Final Capstone Project

Presented by

Satyam

DevOps Training (Batch-1)

Under the Guidance of

Mr. Murali Mohan

Index

Step 1: Jira Workflow

- Each candidate must assign themselves tasks in Jira.
- Track task progress using Jira board (To Do → In Progress → Done).
- Export the board as part of the final submission.

Step 2: GitHub Workflow

- Clone the GitHub repository: https://github.com/akshu20791/Book-My-Show/
- Create a feature branch and make changes.
- Push changes and raise a Pull Request (PR).
- Review peers' PRs and approve before merging into the main branch.
- Submit the final PR link as deliverable.

Step 3: Jenkins CI/CD Pipeline

Pipeline stages to implement:

1. Clean Workspace
2. Checkout Code from GitHub
3. SonarQube Analysis (Quality Gate)
4. Install Dependencies (NPM)
5. Trivy FS Scan (Optional)
6. OWASP Dependency Check (Optional)
7. Docker Build & Push to DockerHub (via Jenkins)
8. Deploy to Docker Container
9. (Optional) Deploy to Kubernetes (EKS)
10. Email Notification on build result

Step 4: Docker Deployment

- Candidates must write their own Dockerfile to build the BMS app image.
- Build Docker image and push to DockerHub via Jenkins.
- Run container locally and validate accessibility on port 3000.

Step 5: Kubernetes Deployment (EKS)

- Candidates must write their own Kubernetes manifests (`deployment.yaml`, `service.yaml`).

- Deploy the application on EKS cluster.
- Expose service using NodePort or LoadBalancer.
- Validate deployment using 'kubectl get pods' and 'kubectl get svc'.

Step 6: Monitoring & Observability

- Install Prometheus and Node Exporter to collect metrics.
- Integrate Jenkins metrics into Prometheus.
- Install Grafana, configure Prometheus as a data source.
- Add dashboards for Node health and Jenkins performance.
- Submit Grafana screenshots in deliverables.

Introduction

This capstone project simulates a real-world DevOps workflow for the Book-My-Show (BMS) application, covering the complete CI/CD lifecycle from code collaboration to deployment and monitoring. The project demonstrates how modern organizations implement automation, security, and observability into their software delivery process.

The workflow begins with project management in Jira, where tasks are created, assigned, and tracked using an Agile board. Version control and collaboration are handled through GitHub, where feature branches, pull requests, and code reviews ensure code quality.

A Jenkins pipeline orchestrates the CI/CD process, which includes:

- Code checkout and quality checks with SonarQube.
- Installing dependencies and performing security scans (Trivy, OWASP).
- Building and pushing Docker images to DockerHub.
- Deploying the application to a Docker container and optionally to an EKS cluster with Kubernetes manifests.

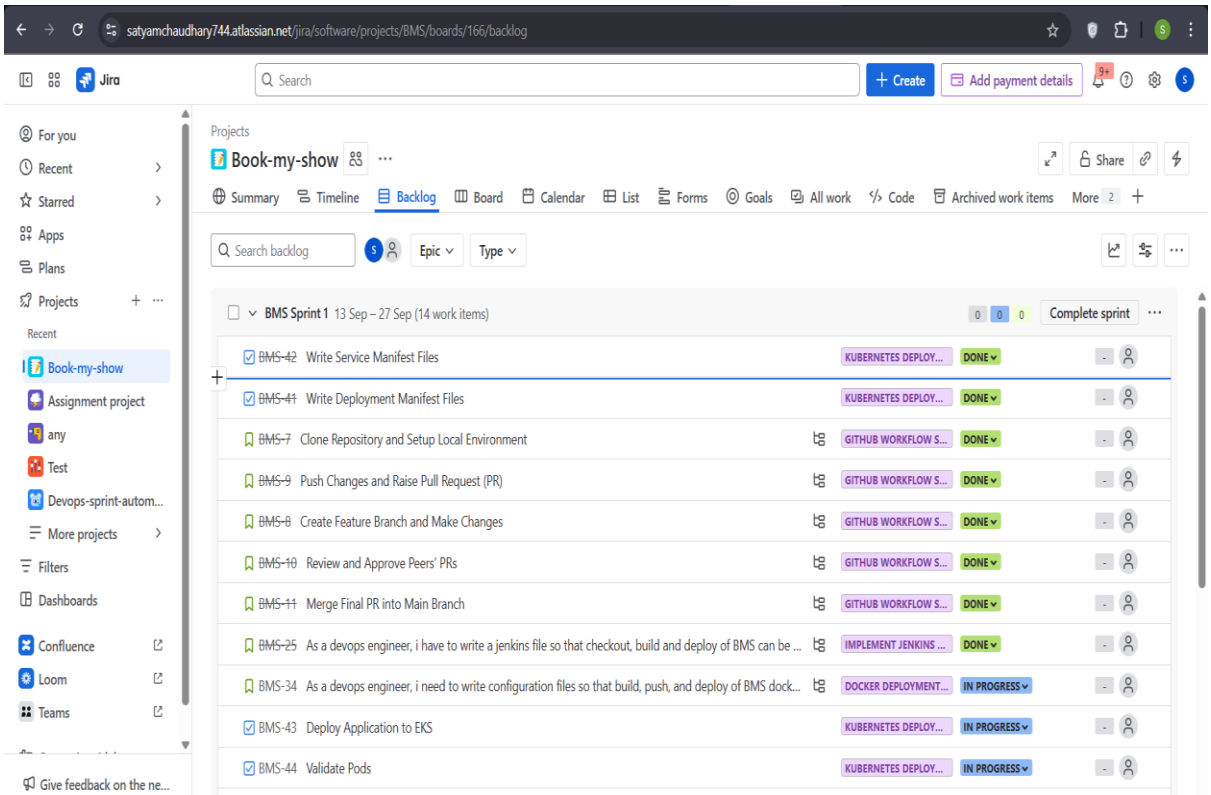
For deployment and scalability, Docker and Kubernetes ensure consistent runtime environments. Finally, Prometheus and Grafana provide monitoring and observability, enabling visibility into system performance, Jenkins pipelines, and infrastructure health.

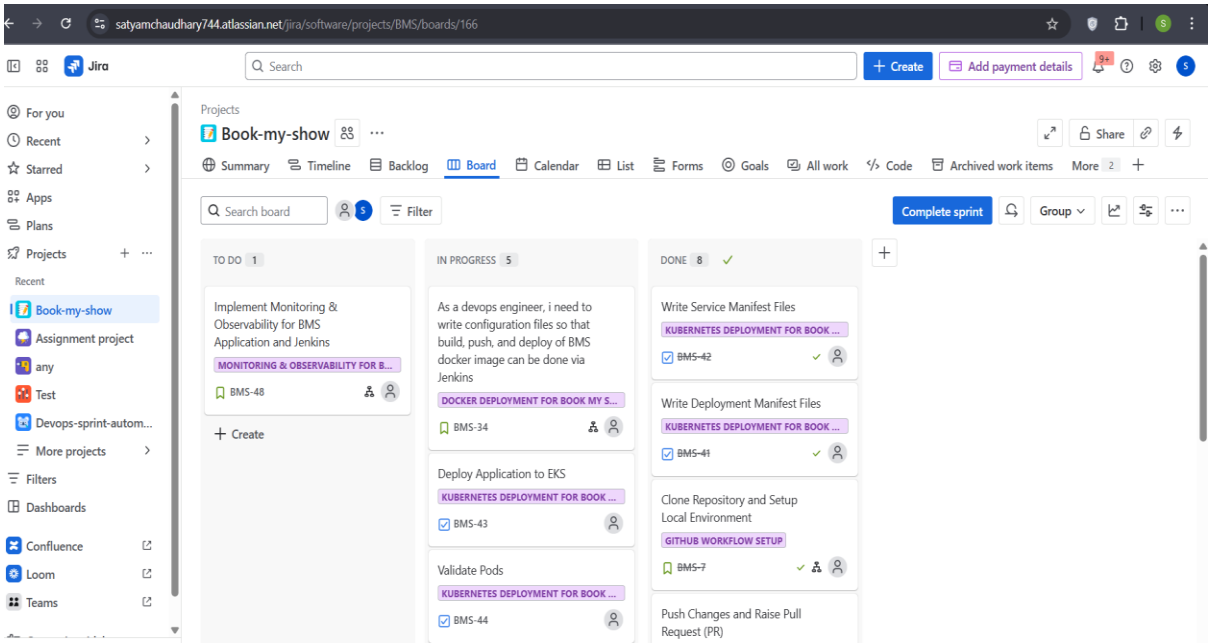
This end-to-end project integrates collaboration, automation, security, deployment, and monitoring, reflecting industry-standard DevOps practices.

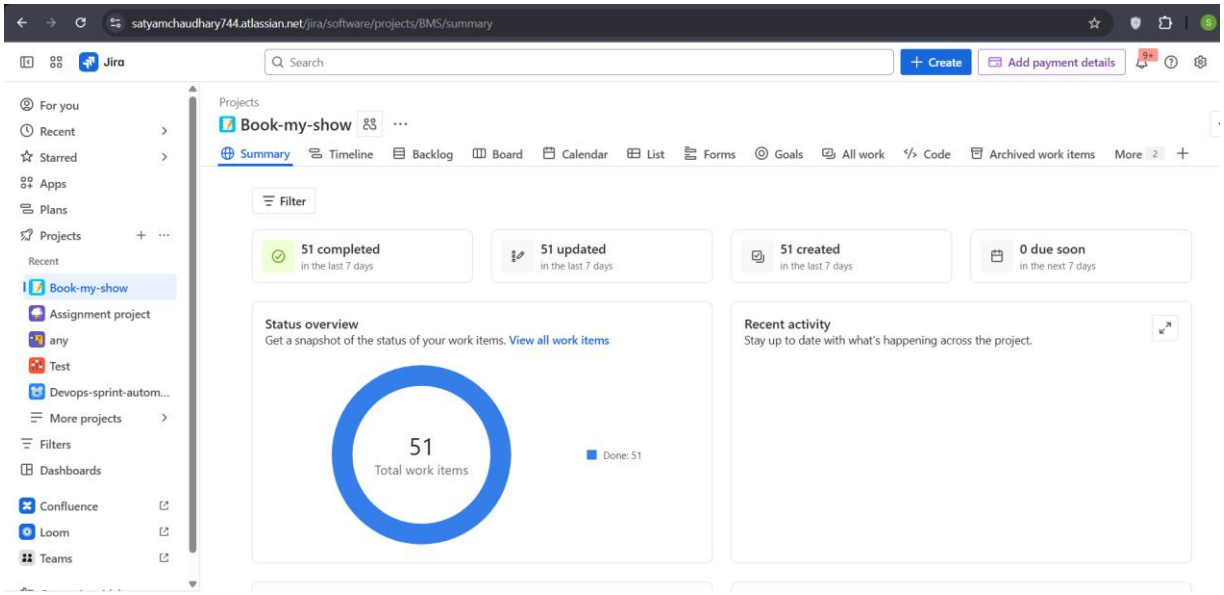
Solution

Step 1

1. Jira was used to manage and track project tasks using an Agile board.
2. All activities required for the Book-My-Show (BMS) DevOps pipeline were broken down into tasks and assigned to myself.
3. The workflow followed the standard Jira board structure.

