws.com:80 was resolved.
* IPv6: (none)
* IPv4: 52.6.253.22, 3.221.240.165
*   Trying 52.6.253.22:80...
* Connected to k8s-prod-eventsph-395841f6f1-1344915046.us-east
-1.elb.amazonaws.com (52.6.253.22) port 80
> GET / HTTP/1.1
> Host: www.enpm818rgroup7.work.gd
> User-Agent: curl/8.5.0
> Accept: */*
>
< HTTP/1.1 301 Moved Permanently
< Server: awselb/2.0
< Date: Sun, 07 Dec 2025 17:08:04 GMT
< Content-Type: text/html
< Content-Length: 134
< Connection: keep-alive
< Location: https://www.enpm818rgroup7.work.gd:443/
<
<html>
<head><title>301 Moved Permanently</title></head>
<body>
<center><h1>301 Moved Permanently</h1></center>
</body>
</html>
* Connection #0 to host k8s-prod-eventsph-395841f6f1-134491504
6.us-east-1.elb.amazonaws.com left intact
```

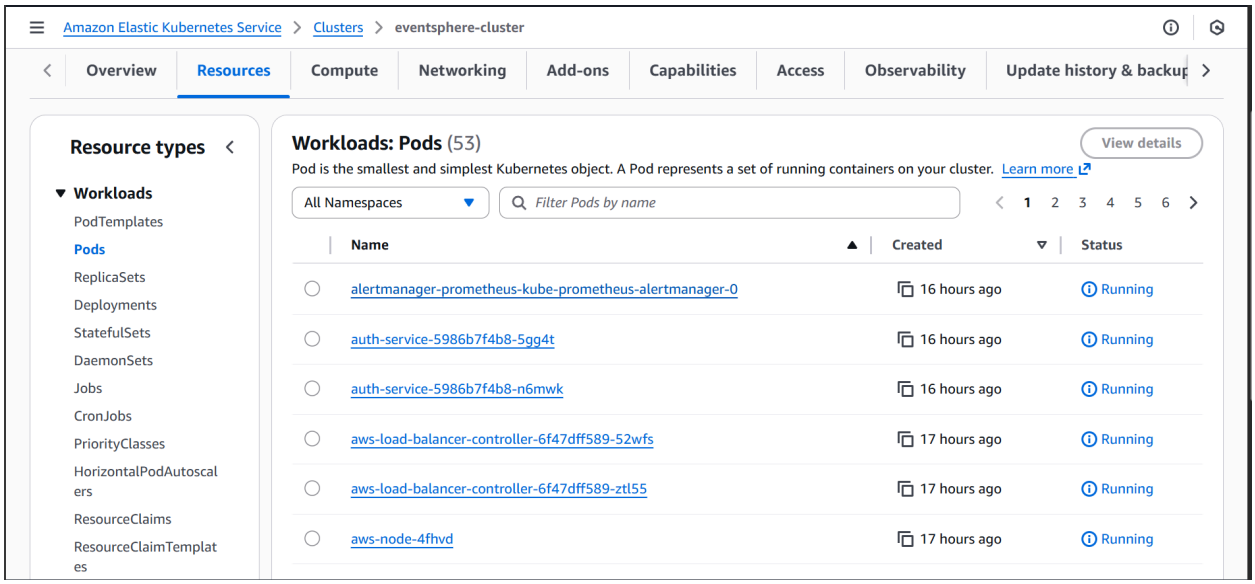
    2. Curl to https://www.enpm818rgroup7.work.gd:443/

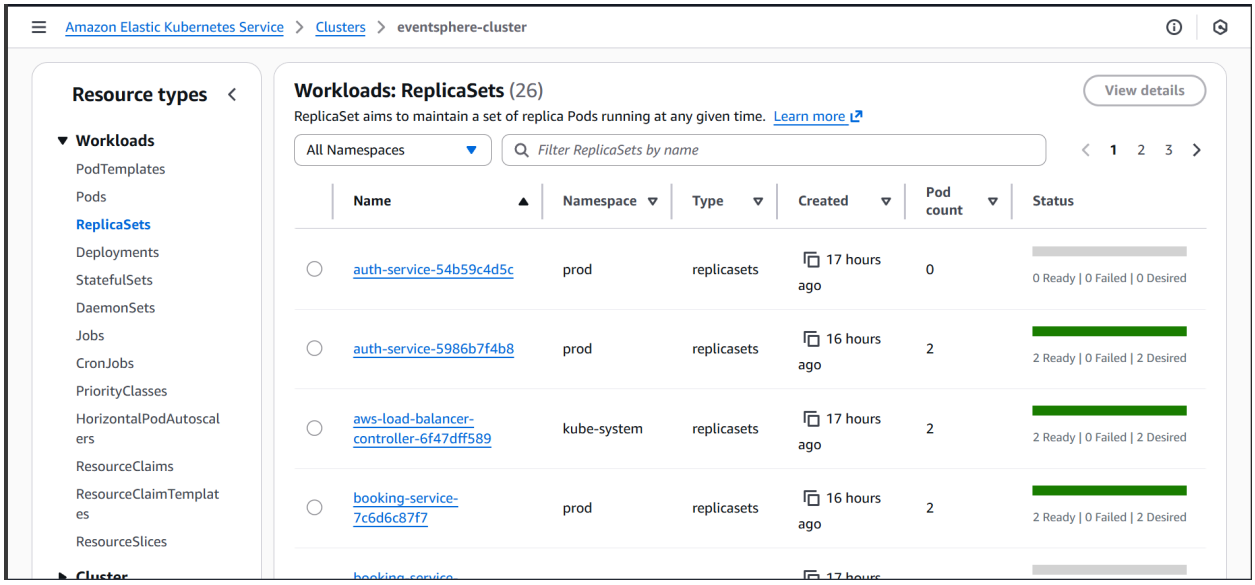
```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventS
phere$ curl -v -H "Host: www.enpm818rgroup7.work.gd" https://www.enpm818rgroup7.work.gd:443/
* Host www.enpm818rgroup7.work.gd:443 was resolved.
* IPv6: (none)
* IPv4: 3.221.240.165, 52.6.253.22
*   Trying 3.221.240.165:443...
* Connected to www.enpm818rgroup7.work.gd (3.221.240.165) port 443
* ALPN: curl offers h2,http/1.1
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
*  CAfile: /etc/ssl/certs/ca-certificates.crt
*  CApath: /etc/ssl/certs
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
* TLSv1.2 (IN), TLS handshake, Server finished (14):
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
* TLSv1.2 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.2 (OUT), TLS handshake, Finished (20):
* TLSv1.2 (IN), TLS handshake, Finished (20):
* SSL connection using TLSv1.2 / ECDHE-RSA-AES128-GCM-SHA256 / prime256v1 / rsaEncryption
* ALPN: server accepted h2
* Server certificate:
*  subject: CN=enpm818rgroup7.work.gd
*  start date: Nov 22 00:00:00 2025 GMT
*  expire date: Dec 21 23:59:59 2026 GMT
*  subjectAltName: host "www.enpm818rgroup7.work.gd" matched cert's "www.enpm818rgroup7.work.gd"
*  issuer: C=US; O=Amazon; CN=Amazon RSA 2048 M04
*  SSL certificate verify ok.
*   Certificate level 0: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
*   Certificate level 1: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
*   Certificate level 2: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
* using HTTP/2
* [HTTP/2] [1] OPENED stream for https://www.enpm818rgroup7.work.gd:443/
* [HTTP/2] [1] [:method: GET]
* [HTTP/2] [1] [:scheme: https]
```

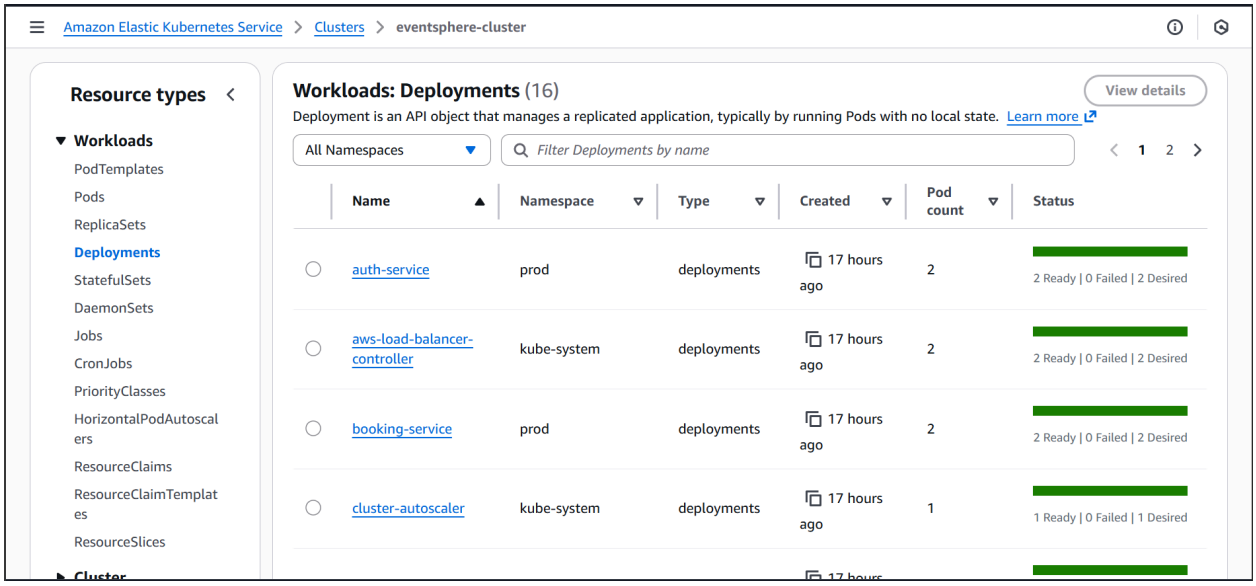
```
*  issuer: C=US; O=Amazon; CN=Amazon RSA 2048 M04
*  SSL certificate verify ok.
*   Certificate level 0: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
*   Certificate level 1: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
*   Certificate level 2: Public key type RSA (2048/112 Bits/secBits), signed using sha256WithRSAEncryption
* using HTTP/2
* [HTTP/2] [1] OPENED stream for https://www.enpm818rgroup7.work.gd:443/
* [HTTP/2] [1] [:method: GET]
* [HTTP/2] [1] [:scheme: https]
* [HTTP/2] [1] [:authority: www.enpm818rgroup7.work.gd]
* [HTTP/2] [1] [:path: /]
* [HTTP/2] [1] [user-agent: curl/8.5.0]
* [HTTP/2] [1] [accept: */*]
> GET / HTTP/2
> Host: www.enpm818rgroup7.work.gd
> User-Agent: curl/8.5.0
> Accept: */*
>
< HTTP/2 200
< date: Sun, 07 Dec 2025 17:11:51 GMT
< content-type: text/html
< content-length: 936
< server: nginx/1.29.2
< last-modified: Thu, 04 Dec 2025 07:32:27 GMT
< etag: "6931390b-3a8"
< accept-ranges: bytes
<
* Connection #0 to host www.enpm818rgroup7.work.gd left intact
<!doctype html><html lang="en"><head><meta charset="utf-8"/><link rel="icon" href="/favicon.ico"/><meta name="viewport" cont
ent="width=device-width,initial-scale=1"/><meta name="theme-color" content="#000000"/><meta name="description" content="Web
site created using create-react-app"/><link rel="apple-touch-icon" href="/logo192.png"/><link rel="manifest" href="/manifest
.json"/><title>Campus Event Management</title><link rel="preconnect" href="https://fonts.googleapis.com"><link rel="preconne
ct" href="https://fonts.gstatic.com" crossorigin><link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;50
0;700&family=Montserrat:wght@400;500;600;700&display=swap" rel="stylesheet"><script defer="defer" src="/static/js/main.81c7b
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventS
```

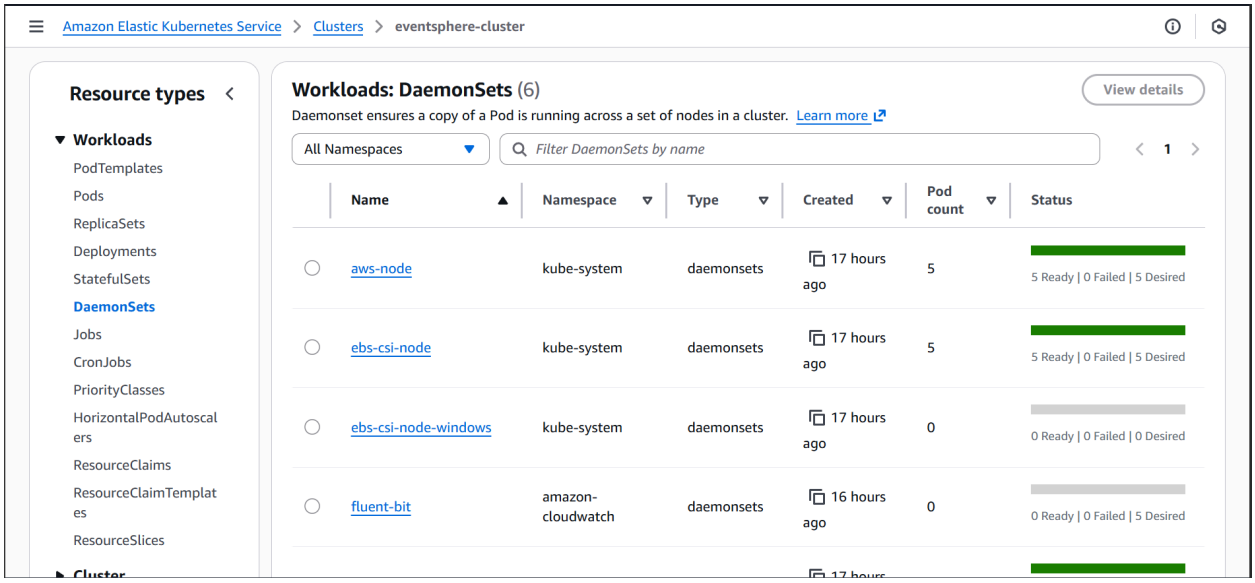
- EKS Console:

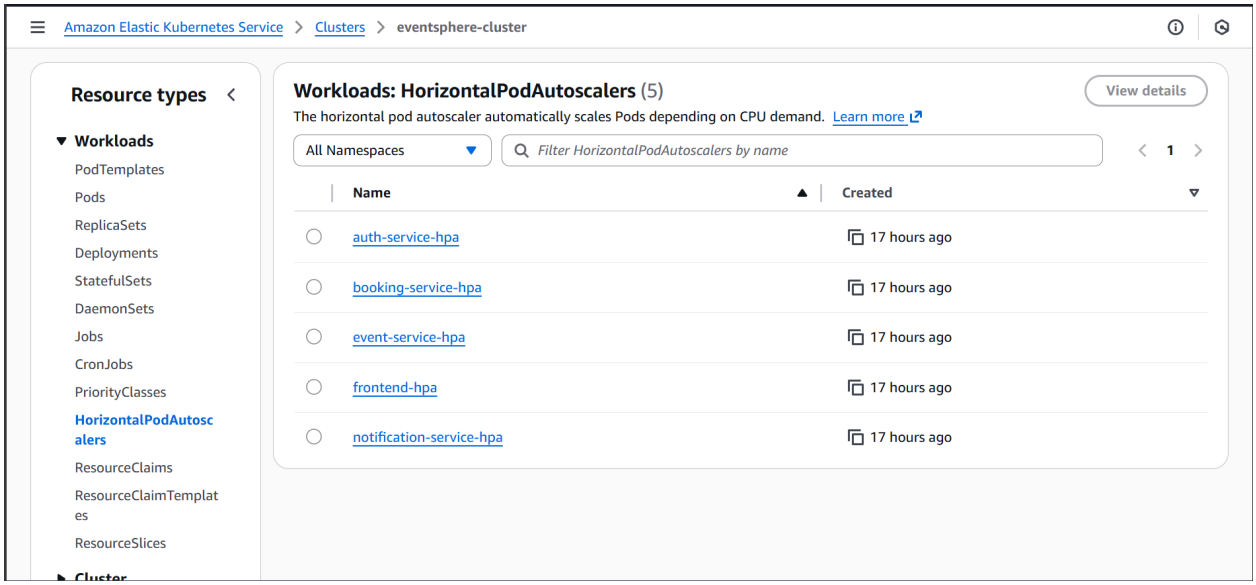

Amazon Elastic Kubernetes Service > Clusters > eventsphere-cluster
eventsphere-cluster
Delete cluster
Monitor cluster
Cluster info Info
Status
Active
Kubernetes version Info
1.34
Support period
Standard support until December 1, 2026
Provider
EKS
Cluster health
0
Upgrade insights
5
Node health issues
0
Capability issues
0
Overview
Resources
Compute
Networking
Add-ons
Capabilities
Access
Observability
Update history & backup
Details
API server endpoint
https://C40D25A8A20A5519B4D11B5D58D0F501.gr7.us-east-1.eks.amazonaws.com
OpenID Connect provider URL
https://oidc.eks.us-east-1.amazonaws.com/id/C40D25A8A20A5519B4D11B5D58D0F501
Created
17 hours ago
Certificate authority
Cluster IAM role ARN
Cluster ARN
arn:aws:eks:us-east-1:277707118728:cluster/events

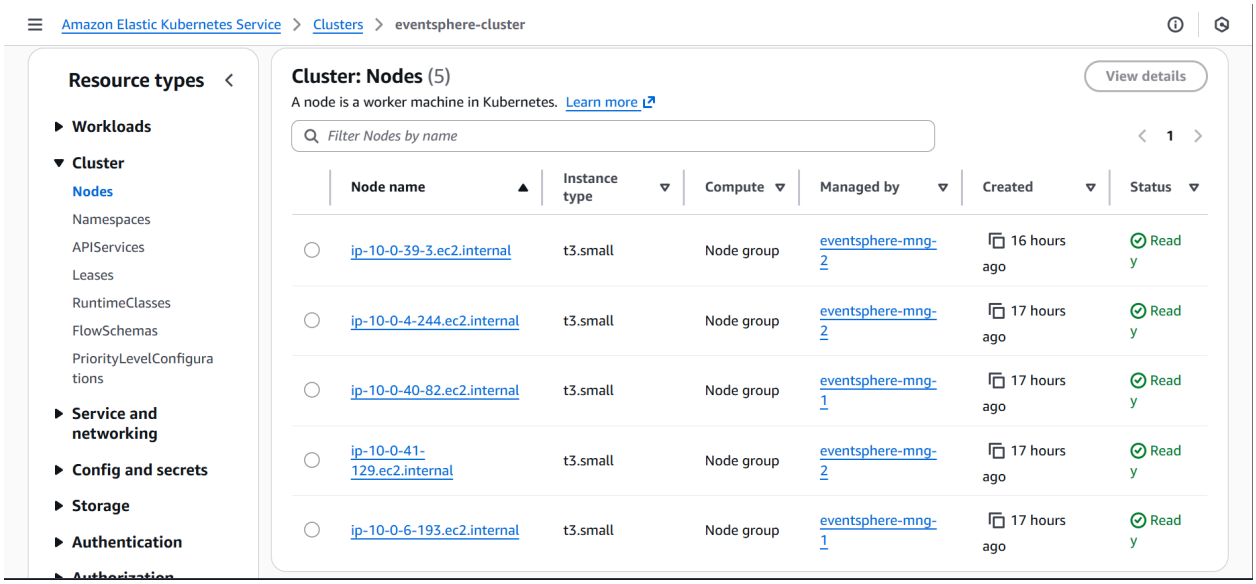

Amazon Elastic Kubernetes Service > Clusters > eventsphere-cluster
Overview
Resources
Compute
Networking
Add-ons
Capabilities
Access
Observability
Update history & backup
Resource types
Workloads
PodTemplates
Pods
ReplicaSets
Deployments
StatefulSets
DaemonSets
Jobs
CronJobs
PriorityClasses
HorizontalPodAutoscalers
ResourceClaims
ResourceClaimTemplates
Workloads: Pods (53)
View details
Pod is the smallest and simplest Kubernetes object. A Pod represents a set of running containers on your cluster. Learn more
All Namespaces
Filter Pods by name
1 2 3 4 5 6
Name
Created
Status
alertmanager-prometheus-kube-prometheus-alertmanager-0
16 hours ago
Running
auth-service-5986b7f4b8-5gg4t
16 hours ago
Running
auth-service-5986b7f4b8-n6mwk
16 hours ago
Running
aws-load-balancer-controller-6f47dff589-52wfs
17 hours ago
Running
aws-load-balancer-controller-6f47dff589-ztl55
17 hours ago
Running
aws-node-4fhvd
17 hours ago
Running

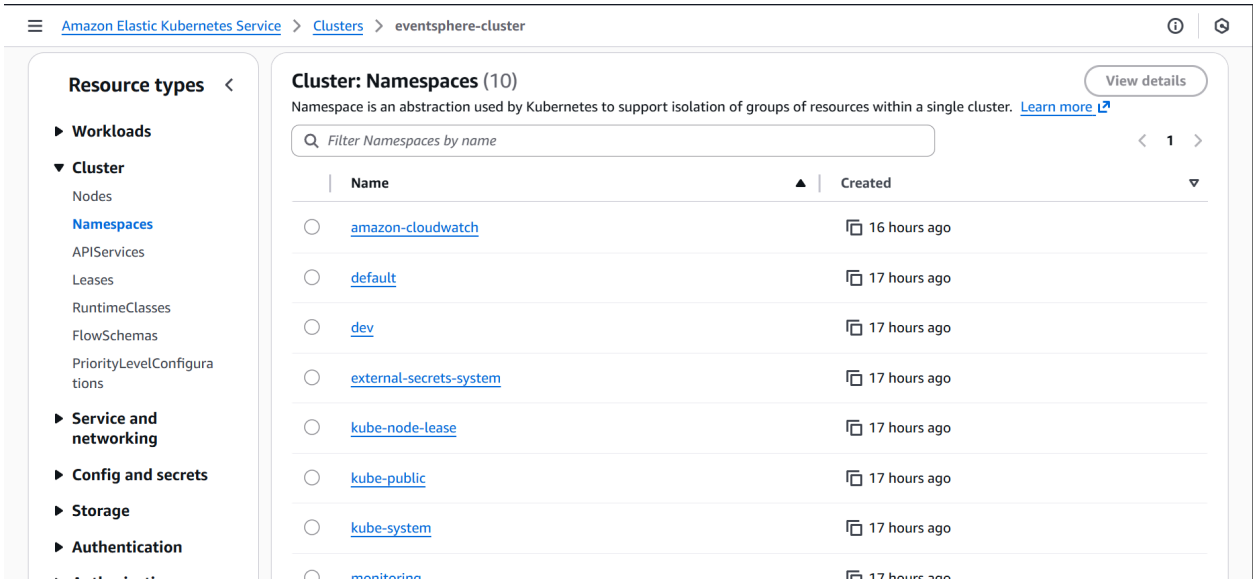

Amazon Elastic Kubernetes Service
Clusters
eventsphere-cluster
Resource types
Workloads
PodTemplates
Pods
ReplicaSets
Deployments
StatefulSets
DaemonSets
Jobs
CronJobs
PriorityClasses
HorizontalPodAutoscalers
ResourceClaims
ResourceClaimTemplates
ResourceSlices
Workloads: ReplicaSets (26)
View details
ReplicaSet aims to maintain a set of replica Pods running at any given time. Learn more
All Namespaces
Filter ReplicaSets by name
1 2 3
Name
Namespace
Type
Created
Pod count
Status
auth-service-54b59c4d5c
prod
replicasets
17 hours ago
0
0 Ready | 0 Failed | 0 Desired
auth-service-5986b7f4b8
prod
replicasets
16 hours ago
2
2 Ready | 0 Failed | 2 Desired
aws-load-balancer-controller-6f47dff589
kube-system
replicasets
17 hours ago
2
2 Ready | 0 Failed | 2 Desired
booking-service-7c6d6c87f7
prod
replicasets
16 hours ago
2
2 Ready | 0 Failed | 2 Desired


Amazon Elastic Kubernetes Service
Clusters
eventsphere-cluster
Resource types
Workloads
PodTemplates
Pods
ReplicaSets
Deployments
StatefulSets
DaemonSets
Jobs
CronJobs
PriorityClasses
HorizontalPodAutoscalers
ResourceClaims
ResourceClaimTemplates
ResourceSlices
Workloads: Deployments (16)
View details
Deployment is an API object that manages a replicated application, typically by running Pods with no local state. Learn more
All Namespaces
Filter Deployments by name
