

Tech Test — Outcome Summary

1.Terraform & Infrastructure (Part 1)

Build infrastructure according to spec, ensure repeatable and extendable code

Used localstack

```
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ cd terraform/
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$ docker-compose up -d localstack
[+] up 2/2
  ✓ Network noosphere-bootstrap_default Created
  ✓ Container localstack-main Created
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$ docker-compose ps
NAME                IMAGE                COMMAND                  SERVICE    CREATED      STATUS      PORTS
localstack-main     localstack/localstack  "docker-entrypoint.sh"  localstack  50 seconds ago  Up 47 seconds (healthy)  127.0.0.1:4510-4559->4510-4559/tcp, 127.0.0.1:4566->4566/tcp
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$
```

VPC with public subnets

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

```
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$ awslocal ec2 describe-vpcs --query 'Vpcs[].{VpcId:VpcId,CIDR:CidrBlock}' --output table
DescribeVpcs
+-----+-----+
| CIDR | VpcId |
+-----+-----+
| 172.31.0.0/16 | vpc-0150081cc103c8cfe |
| 10.0.0.0/16 | vpc-8ab7cbd7a326306f9 |
+-----+-----+
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$
```

```
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$ awslocal ec2 describe-subnets --query 'Subnets[].{SubnetId:SubnetId,CIDR:CidrBlock,VpcId:VpcId}' --output table
DescribeSubnets
+-----+-----+-----+
| CIDR | SubnetId | VpcId |
+-----+-----+-----+
| 172.31.0.0/20 | subnet-36c34dd410b25f883 | vpc-0150081cc103c8cfe |
| 10.0.0.0/24 | subnet-60d572b11346c7a22 | vpc-8ab7cbd7a326306f9 |
| 172.31.16.0/20 | subnet-b93e3a36a087c902b | vpc-0150081cc103c8cfe |
| 172.31.32.0/20 | subnet-a6dfb25606288d48a | vpc-0150081cc103c8cfe |
| 172.31.48.0/20 | subnet-f289d51944ab59c98 | vpc-0150081cc103c8cfe |
| 172.31.64.0/20 | subnet-70775e10341ad9d95 | vpc-0150081cc103c8cfe |
| 172.31.80.0/20 | subnet-887a6a28c1f57eedc | vpc-0150081cc103c8cfe |
+-----+-----+-----+
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$
```

```
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$ awslocal ec2 describe-internet-gateways --query 'InternetGateways[].{IGW:InternetGatewayId,Vpc:Attachments[0].VpcId}' --output table
DescribeInternetGateways
+-----+-----+
| IGW | Vpc |
+-----+-----+
| igw-7e373b4bd2c42b164 | vpc-8ab7cbd7a326306f9 |
+-----+-----+
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$
```

```
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$ awslocal ec2 describe-route-tables --query 'RouteTables[].{RouteTableId:RouteTableId,VpcId:VpcId}' --output table
DescribeRouteTables
+-----+-----+
| RouteTableId | VpcId |
+-----+-----+
| rtb-c9bc04a7f2104695f | vpc-0150081cc103c8cfe |
| rtb-6622e8fab8f8909df | vpc-8ab7cbd7a326306f9 |
| rtb-045d9d732a881fbc4 | vpc-8ab7cbd7a326306f9 |
+-----+-----+
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$
```

EC2 Instances

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

```
=====
=====
```

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

=====

=====

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

```

awscli@1876c48f72b2: /mnt/d/devops-proj$ terraform init -backend=false
export TF_VAR_aws_secret_key="My-VerySecure123-Secret"
export TF_VAR_aws_access_key="test"
export TF_VAR_aws_db_password="MyVulnH2pW05!0ncdY1Ja"
awscli@1876c48f72b2: /mnt/d/devops-proj$ 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 8s [id=328642693]

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.sss_read will be created
+ resource "aws_iam_policy" "sss_read" {
  + arn = (known after apply)
  + id = (known after apply)
  + name = "devops-tech-test-dev-sss-read"
  + name_prefix = (known after apply)
  + path = "/"
  + policy = jsonencode(
    {
      Statement = [
        {
          Action = [
            "s3:GetParameters",
            "s3:GetParameters*",
            "s3:GetParametersByPath",
          ]
          Effect = "Allow"
          Resource = ["arn:aws:sss: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 = {

