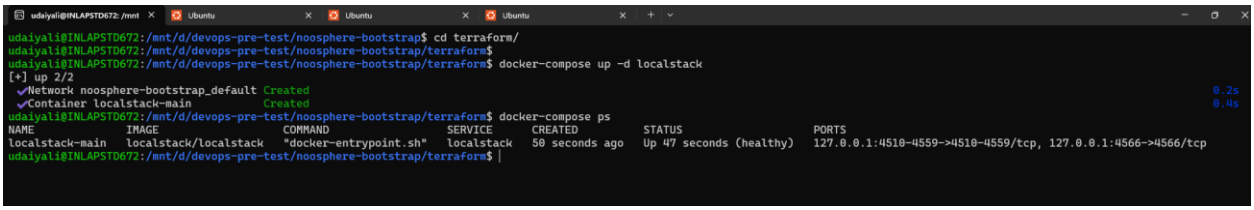
Tech Test — Outcome Summary

1.Terraform & Infrastructure (Part 1)

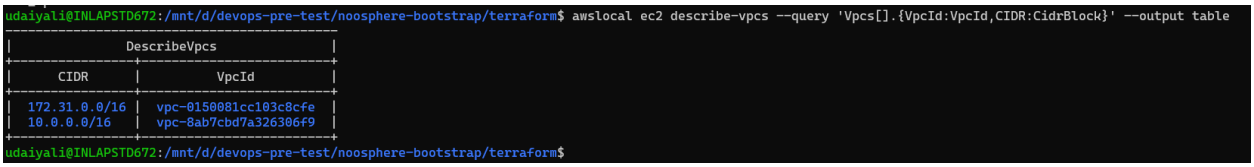
Build infrastructure according to spec, ensure repeatable and extendable code

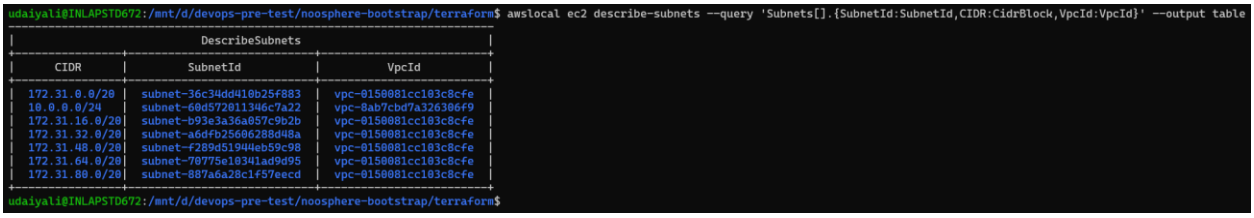
Used localstack

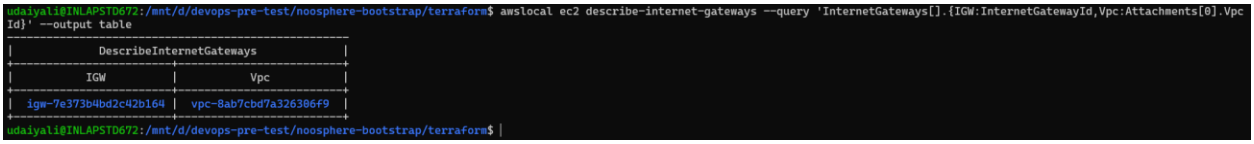


VPC with public subnets

Created VPC and with public subnet. Verified this within LocalStack / AWS CLI.







==========

EC2 Instances

Deployed 3 EC2 instances with distinct configurations using repeatable code that could be extended in future

```
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$ awslocal ec2 describe-instances --query 'Reservations[].Instances[].{Name:Tags[?Key==`Name`][0].Value,InstanceId
:InstanceId,State:State.Name,Type:InstanceType}' --output table
-------------------------------------------------------------------------
|                           DescribeInstances                           |
+----------------------+-------------------------------+---------+------------+
|      InstanceId      |              Name             |  State  |   Type     |
+----------------------+-------------------------------+---------+------------+
|  i-011da8aedde6c0712 |  devops-tech-test-dev-third-ec2  |  running |  t3.medium |
|  i-c0a0e189073cd727f |  devops-tech-test-dev-second-ec2 |  running |  t3.small  |
|  i-efde47e2b6bc14a4e |  devops-tech-test-dev-first-ec2   |  running |  t3.micro  |
+----------------------+-------------------------------+---------+------------+
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$
```

==========

S3 Bucket

Created S3 bucket with lifecycle policy to auto-delete objects after 7 days.

