

LAB 14

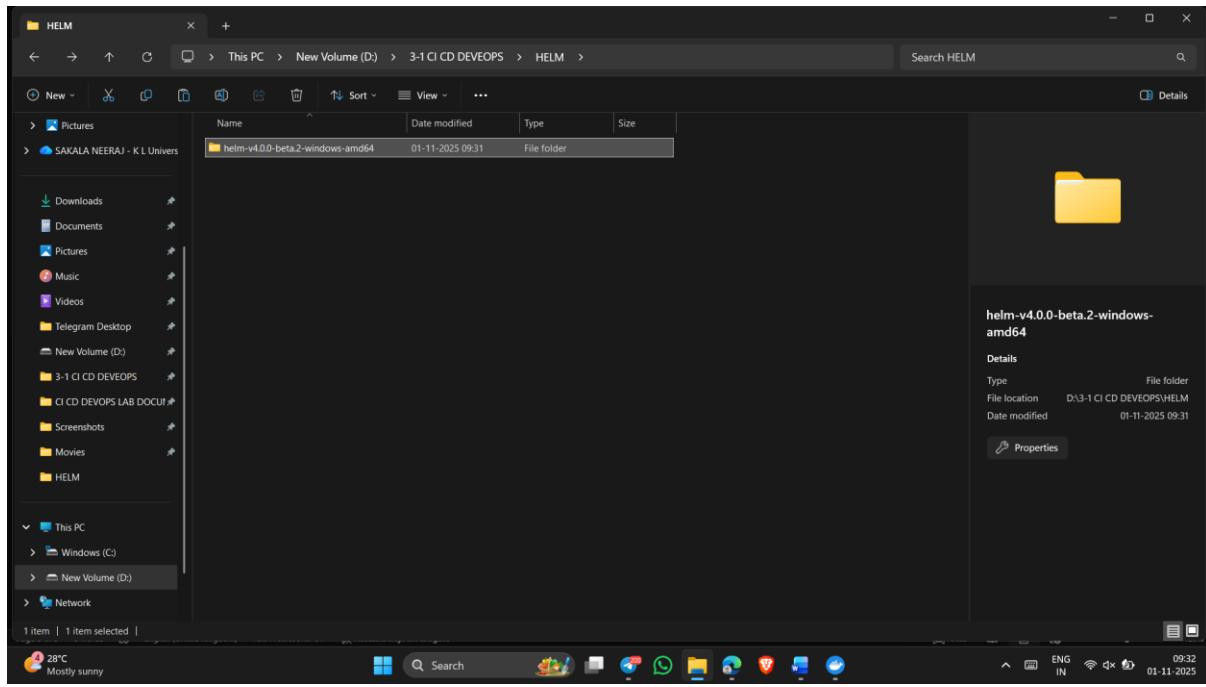
Kubernetes deployment using helm and ingress