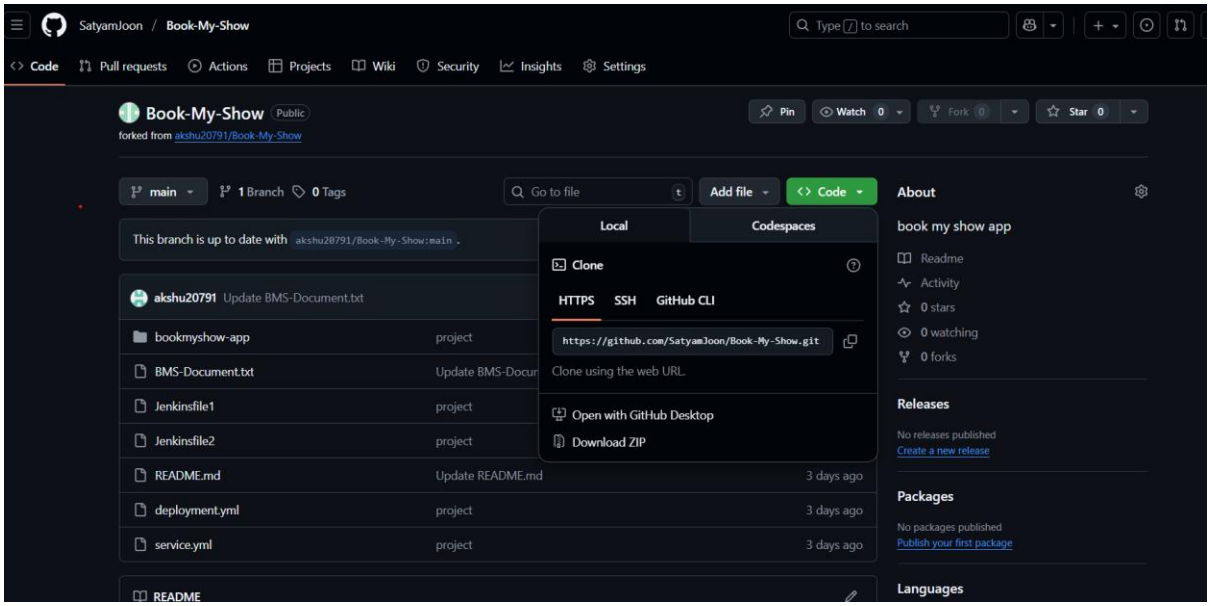






Step 2

1. Fork git repository



Clone repository https://github.com/SatyamJoon/Book-My-Show.git

```
Command Prompt

Microsoft Windows [Version 10.0.22631.5768]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>git clone https://github.com/SatyamJoon/Book-My-Show.git
fatal: destination path 'Book-My-Show' already exists and is not an empty directory.

C:\Users\HP>git clone https://github.com/SatyamJoon/Book-My-Show.git
Cloning into 'Book-My-Show'...
remote: Enumerating objects: 136, done.
remote: Counting objects: 100% (136/136), done.
remote: Compressing objects: 100% (114/114), done.
remote: Total 136 (delta 19), reused 129 (delta 16), pack-reused 0 (from 0)Receiving objects:  89% (122/136)
Receiving objects: 100% (136/136), 1.52 MiB | 3.87 MiB/s, done.
Resolving deltas: 100% (19/19), done.

C:\Users\HP>
```

Editing Dockerfile

```
bookmyshow-app > Dockerfile > ...
FROM node:18

# Set working directory
WORKDIR /app

# Copy package.json and package-lock.json
COPY package.json package-lock.json ./

# Force install a compatible PostCSS version to fix the issue
RUN npm install postcss@8.4.21 postcss-safe-parser@6.0.0 --legacy-peer-deps

#adding chkidar here
RUN npm i chokidar

# Install dependencies
RUN npm install

# Copy the entire project
COPY . .

# Expose port 3000
EXPOSE 3000

# Set environment variable to prevent OpenSSL errors
ENV NODE_OPTIONS=--openssl-legacy-provider
ENV PORT=3000
```

Creating docker-compose.yaml file

```
bookmyshow-app > docker-compose.yaml > {} networks > {} bms-net > driver
docker-compose.yml - The Compose specification establishes a standard for the definition of multi-container platform-agnostic a
version: "3.9"

services:
  bms-app:
    image: satyam744/bms:latest  # Use your Docker Hub image
    container_name: bms-app
    restart: unless-stopped
    ports:
      - "3010:3000"      # Maps container port to host port for local access
    networks:
      - bms-net

networks:
  bms-net:
    driver: bridge
```

Now, I have open terminal move to git bash and use below commands

>> git branch feature

>> git checkout feature

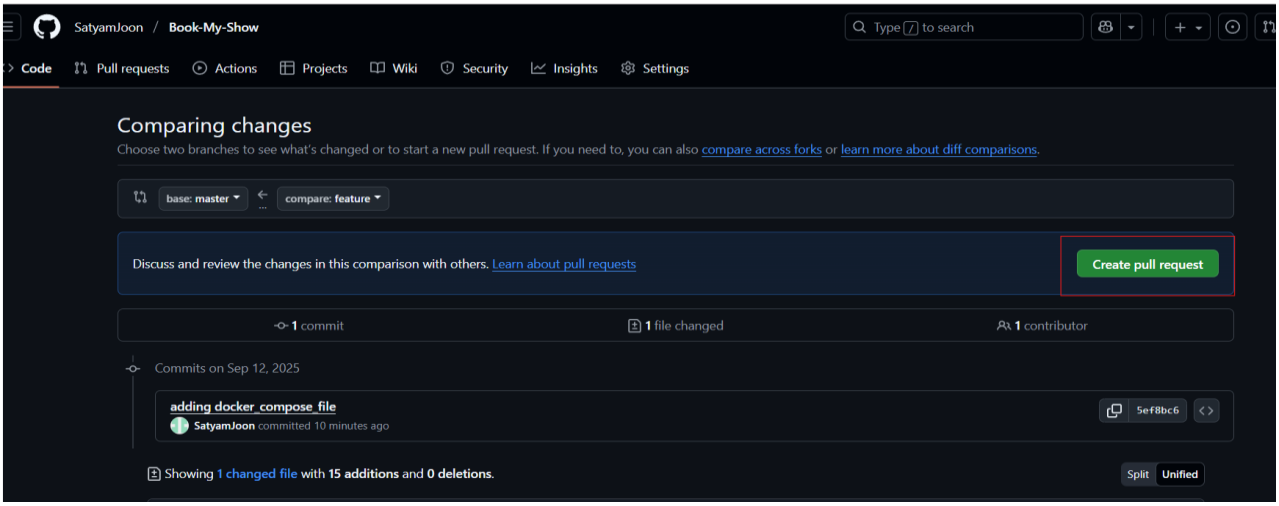
>> git add .