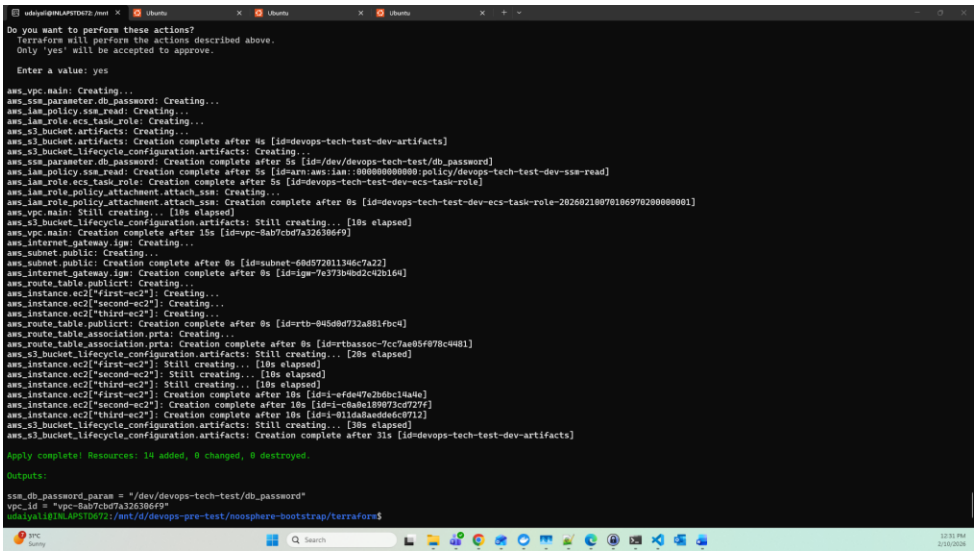
```
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$ awslocal s3 ls
2026-02-10 07:01:05 devops-tech-test-dev-artifacts
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$ awslocal s3api get-bucket-lifecycle-configuration \
    --bucket devops-tech-test-dev-artifacts \
    --query "Rules[].{ID:ID, Status:Status, Prefix:Filter.Prefix, TransitionDays:Transitions[0].Days, StorageClass:Transitions[0].StorageClass, ExpireDays:Expiration.Days}" \
    --output table
-------------------------------------------------------------------------------
|                        GetBucketLifecycleConfiguration                      |
+------------+--------------+---------+----------+----------------+-----------------+
| ExpireDays |      ID      | Prefix  | Status   | StorageClass   | TransitionDays  |
+------------+--------------+---------+----------+----------------+-----------------+
|  7         |  auto-delete |         |  Enabled |  None          |  None           |
+------------+--------------+---------+----------+----------------+-----------------+
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$
```

==========

Terraform Validation

terraform plan and terraform apply successful with no errors. Used env variable to store the secrets and keys

```
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$ export TF_VAR_aws_access_key="test"
export TF_VAR_aws_secret_key="My-VerySecure10x-Secret"
export TF_VAR_db_password='0v*YUVwH#2puW05!0gcdY%3a'
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$ terraform plan -var-file=envs/dev.tfvars
data.aws_iam_policy_document.ecs_task_assume_role: Reading...
data.aws_iam_policy_document.ecs_task_assume_role: Read complete after 0s [id=320642683]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_iam_policy.ssm_read will be created
  + resource "aws_iam_policy" "ssm_read" {
      + arn         = (known after apply)
      + id          = (known after apply)
      + name        = "devops-tech-test-dev-ssm-read"
      + name_prefix = (known after apply)
      + path        = "/"
      + policy      = jsonencode(
            {
              + Statement = [
                  + {
                      + Action   = [
                          + "ssm:GetParameter",
                          + "ssm:GetParameters",
                          + "ssm:GetParametersByPath",
                        ]
                      + Effect   = "Allow"
                      + Resource = "arn:aws:ssm:us-east-1:*:parameter/dev/devops-tech-test/db_password"
                    },
                ]
              + Version   = "2012-10-17"
            }
        )
      + policy_id   = (known after apply)
      + tags_all    = (known after apply)
    }

  # aws_iam_role.ecs_task_role will be created
  + resource "aws_iam_role" "ecs_task_role" {
      + arn                 = (known after apply)
      + assume_role_policy  = jsonencode(
            {
              + Statement = [
                  + {
                      + Action    = "sts:AssumeRole"
                      + Effect    = "Allow"
                      + Principal = {
```




```
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$ terraform state list
data.aws_iam_policy_document.ecs_task_assume_role
aws_iam_policy.ssm_read
aws_iam_role.ecs_task_role
aws_iam_role_policy_attachment.attach_ssm
aws_instance.ec2["first-ec2"]
aws_instance.ec2["second-ec2"]
aws_instance.ec2["third-ec2"]
aws_internet_gateway.igw
aws_route_table.publicrt
aws_route_table_association.prta
aws_s3_bucket.artifacts
aws_s3_bucket_lifecycle_configuration.artifacts
aws_ssm_parameter.db_password
aws_subnet.public
aws_vpc.main
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$
```