```

```
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_vpc.main: Creating...
aws_ssm_parameter.db_password: Creating...
aws_iam_policy.ssm_read: Creating...
aws_iam_role.ecs_task_role: Creating...
aws_s3_bucket.artifacts: Creating...
aws_s3_bucket.artifacts: Creation complete after 4s [id=devops-tech-test-dev-artifacts]
aws_s3_bucket_lifecycle_configuration.artifacts: Creating...
aws_ssm_parameter.db_password: Creation complete after 5s [id=/dev/devops-tech-test/db_password]
aws_iam_policy.ssm_read: Creation complete after 5s [id=arn:aws:iam::808080808080:policy/devops-tech-test-dev-ssm-read]
aws_iam_role.ecs_task_role: Creation complete after 5s [id=devops-tech-test-dev-ecs-task-role]
aws_iam_role_policy_attachment.attach_ssm: Creating...
aws_iam_role_policy_attachment.attach_ssm: Creation complete after 8s [id=devops-tech-test-dev-ecs-task-role-2026021807010697020808080801]
aws_vpc.main: Still creating... [10s elapsed]
aws_s3_bucket_lifecycle_configuration.artifacts: Still creating... [10s elapsed]
aws_vpc.main: Creation complete after 15s [id=vpc-bab7cbd7a326386f9]
aws_internet_gateway.igw: Creating...
aws_subnet.public: Creating...
aws_subnet.public: Creation complete after 0s [id=subnet-6d0572813bdc7a22]
aws_internet_gateway.igw: Creation complete after 8s [id=igw-7a373b4b02c42b164]
aws_route_table.publicrt: Creating...
aws_instance.ec2["first-ec2"]: Creating...
aws_instance.ec2["second-ec2"]: Creating...
aws_instance.ec2["third-ec2"]: Creating...
aws_route_table.publicrt: Creation complete after 8s [id=rtb-0456b0d732a881fbc4]
aws_route_table_association.prta: Creating...
aws_route_table_association.prta: Creation complete after 0s [id=rtbassoc-7cc7a085f978c4u81]
aws_s3_bucket_lifecycle_configuration.artifacts: Still creating... [20s elapsed]
aws_instance.ec2["first-ec2"]: Still creating... [10s elapsed]
aws_instance.ec2["second-ec2"]: Still creating... [10s elapsed]
aws_instance.ec2["third-ec2"]: Still creating... [10s elapsed]
aws_instance.ec2["first-ec2"]: Creation complete after 10s [id=i-efda7e2bdc14a4e]
aws_instance.ec2["second-ec2"]: Creation complete after 10s [id=i-cbae18973c07271]
aws_instance.ec2["third-ec2"]: Creation complete after 10s [id=i-01da8aedd6c0712]
aws_s3_bucket_lifecycle_configuration.artifacts: Still creating... [30s elapsed]
aws_s3_bucket_lifecycle_configuration.artifacts: Creation complete after 31s [id=devops-tech-test-dev-artifacts]

Apply complete! Resources: 14 added, 0 changed, 0 destroyed.

Outputs:
ssm_db_password_param = "/dev/devops-tech-test/db_password"
vpc_id = "vpc-bab7cbd7a326386f9"
udaiyal1@INLAPSTD672: /mnt/d/devops-pre-test/noosphere-bootstrap/terraform$
```

```
udaiyal1@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
udaiyal1@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.

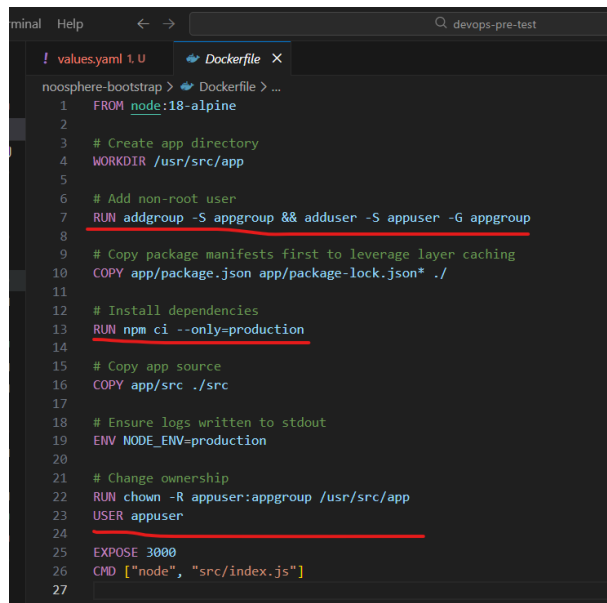
```
udaiyal1@INLAPSTD672: /mnt/d/devops-pre-test/noosphere-bootstrap$ docker build -t local/image-mypai:latest .
[*] Building 57.7s (12/12) FINISHED
=> [internal] lean build definition from Dockerfile
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [internal] load .dockerignore
=> [internal] load context: 75B
=> [1/7] FROM docker.io/library/node:18-alpine@sha256:8d6421d663bdc28fd3ebc498332f24981d1d18945588d0a35cb9cc4b0ca69d9e
=> resolve docker.io/library/node:18-alpine@sha256:8d6421d663bdc28fd3ebc498332f24981d1d18945588d0a35cb9cc4b0ca69d9e
=> [internal] load build context
=> transferring context: 241B
=> CACHED [2/7] WORKDIR /usr/src/app
=> CACHED [3/7] RUN addgroup -S appgroup && adduser -S appuser -G appgroup
=> CACHED [4/7] COPY app/package.json app/package-lock.json* ./
=> CACHED [5/7] RUN npm ci --only=production
=> CACHED [6/7] COPY app/src ./src
=> CACHED [7/7] RUN chown -R appuser:appgroup /usr/src/app
=> exporting to image
=> exporting layers
=> exporting manifest sha256:3cd3a077c08b04a6e389a6f1727b864ae4ec37764013aac8f9c8ab408b9507
=> exporting config sha256:22c08081d3c0171b7aa2485cf833c1bd04d12c76bda9799c18e085faeb4a43
=> exporting attestation manifest sha256:c949367fea521c8acc138a2d58cd368291f5c62124a98f46ef0b4af45404c0f
=> exporting manifest list sha256:e0682d2561c97792772248d6581e92de47e6ca6366c28c97bc251b2befedc9a
=> naming to docker.io/local/image-mypai:latest
=> unpacking to docker.io/local/image-mypai:latest
udaiyal1@INLAPSTD672: /mnt/d/devops-pre-test/noosphere-bootstrap$
```

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

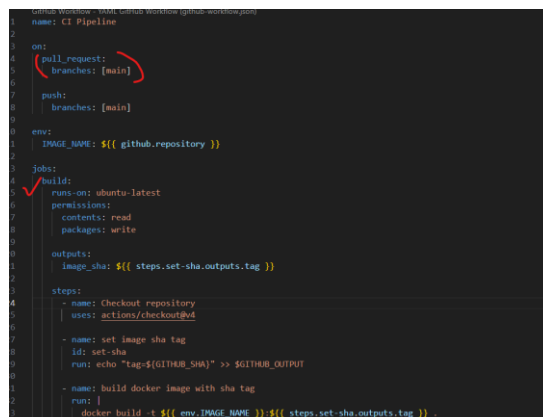