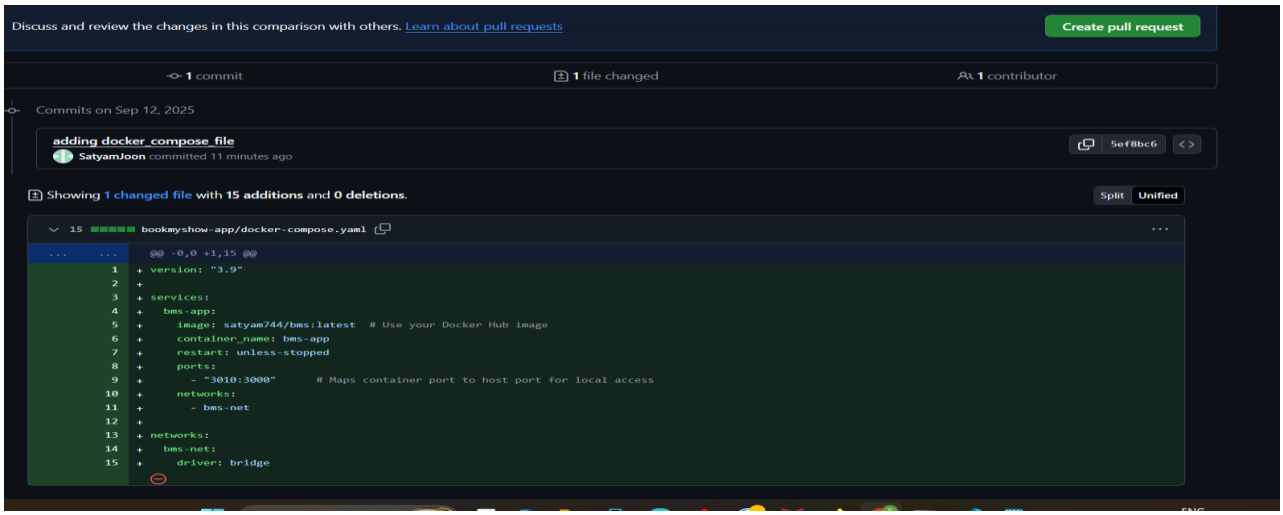
>> git status

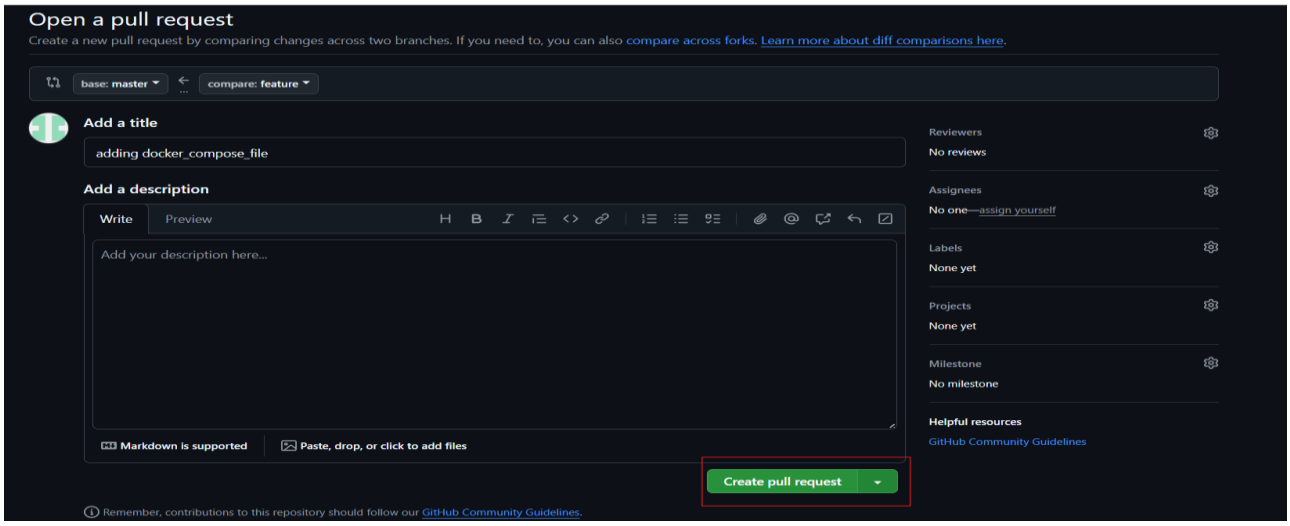
>> git commit -m” docker-compose file ”

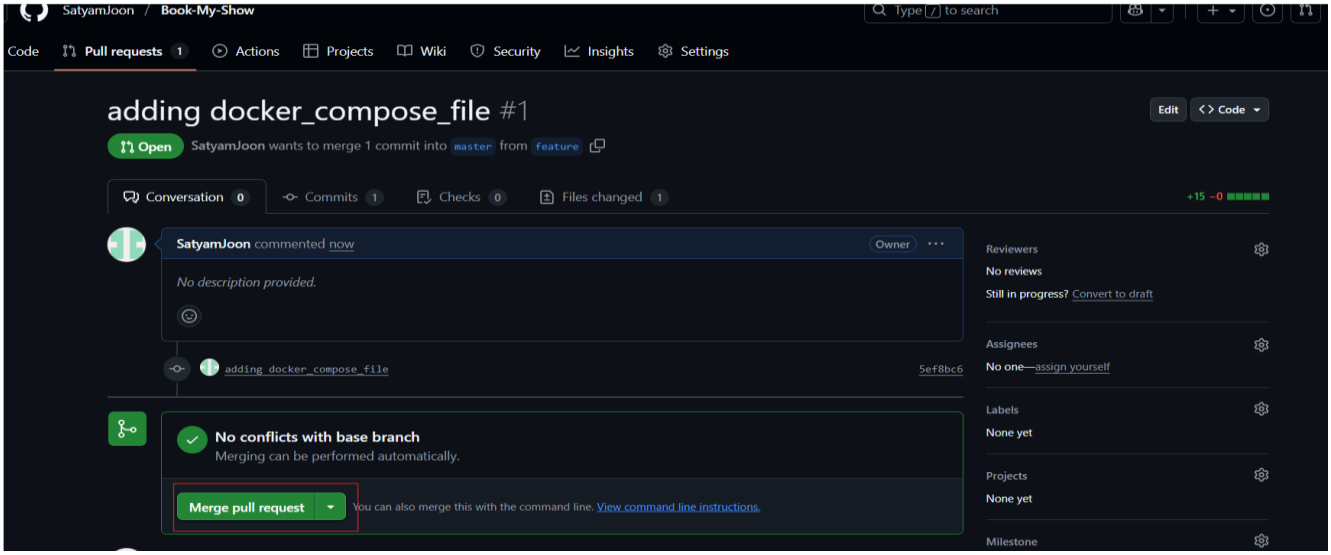
>> git push -u origin feature

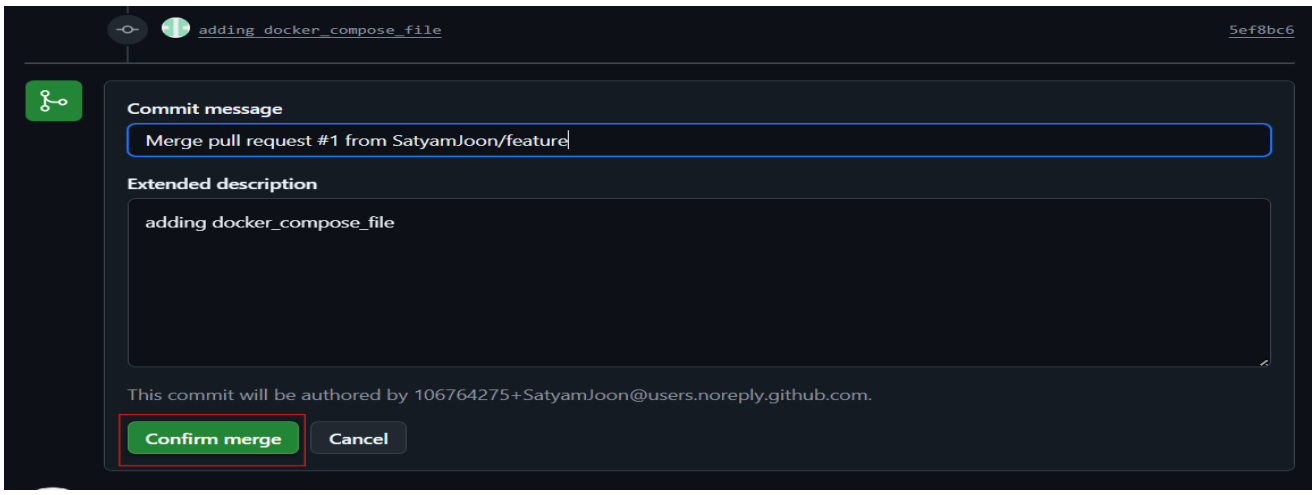
Now go to github and click on compare and pull request

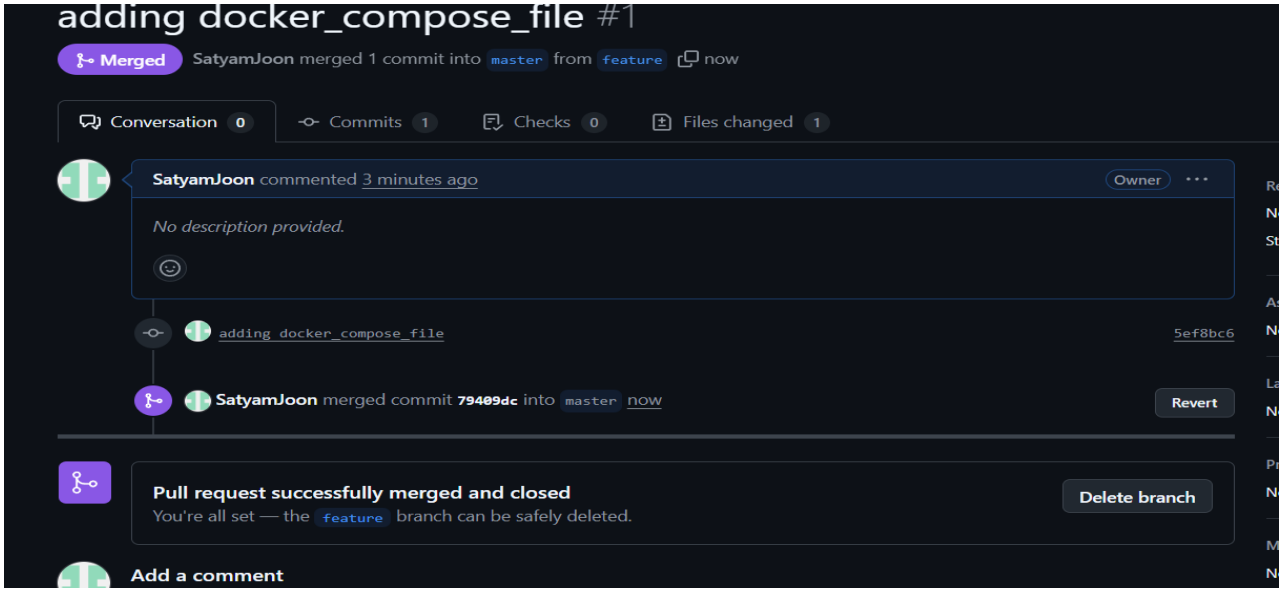


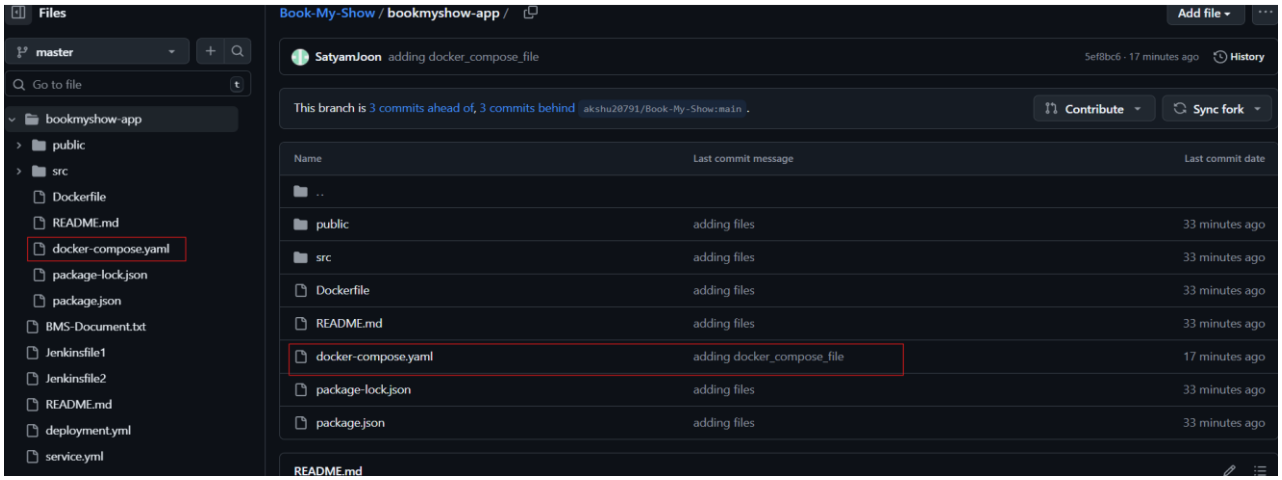



Open a pull request
Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks. Learn more about diff comparisons here.
base: master
compare: feature
Add a title
adding docker_compose_file
Add a description
Write
Preview
Add your description here...
Markdown is supported
Paste, drop, or click to add files
Create pull request
Remember, contributions to this repository should follow our GitHub Community Guidelines.
Reviewers
No reviews
Assignees
No one—assign yourself
Labels
None yet
Projects
None yet
Milestone
No milestone
Helpful resources
GitHub Community Guidelines