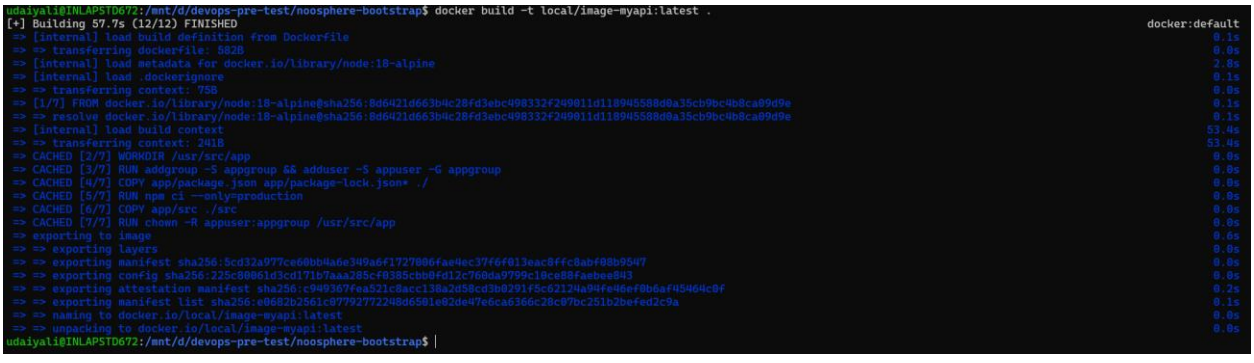
==========

2.Docker & Application Hardening (Part 1)

Ensure secure, minimal container images for deployment

Dockerfile Build

Successfully built Docker image locally.

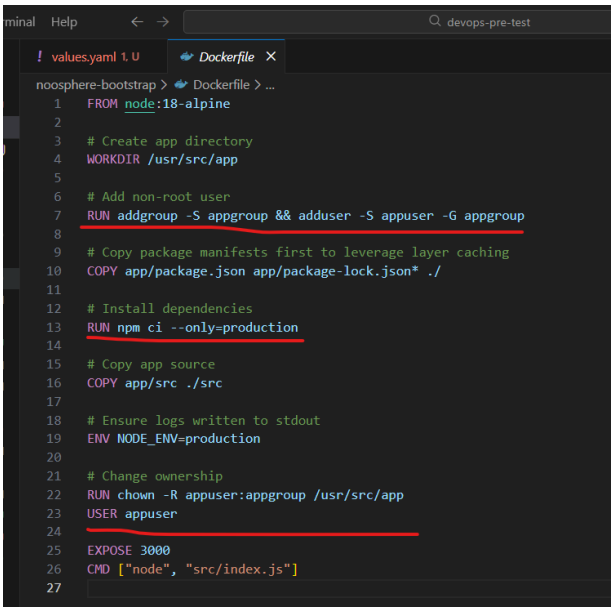


Run as non-root

Verified USER node in Dockerfile; container runs with non-root privileges.

Minimal dependencies

Ensured only what the application requires to run is present

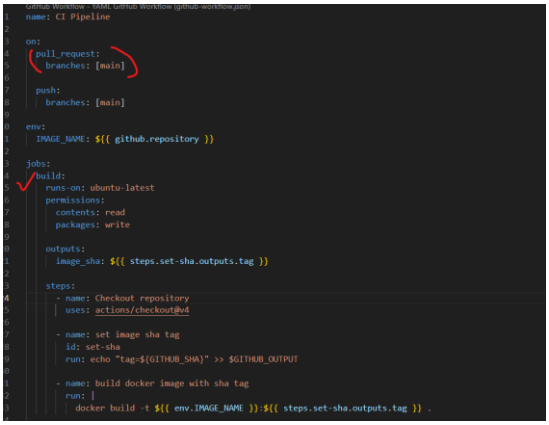


==========

3.CI/CD Pipeline (Part 1)

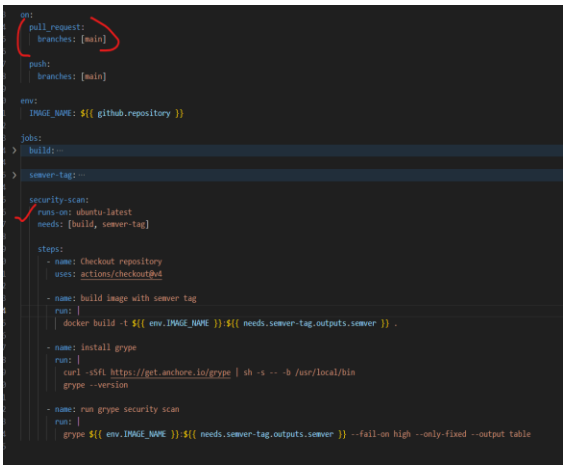
Docker Build

On Pull request Image builds in workflow.



