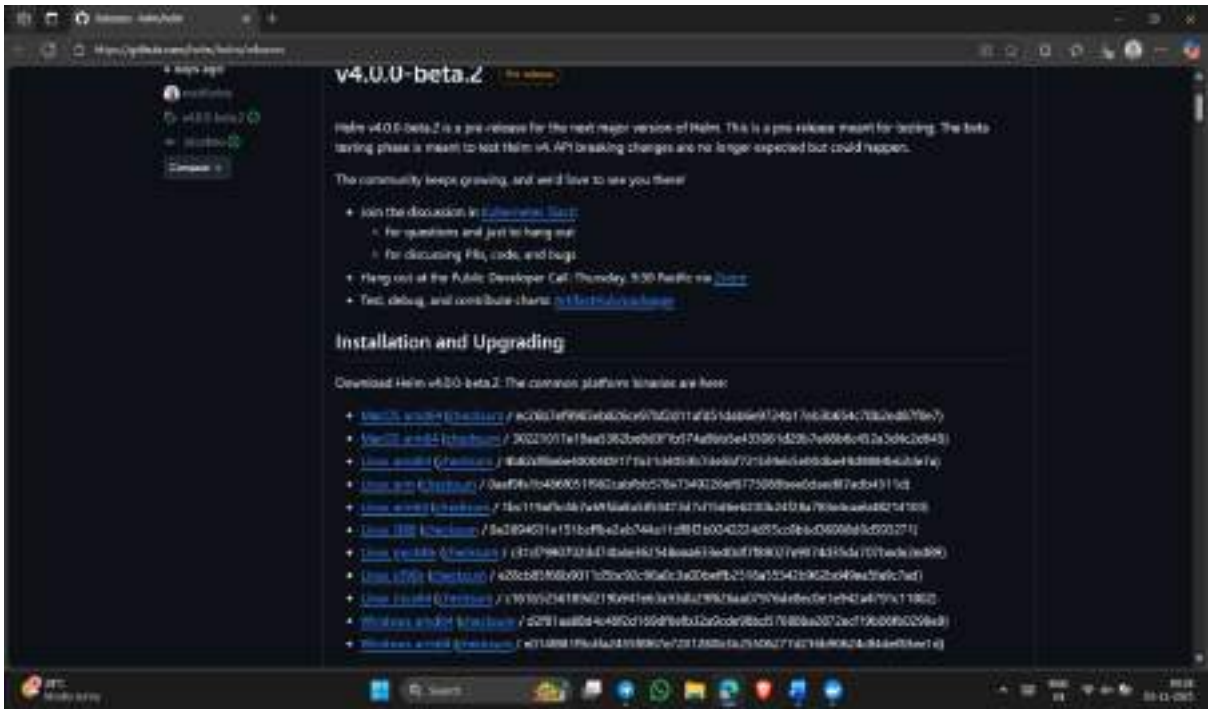
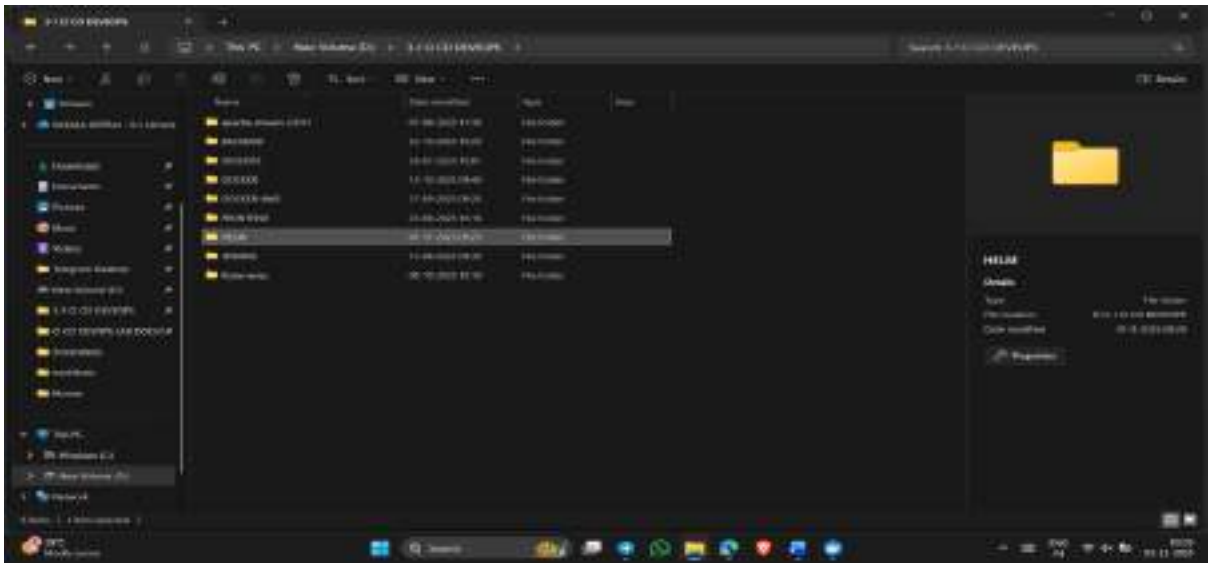
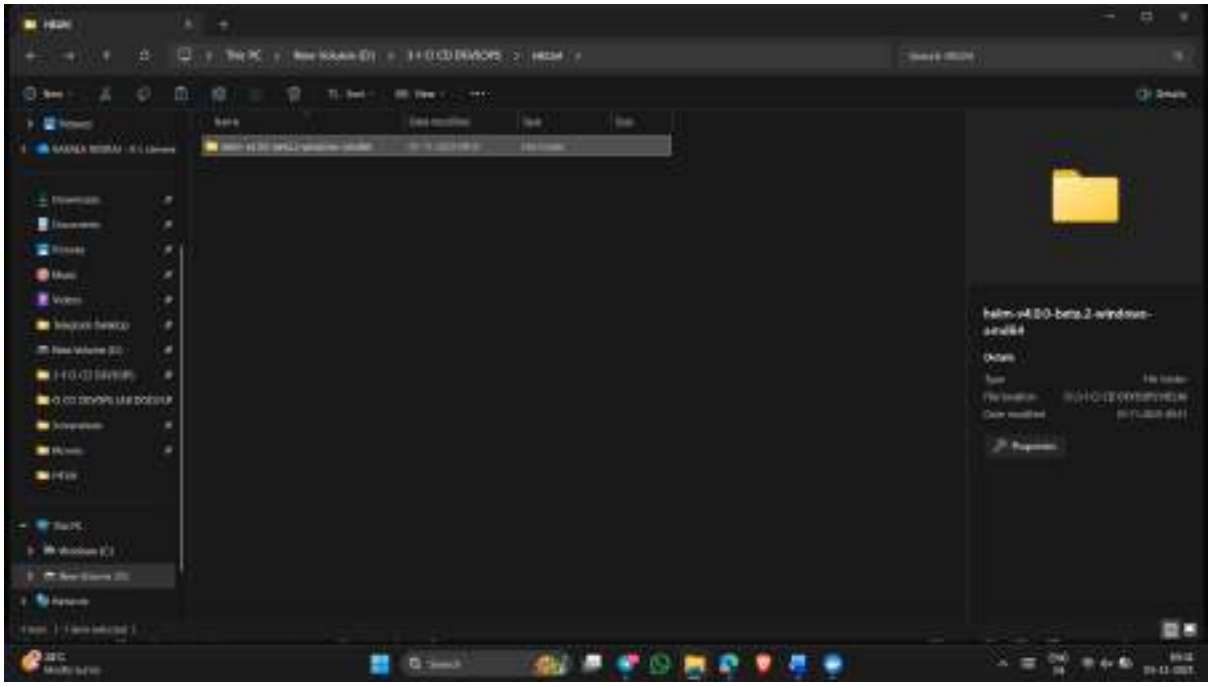
LAB 14

Kubernetes deployment using helm and ingress

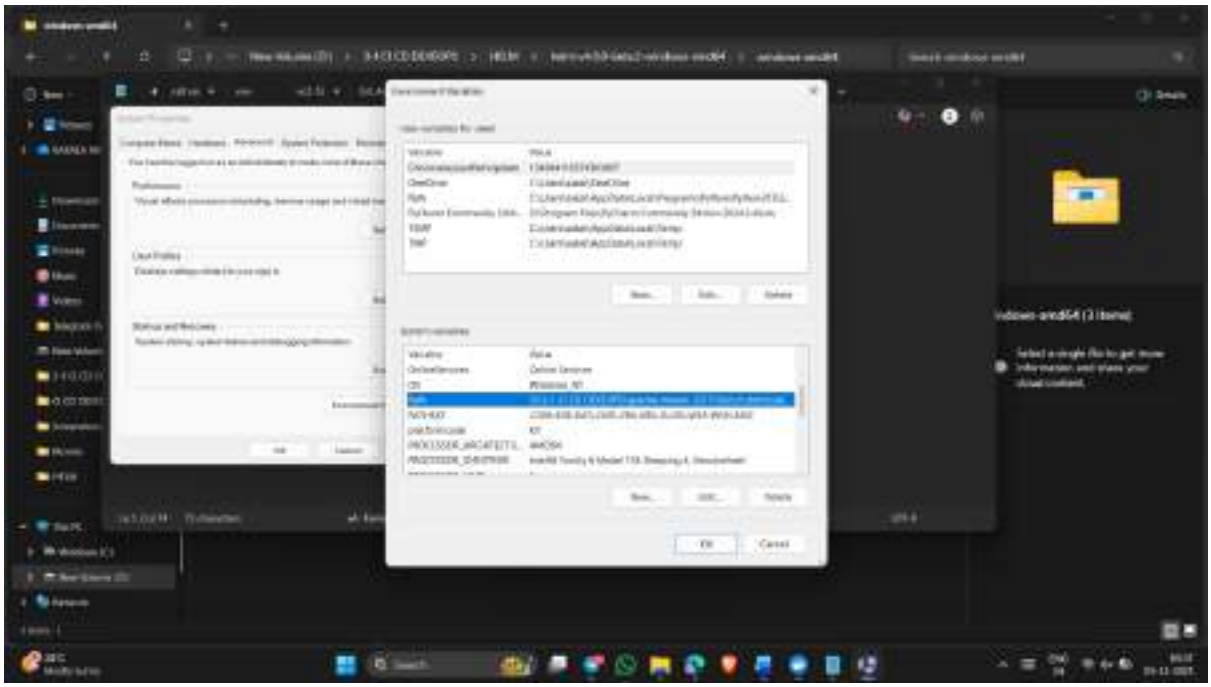
https://github.com/helm/helm/releases

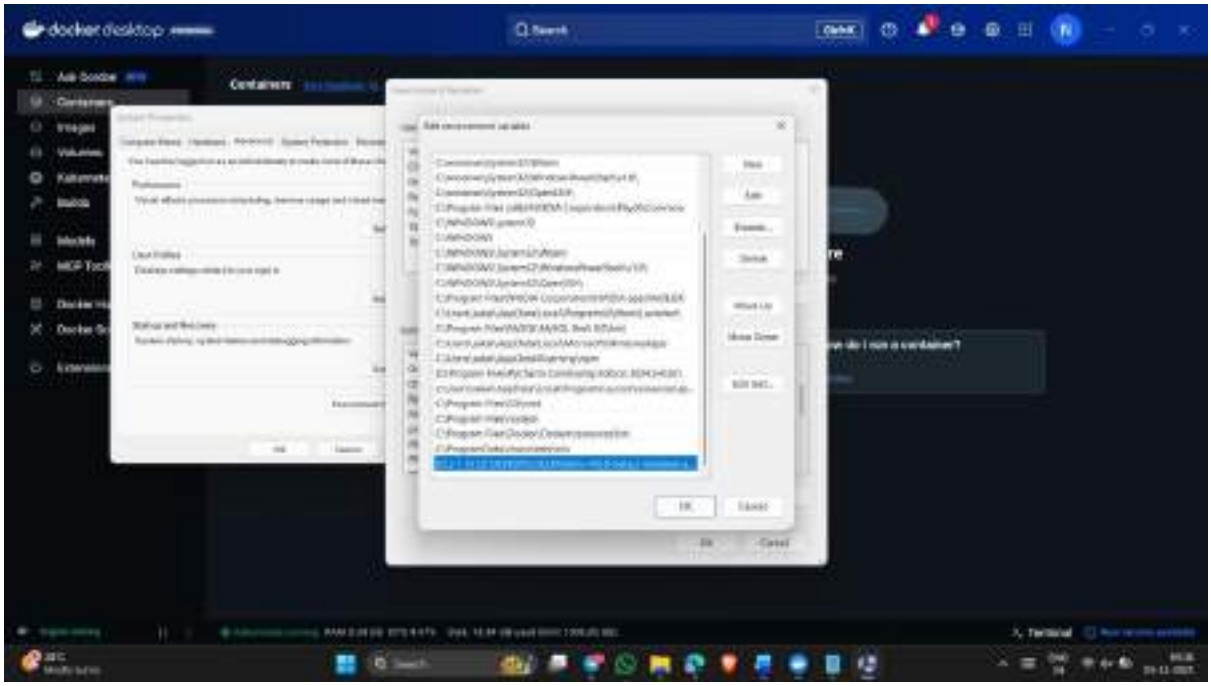


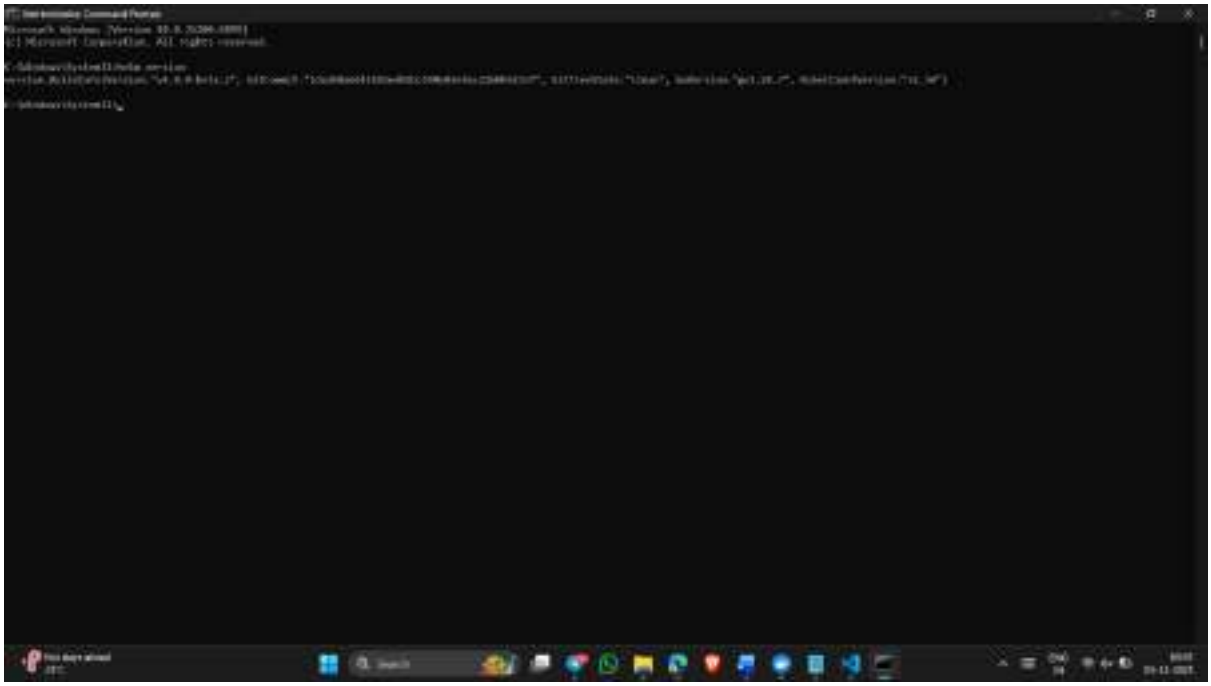


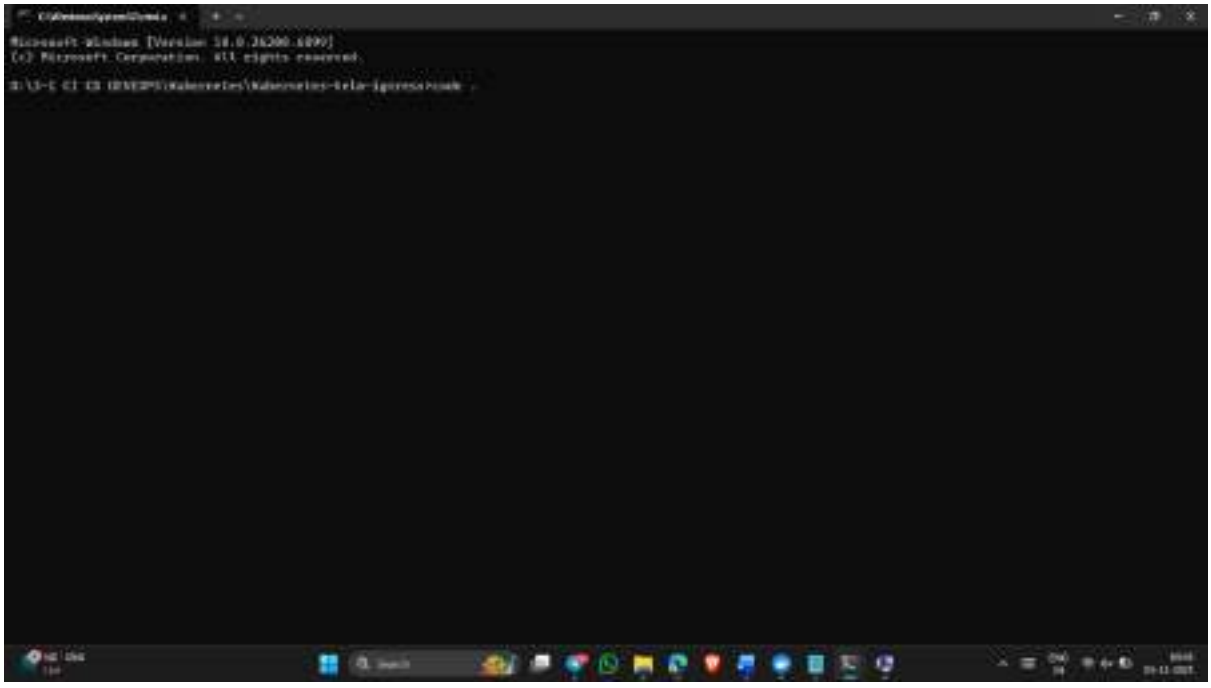


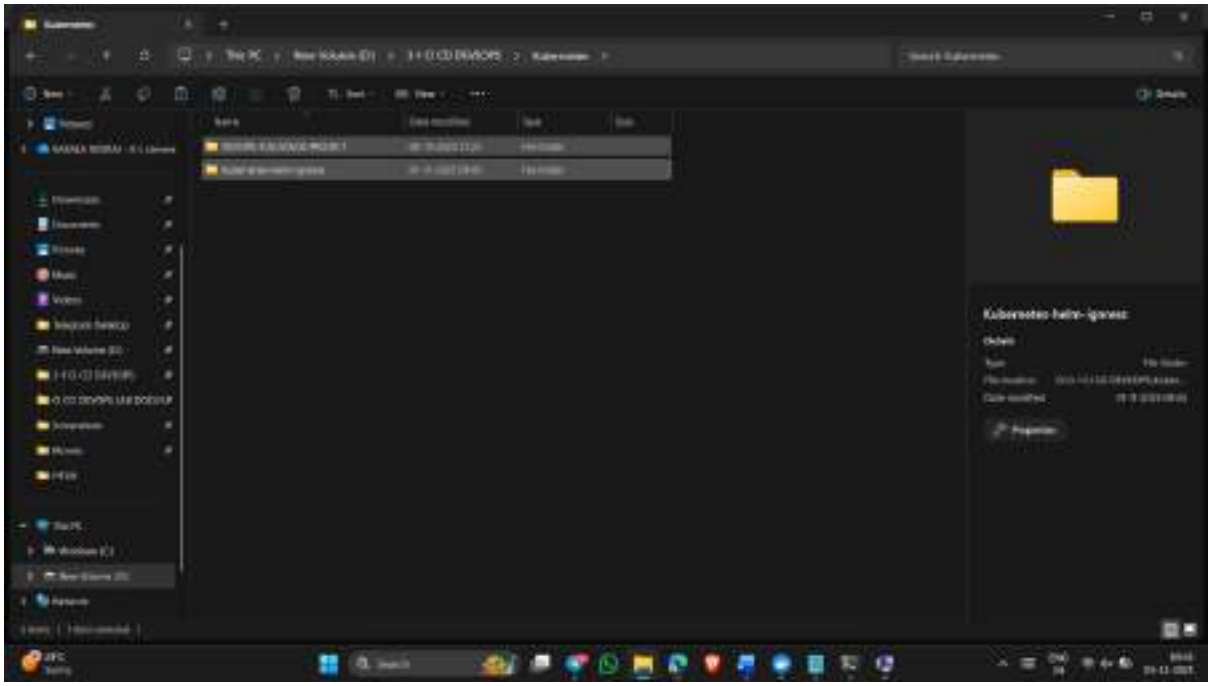
D:\3-1 CI CD DEVEOPS\HELM\helm-v4.0.0-beta.2-windows-amd64\windows-amd64

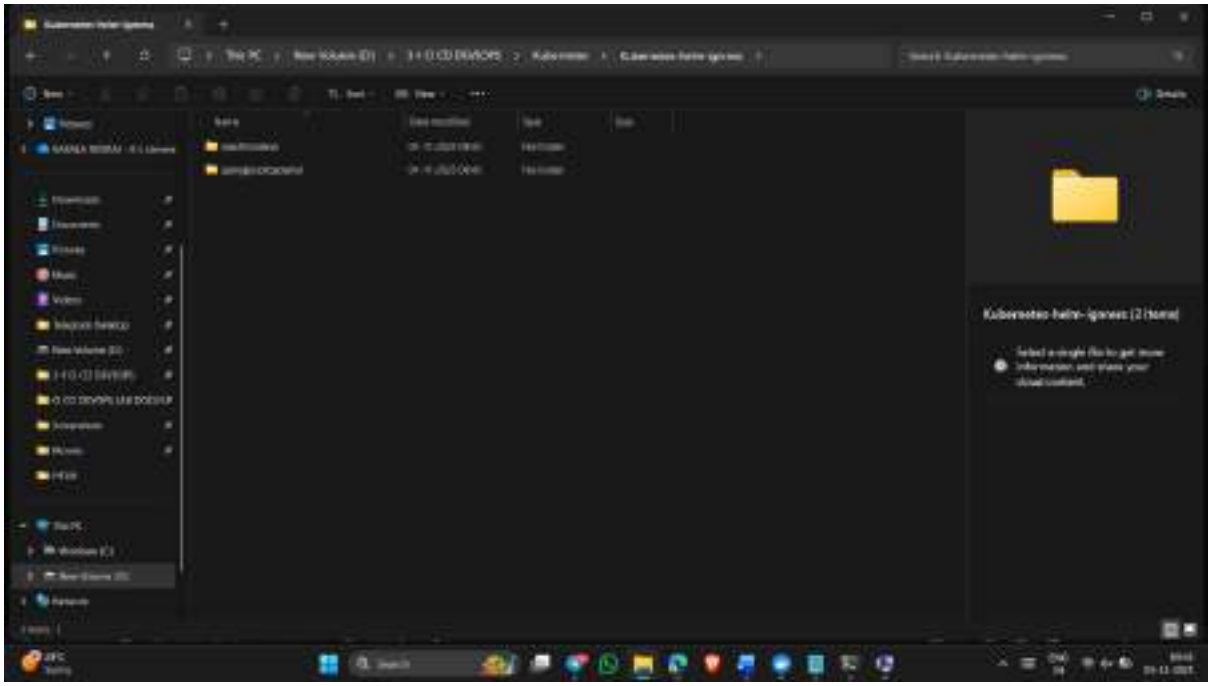


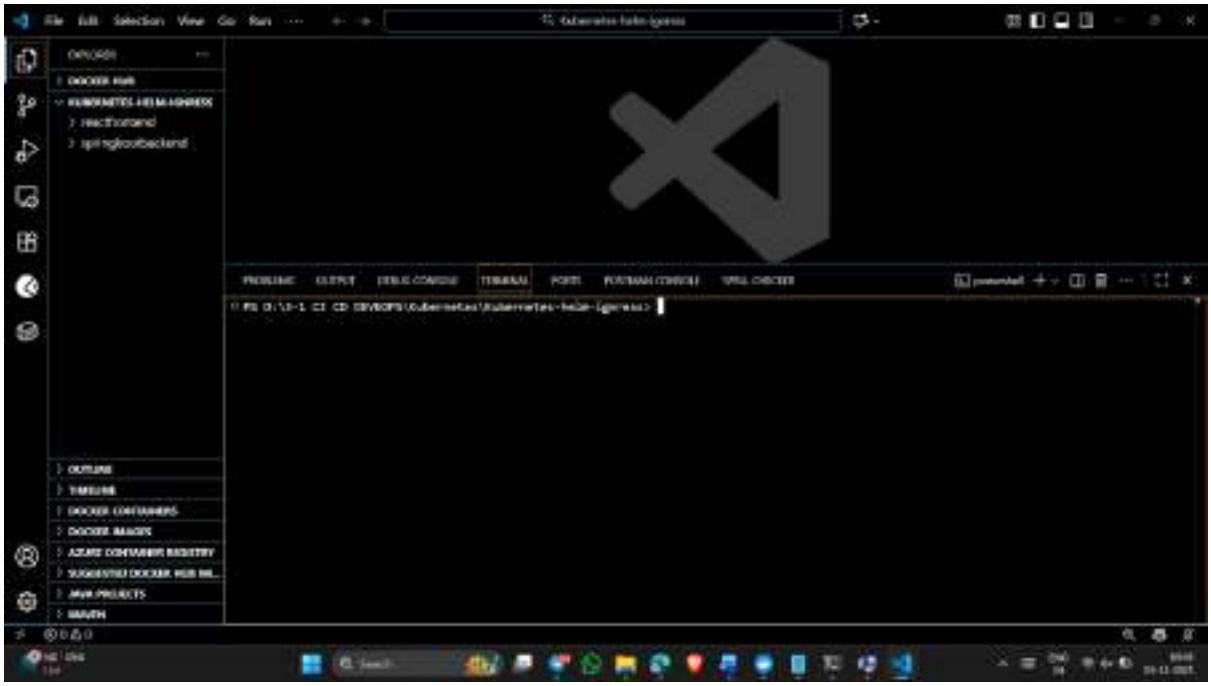

docker desktop
Containers



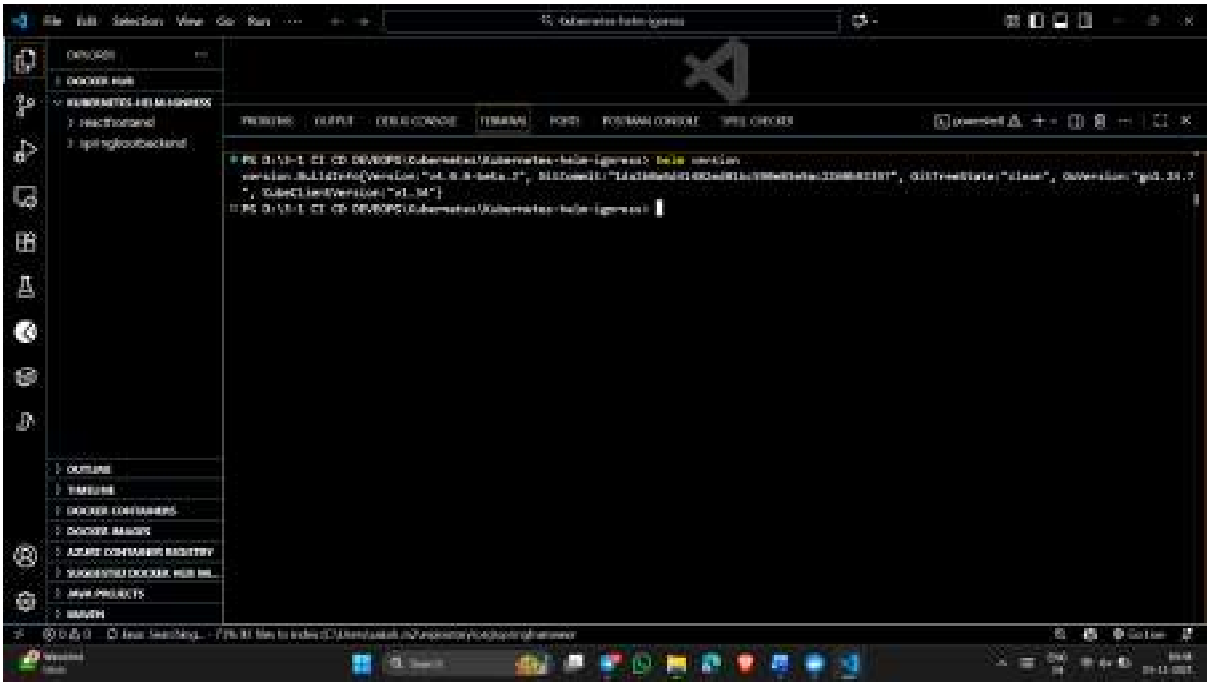

Microsoft Windows [Version 10.0.26200.6899]
(c) Microsoft Corporation. All rights reserved.

