Capstone Project Documentation

GitHub repo: -

https://github.com/Juhi5863/Juhi-Capstone-Project.git

cloud formation template infra repo: -

https://github.com/Juhi5863/CFT.git

Pipeline in us-east-1 project repo: -

https://github.com/Juhi5863/pipeline-east-1-CF.git

Terraform template for creating infra in us-east-2 region repo:-

https://github.com/Juhi5863/terraform-infra.git

Pipeline in us-east-2 project repo: -

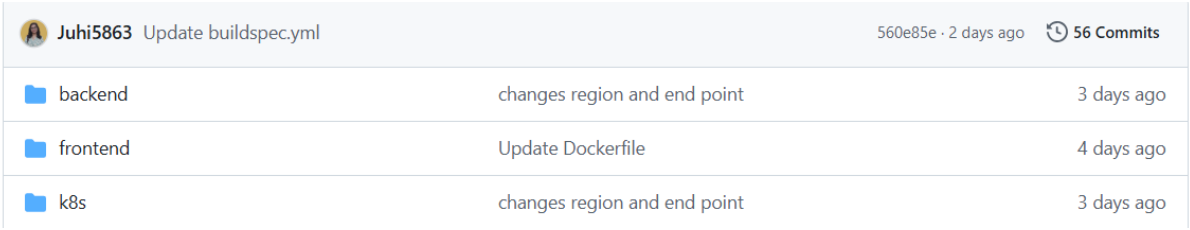
https://github.com/Juhi5863/terraform-east2-deploy.git

project structure

backend (node js)

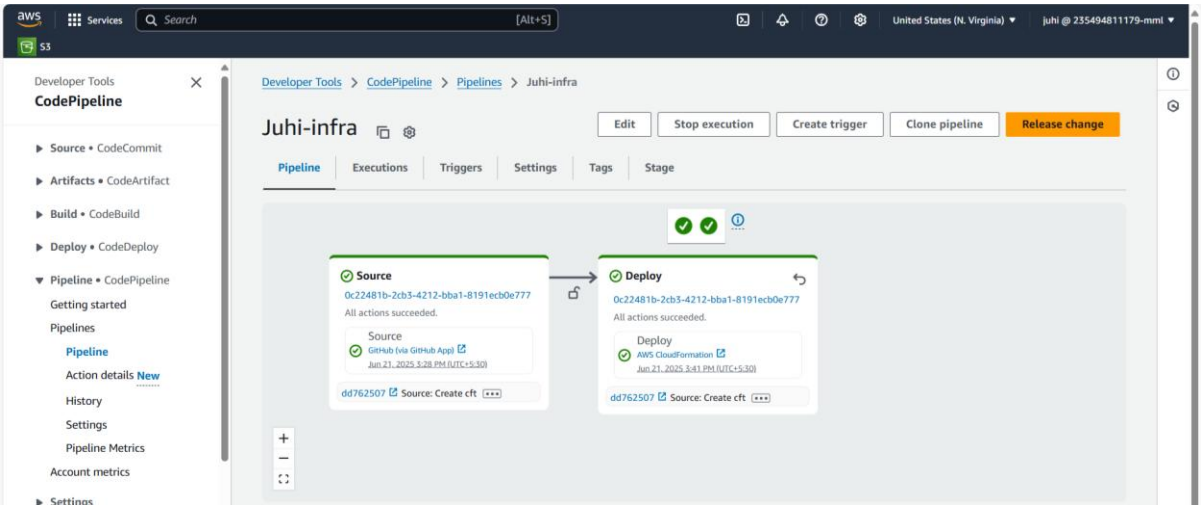
frontend (React)

database (rds MySQL)



Cloud Formation (us-east-1)

1. Infrastructure created in us-east-1 with CFT through Code Pipeline



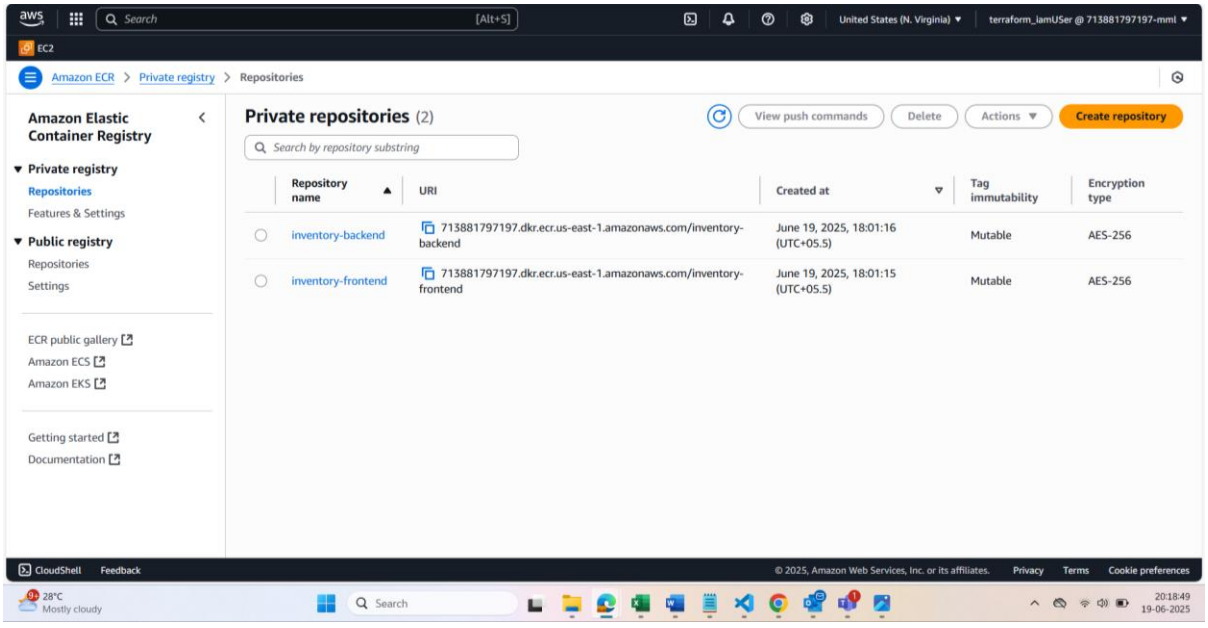
kubectl get nodes

```
root@ip-10-0-11-146:/home/ubuntu# kubectl get nodes
NAME                          STATUS   ROLES    AGE   VERSION
ip-10-0-15-178.ec2.internal   Ready    <none>   26h   v1.32.3-eks-473151a
ip-10-0-66-57.ec2.internal    Ready    <none>   26h   v1.32.3-eks-473151a
```

