

Tech Test — Outcome Summary

1.Terraform & Infrastructure (Part 1)

Build infrastructure according to spec, ensure repeatable and extendable code

Used localstack

```
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ cd terraform/
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ terraform
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ docker-compose up -d localstack
[+] up 2/2
  ✓ Network noosphere-bootstrap_default Created
  ✓ Container localstack-main Created
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraorm$ 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/tcp, 127.0.0.1:4566-4566/tcp
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraorm$
```

VPC with public subnets

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

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



```
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraorm$ awslocal ec2 describe-subnets --query 'Subnets[].{SubnetId:SubnetId,CIDR:CidrBlock,VpcId:VpcId}' --output table
+-----+-----+-----+
| CIDR | SubnetId | VpcId |
+-----+-----+-----+
| 172.31.0.8/28 | subnet-36c34dd410b25ff883 | vpc-0150081cc103c8cfe |
| 10.0.0.0/24 | subnet-60d572011346c7a22 | vpc-8ab7cbd7a326306f9 |
| 172.31.16.0/28 | subnet-b93e3a3a6a057c9b2b | vpc-0150081cc103c8cfe |
| 172.31.32.0/28 | subnet-a6dfb25606288d48a | vpc-0150081cc103c8cfe |
| 172.31.48.0/28 | subnet-f289d5194e6b59c98 | vpc-0150081cc103c8cfe |
| 172.31.64.0/28 | subnet-70775e10341ad9d95 | vpc-0150081cc103c8cfe |
| 172.31.88.0/28 | subnet-887ada28c157eeed | vpc-0150081cc103c8cfe |
+-----+-----+-----+
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraorm$
```



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



```
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraorm$ awslocal ec2 describe-route-tables --query 'RouteTables[].{RouteTableId:RouteTableId,VpcId:VpcId}' --output table
+-----+-----+
| RouteTableId | VpcId |
+-----+-----+
| rtb-c9bc04a7f2184695f | vpc-0150081cc103c8cfe |
| rtb-6622e8fa0fb4999df | vpc-8ab7cbd7a326306f9 |
| rtb-045dd0d732a81fb4c | vpc-8ab7cbd7a326306f9 |
+-----+-----+
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraorm$
```

=====

=====

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==`Name`][0].Value,InstanceId,State,State.Name,Type:InstanceType' --output table
|-----|
|             DescribeInstances           |
|-----|
| InstanceId | Name          | State       | Type        |
|-----|
| i-01lda8aedde6c0712 | devops-tech-test-dev-third-ec2 | running     | t3.medium   |
| i-c8a8e189073cd727f | devops-tech-test-dev-second-ec2 | running     | t3.small    |
| i-e9de47e206bc14a4e | 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[0].{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="MyVerySecure10$Secret"
export AWS_ACCESS_KEY_ID="test"
export AWS_SECRET_ACCESS_KEY="test"
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap/terraform$ terraform plan -var-file=vars/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=3208642683]

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    = "arn:aws:iam::aws:policy: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_taskRole will be created
resource "aws_iam_role" "ecs_taskRole" {
  + arn      = (known after apply)
  + assume_role_policy = jsonencode(
      + Statement = [
          + {
              + Action  = "sts:AssumeRole"
              + Effect  = "Allow"
              + Principal =

```

```

udaiali@INLAPSTD672:~$ terraform apply
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.iam_policy: 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_ssm_parameter.db_password: Creation complete after 5s [id=/dev/devops-tech-test/db_password]
aws_iam_policy.ssa_read: Creation complete after 5s [id=arn:aws:iam::000000000000:policy/devops-tech-test-dev-ssm-read]
aws_iam_role.eks_task_role: Creation complete after 5s [id=devops-tech-test-dev-eks-task-role]
aws_iam_role_policy.attachment.attach_ssm: Creation complete after 8s [id=devops-tech-test-dev-ecs-task-role-20260218070106970200000001]
aws_vpc.main: Still creating... [10s elapsed]
aws_vpc.main: Creation complete after 15s [id=vpc-8ab7cd7a326306f9]
aws_internet_gateway.igw: Creating...
aws_subnet.public: Creating...
aws_route_table.publicrt: Creating...
aws_instance.ec2["second-ec2"]: Creating...
aws_instance.ec2["third-ec2"]: Creating...
aws_route_table_publicrt: Creation complete after 8s [id=rtb-045d0d732a881fb4]
aws_route_table_association.prt: Creation complete after 8s [id=rta-bassos-7ccfae05f078c4481]
aws_s3_bucket.lifecycle_configuration.artifacts: Still creating... [20s elapsed]
aws_instance.ec2["first-ec2"]: Creating...
aws_instance.ec2["third-ec2"]: Still creating... [10s elapsed]
aws_instance.ec2["first-ec2"]: Creation complete after 10s [id=rtb-045d0d732a881fb4]
aws_instance.ec2["second-ec2"]: Creation complete after 10s [id=rtb-045d0d732a881fb4]
aws_instance.ec2["third-ec2"]: Creation complete after 10s [id=rtb-045d0d732a881fb4]
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-8ab7cd7a326306f9"

```