Grype Container Scan

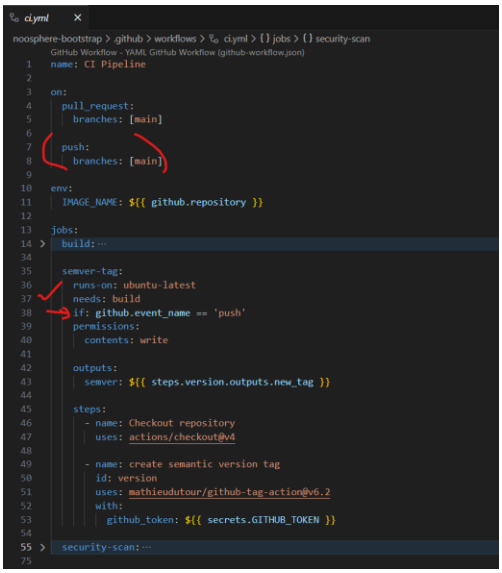
On Pull request Security scan runs and reports vulnerabilities.



```yaml
on:
  pull_request:
    branches: [main]

  push:
    branches: [main]

env:
  IMAGE_NAME: ${{ github.repository }}

jobs:
  build: ···

  semver-tag: ···

  security-scan:
    runs-on: ubuntu-latest
    needs: [build, semver-tag]

    steps:
      - name: Checkout repository
        uses: actions/checkout@v4

      - name: build image with semver tag
        run: |
          docker build -t ${{ env.IMAGE_NAME }}:${{ needs.semver-tag.outputs.semver }} .

      - name: install grype
        run: |
          curl -sSfL https://get.anchore.io/grype | sh -s -- -b /usr/local/bin
          grype --version

      - name: run grype security scan
        run: |
          grype ${{ env.IMAGE_NAME }}:${{ needs.semver-tag.outputs.semver }} --fail-on high --only-fixed --output table
```

Semantic Versioning

On push Tags applied automatically when the push event triggers



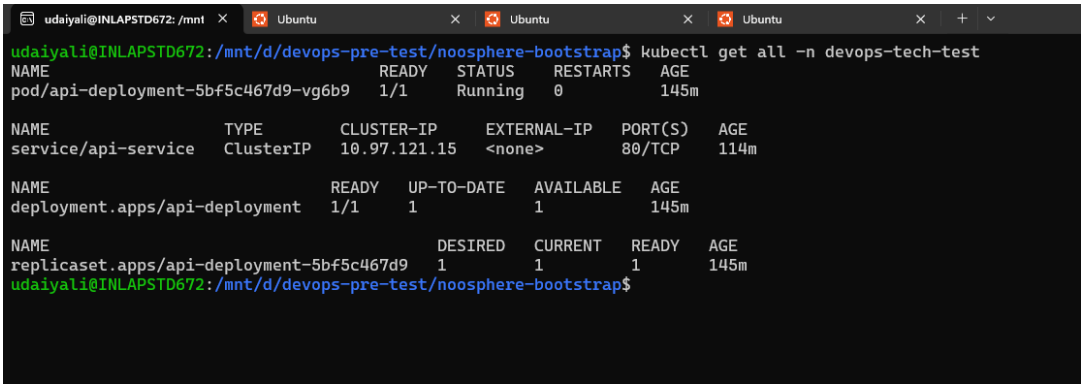
```yaml
ci.yml
noosphere-bootstrap > .github > workflows > ci.yml > {} jobs > {} security-scan
GitHub Workflow - YAML GitHub Workflow (github-workflow.json)
name: CI Pipeline

on:
  pull_request:
    branches: [main]

  push:
    branches: [main]

env:
  IMAGE_NAME: ${{ github.repository }}

jobs:
  build: ···

  semver-tag:
    runs-on: ubuntu-latest
    needs: build
    if: github.event_name == 'push'
    permissions:
      contents: write

    outputs:
      semver: ${{ steps.version.outputs.new_tag }}

    steps:
      - name: Checkout repository
        uses: actions/checkout@v4

      - name: create semantic version tag
        id: version
        uses: mathieudutour/github-tag-action@v6.2
        with:
          github_token: ${{ secrets.GITHUB_TOKEN }}

  security-scan: ···
```