```
1 FROM node:18-alpine
2
3 # Create app directory
4 WORKDIR /usr/src/app
5
6 # Add non-root user
7 RUN addgroup -S appgroup && adduser -S appuser -G appgroup
8
9 # Copy package manifests first to leverage layer caching
10 COPY app/package.json app/package-lock.json* ./
11
12 # Install dependencies
13 RUN npm ci --only=production
14
15 # Copy app source
16 COPY app/src ./src
17
18 # Ensure logs written to stdout
19 ENV NODE_ENV=production
20
21 # Change ownership
22 RUN chown -R appuser:appgroup /usr/src/app
23 USER appuser
24
25 EXPOSE 3000
26 CMD ["node", "src/index.js"]
27
```

3.CI/CD Pipeline (Part 1)

Docker Build

On Pull request Image builds in workflow.



```
1 name: CI Pipeline
2
3 on:
4   pull_request:
5     branches: [main]
6
7   push:
8     branches: [main]
9
10 env:
11   IMAGE_NAME: ${GITHUB_REPOSITORY}
12
13 jobs:
14   build:
15     runs-on: ubuntu-latest
16     permissions:
17       contents: read
18       packages: write
19     outputs:
20       image_sha: ${GITHUB_OUTPUT}
21
22     steps:
23       - name: Checkout repository
24         uses: actions/checkout@v4
25
26       - name: set image sha tag
27         id: set-sha
28         run: echo "tag=${GITHUB_SHA}" >> $GITHUB_OUTPUT
29
30       - name: build docker image with sha tag
31         run: docker build -t ${{ env.IMAGE_NAME }}:${{ steps.set-sha.outputs.tag }} .
```

Grype Container Scan

On Pull request Security scan runs and reports vulnerabilities.

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

env:
  IMAGE_NAME: ${ github.repository }

jobs:
  build:
    runs-on: ubuntu-latest
    needs: [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 -sSf 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