```

udaiali@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.prt
aws_s3_bucket.artifacts
aws_s3_bucket.lifecycle_configuration.artifacts
aws_ssm_parameter.db_password
aws_subnet.public
aws_vpc.main
udaiali@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.

```

udaiali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$ docker build -t local/image-myapi:latest .
[*] Building 57.7s (12/12) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] transfer Dockerfile: 582B
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [internal] load .dockerrcignore
=> [internal] transfer context: 758
(1/1) FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d1b945588d0a35cb9bc4b8ca09d9e
--> resolve docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d1b945588d0a35cb9bc4b8ca09d9e
=> [internal] load build context
=> [internal] transfer 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 tar image
=> exporting layers
=> exporting manifest sha256:5cd32a977ce600b0a4e3d9a6f1727806fae4ec37ff6f013eac8ffc8abfb88b9507
=> exporting config sha256:225c80881d3cd171b7a285cf0385ccb9f11c768da9799c19c9e8bf9ebe8813
=> exporting attestation manifest sha256:c949307fea521c8acc138a2d5fc3b0291f5c62124a94fe46ef7b6af03464c0f
=> exporting manifest list sha256:0682b2561c07792772248d6501e2de47e6ca6366c28c97bc251b2befed2c9a
=> naming to docker.io/local/image-myapi:latest
=> unpacking to docker.io/local/image-myapi:latest

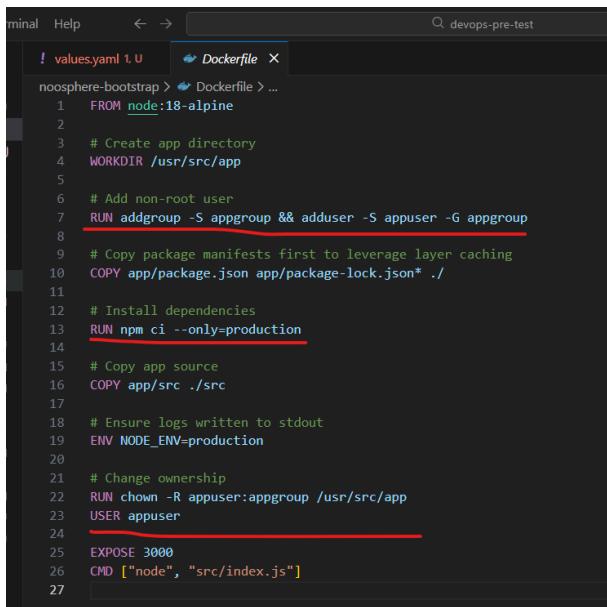
```