SatyamJoon / Book-My-Show
Type / to search
Code
Pull requests 1
Actions
Projects
Wiki
Security
Insights
Settings
adding docker_compose_file #1
Edit
Code
Open
SatyamJoon wants to merge 1 commit into master from feature
Conversation 0
Commits 1
Checks 0
Files changed 1
+15 −0
SatyamJoon commented now
Owner
No description provided.
adding docker_compose_file
5ef8bc6
No conflicts with base branch
Merging can be performed automatically.
Merge pull request
You can also merge this with the command line. View command line instructions.
Reviewers
No reviews
Still in progress? Convert to draft
Assignees
No one—assign yourself
Labels
None yet
Projects
None yet
Milestone


adding docker_compose_file
5ef8bc6
Commit message
Merge pull request #1 from SatyamJoon/feature
Extended description
adding docker_compose_file
This commit will be authored by 106764275+SatyamJoon@users.noreply.github.com.
Confirm merge
Cancel

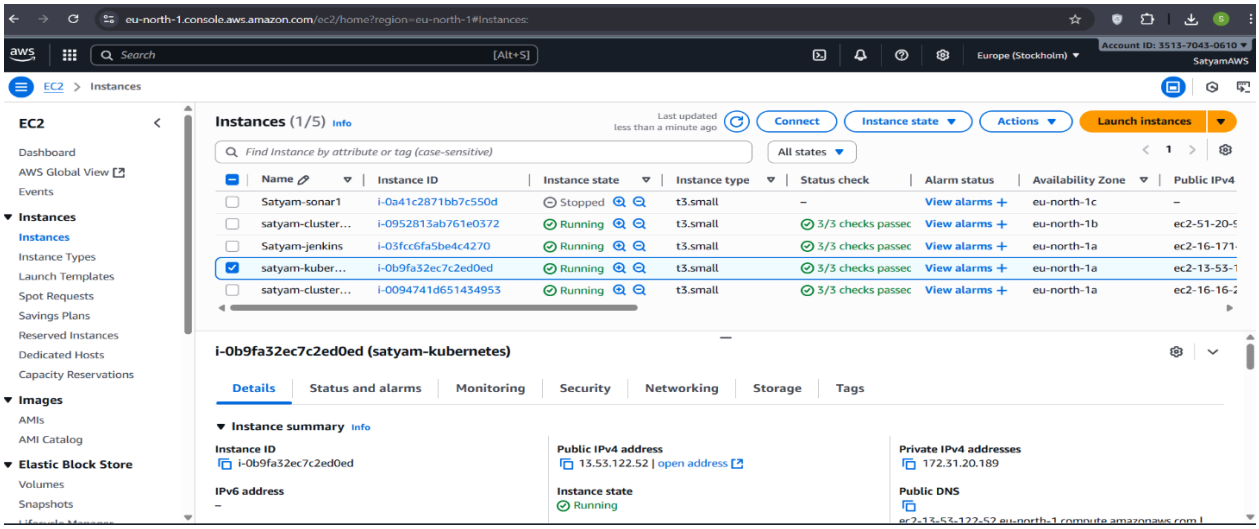




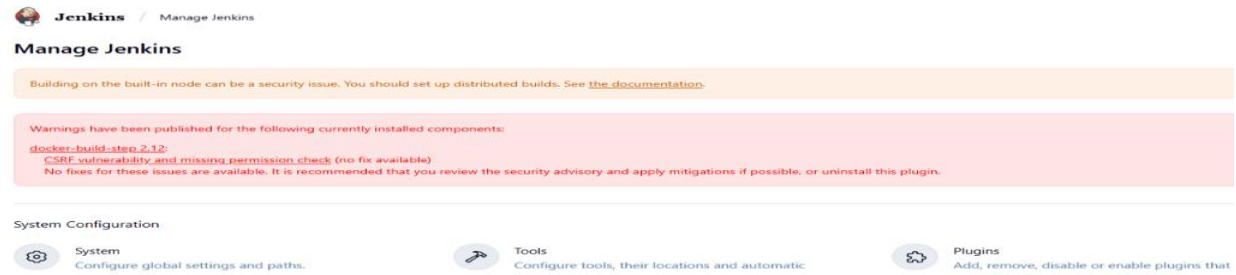
Link for the pull request → https://github.com/SatyamJoon/Book-My-Show/pull/1

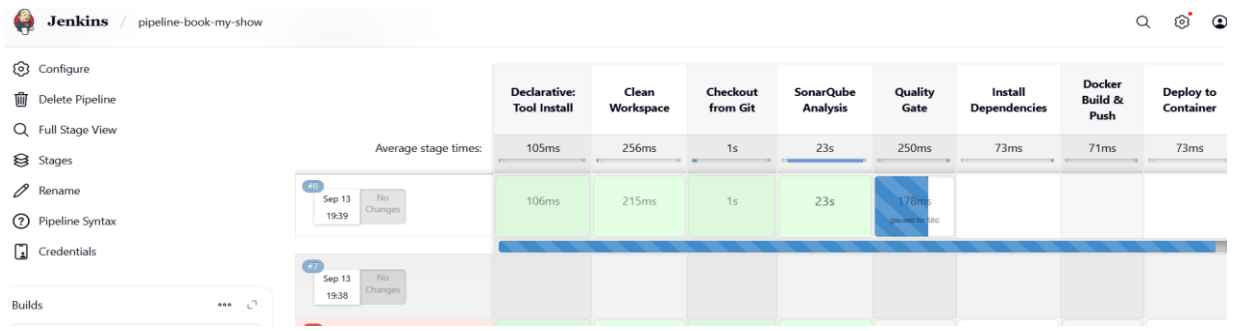
Step 3

1. Launch the 2 ec2 server for t2.medium and storage upto 20gb. One for Jenkins and one for SonarQube.
2. Install Jenkins, Java and Docker on Jenkins server, and Java, Docker and Sonar on sonar server.
3. For launching SonarQube , I use command and https://<public ip>:9000 .
4. >> docker run -d –name sonarqube -p 9000:9000 sonarqube:community
5. In Jenkins plugin, I have installed plugins for Docker, Pipeline and SonarQube.
6. Configure the Jenkins along with credentials of Docker-hub and SonarQube.
7. Writing the pipeline script and start build.