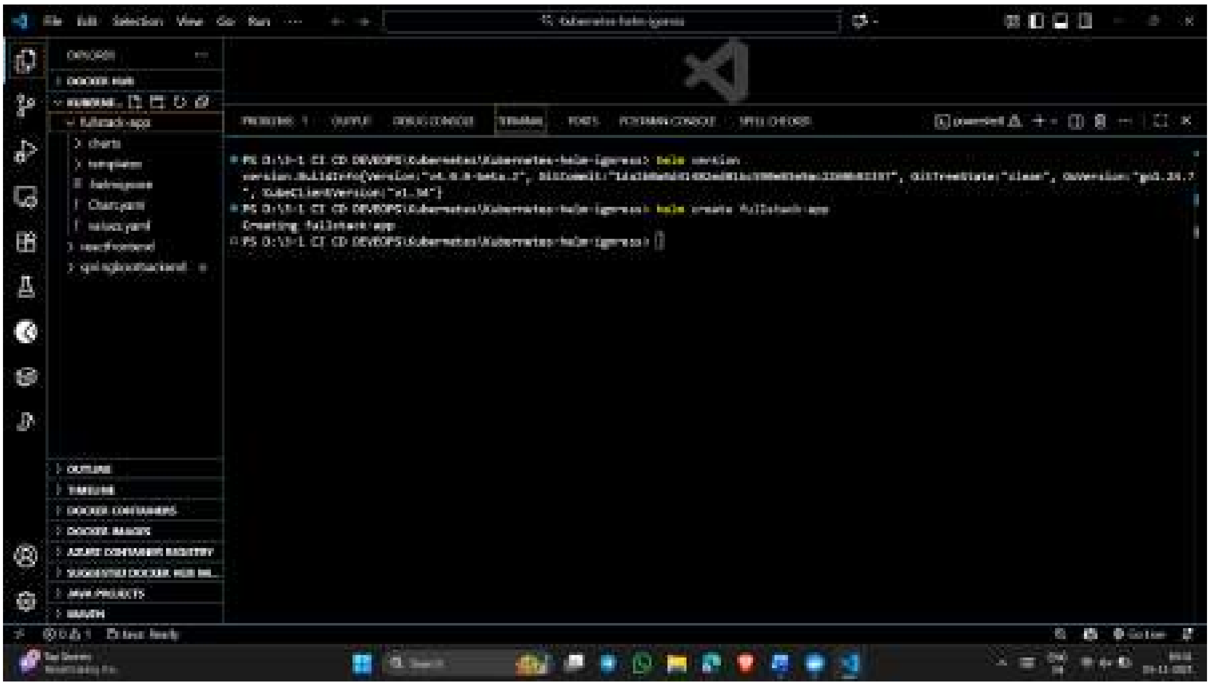


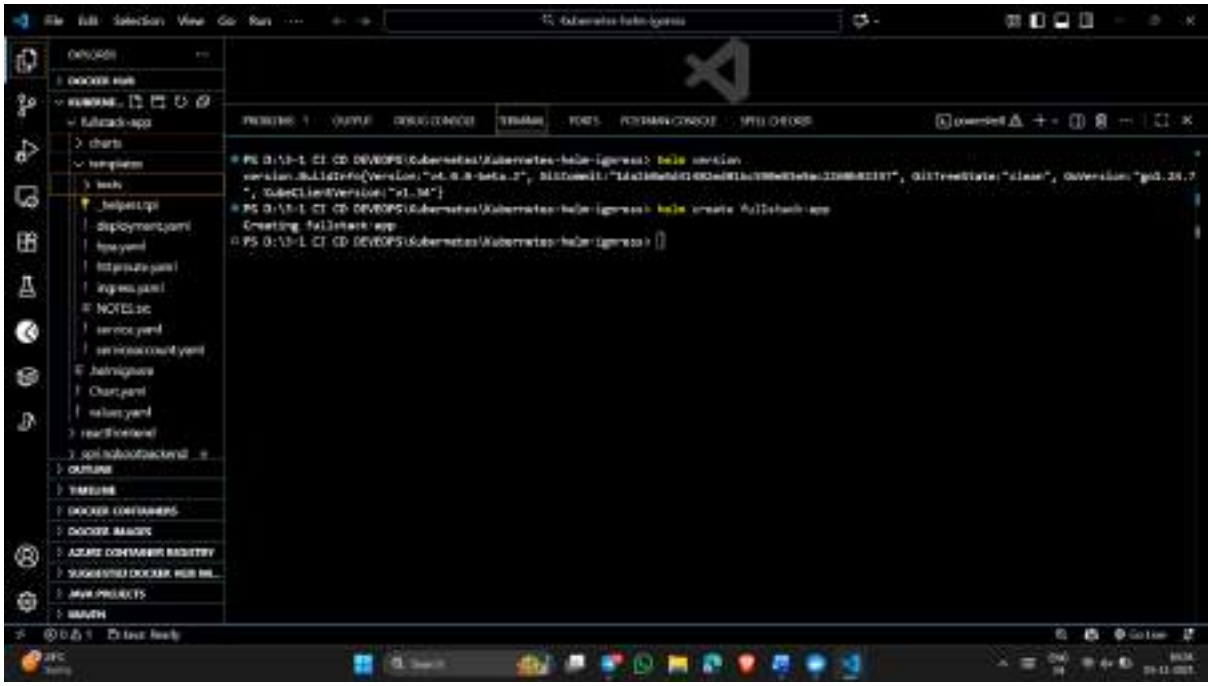




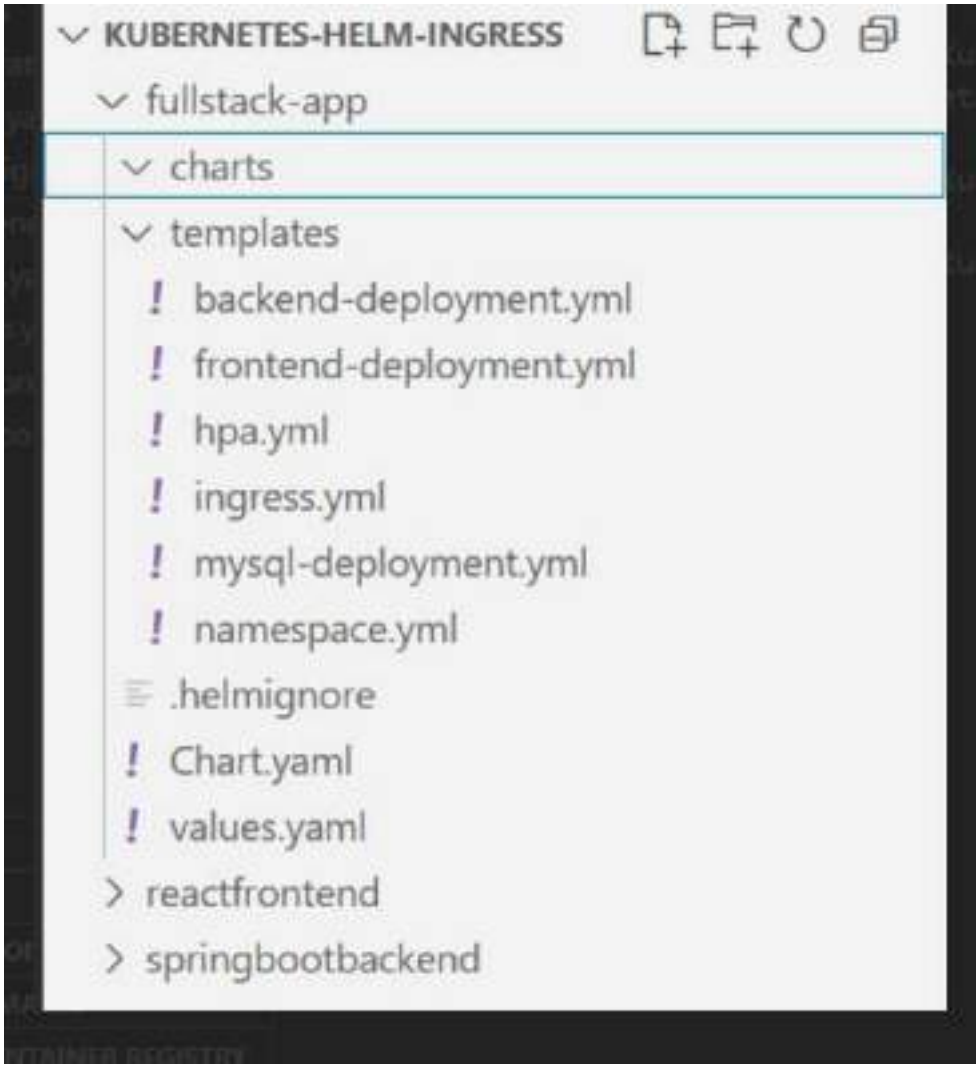
->Build docker images for both frontend and backend push them into docker hub

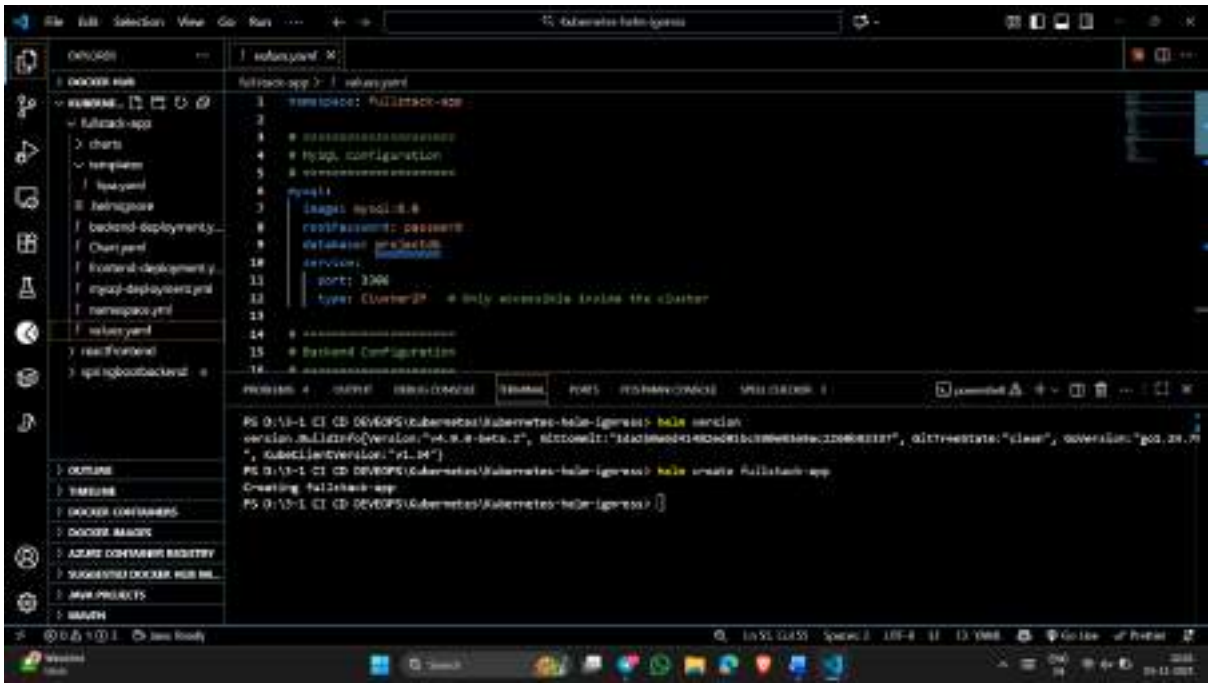

File Edit Selection View Go Run
DOCKER HUB
reactfrontend
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
AZURE CONTAINER REGISTRY
JAVA PROJECTS
MAVEN
TERMINAL
helm version

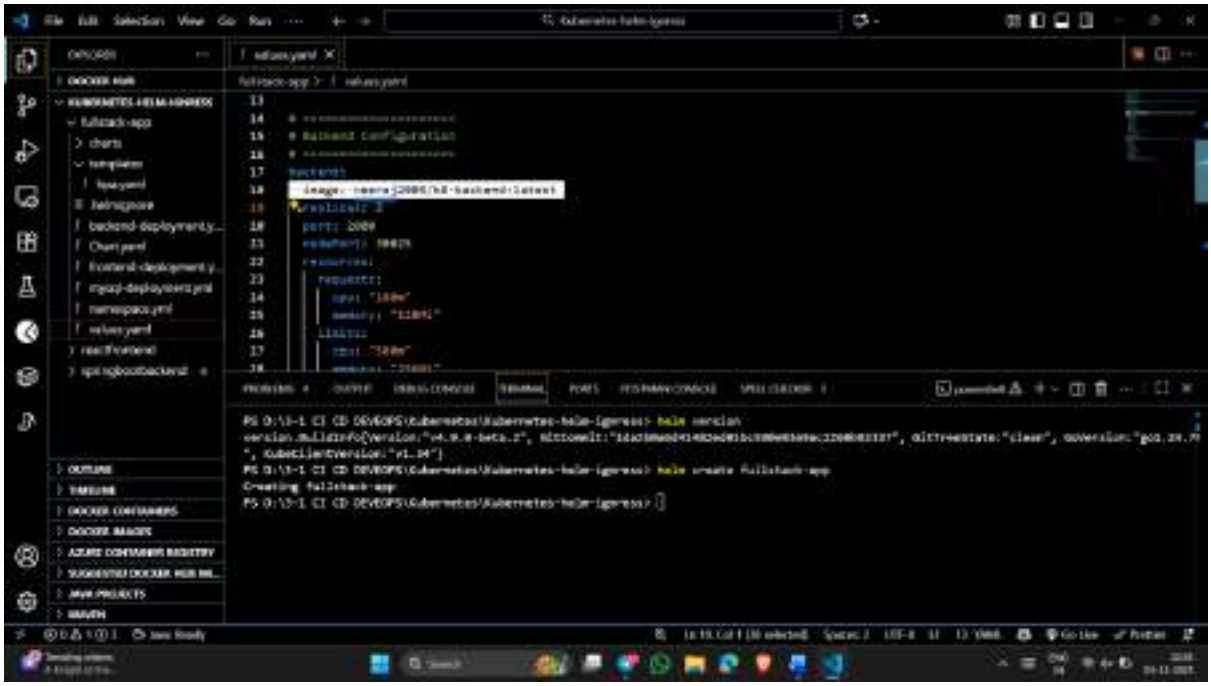

File Edit Selection View Go Run
DOCKER HUB
fullstack-app
charts
templates
reactfrontend
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
AZURE CONTAINER REGISTRY
JAVA PROJECTS
MAVEN
TERMINAL
helm version
helm create fullstack-app
Creating fullstack-app



->remove tests , helpers, deployment.yml,httproute.yml,nodes.txt,service.yml,seriveaccount.yml


KUBERNETES-HELM-INGRESS
fullstack-app
charts
templates
backend-deployment.yml
frontend-deployment.yml
hpa.yml
ingress.yml
mysql-deployment.yml
namespace.yml
.helmignore
Chart.yaml
values.yaml
reactfrontend
springbootbackend





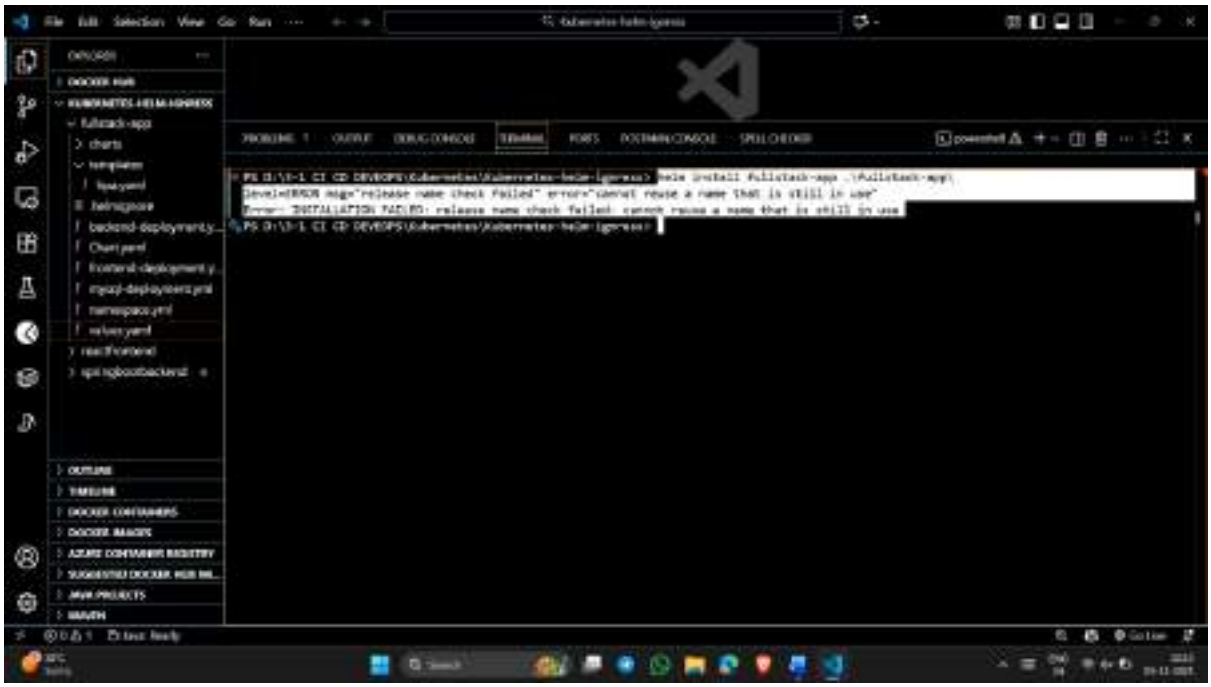
->helm install fullstack-app ./fullstack-app

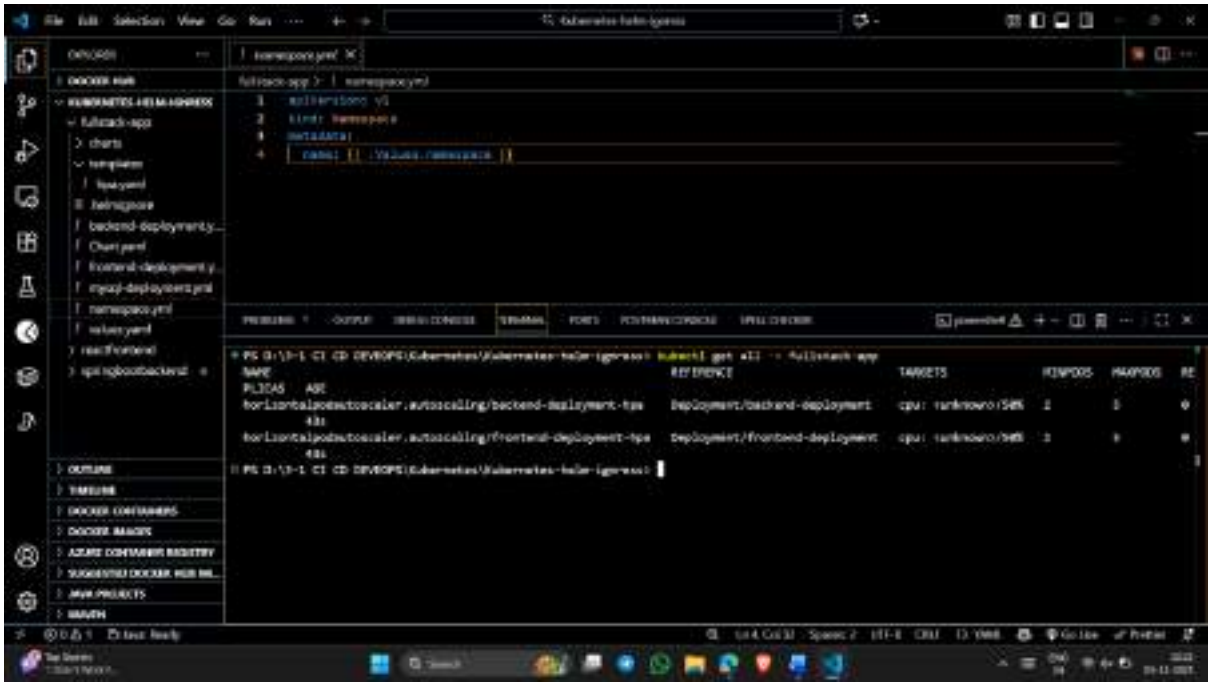
helm means cli tool

install means deploy a new release

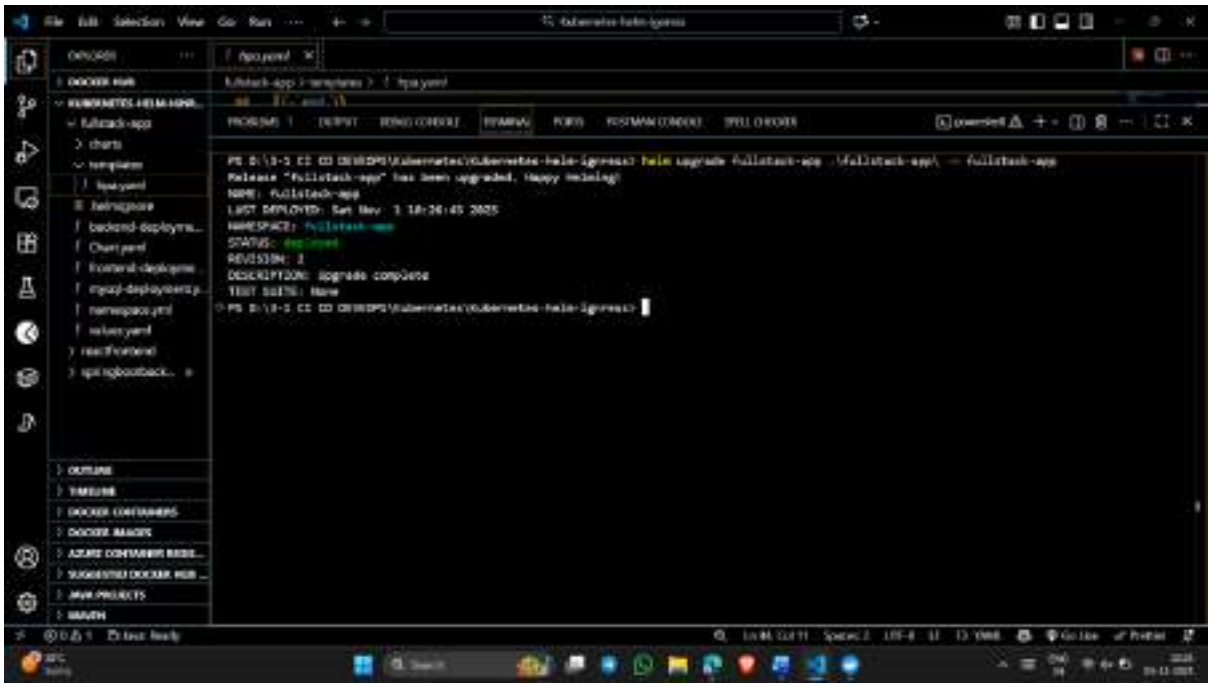
fullstack-app means name of the release name

./fullstack-app/ means helm directory

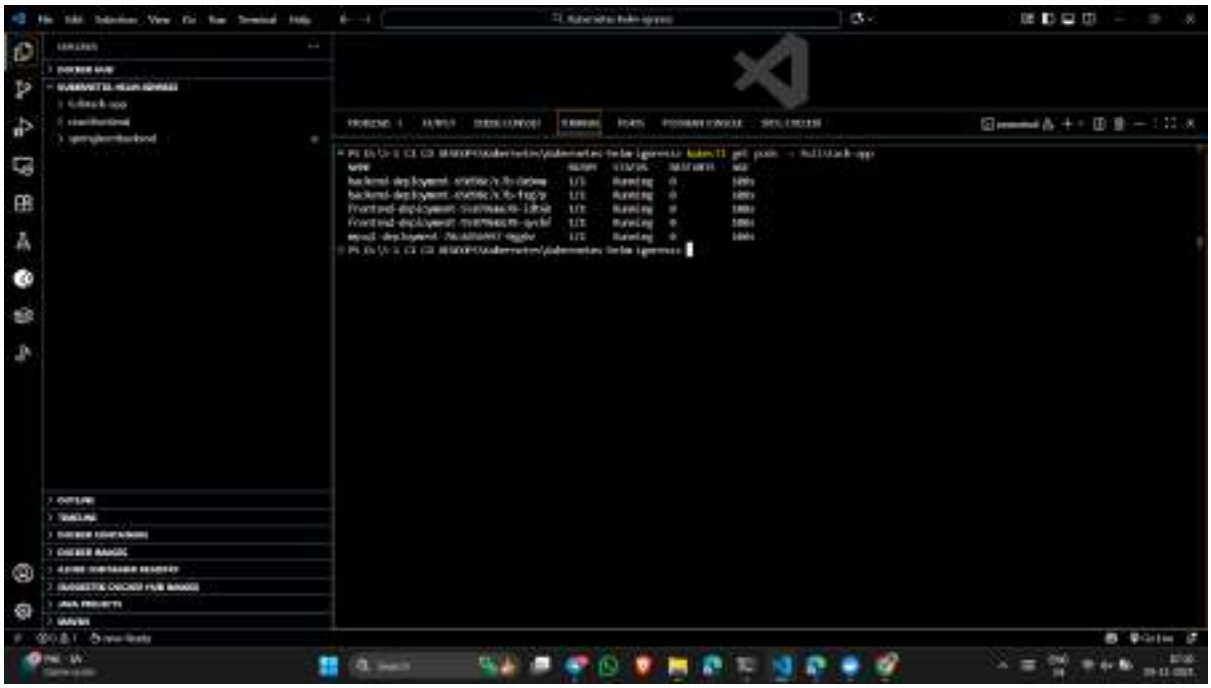


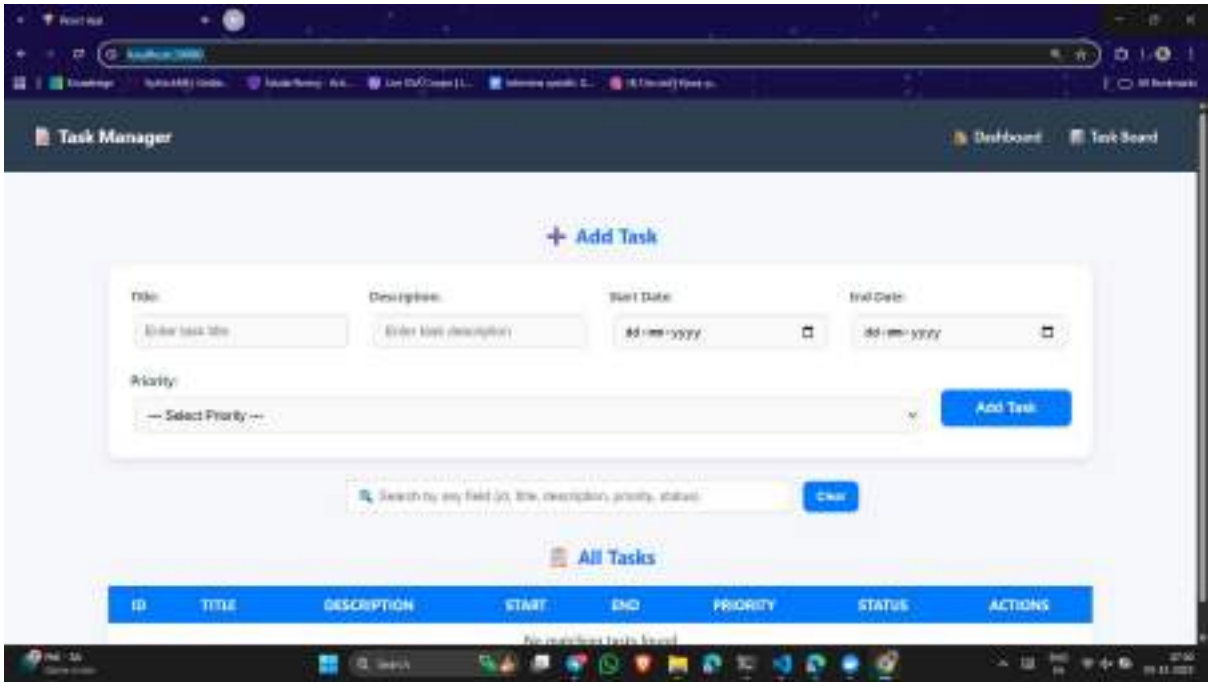


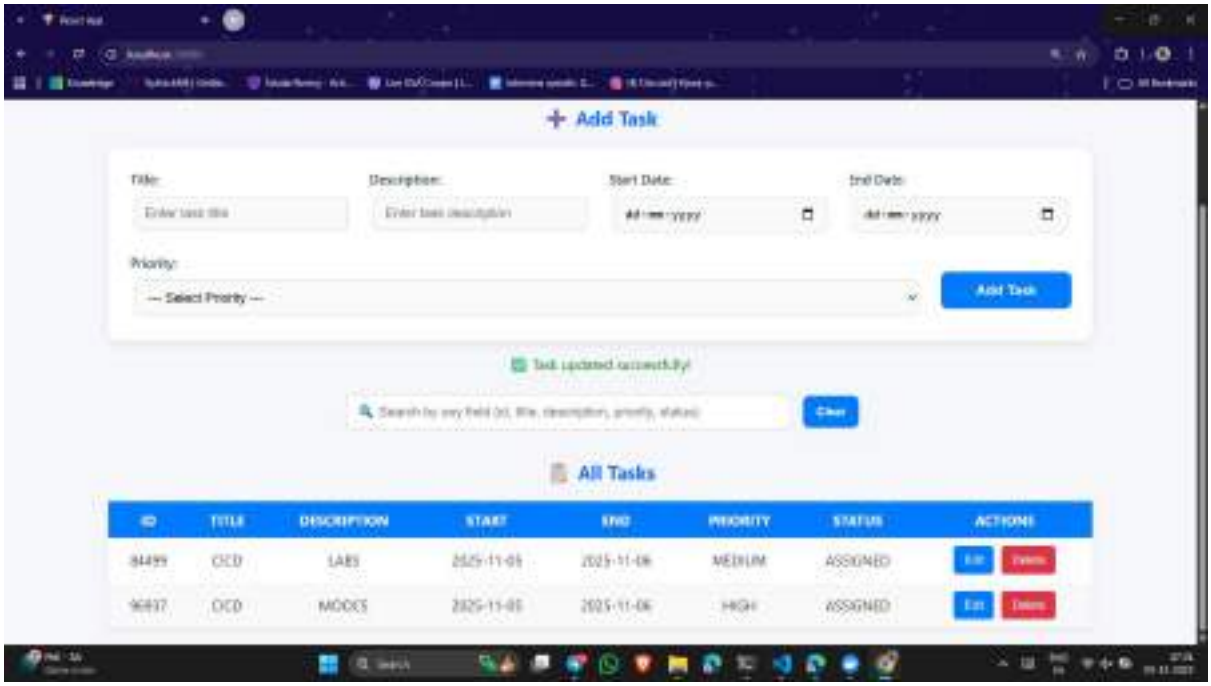
helm upgrade fullstack-app .\fullstack-app