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



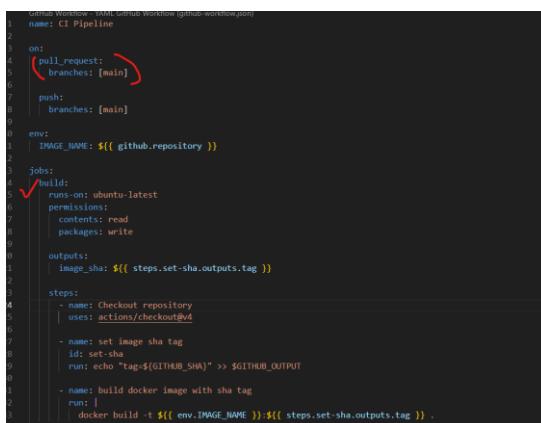
```
mininal Help ← → Dockerfile x
values.yaml 1.0 noosphere-bootstrap > Dockerfile > ...
1 FROM node:18-alpine
2
3 # Create app directory
4 WORKDIR /usr/src/app
5
6 # Add non-root user
7 RUN addgroup -S appgroup && adduser -S appuser -G appgroup
8
9 # Copy package manifests first to leverage layer caching
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

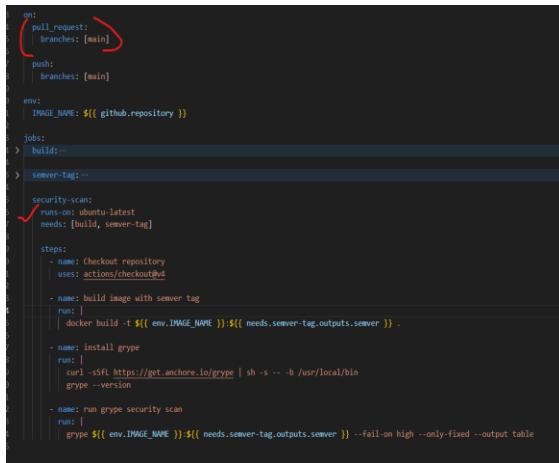
On Pull request Image builds in workflow.



```
name: CI Pipeline
on:
  pull_request:
    branches: [main]
  push:
    branches: [main]
env:
  IMAGE_NAME: ${{ github.repository }}
jobs:
  build:
    runs-on: ubuntu-latest
    permissions:
      contents: read
      packages: write
    output:
      image_sha: ${{ steps.set-sha.outputs.tag }}
    steps:
      - name: Checkout repository
        uses: actions/checkout@v4
      - name: set image sha tag
        id: set-sha
        run: echo "tag=${GITHUB_SHA}" >> $GITHUB_OUTPUT
      - name: build docker image with sha tag
        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]

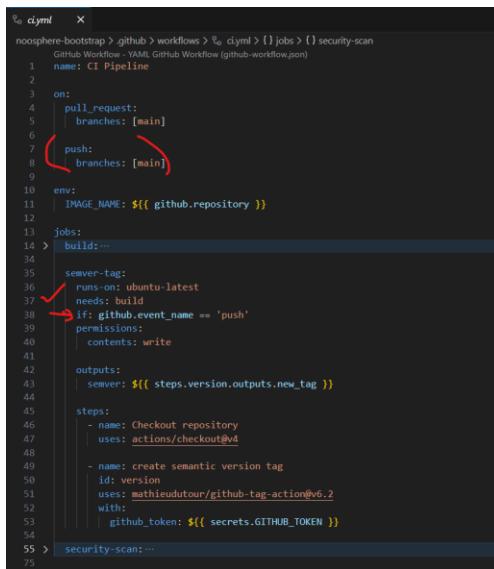
env:
  IMAGE_NAME: ${github.repository}

jobs:
  > build:...
  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 - YAML 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: mathieuendutour/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

```
udaiyali@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
service/api-service   ClusterIP  10.97.121.15  <none>      80/TCP   114m
deployment.apps/api-deployment   1/1     1         1          145m
replicaset.apps/api-deployment-5bf5c467d9   1       1         1          145m
udaiyali@INLAPSTD672:/mnt/d/devops-pre-test/noosphere-bootstrap$
```

Service Exposure

Port-forwarded service, endpoints verified with curl.

```
udaiyali@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 [::]:8080 -> 3000
Handling connection for 8080
Handling connection for 8080
```

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

Security Context

Container runs as non-root with restricted permissions.