==========
=======

4. Kubernetes Deployment (Part 1)

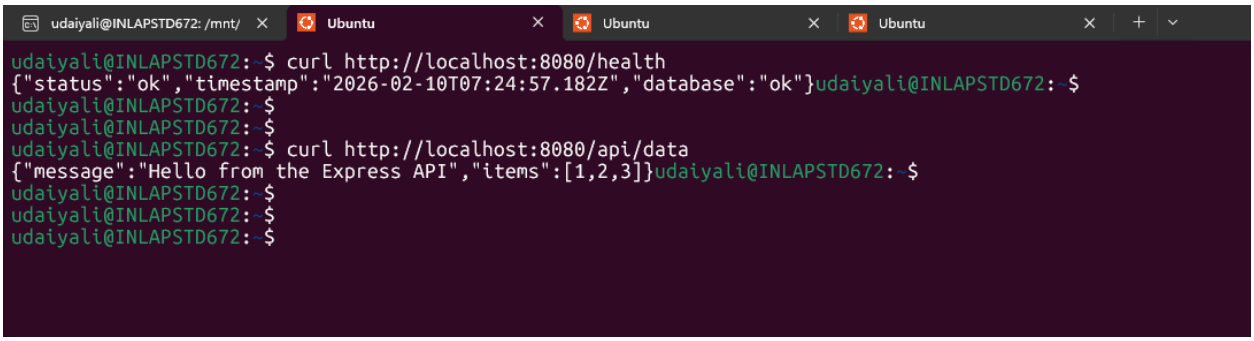
Namespace & Deployment

Updated the manifest files and Applied k8s/namespace.yaml and k8s/deployment.yaml k8s/service.yaml



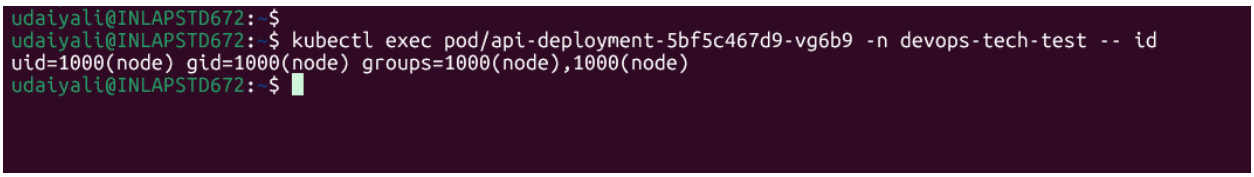
Service Exposure

Port-forwarded service, endpoints verified with curl.



Security Context

Container runs as non-root with restricted permissions.



Commands Tested:

kubectl -n devops-tech-test port-forward svc/api-service 8080:80

curl http://localhost:8080/health

curl http://localhost:8080/api/data

==========

5.Git & Version Control

Conventional commits used for all changes.

Git tags applied automatically via Github Actions workflow.

Atomic commits maintained for clarity and traceability.

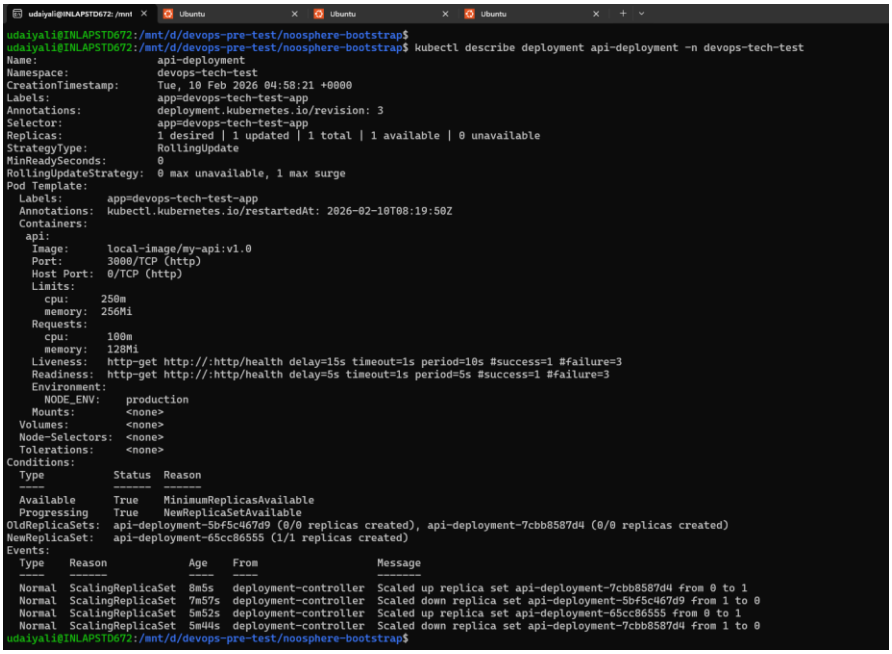
==========

6.Optional Extension (Part 2 — Platform resilience)

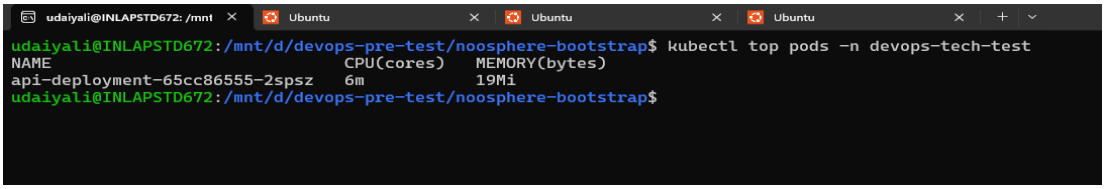
HPA Scaling

Horizontal Pod Autoscaler deployed; app scales under load.