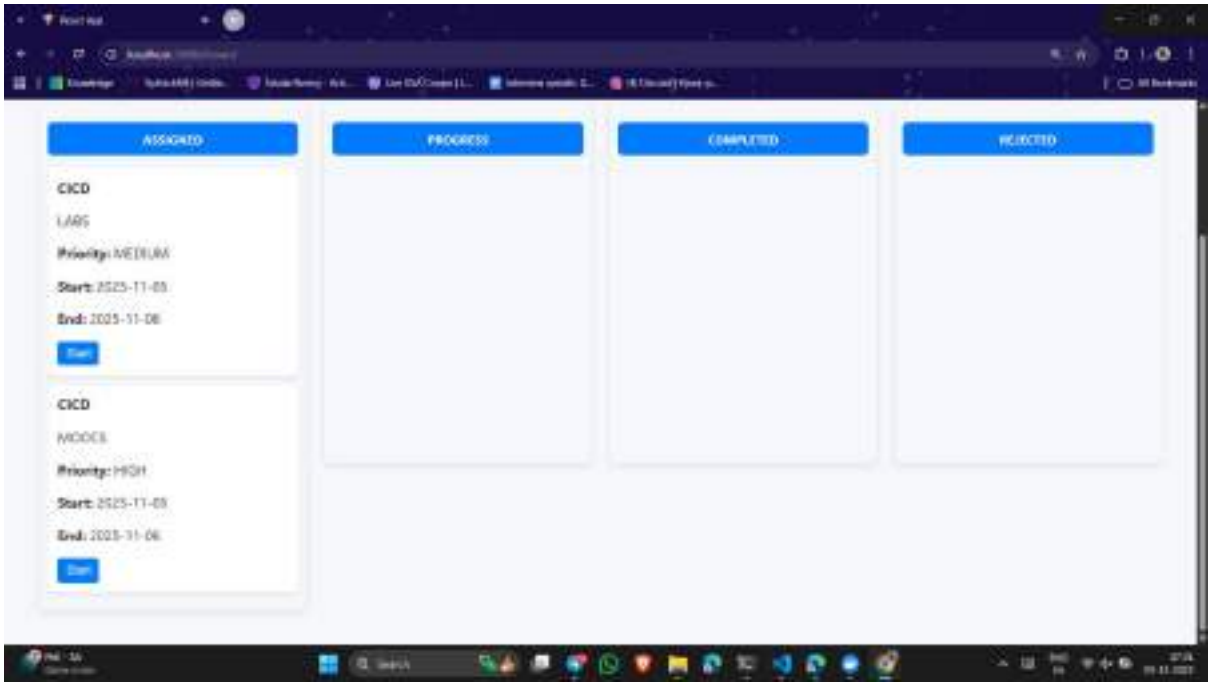


kubectl get all -n fullstack-app

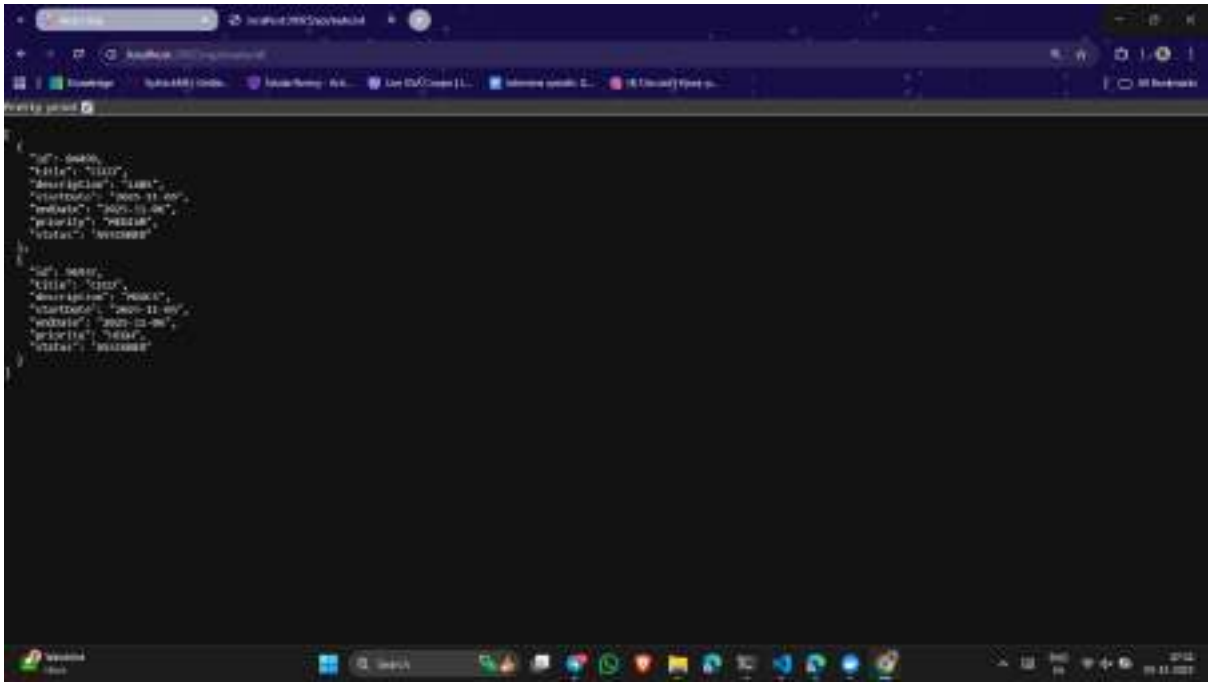


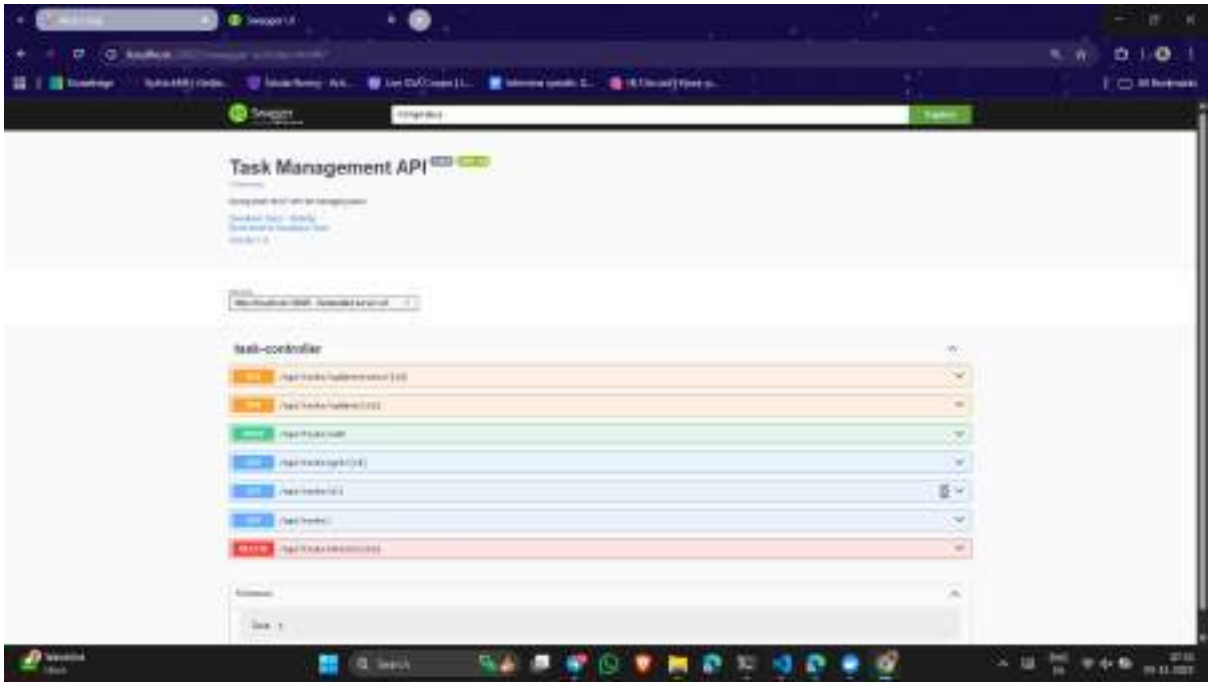

Task Manager
Dashboard
Task Board
Add Task
Title:
Description:
Start Date:
End Date:
dd-mm-yyyy
Priority:
--- Select Priority ---
Add Task
Clear
All Tasks
ID
TITLE
DESCRIPTION
START
END
PRIORITY
STATUS
ACTIONS

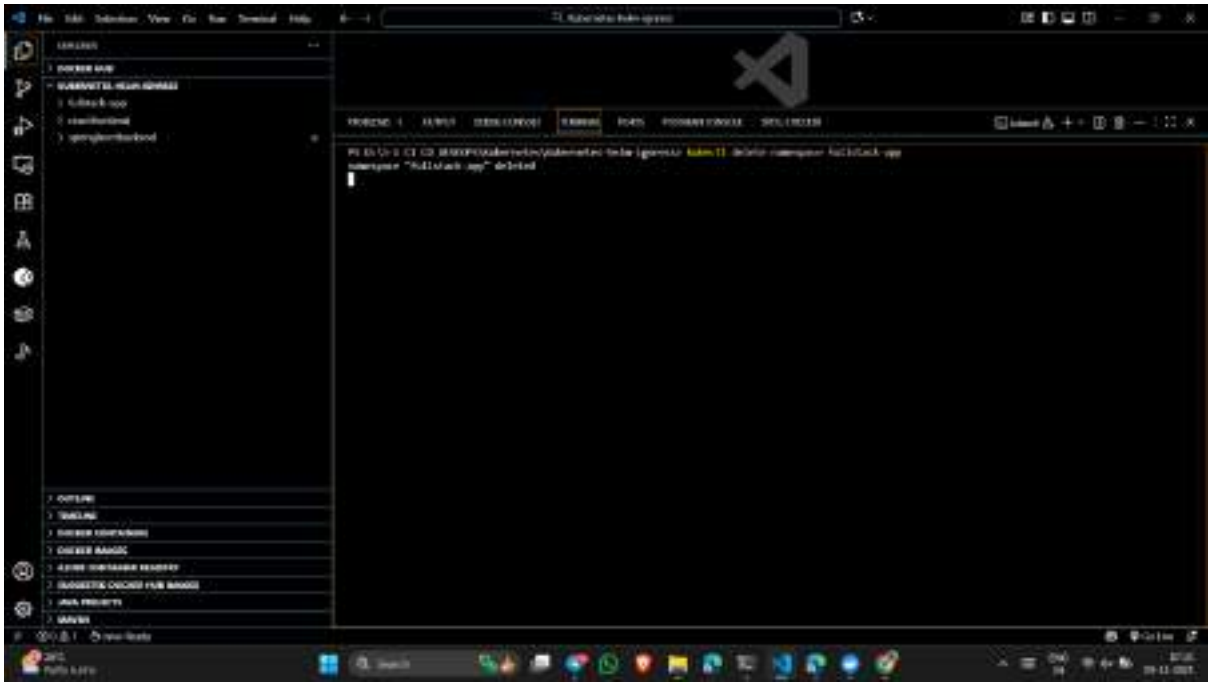

Add Task
Title:
Description:
Start Date:
End Date:
dd-mm-yyyy
Priority:
--- Select Priority ---
Add Task
Clear
All Tasks
ID
TITLE
DESCRIPTION
START
END
PRIORITY
STATUS
ACTIONS
84499
OCD
LABS
2025-11-05
2025-11-06
MEDIUM
ASSIGNED
Edit
Delete
96497
OCD
MOCKS
2025-11-05
2025-11-06
HIGH
ASSIGNED
Edit
Delete

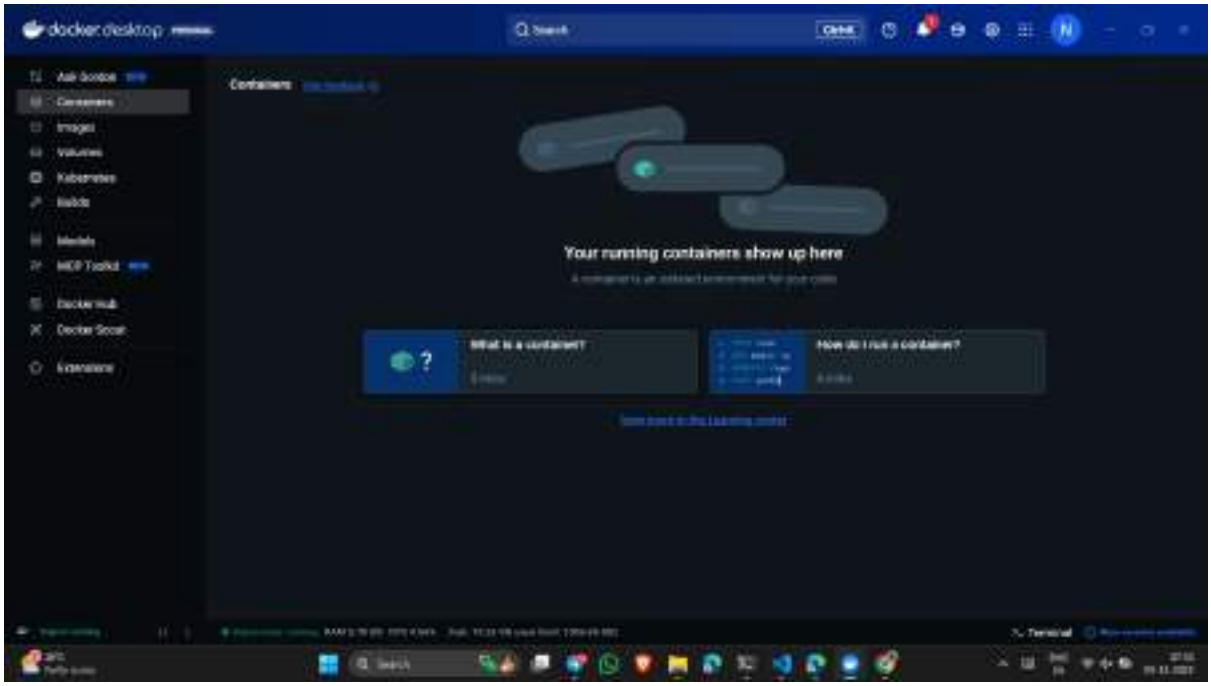


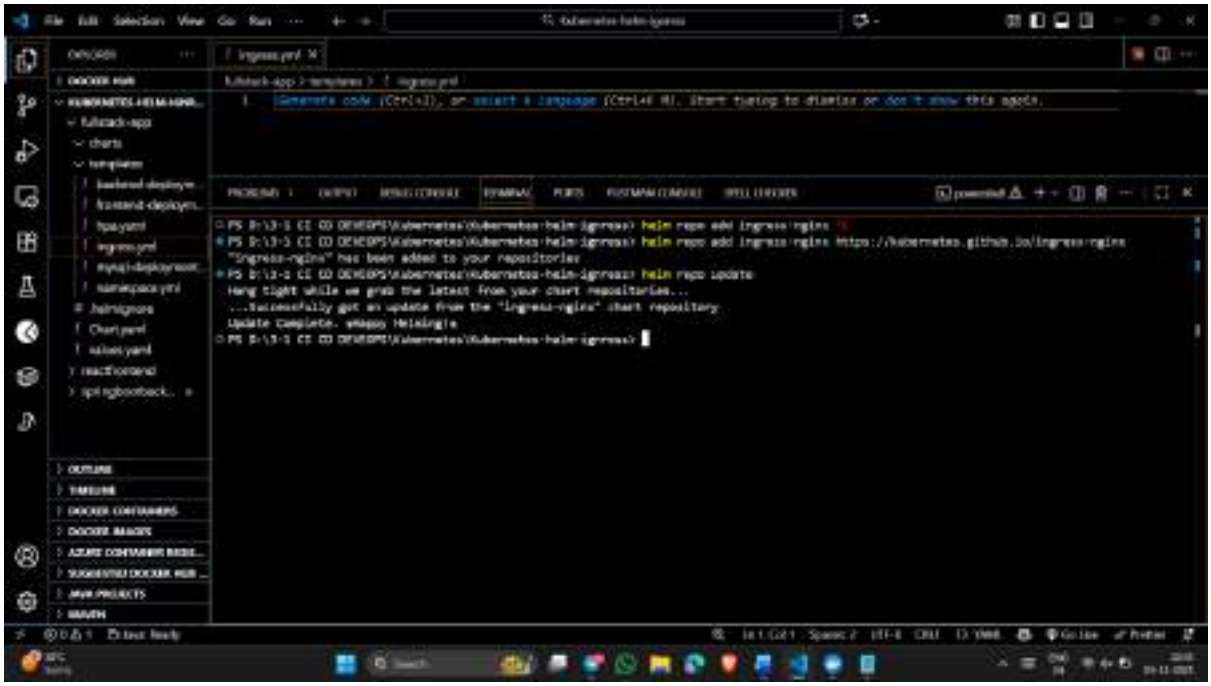
Task API Docs:



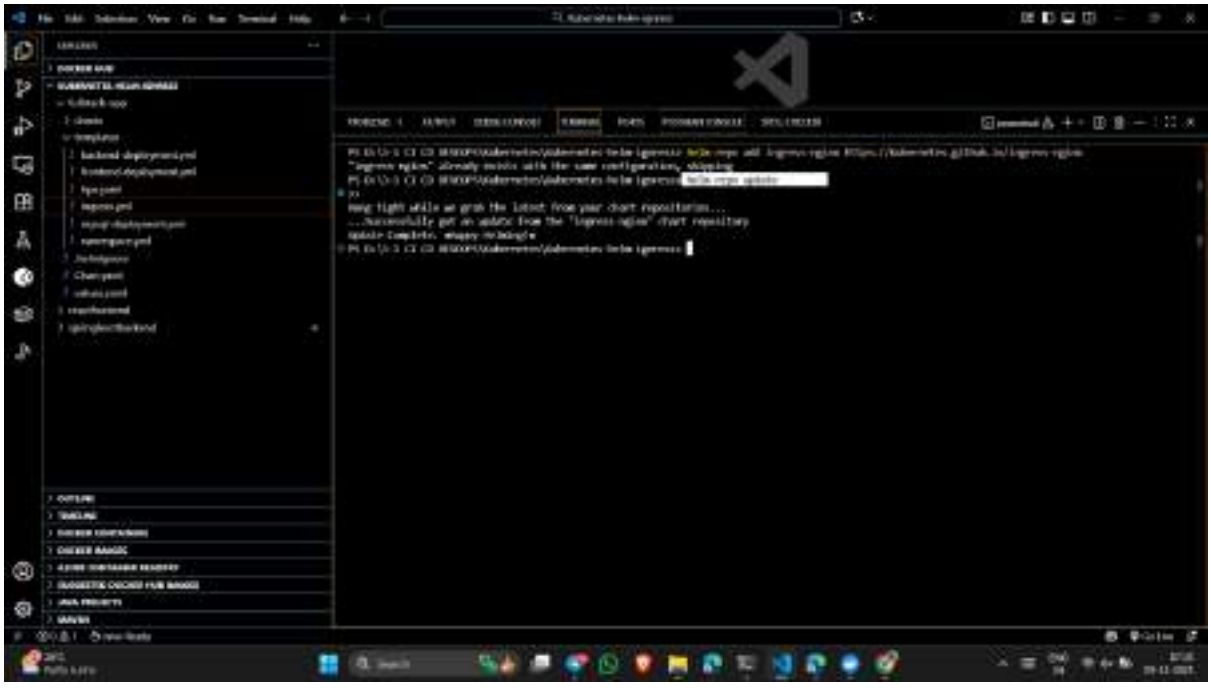

Task Management API
task-controller

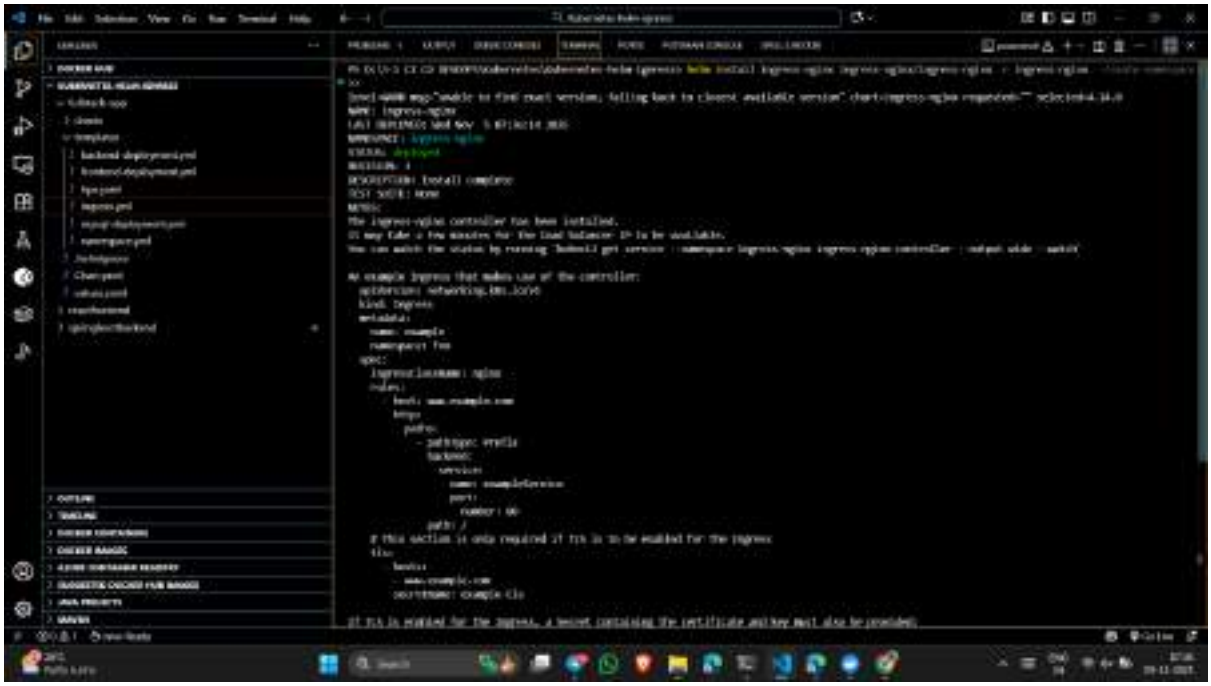


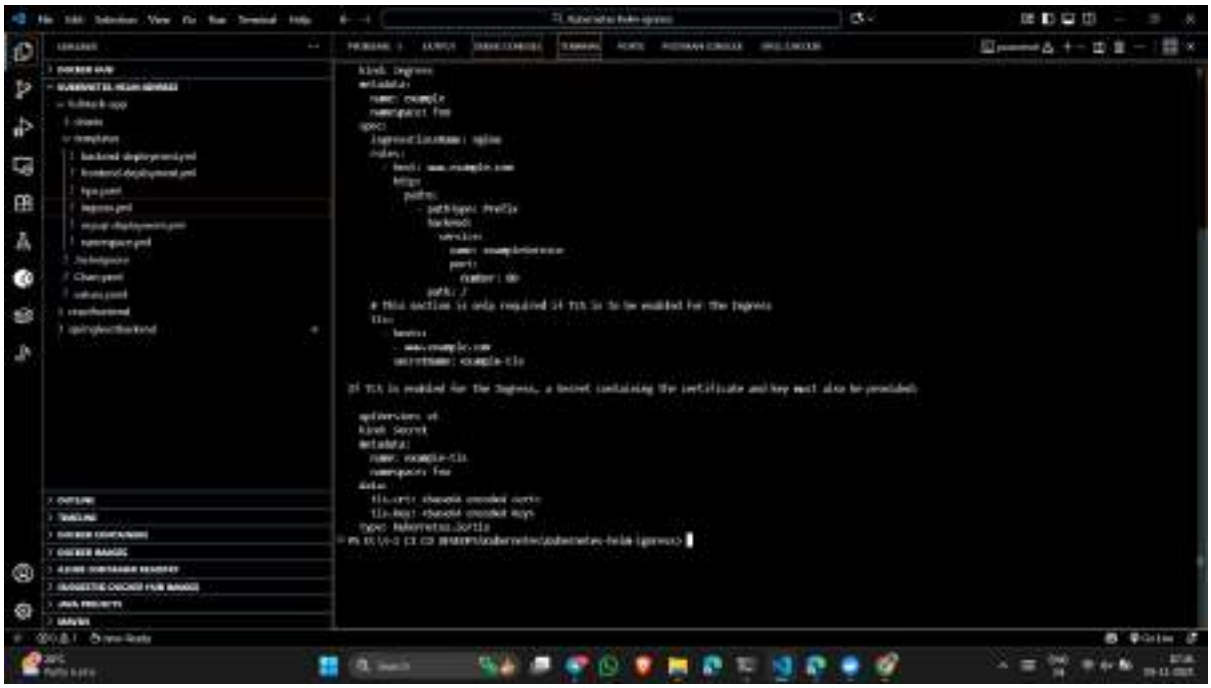

docker desktop
Containers
Your running containers show up here
What is a container?
How do I run a container?