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

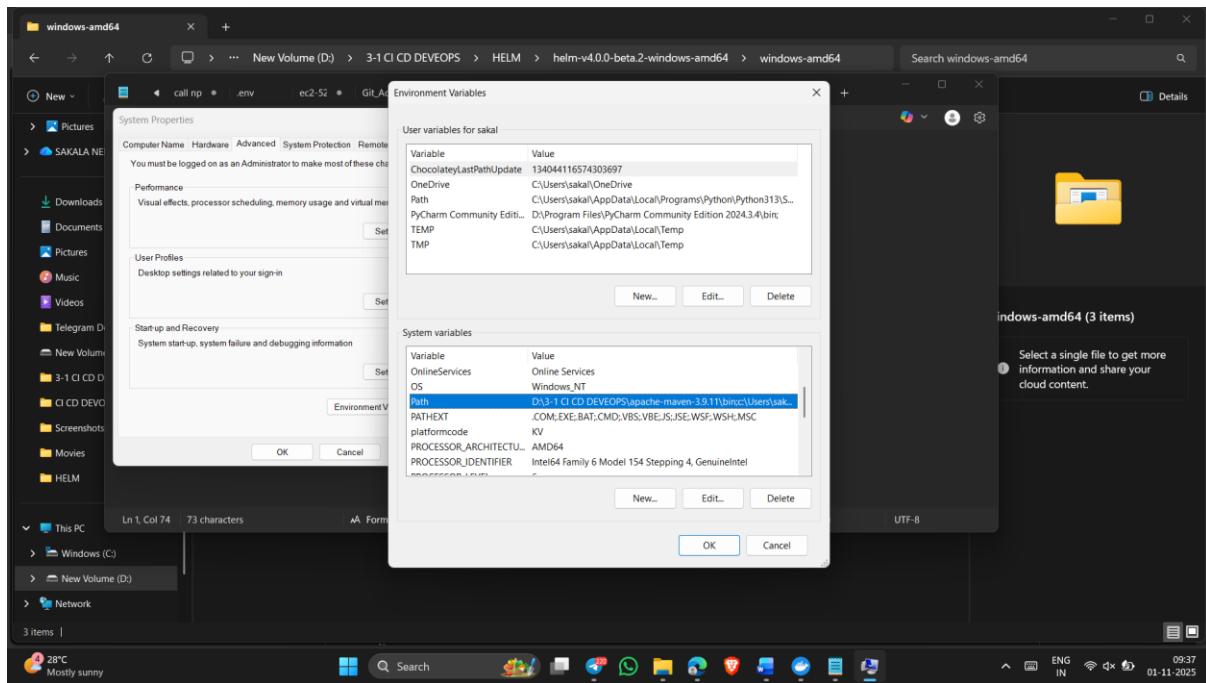
The screenshot shows the GitHub release page for Helm v4.0.0-beta.2. The page header indicates it's a pre-release from 4 days ago. The main content area is titled "v4.0.0-beta.2" and includes a note about the beta testing phase. It lists several ways to get involved: joining the Kubernetes Slack, hanging out at the Public Developer Call, and contributing to ArtifactHub. Below this is a section titled "Installation and Upgrading" which provides download links for common platforms. The links include checksums for each file.

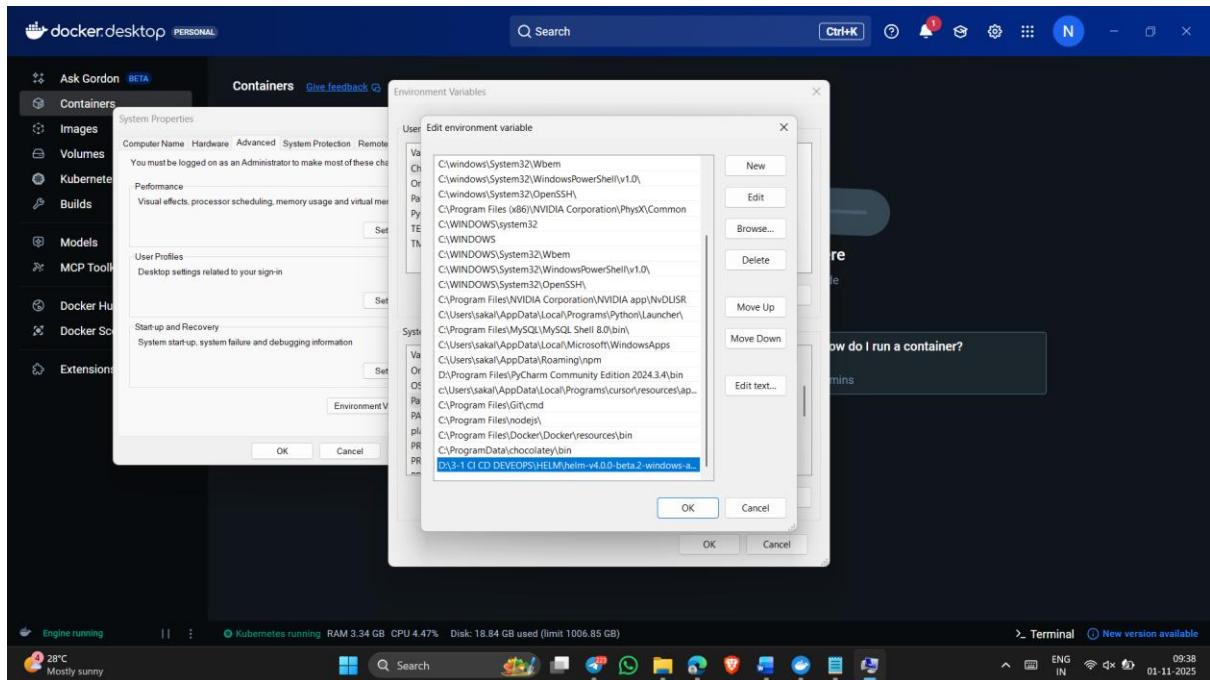
Platform	Link	Checksum
MacOS amd64	MacOS amd64	checksum
MacOS arm64	MacOS arm64	checksum
Linux amd64	Linux amd64	checksum
Linux arm	Linux arm	checksum
Linux arm64	Linux arm64	checksum
i386	i386	checksum
ppc64le	ppc64le	checksum
s390x	s390x	checksum
riscv64	riscv64	checksum
Windows amd64	Windows amd64	checksum
Windows arm64	Windows arm64	checksum