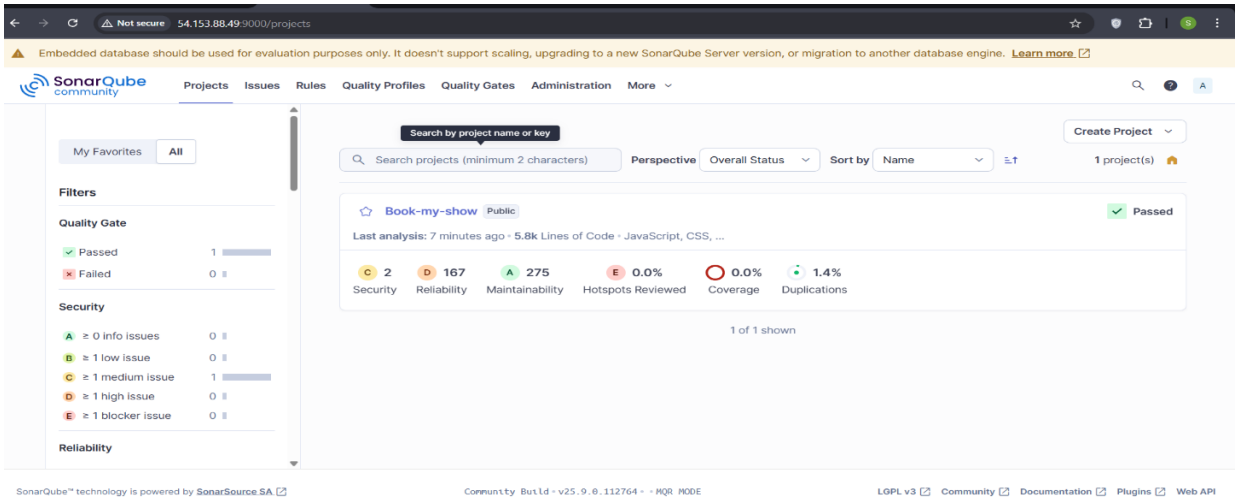


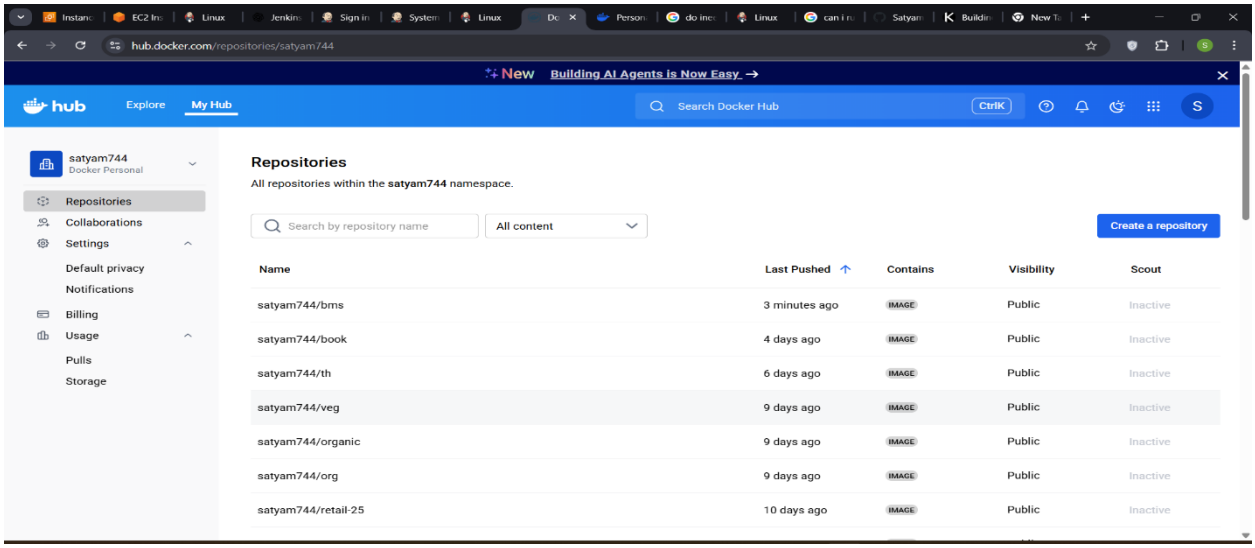

```
OpenJDK 64-Bit Server VM (build 21.0.8+9-Ubuntu-0ubuntu124.04.1, mixed mode, sharing)
root@ip-172-31-1-240:/home/ubuntu# sudo systemctl enable jenkins
Synchronizing state of jenkins.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable jenkins
root@ip-172-31-1-240:/home/ubuntu# sudo systemctl start jenkins
root@ip-172-31-1-240:/home/ubuntu# sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-09-13 05:28:54 UTC; 14s ago
   Main PID: 5545 (java)
      Tasks: 44 (limit: 2329)
     Memory: 393.6M (peak: 399.0M)
        CPU: 15.075s
     CGroup: /system.slice/jenkins.service
             └─5545 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/

Sep 13 05:28:48 ip-172-31-1-240 jenkins[5545]: ceae666609254a909dfb00573e16c1f8
Sep 13 05:28:48 ip-172-31-1-240 jenkins[5545]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Sep 13 05:28:48 ip-172-31-1-240 jenkins[5545]: *************************************************************
Sep 13 05:28:48 ip-172-31-1-240 jenkins[5545]: *************************************************************
Sep 13 05:28:48 ip-172-31-1-240 jenkins[5545]: *************************************************************
Sep 13 05:28:54 ip-172-31-1-240 jenkins[5545]: 2025-09-13 05:28:54.355+0000 [id=32]	INFO	jenkins.InitReacto
Sep 13 05:28:54 ip-172-31-1-240 jenkins[5545]: 2025-09-13 05:28:54.385+0000 [id=24]	INFO	hudson.lifecycle.I
Sep 13 05:28:54 ip-172-31-1-240 systemd[1]: Started jenkins.service - Jenkins Continuous Integration Server.
Sep 13 05:28:55 ip-172-31-1-240 jenkins[5545]: 2025-09-13 05:28:55.129+0000 [id=48]	INFO	h.m.DownloadServic
```






clean workspace
git checkout
sonarqube analysis
install dependencies
docker build and push
deploy to container
Declarative: Post Actions
Average stage times:
(full run time: ~4min 43s)
1s
728ms
11s
39s
59s
201ms
1s
#5 Sep 14 14:35 1 commit
3s
618ms
11s
49s
3min 36s
564ms
1s
Success
Logs
#4 Sep 14 14:22 1 commit
3s
10s
48s
18s failed
71ms failed
1s
#3 Sep 14 14:13 No Changes
101ms
746ms
9s
58s
2s failed
100ms failed
1s
#2 Sep 14 14:06 No Changes
77ms
940ms
14s
399ms failed
104ms failed
69ms failed
1s
#1 Sep 14 14:06 No Changes

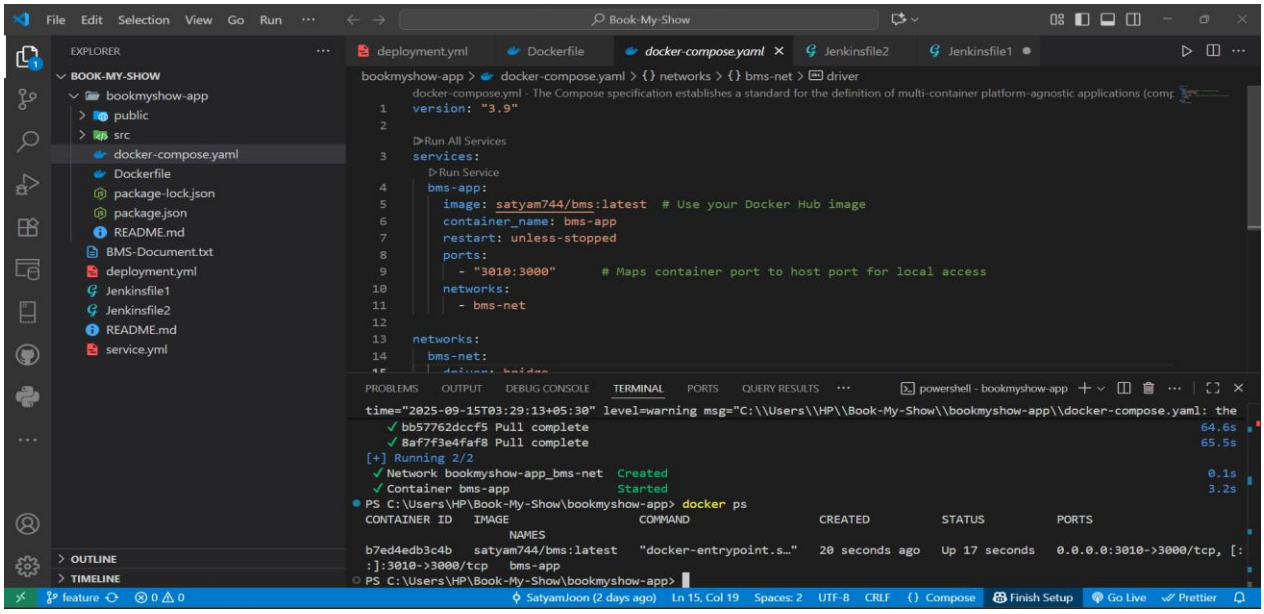

Not secure 54.153.88.49:9000/projects
Embedded database should be used for evaluation purposes only. It doesn't support scaling, upgrading to a new SonarQube Server version, or migration to another database engine. Learn more
SonarQube community
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
My Favorites
All
Filters
Quality Gate
Passed 1
Failed 0
Security
≥ 0 info issues 0
≥ 1 low issue 0
≥ 1 medium issue 1
≥ 1 high issue 0
≥ 1 blocker issue 0
Reliability
Search by project name or key
Search projects (minimum 2 characters)
Perspective
Overall Status
Sort by
Name
Create Project
1 project(s)
Book-my-show Public
Passed
Last analysis: 7 minutes ago · 5.8k Lines of Code · JavaScript, CSS, ...
2 Security
167 Reliability
275 Maintainability
0.0% Hotspots Reviewed
0.0% Coverage
1.4% Duplications
1 of 1 shown
SonarQube™ technology is powered by SonarSource SA
Community Build · v25.9.0.112764 · MQR MODE
LGPL v3
Community
Documentation
Plugins
Web API

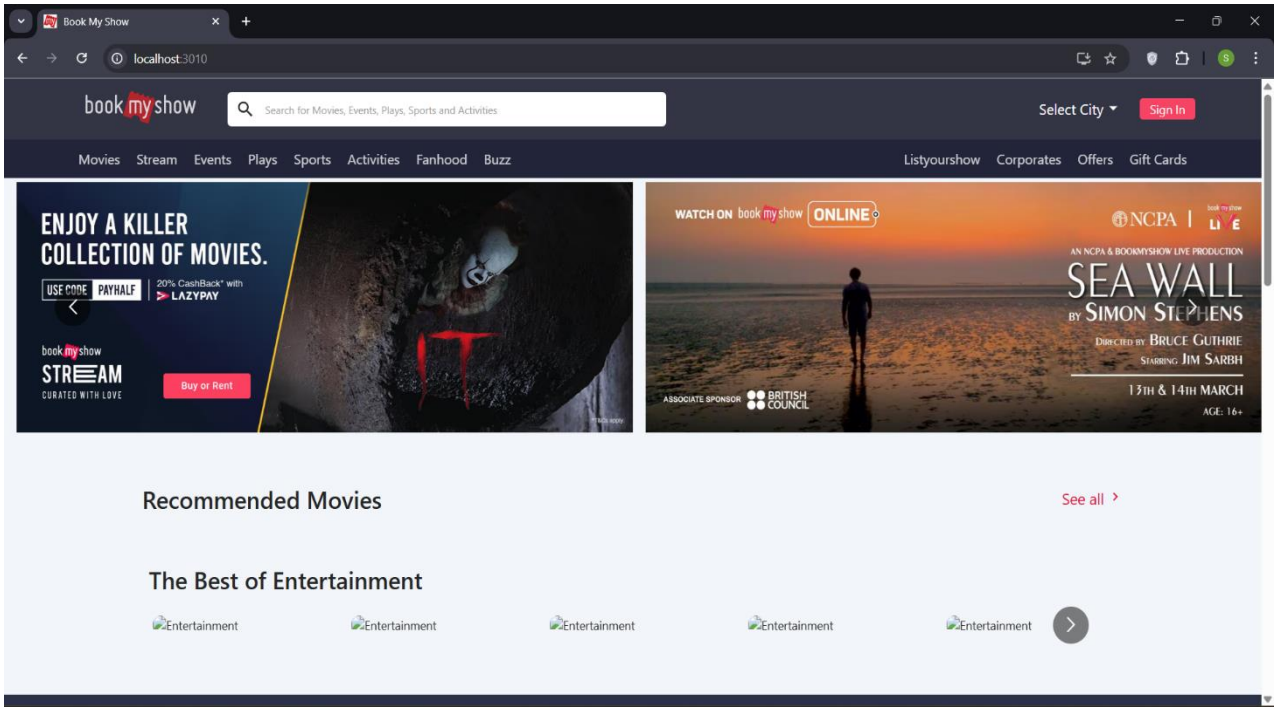

hub.docker.com/repositories/satyam744
New Building AI Agents is Now Easy
hub
Explore
My Hub
Search Docker Hub
Ctrl K
satyam744
Docker Personal
Repositories
Collaborations
Settings
Default privacy
Notifications
Billing
Usage
Pulls
Storage
Repositories
All repositories within the satyam744 namespace.
Search by repository name
All content
Create a repository
Name
Last Pushed
Contains
Visibility
Scout
satyam744/bms 3 minutes ago IMAGE Public Inactive
satyam744/book 4 days ago IMAGE Public Inactive
satyam744/th 6 days ago IMAGE Public Inactive
satyam744/veg 9 days ago IMAGE Public Inactive
satyam744/organic 9 days ago IMAGE Public Inactive
satyam744/org 9 days ago IMAGE Public Inactive
satyam744/retail-25 10 days ago IMAGE Public Inactive

Stage 4

1. First I write the Dockerfile for building the docker image and pushed the image to the Docker-registry via Jenkins.
2. Using Docker desktop, run the container locally.