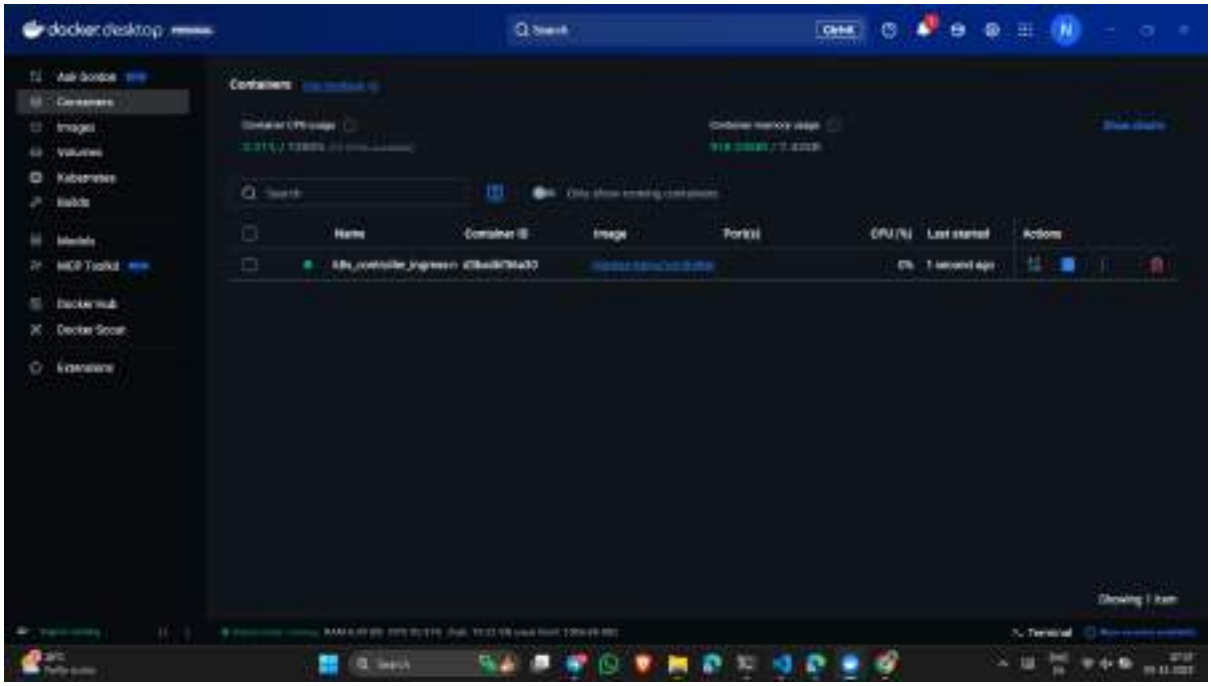


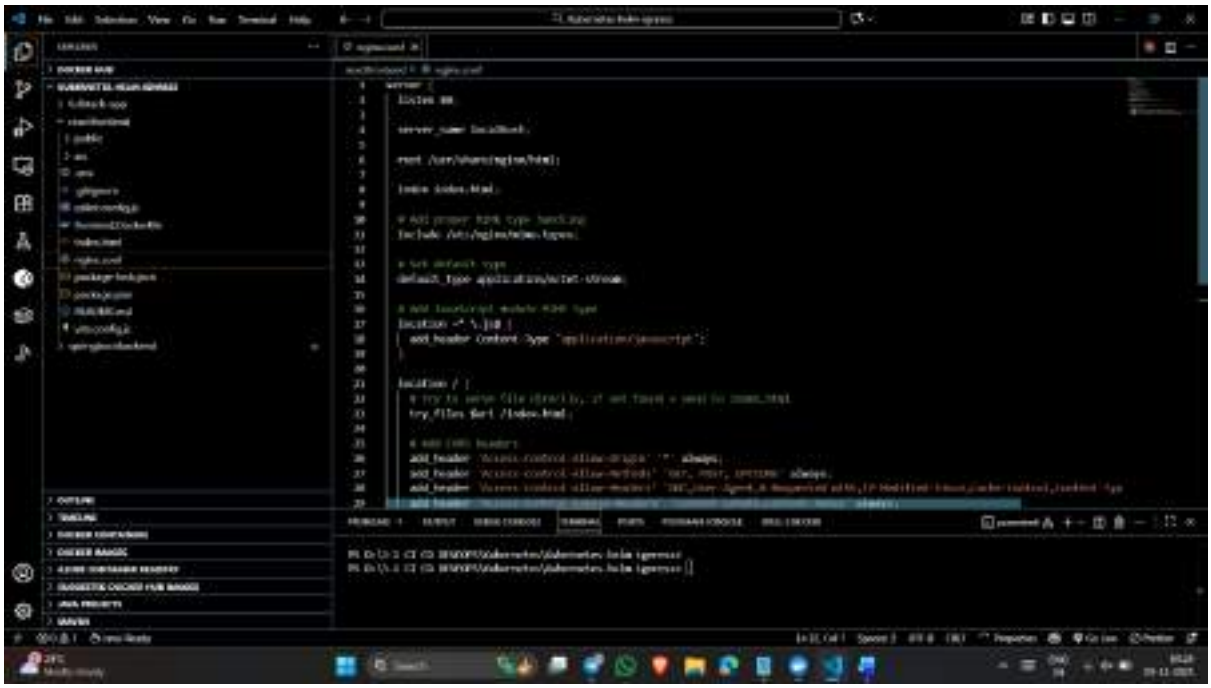
Helm with ingress

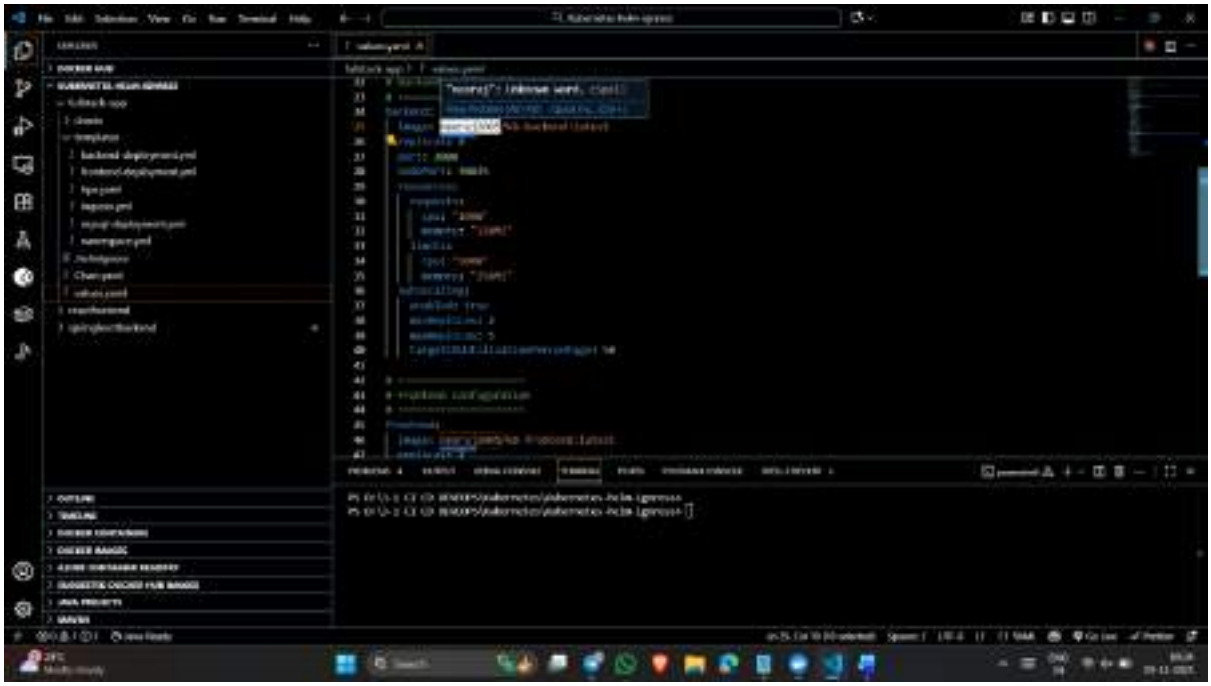


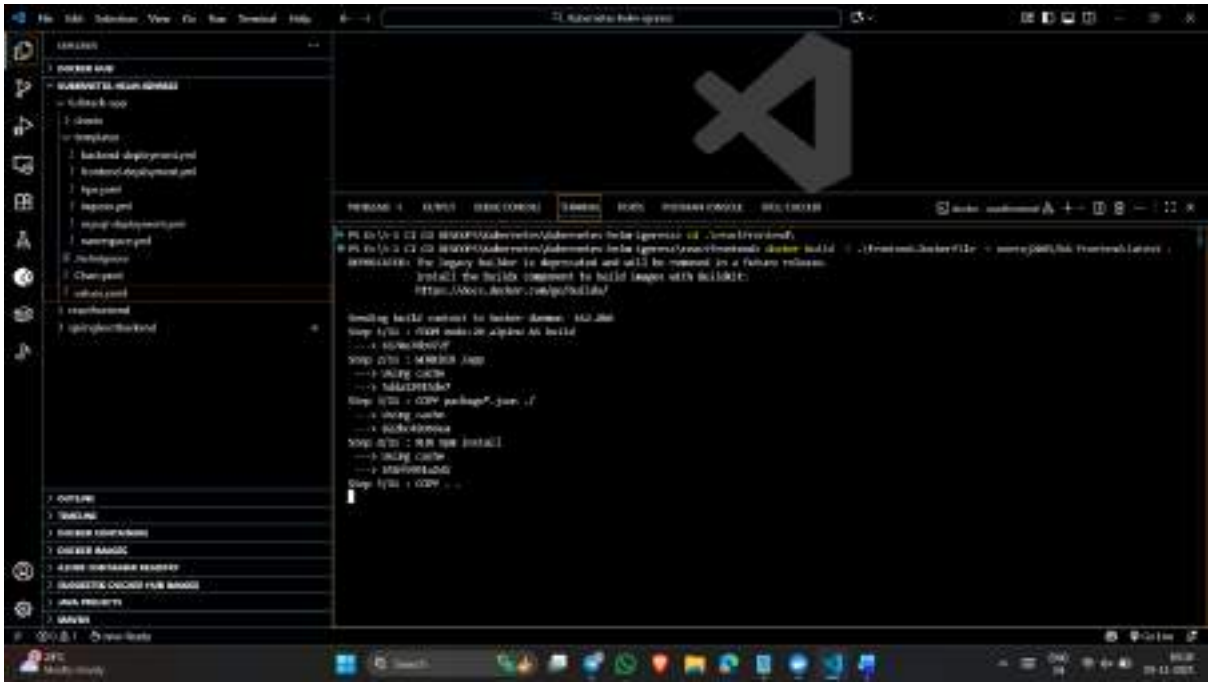


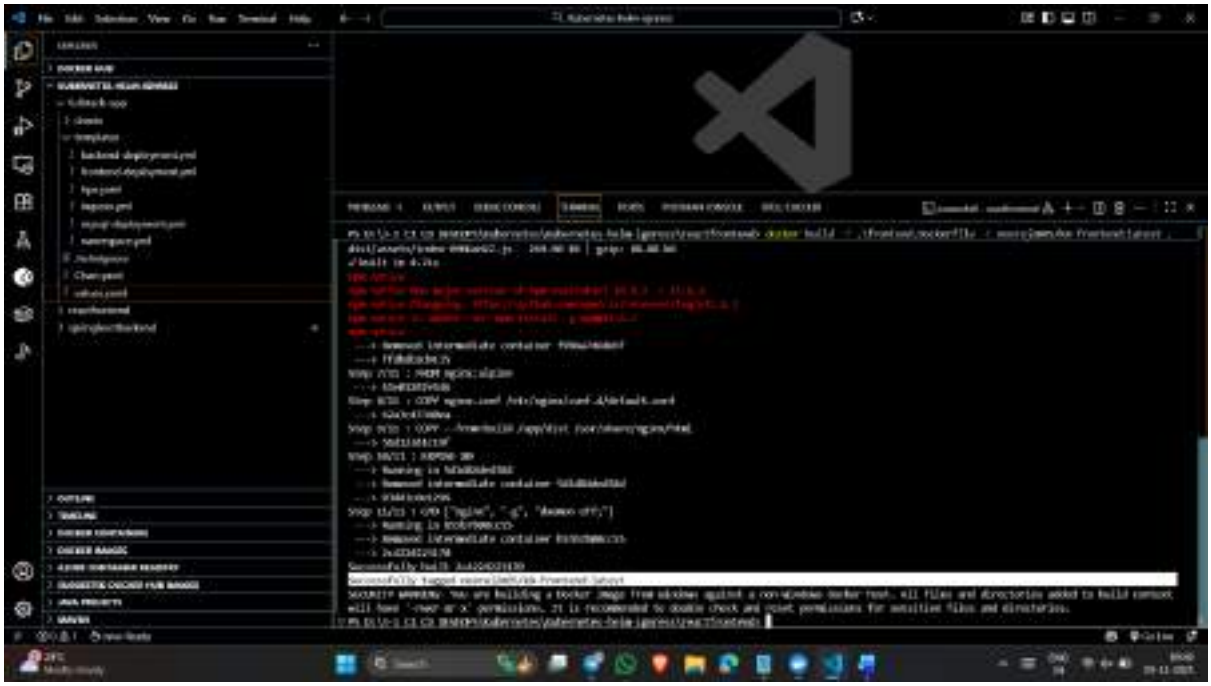


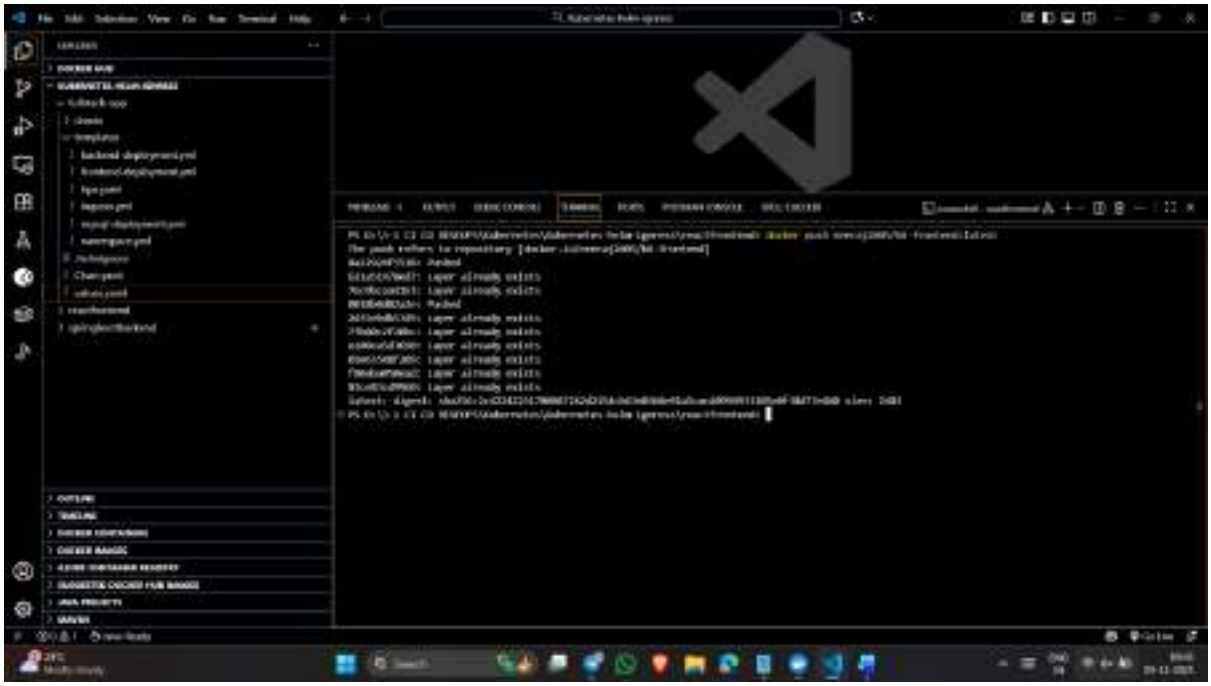

docker desktop
Containers
Name
Container ID
Image
Port(s)
CPU (%)
Last started
Actions
Showing 1 item