The screenshot shows a Windows File Explorer window with a dark theme. The left sidebar shows a navigation tree with "3-1 CI CD DEVEOPS" as the current folder. Inside, there are several subfolders: "apache-maven-3.9.11", "BACKEND", "DEVEOPS", "DOCKER-AWS", "FRONTEND", "HELM" (which is currently selected), "JENKINS", and "Kubernetes". The right pane displays detailed information for the selected "HELM" folder, including its type (File folder), file location (D:\3-1 CI CD DEVEOPS), and date modified (01-11-2025 09:29). There are also "Details" and "Properties" buttons.



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



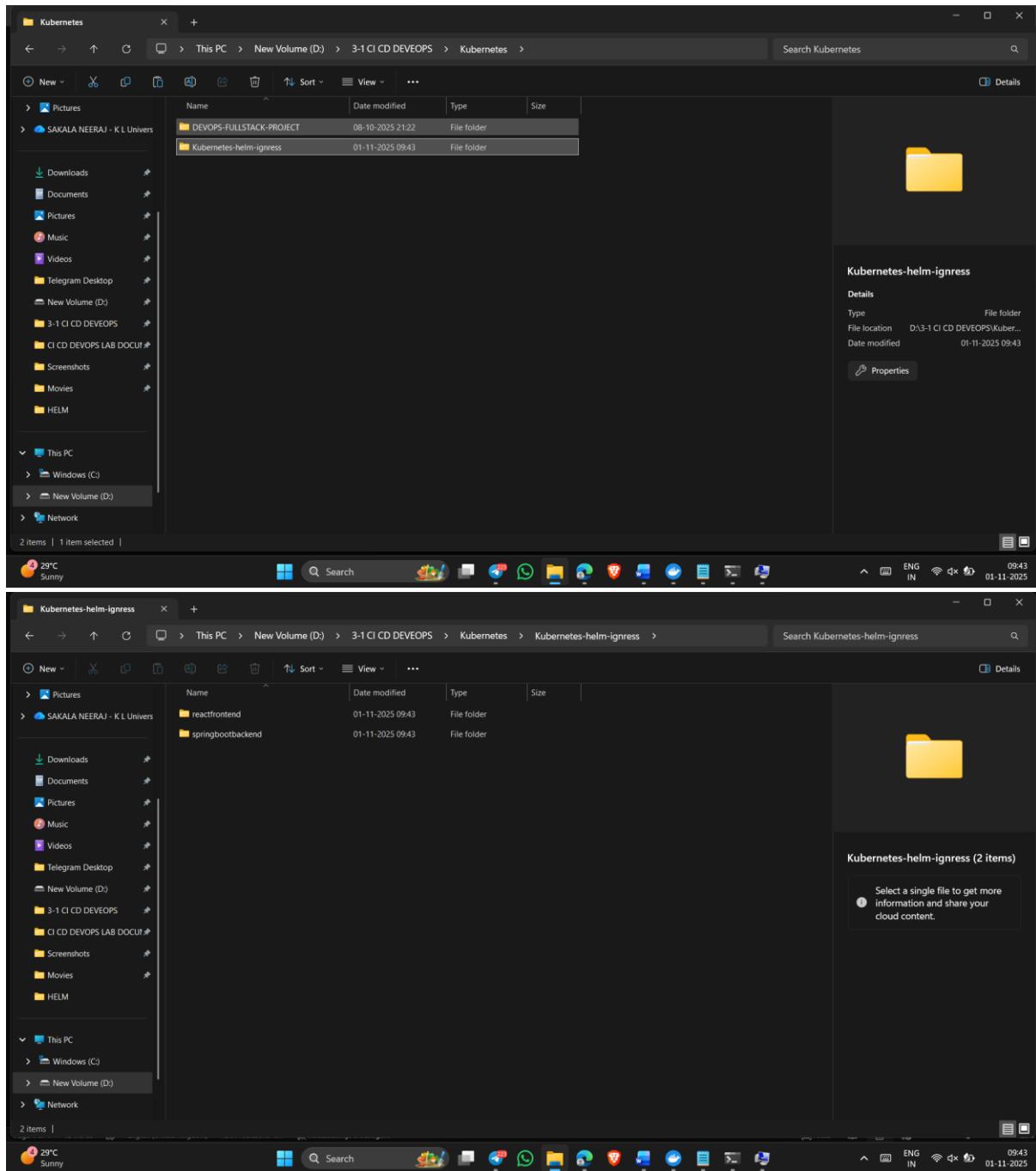


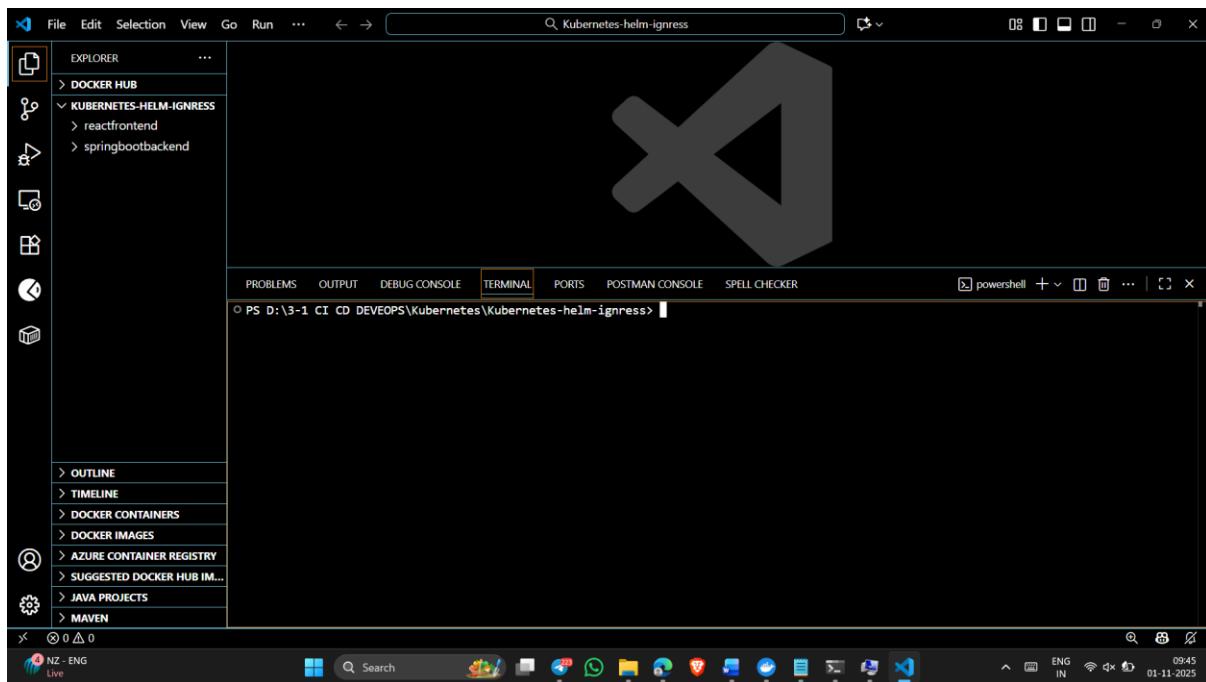
The image shows three separate windows of the Windows Command Prompt (cmd.exe) running as Administrator. Each window displays the same command and its output.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.26200.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>helm version
version.BuildInfo{Version:"v4.0.0-beta.2", GitCommit:"1da2b0a6d41482ed01bc590e83e9ac2260b92337", GitTreeState:"clean", GoVersion:"go1.24.7", KubeClientVersion:"v1.34"}
C:\Windows\System32>