1 2
Name
Namespace
Type
Created
Pod count
Status
auth-service
prod
deployments
17 hours ago
2
2 Ready | 0 Failed | 2 Desired
aws-load-balancer-controller
kube-system
deployments
17 hours ago
2
2 Ready | 0 Failed | 2 Desired
booking-service
prod
deployments
17 hours ago
2
2 Ready | 0 Failed | 2 Desired
cluster-autoscaler
kube-system
deployments
17 hours ago
1
1 Ready | 0 Failed | 1 Desired


Amazon Elastic Kubernetes Service
Clusters
eventsphere-cluster
Resource types
Workloads
PodTemplates
Pods
ReplicaSets
Deployments
StatefulSets
DaemonSets
Jobs
CronJobs
PriorityClasses
HorizontalPodAutoscalers
ResourceClaims
ResourceClaimTemplates
ResourceSlices
Cluster
Workloads: DaemonSets (6)
View details
Daemonset ensures a copy of a Pod is running across a set of nodes in a cluster. Learn more
All Namespaces
Filter DaemonSets by name
1
Name
Namespace
Type
Created
Pod count
Status
aws-node
kube-system
daemonsets
17 hours ago
5
5 Ready | 0 Failed | 5 Desired
ebs-csi-node
kube-system
daemonsets
17 hours ago
5
5 Ready | 0 Failed | 5 Desired
ebs-csi-node-windows
kube-system
daemonsets
17 hours ago
0
0 Ready | 0 Failed | 0 Desired
fluent-bit
amazon-cloudwatch
daemonsets
16 hours ago
0
0 Ready | 0 Failed | 5 Desired


Amazon Elastic Kubernetes Service
Clusters
eventsphere-cluster
Resource types
Workloads
PodTemplates
Pods
ReplicaSets
Deployments
StatefulSets
DaemonSets
Jobs
CronJobs
PriorityClasses
HorizontalPodAutoscalers
ResourceClaims
ResourceClaimTemplates
ResourceSlices
Cluster
Workloads: HorizontalPodAutoscalers (5)
View details
The horizontal pod autoscaler automatically scales Pods depending on CPU demand. Learn more
All Namespaces
Filter HorizontalPodAutoscalers by name
1
Name
Created
auth-service-hpa
17 hours ago
booking-service-hpa
17 hours ago
event-service-hpa
17 hours ago
frontend-hpa
17 hours ago
notification-service-hpa
17 hours ago





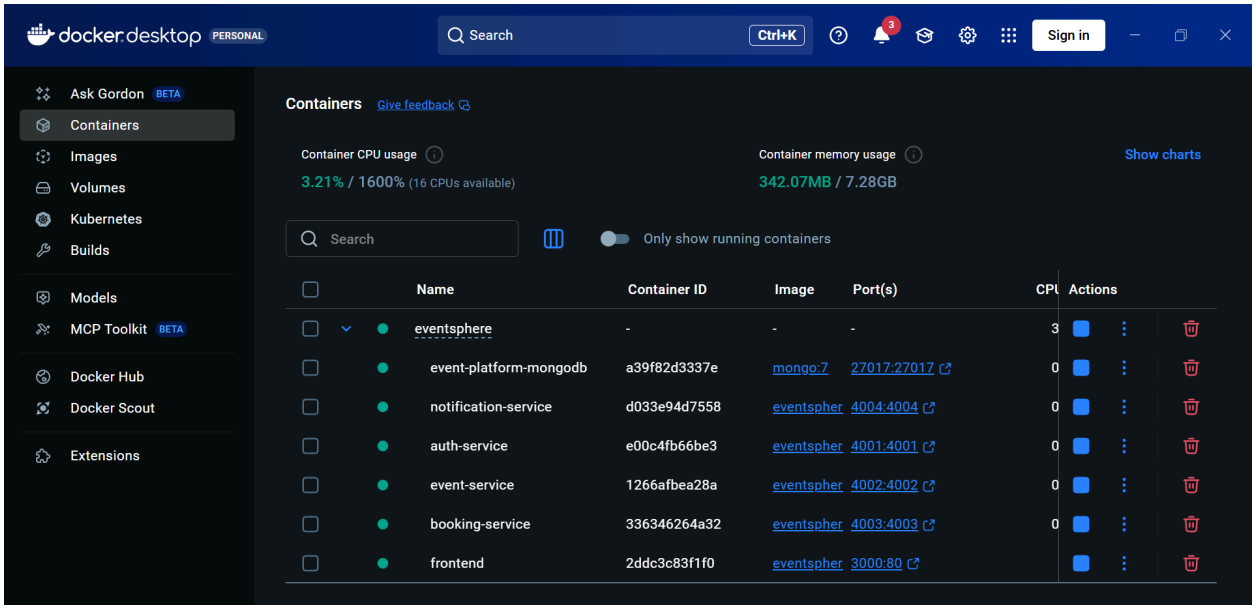
kubectl get pvc,pv outputs showing Bound state:

```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventSphere$ kubectl get pvc -n prod
NAME                     STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
mongodb-data-mongodb-0   Bound    pvc-4273ef6d-40aa-4766-839a-e2d128954985   20Gi       RWO            mongodb-ebs    <unset>                 2d4h
```

```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventSphere$ kubectl get pv -n prod
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                            STORAGECLASS   VOLUMEATTRIBUTESCLASS   REASON   AGE
pvc-4273ef6d-40aa-4766-839a-e2d128954985   20Gi       RWO            Retain           Bound    prod/mongodb-data-mongodb-0      mongodb-ebs    <unset>                          2d4h
```

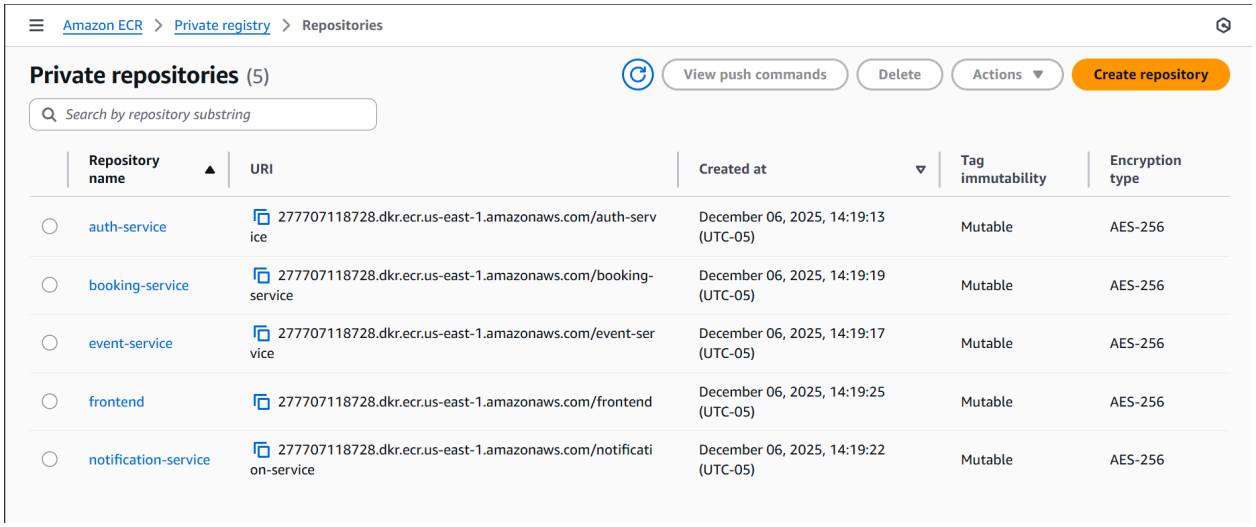
Design and Build Dockerized Microservices

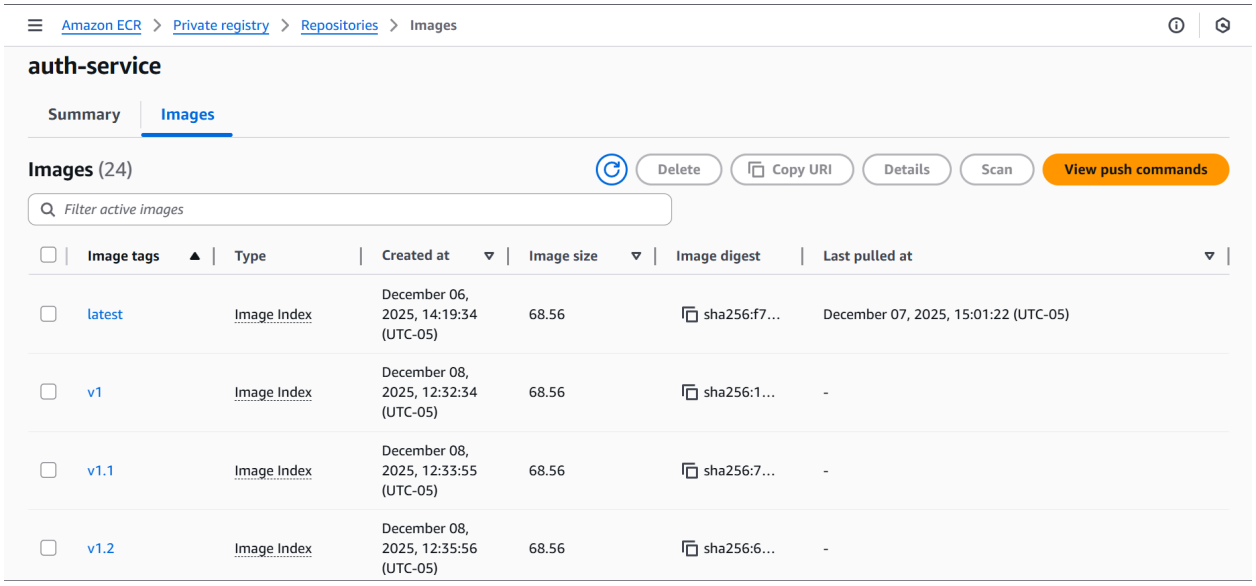
Screenshot of containers running locally:




```
 => => unpacking to docker.io/library/eventsphere-frontend:latest                                                0.0s
 => [notification-service] resolving provenance for metadata file                                                0.1s
 => [auth-service] resolving provenance for metadata file                                                        0.1s
 => [event-service] resolving provenance for metadata file                                                       0.1s
 => [booking-service] resolving provenance for metadata file                                                     0.1s
 => [frontend] resolving provenance for metadata file                                                            0.0s
[+] Running 13/13
 ✔eventsphere-frontend                       Built                                                              0.0s
 ✔eventsphere-notification-service           Built                                                              0.0s
 ✔eventsphere-auth-service                   Built                                                              0.0s
 ✔eventsphere-event-service                  Built                                                              0.0s
 ✔eventsphere-booking-service                Built                                                              0.0s
 ✔Network eventsphere_event-platform-network Created                                                            0.1s
 ✔Volume eventsphere_mongo-data              Created                                                            0.0s
 ✔Container event-platform-mongodb           Healthy                                                           13.4s
 ✔Container notification-service             Started                                                            0.8s
 ✔Container auth-service                     Started                                                           12.0s
 ✔Container event-service                    Started                                                           12.6s
 ✔Container booking-service                  Started                                                           13.3s
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventS
phere$ docker ps
CONTAINER ID   IMAGE                              COMMAND                  CREATED         STATUS                     PORTS
                                 NAMES
2ddc3c83f1f0   eventsphere-frontend               "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes (unhealthy)   0.0.0.
0:3000->80/tcp, [::]:3000->80/tcp                frontend
336346264a32   eventsphere-booking-service        "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes (healthy)     0.0.0.
0:4003->4003/tcp, [::]:4003->4003/tcp            booking-service
1266afbea28a   eventsphere-event-service          "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes (healthy)     0.0.0.
0:4002->4002/tcp, [::]:4002->4002/tcp            event-service
e00c4fb66be3   eventsphere-auth-service           "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes (healthy)     0.0.0.
0:4001->4001/tcp, [::]:4001->4001/tcp            auth-service
d033e94d7558   eventsphere-notification-service   "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes (healthy)     0.0.0.
0:4004->4004/tcp, [::]:4004->4004/tcp            notification-service
a39f82d3337e   mongo:7                            "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes (healthy)     0.0.0.
0:27017->27017/tcp, [::]:27017->27017/tcp        event-platform-mongodb
```

ECR listing with tagged versions:





Trivy scan reports (no Critical vulnerabilities):

```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventSphere$ ./bin/trivy image auth-service:latest
2025-12-08T13:30:22-05:00	INFO	[vuln] Vulnerability scanning is enabled
2025-12-08T13:30:22-05:00	INFO	[secret] Secret scanning is enabled
2025-12-08T13:30:22-05:00	INFO	[secret] If your scanning is slow, please try '--scanners vuln' to disable secret scanning
2025-12-08T13:30:22-05:00	INFO	[secret] Please see https://trivy.dev/docs/v0.68/guide/scanner/secret#recommendation for faster secret detection
2025-12-08T13:30:22-05:00	INFO	Detected OS	family="alpine" version="3.23.0"
2025-12-08T13:30:22-05:00	WARN	This OS version is not on the EOL list	family="alpine" version="3.23"
2025-12-08T13:30:22-05:00	INFO	[alpine] Detecting vulnerabilities...	os_version="3.23" repository="3.23" pkg_num=18
2025-12-08T13:30:22-05:00	INFO	Number of language-specific files	num=1
2025-12-08T13:30:22-05:00	INFO	[node-pkg] Detecting vulnerabilities...
2025-12-08T13:30:22-05:00	INFO	Table result includes only package filenames. Use '--format json' option to get the full path to the package file.