   >> docker-compose up -d





Stage 5

1. First I have create a ec2 instance then install ekctl, kubectl and awscli.

2. Setup IAM user and role for cluster.
3. Attach the IAM role with the ec2 instance and configure the credentials,then using ekctl create command I have created the cluster.

```
>> eksctl create cluster \

 --name satyam-cluster \

 --region eu-north-1 \

 --nodegroup-name satyam-nodes \

 --node-type t3.small \

 --zones eu-north-1a,eu-north-1b \

 --nodes 2
```

4. First create namespace as satyam-ns then create the deployment.yaml and service.yaml file for deploying the application on EKS cluster using below commands.

```
>> kubectl create namespace satyam-ns

>> kubectl apply -f deployment.yaml -n satyam-ns

>> kubectl apply -f service.yaml -n satyam-ns

>> kubectl get pods -n satyam-ns

>> kubectl get svc -n satyam-ns
```

5. After these commands my command and service for application start running and I getthe external ip address for access the application.
6. At last, I delete the cluster using below command.

```
>> eksctl delete cluster \

 --name satyam-cluster \

 --region eu-north-1
```

eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#Instances:

Account ID: 3513-7043-0610 SatyamAWS

EC2 > Instances

Instances (1/7) Info

Last updated 2 minutes ago | Connect | Instance state | Actions | Launch instances

Find Instance by attribute or tag (case-sensitive) | All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
Satyam-jenkins	i-03fcc6fa5be4c4270	Stopped	t3.small	–	View alarms +	eu-north-1a	–
satyam-kuber...	i-0b9fa32ec7c2ed0ed	Running	t3.small	3/3 checks passed	View alarms +	eu-north-1a	ec2-13-61-1
satyam-cluster...	i-0094741d651434953	Terminated	t3.small	–	View alarms +	eu-north-1a	–

i-0b9fa32ec7c2ed0ed (satyam-kubernetes)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

Instance summary Info

Instance ID: i-0b9fa32ec7c2ed0ed

Public IPv4 address: 13.61.143.164 | open address

Private IPv4 addresses: 172.31.20.189

IPv6 address: –

Instance state: Running

Public DNS: ec2-13-61-143-164.eu-north-1.compute.amazonaws.com | open address

Hostname type

Private IP DNS name (IPv4 only)

eu-north-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0b9fa32ec7c2ed0ed&osUser=ubuntu&region=eu-n...

Account ID: 3513-7043-0610 SatyamAWS

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: bms-app
  namespace: satyam-ns
  labels:
    app: bms
spec:
  replicas: 1
  selector:
    matchLabels:
      app: bms
  template:
    metadata:
      labels:
        app: bms
    spec:
      containers:
      - name: bms-container
        image: satyam744/bms:latest  # Replace with your Docker image
        ports:
        - containerPort: 3000
```

i-0b9fa32ec7c2ed0ed (satyam-kubernetes)

PublicIPs: 13.61.143.164 PrivateIPs: 172.31.20.189

eu-north-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0b9fa32ec7c2ed0ed&osUser=ubuntu&region=eu-n...

Account ID: 3513-7043-0610 ▾ SatyamAWS

```
apiVersion: v1
kind: Service
metadata:
  name: bms-service
  namespace: satyam-ns
  labels:
    app: bms
spec:
  type: LoadBalancer
  ports:
  - port: 80
    targetPort: 3000  # Replace with the port your app runs on
  selector:
    app: bms
~
~
~
~
~
~
~
~
~
~
~
~
"service.yaml" 14L, 242B                                                    5,22          All
```

i-0b9fa32ec7c2ed0ed (satyam-kubernetes)

PublicIPs: 13.61.143.164 PrivateIPs: 172.31.20.189

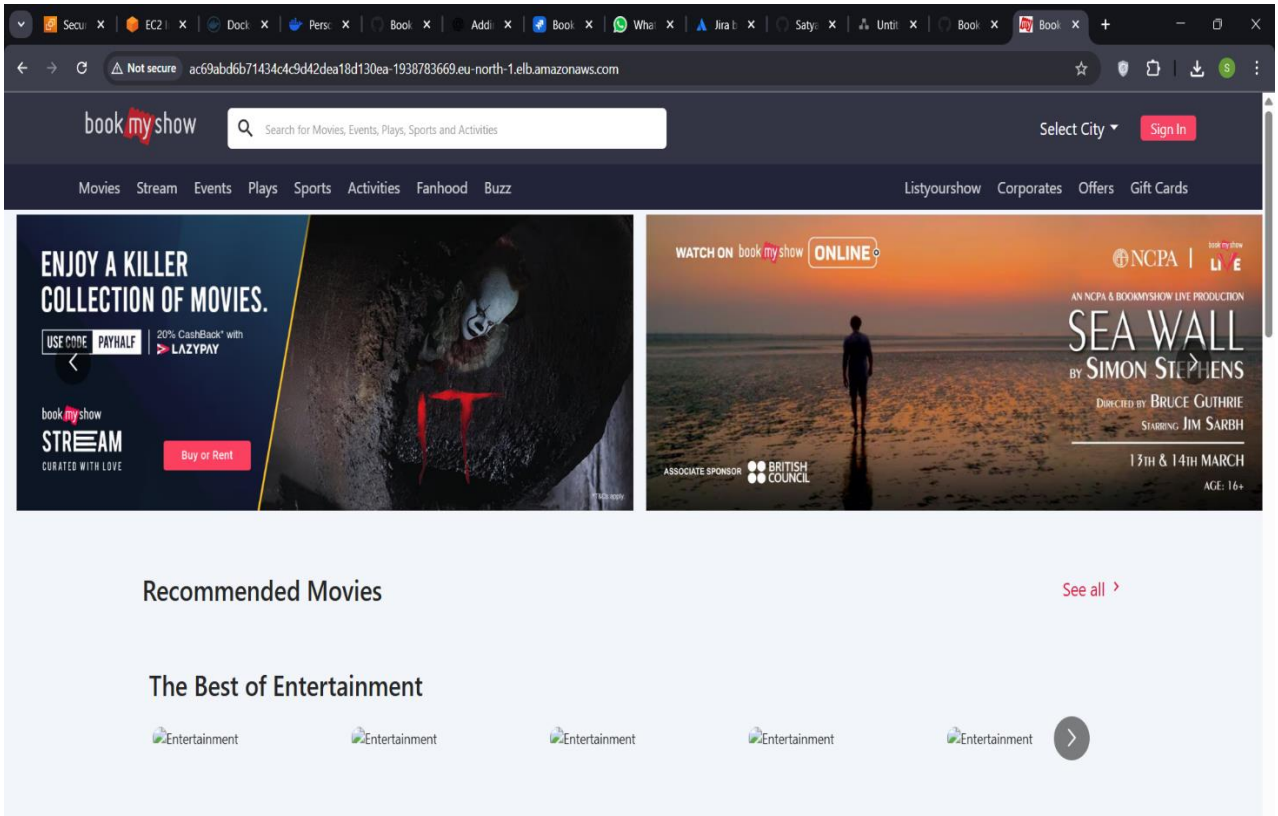
eu-north-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0b9fa32ec7c2ed0ed&osUser=ubuntu&region=eu-n...

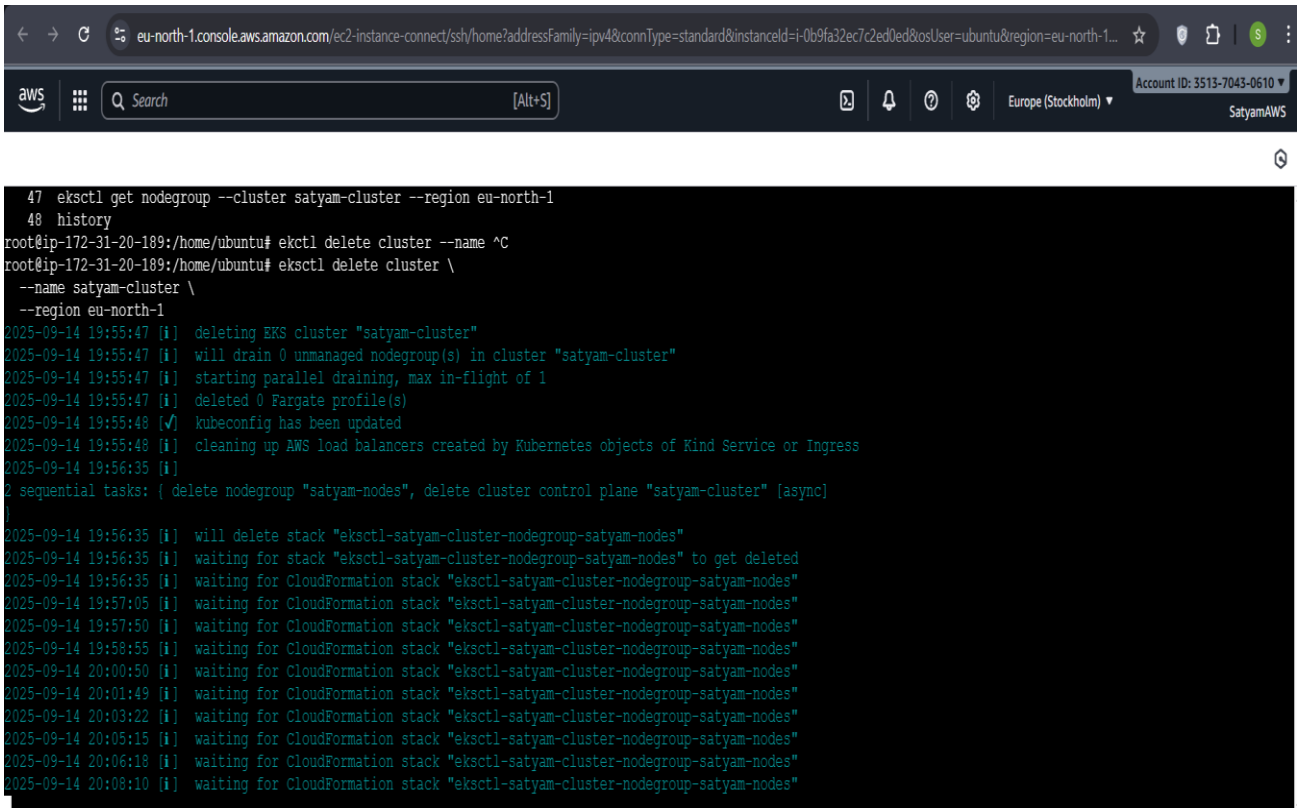
Account ID: 3513-7043-0610 ▾ SatyamAWS