```

Each window has a title bar showing "Administrator: Command Prompt". The taskbar at the bottom of the screen also shows the cmd.exe icon and the system tray with network, battery, and date/time indicators.





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

Kubernetes-helm-ingress

File Edit Selection View Go Run ... ← → 🔍 Kubernetes-helm-ingress

EXPLORER ⚡ DOCKER HUB KUBERNETES-HELM-INGRESS reactfrontend springbootbackend

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SPELL CHECKER powershell +

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> **helm** version
version.BuildInfo{Version:"v4.0.0-beta.2", GitCommit:"1da2b0a6d41482ed01bc590e83e9ac2260b92337", GitTreeState:"clean", GoVersion:"go1.24.7", KubeClientVersion:"v1.34"}

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>

OUTLINE TIMELINE DOCKER CONTAINERS DOCKER IMAGES AZURE CONTAINER REGISTRY SUGGESTED DOCKER HUB IM... JAVA PROJECTS MAVEN

Watchlist Ideas

Java: Searching... - 73% 92 files to index (C:\Users\sakal\m2\repository\org\springframework)

File Edit Selection View Go Run ... ← → 🔍 Kubernetes-helm-ingress

EXPLORER ⚡ DOCKER HUB KUBERNETES-HELM-INGRESS fullstack-app