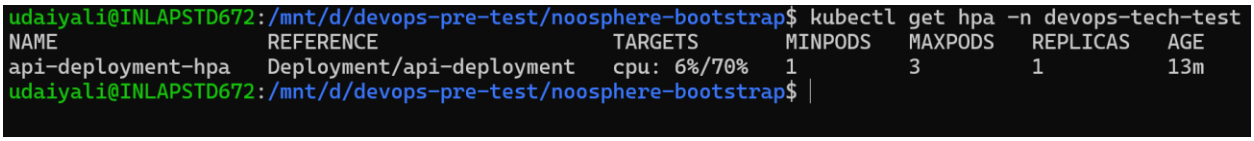
Deployed the application to a Docker Desktop Kubernetes cluster using a rolling update strategy to ensure zero downtime and rollback support.



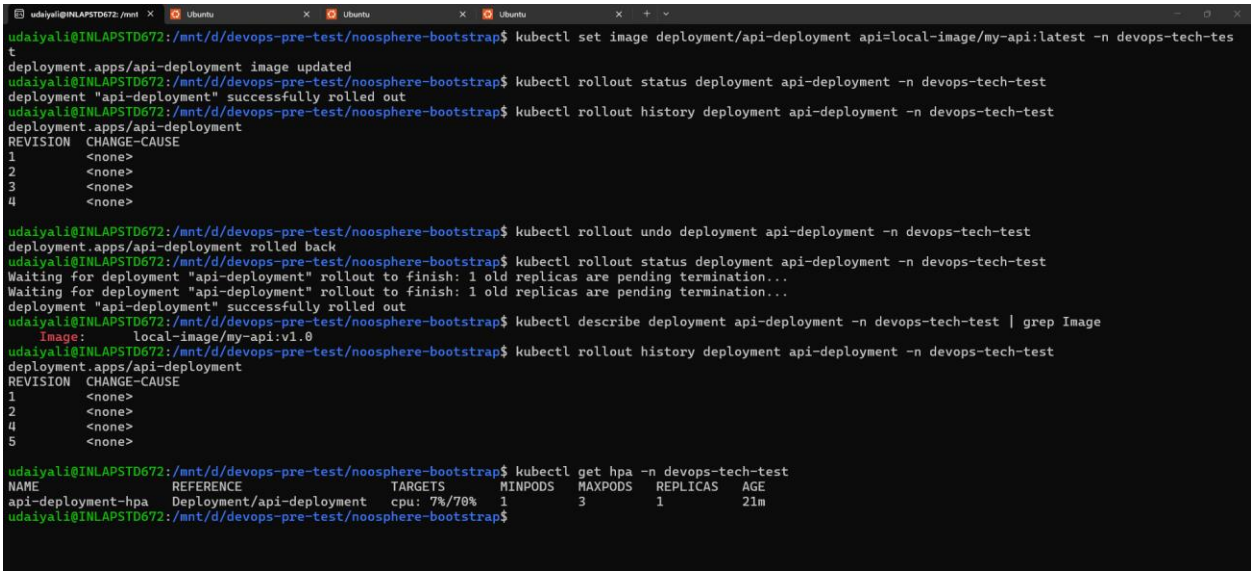
Installed and verified Metrics Server, then configured a Horizontal Pod Autoscaler to scale replicas between 1 and 3 based on CPU utilization.



Enforced resource requests/limits and ran the container as a non-root user and HPA configured and active



Validated the setup using kubectl top, HPA status, and rollout history, with screenshots attached as evidence.