```
% ci.yml X
noosphere-bootstrap > github > workflows > % ci.yml > {} jobs > {} security-scan
Github Workflow - Basic, GitHub Workflow (github-workflow.json)
1  name: CI Pipeline
2
3  on:
4    pull_request:
5      branches: [main]
6
7    push:
8      branches: [main]
9
10 env:
11   IMAGE_NAME: ${ github.repository }
12
13 jobs:
14   build:
15
16   semver-tag:
17     runs-on: ubuntu-latest
18     needs: build
19     if: github.event_name == 'push'
20     permissions:
21       contents: write
22
23     outputs:
24       semver: ${ steps.version.outputs.new_tag }
25
26     steps:
27       - name: Checkout repository
28         uses: actions/checkout@v4
29
30       - name: create semantic version tag
31         id: version
32         uses: mathieudutour/github-tag-action@v6.2
33         with:
34           github_token: ${ secrets.GITHUB_TOKEN }
35
36   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

```
udaiali@INLAPSTD672: /mnt/ x Ubuntu x Ubuntu x Ubuntu
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ kubectl get all -n devops-tech-test
NAME                                READY    STATUS    RESTARTS   AGE
pod/api-deployment-5bf5c467d9-vg6b9  1/1      Running   0           145m

NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
service/api-service                 ClusterIP      10.97.121.15   <none>        80/TCP      114m

NAME                                READY    UP-TO-DATE   AVAILABLE   AGE
deployment.apps/api-deployment      1/1      1             1           145m

NAME                                DESIRED    CURRENT    READY   AGE
replicaset.apps/api-deployment-5bf5c467d9  1          1          1       145m
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$
```

Service Exposure

Port-forwarded service, endpoints verified with curl.

```
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ kubectl -n devops-tech-test port-forward svc/api-service 8080:80
Forwarding from 127.0.0.1:8080 -> 3000
Forwarding from [::1]:8080 -> 3000
Handling connection for 8080
Handling connection for 8080
```

```
udaiali@INLAPSTD672: $ curl http://localhost:8080/health
{"status":"ok","timestamp":"2026-02-10T07:24:57.182Z","database":"ok"}udaiali@INLAPSTD672: $
udaiali@INLAPSTD672: $
udaiali@INLAPSTD672: $ curl http://localhost:8080/api/data
{"message":"Hello from the Express API","items":[1,2,3]}udaiali@INLAPSTD672: $
udaiali@INLAPSTD672: $
udaiali@INLAPSTD672: $
udaiali@INLAPSTD672: $
```

Security Context

Container runs as non-root with restricted permissions.

```
udaiali@INLAPSTD672: $
udaiali@INLAPSTD672: $ kubectl exec pod/api-deployment-5bf5c467d9-vg6b9 -n devops-tech-test -- id
uid=1000(node) gid=1000(node) groups=1000(node),1000(node)
udaiali@INLAPSTD672: $ █
```

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.

```
ufaiyal@INLAPST0672: /mnt/d/devops-pre-test/noosphere-bootstrap$
ufaiyal@INLAPST0672: /mnt/d/devops-pre-test/noosphere-bootstrap$ kubectl describe deployment api-deployment -n devops-tech-test
Name:          api-deployment
Namespace:     devops-tech-test
CreationTimestamp: Tue, 10 Feb 2026 04:58:21 +0800
Labels:        app=devops-tech-test-app
Annotations:    deployment.kubernetes.io/revision: 3
Selector:      app=devops-tech-test-app
Replicas:      1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:  RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 0 max unavailable, 1 max surge
Pod Template:
  Labels:  app=devops-tech-test-app
  Annotations:  kubectl.kubernetes.io/restartedAt: 2026-02-10T08:19:50Z
  Containers:
    api:
      Image:  local-image/my-api:v1.0
      Port:   3088/TCP (http)
      Host Port: 0/TCP (http)
      Limits:
        cpu:      250m
        memory: 256Mi
      Requests:
        cpu:      100m
        memory: 128Mi
      Liveness:  http-get http://:http/health delay=15s timeout=1s period=10s #success=1 #failure=3
      Readiness: http-get http://:http/health delay=5s timeout=1s period=5s #success=1 #failure=3
  Environment:
    NODE_ENV:  production
  Mounts:      <none>
  Volumes:     <none>
  Node-Selectors:  <none>
  Tolerations:    <none>
Conditions:
  Type            Status  Reason
  ----            -
  Available       True    MinimumReplicasAvailable
  Progressing     True    NewReplicaSetAvailable
OldReplicaSets:  api-deployment-5bf5c467d9 (0/0 replicas created), api-deployment-7cbb8587d4 (0/0 replicas created)
NewReplicaSet:   api-deployment-65cc86555 (1/1 replicas created)
Events:
  Type    Reason      Age    From          Message
  ----    -
  Normal  ScalingReplicaSet  8m5s  deployment-controller  Scaled up replica set api-deployment-7cbb8587d4 from 0 to 1
  Normal  ScalingReplicaSet  7m57s  deployment-controller  Scaled down replica set api-deployment-5bf5c467d9 from 1 to 0
  Normal  ScalingReplicaSet  5m52s  deployment-controller  Scaled up replica set api-deployment-65cc86555 from 0 to 1
  Normal  ScalingReplicaSet  5m44s  deployment-controller  Scaled down replica set api-deployment-7cbb8587d4 from 1 to 0
ufaiyal@INLAPST0672: /mnt/d/devops-pre-test/noosphere-bootstrap$
```