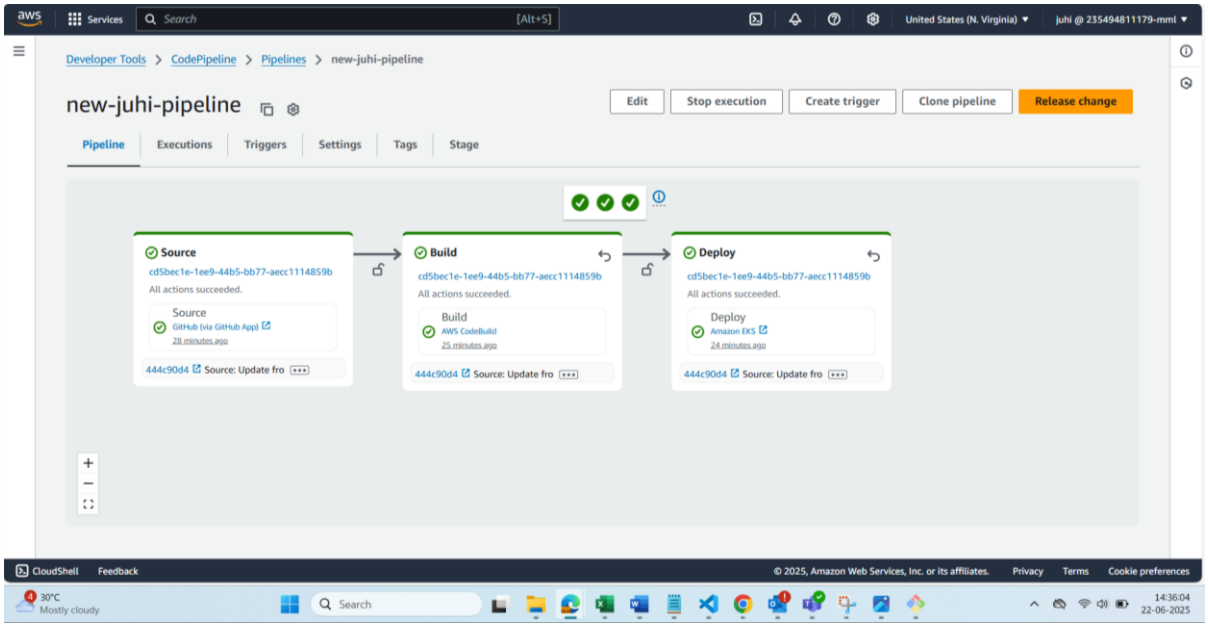
mysql endpoint -u admin -p

```
root@ip-10-0-11-146:/home/ubuntu# mysql -h mysqldatabase.cn0us0ksu6a4.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.41 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use inventory;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+----------------------+
| Tables_in_inventory |
+----------------------+
| items                |
+----------------------+
```

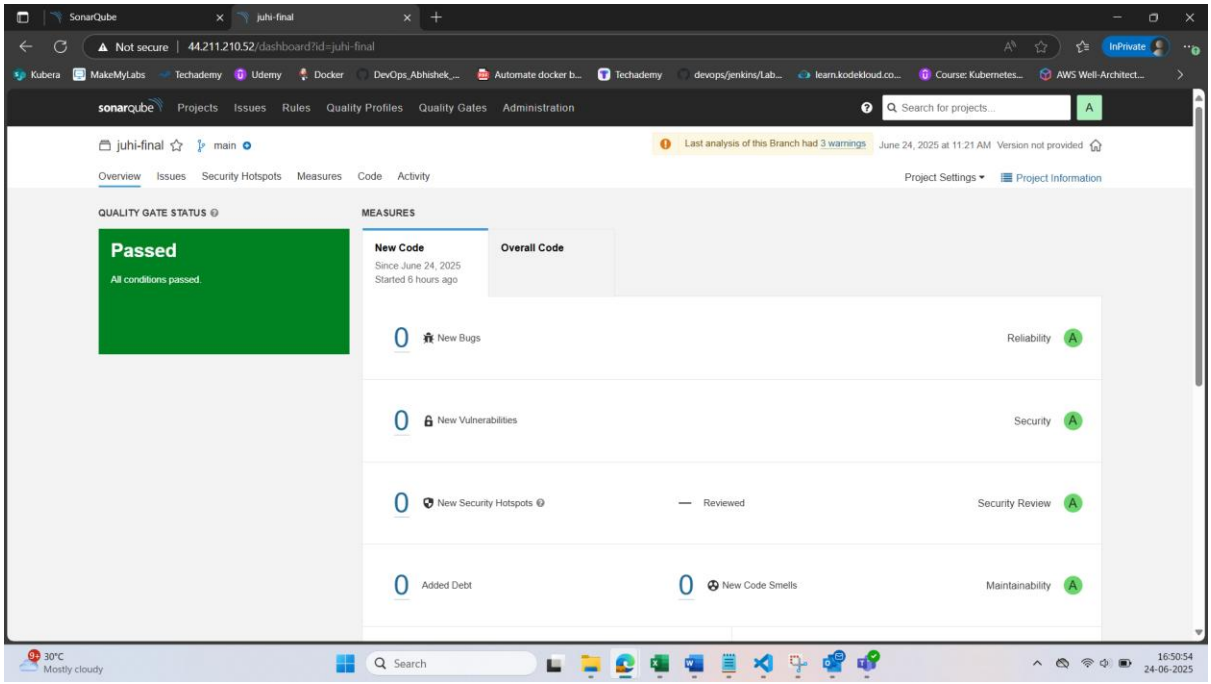
2. Ecr created from both backend and frontend image