==========

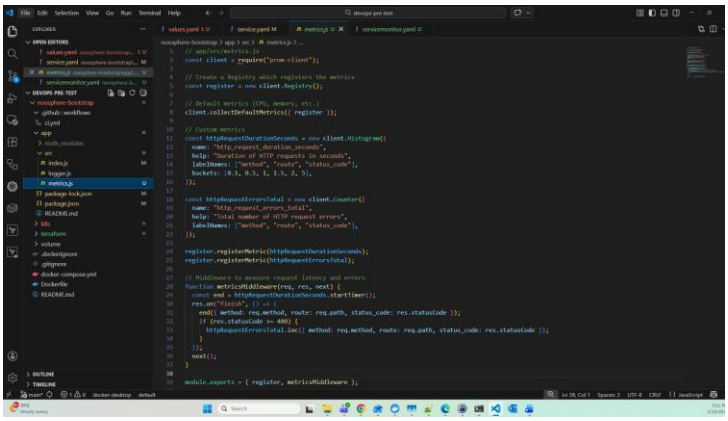
Observability

Prometheus Metrics

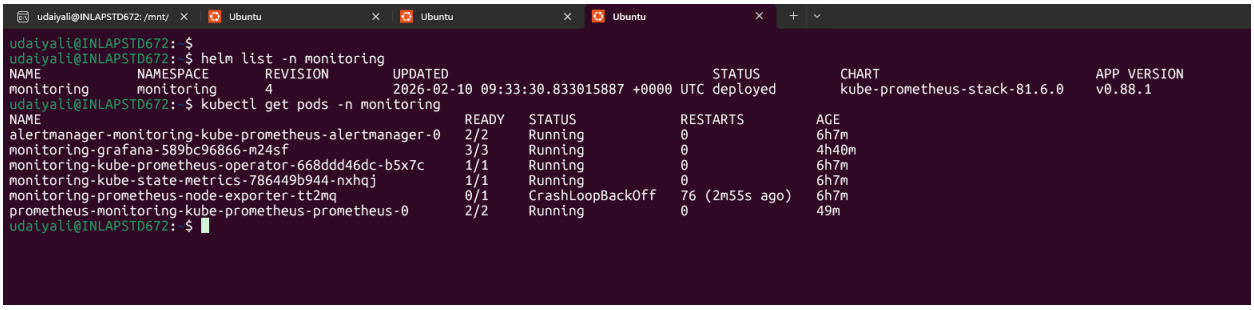
Metrics integrated into Express app (latency, error rate).

Summary (with help of LLM guidance):

Added Prometheus metrics to Express app. Used metrics.js and updated index.js to expose metrics like request latency and error counts.



Deployed Prometheus & Grafana using Helm

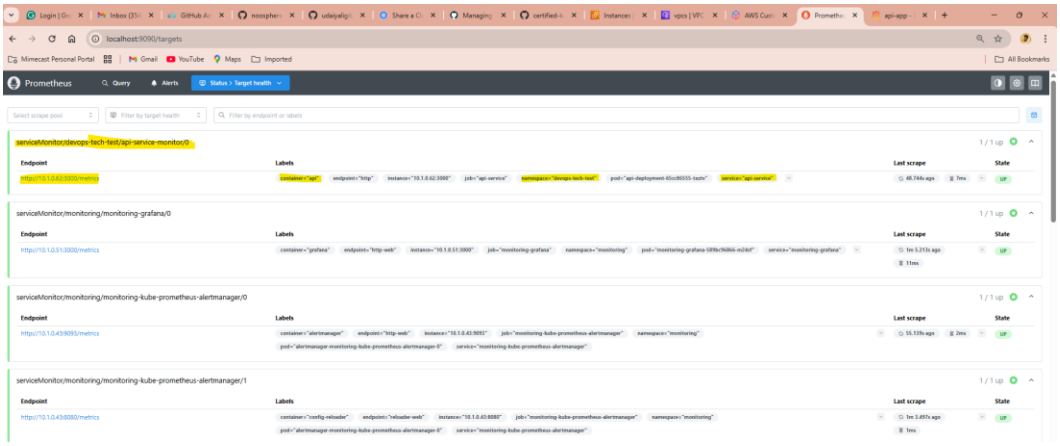


```
udaiyali@INLAPSTD672: /mnt/ | Ubuntu | Ubuntu | Ubuntu

udaiyali@INLAPSTD672:~$
udaiyali@INLAPSTD672:~$ helm list -n monitoring
NAME            NAMESPACE       REVISION        UPDATED                                 STATUS          CHART                           APP VERSION
monitoring      monitoring      4               2026-02-10 09:33:30.833015887 +0000 UTC deployed        kube-prometheus-stack-81.6.0    v0.88.1
udaiyali@INLAPSTD672:~$ kubectl get pods -n monitoring
NAME                                                     READY   STATUS             RESTARTS          AGE
alertmanager-monitoring-kube-prometheus-alertmanager-0   2/2     Running            0                 6h7m
monitoring-grafana-589bc96866-m24sf                      3/3     Running            0                 4h40m
monitoring-kube-prometheus-operator-668ddd46dc-b5x7c     1/1     Running            0                 6h7m
monitoring-kube-state-metrics-786449b944-nxhqj           1/1     Running            0                 6h7m
monitoring-prometheus-node-exporter-tt2mq                0/1     CrashLoopBackOff   76 (2m55s ago)    6h7m
prometheus-monitoring-kube-prometheus-prometheus-0       2/2     Running            0                 49m
udaiyali@INLAPSTD672:~$
```

Wrote servicemonitor.yaml matching deployment labels (app: devops-tech-test-app) and created for scraping.