Report Summary
```

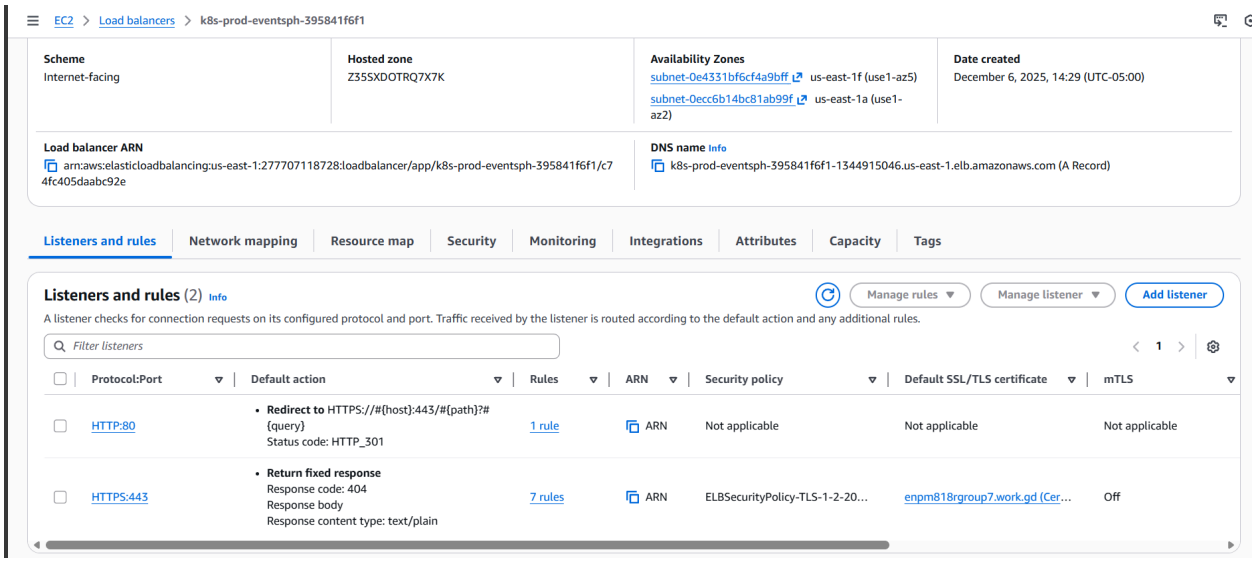
Target	Type	Vulnerabilities	Secrets
auth-service:latest (alpine 3.23.0)	alpine	0	-
app/node_modules/@isaacs/cliui/package.json	node-pkg	0	-
app/node_modules/@isaacs/fs-minipass/package.json	node-pkg	0	-
app/node_modules/@mongodb-js/saslprep/package.json	node-pkg	0	-
app/node_modules/@opentelemetry/api/package.json	node-pkg	0	-
app/node_modules/@pkgjs/parseargs/package.json	node-pkg	0	-
app/node_modules/@types/webidl-conversions/package.json	node-pkg	0	-
app/node_modules/@types/whatwg-url/package.json	node-pkg	0	-
app/node_modules/accepts/package.json	node-pkg	0	-
app/node_modules/ansi-regex/package.json	node-pkg	0	-
app/node_modules/ansi-styles/package.json	node-pkg	0	-
app/node_modules/asynckit/package.json	node-pkg	0	-
app/node_modules/axios/package.json	node-pkg	0	-
app/node_modules/balanced-match/package.json	node-pkg	0	-
usr/local/lib/node_modules/npm/node_modules/yallist/package.json	node-pkg	0	-
usr/local/lib/node_modules/npm/package.json	node-pkg	0	-

```
Legend:
- '-': Not scanned
- '0': Clean (no security findings detected)

Node.js (node-pkg)

Total: 3 (UNKNOWN: 0, LOW: 0, MEDIUM: 1, HIGH: 2, CRITICAL: 0)
```

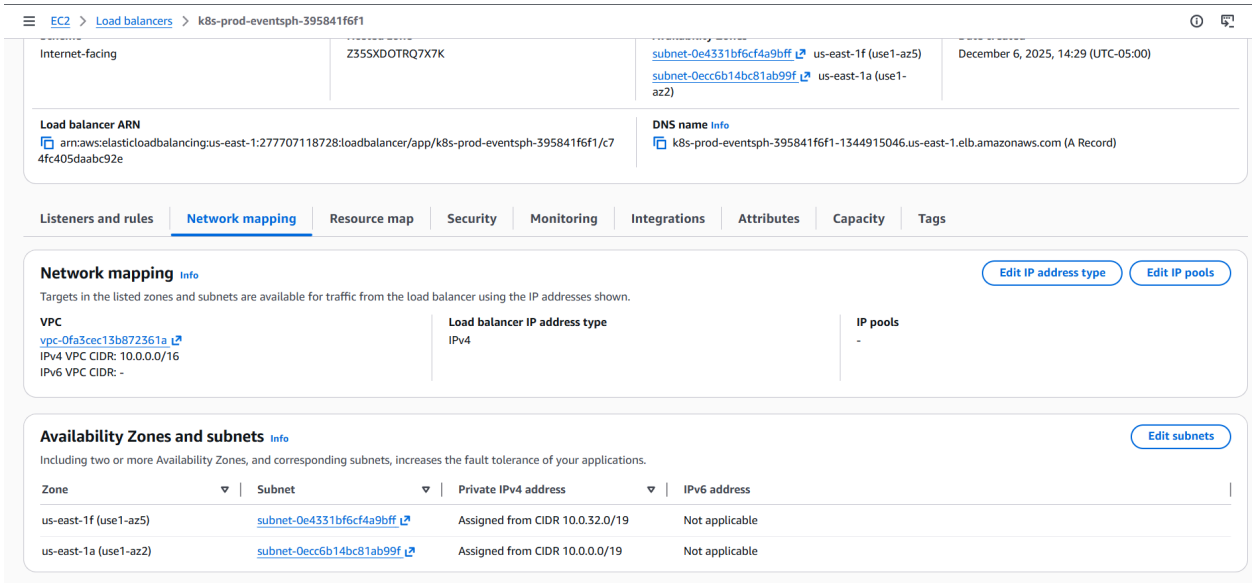
Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
glob (package.json)	CVE-2025-64756	HIGH	fixed	10.4.5	11.1.0, 10.5.0	glob: glob: Command Injection Vulnerability via Malicious Filenames https://avd.aquasec.com/nvd/cve-2025-64756
				11.0.3		
tar (package.json)	CVE-2025-64118	MEDIUM		7.5.1	7.5.2	node-tar has a race condition leading to uninitialized memory exposure https://avd.aquasec.com/nvd/cve-2025-64118

Load Balancing & Networking

ALB:



ALB Network Mapping:

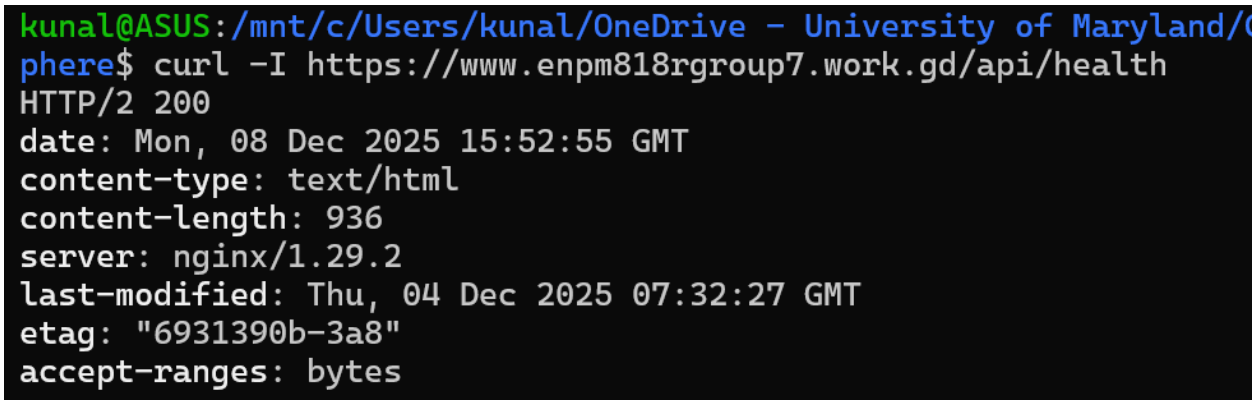


ALB Resource Map (target health OK):

Output of kubectl describe ingress:

```
kunal@ASUS:/mnt/c/Users/kunal/OneDrive - University of Maryland/Cyber/Sem 3/ENPM818R - Virtualization/Final project 2/EventS
phere$ kubectl describe ingress -n prod
Name:             eventsphere-ingress
Labels:           app=eventsphere
                  environment=production
Namespace:        prod
Address:          k8s-prod-eventsph-395841f6f1-1344915046.us-east-1.elb.amazonaws.com
Ingress Class:    alb
Default backend:  <default>
Rules:
  Host                        Path  Backends
  ----                        ----  --------
  www.enpm818rgroup7.work.gd
                              /   frontend:80 (10.0.4.135:80,10.0.57.206:80)
  api.enpm818rgroup7.work.gd
                              /api/auth            auth-service:4001 (10.0.55.26:4001,10.0.39.74:4001)
                              /api/events          event-service:4002 (10.0.23.181:4002,10.0.27.88:4002)
                              /api/bookings        booking-service:4003 (10.0.47.216:4003,10.0.57.233:4003)
                              /api/notifications   notification-service:4004 (10.0.54.188:4004,10.0.41.84:4004)
                              /health              auth-service:4001 (10.0.55.26:4001,10.0.39.74:4001)
Annotations:                  alb.ingress.kubernetes.io/certificate-arn: arn:aws:acm:us-east-1:277707118728:certificate/2ab5
f4c8-842c-447b-842f-b741fbdb2126
                              alb.ingress.kubernetes.io/healthcheck-interval-seconds: 30
                              alb.ingress.kubernetes.io/healthcheck-path: /health
                              alb.ingress.kubernetes.io/healthcheck-timeout-seconds: 5
                              alb.ingress.kubernetes.io/healthy-threshold-count: 2
                              alb.ingress.kubernetes.io/listen-ports: [{"HTTP": 80}, {"HTTPS": 443}]
                              alb.ingress.kubernetes.io/load-balancer-attributes: idle_timeout.timeout_seconds=60
                              alb.ingress.kubernetes.io/scheme: internet-facing
                              alb.ingress.kubernetes.io/ssl-policy: ELBSecurityPolicy-TLS-1-2-2017-01
                              alb.ingress.kubernetes.io/ssl-redirect: 443
                              alb.ingress.kubernetes.io/tags: Environment=production,Project=EventSphere
                              alb.ingress.kubernetes.io/target-type: ip
                              alb.ingress.kubernetes.io/unhealthy-threshold-count: 3
Events:                       <none>
```

Curl test results of API health endpoint confirm that the ALB is responding:



CI/CD Pipeline and Version Control

To manage the development across our team, we set up a dedicated GitHub Organization to centralize our code repositories and infrastructure files. We enforced a strict workflow where no developer was allowed to push code directly to the main branch.

Instead, every update, whether it was a fix for a microservice or a change to a Helm chart, required a Pull Request (PR). This allowed us to review each other’s code for logic errors and potential security issues before integrating it.

We connected these PRs directly to our CI/CD pipeline. Whenever a pull request was created, our automated testing suite (GitHub Actions) would run immediately. We configured the repository settings so that a PR could only be merged into main after all the pipeline tests passed successfully. This "quality gate" ensured that we never accidentally broke the production build with faulty code.

Observability

Observability Implementation

The system implements a complete observability stack using **Prometheus, Grafana, and AWS CloudWatch**, deployed in an **Amazon EKS cluster using Helm**.

Metrics and Visualization

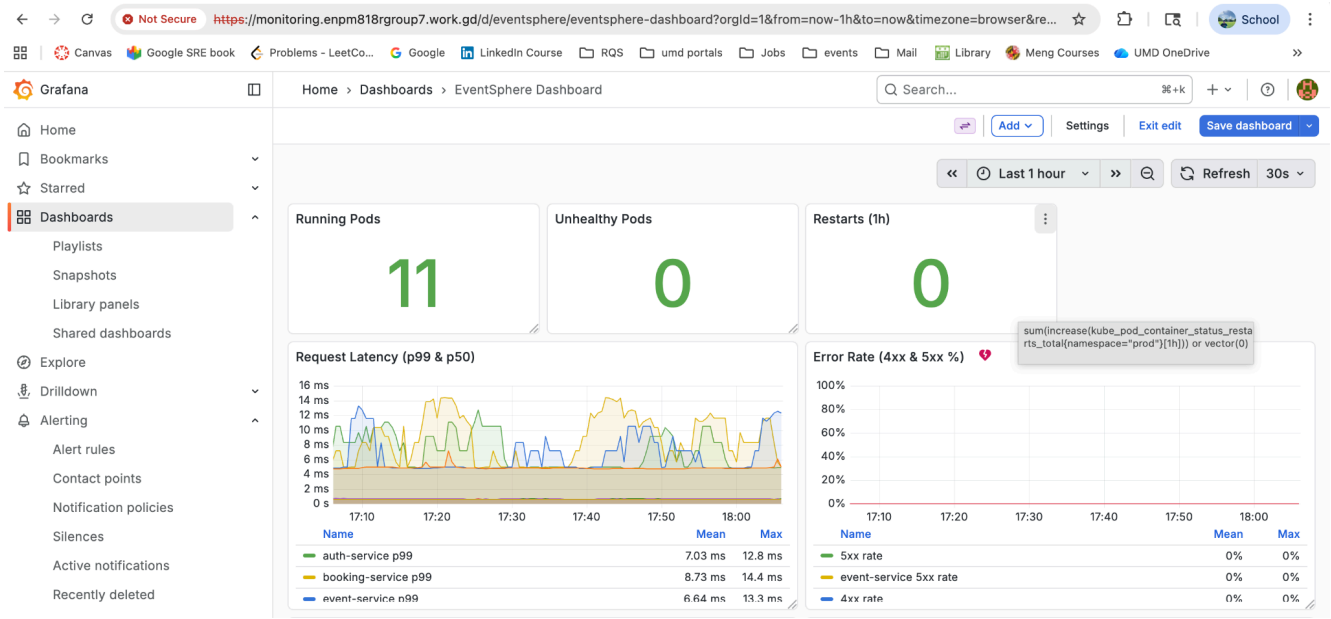
Prometheus scrapes Kubernetes and application-level metrics from all microservices running in the production namespace. Each service exposes a /metrics endpoint instrumented using Prometheus client libraries.

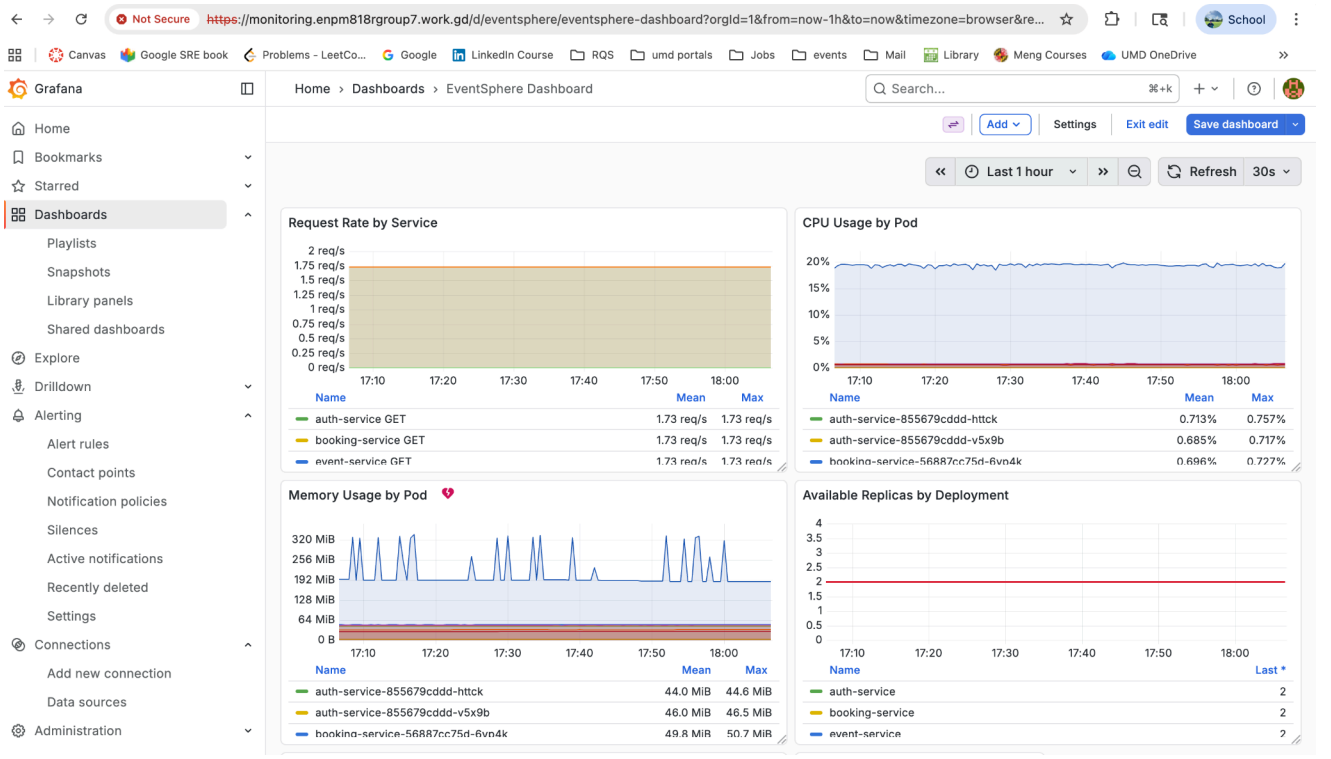
Grafana is used for metric visualization and alert evaluation through **Grafana Unified Alerting**. Unlike a port-forward–only setup, Grafana is exposed externally through an **AWS Application Load Balancer (ALB)** and accessed via DNS.

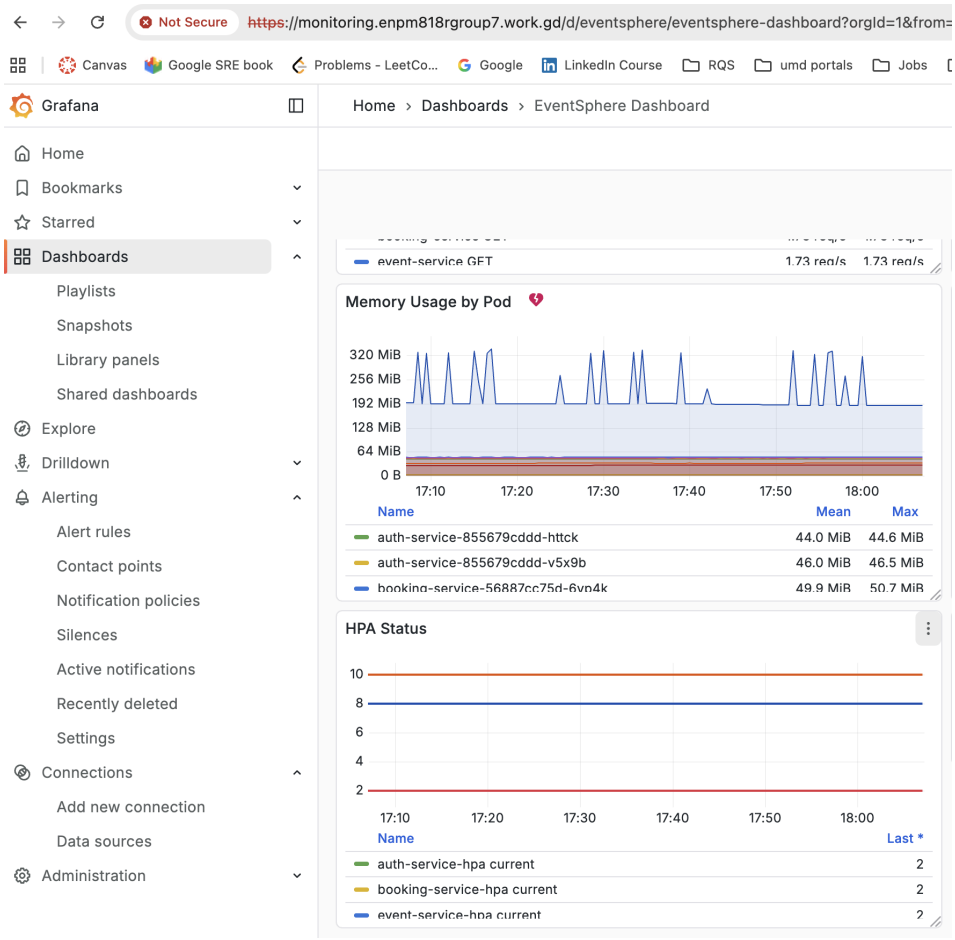
- **Grafana URL:** https://monitoring.enpm818rgroup7.work.gd/
- **Username:** admin
- **Password:** EventSphere2024
- **Dashboard:** *EventSphere Dashboard*

Grafana dashboards visualize:

1. Request latency (p99 and p50)
2. HTTP error rate (4xx & 5xx)
3. CPU and memory utilization by pod
4. Request rate per service
5. Pod status, restarts, and HPA state


Not Secure https://monitoring.enpm818rgroup7.work.gd/d/eventsphere/eventsphere-dashboard?orgId=1&from=now-1h&to=now&timezone=browser&re...
School
Canvas
Google SRE book
Problems - LeetCo...
Google
LinkedIn Course
RQS
umd portals
Jobs
events
Mail
Library
Meng Courses
UMD OneDrive
Grafana
Home › Dashboards › EventSphere Dashboard
Search...
⌘+k
Add
Settings
Exit edit
Save dashboard
Last 1 hour
Refresh
30s
Home
Bookmarks
Starred
Dashboards
Playlists
Snapshots
Library panels
Shared dashboards
Explore
Drilldown
Alerting
Alert rules
Contact points
Notification policies
Silences
Active notifications
Recently deleted
Running Pods
11
Unhealthy Pods
0
Restarts (1h)
0
sum(increase(kube_pod_container_status_restarts_total{namespace="prod"}[1h])) or vector(0)
Request Latency (p99 & p50)
16 ms
14 ms
12 ms
10 ms
8 ms
6 ms
4 ms
2 ms
0 s
17:10
17:20
17:30
17:40
17:50
18:00
Name
Mean
Max
auth-service p99
7.03 ms
12.8 ms
booking-service p99
8.73 ms
14.4 ms
event-service p99
6.64 ms
13.3 ms
Error Rate (4xx & 5xx %)
100%
80%
60%
40%
20%
0%
5xx rate
0%
0%
event-service 5xx rate
0%
0%
4xx rate
0%
0%


Not Secure https://monitoring.enpm818rgroup7.work.gd/d/eventsphere/eventsphere-dashboard?orgId=1&from=now-1h&to=now&timezone=browser&re...
School
Canvas
Google SRE book
Problems - LeetCo...
Google
LinkedIn Course
RQS
umd portals
Jobs
events
Mail
Library
Meng Courses
UMD OneDrive
Grafana
Home › Dashboards › EventSphere Dashboard
Search...