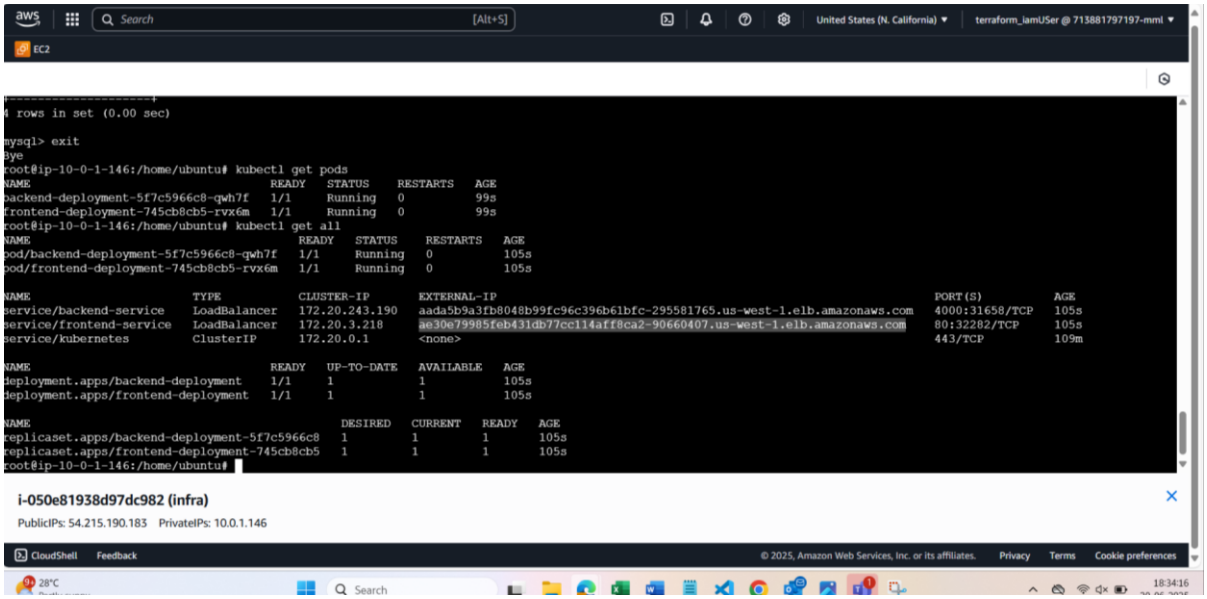
3. Pipeline for application deployment



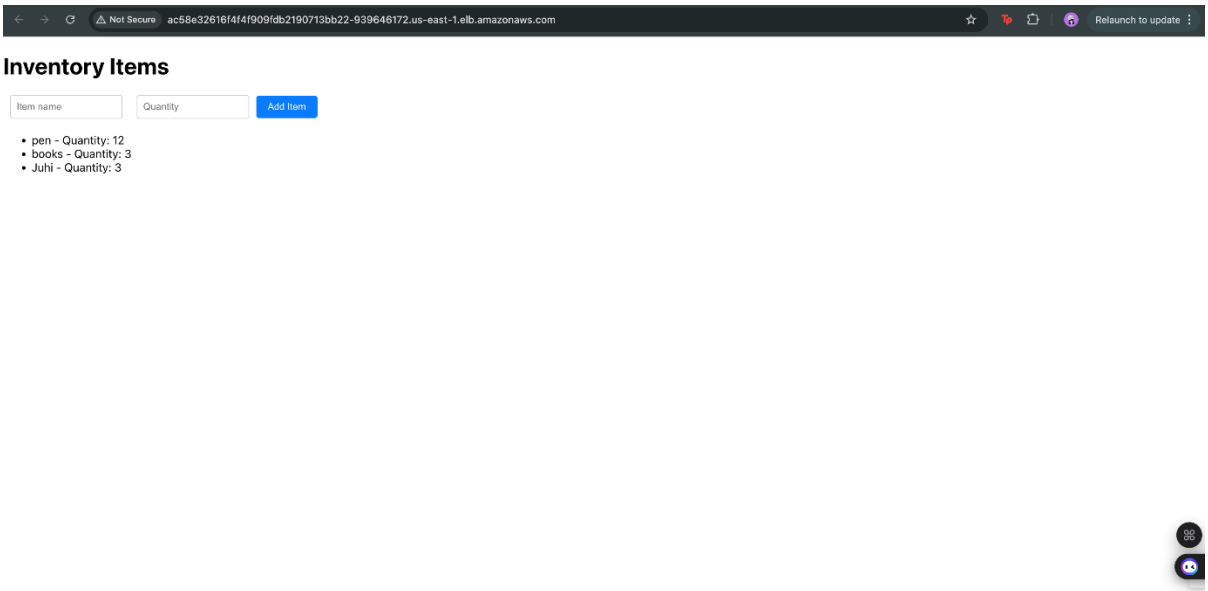
SonarQube – added in buildspec.yml



4. Check in ec2 instance everything pod and load balancer external dns to access the link



5. Application access through external-ip of frontend-service

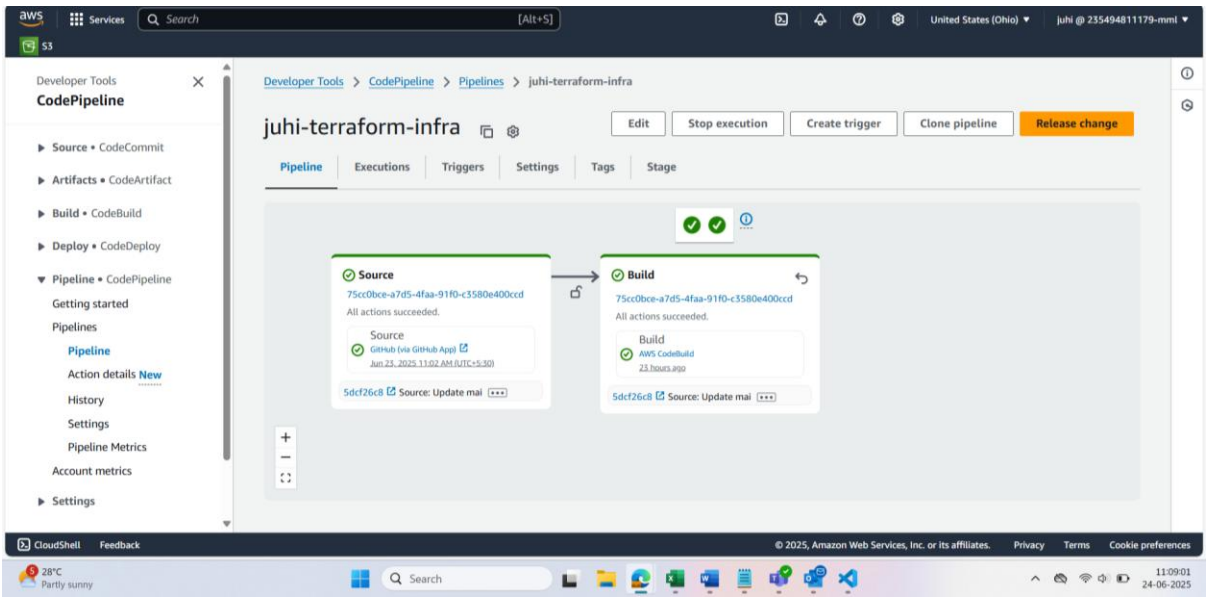


7. Access through backend database

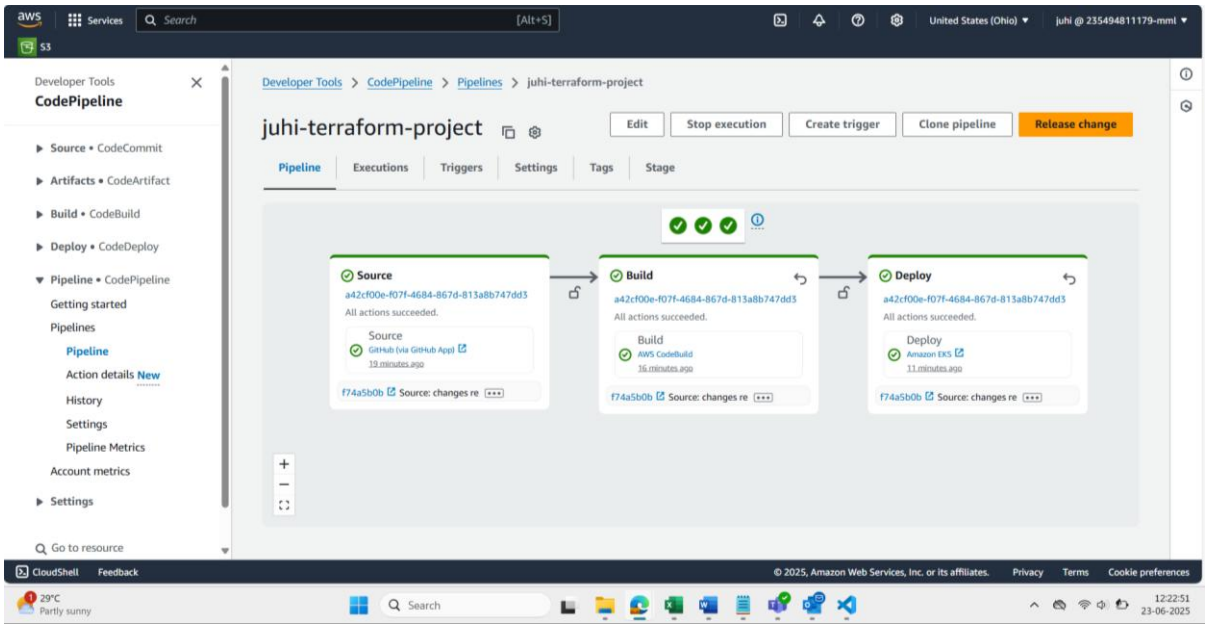
```
mysql> use inventory;
Database changed
mysql> show tables;
+---------------------+
| Tables_in_inventory |
+---------------------+
| clients             |
| items               |
+---------------------+
2 rows in set (0.01 sec)

mysql> select * from items;
+----+--------+----------+---------------------+
| id | name   | quantity | created_at          |
+----+--------+----------+---------------------+
|  1 | pen    |       12 | 2025-06-19 14:20:33 |
|  2 | books  |        3 | 2025-06-19 14:26:17 |
|  3 | Juhi   |        3 | 2025-06-19 14:29:03 |
+----+--------+----------+---------------------+
3 rows in set (0.00 sec)

mysql>
```

Terraform (us-east-2)

1. pipeline – terraform infra



2. Terraform pipeline deployment application



Application access through external-ip of frontend-service

And we can see the data from backend what items added

```
mysql> use inventory;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from items;
+----+-------------+----------+
| id | name        | quantity |
+----+-------------+----------+
|  1 | test        |        5 |
|  2 | Sample Item |        3 |
|  3 | testUU      |     2345 |
|  4 | JBNHBYHGBG  |  2345678 |
|  5 | books       |        5 |
|  6 | bags        |        5 |
|  7 | Shoes       |        3 |
|  8 | Phone       |        1 |
|  9 | Baggg       |        8 |
| 10 | Laptop      |        2 |
| 11 | Card        |        2 |
| 12 | Chair       |        3 |
| 13 | Mouse       |        9 |
| 14 | Balls       |        5 |
+----+-------------+----------+
14 rows in set (0.00 sec)
```

setup monitoring on Kubernetes Cluster using Prometheus and Grafana