```
root@ip-172-31-20-189:/home/ubuntu# cd satyam
root@ip-172-31-20-189:/home/ubuntu/satyam# ls
deployment.yaml  service.yaml
root@ip-172-31-20-189:/home/ubuntu/satyam# kubectl get pods
No resources found in default namespace.
root@ip-172-31-20-189:/home/ubuntu/satyam# kubectl get pods -n satyam-ns
No resources found in satyam-ns namespace.
root@ip-172-31-20-189:/home/ubuntu/satyam# vi deploymeny.yaml
root@ip-172-31-20-189:/home/ubuntu/satyam# vi service.yaml
root@ip-172-31-20-189:/home/ubuntu/satyam# rm -rf deploymeny.yaml
root@ip-172-31-20-189:/home/ubuntu/satyam# ls
deployment.yaml  service.yaml
root@ip-172-31-20-189:/home/ubuntu/satyam# vi deployment.yaml
root@ip-172-31-20-189:/home/ubuntu/satyam# ls
deployment.yaml  service.yaml
root@ip-172-31-20-189:/home/ubuntu/satyam# kubectl apply -f satyam-ns
error: the path "satyam-ns" does not exist
root@ip-172-31-20-189:/home/ubuntu/satyam# kubectl apply -f deployment.yaml -n satyam-ns
deployment.apps/bms-app created
root@ip-172-31-20-189:/home/ubuntu/satyam# kubectl apply -f service.yaml -n satyam-ns
service/bms-service created
root@ip-172-31-20-189:/home/ubuntu/satyam# kubectl get svc -n satyam-ns
NAME          TYPE           CLUSTER-IP     EXTERNAL-IP                                                              PORT(S)        AGE
bms-service   LoadBalancer   10.100.110.8   ac69abd6b71434c4c9d42dea18d130ea-1938783669.eu-north-1.elb.amazonaws.com   80:30840/TCP   22s
root@ip-172-31-20-189:/home/ubuntu/satyam# kubectl get pods -n satyam-ns
NAME                       READY   STATUS    RESTARTS   AGE
bms-app-5cf86654d9-9x2qs   1/1     Running   0          4m53s
root@ip-172-31-20-189:/home/ubuntu/satyam#
```

i-0b9fa32ec7c2ed0ed (satyam-kubernetes)

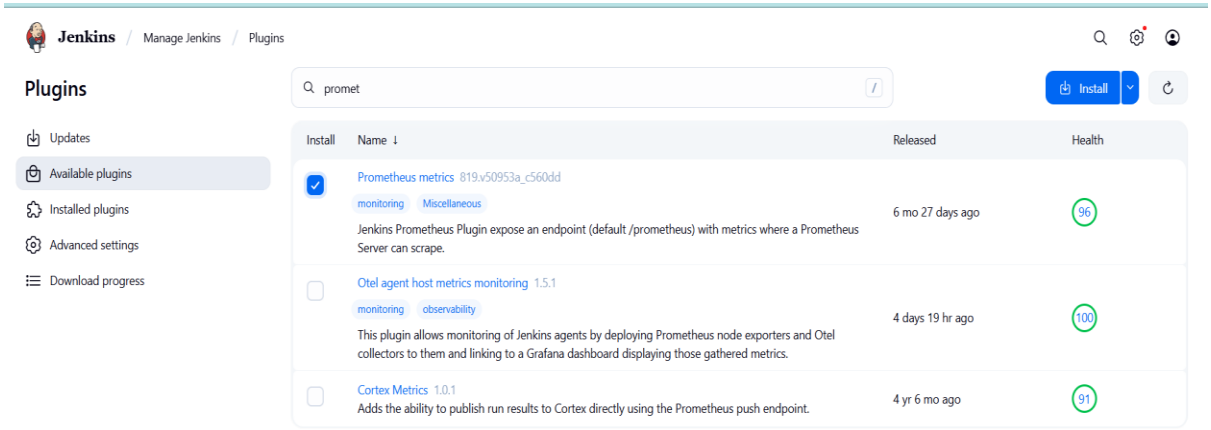
PublicIPs: 13.61.143.164 PrivateIPs: 172.31.20.189

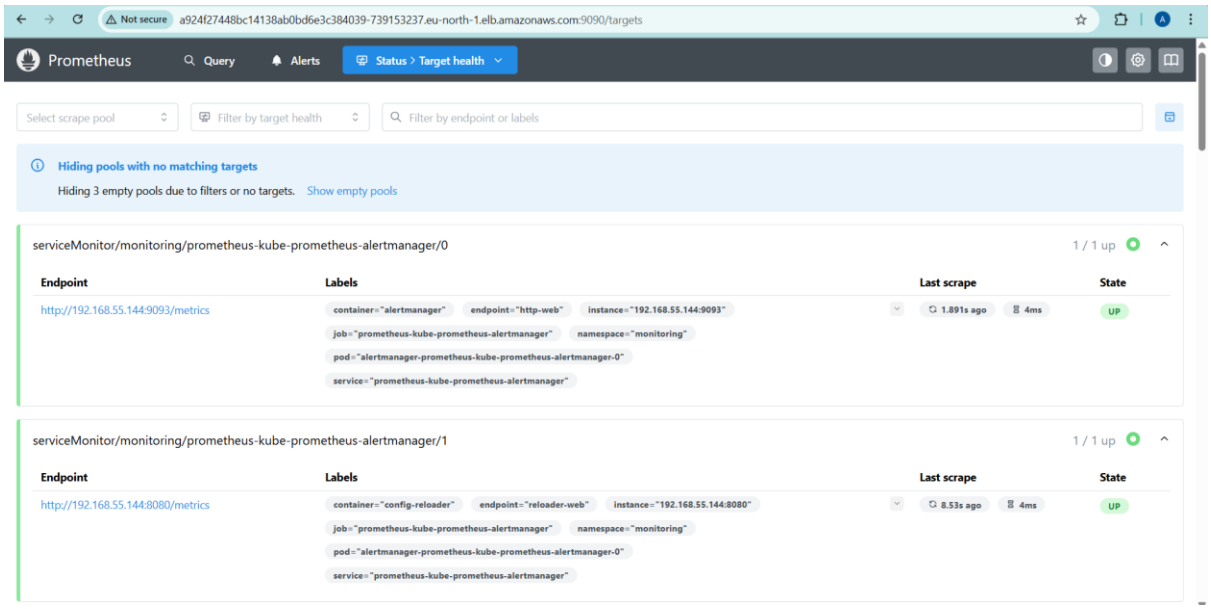

Not secure
ac69abd6b71434c4c9d42dea18d130ea-1938783669.eu-north-1.elb.amazonaws.com
book my show
Search for Movies, Events, Plays, Sports and Activities
Select City
Sign In
Movies Stream Events Plays Sports Activities Fanhood Buzz
Listyourshow Corporates Offers Gift Cards
ENJOY A KILLER COLLECTION OF MOVIES.
USE CODE PAYHALF
20% CashBack* with LAZYPAY
book my show STREAM
CURATED WITH LOVE
Buy or Rent
WATCH ON book my show ONLINE
NCPA
AN NCPA & BOOKMYSHOW LIVE PRODUCTION
SEA WALL
BY SIMON STEPHENS
DIRECTED BY BRUCE GUTHRIE
STARRING JIM SARBH
13TH & 14TH MARCH
AGE: 16+
ASSOCIATE SPONSOR BRITISH COUNCIL
Recommended Movies
See all
The Best of Entertainment
Entertainment
Entertainment
Entertainment
Entertainment
Entertainment

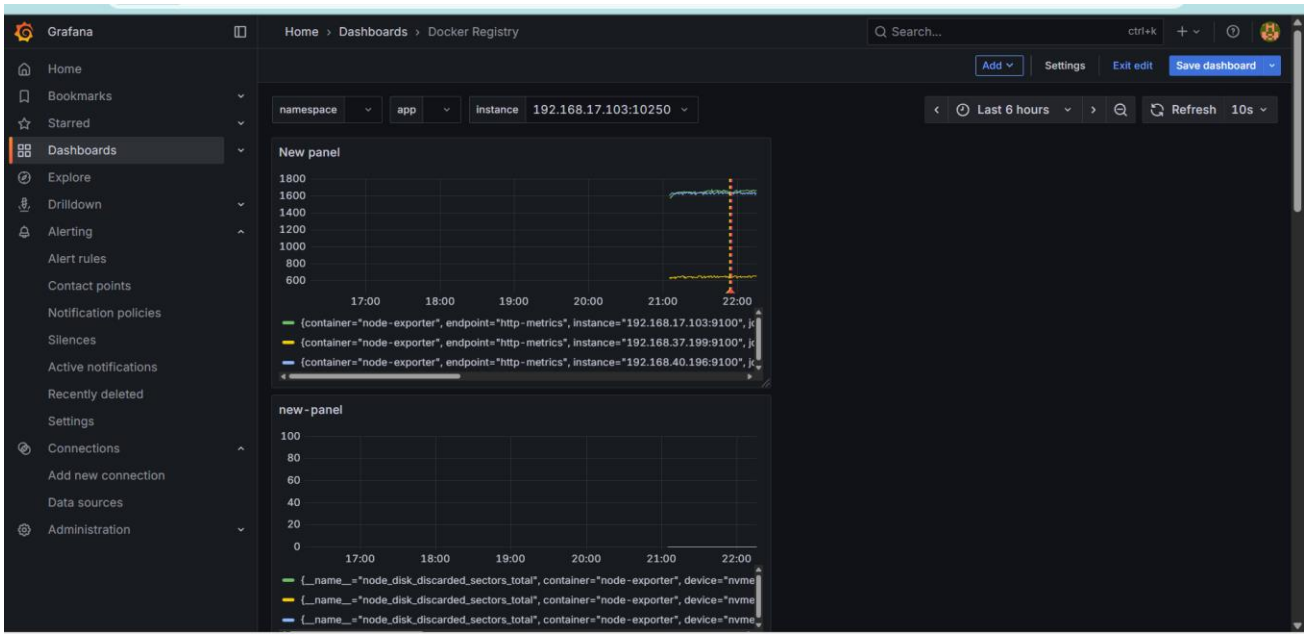

eu-north-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0b9fa32ec7c2ed0ed&osUser=ubuntu&region=eu-north-1...
aws
Search
[Alt+S]
Europe (Stockholm)
Account ID: 3513-7043-0610
SatyamAWS
47 eksctl get nodegroup --cluster satyam-cluster --region eu-north-1
48 history
root@ip-172-31-20-189:/home/ubuntu# ekctl delete cluster --name ^C
root@ip-172-31-20-189:/home/ubuntu# eksctl delete cluster \
--name satyam-cluster \
--region eu-north-1
2025-09-14 19:55:47 [i] deleting EKS cluster "satyam-cluster"
2025-09-14 19:55:47 [i] will drain 0 unmanaged nodegroup(s) in cluster "satyam-cluster"
2025-09-14 19:55:47 [i] starting parallel draining, max in-flight of 1
2025-09-14 19:55:47 [i] deleted 0 Fargate profile(s)
2025-09-14 19:55:48 [✓] kubeconfig has been updated
2025-09-14 19:55:48 [i] cleaning up AWS load balancers created by Kubernetes objects of Kind Service or Ingress
2025-09-14 19:56:35 [i]
2 sequential tasks: { delete nodegroup "satyam-nodes", delete cluster control plane "satyam-cluster" [async]
}