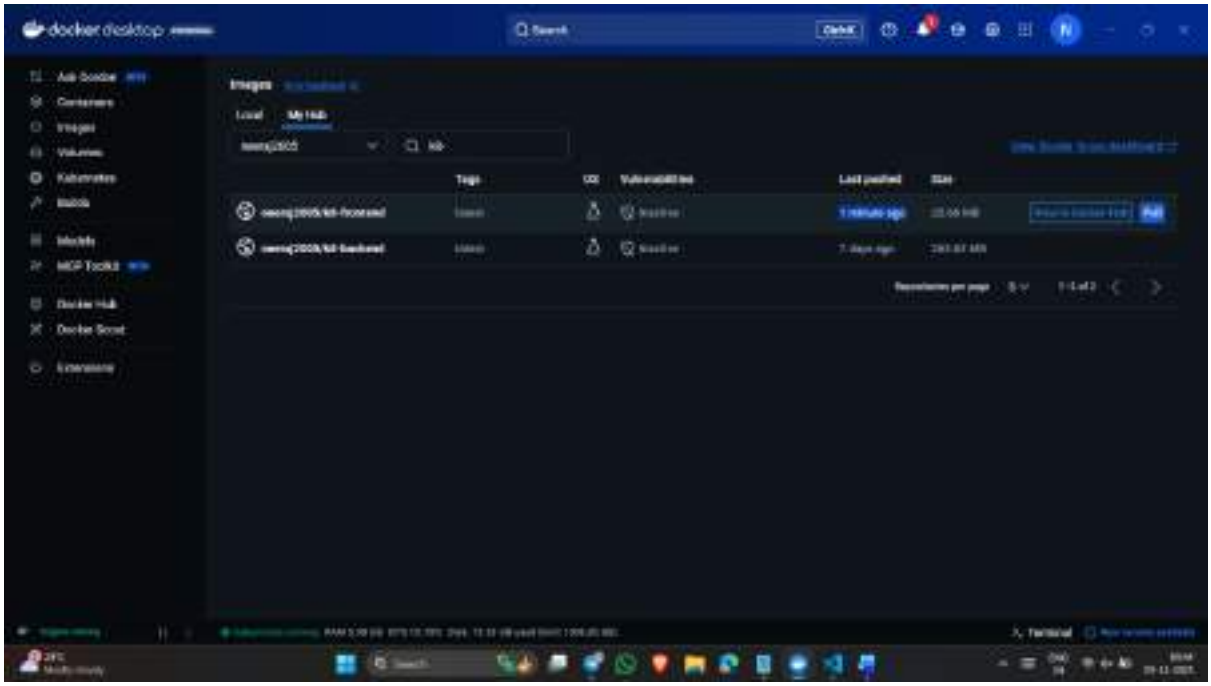


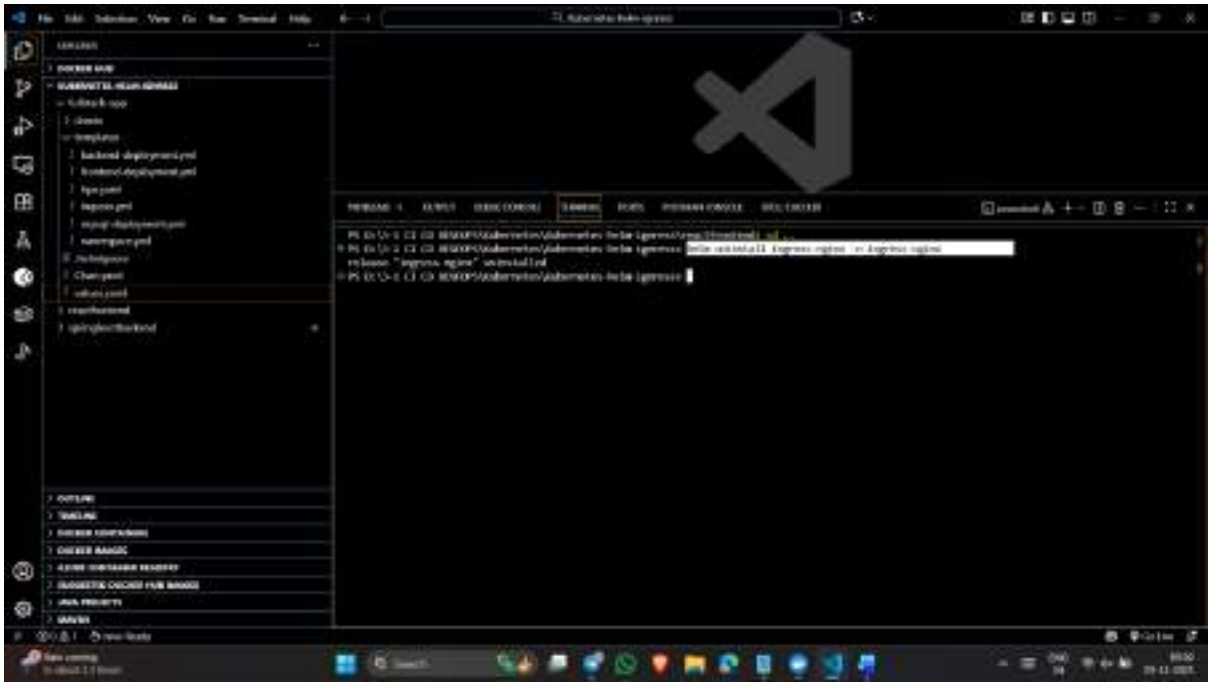


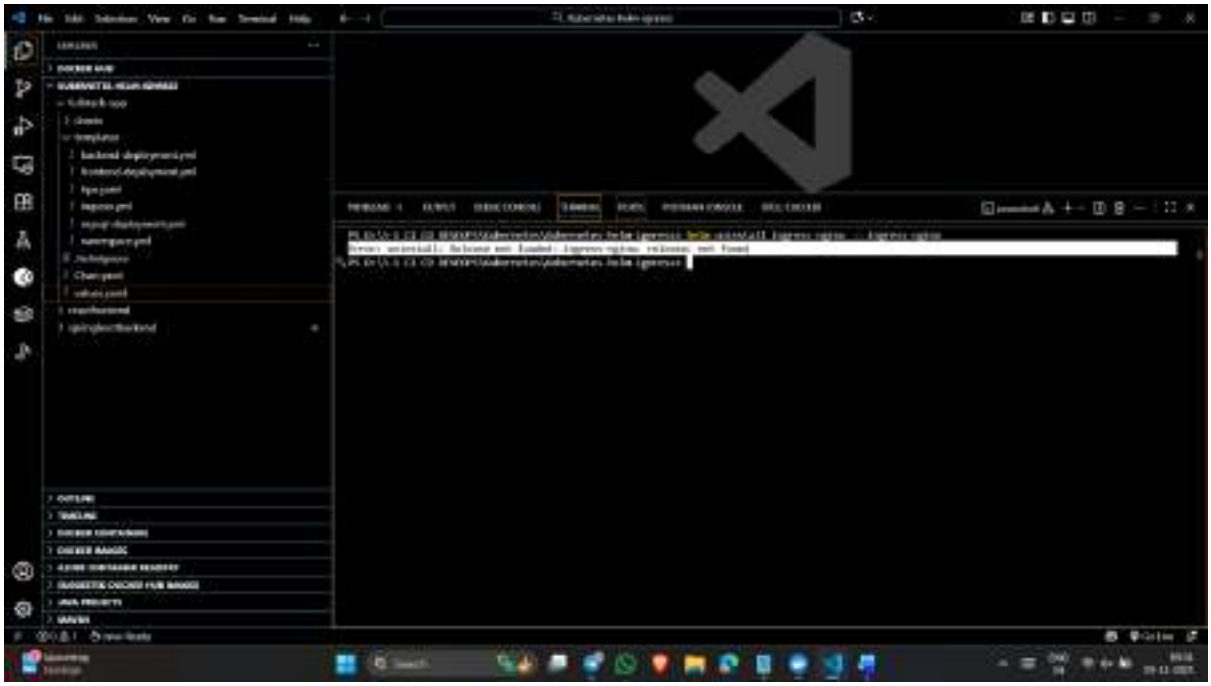


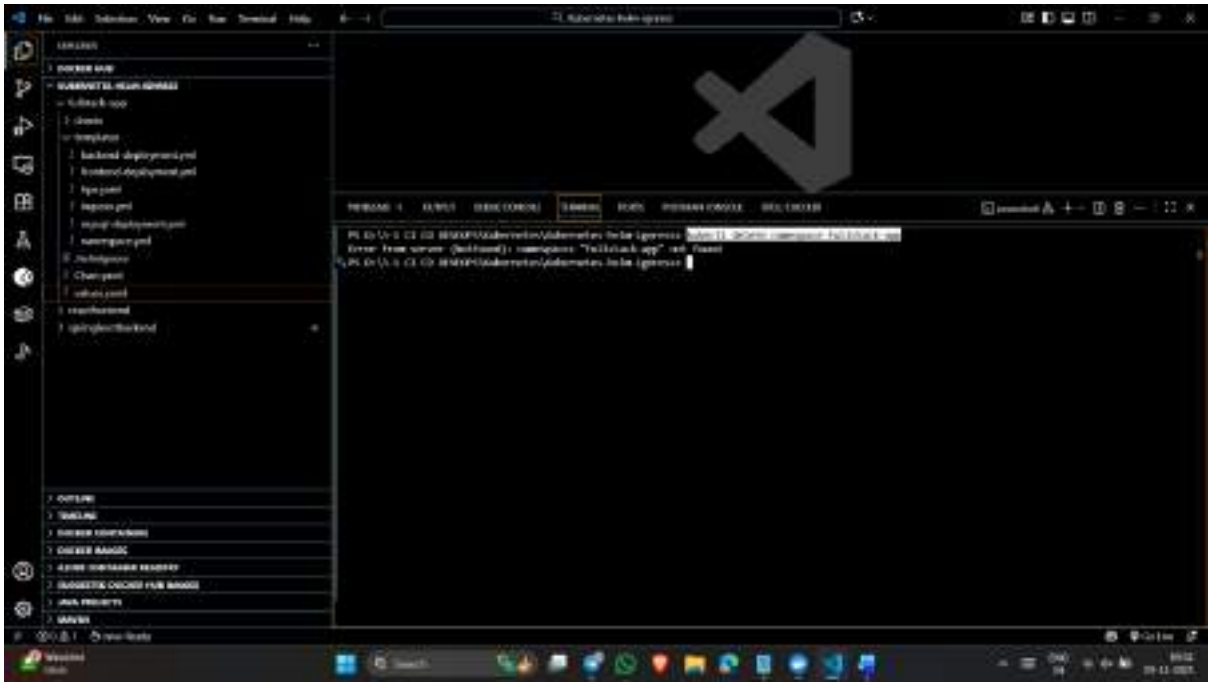


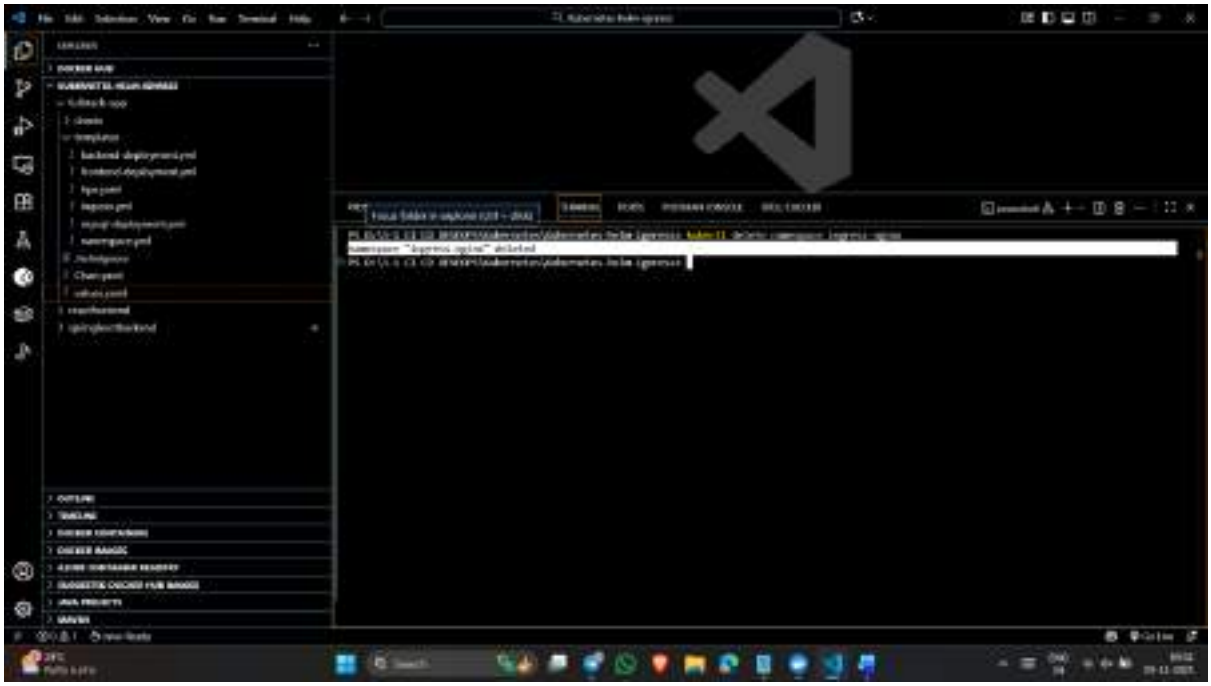


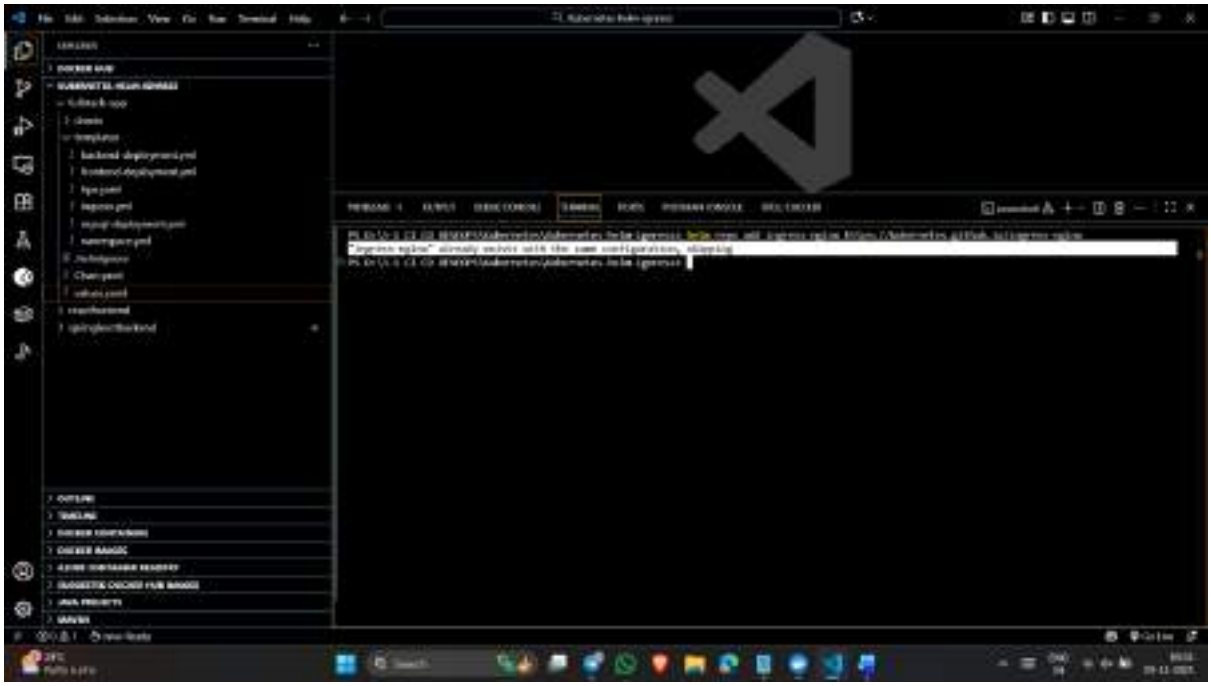

docker desktop
Images
Local
My Hub

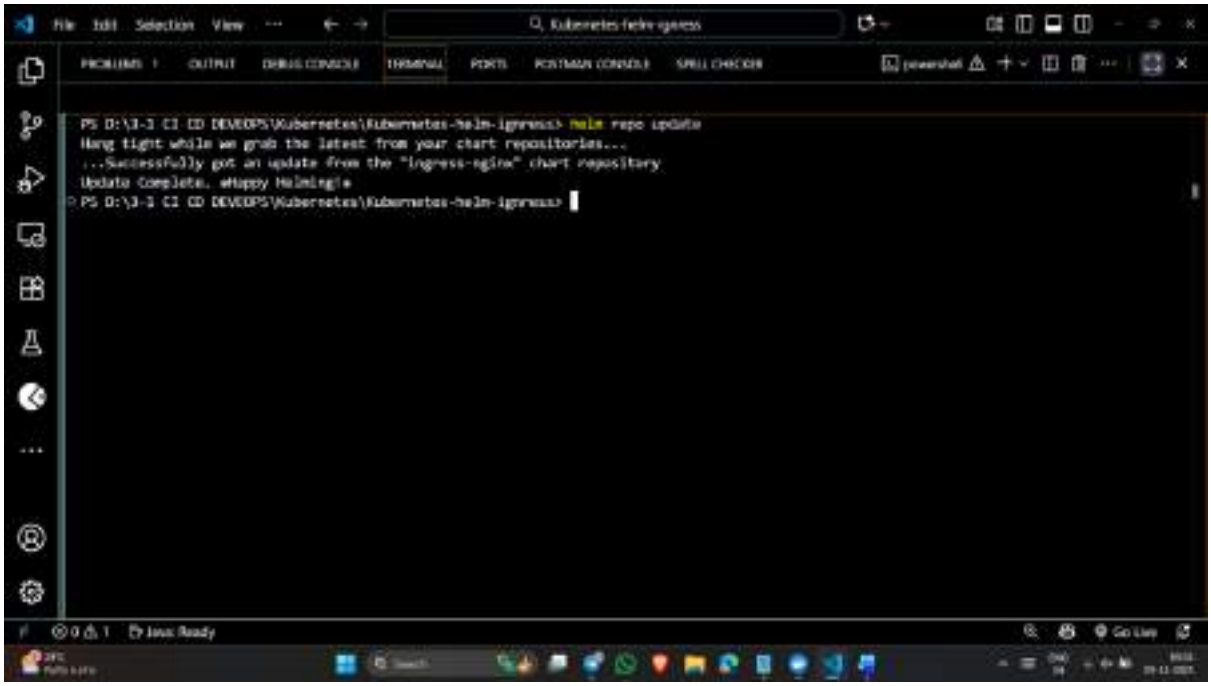


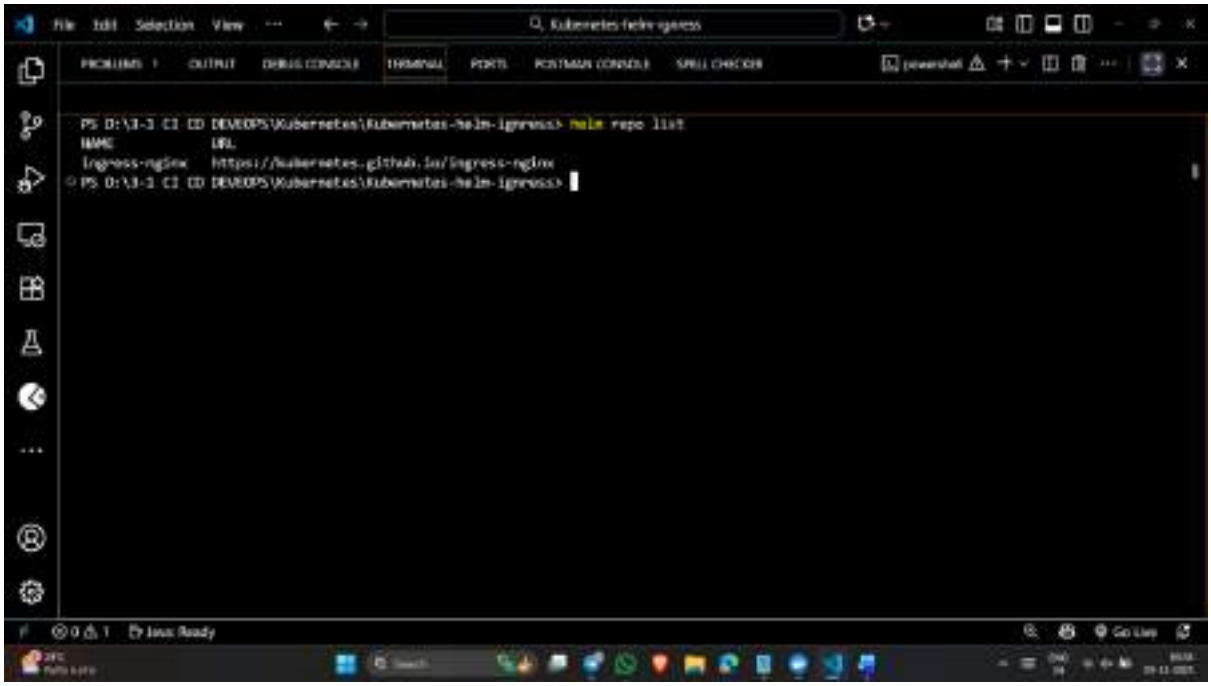


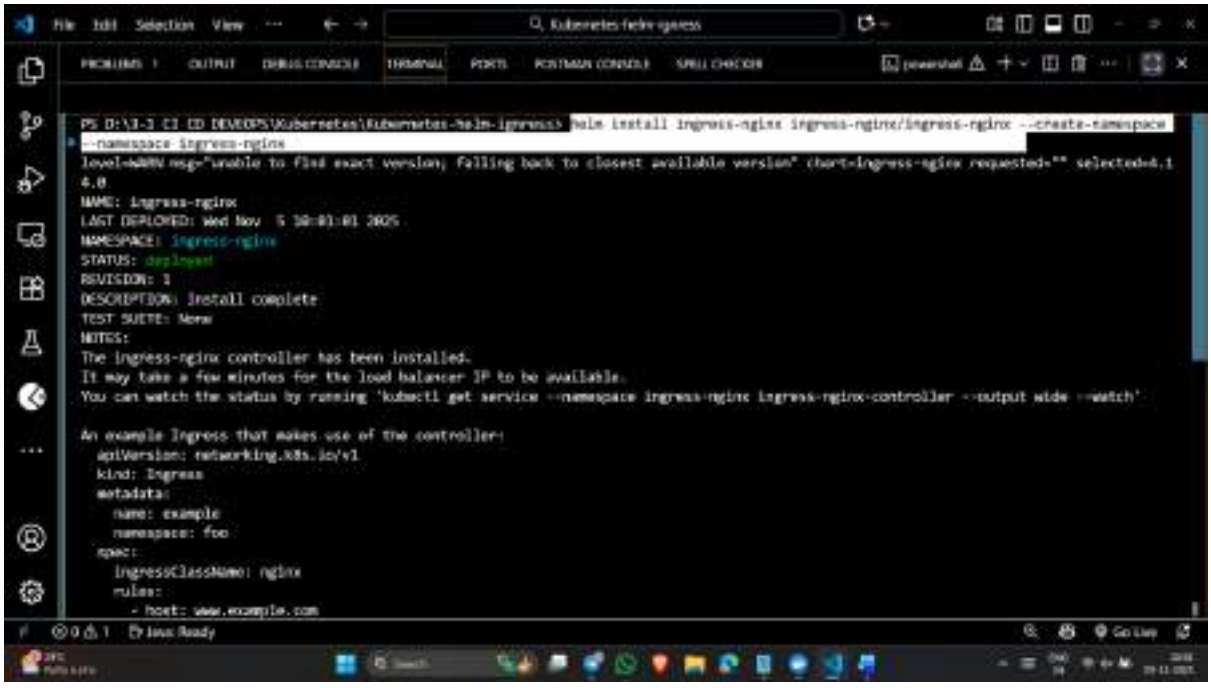


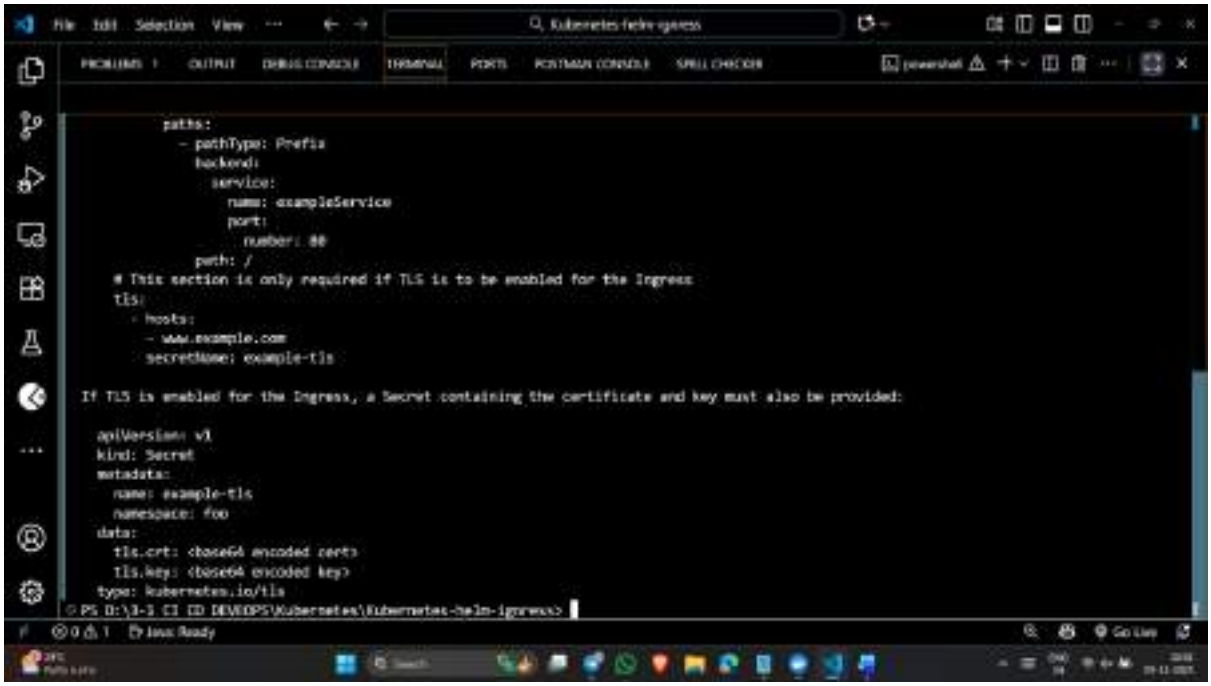


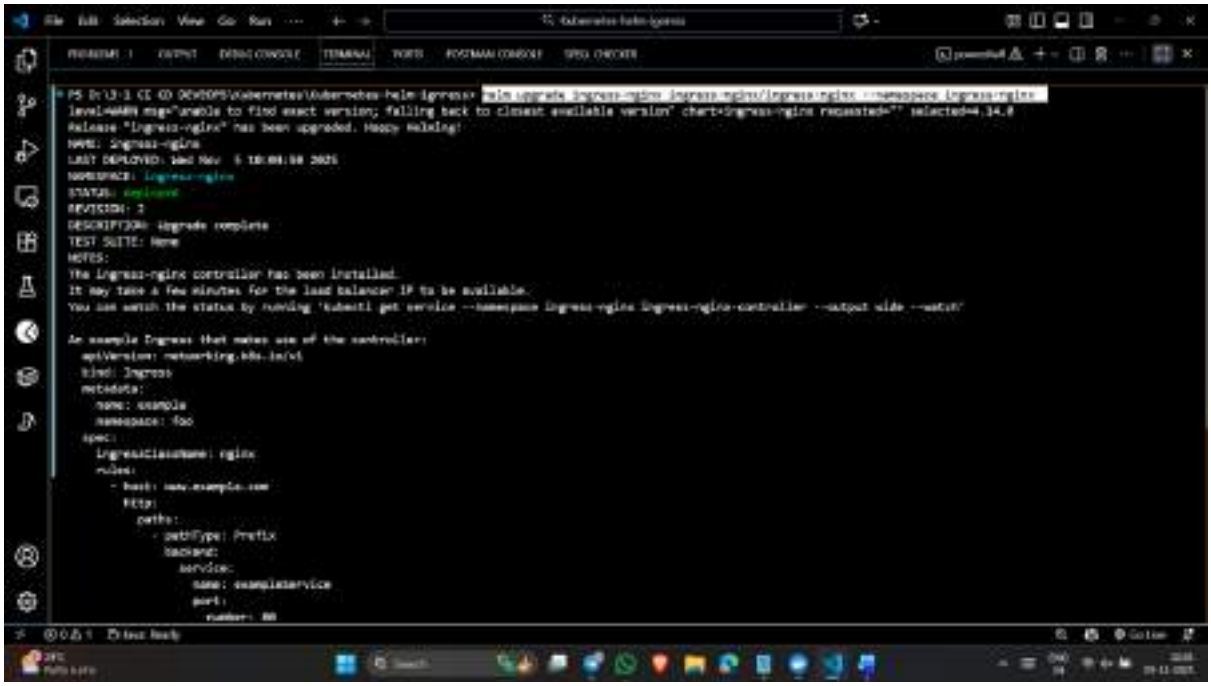


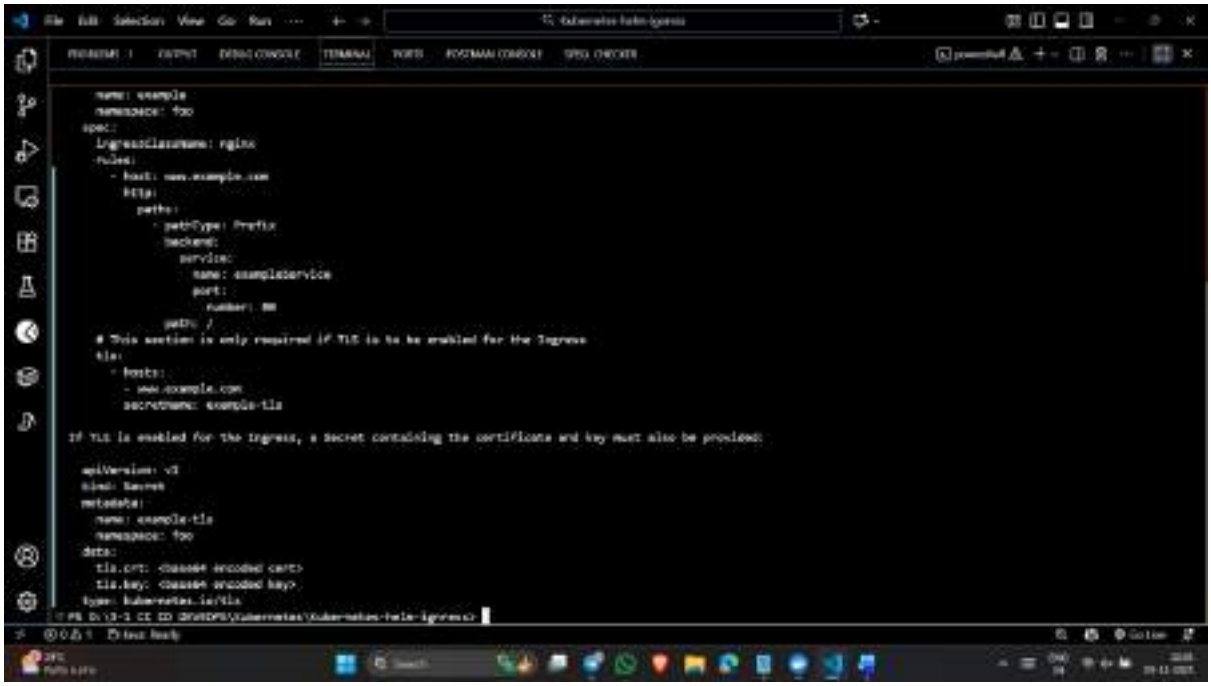

File Edit Selection View
Kubernetes-helm-ingress
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
POSTMAN CONSOLE
SPELL CHECKER
powershell
PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress> helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "ingress-nginx" chart repository
Update Complete. ⎈Happy Helming!⎈
PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress>
Java: Ready
Go Live

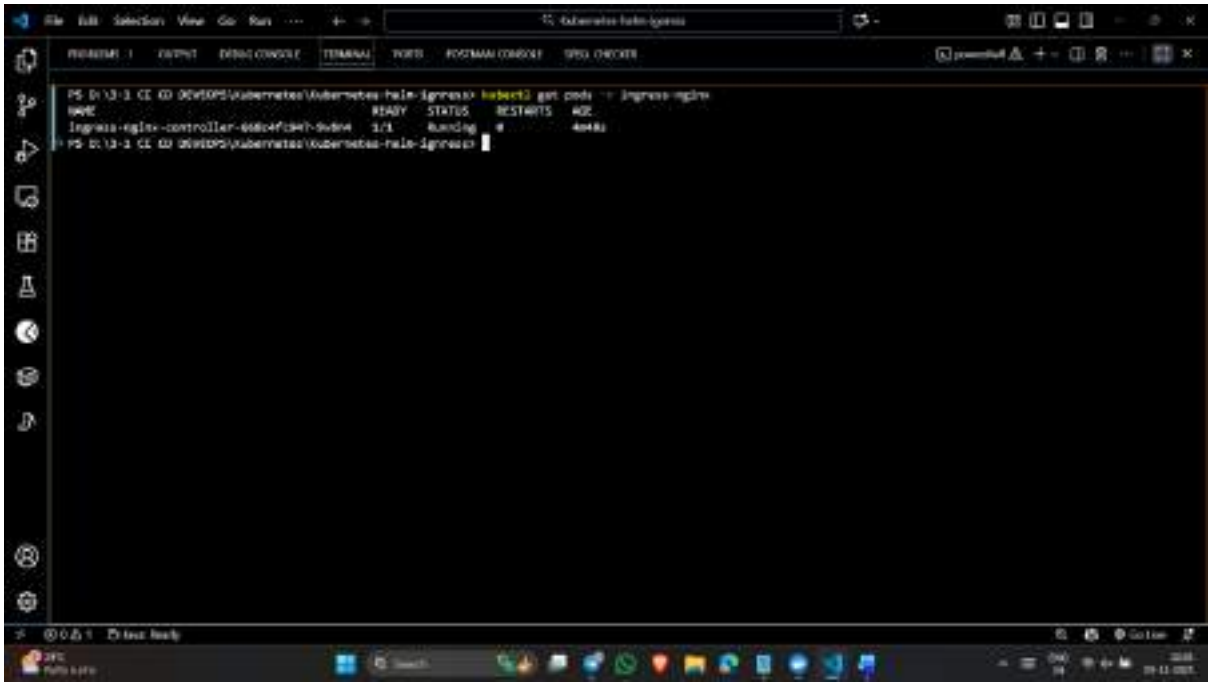

PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress> helm repo list
NAME URL
ingress-nginx https://kubernetes.github.io/ingress-nginx
PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress>

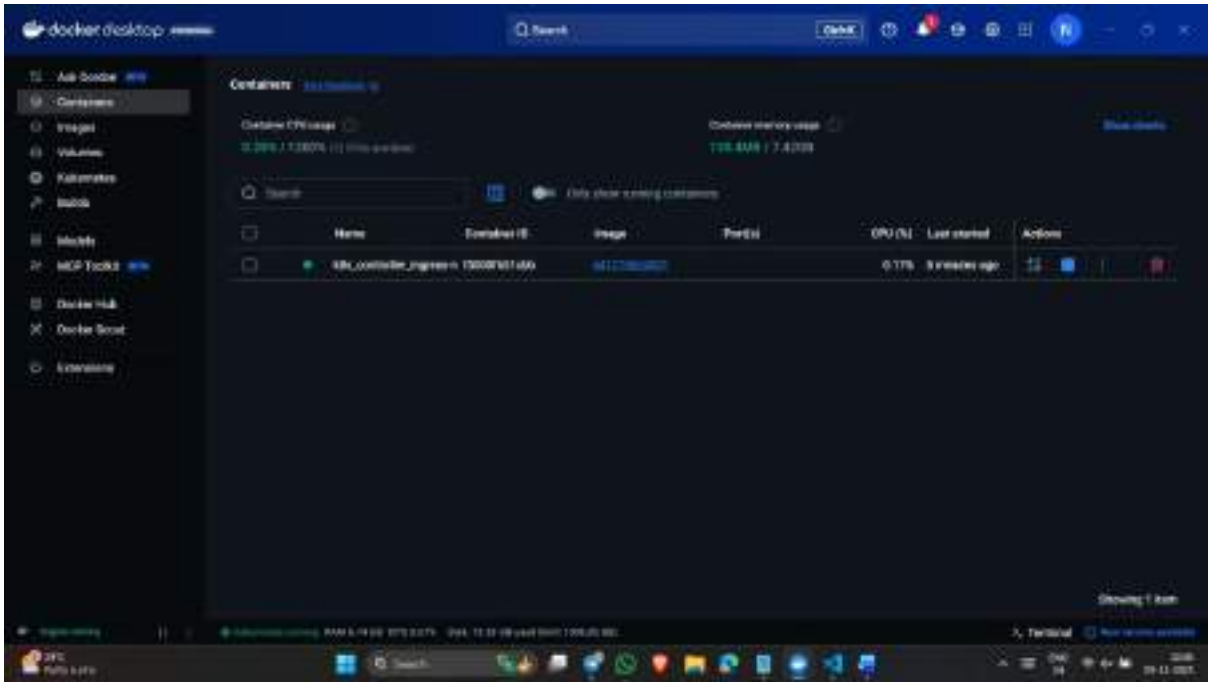

PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress> helm install ingress-nginx ingress-nginx/ingress-nginx --create-namespace
--namespace ingress-nginx
level=WARN msg="unable to find exact version; falling back to closest available version" chart=ingress-nginx requested="" selected=4.1
4.0
NAME: ingress-nginx
LAST DEPLOYED: Wed Nov 5 10:01:01 2025
NAMESPACE: ingress-nginx
STATUS: deployed
REVISION: 1
DESCRIPTION: Install complete
TEST SUITE: None
NOTES:
The ingress-nginx controller has been installed.
It may take a few minutes for the load balancer IP to be available.
You can watch the status by running 'kubectl get service --namespace ingress-nginx ingress-nginx-controller --output wide --watch'
An example Ingress that makes use of the controller:
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
name: example
namespace: foo
spec:
ingressClassName: nginx
rules:
- host: www.example.com

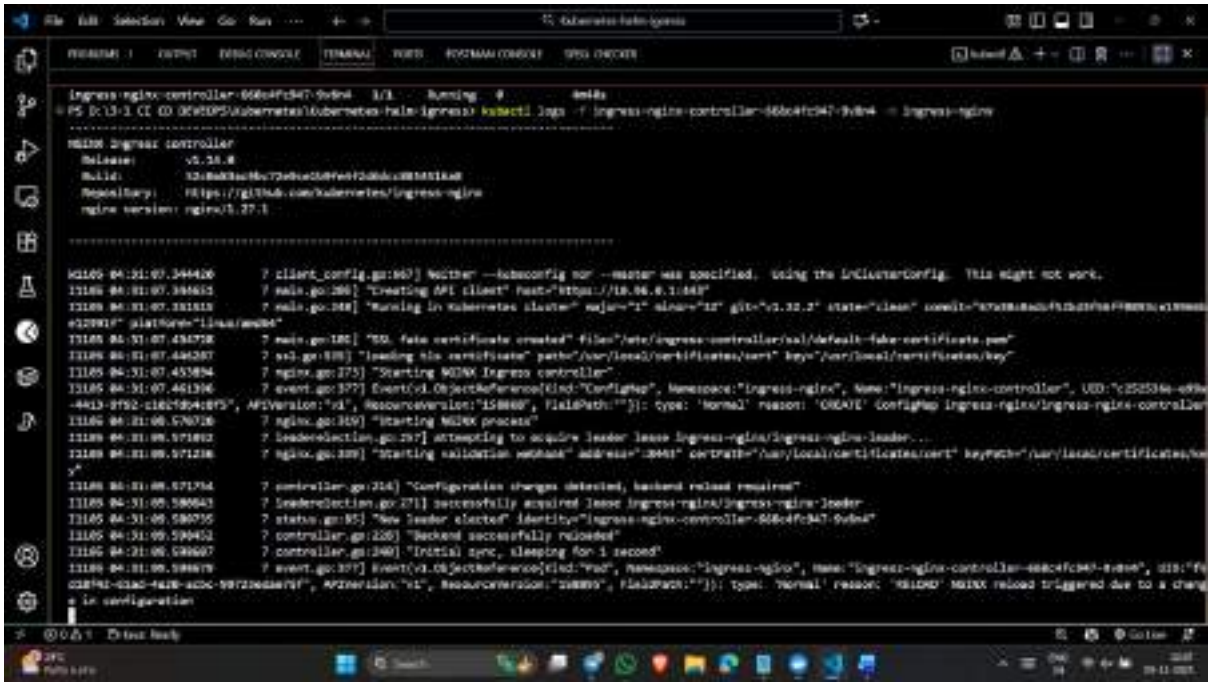

paths:
- pathType: Prefix
backend:
service:
name: exampleService
port:
number: 80
path: /
This section is only required if TLS is to be enabled for the Ingress
tls:
- hosts:
- www.example.com
secretName: example-tls
If TLS is enabled for the Ingress, a Secret containing the certificate and key must also be provided:
apiVersion: v1
kind: Secret
metadata:
name: example-tls
namespace: foo
data:
tls.crt: <base64 encoded cert>
tls.key: <base64 encoded key>
type: kubernetes.io/tls
PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-igress>

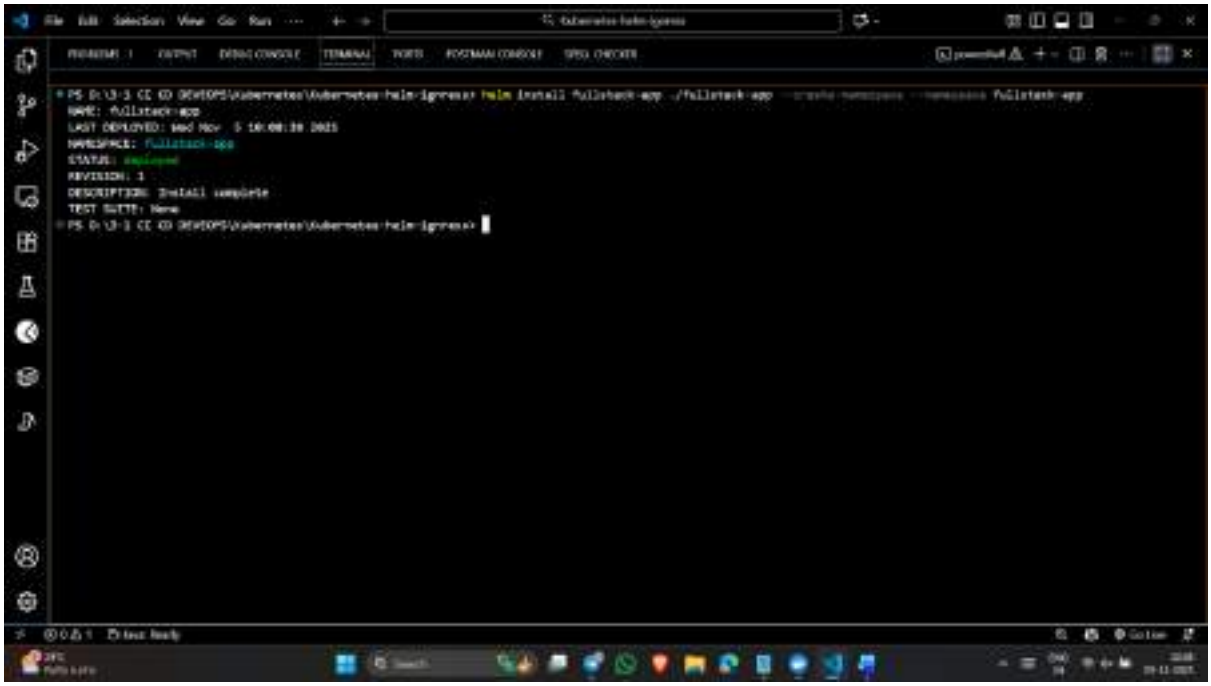

PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-igress> helm upgrade ingress-nginx ingress-nginx/ingress-nginx --namespace ingress-nginx
Release "ingress-nginx" has been upgraded. Happy Helming!
NAME: ingress-nginx
NAMESPACE: ingress-nginx
REVISION: 3
DESCRIPTION: Upgrade complete
TEST SUITE: None
NOTES:
The ingress-nginx controller has been installed.
It may take a few minutes for the load balancer IP to be available.
You can watch the status by running 'kubectl get service --namespace ingress-nginx ingress-nginx-controller --output wide --watch'
An example Ingress that makes use of the controller:
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
name: example
namespace: foo
spec:
ingressClassName: nginx
rules:
- host: www.example.com
http:
paths:
- pathType: Prefix
backend:
service:
name: exampleservice
port:

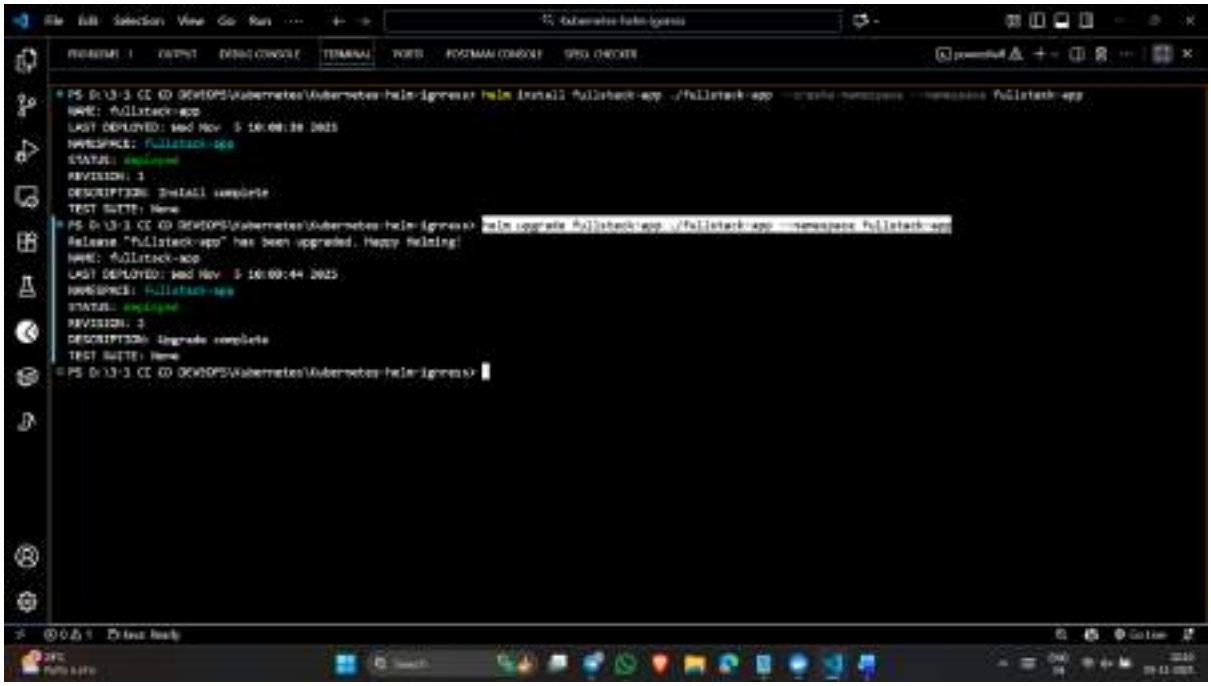

name: example
namespace: foo
spec:
ingressClassName: nginx
rules:
- host: www.example.com
http:
paths:
- pathType: Prefix
backend:
service:
name: exampleservice
port:
number: 80
path: /
This section is only required if TLS is to be enabled for the Ingress
tls:
- hosts:
- www.example.com
secretName: example-tls
If TLS is enabled for the Ingress, a Secret containing the certificate and key must also be provided:
apiVersion: v1
kind: Secret
metadata:
name: example-tls
namespace: foo
data:
tls.crt: <base64 encoded cert>
tls.key: <base64 encoded key>
type: kubernetes.io/tls