region – us-east-1

Pod

```
root@ip-10-0-11-146:/home/ubuntu# kubectl get pods -n prometheus
NAME                                                     READY   STATUS    RESTARTS   AGE
alertmanager-stable-kube-prometheus-sta-alertmanager-0   2/2     Running   0          3m19s
prometheus-stable-kube-prometheus-sta-prometheus-0       2/2     Running   0          3m19s
stable-grafana-56bdbb9b4b-gr9w9                          3/3     Running   0          3m23s
stable-kube-prometheus-sta-operator-66847554f8-4z716     1/1     Running   0          3m23s
stable-kube-state-metrics-7fc6b5c5d4-wdbfq               1/1     Running   0          3m23s
stable-prometheus-node-exporter-kdkzd                    1/1     Running   0          3m23s
stable-prometheus-node-exporter-wrxz7                    1/1     Running   0          3m23s
```

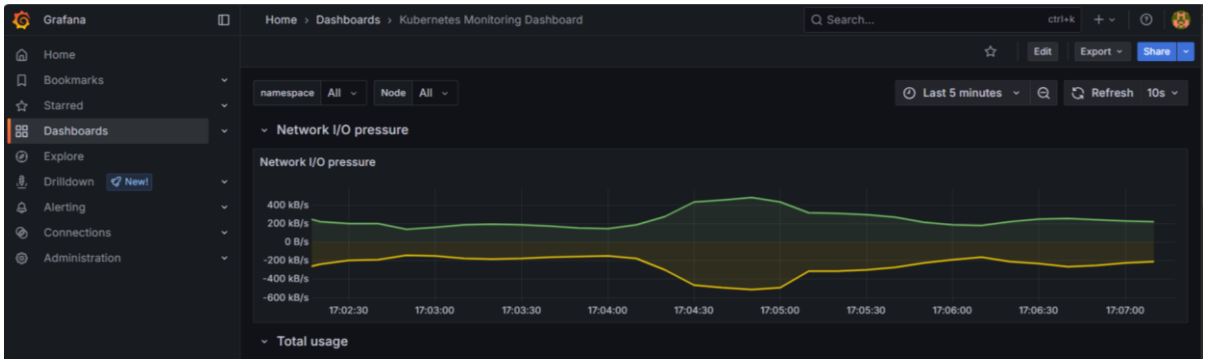
Svc

```
root@ip-10-0-11-146:/home/ubuntu# kubectl get svc -n prometheus
NAME                                      TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                      AGE
alertmanager-operated                     ClusterIP   None             <none>        9093/TCP,9094/TCP,9094/UDP   3m25s
prometheus-operated                       ClusterIP   None             <none>        9090/TCP                     3m25s
stable-grafana                            ClusterIP   172.20.215.19    <none>        80/TCP                       3m29s
stable-kube-prometheus-sta-alertmanager   ClusterIP   172.20.254.168   <none>        9093/TCP,8080/TCP            3m29s
stable-kube-prometheus-sta-operator       ClusterIP   172.20.77.48     <none>        443/TCP                      3m29s
stable-kube-prometheus-sta-prometheus     ClusterIP   172.20.60.96     <none>        9090/TCP,8080/TCP            3m29s
stable-kube-state-metrics                 ClusterIP   172.20.166.118   <none>        8080/TCP                     3m29s
stable-prometheus-node-exporter           ClusterIP   172.20.211.46    <none>        9100/TCP                     3m29s
```