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

```
udaiali@INLAPSTD672: /mnt x Ubuntu x Ubuntu x Ubuntu
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ kubectl top pods -n devops-tech-test
NAME                                CPU(cores)  MEMORY(bytes)
api-deployment-65cc86555-2spsz      6m          19Mi
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$
```

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

```
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ kubectl get hpa -n devops-tech-test
NAME                                REFERENCE                TARGETS      MINPODS  MAXPODS  REPLICAS  AGE
api-deployment-hpa                 Deployment/api-deployment  cpu: 6%/70%    1         3         1         13m
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ |
```

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

```
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ kubectl set image deployment/api-deployment api=local-image/my-api:latest -n devops-tech-test
deployment.apps/api-deployment image updated
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ kubectl rollout status deployment api-deployment -n devops-tech-test
deployment "api-deployment" successfully rolled out
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ kubectl rollout history deployment api-deployment -n devops-tech-test
deployment.apps/api-deployment
REVISION  CHANGE-CAUSE
1          <none>
2          <none>
3          <none>
4          <none>

udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ kubectl rollout undo deployment api-deployment -n devops-tech-test
deployment.apps/api-deployment rolled back
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ kubectl rollout status deployment api-deployment -n devops-tech-test
Waiting for deployment "api-deployment" rollout to finish: 1 old replicas are pending termination...
Waiting for deployment "api-deployment" rollout to finish: 1 old replicas are pending termination...
deployment "api-deployment" successfully rolled out
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ kubectl describe deployment api-deployment -n devops-tech-test | grep Image
Image:          local-image/my-api:v1.0
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ kubectl rollout history deployment api-deployment -n devops-tech-test
deployment.apps/api-deployment
REVISION  CHANGE-CAUSE
1          <none>
2          <none>
4          <none>
5          <none>

udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ kubectl get hpa -n devops-tech-test
NAME                                REFERENCE                TARGETS      MINPODS  MAXPODS  REPLICAS  AGE
api-deployment-hpa                 Deployment/api-deployment  cpu: 7%/70%    1         3         1         21m
udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$
```