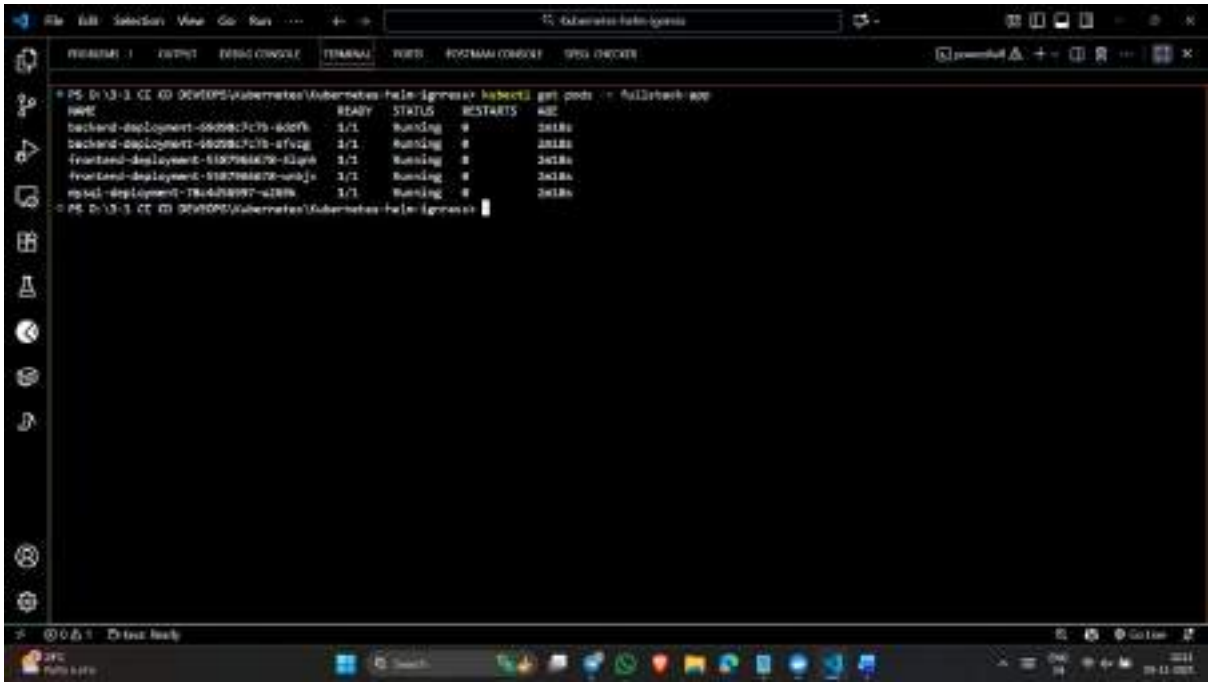

kubectl get pods -n ingress-nginx
NAME READY STATUS RESTARTS AGE
Ingress-nginx-controller-6b8c4f1347-9vbn4 1/1 Running 0 4m46s

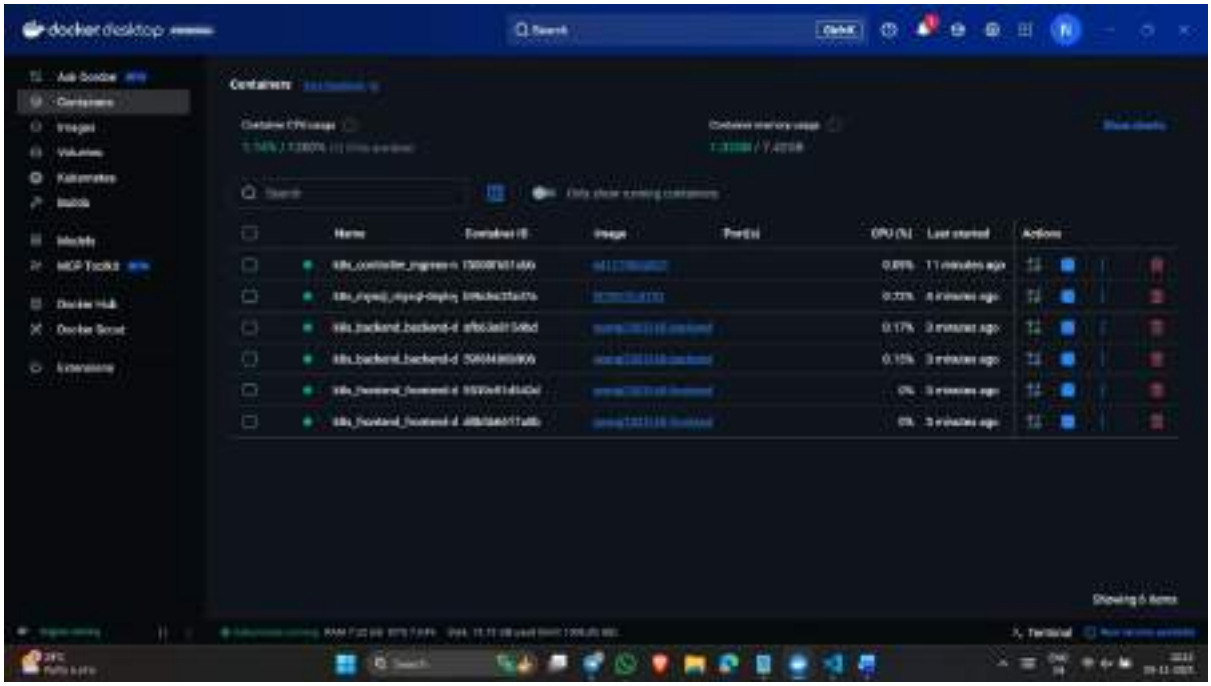

docker desktop
Containers
Name
Container ID
Image
Port(s)
CPU (%)
Last started
Actions

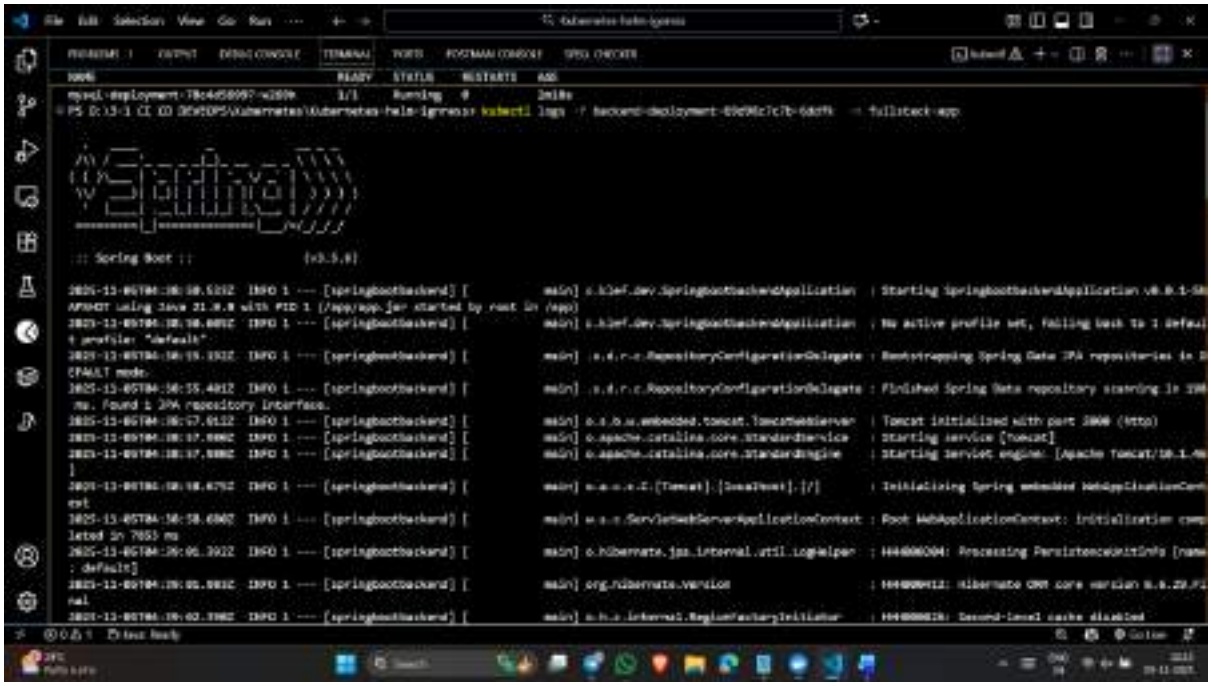

ingress-nginx-controller-668c4fc947-9v8n4 1/1 Running 0 4m48s
PS D:\3-1 CE ED DEVOPS\Kubernetes\kubernetes-helm-ingress> kubectl logs -f ingress-nginx-controller-668c4fc947-9v8n4 -n ingress-nginx
NGINX Ingress controller
Release: v1.14.0
Build:
Repository: https://github.com/kubernetes/ingress-nginx
nginx version: nginx/1.27.1
W1105 04:31:07.344420 7 client_config.go:667] Neither --kubeconfig nor --master was specified. Using the InClusterConfig. This might not work.
I1105 04:31:07.344611 7 main.go:205] "Creating API client" host="https://10.96.0.1:443"
I1105 04:31:07.381613 7 main.go:248] "Running in Kubernetes cluster" major="1" minor="32" git="v1.32.2" state="clean" commit="
e120814" platform="linux/amd64"
I1105 04:31:07.454738 7 main.go:101] "SSL fake certificate created" file="/etc/ingress-controller/ssl/default-fake-certificate.pem"
I1105 04:31:07.466207 7 ssl.go:535] "loading tls certificate" path="/usr/local/certificates/cert" key="/usr/local/certificates/key"
I1105 04:31:07.457854 7 nginx.go:273] "Starting NGINX Ingress controller"
I1105 04:31:07.461316 7 event.go:377] Event(v1.ObjectReference{Kind:"ConfigMap", Namespace:"ingress-nginx", Name:"ingress-nginx-controller", UID:"c2525346-e08e
-4413-9752-c1617db4c075", APIVersion:"v1", ResourceVersion:"158868", FieldPath:""}): type: 'Normal' reason: 'CREATE' ConfigMap ingress-nginx/ingress-nginx-controller
I1105 04:31:08.570728 7 nginx.go:319] "Starting NGINX process"
I1105 04:31:08.571052 7 leaderelection.go:257] attempting to acquire leader lease ingress-nginx/ingress-nginx-leader...
I1105 04:31:08.571238 7 nginx.go:339] "Starting validation webhook" address=":8443" certPath="/usr/local/certificates/cert" keyPath="/usr/local/certificates/ke
y"
I1105 04:31:08.571754 7 controller.go:214] "Configuration changes detected, backend reload required"
I1105 04:31:08.586643 7 leaderelection.go:271] successfully acquired lease ingress-nginx/ingress-nginx-leader
I1105 04:31:08.586756 7 status.go:85] "New leader elected" identity="ingress-nginx-controller-668c4fc947-9v8n4"
I1105 04:31:08.590451 7 controller.go:230] "Backend successfully reloaded"
I1105 04:31:08.590607 7 controller.go:240] "Initial sync, sleeping for 1 second"
I1105 04:31:08.590679 7 event.go:377] Event(v1.ObjectReference{Kind:"Pod", Namespace:"ingress-nginx", Name:"ingress-nginx-controller-668c4fc947-9v8n4", UID:"f
c1d7e2-61ac-4a38-ac6c-9972ce6ae7d7", APIVersion:"v1", ResourceVersion:"158859", FieldPath:""}): type: 'Normal' reason: 'RELOAD' NGINX reload triggered due to a chang
e in configuration

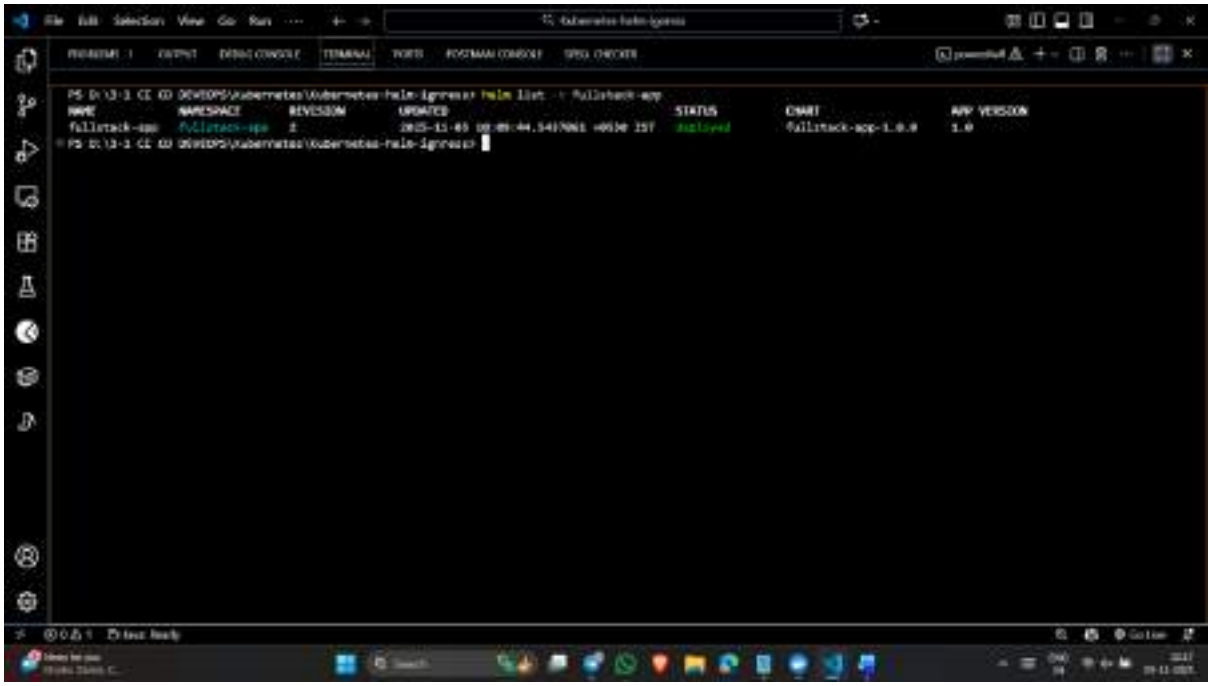

PS D:\3-1 CI CD DEVOPS\kubernetes\kubernetes-helm-ingress> helm install fullstack-app ./fullstack-app --create-namespace --namespace fullstack-app
NAME: fullstack-app
LAST DEPLOYED: Wed Nov 5 10:08:39 2025
NAMESPACE: fullstack-app
STATUS: deployed
REVISION: 1
DESCRIPTION: Install complete
TEST SUITE: None
PS D:\3-1 CI CD DEVOPS\kubernetes\kubernetes-helm-ingress>

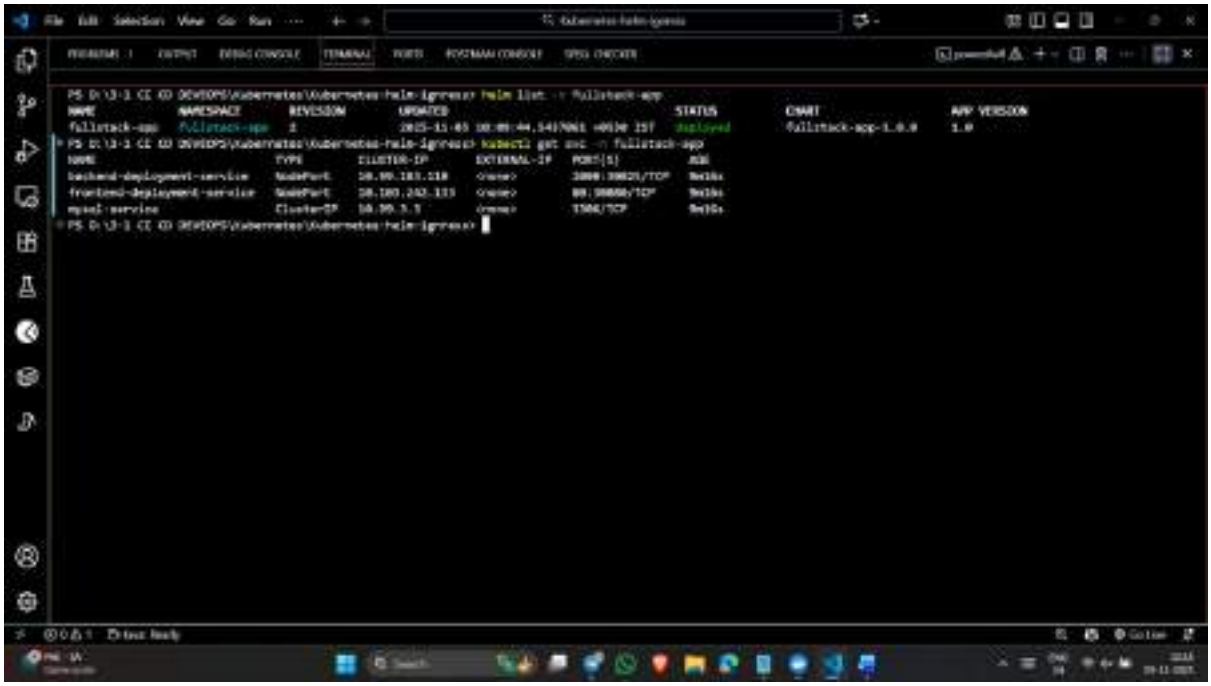

PS D:\3-1 CI CD DEVOPS\kubernetes\kubernetes-helm-ingress> helm install fullstack-app ./fullstack-app --create-namespace --namespace fullstack-app
NAME: fullstack-app
LAST DEPLOYED: Wed Nov 5 10:08:39 2025
NAMESPACE: fullstack-app
STATUS: deployed
REVISION: 1
DESCRIPTION: Install complete
TEST SUITE: None
PS D:\3-1 CI CD DEVOPS\kubernetes\kubernetes-helm-ingress> helm upgrade fullstack-app ./fullstack-app --namespace fullstack-app
Release "fullstack-app" has been upgraded. Happy Helming!
NAME: fullstack-app
LAST DEPLOYED: Wed Nov 5 10:09:44 2025
NAMESPACE: fullstack-app
STATUS: deployed
REVISION: 2
DESCRIPTION: Upgrade complete
TEST SUITE: None
PS D:\3-1 CI CD DEVOPS\kubernetes\kubernetes-helm-ingress>

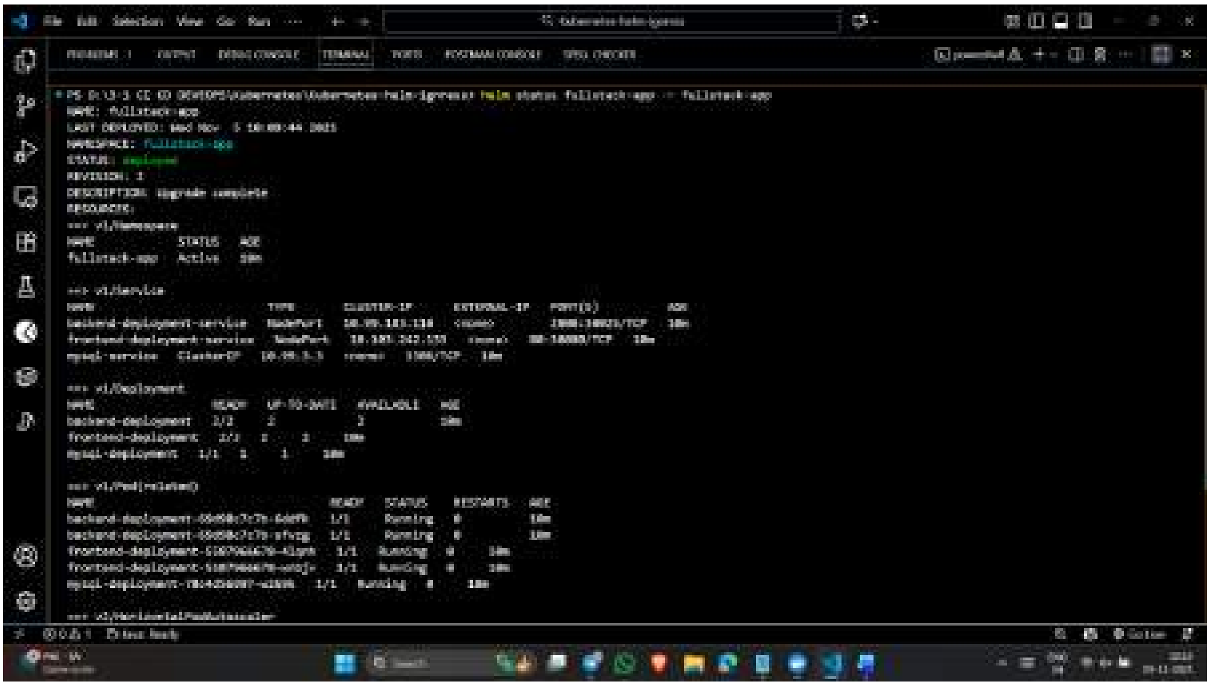


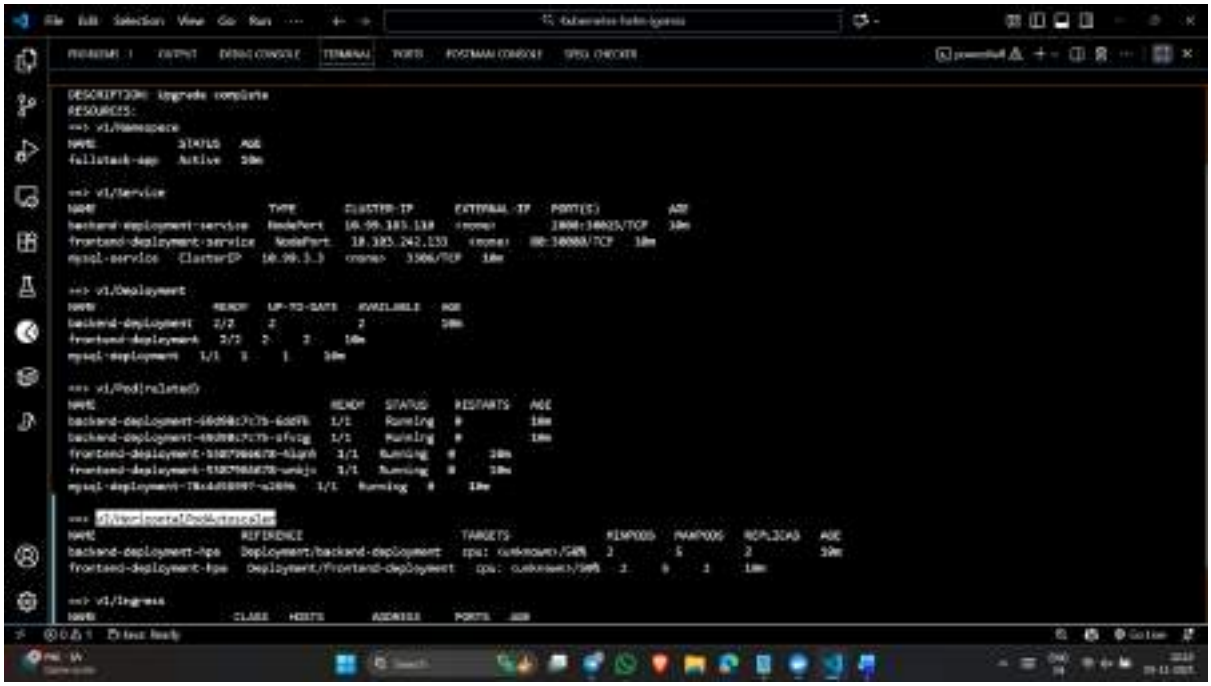


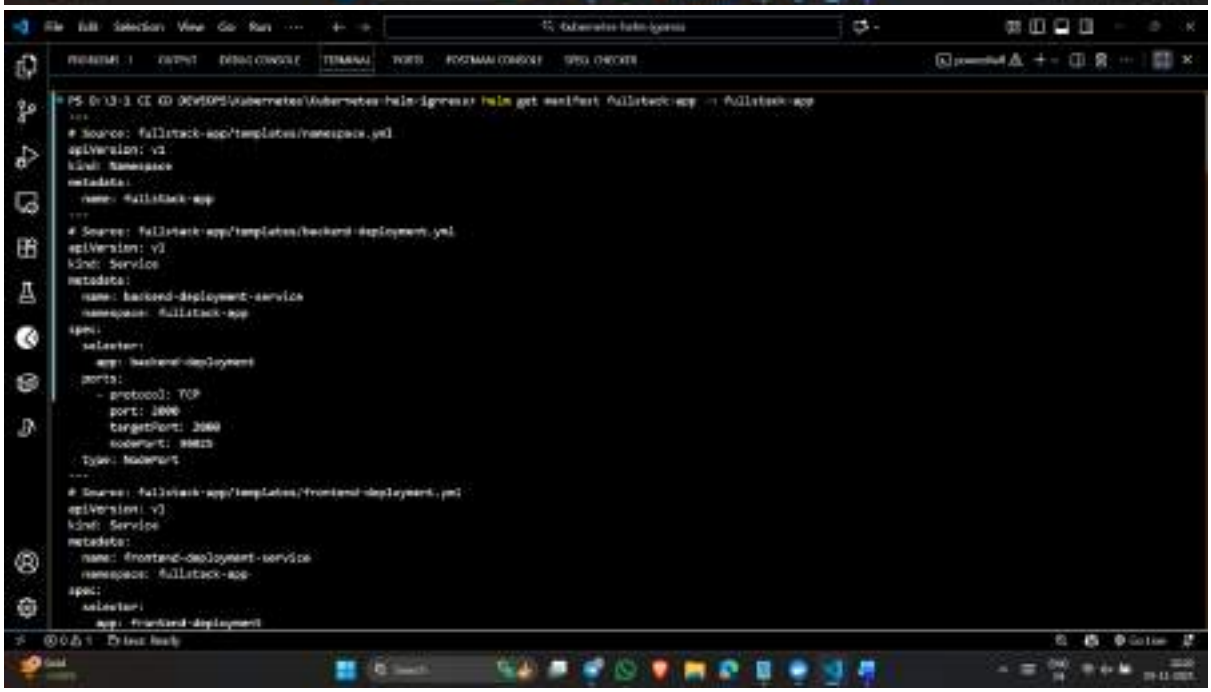

Project Backend is Running Successfully .....