2025-09-14 19:56:35 [i] will delete stack "eksctl-satyam-cluster-nodegroup-satyam-nodes"
2025-09-14 19:56:35 [i] waiting for stack "eksctl-satyam-cluster-nodegroup-satyam-nodes" to get deleted
2025-09-14 19:56:35 [i] waiting for CloudFormation stack "eksctl-satyam-cluster-nodegroup-satyam-nodes"
2025-09-14 19:57:05 [i] waiting for CloudFormation stack "eksctl-satyam-cluster-nodegroup-satyam-nodes"
2025-09-14 19:57:50 [i] waiting for CloudFormation stack "eksctl-satyam-cluster-nodegroup-satyam-nodes"
2025-09-14 19:58:55 [i] waiting for CloudFormation stack "eksctl-satyam-cluster-nodegroup-satyam-nodes"
2025-09-14 20:00:50 [i] waiting for CloudFormation stack "eksctl-satyam-cluster-nodegroup-satyam-nodes"
2025-09-14 20:01:49 [i] waiting for CloudFormation stack "eksctl-satyam-cluster-nodegroup-satyam-nodes"
2025-09-14 20:03:22 [i] waiting for CloudFormation stack "eksctl-satyam-cluster-nodegroup-satyam-nodes"
2025-09-14 20:05:15 [i] waiting for CloudFormation stack "eksctl-satyam-cluster-nodegroup-satyam-nodes"
2025-09-14 20:06:18 [i] waiting for CloudFormation stack "eksctl-satyam-cluster-nodegroup-satyam-nodes"
2025-09-14 20:08:10 [i] waiting for CloudFormation stack "eksctl-satyam-cluster-nodegroup-satyam-nodes"

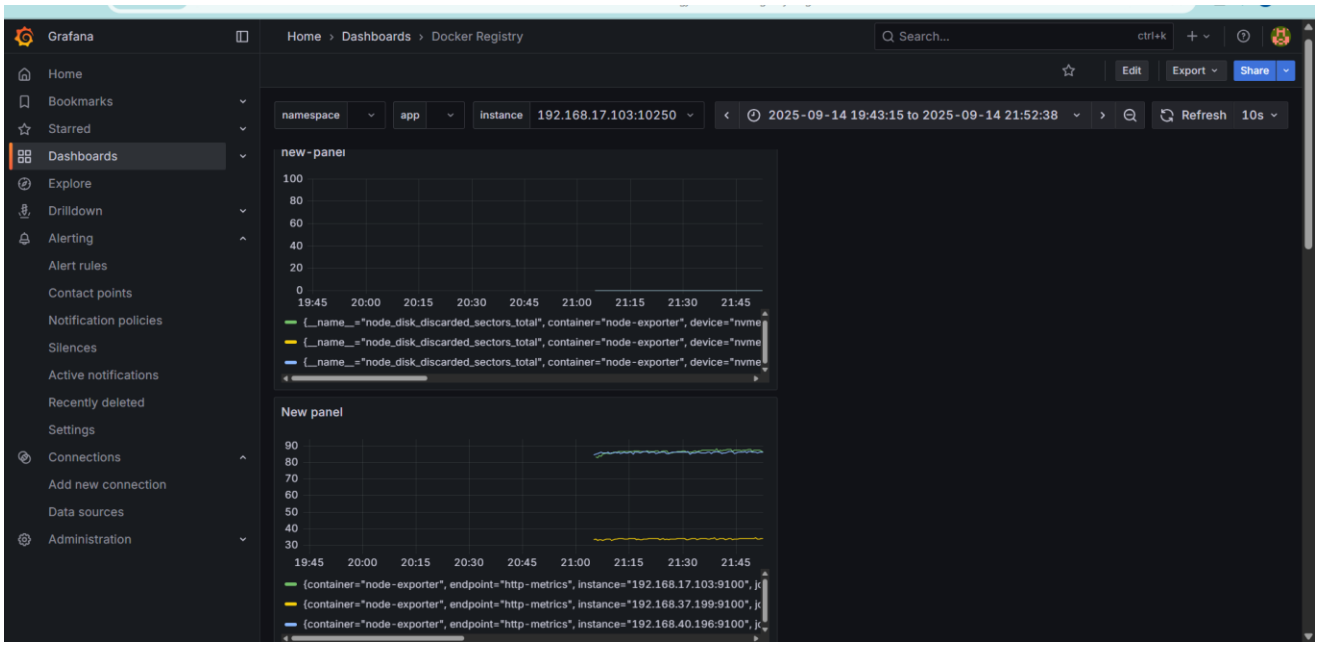
Stage 6

1. Host the application using EKS cluster, then install Promethius and Grafana.
2. Install plugins for Prometheus.
3. Create a data source, contact point along with alerting rules.
4. Go dashboard section and create the dashboard for the application.






Grafana
Home
Bookmarks
Starred
Dashboards
Explore
Drilldown
Alerting
Alert rules
Contact points
Notification policies
Silences
Active notifications
Recently deleted
Settings
Connections
Add new connection
Data sources
Administration
Home › Dashboards › Docker Registry
Search...
ctrl+k
Add
Settings
Exit edit
Save dashboard
namespace
app
instance
192.168.17.103:10250
Last 6 hours
Refresh
10s
New panel
1800
1600
1400
1200
1000
800
600
17:00
18:00
19:00
20:00
21:00
22:00
{container="node-exporter", endpoint="http-metrics", instance="192.168.17.103:9100", jo
{container="node-exporter", endpoint="http-metrics", instance="192.168.37.199:9100", jo
{container="node-exporter", endpoint="http-metrics", instance="192.168.40.196:9100", jo
new-panel
100
80
60
40
20
0
17:00
18:00
19:00
20:00
21:00
22:00
{__name__="node_disk_discarded_sectors_total", container="node-exporter", device="nvme
{__name__="node_disk_discarded_sectors_total", container="node-exporter", device="nvme
{__name__="node_disk_discarded_sectors_total", container="node-exporter", device="nvme


Grafana
Home
Bookmarks
Starred
Dashboards
Explore
Drilldown
Alerting
Alert rules
Contact points
Notification policies
Silences
Active notifications
Recently deleted
Settings
Connections
Add new connection
Data sources
Administration
Home › Dashboards › Docker Registry
Search...
ctrl+k
Edit
Export
Share
namespace
app
instance
192.168.17.103:10250
2025-09-14 19:43:15 to 2025-09-14 21:52:38
Refresh
10s
new-panel
100
80
60
40
20
0
19:45
20:00
20:15
20:30
20:45
21:00
21:15
21:30
21:45
{__name__="node_disk_discarded_sectors_total", container="node-exporter", device="nvme
{__name__="node_disk_discarded_sectors_total", container="node-exporter", device="nvme
{__name__="node_disk_discarded_sectors_total", container="node-exporter", device="nvme
New panel
90
80
70
60
50
40
30
19:45
20:00
20:15
20:30
20:45
21:00
21:15
21:30
21:45
{container="node-exporter", endpoint="http-metrics", instance="192.168.17.103:9100", jo
{container="node-exporter", endpoint="http-metrics", instance="192.168.37.199:9100", jo
{container="node-exporter", endpoint="http-metrics", instance="192.168.40.196:9100", jo