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

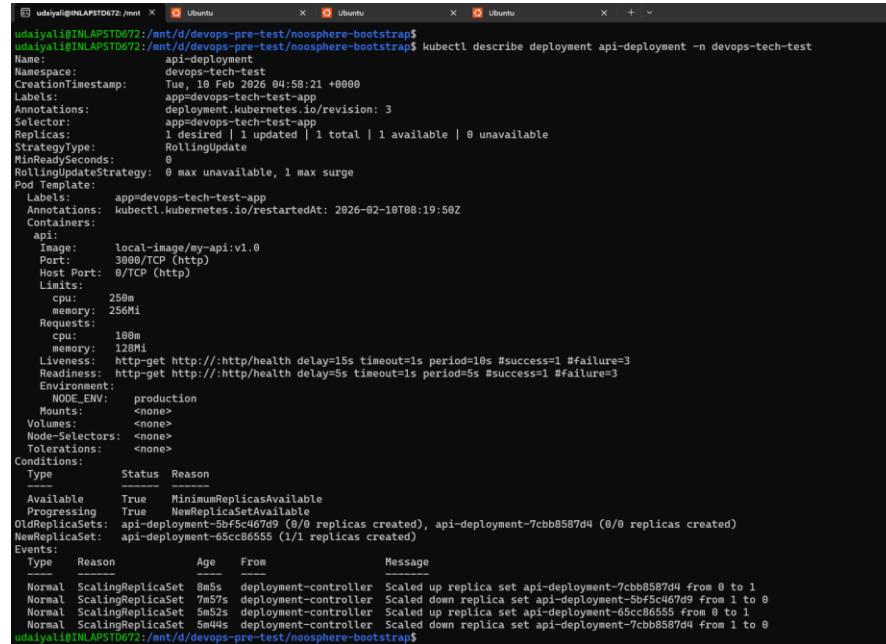
Atomic commits maintained for clarity and traceability.

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.



```
u@u@INLAPSTD672:~$ kubectl get hpa  
NAME      REFERENCE   CURRENT   TARGETS  
api       api-deployment   1         250m  
u@u@INLAPSTD672:~$ kubectl get deployment api-deployment -n devops-tech-test  
Name:           api-deployment  
Namespace:     devops-tech-test  
CreationTimestamp: Tue, 10 Feb 2026 04:58:21 +0000  
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:       3000/TCP (http)  
      Host Port:  8/TCP (http)  
      Limits:  
        cpu: 250m  
        memory: 256Mi  
      Requests:  
        cpu: 100m  
        memory: 128Mi  
      Livecheck:  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-7ccb8587d4 (0/0 replicas created)  
NewReplicaSet:  api-deployment-d5cc86555 (1/1 replicas created)  
Events:  
  Type  Reason  Age  From  Message  
  ----  ----  --  --  --  
  Normal  ScalingReplicaSet  8m5s  deployment-controller  Scaled up replica set api-deployment-7ccb8587d4 from 0 to 1  
  Normal  ScalingReplicaSet  7m57s  deployment-controller  Scaled down replica set api-deployment-5bf5c467d9 from 1 to 0  
  Normal  ScalingReplicaSet  8m52s  deployment-controller  Scaled up replica set api-deployment-d5cc86555 from 0 to 1  
  Normal  ScalingReplicaSet  8m44s  deployment-controller  Scaled down replica set api-deployment-7ccb8587d4 from 1 to 0
```

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/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:v1.8
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.

```

1 // Create a Registry which registers the metrics
2 const register = new ClientMetricsRegistry();
3
4 // Create a Registry which registers the metrics
5 const register = new ClientMetricsRegistry();
6
7 // Default metrics (CPU, memory, etc)
8 client.collectDefaultMetrics(register);
9
10 // Custom metrics
11 const httpRequestDurationSeconds = new ClientHistogram({
12   name: "http.request.duration.seconds",
13   help: "Duration of HTTP requests in seconds",
14   buckets: [0.1, 0.5, 1, 1.5, 2, 5],
15 });
16
17 const httpStatusCodes = new ClientCounter({
18   name: "http.request.error",
19   help: "Total value of HTTP request errors",
20   labels: ["method", "route", "status_code"],
21 });
22
23
24 register.registerMetric(httpRequestDurationSeconds);
25 register.registerMetric(httpStatusCodes);
26
27 // Middleware to measure request latency and errors
28 function metricMiddleware(req, res, next) {
29   const end = httpRequestDurationSeconds.start();
30   res.on("finish", () => {
31     end();
32     const { method, route, status_code: resStatusCode } = req;
33     if (res.statusCode >= 400) {
34       httpStatusCodes.labels({ method, route, status_code: res.StatusCode }).inc();
35     }
36   });
37   next();
38 }
39
40 module.exports = { register, metricMiddleware };