⌘+k
Add
Settings
Exit edit
Save dashboard
Last 1 hour
Refresh
30s
Home
Bookmarks
Starred
Dashboards
Playlists
Snapshots
Library panels
Shared dashboards
Explore
Drilldown
Alerting
Alert rules
Contact points
Notification policies
Silences
Active notifications
Recently deleted
Settings
Connections
Add new connection
Data sources
Administration
Request Rate by Service
2 req/s
1.75 req/s
1.5 req/s
1.25 req/s
1 req/s
0.75 req/s
0.5 req/s
0.25 req/s
0 req/s
17:10
17:20
17:30
17:40
17:50
18:00
Name
Mean
Max
auth-service GET
1.73 req/s
1.73 req/s
booking-service GET
1.73 req/s
1.73 req/s
event-service GET
1.73 req/s
1.73 req/s
CPU Usage by Pod
20%
15%
10%
5%
0%
Name
Mean
Max
auth-service-855679cddd-httck
0.713%
0.757%
auth-service-855679cddd-v5x9b
0.685%
0.717%
booking-service-56887cc75d-6vn4k
0.696%
0.727%
Memory Usage by Pod
320 MiB
256 MiB
192 MiB
128 MiB
64 MiB
0 B
Name
Mean
Max
auth-service-855679cddd-httck
44.0 MiB
44.6 MiB
auth-service-855679cddd-v5x9b
46.0 MiB
46.5 MiB
booking-service-56887cc75d-6vn4k
49.8 MiB
50.7 MiB
Available Replicas by Deployment
4
3.5
3
2.5
2
1.5
1
0.5
0
Name
Last *
auth-service
2
booking-service
2
event-service
2

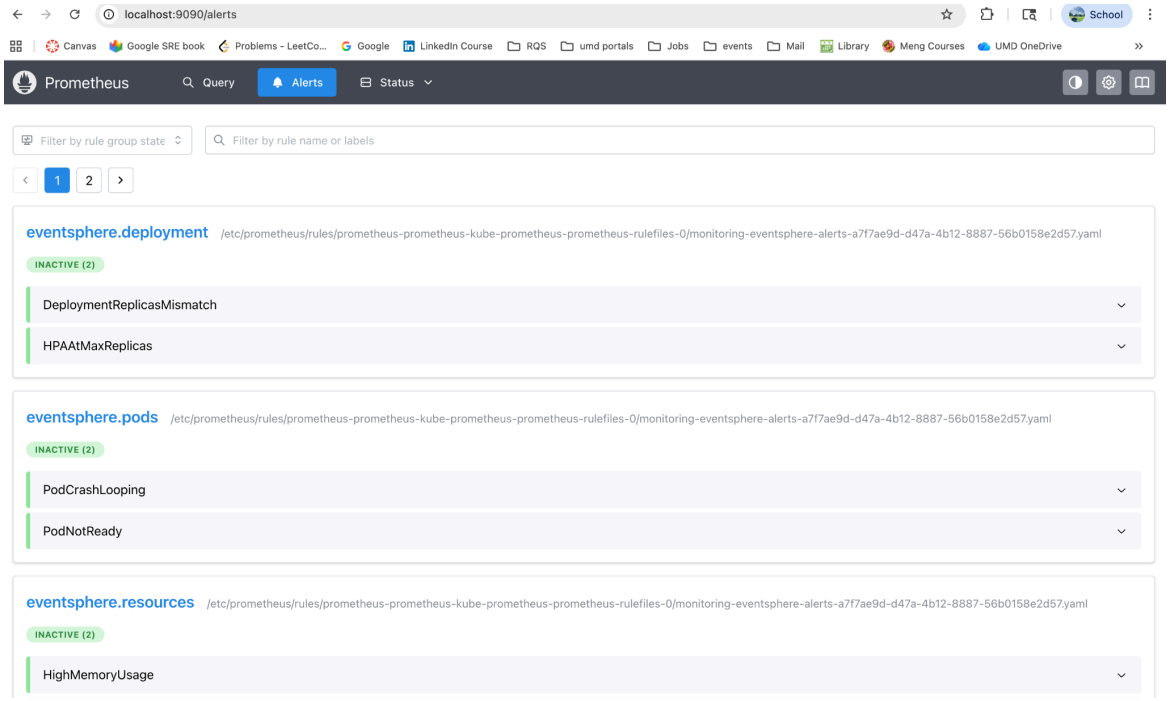


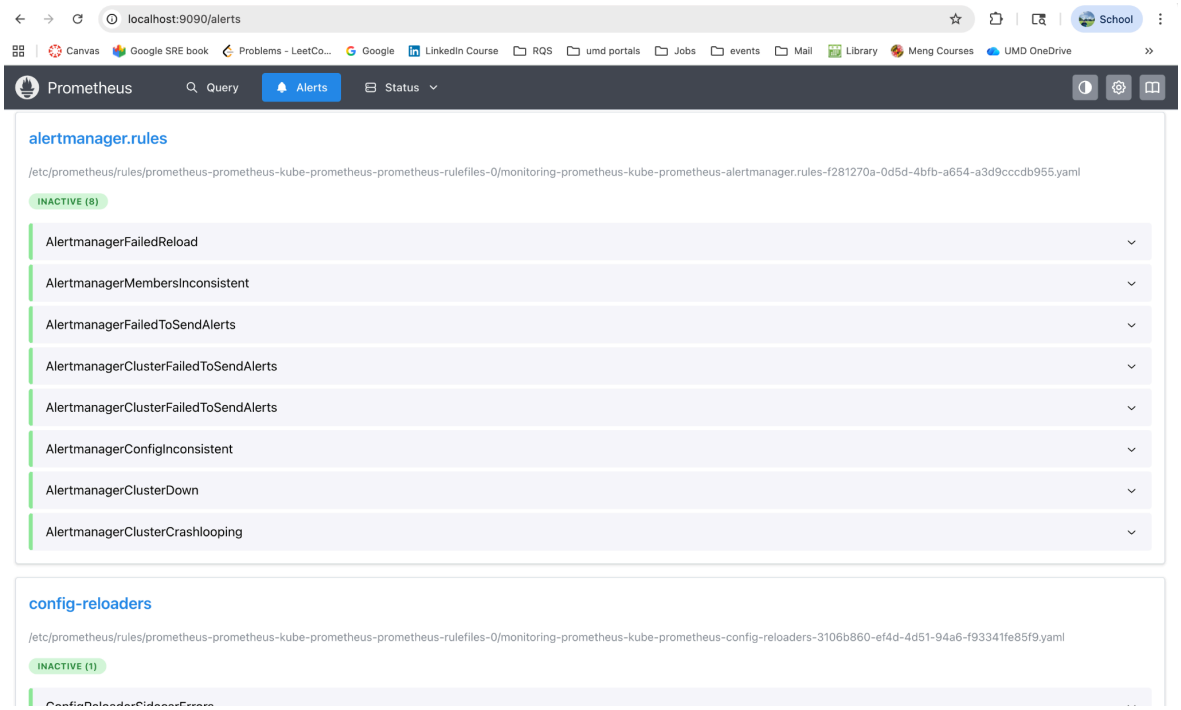
Prometheus Access

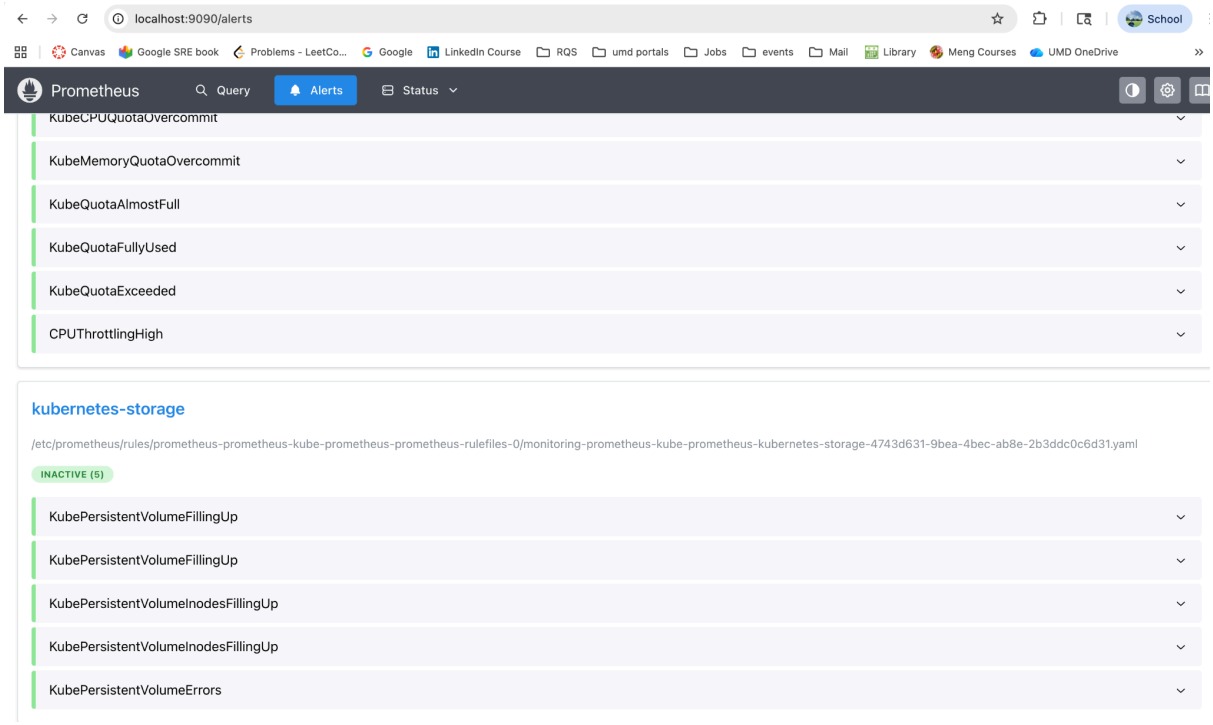
The Prometheus server itself is not publicly exposed and is accessed securely using port forwarding when needed for inspection and validation.

kubectl port-forward -n monitoring svc/prometheus-kube-prometheus-prometheus 9090:9090
Prometheus UI:

http://localhost:9090/alerts


localhost:9090/alerts
School
Canvas
Google SRE book
Problems - LeetCo...
Google
LinkedIn Course
RQS
umd portals
Jobs
events
Mail
Library
Meng Courses
UMD OneDrive
Prometheus
Query
Alerts
Status
Filter by rule group state
Filter by rule name or labels
1
2
eventsphere.deployment
/etc/prometheus/rules/prometheus-prometheus-kube-prometheus-prometheus-rulefiles-0/monitoring-eventsphere-alerts-a7f7ae9d-d47a-4b12-8887-56b0158e2d57.yaml
INACTIVE (2)
DeploymentReplicasMismatch
HPAAtMaxReplicas
eventspher.pods
/etc/prometheus/rules/prometheus-prometheus-kube-prometheus-prometheus-rulefiles-0/monitoring-eventsphere-alerts-a7f7ae9d-d47a-4b12-8887-56b0158e2d57.yaml
INACTIVE (2)
PodCrashLooping
PodNotReady
eventsphere.resources
/etc/prometheus/rules/prometheus-prometheus-kube-prometheus-prometheus-rulefiles-0/monitoring-eventsphere-alerts-a7f7ae9d-d47a-4b12-8887-56b0158e2d57.yaml
INACTIVE (2)
HighMemoryUsage


localhost:9090/alerts
School
Canvas
Google SRE book
Problems - LeetCo...
Google
LinkedIn Course
RQS
umd portals
Jobs
events
Mail
Library
Meng Courses
UMD OneDrive
Prometheus
Query
Alerts
Status
alertmanager.rules
/etc/prometheus/rules/prometheus-prometheus-kube-prometheus-prometheus-rulefiles-0/monitoring-prometheus-kube-prometheus-alertmanager.rules-f281270a-0d5d-4bfb-a654-a3d9cccdb955.yaml
INACTIVE (8)
AlertmanagerFailedReload
AlertmanagerMembersInconsistent
AlertmanagerFailedToSendAlerts
AlertmanagerClusterFailedToSendAlerts
AlertmanagerClusterFailedToSendAlerts
AlertmanagerConfigInconsistent
AlertmanagerClusterDown
AlertmanagerClusterCrashlooping
config-reloaders
/etc/prometheus/rules/prometheus-prometheus-kube-prometheus-prometheus-rulefiles-0/monitoring-prometheus-kube-prometheus-config-reloaders-3106b860-ef4d-4d51-94a6-f93341fe85f9.yaml
INACTIVE (1)



Logging Pipeline

CloudWatch is used for centralized log ingestion and log analytics.

Application logs, Kubernetes events, and alerting-related logs are ingested into **CloudWatch** and stored in **structured JSON format**, enabling filtering and correlation by pod, service, and timestamp.

Role of CloudWatch in the Observability Stack

CloudWatch is used to:

- Store and query **application and Kubernetes control-plane logs**
- Enable **on-demand log analysis** using CloudWatch Logs and Logs Insights
- Provide visibility into Kubernetes audit logs and request-level events

This complements, but does not duplicate, the Prometheus–Grafana metrics pipeline.

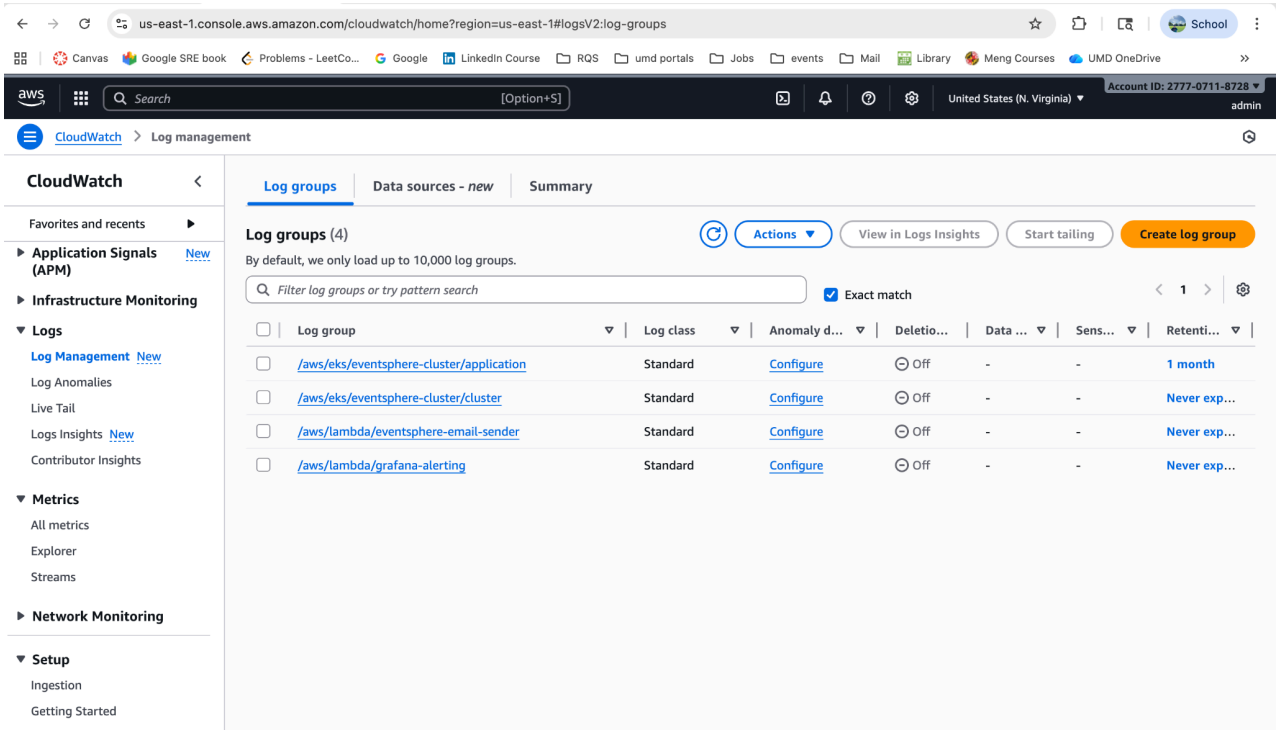
Relevant log groups include:

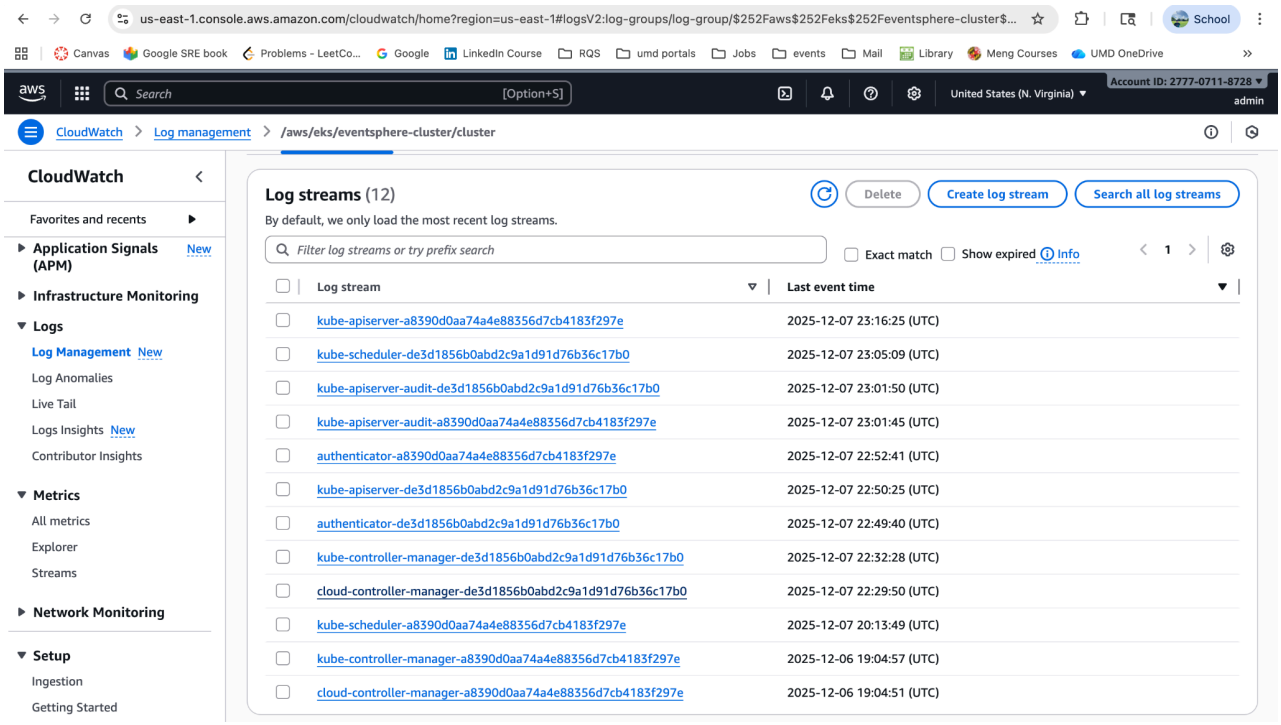
- /aws/eks/eventsphere-cluster/cluster
- /aws/lambda/eventsphere-email-sender
- /aws/lambda/grafana-alerting

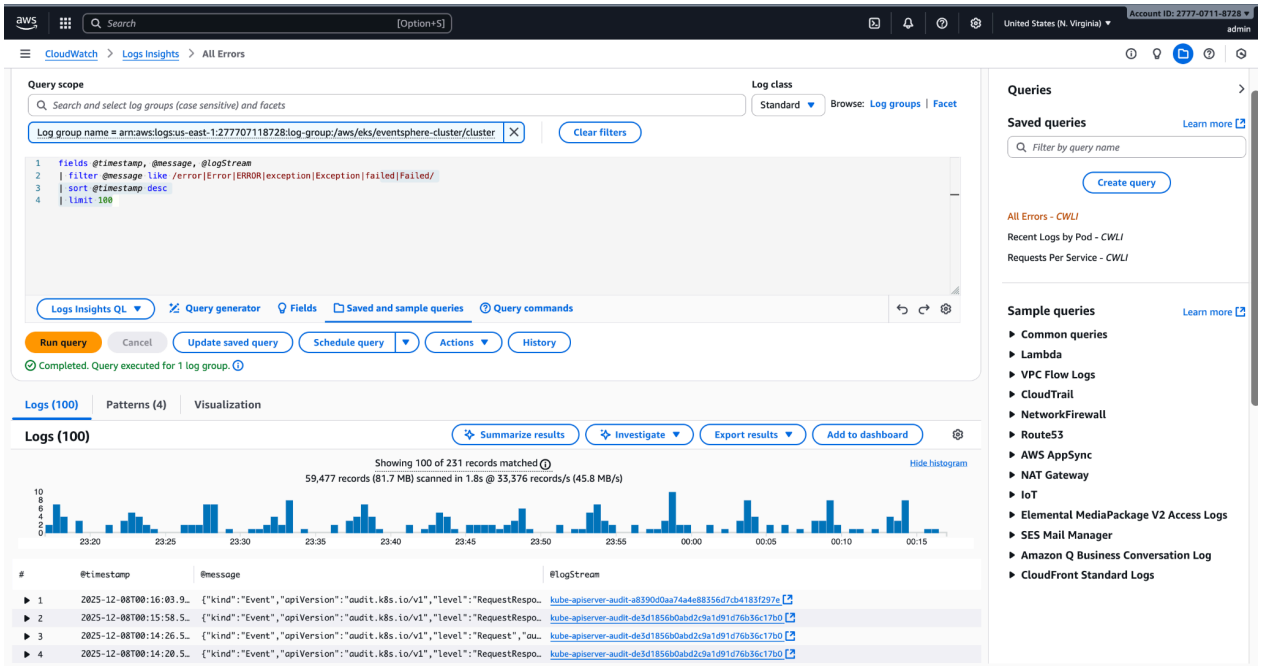
In this architecture:

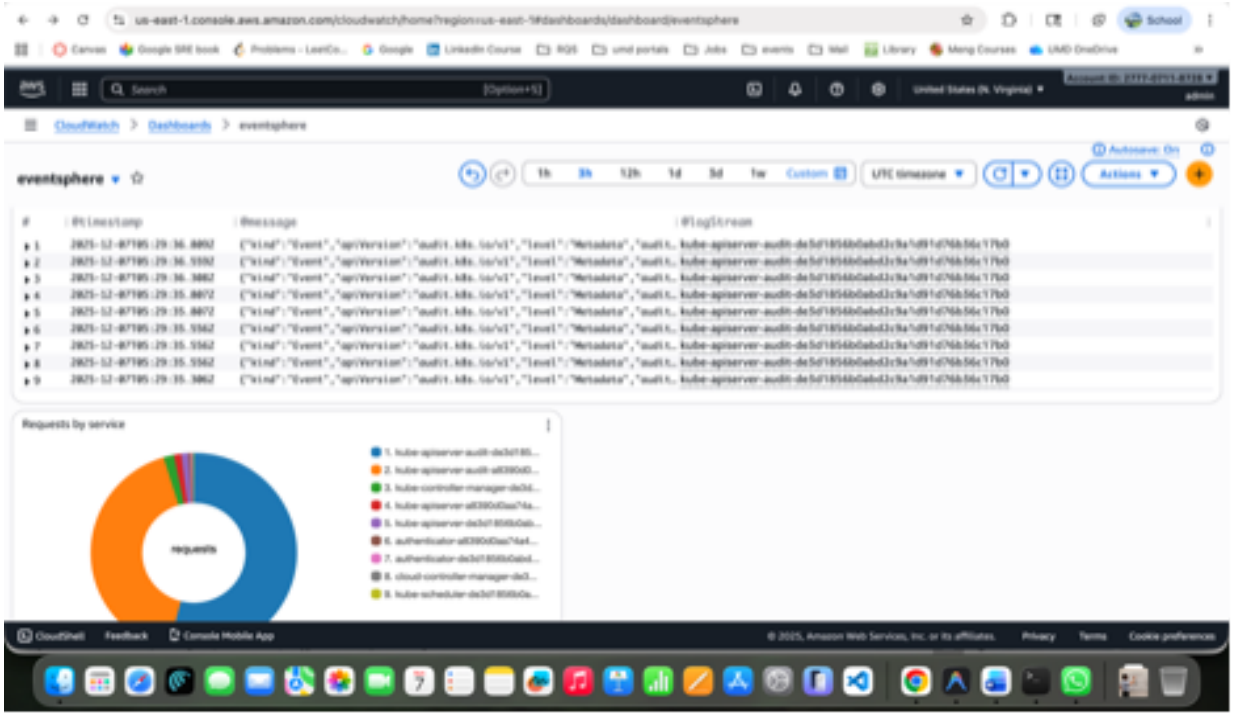
- **Metrics and alerting** are handled by Prometheus and Grafana
- **Logs and log analytics** are handled by CloudWatch

This separation follows recommended cloud-native observability practices.