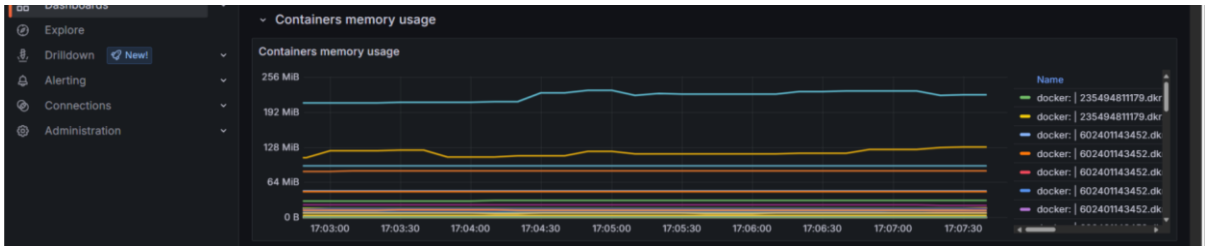
Load balancer

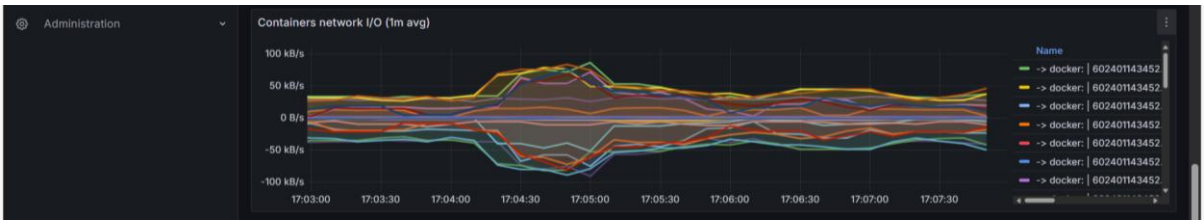
```
root@ip-10-0-11-146:/home/ubuntu# kubectl edit svc stable-grafana -n prometheus
service/stable-grafana edited
root@ip-10-0-11-146:/home/ubuntu# kubectl get svc -n prometheus
NAME                                      TYPE           CLUSTER-IP       EXTERNAL-IP                                                              PORT(S)
          AGE
alertmanager-operated                     ClusterIP      None             <none>                                                                   9093/TCP,9094/TCP,
9094/UDP      6m20s
prometheus-operated                       ClusterIP      None             <none>                                                                   9090/TCP
          6m20s
stable-grafana                            LoadBalancer   172.20.215.19    a432af93c9ab946f6843a810ec4caadf-1255529578.us-east-1.elb.amazonaws.com   80:32273/TCP
          6m24s
stable-kube-prometheus-sta-alertmanager   ClusterIP      172.20.254.168   <none>                                                                   9093/TCP,8080/TCP
          6m24s
stable-kube-prometheus-sta-operator       ClusterIP      172.20.77.48     <none>                                                                   443/TCP
          6m24s
stable-kube-prometheus-sta-prometheus     LoadBalancer   172.20.60.96     a7be0f946f8af4be785bbbd138900766-1896687863.us-east-1.elb.amazonaws.com   9090:31896/TCP,808
0:30277/TCP   6m24s
stable-kube-state-metrics                 ClusterIP      172.20.166.118   <none>                                                                   8080/TCP
          6m24s
stable-prometheus-node-exporter           ClusterIP      172.20.211.46    <none>                                                                   9100/TCP
          6m24s
root@ip-10-0-11-146:/home/ubuntu#
```

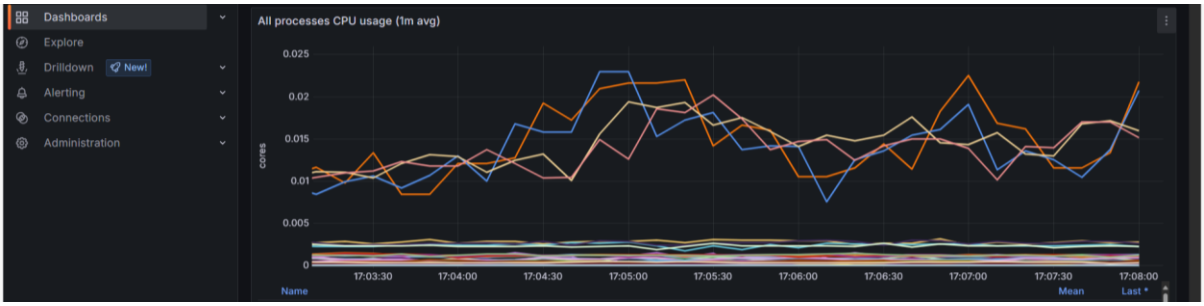
Kubernetes Monitoring Dashboard - Dashboards - Grafana




Explore
Drilldown
New!
Alerting
Connections
Administration
Containers CPU usage
Containers CPU usage (1m avg)
cores
0.025
0.02
0.015
0.01
0.005
0
17:03:00
17:04:00
17:05:00
17:06:00
17:07:00
Name
docker: | 235494811179.dkr.ecr.us-east-1.amazonaws.com/inventory-backend:latest (037a8020e6a8...
docker: | 235494811179.dkr.ecr.us-east-1.amazonaws.com/inventory-frontend:latest (ffcf7cc6a1fdfd6...
docker: | 602401143452.dkr.ecr.us-east-1.amazonaws.com/amazon-k8s-cni:v1.19.2-eksbuild.1 (1a15a...
docker: | 602401143452.dkr.ecr.us-east-1.amazonaws.com/amazon-k8s-cni:v1.19.2-eksbuild.1 (65c9...
docker: | 602401143452.dkr.ecr.us-east-1.amazonaws.com/amazon/aws-network-policy-agent:v1.1.6...
docker: | 602401143452.dkr.ecr.us-east-1.amazonaws.com/amazon/aws-network-policy-agent:v1.1.6...
docker: | 602401143452.dkr.ecr.us-east-1.amazonaws.com/eks/coredns:v1.11.4-eksbuild.2 (2fd32494...


Explore
Drilldown
New!
Alerting
Connections
Administration
Containers memory usage
Containers memory usage
256 MiB
192 MiB
128 MiB
64 MiB
0 B
17:03:00
17:03:30
17:04:00
17:04:30
17:05:00
17:05:30
17:06:00
17:06:30
17:07:00
17:07:30
Name


Administration
Containers network I/O (1m avg)
100 kB/s
50 kB/s
0 B/s
-50 kB/s
-100 kB/s
17:03:00
17:03:30
17:04:00
17:04:30
17:05:00
17:05:30
17:06:00
17:06:30
17:07:00
17:07:30
Name
-> docker: | 602401143452
-> docker: | 602401143452
-> docker: | 602401143452
-> docker: | 602401143452
-> docker: | 602401143452
-> docker: | 602401143452
-> docker: | 602401143452


Dashboards
Explore
Drilldown
New!
Alerting
Connections
Administration
All processes CPU usage (1m avg)
cores
0.025
0.02
0.015
0.01
0.005
0
17:03:30
17:04:00
17:04:30
17:05:00
17:05:30
17:06:00
17:06:30
17:07:00
17:07:30
17:08:00
Name
Mean
Last *


Drilldown
New!
Alerting
Connections
Administration
All processes memory usage
320 MiB
256 MiB
192 MiB
128 MiB
64 MiB
0 B
17:03:30
17:04:00
17:04:30
17:05:00
17:05:30
17:06:00
17:06:30
17:07:00
17:07:30
17:08:00


Explore
Drilldown
New!
Alerting
Connections
Administration
All processes network I/O (1m avg)
100 kB/s
75 kB/s
50 kB/s
25 kB/s
0 B/s
-25 kB/s
-50 kB/s
-75 kB/s
-100 kB/s
17:03:30
17:04:00
17:04:30
17:05:00
17:05:30
17:06:00
17:06:30
17:07:00
17:07:30
17:08:00