```

Deployed Prometheus & Grafana using Helm

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

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

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:~$ 

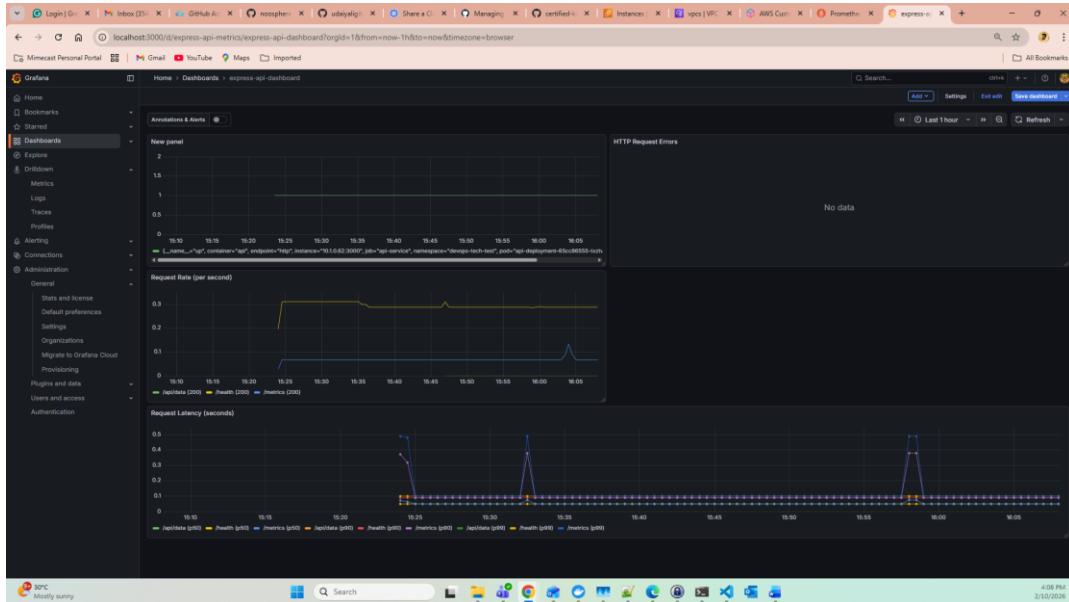
```

Prometheus discovered and scraped API service

The screenshot shows the Prometheus interface with several discovered targets listed:

- serviceMonitor/tech-test-api-service-monitor/0**:
 - Endpoint: <http://10.0.1.45:3000/metrics>
 - Labels: container="api", endpoint="http", instance="10.0.1.45:3000", job="api-service", namespace="devops-tech-test", pod="api-deployment-65c80555-lshtz", service="api-service".
 - Last scrape: 48.746s ago
 - State: UP
- serviceMonitor/monitoring-grafana/0**:
 - Endpoint: <http://10.0.1.30:3000/metrics>
 - Labels: container="grafana", endpoint="http-web", instance="10.0.1.30:3000", job="monitoring-grafana", namespace="monitoring", pod="monitoring-grafana-5f8cfc666-mxkbt", service="monitoring-grafana".
 - Last scrape: 1m 3.33s ago
 - State: UP
- serviceMonitor/monitoring/monitoring-kube-prometheus-alertmanager/0**:
 - Endpoint: <http://10.0.43:9093/metrics>
 - Labels: container="alertmanager", endpoint="http-web", instance="10.0.43:9093", job="monitoring-kube-prometheus-alertmanager", namespace="monitoring", pod="alertmanager monitoring-kube-prometheus-alertmanager-0", service="monitoring-kube-prometheus-alertmanager".
 - Last scrape: 55.335s ago
 - State: UP
- serviceMonitor/monitoring/monitoring-kube-prometheus-alertmanager/1**:
 - Endpoint: <http://10.0.43:9090/metrics>
 - Labels: container="config-reloader", endpoint="reloader-web", instance="10.0.43:9090", job="monitoring-kube-prometheus-alertmanager", namespace="monitoring", pod="alertmanager monitoring-kube-prometheus-alertmanager-0", service="monitoring-kube-prometheus-alertmanager".
 - Last scrape: 55.335s ago
 - State: UP

Created Grafana dashboard



Ensured persistence through restarts

```
ubuntu@INLAPOSTD672:~$ kubectl rollout restart deployment devops-tech-test-app -n devops-tech-test
Error from server (NotFound): deployments.apps "devops-tech-test-app" not found
ubuntu@INLAPOSTD672:~$ kubectl rollout restart deployment api-deployment -n devops-tech-test
deployment.apps/api-deployment restarted
ubuntu@INLAPOSTD672:~$ kubectl rollout restart deployment monitoring-grafana -n monitoring
deployment.apps/monitoring-grafana restarted
ubuntu@INLAPOSTD672:~$
```