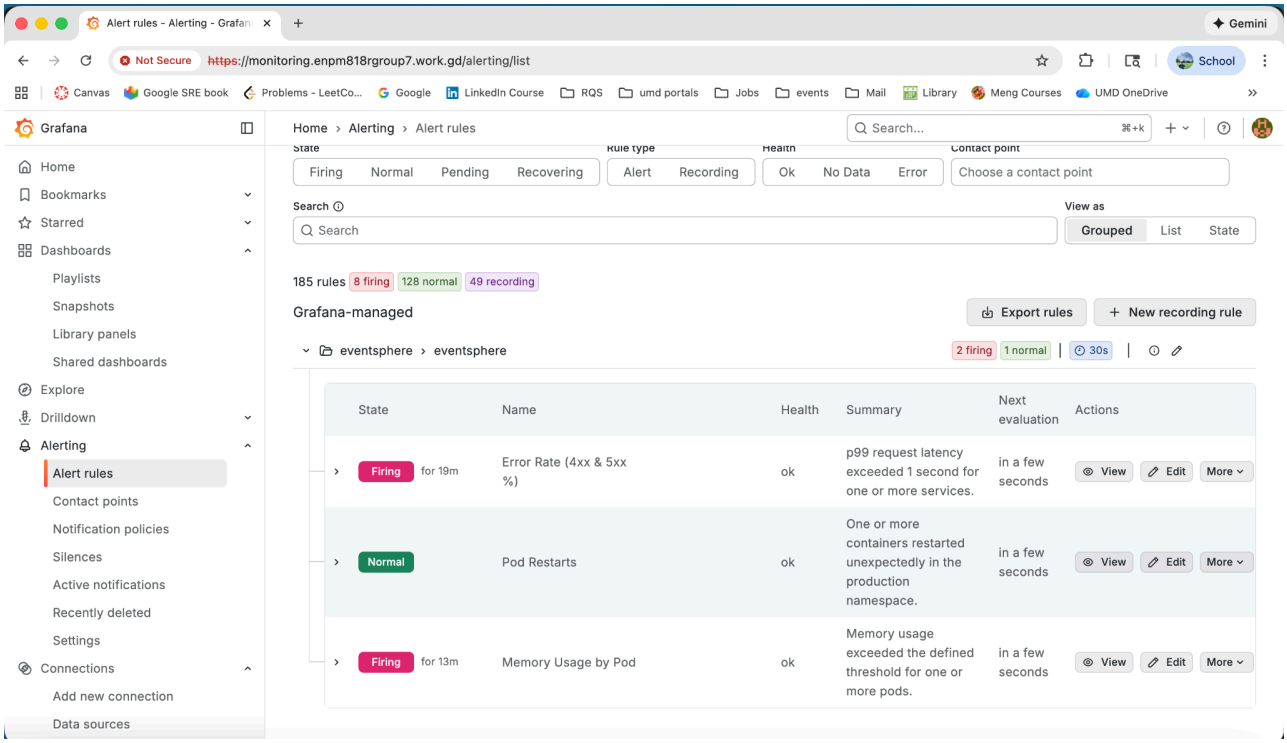

CloudWatch > Logs Insights > All Errors
Query scope
Search and select log groups (case sensitive) and facets
Log group name = arn:aws:logs:us-east-1:277707118728:log-group:/aws/eks/eventsphere-cluster/cluster
Clear filters
Log class
Standard
Browse: Log groups | Facet
fields @timestamp, @message, @logStream
| filter @message like /error|Error|ERROR|exception|Exception|failed|Failed/
| sort @timestamp desc
| limit 100
Logs Insights QL
Query generator
Fields
Saved and sample queries
Query commands
Run query
Cancel
Update saved query
Schedule query
Actions
History
Completed. Query executed for 1 log group.
Logs (100)
Patterns (4)
Visualization
Logs (100)
Summarize results
Investigate
Export results
Add to dashboard
Showing 100 of 231 records matched
59,477 records (81.7 MB) scanned in 1.8s @ 33,376 records/s (45.8 MB/s)
Hide histogram
Queries
Saved queries
Learn more
Filter by query name
Create query
All Errors - CWLI
Recent Logs by Pod - CWLI
Requests Per Service - CWLI
Sample queries
Learn more
Common queries
Lambda
VPC Flow Logs
CloudTrail
NetworkFirewall
Route53
AWS AppSync
NAT Gateway
IoT
Elemental MediaPackage V2 Access Logs
SES Mail Manager
Amazon Q Business Conversation Log
CloudFront Standard Logs


CloudWatch > Dashboards > eventsphere
eventsphere
Actions
Requests by service
requests

Alerting and Notifications

Alert rules are evaluated using **Grafana Unified Alerting**, and notifications are delivered through an AWS-native pipeline:

Grafana → Webhook → AWS Lambda → Amazon SNS → Email



Alerting and Runbooks Implementation

We implemented **three production-grade alert rules** using **Grafana Unified Alerting (Grafana-managed Alert manager)**. Alert notifications are delivered through an **AWS-integrated pipeline**, consisting of **Grafana webhook notifications routed through AWS Lambda and delivered via Amazon SNS (email)**.

The alerts are actively evaluated and observed in a **firing state**, as shown in the Grafana Alert Rules view.

Configured Alerts

1. **Error Rate (4xx & 5xx)**
    - Monitors elevated HTTP error rates across application services
    - Triggers when the error percentage exceeds the defined SLO threshold
2. **Memory Usage by Pod**
    - Tracks pod-level memory consumption using Kubernetes container metrics
    - Triggers when memory usage exceeds the configured threshold for a sustained period

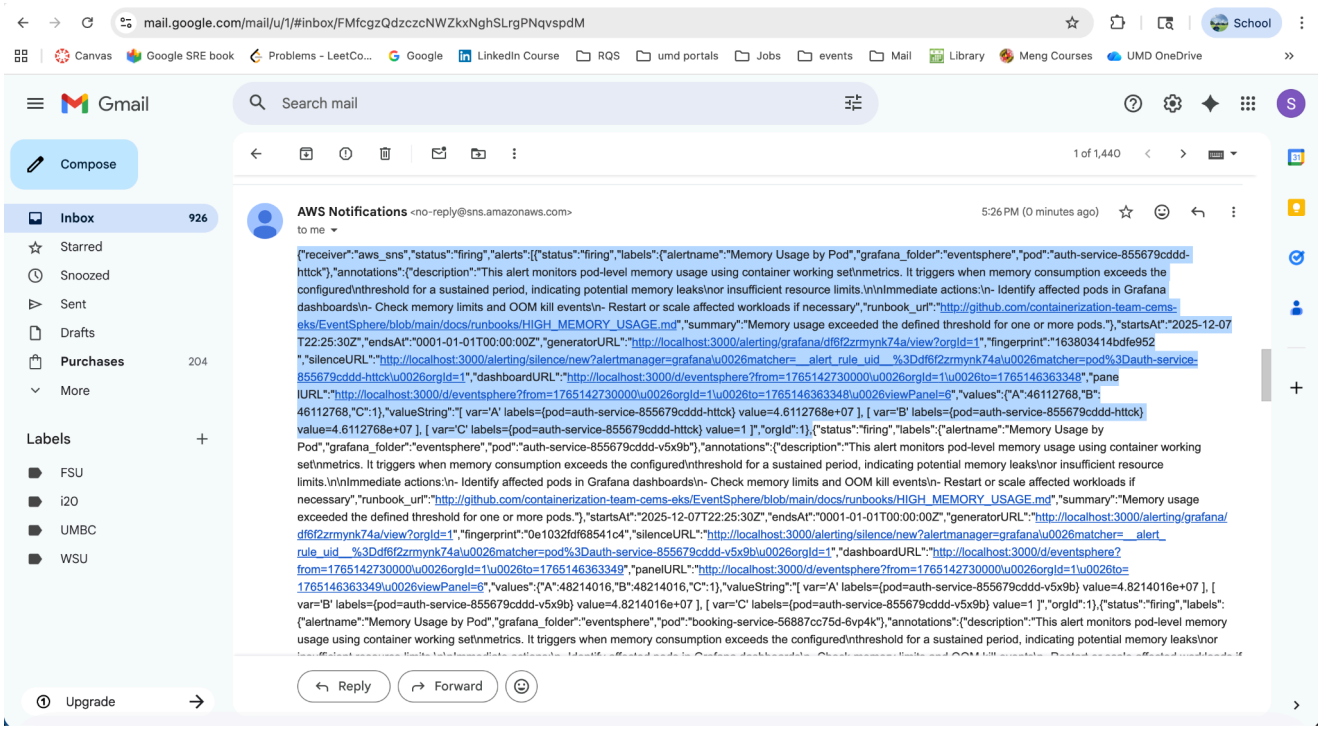
3. **Pod Restart Spike**
    - Detects unexpected container restarts within the production namespace
    - Indicates potential crash loops, failed deployments, or resource misconfigurations

All alert rules are evaluated by **Grafana Alert manager**, while notifications are delivered using **Amazon SNS**, invoked via an **AWS Lambda webhook integration**.

Runbooks

For each alert, a corresponding operational runbook has been created. Each runbook documents possible root causes, investigation steps, and remediation actions.

- **Error Rate Alert Runbook** https://github.com/containerization-team-cems-eks/EventSphere/blob/main/docs/runbooks/ERROR_RATE.md
- **Pod Restart Spike Runbook** https://github.com/containerization-team-cems-eks/EventSphere/blob/main/docs/runbooks/POD_RESTART_SPIKE.md
- **High Memory Usage Runbook** https://github.com/containerization-team-cems-eks/EventSphere/blob/main/docs/runbooks/HIGH_MEMORY_USAGE.md



This email is an automated **AWS SNS notification triggered by a Grafana alert** indicating **high memory usage at the pod level**. The alert fires when one or more Kubernetes pods

exceed the configured memory threshold for a sustained period, signaling potential memory leaks, misconfigured resource limits, or abnormal traffic patterns. Along with detailed metadata such as the affected pod, timestamps, and alert labels, the notification **includes links to Grafana dashboards and an attached runbook**, providing clear, predefined steps for investigation and remediation to reduce response time and operational guesswork.

Three Pillars of Observability

1. Metrics (Prometheus + Grafana)

Component	Description
prom-client middleware	Each service exposes /metrics with latency histograms and request counters
Prometheus	Scrapes metrics from all pods and stores time-series data
Grafana	Custom dashboard (eventsphere-dashboard.json) for visualization
Alertmanager	Evaluates alert rules and triggers notifications

Key metrics collected:

- http_request_duration_seconds – latency histogram
- http_requests_total – request count by method/route/status
- Default Node.js metrics (CPU, memory, event loop)

2. Logs (Fluent Bit → CloudWatch)

Component	Description
Fluent Bit	DaemonSet tailing /var/log/containers/*_prod_*.log
CloudWatch	Centralized log storage and search

3. Alerting

Alert rules are defined and managed using Helm configuration and evaluated by Grafana Alertmanager.

Configured Alerts:

Alert	Trigger	Severity

PodCrashLooping	Restart rate > 0 for 5m	Critical
PodNotReady	Pending/Failed > 10m	Warning
HighMemoryUsage	> 90% limit for 5m	Warning
HighCPUUsage	> 80% limit for 10m	Warning
DeploymentReplicasMismatch	Desired ≠ available for 10m	Warning
HPAAtMaxReplicas	Max replicas for 15m	Warning

Deployment Using Helm

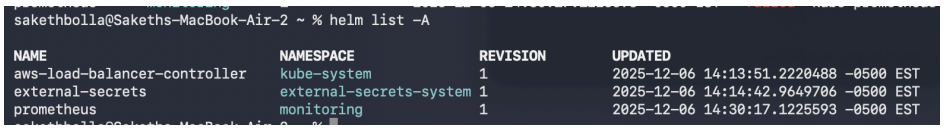
All observability components in the EventSphere platform are deployed to the Amazon EKS cluster using **Helm**, ensuring consistency, repeatability, and version-controlled configuration.

Helm charts are used to deploy and manage:

- Prometheus (metrics scraping and storage)
- Grafana (dashboards and alerting)
- Alert rules (Prometheus Rule and Grafana alert definitions)
- Fluent Bit (log forwarding to AWS CloudWatch)

Using Helm allows centralized configuration management, simplified upgrades, and reproducible deployments across environments.

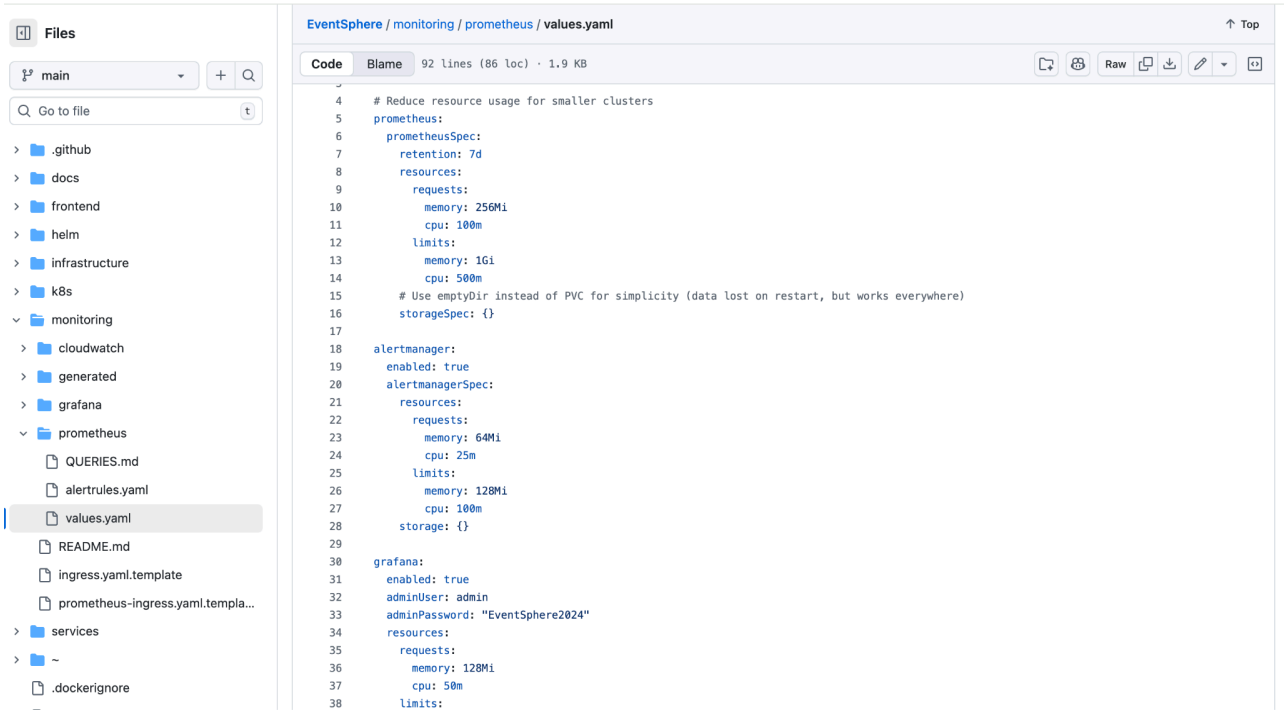
The Helm-based deployment confirms that the observability stack follows cloud-native and production-aligned deployment practices rather than ad-hoc manual configuration.