PS D:\3-1 CI CD DEVOPS\kubernetes\kubernetes-helm-ingress> helm list -n fullstack-app
NAME NAMESPACE REVISION UPDATED STATUS CHART APP VERSION
fullstack-app fullstack-app 2 2025-11-05 10:09:44.5497061 +0530 IST deployed fullstack-app-1.0.0 1.0
PS D:\3-1 CI CD DEVOPS\kubernetes\kubernetes-helm-ingress>


PS D:\3-1 CI CD DEVOPS\kubernetes\kubernetes-helm-ingress> helm list -n fullstack-app
NAME NAMESPACE REVISION UPDATED STATUS CHART APP VERSION
fullstack-app fullstack-app 2 2025-11-05 10:09:44.5497061 +0530 IST deployed fullstack-app-1.0.0 1.0
PS D:\3-1 CI CD DEVOPS\kubernetes\kubernetes-helm-ingress> kubectl get svc -n fullstack-app
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
backend-deployment-service NodePort 10.99.183.118 <none> 3000:30025/TCP 9m16s
frontend-deployment-service NodePort 10.103.242.133 <none> 80:30000/TCP 9m16s
mysql-service ClusterIP 10.99.3.1 <none> 3306/TCP 9m16s
PS D:\3-1 CI CD DEVOPS\kubernetes\kubernetes-helm-ingress>

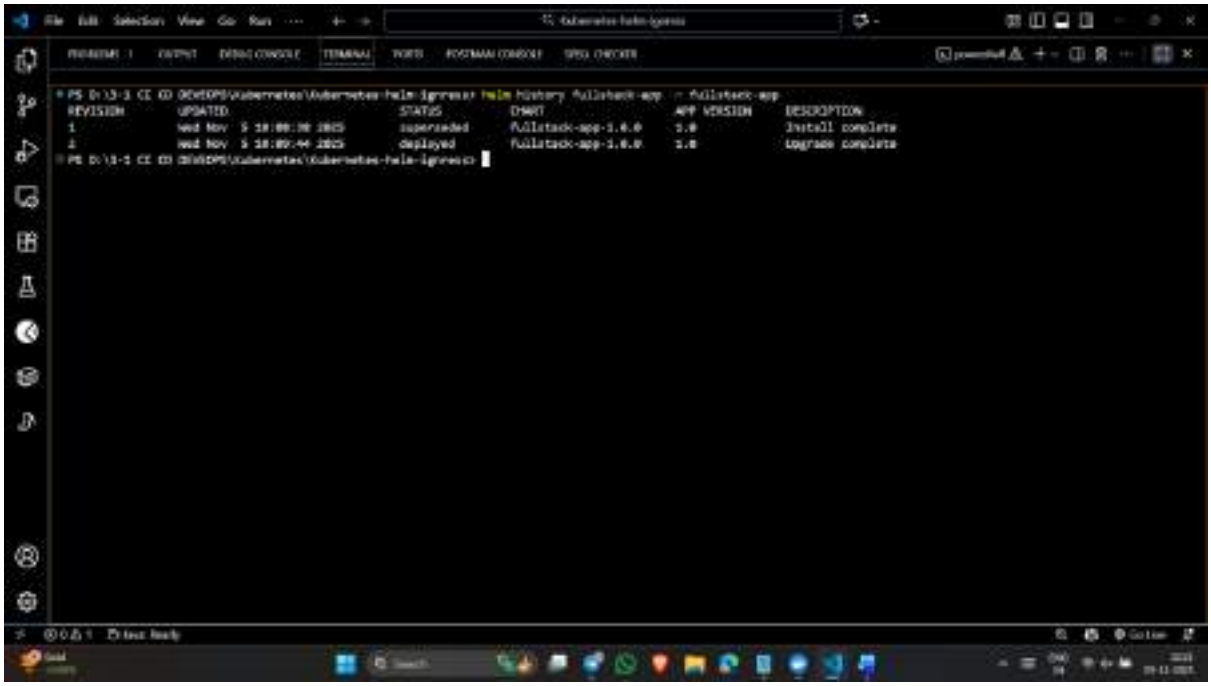

File Edit Selection View Go Run
OUTPUT
TERMINAL
NAME: fullstack-app
NAMESPACE: fullstack-app
REVISION: 1
DESCRIPTION: Upgrade complete
RESOURCES:
NAME STATUS AGE
fullstack-app Active 10m
==> v1/Service
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
backend-deployment-service NodePort
frontend-deployment-service NodePort
mysql-service ClusterIP
==> v1/Deployment
NAME READY UP-TO-DATE AVAILABLE AGE
backend-deployment 2/2 2 2 10m
frontend-deployment 2/2 2 2 10m
mysql-deployment 1/1 1 1 10m
==> v1/Pod(related)
NAME READY STATUS RESTARTS AGE
==> v1/HorizontalPodAutoscaler

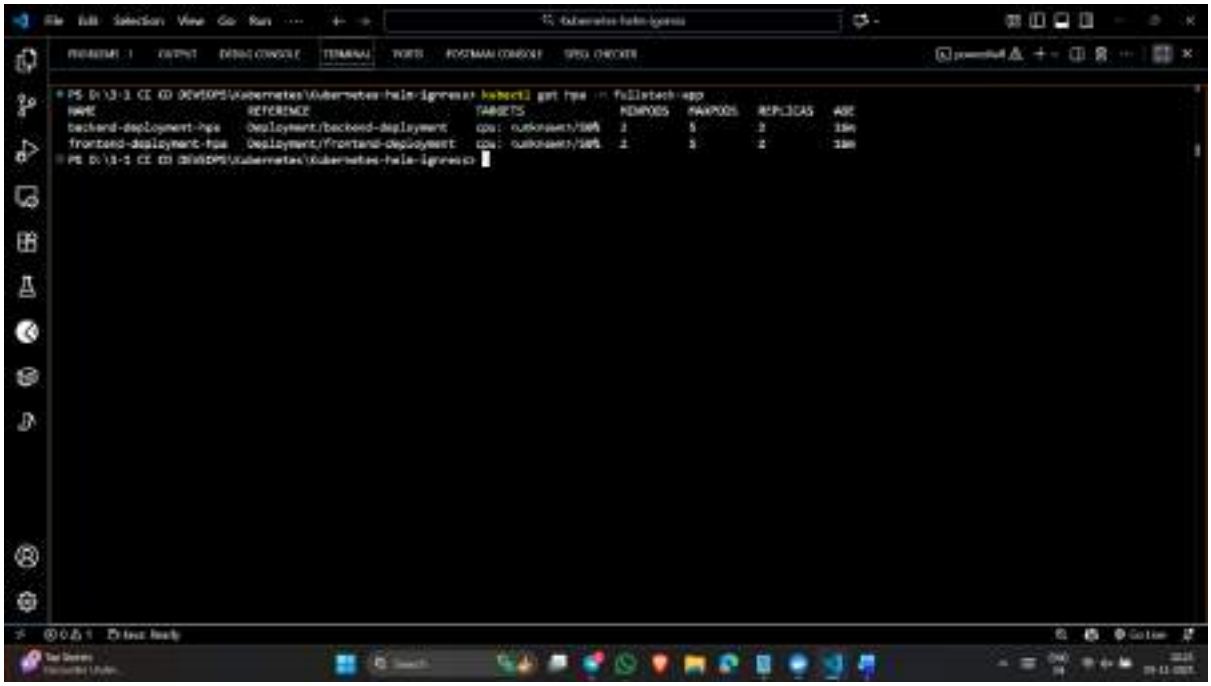

DESCRIPTION: Upgrade complete
RESOURCES:
==> v1/Namespace
NAME STATUS AGE
fullstack-app Active 10m
==> v1/Service
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
backend-deployment-service NodePort 10.99.185.138 <none> 2000:30025/TCP 10m
frontend-deployment-service NodePort 10.105.242.133 <none> 80:30080/TCP 10m
mysql-service ClusterIP 10.99.3.3 <none> 3306/TCP 10m
==> v1/Deployment
NAME READY UP-TO-DATE AVAILABLE AGE
backend-deployment 2/2 2 2 10m
frontend-deployment 3/3 3 3 10m
mysql-deployment 1/1 1 1 10m
==> v1/Pod(related)
NAME READY STATUS RESTARTS AGE
backend-deployment-66d98c7c7b-6d2fk 1/1 Running 0 10m
backend-deployment-66d98c7c7b-sfvig 1/1 Running 0 10m
frontend-deployment-5587f8667b-hlqn9 1/1 Running 0 10m
frontend-deployment-5587f8667b-wnbjs 1/1 Running 0 10m
mysql-deployment-78c4d58897-s2b9k 1/1 Running 0 10m
==> v1/HorizontalPodAutoscaler
NAME REFERENCE TARGETS MINPODS MAXPODS REPLICAS AGE
backend-deployment-hpa Deployment/backend-deployment cpu: <unknown>/50% 2 5 2 10m
frontend-deployment-hpa Deployment/frontend-deployment cpu: <unknown>/50% 2 5 2 10m
==> v1/Ingress
NAME CLASS HOSTS ADDRESS PORTS AGE


PS D:\3-1 CE CD DEVOPS\Kubernetes\Kubernetes-helm-ingress> helm get manifest fullstack-app -n fullstack-app
---
Source: fullstack-app/templates/namespace.yml
apiVersion: v1
kind: Namespace
metadata:
name: fullstack-app

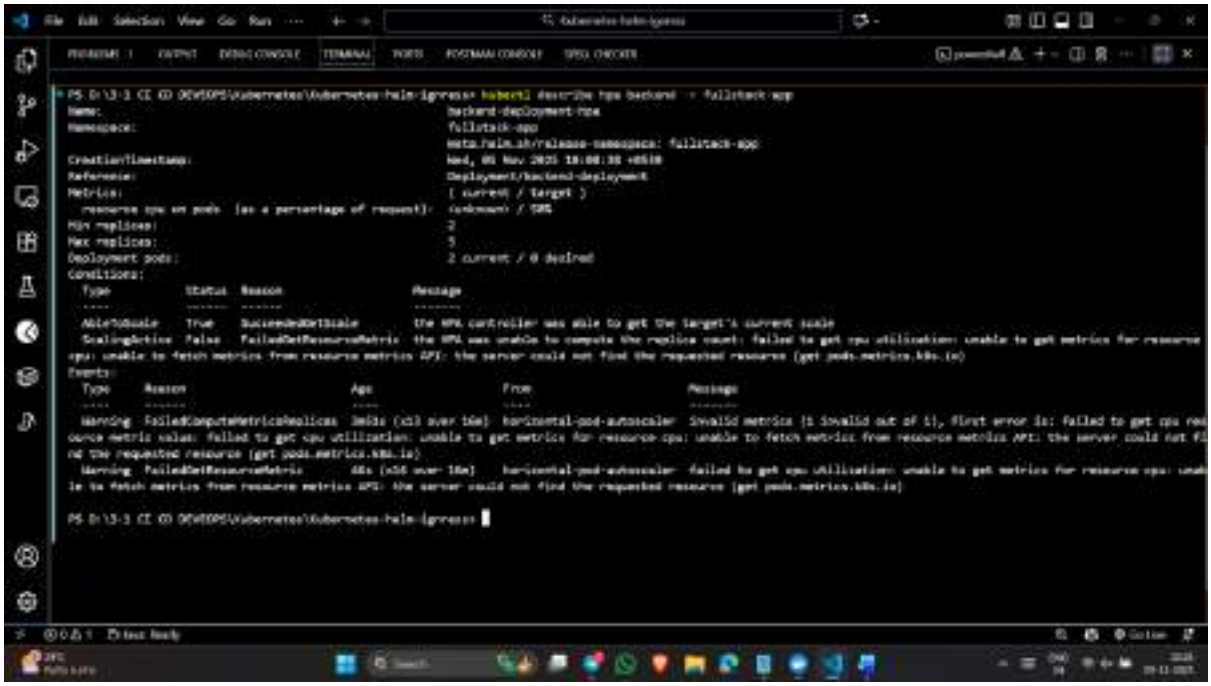
Source: fullstack-app/templates/backend-deployment.yml
apiVersion: v1
kind: Service
metadata:
name: backend-deployment-service
namespace: fullstack-app
spec:
selector:
app: backend-deployment
ports:
- protocol: TCP
port: 2000
targetPort: 2000
nodeport: 30025
type: NodePort

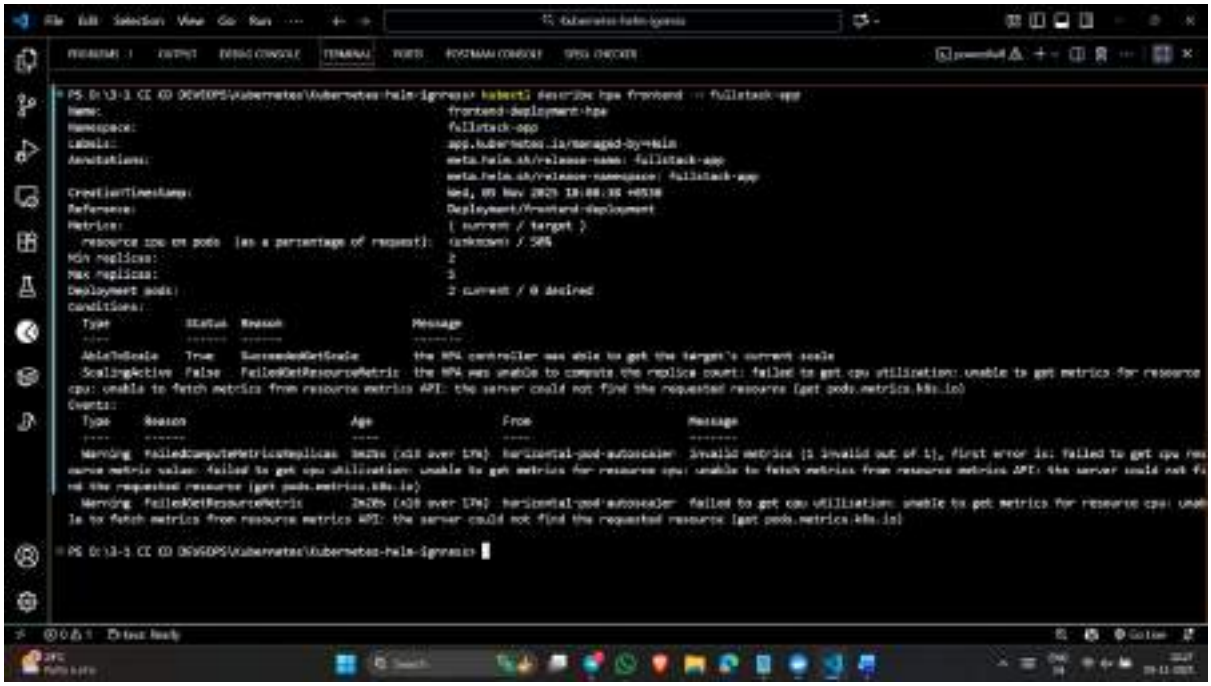
Source: fullstack-app/templates/frontend-deployment.yml
apiVersion: v1
kind: Service
metadata:
name: frontend-deployment-service
namespace: fullstack-app
spec:
selector:
app: frontend-deployment

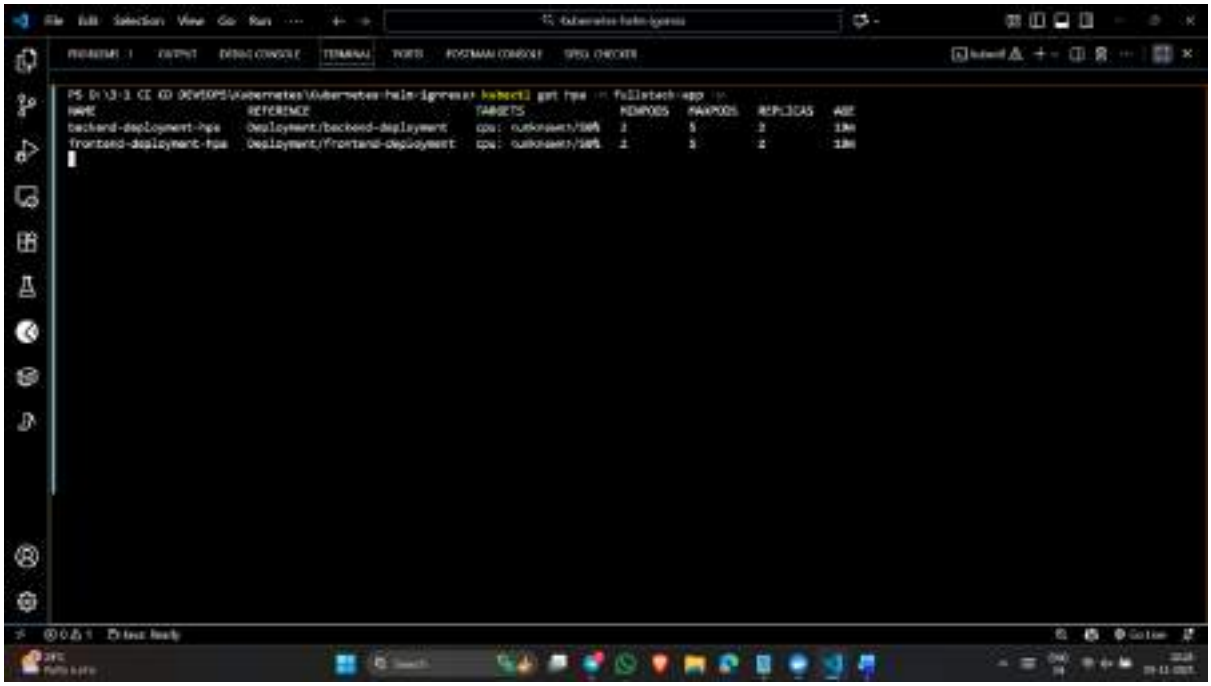

PS D:\3-1 CE ED DEVOPS\Kubernetes\Kubernetes-helm-ingress> helm history fullstack-app -n fullstack-app
REVISION UPDATED STATUS CHART APP VERSION DESCRIPTION
1 Wed Nov 5 10:09:30 2025 superseded fullstack-app-1.0.0 1.0 Install complete
2 Wed Nov 5 10:09:44 2025 deployed fullstack-app-1.0.0 1.0 Upgrade complete
PS D:\3-1 CE ED DEVOPS\Kubernetes\Kubernetes-helm-ingress>

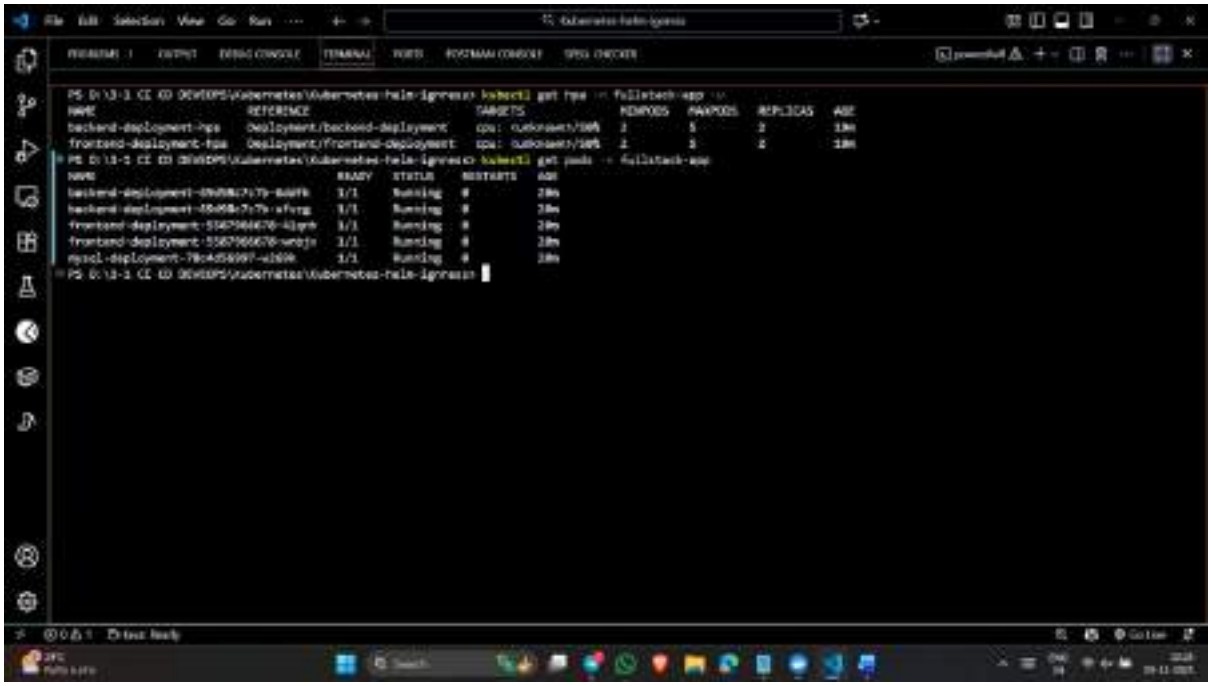

PS D:\3-1 CE ED DEVOPS\Kubernetes\Kubernetes-helm-ingress> kubectl get hpa -n fullstack-app
NAME REFERENCE TARGETS MINPODS MAXPODS REPLICAS AGE
backend-deployment-hpa Deployment/backend-deployment cpu: <unknown>/50% 2 5 2 16m
frontend-deployment-hpa Deployment/frontend-deployment cpu: <unknown>/50% 2 5 2 16m
PS D:\3-1 CE ED DEVOPS\Kubernetes\Kubernetes-helm-ingress>

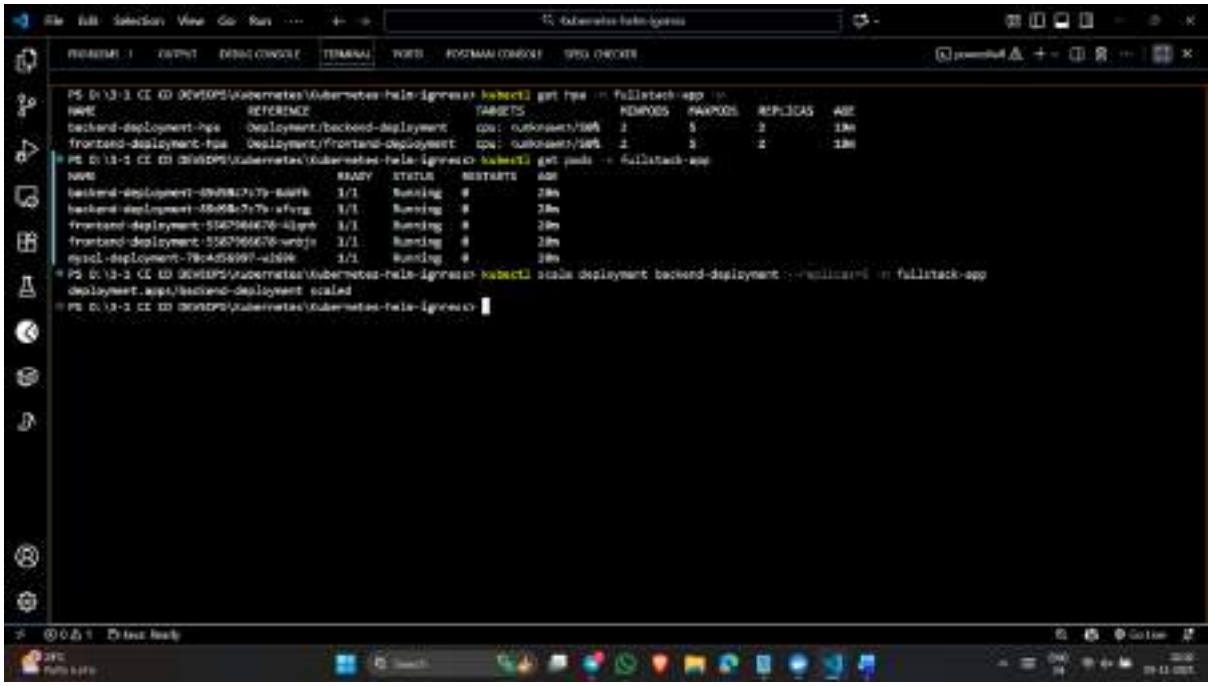
Horizontal pod autoscaling

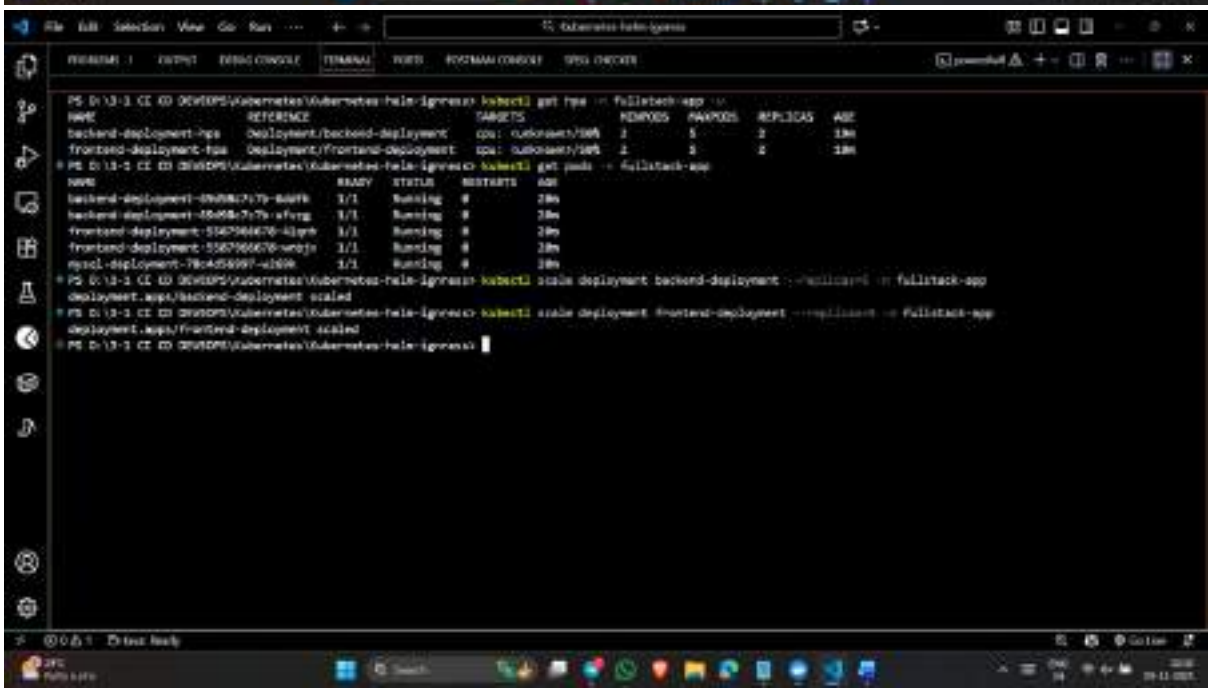




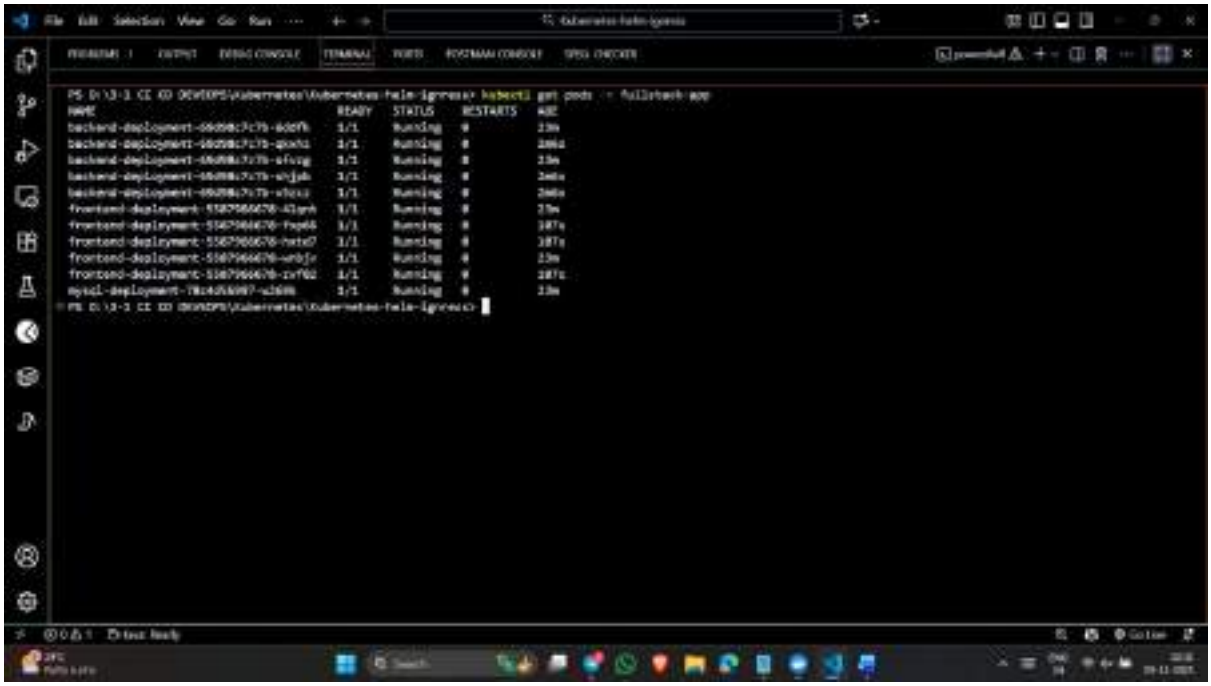

PS D:\3-1 CI CD DEVOPS\kubernetes\kubernetes-helm-ingress> kubectl get hpa -n fullstack-app
NAME REFERENCE TARGETS MINPODS MAXPODS REPLICAS AGE
backend-deployment-hpa Deployment/backend-deployment cpu: <unknown>/50% 2 5 2 19m
frontend-deployment-hpa Deployment/frontend-deployment cpu: <unknown>/50% 2 5 2 19m

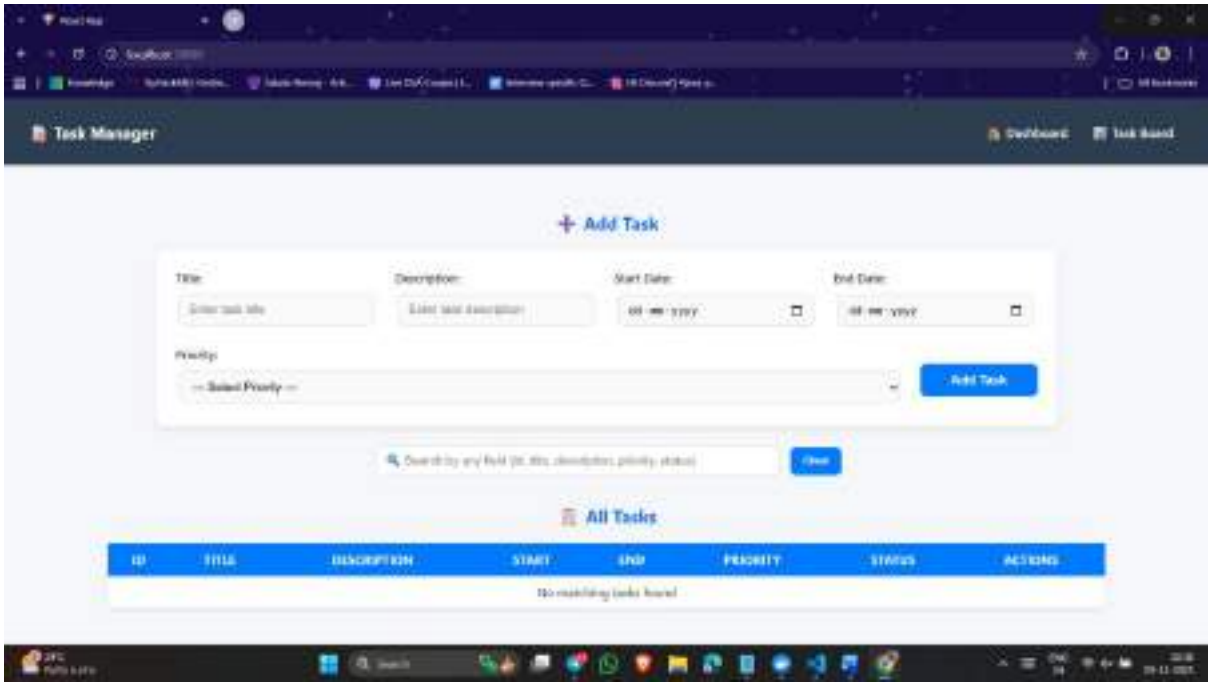

PS D:\3-1 CI CD DEVOPS\kubernetes\kubernetes-helm-ingress> kubectl get hpa -n fullstack-app
NAME REFERENCE TARGETS MINPODS MAXPODS REPLICAS AGE
backend-deployment-hpa Deployment/backend-deployment cpu: <unknown>/50% 2 5 2 19m
frontend-deployment-hpa Deployment/frontend-deployment cpu: <unknown>/50% 2 5 2 19m
PS D:\3-1 CI CD DEVOPS\kubernetes\kubernetes-helm-ingress> kubectl get pods -n fullstack-app
NAME READY STATUS RESTARTS AGE
backend-deployment-69d98c7c7b-6s8fk 1/1 Running 0 19m
backend-deployment-69d98c7c7b-xfvzg 1/1 Running 0 19m
frontend-deployment-5567966678-4lqrh 1/1 Running 0 19m
frontend-deployment-5567966678-wndjv 1/1 Running 0 19m
mysql-deployment-78c4d56997-xl6tk 1/1 Running 0 19m
PS D:\3-1 CI CD DEVOPS\kubernetes\kubernetes-helm-ingress>