Cloud watch - In us-east-2

1. (One-Time) Enable IAM OIDC Provider for Your EKS Cluster

```
eksctl utils associate-iam-oidc-provider \
 --cluster MyEKSCluster  \
 --region us-east-2 \
 --approve
```

2. Create the IAM Policy for the Agent (One-Time Per Account)

```
cat <<EOF > cwagent-policy.json
{
 "Version": "2012-10-17",
 "Statement": [{
  "Effect": "Allow",
  "Action": [
   "logs:PutLogEvents",
   "logs:DescribeLogStreams",
   "logs:DescribeLogGroups",
   "logs:CreateLogStream",
   "logs:CreateLogGroup",
   "logs:PutRetentionPolicy",
   "cloudwatch:PutMetricData"
  ],
  "Resource": "*"
 }]
}
EOF

aws iam create-policy \
 --policy-name CloudWatchAgentServerPolicy \
 --policy-document file://cwagent-policy.json
```

3. **Create the IAM Service Account for the Agent**

```
eksctl create iamserviceaccount \
 --cluster MyEKSCluster \
 --namespace amazon-cloudwatch \
 --name cloudwatch-agent \
 --attach-policy-arn arn:aws:iam::<your-account-id>:policy/CloudWatchAgentServerPolicy \
 --approve \
 --region us-east-2
```

4. I**nstall the CloudWatch Observability EKS Add-on (This is the AWS-recommended, auto-managed way!)**

```
aws eks create-addon \
 --cluster-name MyEKSCluster \
 --addon-name amazon-cloudwatch-observability \
 --resolve-conflicts OVERWRITE \
 --region us-east-2
```

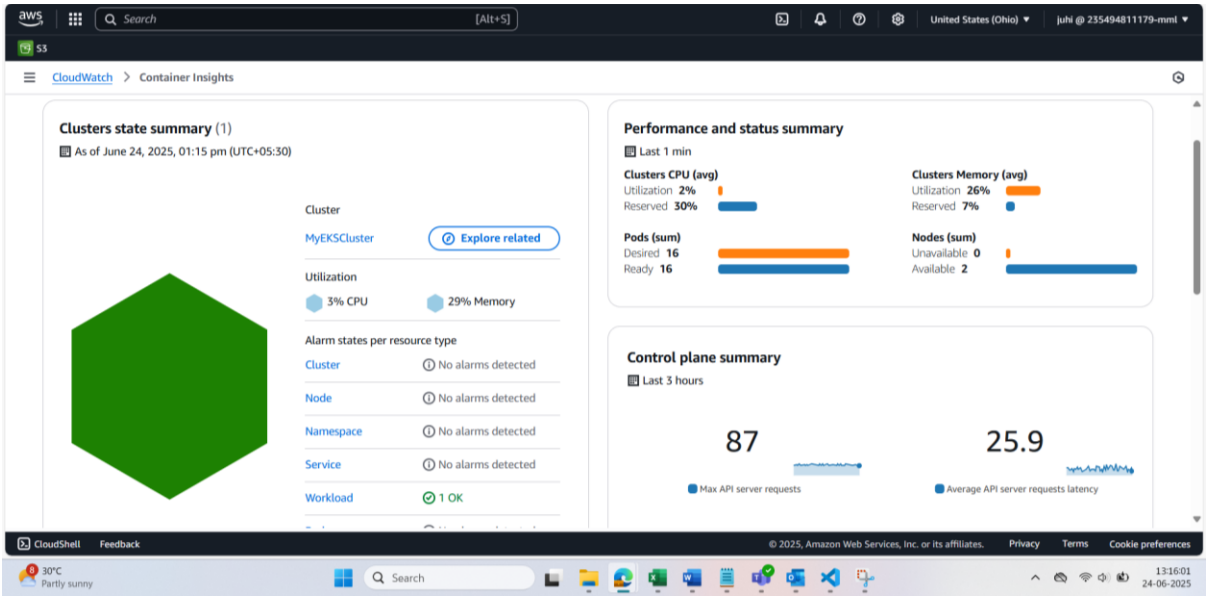
5. Watch for Pod Readiness
kubectl get pods -n amazon-cloudwatch -w

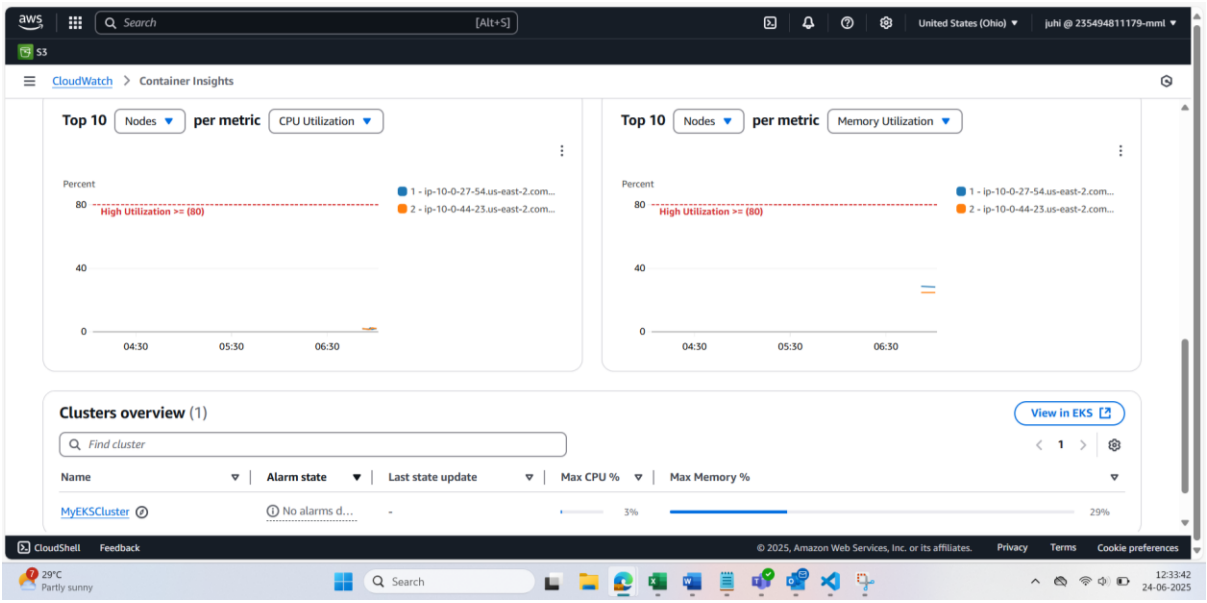
aws iam attach-role-policy \
  --role-name EKSNodeGroupRole-v2 \
  --policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy

pods

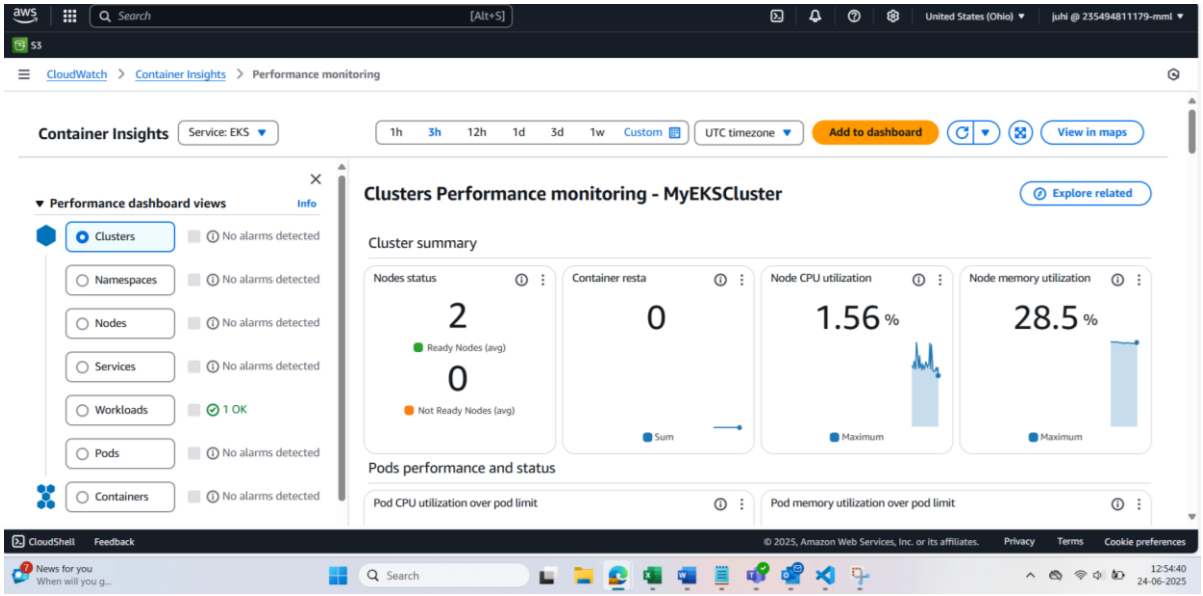
```
root@ip-10-0-2-225:/home/ubuntu# kubectl get pods -n amazon-cloudwatch
NAME                                                           READY   STATUS    RESTARTS   AGE
amazon-cloudwatch-observability-controller-manager-666d994jpspc   1/1     Running   0          20m
cloudwatch-agent-69z7k                                         1/1     Running   0          20m
cloudwatch-agent-gq2sx                                         1/1     Running   0          20m
fluent-bit-2k65f                                               1/1     Running   0          20m
fluent-bit-9xws7                                               1/1     Running   0          20m
root@ip-10-0-2-225:/home/ubuntu#
```