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SPELL CHECKER powershell +

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> **helm** version
version.BuildInfo{Version:"v4.0.0-beta.2", GitCommit:"1da2b0a6d41482ed01bc590e83e9ac2260b92337", GitTreeState:"clean", GoVersion:"go1.24.7", KubeClientVersion:"v1.34"}

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> **helm** create fullstack-app
Creating fullstack-app

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>

OUTLINE TIMELINE DOCKER CONTAINERS DOCKER IMAGES AZURE CONTAINER REGISTRY SUGGESTED DOCKER HUB IM... JAVA PROJECTS MAVEN

Java: Ready

Kubernetes-helm-ingress

File Edit Selection View Go Run ... ← → 🔍 Kubernetes-helm-ingress

EXPLORER ⚡ DOCKER HUB KUBERNETES-HELM-INGRESS fullstack-app

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SPELL CHECKER powershell +

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> **helm** version
version.BuildInfo{Version:"v4.0.0-beta.2", GitCommit:"1da2b0a6d41482ed01bc590e83e9ac2260b92337", GitTreeState:"clean", GoVersion:"go1.24.7", KubeClientVersion:"v1.34"}

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> **helm** create fullstack-app
Creating fullstack-app

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>

OUTLINE TIMELINE DOCKER CONTAINERS DOCKER IMAGES AZURE CONTAINER REGISTRY SUGGESTED DOCKER HUB IM... JAVA PROJECTS MAVEN

Top Stories Breathtaking fra...

File Edit Selection View Go Run ... ← → 🔍 Kubernetes-helm-ingress

EXPLORER ⚡ DOCKER HUB KUBERNETES-HELM-INGRESS fullstack-app

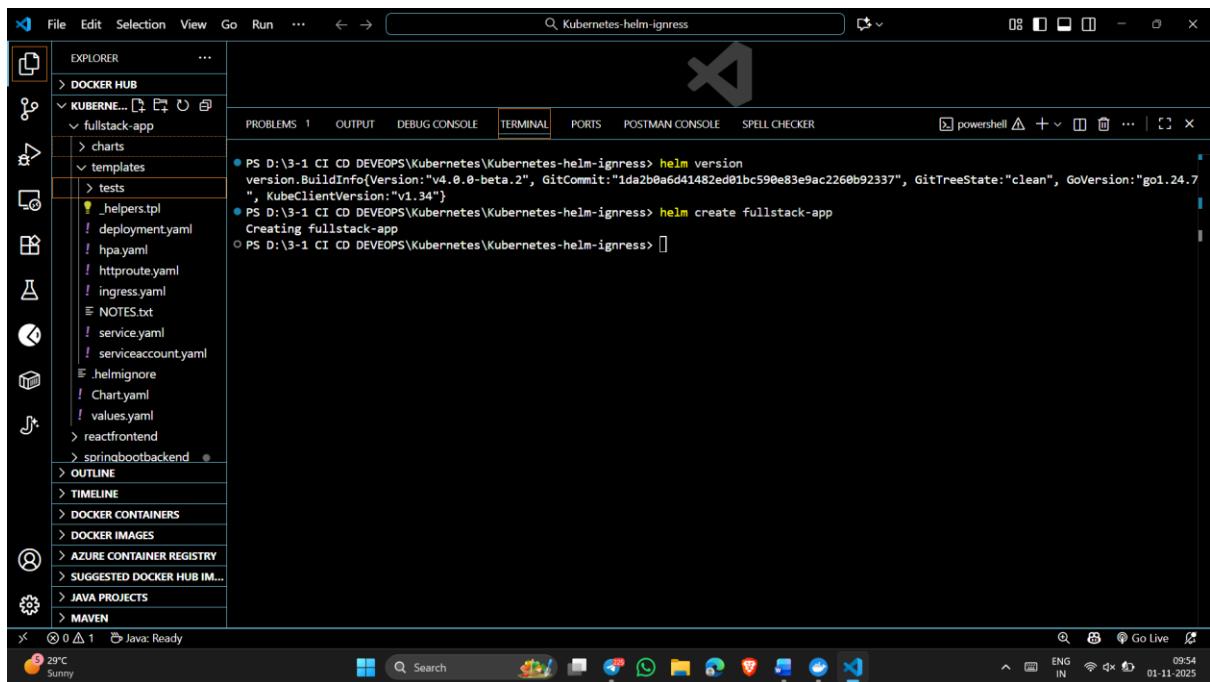
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SPELL CHECKER powershell +

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> **helm** version
version.BuildInfo{Version:"v4.0.0-beta.2", GitCommit:"1da2b0a6d41482ed01bc590e83e9ac2260b92337", GitTreeState:"clean", GoVersion:"go1.24.7", KubeClientVersion:"v1.34"}

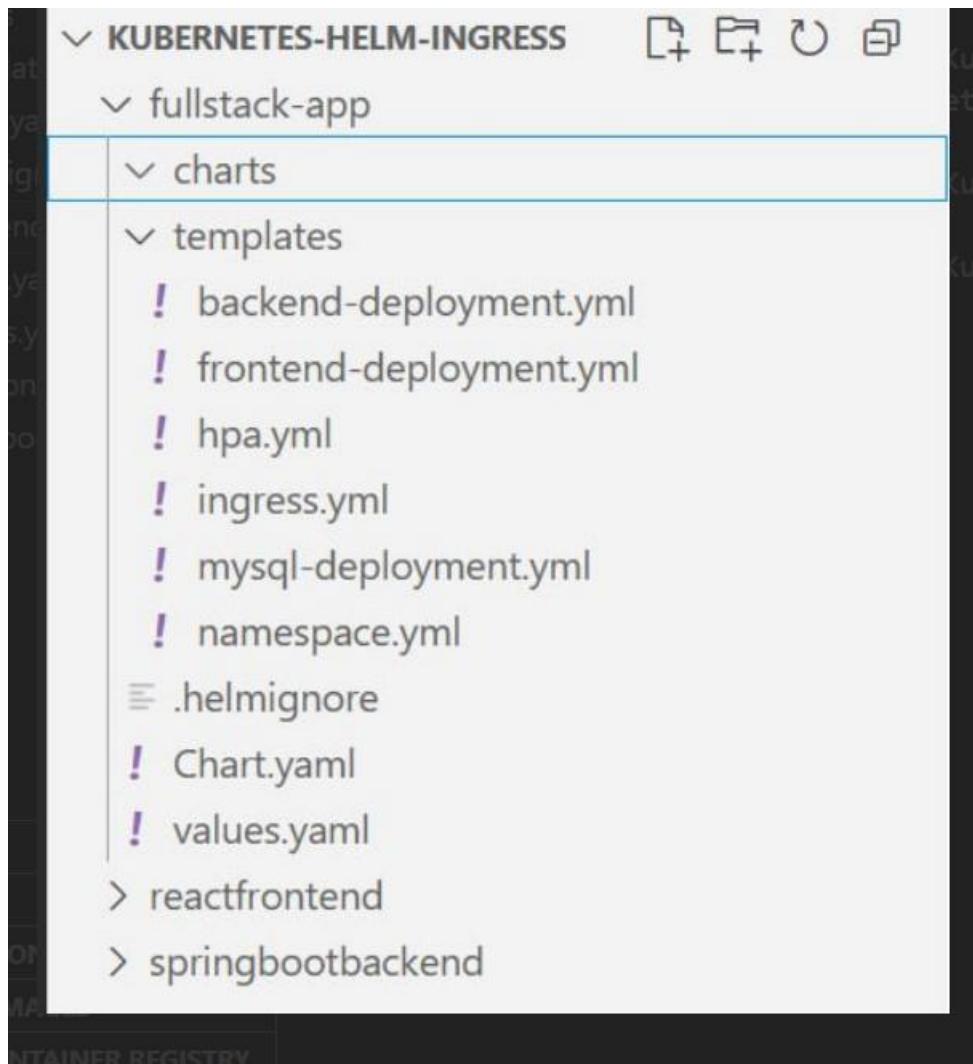
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> **helm** create fullstack-app
Creating fullstack-app

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>

OUTLINE TIMELINE DOCKER CONTAINERS DOCKER IMAGES AZURE CONTAINER REGISTRY SUGGESTED DOCKER HUB IM... JAVA PROJECTS MAVEN



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



The screenshot shows an IDE terminal window with the following output:

```
PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress> helm version
version.BuildInfo{Version:"v4.0.0-beta.2", GitCommit:"1da2b0a6d41482ed01bc590e83e9ac2260b92337", GitTreeState:"clean", GoVersion:"go1.24.7", KubeClientVersion:"v1.34"}
PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress> helm create fullstack-app
Creating fullstack-app
PS D:\3-1 CI CD DEVOPS\Kubernetes\Kubernetes-helm-ingress>
```

```

13
14 # =====
15 # Backend Configuration
16 # =====
17 backend:
18   - image: neeraj2005/k8-backend:latest
19   replicas: 2
20   port: 2000
21   nodePort: 30025
22   resources:
23     requests:
24       cpu: "100m"
25       memory: "128Mi"
26     limits:
27       cpu: "500m"
28       memory: "256Mi"

```

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> helm version
version.BuildInfo{Version:"v4.0.0-beta.2", GitCommit:"dca2b0a6d41482ed01bc590e83e9ac2260b92337", GitTreeState:"clean", GoVersion:"go1.24.7", KubeClientVersion:"v1.34"}
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> helm create fullstack-app
Creating fullstack-app
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>

->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

```

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> helm install fullstack-app ./fullstack-app
Level=ERROR msg="release name check failed" error="cannot reuse a name that is still in use"
Error: INSTALLATION FAILED: release name check failed: cannot reuse a name that is still in use
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>

```

```

apiVersion: v1
kind: Namespace
metadata:
  name: {{ .Values.namespace }}

```

helm upgrade fullstack-app ./fullstack-app

```

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> helm upgrade fullstack-app ./fullstack-app\ -n fullstack-app
Release "fullstack-app" has been upgraded. Happy Helm-ing!
NAME: fullstack-app
LAST DEPLOYED: Sat Nov 1 10:26:45 2025
NAMESPACE: fullstack-app
STATUS: deployed
REVISION: 2
DESCRIPTION: Upgrade complete
TEST SUITE: None

```

kubectl get all -n fullstack-app

```
PS D:\3-1 CI CD DEVEOPS\kubernetes\kubernetes-helm-ingress> kubectl get all -n fullstack-app
NAME                                         READY   STATUS    RESTARTS   AGE
pod/backend-deployment-69d98c7cb-8xbnw   1/1    Running   0          86s
pod/backend-deployment-69d98c7cb-fxg7p   1/1    Running   0          86s
pod/frontend-deployment-5587966678-1dt6b  1/1    Running   0          86s
pod/frontend-deployment-5587966678-qvcbf  1/1    Running   0          86s
pod/mysql-deployment-78c4d56997-9ggbv   1/1    Running   0          86s

NAME                           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
service/backend-deployment-service  NodePort   10.103.132.233 <none>       2800:30025/TCP   86s
service/frontend-deployment-service  NodePort   10.111.199.89  <none>       80:30000/TCP   86s
service/mysql-service              ClusterIP  10.111.63.156  <none>       3306/TCP     86s