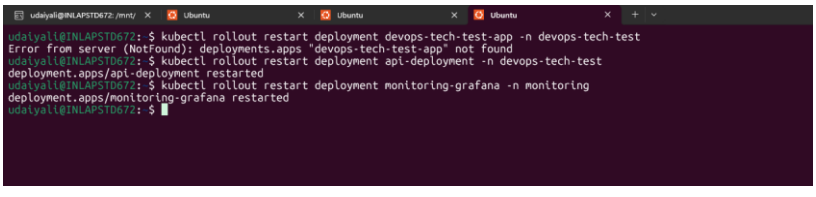
```
udaiyali@INLAPSTD672:~$ kubectl get servicemonitor -n monitoring
NAME                                              AGE
monitoring-grafana                                6h10m
monitoring-kube-prometheus-alertmanager           6h10m
monitoring-kube-prometheus-apiserver              6h10m
monitoring-kube-prometheus-coredns                6h10m
monitoring-kube-prometheus-kube-controller-manager   6h10m
monitoring-kube-prometheus-kube-etcd              6h10m
monitoring-kube-prometheus-kube-proxy             6h10m
monitoring-kube-prometheus-kube-scheduler         6h10m
monitoring-kube-prometheus-kubelet                6h10m
monitoring-kube-prometheus-operator               6h10m
monitoring-kube-prometheus-prometheus             6h10m
monitoring-kube-state-metrics                     6h10m
monitoring-prometheus-node-exporter               6h10m
udaiyali@INLAPSTD672:~$ kubectl describe servicemonitor api-service-monitor -n devops-tech-test
Name:         api-service-monitor
Namespace:    devops-tech-test
Labels:       release=monitoring
Annotations:  <none>
API Version:  monitoring.coreos.com/v1
Kind:         ServiceMonitor
Metadata:
  Creation Timestamp:  2026-02-10T05:55:47Z
  Generation:          1
  Resource Version:    35974
  UID:                 466a3a45-486b-43dd-995f-1bb576582368
Spec:
  Endpoints:
    Interval:  15s
    Path:      /metrics
    Port:      http
  Namespace Selector:
    Match Names:
      devops-tech-test
  Selector:
    Match Labels:
      App:  devops-tech-test-app
Events:     <none>
udaiyali@INLAPSTD672:~$
```

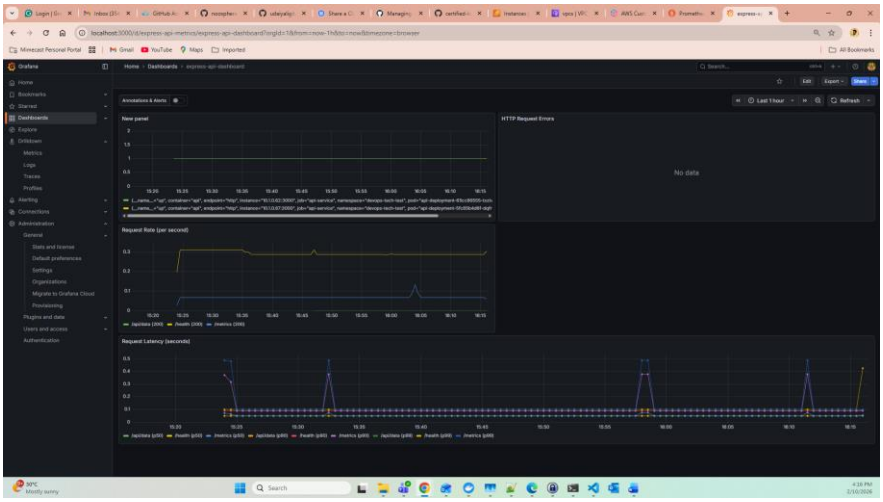
Prometheus discovered and scraped API service



Created Grafana dashboard

Ensured persistence through restarts




Grafana
Home
Bookmarks
Starred
Dashboards
Explore
Drilldown
Metrics
Logs
Traces
Profiles
Alerting
Connections
Administration
General
Stats and license
Default preferences
Settings
Organizations
Migrate to Grafana Cloud
Provisioning
Plugins and data
Users and access
Authentication
Home › Dashboards › express-api-dashboard
Edit
Export
Share
Annotations & Alerts
Last 1 hour
Refresh
New panel
HTTP Request Errors
No data
Request Rate (per second)
Request Latency (seconds)