```
sakethbolla@Saketh-MacBook-Air-2 ~ % kubectl get pods -n monitoring

NAME                                                     READY   STATUS    RESTARTS   AGE
alertmanager-prometheus-kube-prometheus-alertmanager-0   2/2     Running   0          3h57m
prometheus-grafana-5965ff89c5-ld998                      3/3     Running   0          3h57m
prometheus-kube-prometheus-operator-6fc45cb74f-h7t6g     1/1     Running   0          3h57m
prometheus-kube-state-metrics-5d7bdb7c86-7fhbl           1/1     Running   0          3h57m
prometheus-prometheus-kube-prometheus-prometheus-0       2/2     Running   0          3h57m
prometheus-prometheus-node-exporter-4cjb6                1/1     Running   0          3h57m
prometheus-prometheus-node-exporter-lhxgf                1/1     Running   0          3h57m
prometheus-prometheus-node-exporter-svfbd                1/1     Running   0          3h56m
prometheus-prometheus-node-exporter-x4jbp                1/1     Running   0          17h
prometheus-prometheus-node-exporter-xj5zr                0/1     Pending   0          3h57m
```

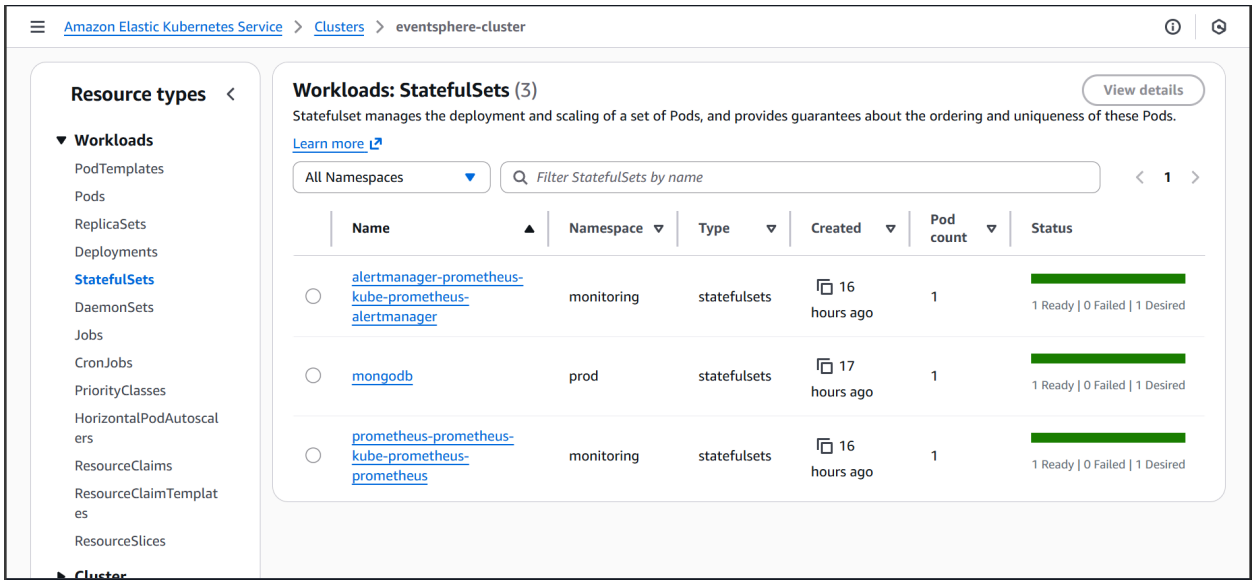
```
sakethbolla@Saketh-MacBook-Air-2 ~ % kubectl get svc -n monitoring

NAME                                      TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                      AGE
alertmanager-operated                     ClusterIP   None             <none>        9093/TCP,9094/TCP,9094/UDP   28h
prometheus-grafana                        ClusterIP   172.20.65.33     <none>        80/TCP                       28h
prometheus-kube-prometheus-alertmanager   ClusterIP   172.20.35.219    <none>        9093/TCP,8080/TCP            28h
prometheus-kube-prometheus-operator       ClusterIP   172.20.31.130    <none>        443/TCP                      28h
prometheus-kube-prometheus-prometheus     ClusterIP   172.20.51.12     <none>        9090/TCP,8080/TCP            28h
prometheus-kube-state-metrics             ClusterIP   172.20.235.236   <none>        8080/TCP                     28h
prometheus-operated                       ClusterIP   None             <none>        9090/TCP                     28h
prometheus-prometheus-node-exporter       ClusterIP   172.20.139.39    <none>        9100/TCP                     28h
sakethbolla@Saketh-MacBook-Air-2 ~ %
```

Deliverable	File Path
Prometheus/Grafana Helm Values	monitoring/prometheus/values.yaml
PrometheusRule (Alerts)	monitoring/prometheus/alertrules.yaml
Grafana Dashboard JSON	monitoring/grafana/dashboards/eventsphere-dashboard.json
Fluent Bit Config	monitoring/cloudwatch/fluent-bit-config.yaml



Deployment Methodology

All observability components, including **Prometheus, Grafana, Alerting rules, and Fluent Bit**, are deployed into the EKS cluster **using Helm charts**, ensuring reproducibility and consistency.


Amazon Elastic Kubernetes Service
Clusters
eventsphere-cluster
Resource types
Workloads
PodTemplates
Pods
ReplicaSets
Deployments
StatefulSets
DaemonSets
Jobs
CronJobs
PriorityClasses
HorizontalPodAutoscalers
ResourceClaims
ResourceClaimTemplates
ResourceSlices
Cluster
Workloads: StatefulSets (3)
View details
Statefulset manages the deployment and scaling of a set of Pods, and provides guarantees about the ordering and uniqueness of these Pods.
Learn more
All Namespaces
Filter StatefulSets by name
1
Name
Namespace
Type
Created
Pod count
Status
alertmanager-prometheus-kube-prometheus-alertmanager
monitoring
statefulsets
16 hours ago
1
1 Ready | 0 Failed | 1 Desired
mongodb
prod
statefulsets
17 hours ago
1
1 Ready | 0 Failed | 1 Desired
prometheus-prometheus-kube-prometheus-prometheus
monitoring
statefulsets
16 hours ago
1
1 Ready | 0 Failed | 1 Desired

Security and Compliance

All Images are taken from trusted sources. Images are scanned for vulnerabilities with trivy, and rejected if there are critical vulnerabilities. Database is encrypted, and TLS is enforced for all data in transit. Trivy Scan reports:

---

eventsphere-auth-service:latest (alpine 3.23.0) - Trivy Report - 2025-12-09 12:03:33.33226948 -0500 EST m=+0.794909274

alpine					
No Vulnerabilities found					
No Misconfigurations found					
node-pkg					
Package	**Vulnerability ID**	**Severity**	**Installed Version**	**Fixed Version**	**Links**
glob	CVE-2025-64756	HIGH	10.4.5	11.1.0, 10.5.0	https://access.redhat.com/security/cve/CVE-2025-64756 https://github.com/isaacs/node-glob https://github.com/isaacs/node-glob/commit/1e4e297342a09f2aa0ced87fcd4a70ddc325d75f Toggle more links
glob	CVE-2025-64756	HIGH	11.0.3	11.1.0, 10.5.0	https://access.redhat.com/security/cve/CVE-2025-64756 https://github.com/isaacs/node-glob https://github.com/isaacs/node-glob/commit/1e4e297342a09f2aa0ced87fcd4a70ddc325d75f Toggle more links
tar	CVE-2025-64118	MEDIUM	7.5.1	7.5.2	https://github.com/isaacs/node-tar https://github.com/isaacs/node-tar/commit/5330eb04bc43014f216e5c271b40d5c00d45224d https://github.com/isaacs/node-tar/commit/5e1a8e638600d3c3a2969b4de6a6ec44fa8d74c9 Toggle more links
No Misconfigurations found					

eventsphere-booking-service:latest (alpine 3.23.0) - Trivy Report - 2025-12-09 12:08:07.252320747 -0500 EST m=+1.756425588

alpine					
No Vulnerabilities found					
No Misconfigurations found					
node-pkg					
Package	**Vulnerability ID**	**Severity**	**Installed Version**	**Fixed Version**	**Links**
glob	CVE-2025-64756	HIGH	10.4.5	11.1.0, 10.5.0	https://access.redhat.com/security/cve/CVE-2025-64756 https://github.com/isaacs/node-glob https://github.com/isaacs/node-glob/commit/1e4e297342a09f2aa0ced87fcd4a70ddc325d75f Toggle more links
glob	CVE-2025-64756	HIGH	11.0.3	11.1.0, 10.5.0	https://access.redhat.com/security/cve/CVE-2025-64756 https://github.com/isaacs/node-glob https://github.com/isaacs/node-glob/commit/1e4e297342a09f2aa0ced87fcd4a70ddc325d75f Toggle more links
tar	CVE-2025-64118	MEDIUM	7.5.1	7.5.2	https://github.com/isaacs/node-tar https://github.com/isaacs/node-tar/commit/5330eb04bc43014f216e5c271b40d5c00d45224d https://github.com/isaacs/node-tar/commit/5e1a8e638600d3c3a2969b4de6a6ec44fa8d74c9 Toggle more links
No Misconfigurations found					

eventsphere-event-service:latest (alpine 3.23.0) - Trivy Report - 2025-12-09 12:08:01.892374016 -0500 EST m=+2.317336817

alpine					
No Vulnerabilities found					
No Misconfigurations found					
node-pkg					
Package	**Vulnerability ID**	**Severity**	**Installed Version**	**Fixed Version**	**Links**
glob	CVE-2025-64756	HIGH	10.4.5	11.1.0, 10.5.0	https://access.redhat.com/security/cve/CVE-2025-64756 https://github.com/isaacs/node-glob https://github.com/isaacs/node-glob/commit/1e4e297342a09f2aa0ced87fcd4a70ddc325d75f Toggle more links
glob	CVE-2025-64756	HIGH	11.0.3	11.1.0, 10.5.0	https://access.redhat.com/security/cve/CVE-2025-64756 https://github.com/isaacs/node-glob https://github.com/isaacs/node-glob/commit/1e4e297342a09f2aa0ced87fcd4a70ddc325d75f Toggle more links
tar	CVE-2025-64118	MEDIUM	7.5.1	7.5.2	https://github.com/isaacs/node-tar https://github.com/isaacs/node-tar/commit/5330eb04bc43014f216e5c271b40d5c00d45224d https://github.com/isaacs/node-tar/commit/5e1a8e638600d3c3a2969b4de6a6ec44fa8d74c9 Toggle more links
No Misconfigurations found					

eventsphere-notification-service:latest (alpine 3.23.0) - Trivy Report - 2025-12-09 12:03:50.955954529 -0500 EST m=+0.496271750

alpine					
No Vulnerabilities found					
No Misconfigurations found					
node-pkg					
Package	**Vulnerability ID**	**Severity**	**Installed Version**	**Fixed Version**	**Links**
glob	CVE-2025-64756	HIGH	10.4.5	11.1.0, 10.5.0	https://access.redhat.com/security/cve/CVE-2025-64756 https://github.com/isaacs/node-glob https://github.com/isaacs/node-glob/commit/1e4e297342a09f2aa0ced87fcd4a70ddc325d75f Toggle more links
glob	CVE-2025-64756	HIGH	11.0.3	11.1.0, 10.5.0	https://access.redhat.com/security/cve/CVE-2025-64756 https://github.com/isaacs/node-glob https://github.com/isaacs/node-glob/commit/1e4e297342a09f2aa0ced87fcd4a70ddc325d75f Toggle more links
tar	CVE-2025-64118	MEDIUM	7.5.1	7.5.2	https://github.com/isaacs/node-tar https://github.com/isaacs/node-tar/commit/5330eb04bc43014f216e5c271b40d5c00d45224d https://github.com/isaacs/node-tar/commit/5e1a8e638600d3c3a2969b4de6a6ec44fa8d74c9 Toggle more links
No Misconfigurations found					

eventsphere-frontend:latest (alpine 3.22.2) - Trivy Report - 2025-12-09 11:59:09.832655202 -0500 EST m=+0.854720586

alpine					
Package	**Vulnerability ID**	**Severity**	**Installed Version**	**Fixed Version**	**Links**
busybox	CVE-2024-58251	MEDIUM	1.37.0-r19	1.37.0-r20	http://www.openwall.com/lists/oss-security/2025/04/23/6 https://bugs.busybox.net/show_bug.cgi?id=15922 https://www.busybox.net Toggle more links
busybox	CVE-2025-46394	LOW	1.37.0-r19	1.37.0-r20	http://www.openwall.com/lists/oss-security/2025/04/23/5 http://www.openwall.com/lists/oss-security/2025/04/24/3 https://bugs.busybox.net/show_bug.cgi?id=16018 Toggle more links
busybox-binsh	CVE-2024-58251	MEDIUM	1.37.0-r19	1.37.0-r20	http://www.openwall.com/lists/oss-security/2025/04/23/6 https://bugs.busybox.net/show_bug.cgi?id=15922 https://www.busybox.net Toggle more links
busybox-binsh	CVE-2025-46394	LOW	1.37.0-r19	1.37.0-r20	http://www.openwall.com/lists/oss-security/2025/04/23/5 http://www.openwall.com/lists/oss-security/2025/04/24/3 https://bugs.busybox.net/show_bug.cgi?id=16018 Toggle more links
libpng	CVE-2025-64720	HIGH	1.6.47-r0	1.6.51-r0	https://access.redhat.com/security/cve/CVE-2025-64720 https://github.com/pnggroup/libpng/commit/08da33b4c88cfcd36e5a706558a8d7e0e4773643 https://github.com/pnggroup/libpng/issues/686 Toggle more links
libpng	CVE-2025-65018	HIGH	1.6.47-r0	1.6.51-r0	https://access.redhat.com/security/cve/CVE-2025-65018 https://github.com/pnggroup/libpng/commit/16b5e3823918840aae65c0a6da57c78a5a496a4d https://github.com/pnggroup/libpng/commit/218612ddd6b17944e21eda56caf8b4bf7779d1ea Toggle more links
libpng	CVE-2025-66293	HIGH	1.6.47-r0	1.6.53-r0	http://www.openwall.com/lists/oss-security/2025/12/03/6 http://www.openwall.com/lists/oss-security/2025/12/03/7 http://www.openwall.com/lists/oss-security/2025/12/03/8 Toggle more links
libpng	CVE-2025-64505	MEDIUM	1.6.47-r0	1.6.51-r0	https://access.redhat.com/security/cve/CVE-2025-64505 https://github.com/pnggroup/libpng/commit/6a528eb5fd0dd7f6de1c39d30de0e41473431c37 https://github.com/pnggroup/libpng/commit/6a528eb5fd0dd7f6de1c39d30de0e41473431c37 (v1.6.51) Toggle more links
libpng	CVE-2025-64506	MEDIUM	1.6.47-r0	1.6.51-r0	https://access.redhat.com/security/cve/CVE-2025-64506 https://github.com/pnggroup/libpng/commit/2bd84c019c300b78e811743fbcddb67c9d9bf821 https://github.com/pnggroup/libpng/pull/749 Toggle more links
ssl_client	CVE-2024-58251	MEDIUM	1.37.0-r19	1.37.0-r20	http://www.openwall.com/lists/oss-security/2025/04/23/6 https://bugs.busybox.net/show_bug.cgi?id=15922 https://www.busybox.net Toggle more links
ssl_client	CVE-2025-46394	LOW	1.37.0-r19	1.37.0-r20	http://www.openwall.com/lists/oss-security/2025/04/23/5 http://www.openwall.com/lists/oss-security/2025/04/24/3 https://bugs.busybox.net/show_bug.cgi?id=16018 Toggle more links
No Misconfigurations found					

Once we analyze the trivy reports, we mitigate each of the vulnerabilities found and do a report once again. If a fix isn't available, we can setup alerts and monitor those images closely with our monitoring stack, while waiting for the vendor to publish a fix.

Innovation

In developing EventSphere, our team moved beyond simple containerization to implement an operations-ready platform. The primary innovation lies in our automated resiliency and proactive security capabilities.

Dynamic Auto-Scaling instead of static provisioning: We implemented a dual-layer scaling strategy. We configured the Cluster Autoscaler to watch for pending pods and automatically provision new EC2 nodes when resources run low. We validated this by running a load test that pushed the cluster from 2 nodes up to 5 nodes automatically when traffic spiked, and then scaled it back down when the load was removed. We also implemented Horizontal Pod Autoscalers (HPA) for individual services like the auth-service and booking-service to scale pod replicas based on CPU demand.

DevSecOps Integration: We integrated security directly into our pipeline. We used Trivy to scan our container images for vulnerabilities before deployment. For example, our scans of the auth-service identified specific package vulnerabilities, allowing us to address risks before they hit production.

Production-Grade Alerting & Runbooks: Rather than just staring at dashboards, we set up an automated alerting pipeline. If error rates spike (4xx/5xx errors) or memory usage gets too high, Grafana triggers a webhook that routes through AWS Lambda to Amazon SNS, sending us an email notification immediately. Crucially, each alert is paired with a specific Runbook (hosted on GitHub) that tells the on-call engineer exactly how to diagnose the crash loop or memory leak, significantly reducing time-to-recovery.

Notification Service

1. Introduction:

The Notification Service is a cloud-native microservice responsible for delivering transactional emails—including welcome emails and booking confirmations—in the EventSphere system. It leverages Amazon SNS for message brokering, AWS Lambda for serverless processing, and Amazon SES for authenticated email delivery. The service runs on AWS EKS and uses IRSA (IAM Roles for Service Accounts) for secure AWS integration.

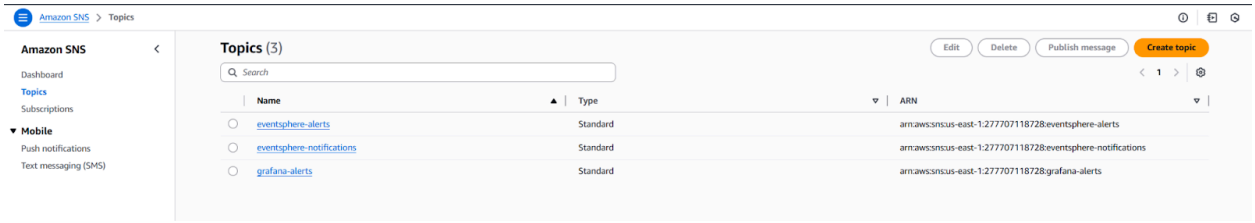
2. Architecture Overview

The notification pipeline follows an event-driven architecture:
User registers / books an event
Auth or Booking Service calls Notification Service REST endpoint
Notification Service publishes a structured message to SNS
SNS invokes Lambda
Lambda sends email via SES
SES delivers to end-user inbox

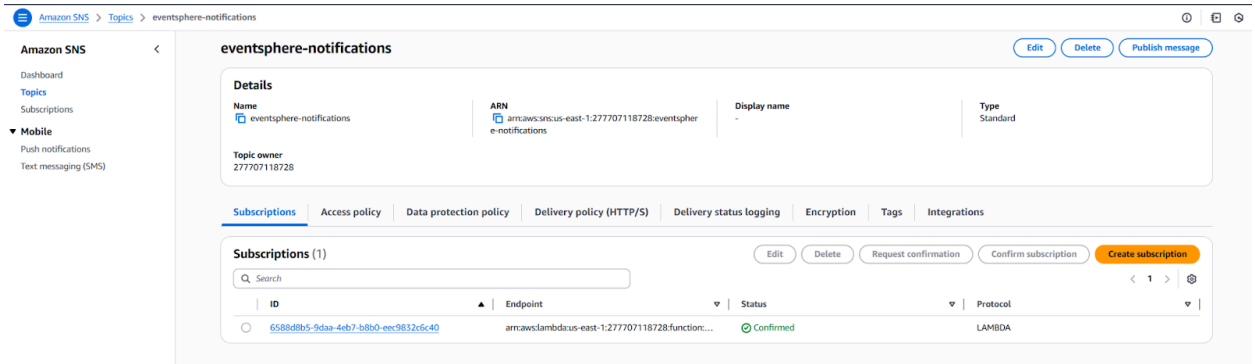
3. Technology Stack
Node.js + Express backend service
AWS SNS – asynchronous messaging
AWS Lambda – email processing
AWS SES – email delivery
Amazon EKS – container orchestration
Amazon ECR – container registry
Kubernetes HPA – auto-scaling
IAM + IRSA – identity and permissions

4.1 Notification Service (Microservice)
Handles REST API requests and publishes messages to SNS.
Endpoints:
POST /api/notifications/welcome
POST /api/notifications/booking-confirmation
GET /health

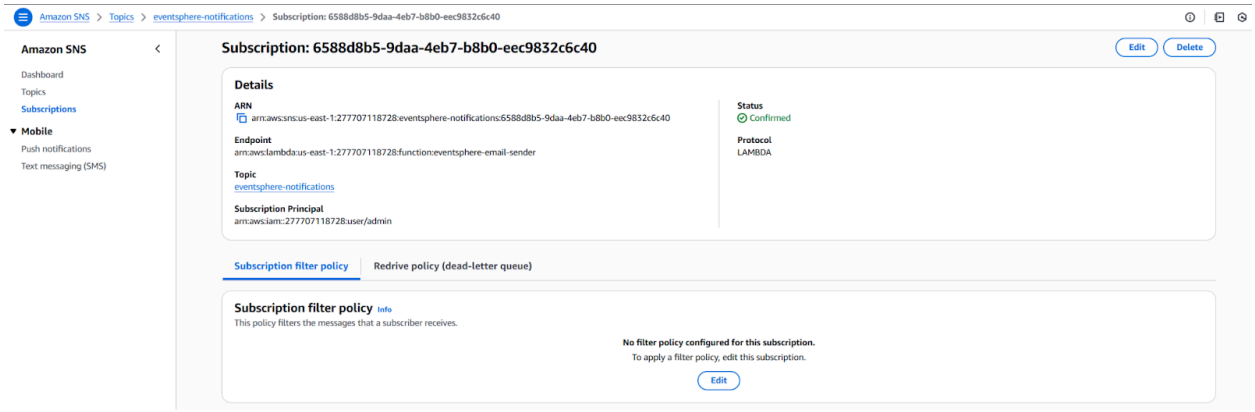
4.2 Amazon SNS (Message Broker)
SNS topic receives messages from the Notification Service and triggers Lambda.
Displays the SNS Topic List page, confirming that the topic *eventsphere-notifications* exists in the correct region.



Shows the SNS Topic Details page, including the Topic ARN and the number of connected subscriptions, verifying correct configuration and availability.