Database – container insights



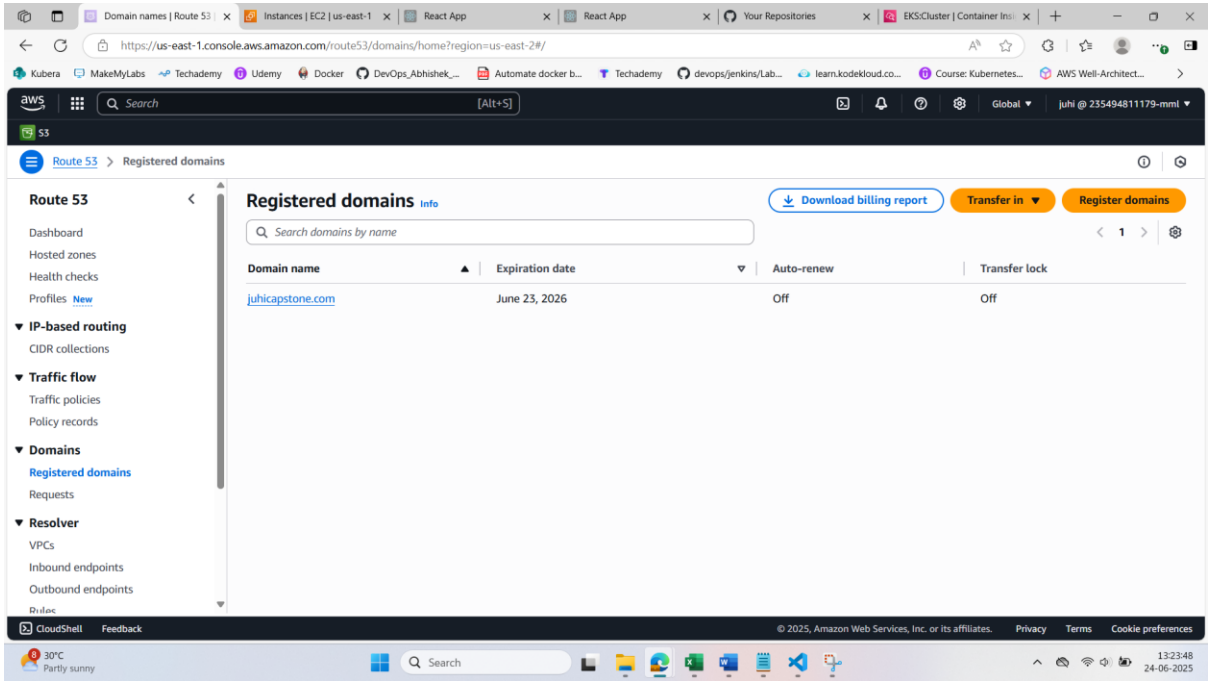


Performances

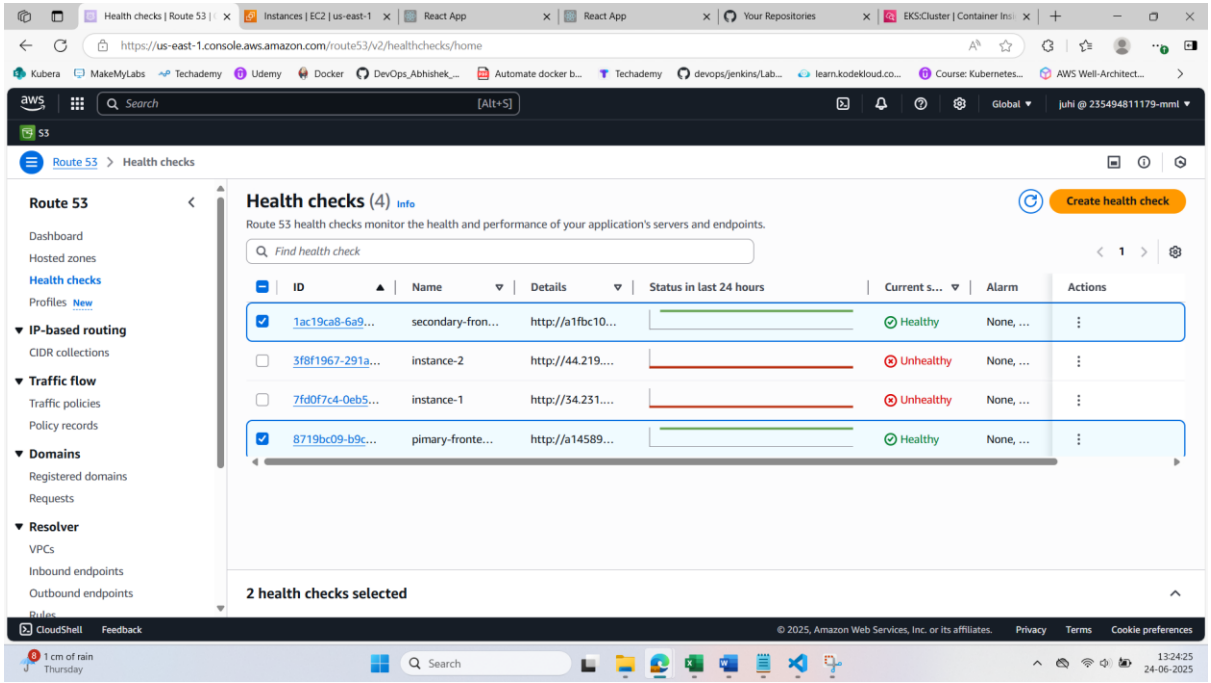


Route 53 for disaster recovery and high availability

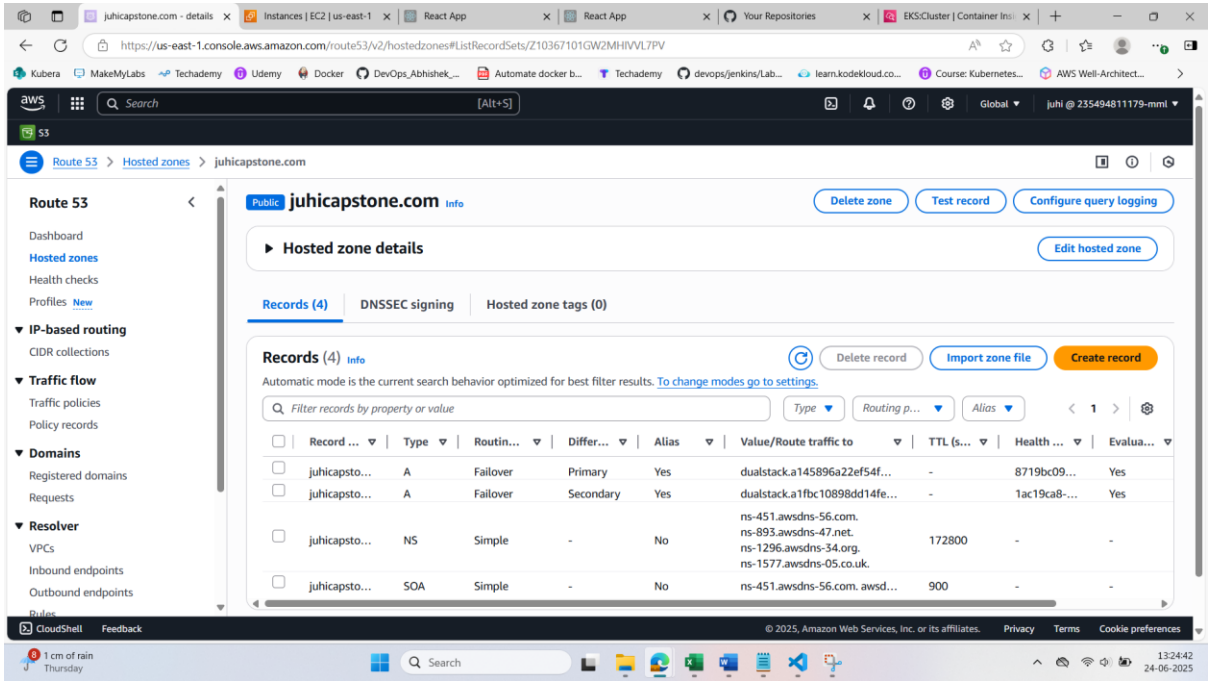
- Domain juhicapstone.com



Health checks for both frontend load balancer

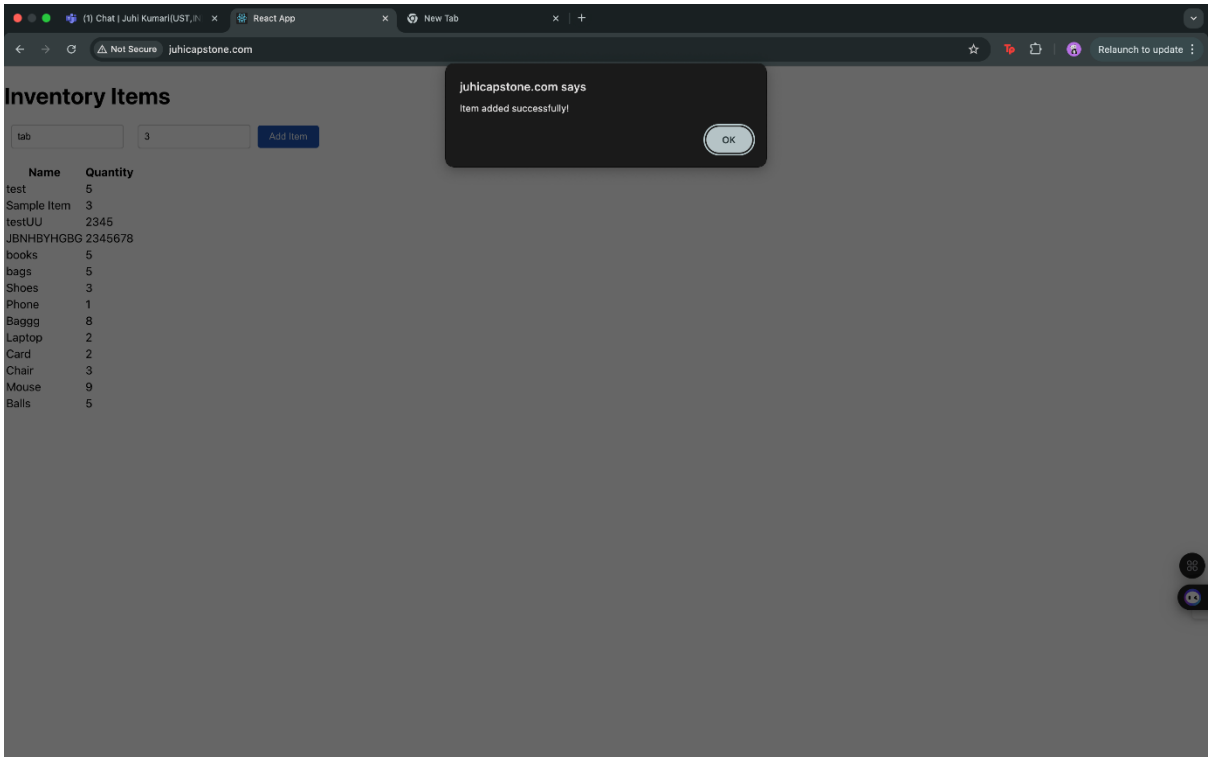


Records added for both primary and secondary



Application is only we can access

Added items



Items visible