=====

=====

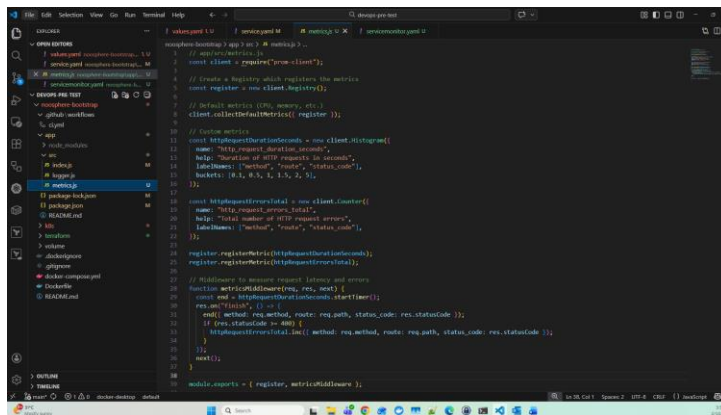
Observability

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/ $ 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-mxhqj      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

```

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

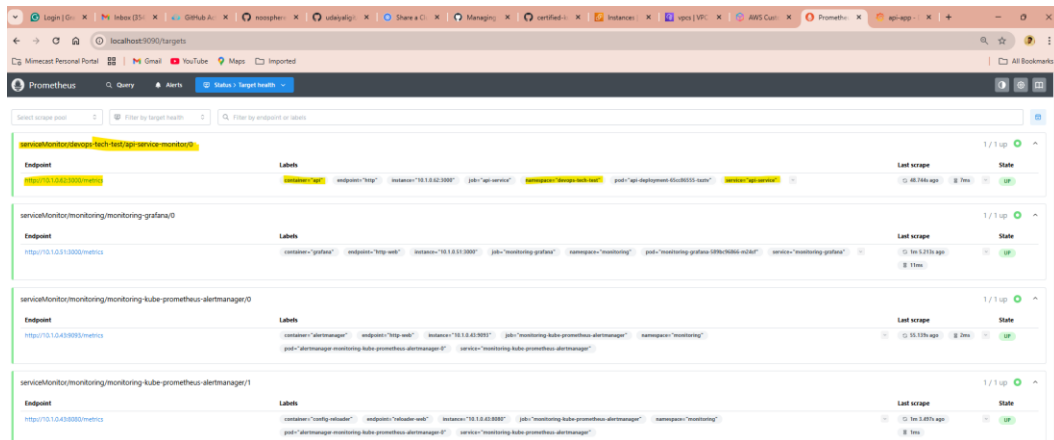
```

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

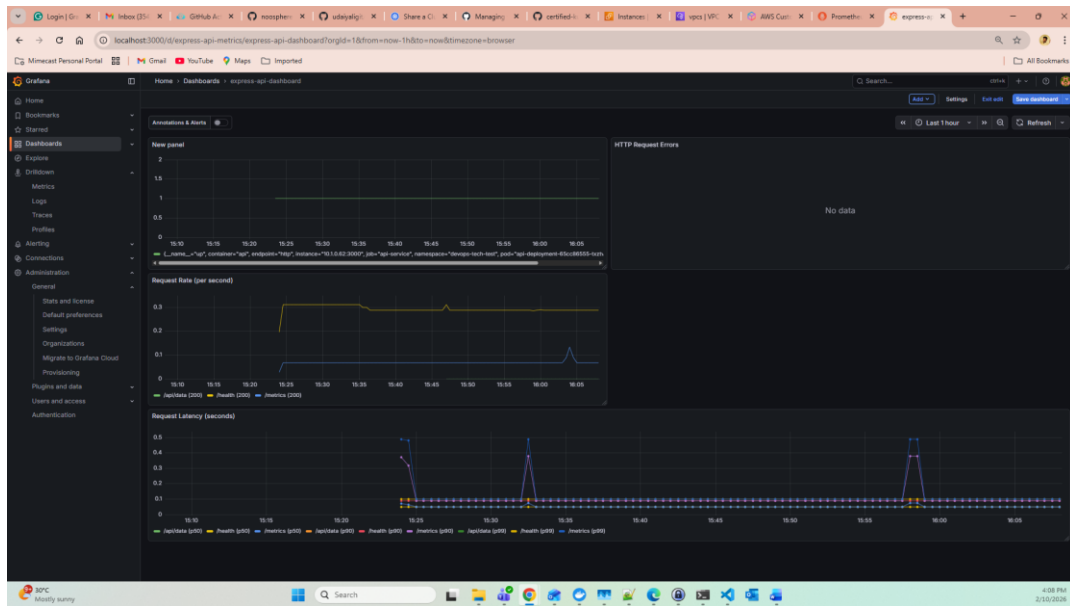
udaiyali@INLAPSTD672: $ kubectl describe servicemonitor api-service-monitor -n devops-tech-test
Name:          api-service-monitor
Namespace:     devops-tech-test
Labels:        release=monitoring
Annotations:   <none>
API Version:   monitoring.coreos.com/v1
Kind:          ServiceMonitor
Metadata:
  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>

```

Prometheus discovered and scraped API service



Created Grafana dashboard



Ensured persistence through restarts

```
udaiyali@INLAPST0672: /mnt/ $ kubectl rollout restart deployment devops-tech-test-app -n devops-tech-test
Error from server (NotFound): deployments.apps "devops-tech-test-app" not found
udaiyali@INLAPST0672: /mnt/ $ kubectl rollout restart deployment api-deployment -n devops-tech-test
deployment.apps/api-deployment restarted
udaiyali@INLAPST0672: /mnt/ $ kubectl rollout restart deployment monitoring-grafana -n monitoring
deployment.apps/monitoring-grafana restarted
udaiyali@INLAPST0672: /mnt/ $
```