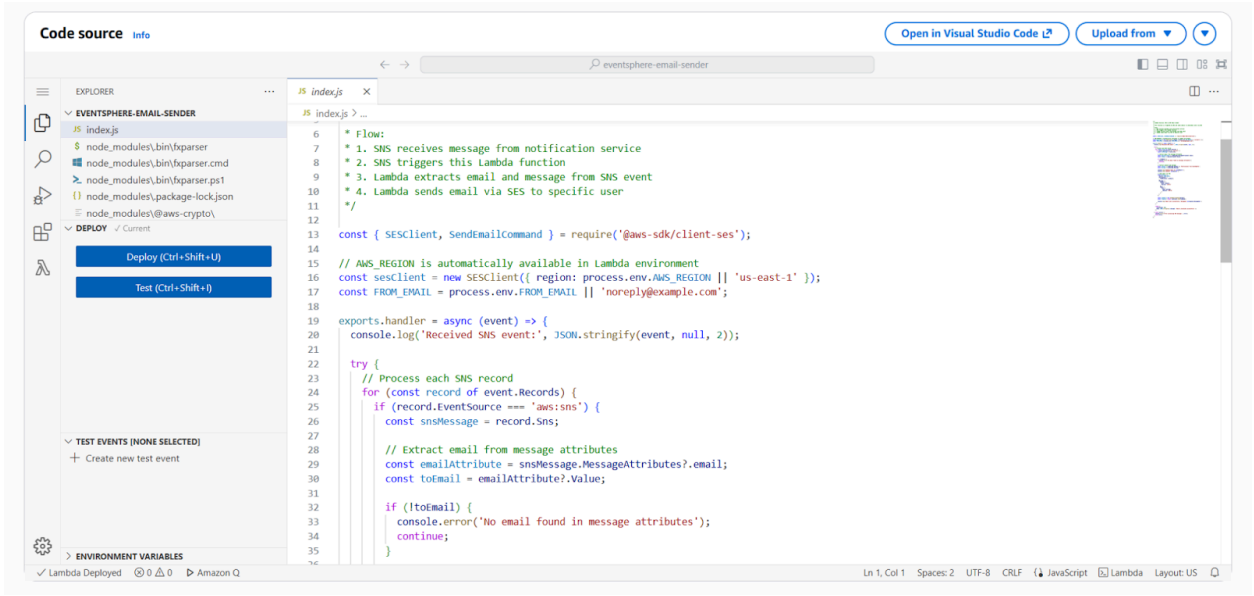
Provides the SNS Subscriptions page, confirming that the Notification Service's SNS topic is successfully subscribed to the Lambda function and that the subscription status is *Confirmed*.



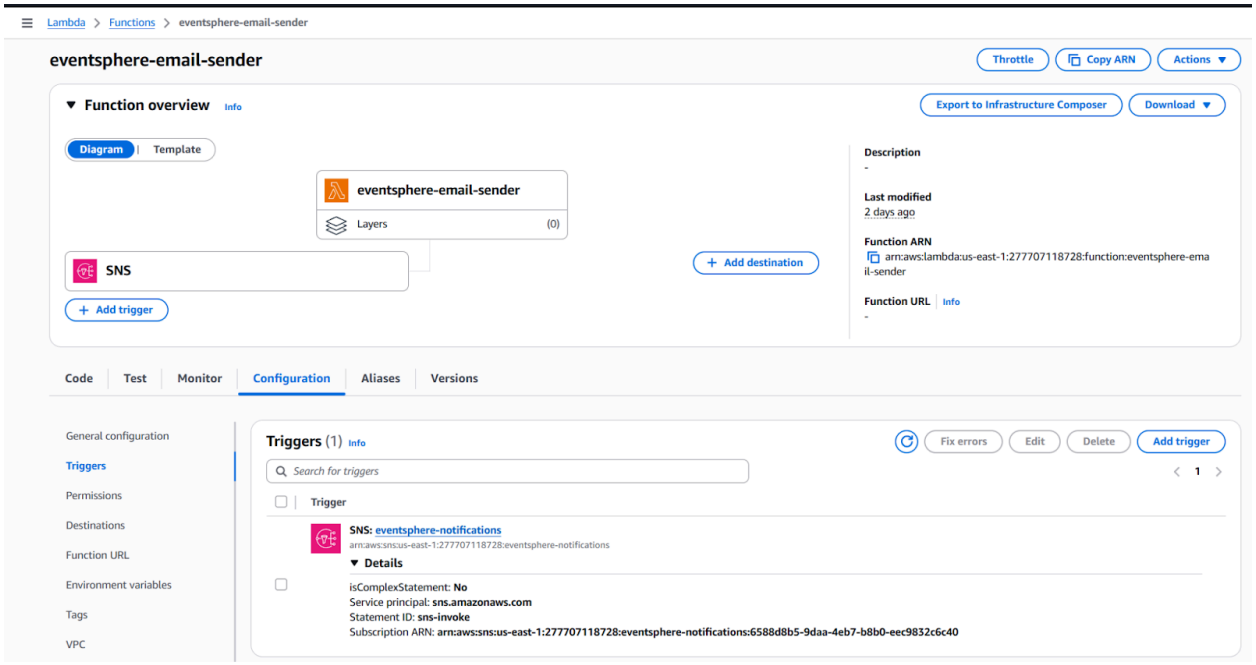
4.3 AWS Lambda (Email Processor)

Lambda receives SNS messages and sends emails using SES.

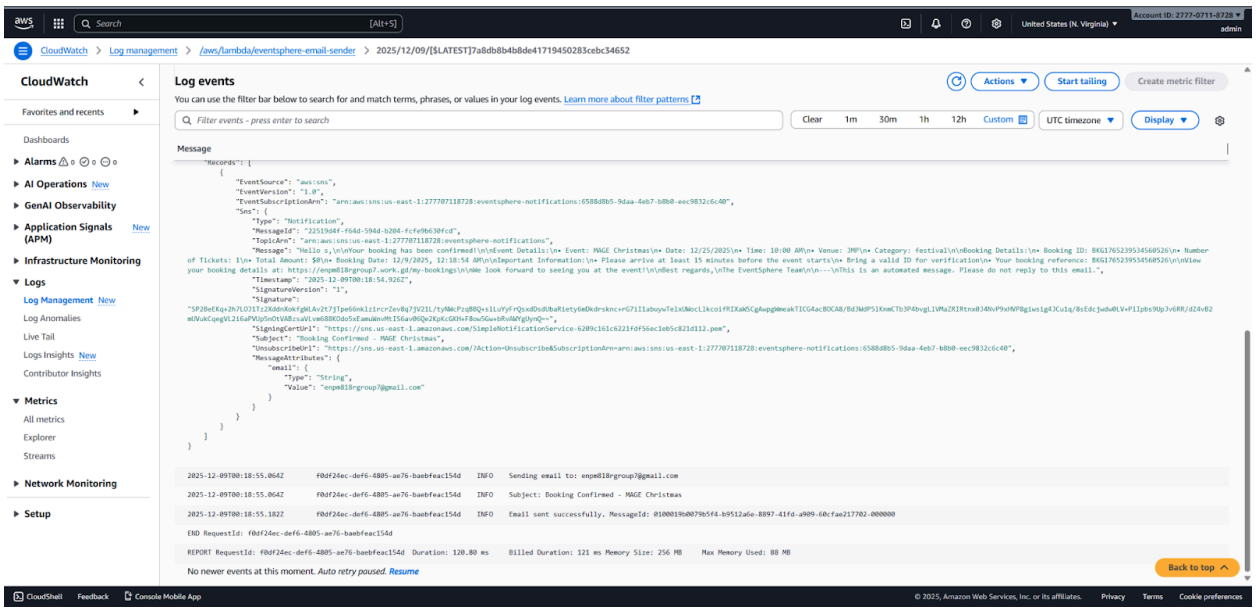
Shows the Lambda Code interface, index.js file implements SNS event handling and SES email sending.



Displays the Lambda Trigger configuration, confirming that the Lambda function is correctly triggered by the SNS topic defined earlier.



Shows the CloudWatch Lambda log stream, containing a log entry such as *"Email sent successfully. MessageId: ..."*, proving that the Lambda processor executed successfully and SES accepted the outgoing email.



Email received evidences

Welcome to EventSphere! Inbox ×

noreply@enpm818rgroup7.work.gd
to me

7:18 PM (4 minutes ago)

Hello s,

Welcome to EventSphere - Your Gateway to Amazing Events!

We're thrilled to have you join our community. EventSphere makes it easy to discover, book, and manage tickets for exciting events.

Here's what you can do:
• Browse upcoming events across various categories
• Book tickets instantly with our seamless booking system
• Manage your bookings and view your event history
• Get real-time updates about your events

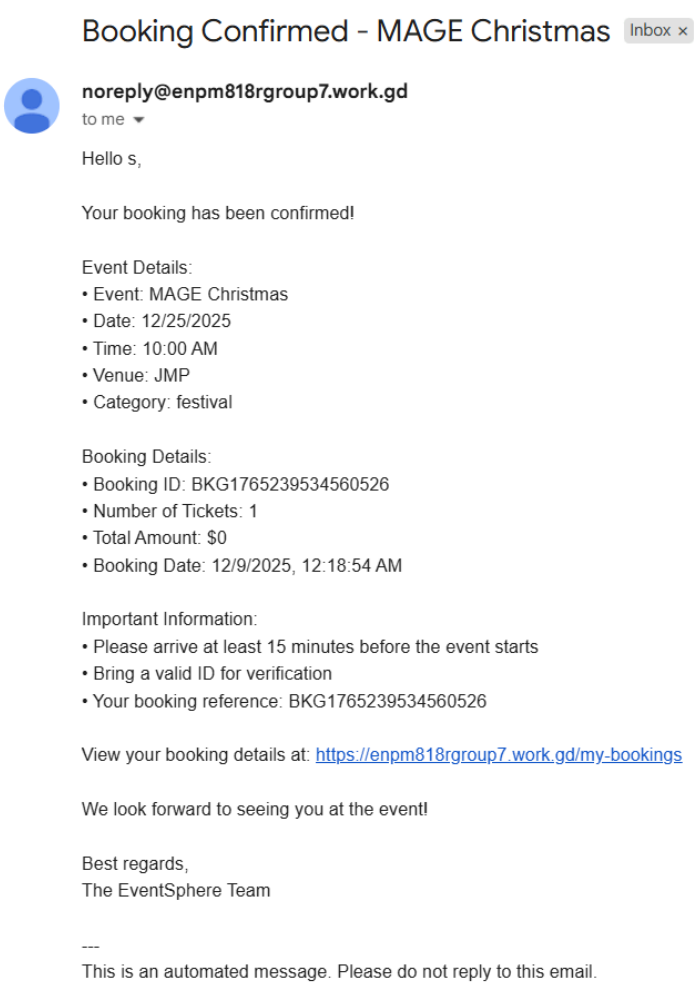
Start exploring events now at https://enpm818rgroup7.work.gd

If you have any questions, feel free to reach out to our support team.

Happy event hunting!

Best regards,
The EventSphere Team

This is an automated message. Please do not reply to this email.



Booking Confirmed - MAGE Christmas Inbox ×

noreply@enpm818rgroup7.work.gd
to me

Hello s,

Your booking has been confirmed!

Event Details:
• Event: MAGE Christmas
• Date: 12/25/2025
• Time: 10:00 AM
• Venue: JMP
• Category: festival

Booking Details:
• Booking ID: BKG1765239534560526
• Number of Tickets: 1
• Total Amount: $0
• Booking Date: 12/9/2025, 12:18:54 AM

Important Information:
• Please arrive at least 15 minutes before the event starts
• Bring a valid ID for verification
• Your booking reference: BKG1765239534560526

View your booking details at: https://enpm818rgroup7.work.gd/my-bookings

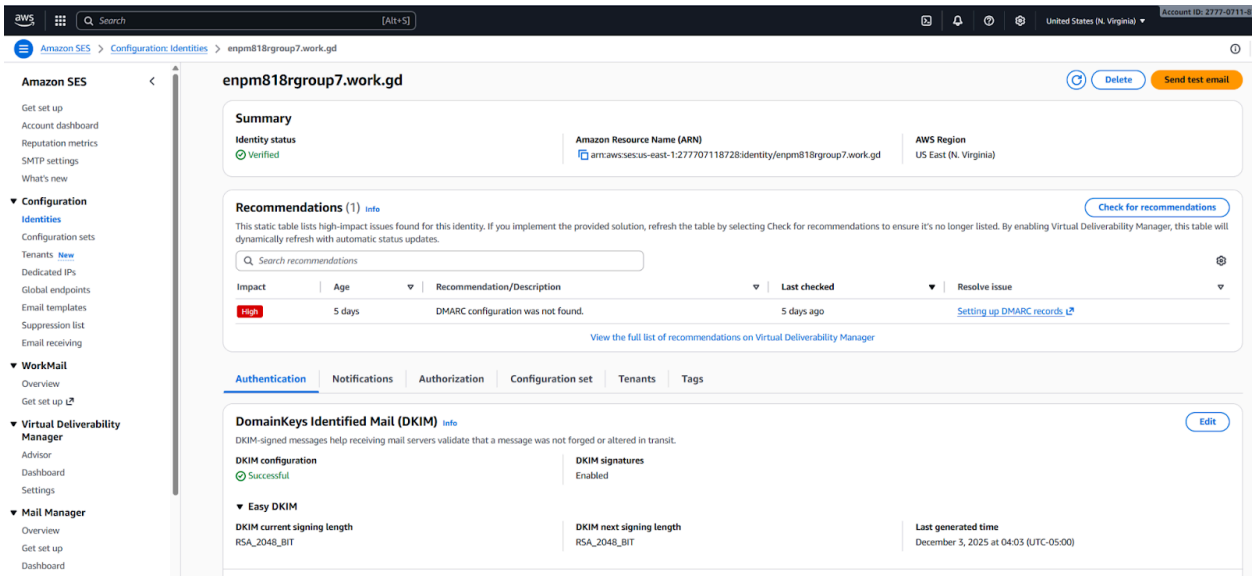
We look forward to seeing you at the event!

Best regards,
The EventSphere Team

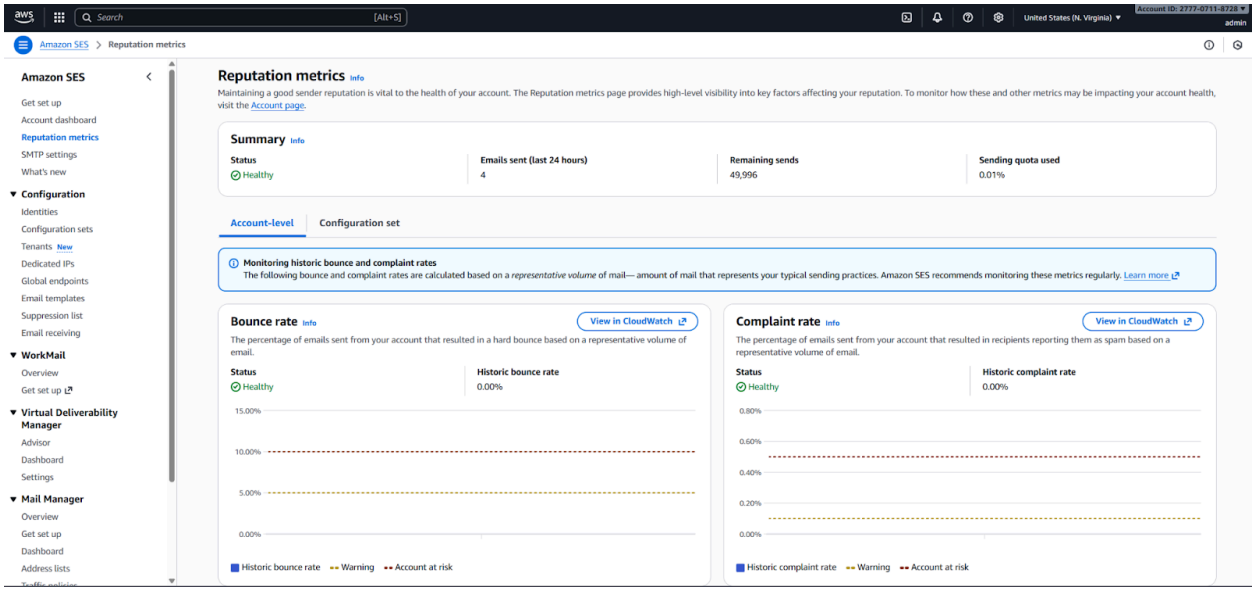
This is an automated message. Please do not reply to this email.

4.4 Amazon SES (Email Delivery)

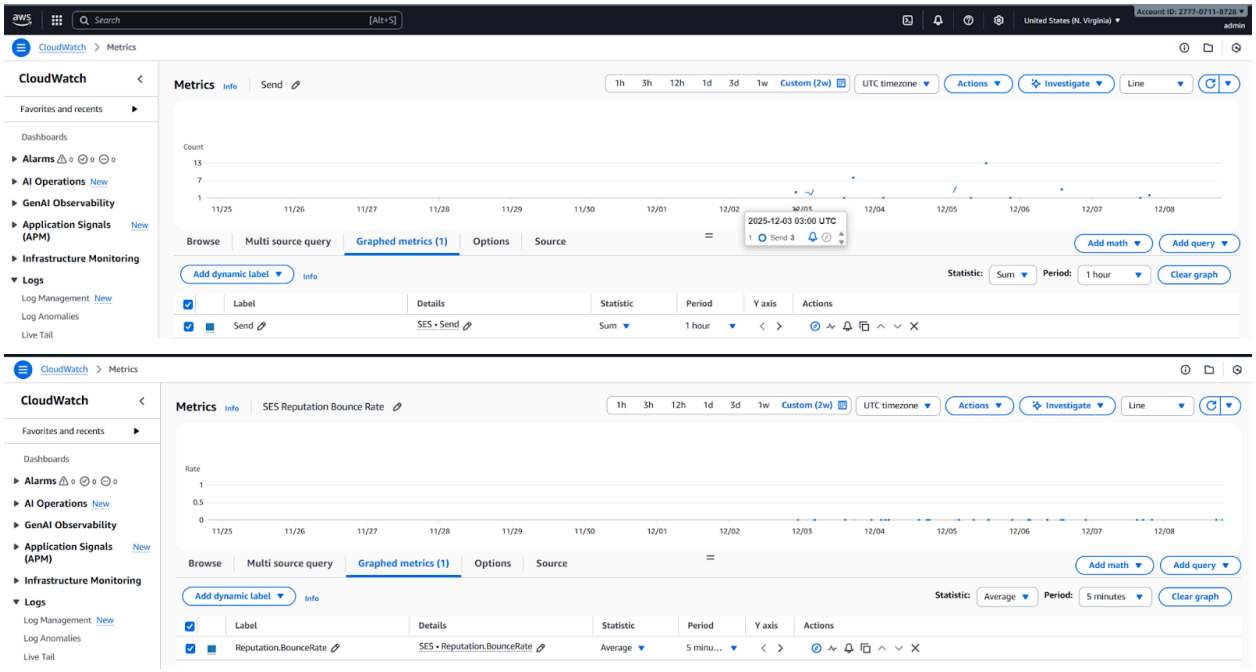
Displays the SES Sending Statistics dashboard, showing metrics such as total messages sent, delivery success rate, and bounce rates. This evidence confirms that SES is actively processing and delivering outgoing emails.



Shows the **SES Reputation Metrics dashboard**, confirming that SES is active and healthy.

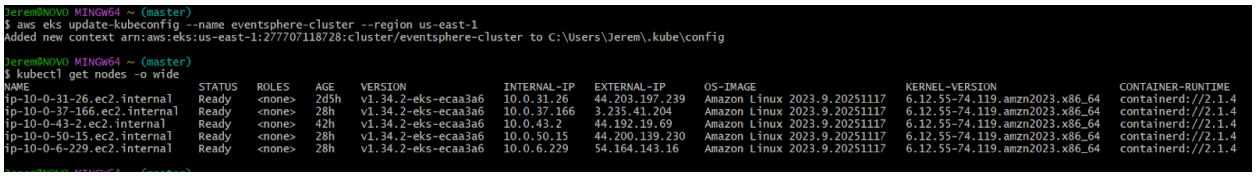


Metrics showing email sent and bounce rates in Cloudwatch.