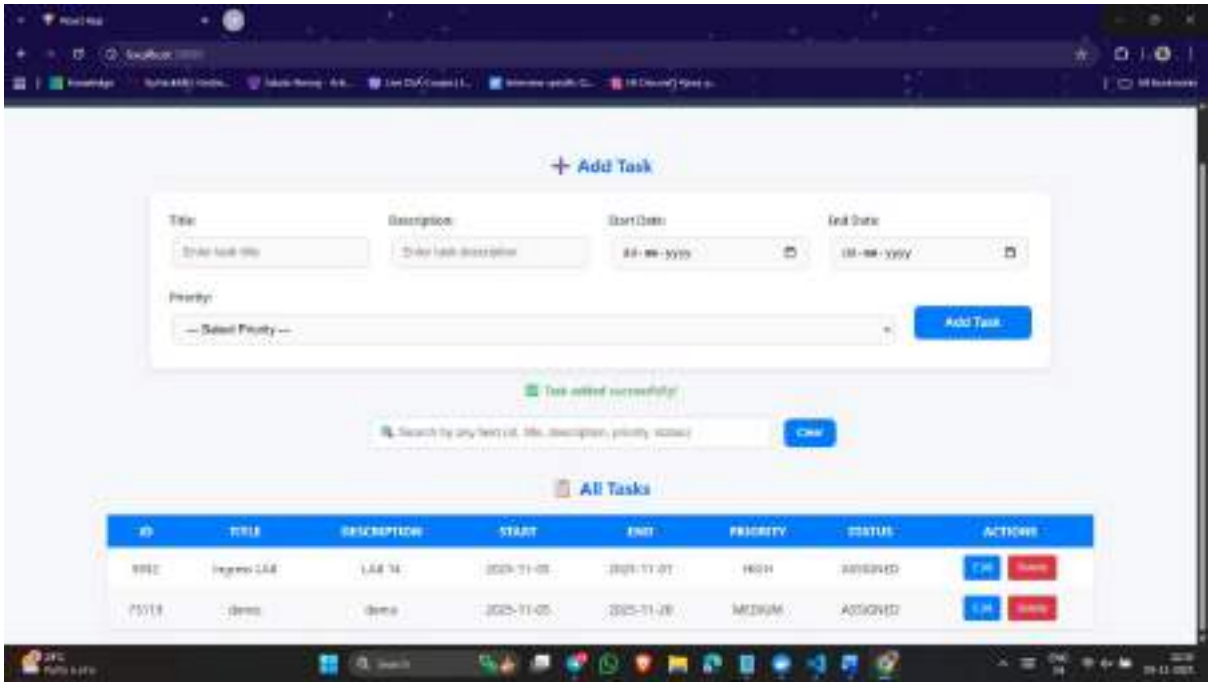

PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress> kubectl get hpa -n fullstack-app
NAME REFERENCE TARGETS MINPODS MAXPODS REPLICAS AGE
backend-deployment-hpa Deployment/backend-deployment cpu: <unknown>/50% 2 5 2 19m
frontend-deployment-hpa Deployment/frontend-deployment cpu: <unknown>/50% 2 5 2 19m
PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress> kubectl get pods -n fullstack-app
NAME READY STATUS RESTARTS AGE
backend-deployment-69d58c7c7b-6s6fb 1/1 Running 0 20m
backend-deployment-69d58c7c7b-xfvzg 1/1 Running 0 20m
frontend-deployment-5567966678-4lgnh 1/1 Running 0 20m
frontend-deployment-5567966678-wndjs 1/1 Running 0 20m
mysql-deployment-78c4d56997-xl6hh 1/1 Running 0 20m
PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress> kubectl scale deployment backend-deployment --replicas=4 -n fullstack-app
deployment.apps/backend-deployment scaled
PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress>

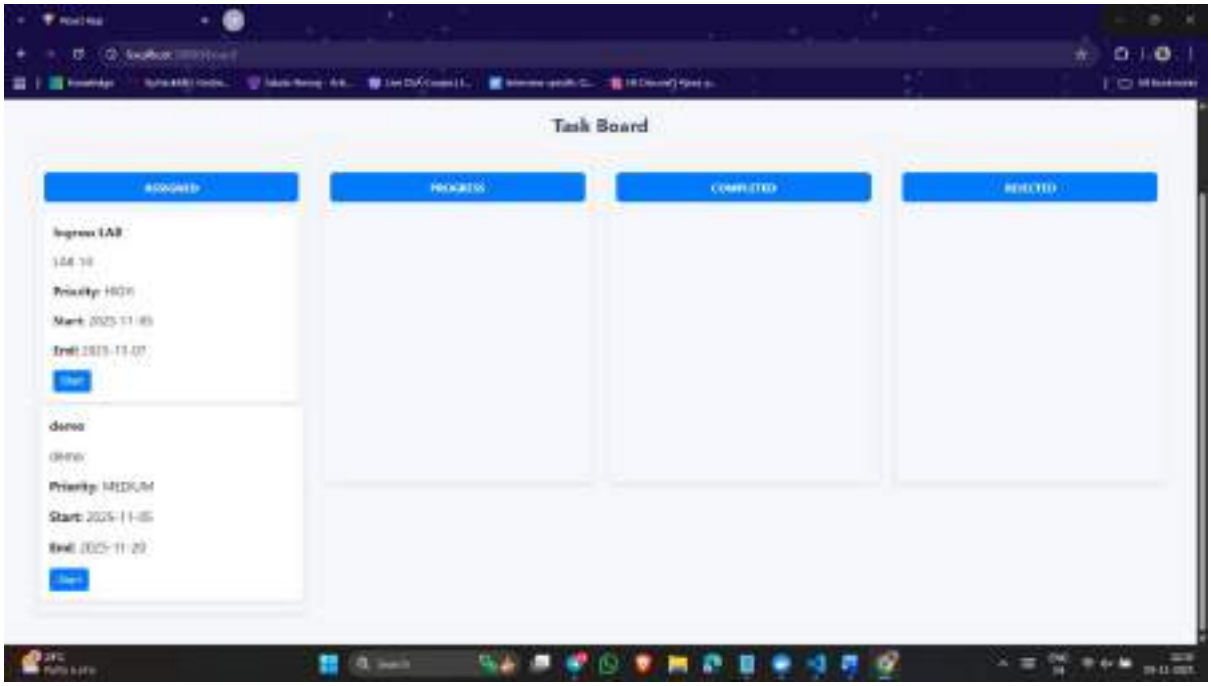

PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress> kubectl get hpa -n fullstack-app
NAME REFERENCE TARGETS MINPODS MAXPODS REPLICAS AGE
backend-deployment-hpa Deployment/backend-deployment cpu: <unknown>/50% 2 5 2 19m
frontend-deployment-hpa Deployment/frontend-deployment cpu: <unknown>/50% 2 5 2 19m
PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress> kubectl get pods -n fullstack-app
NAME READY STATUS RESTARTS AGE
backend-deployment-69d58c7c7b-6s6fb 1/1 Running 0 20m
backend-deployment-69d58c7c7b-xfvzg 1/1 Running 0 20m
frontend-deployment-5567966678-4lgnh 1/1 Running 0 20m
frontend-deployment-5567966678-wndjs 1/1 Running 0 20m
mysql-deployment-78c4d56997-xl6hh 1/1 Running 0 20m
PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress> kubectl scale deployment backend-deployment --replicas=4 -n fullstack-app
deployment.apps/backend-deployment scaled
PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress> kubectl scale deployment frontend-deployment --replicas=4 -n fullstack-app
deployment.apps/frontend-deployment scaled
PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress>

->If Image Pull back off there in the status make sure to run the command again and wait for running status

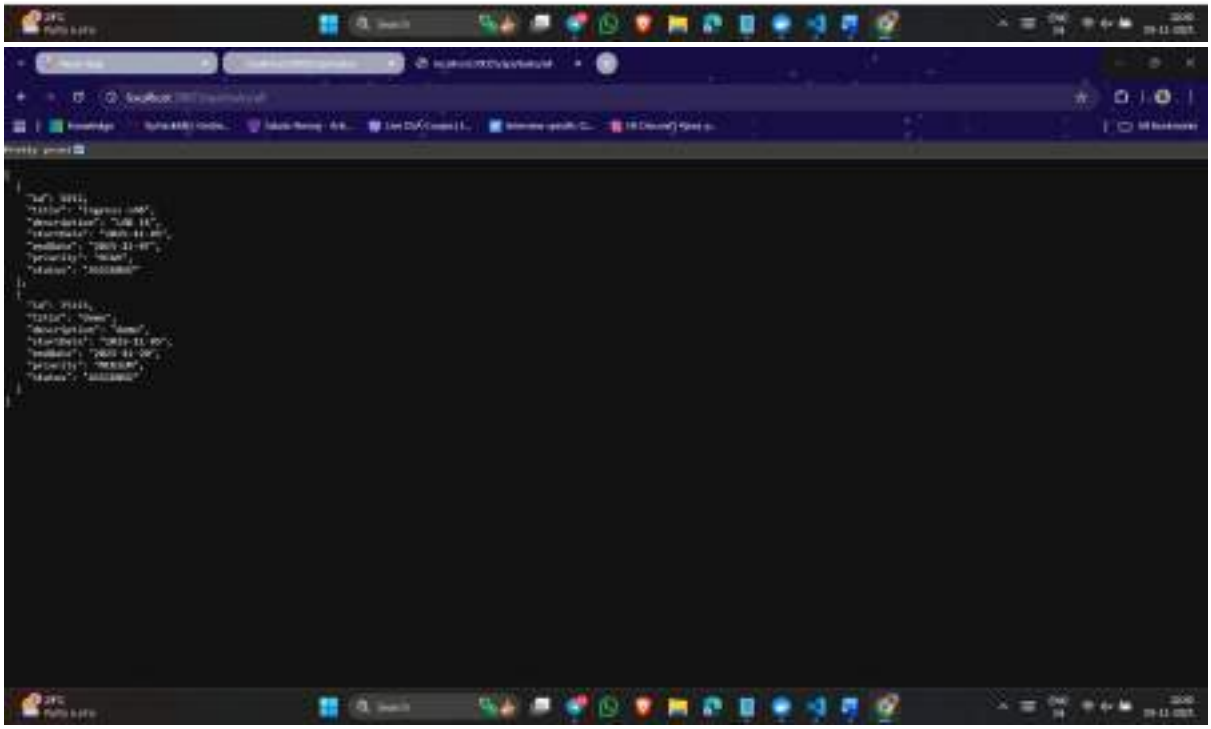


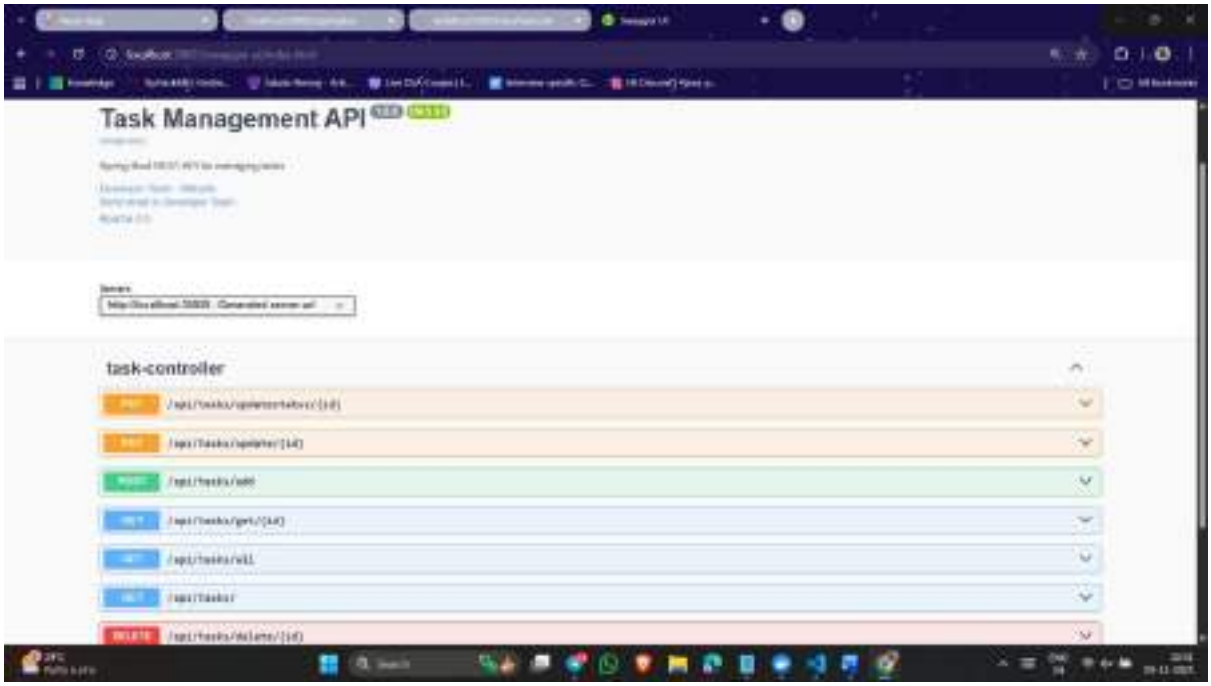

Task Manager
Dashboard
Task Board
Add Task
Title:
Description:
Start Date:
End Date:
Priority:
-- Select Priority --
Add Task
All Tasks
ID
TITLE
DESCRIPTION
START
END
PRIORITY
STATUS
ACTIONS

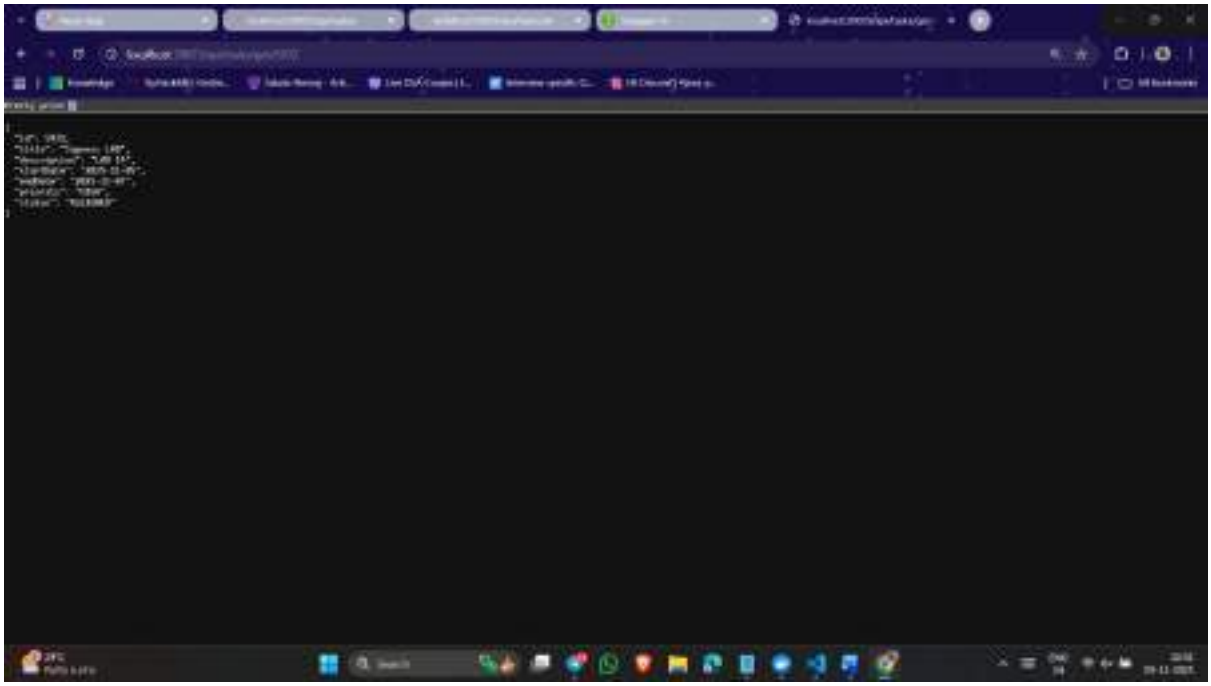

Add Task
Title:
Description:
Start Date:
End Date:
Priority:
-- Select Priority --
Add Task
All Tasks
ID
TITLE
DESCRIPTION
START
END
PRIORITY
STATUS
ACTIONS

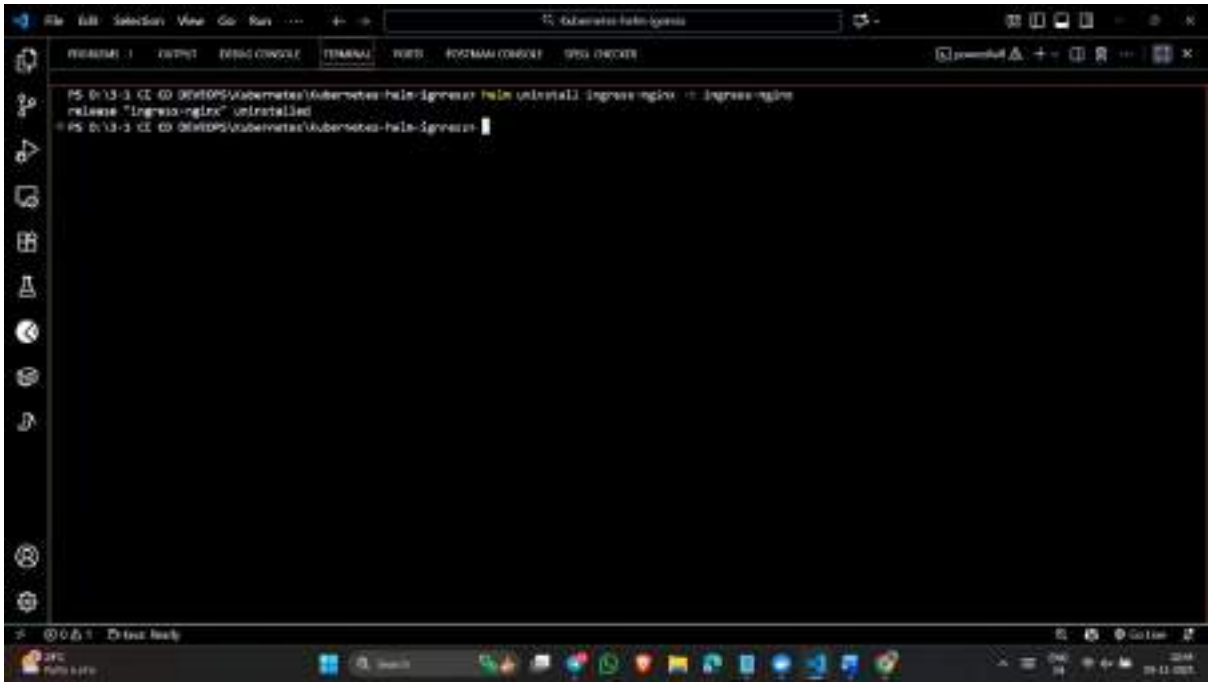

Task Board
ASSIGNED
PROGRESS
COMPLETED
REJECTED
Priority: MEDIUM

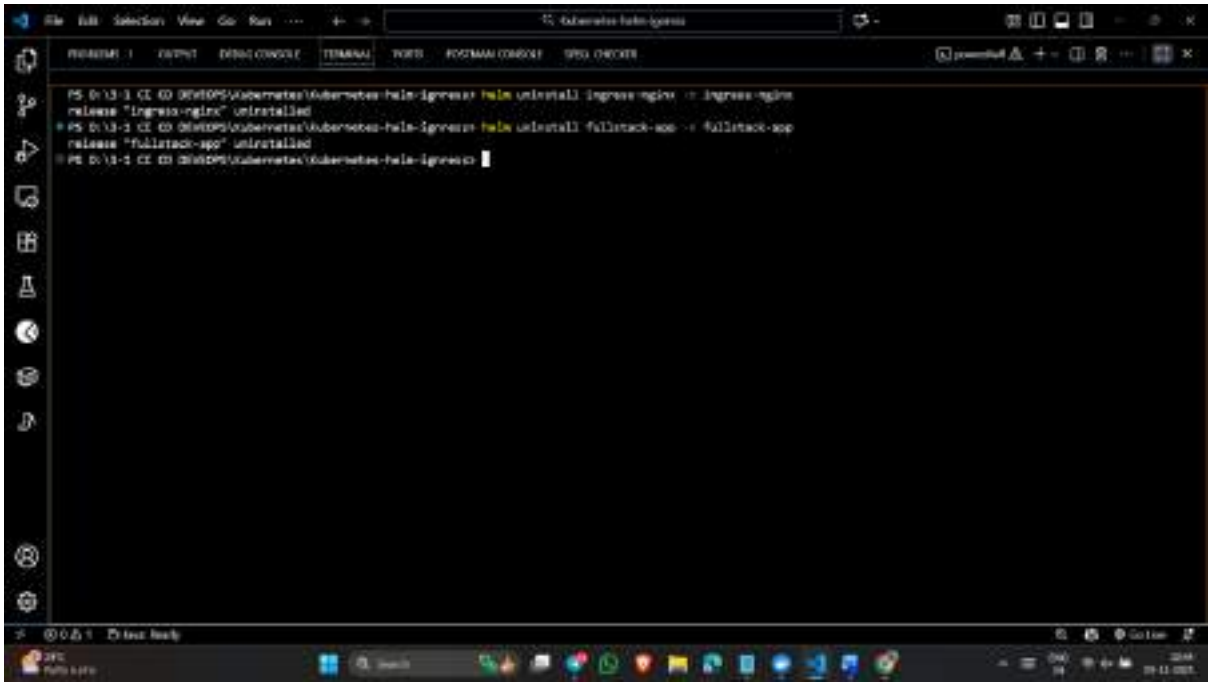
Task API Demo

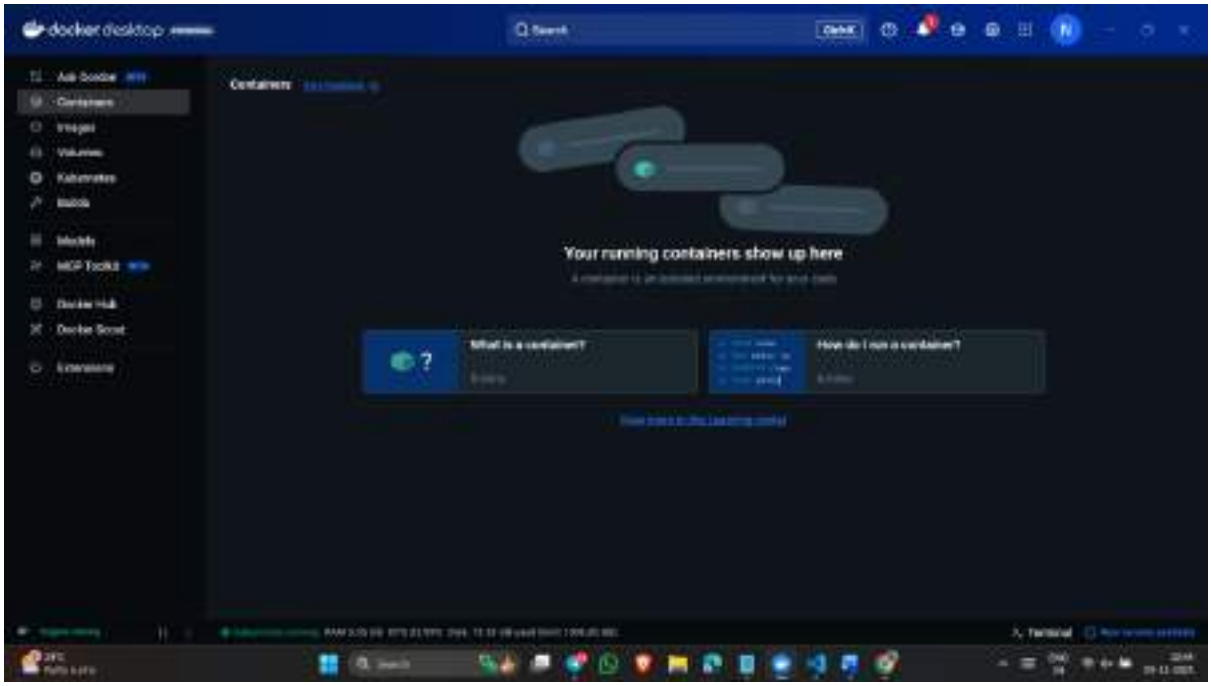


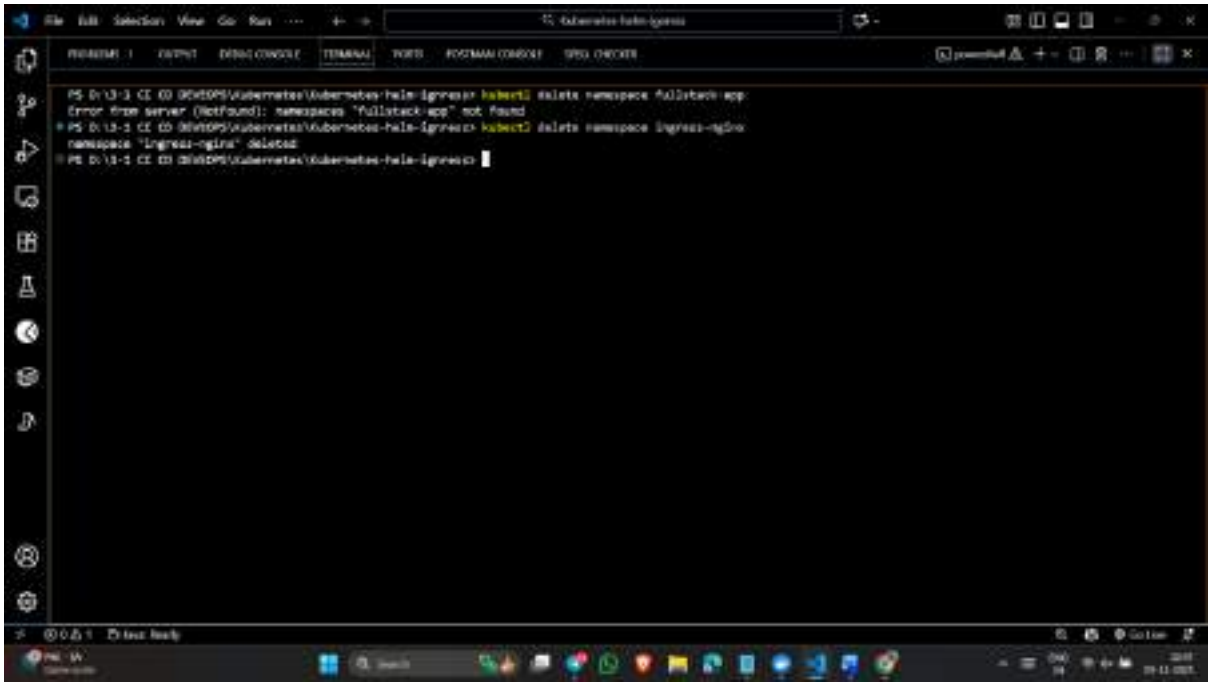

Task Management API
task-controller




PS D:\3-1 CI CD DEVOPS\kubernetes\kubernetes-helm-ingress> helm uninstall ingress-nginx -n ingress-nginx
release "ingress-nginx" uninstalled
PS D:\3-1 CI CD DEVOPS\kubernetes\kubernetes-helm-ingress>


release "ingress-nginx" uninstalled
release "fullstack-app" uninstalled


Containers
Your running containers show up here
What is a container?
How do I run a container?



🟥 **Step 1: helm create fullstack-app [This command for one time only]**

This will generate a folder named fullstack-app/ with the following structure:

fullstack-app/

├── charts/

├── templates/

├── Chart.yaml

├── values.yaml

You can then modify values.yaml and templates as per your fullstack app (frontend, backend, MySQL, etc.).

🟥 **Step 2: Add and Update Helm Repositories [These commands for first time only]**

helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx

helm repo update

helm repo list

🟥 Step 3: Install NGINX Ingress Controller

🟢 First time (creates namespace)

helm install ingress-nginx ingress-nginx/ingress-nginx --create-namespace --namespace ingress-nginx

🔁 Next time (upgrade without creating namespace)

helm upgrade ingress-nginx ingress-nginx/ingress-nginx --namespace ingress-nginx

🔍 Check Ingress Controller Pods

kubectl get pods -n ingress-nginx

🪵 View Ingress Logs

kubectl logs -f <ingress-pod-name> -n ingress-nginx

🟥 **Step 4: Install or Upgrade Your Fullstack App**

🟢 **First time (creates namespace)**

helm install fullstack-app ./fullstack-app --create-namespace --namespace fullstack-app

🔁 **Next time (upgrade without recreating namespace)**

helm upgrade fullstack-app ./fullstack-app --namespace fullstack-app

🔍 **Check Application Pods**

kubectl get pods -n fullstack-app

🪵 **View Application Logs**

kubectl logs -f <backend-pod-name> -n fullstack-app [this one important]

kubectl logs -f <frontend-pod-name> -n fullstack-app

kubectl logs -f <mysql-pod-name> -n fullstack-app

🟥 **Step 5: List Helm Releases**

helm list -n fullstack-app

🟥 **Step 6: List the services (svc) in the namespace (fullstack-app)**

kubectl get svc -n fullstack-app

🟥 **Step 7: Check Release Status**

helm status fullstack-app -n fullstack-app

🟥 **Step 8: View Manifest**

helm get manifest fullstack-app -n fullstack-app

🟥 Step 9: Check Release History

helm history fullstack-app -n fullstack-app

🟥 Step 10: Horizontal Pod Autoscaler (HPA)

🧩 Check All HPAs

kubectl get hpa -n fullstack-app

📋 Describe a Specific HPA

kubectl describe hpa backend -n fullstack-app

kubectl describe hpa frontend -n fullstack-app

📈 Watch Scaling in Real Time

kubectl get hpa -n fullstack-app -w

🔍 Check Current Pods and Resource Usage

kubectl get pods -n fullstack-app

⚙️ **Manually Scale (Optional)**

kubectl scale deployment backend-deployment --replicas=5 -n fullstack-app

kubectl scale deployment frontend-deployment --replicas=5 -n fullstack-app

🪵 **Monitor Logs During Scaling**

kubectl logs -f <backend-pod-name> -n fullstack-app

🟥 **Step 11: Monitor Pods and Scaling Activity**

kubectl get pods -n fullstack-app -w

🟥 **Step 12: Uninstall the Fullstack App**

helm uninstall fullstack-app -n fullstack-app

🟥 Step 13: Uninstall NGINX Ingress Controller

helm uninstall ingress-nginx -n ingress-nginx

🟥 Step 14: Delete Namespaces (Cleanup)

kubectl delete namespace fullstack-app

kubectl delete namespace ingress-nginx