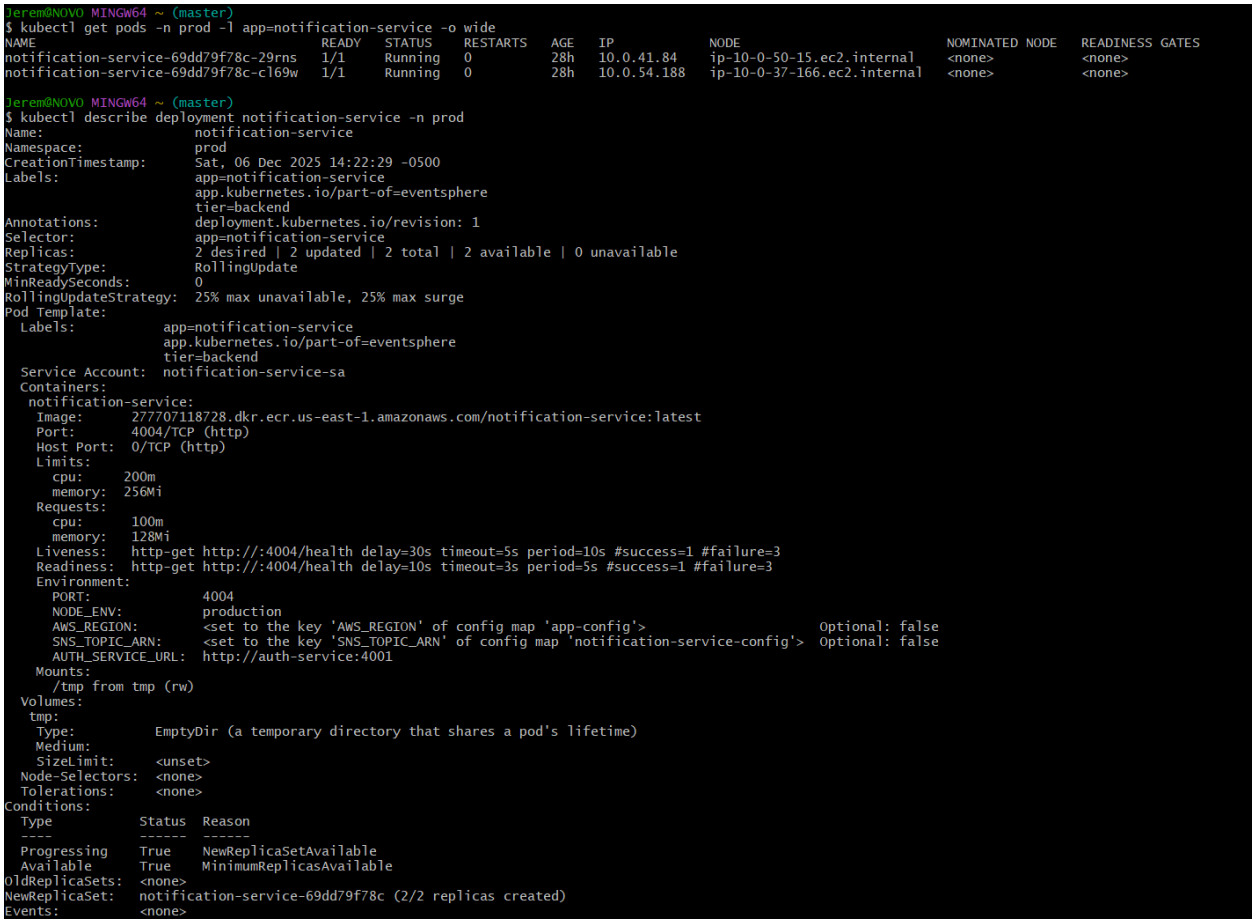
5. Infrastructure & Deployment (Kubernetes + AWS)

This screenshot shows the **EKS worker nodes** successfully registered with the cluster. All nodes report a Ready status, confirming that the EKS node group is functioning correctly and able to run workloads.



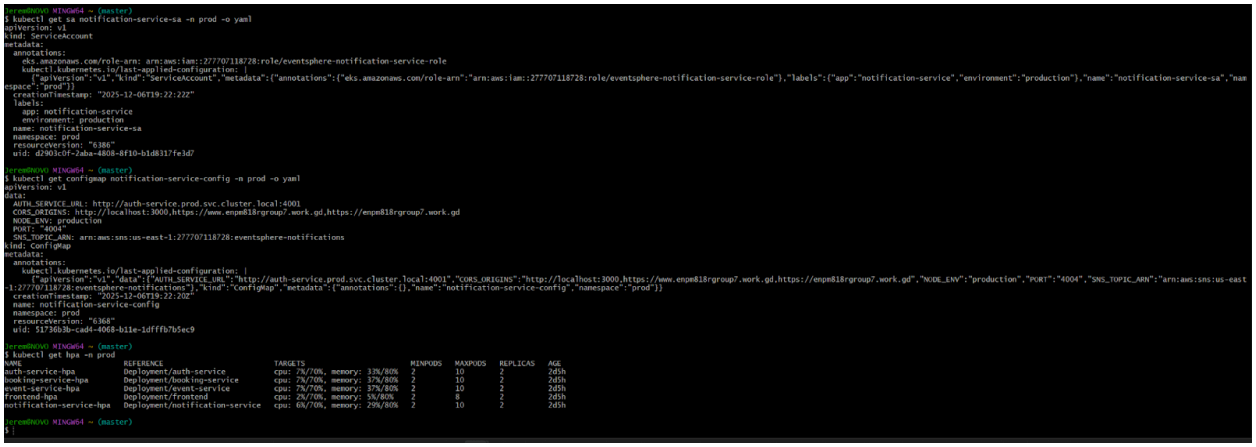
```
Jerem@NOVO MINGW64 ~ (master)
$ aws eks update-kubeconfig --name eventsphere-cluster --region us-east-1
Added new context arn:aws:eks:us-east-1:277707118728:cluster/eventsphere-cluster to C:\Users\Jerem\.kube\config

Jerem@NOVO MINGW64 ~ (master)
$ kubectl get nodes -o wide
NAME                          STATUS   ROLES    AGE    VERSION               INTERNAL-IP   EXTERNAL-IP      OS-IMAGE                        KERNEL-VERSION                  CONTAINER-RUNTIME
ip-10-0-31-26.ec2.internal    Ready    <none>   2d5h   v1.34.2-eks-ecaa3a6   10.0.31.26    44.203.197.239   Amazon Linux 2023.9.20251117    6.12.55-74.119.amzn2023.x86_64  containerd://2.1.4
ip-10-0-37-166.ec2.internal   Ready    <none>   28h    v1.34.2-eks-ecaa3a6   10.0.37.166   3.235.41.204     Amazon Linux 2023.9.20251117    6.12.55-74.119.amzn2023.x86_64  containerd://2.1.4
ip-10-0-43-2.ec2.internal     Ready    <none>   42h    v1.34.2-eks-ecaa3a6   10.0.43.2     44.192.19.69     Amazon Linux 2023.9.20251117    6.12.55-74.119.amzn2023.x86_64  containerd://2.1.4
ip-10-0-50-15.ec2.internal    Ready    <none>   28h    v1.34.2-eks-ecaa3a6   10.0.50.15    44.200.139.230   Amazon Linux 2023.9.20251117    6.12.55-74.119.amzn2023.x86_64  containerd://2.1.4
ip-10-0-6-229.ec2.internal    Ready    <none>   28h    v1.34.2-eks-ecaa3a6   10.0.6.229    54.164.143.16    Amazon Linux 2023.9.20251117    6.12.55-74.119.amzn2023.x86_64  containerd://2.1.4
```

This screenshot confirms that the Notification Service has **two running pods** in the prod namespace and displays the deployment configuration, including image version, replicas, environment variables, and health probes.



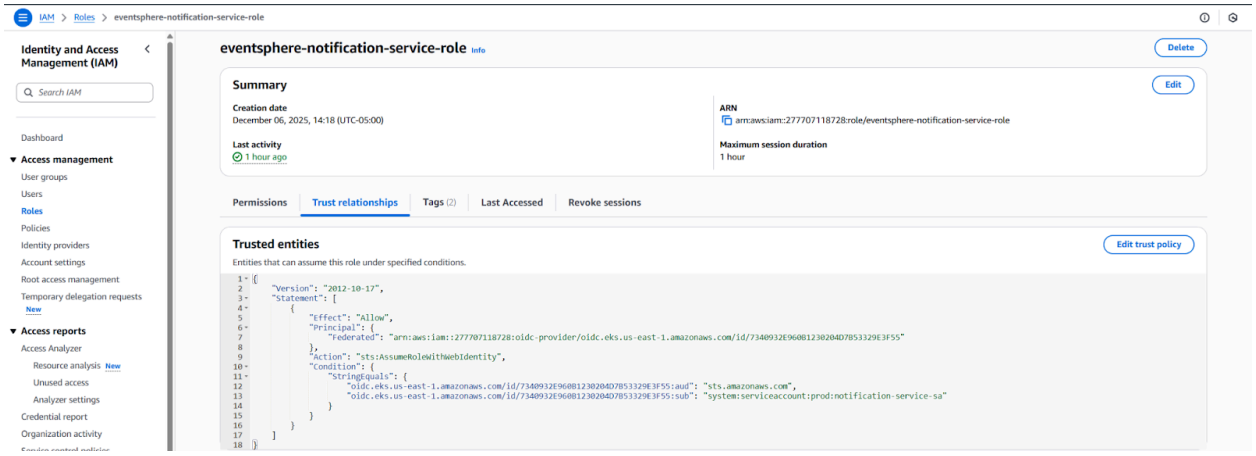
```
Jerem@NOVO MINGW64 ~ (master)
$ kubectl get pods -n prod -l app=notification-service -o wide
NAME                                    READY   STATUS    RESTARTS   AGE   IP            NODE                           NOMINATED NODE   READINESS GATES
notification-service-69dd79f78c-29rns   1/1     Running   0          28h   10.0.41.84    ip-10-0-50-15.ec2.internal     <none>           <none>
notification-service-69dd79f78c-cl69w   1/1     Running   0          28h   10.0.54.188   ip-10-0-37-166.ec2.internal    <none>           <none>

Jerem@NOVO MINGW64 ~ (master)
$ kubectl describe deployment notification-service -n prod
Name:                   notification-service
Namespace:              prod
CreationTimestamp:      Sat, 06 Dec 2025 14:22:29 -0500
Labels:                 app=notification-service
                        app.kubernetes.io/part-of=eventsphere
                        tier=backend
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=notification-service
Replicas:               2 desired | 2 updated | 2 total | 2 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:           app=notification-service
                    app.kubernetes.io/part-of=eventsphere
                    tier=backend
  Service Account:  notification-service-sa
  Containers:
   notification-service:
    Image:      277707118728.dkr.ecr.us-east-1.amazonaws.com/notification-service:latest
    Port:       4004/TCP (http)
    Host Port:  0/TCP (http)
    Limits:
      cpu:     200m
      memory:  256Mi
    Requests:
      cpu:      100m
      memory:   128Mi
    Liveness:   http-get http://:4004/health delay=30s timeout=5s period=10s #success=1 #failure=3
    Readiness:  http-get http://:4004/health delay=10s timeout=3s period=5s #success=1 #failure=3
    Environment:
      PORT:              4004
      NODE_ENV:          production
      AWS_REGION:        <set to the key 'AWS_REGION' of config map 'app-config'>                    Optional: false
      SNS_TOPIC_ARN:     <set to the key 'SNS_TOPIC_ARN' of config map 'notification-service-config'>  Optional: false
      AUTH_SERVICE_URL:  http://auth-service:4001
    Mounts:
      /tmp from tmp (rw)
  Volumes:
   tmp:
    Type:          EmptyDir (a temporary directory that shares a pod's lifetime)
    Medium:
    SizeLimit:     <unset>
  Node-Selectors:  <none>
  Tolerations:     <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Progressing    True    NewReplicaSetAvailable
  Available      True    MinimumReplicasAvailable
OldReplicaSets:  <none>
NewReplicaSet:   notification-service-69dd79f78c (2/2 replicas created)
Events:          <none>
```

This screenshot shows the Notification Service's **IRSA-enabled ServiceAccount**, its ConfigMap with environment settings, and the Horizontal Pod Autoscaler, confirming proper AWS integration, externalized configuration, and autoscaling.

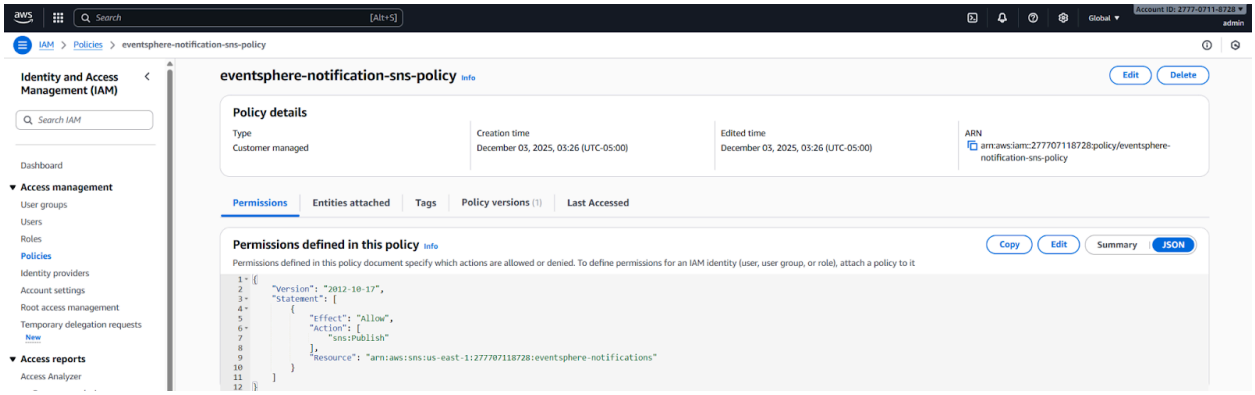


5.1 IAM Role

IAM Role trust relationship for eventsphere-notification-service-role. This policy shows that only the EKS OIDC provider and the Kubernetes ServiceAccount prod:notification-service-sa are allowed to assume this role via sts:AssumeRoleWithWebIdentity. This confirms that the Notification Service uses IRSA for secure, credential-free access to AWS services.



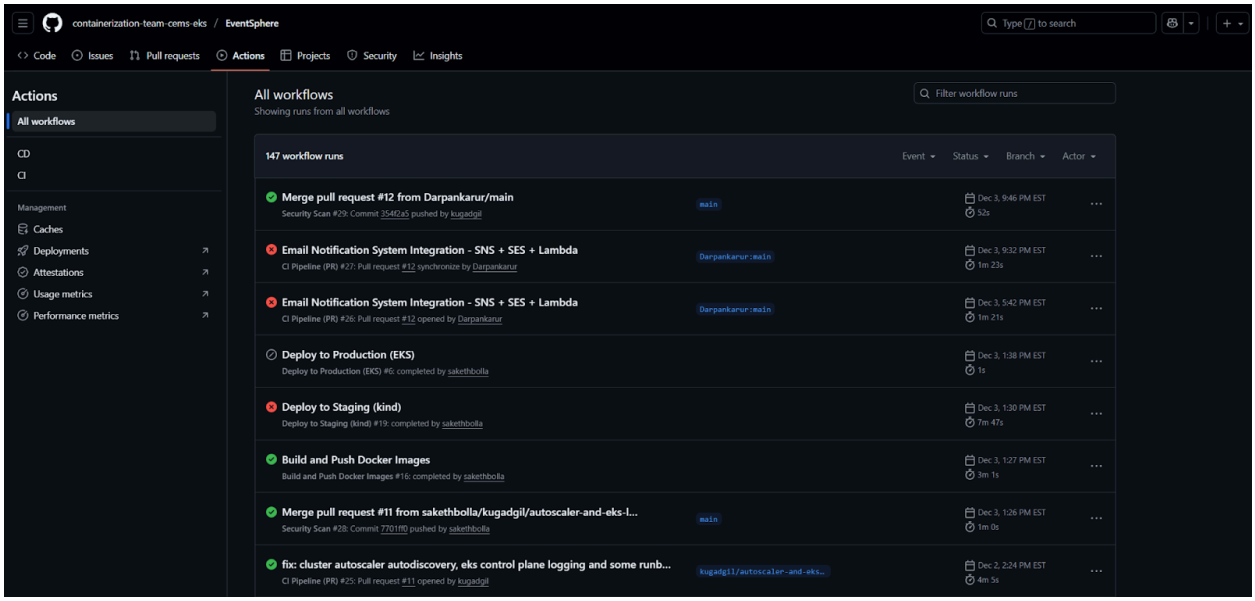
IAM policy eventsphere-notification-sns-policy attached to the Notification Service role. The policy grants the minimum required permission—sns:Publish—allowing the service to publish notification messages to the SNS topic eventsphere-notifications.



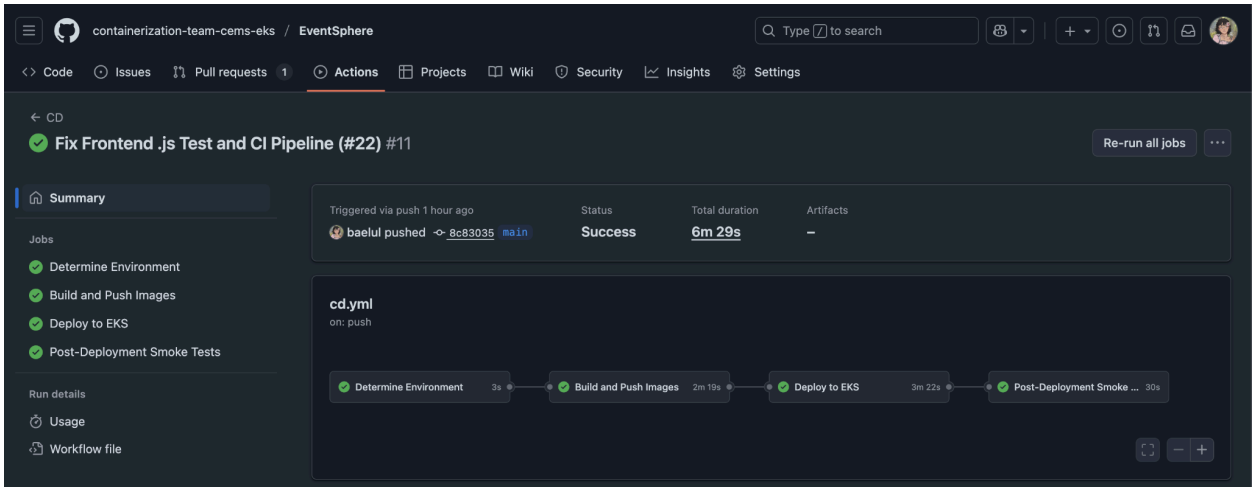
6.CI/CD Pipeline (Notification Service)

The Notification Service is integrated into the EventSphere GitHub Actions **CI/CD pipeline.** Any change in the services/notification-service/ directory triggers the CI workflow, which runs security scans, validates Kubernetes manifests, and builds the container image. After changes are merged into main, the CD pipeline builds and pushes the final Docker image to ECR and deploys it to EKS using Helm with atomic rollouts and automated health checks.

Below screenshot shows the full set of **CI/CD workflows running** in the EventSphere repository, including the Notification Service integration pipeline, the "Build and Push Docker Images" job, and the "Deploy to Production (EKS)" workflow. The presence of successful and failed runs demonstrates that automated pipelines enforce quality control and ensure reliable deployments to EKS. too long



This screenshot displays the **complete GitHub Actions CD pipeline** for the EventSphere project. The workflow includes environment selection, image build and push to ECR, Helm deployment to EKS, and post-deployment smoke tests. All steps succeeded, confirming reliable automated deployment of the Notification Service to the cluster.


containerization-team-cems-eks / EventSphere
Type / to search
Code
Issues
Pull requests 1
Actions
Projects
Wiki
Security
Insights
Settings
CD
Fix Frontend .js Test and CI Pipeline (#22) #11
Re-run all jobs
Summary
Jobs
Determine Environment
Build and Push Images
Deploy to EKS
Post-Deployment Smoke Tests
Run details
Usage
Workflow file
Triggered via push 1 hour ago
baelul pushed 8c83035 main
Status
Success
Total duration
6m 29s
Artifacts
–
cd.yml
on: push
Determine Environment 3s
Build and Push Images 2m 19s
Deploy to EKS 3m 22s
Post-Deployment Smoke ... 30s

Reflection

The successful deployment of EventSphere was a huge learning process for the team. We learnt how to build a scalable, micro-service architecture platform, with monitoring, CI/CD pipeline, and version control system. We learnt how to use Kubernetes in the cloud with AWS EKS, and how to build and use it securely. Our CI/CD pipeline allowed us to update our production servers/pods live, with no downtime, which gave us insights into how real world applications are built and maintained. We deployed an auto-scaling cluster that scaled up and down based on traffic and demand, which meant low costs during low traffic, and high availability during peak traffic times. We learnt how to build and integrate microservices together, and how we could use pods that scale automatically based on each specific microservice's demands. With SNS and SES, we developed a robust system that could both send emails and notifications to users regarding events, and developers regarding operational triggers, reducing response time to errors/problems that occur during live production.

Overall, this was a very challenging, yet rewarding project, that taught each one of us how to deploy and maintain a microservice, autoscaling application in the cloud.