NAME                           DESIRED  CURRENT  READY   AGE
deployment.apps/backend-deployment  2/2      2        2        87s
deployment.apps/frontend-deployment 2/2      2        2        87s
deployment.apps/mysql-deployment    1/1      1        1        87s

NAME                                     REFERENCE           TARGETS             MINPODS   MAXPODS   REPLICAS   AGE
horizontalpodautoscaler.autoscaling/backend-deployment   deployment/backend-deployment   cpu: <unknown>/50%  2          5          2          87s
horizontalpodautoscaler.autoscaling/frontend-deployment  deployment/frontend-deployment  cpu: <unknown>/50%  2          5          2          87s
○ PS D:\3-1 CI CD DEVEOPS\kubernetes\kubernetes-helm-ingress>
```

```
PS D:\3-1 CI CD DEVEOPS\kubernetes\kubernetes-helm-ingress> kubectl get pods -n fullstack-app
NAME                                         READY   STATUS    RESTARTS   AGE
backend-deployment-69d98c7cb-8xbnw   1/1    Running   0          108s
backend-deployment-69d98c7cb-fxg7p   1/1    Running   0          108s
frontend-deployment-5587966678-1dt6b  1/1    Running   0          108s
frontend-deployment-5587966678-qvcbf  1/1    Running   0          108s
mysql-deployment-78c4d56997-9ggbv   1/1    Running   0          108s
○ PS D:\3-1 CI CD DEVEOPS\kubernetes\kubernetes-helm-ingress>
```

React App

localhost:30080

Sakala Neeraj - Acti... Live DSA Course | L... Interview specific Q... (4) Discord | #java-p...

Task Manager

Dashboard Task Board

Add Task

Title: Description: Start Date: End Date:

Enter task title Enter task description dd-mm-yyyy dd-mm-yyyy

Priority: --- Select Priority ---

Add Task

Search by any field (id, title, description, priority, status)

All Tasks

ID	TITLE	DESCRIPTION	START	END	PRIORITY	STATUS	ACTIONS
No matching tasks found.							

PAK - SA Game score

Search

07:30 05.11.2025 ENG IN

This screenshot shows a task management application running on a local host. The interface includes a header with links to 'Dashboard' and 'Task Board'. Below the header is a 'Add Task' form with fields for Title, Description, Start Date, End Date, and Priority. A search bar is also present. The main area displays a table titled 'All Tasks' with a single row indicating 'No matching tasks found.' At the bottom, a system tray shows the date and time as 07:30 on 05.11.2025, and language settings as ENG IN.

React App

localhost:30080

Sakala Neeraj - Acti... Live DSA Course | L... Interview specific Q... (4) Discord | #java-p...

Add Task

Title: Description: Start Date: End Date:

Enter task title Enter task description dd-mm-yyyy dd-mm-yyyy

Priority: --- Select Priority ---

Add Task

Task updated successfully!

Search by any field (id, title, description, priority, status)

All Tasks

ID	TITLE	DESCRIPTION	START	END	PRIORITY	STATUS	ACTIONS
84499	CICD	LABS	2025-11-05	2025-11-06	MEDIUM	ASSIGNED	<button>Edit</button> <button>Delete</button>
96937	CICD	MOOCS	2025-11-05	2025-11-06	HIGH	ASSIGNED	<button>Edit</button> <button>Delete</button>

PAK - SA Game score

Search

07:31 05.11.2025 ENG IN

This screenshot shows the same task management application after two tasks have been added. The 'All Tasks' table now contains two rows: one for 'CICD' with 'LABS' description and another for 'CICD' with 'MOOCS' description. Both tasks are set to start on 2025-11-05 and end on 2025-11-06, with a priority of 'MEDIUM' and 'HIGH' respectively, and their status is 'ASSIGNED'. The success message 'Task updated successfully!' is visible above the search bar. The system tray at the bottom remains the same as in the previous screenshot.

ASSIGNED

CICD LABS
Priority: MEDIUM
Start: 2025-11-05
End: 2025-11-06
Start

CICD MOOCs
Priority: HIGH
Start: 2025-11-05
End: 2025-11-06
Start

PROGRESS

COMPLETED

REJECTED

PAK - SA
Game score

React App | localhost:30025/api/tasks/

Search

07:31 05-11-2025

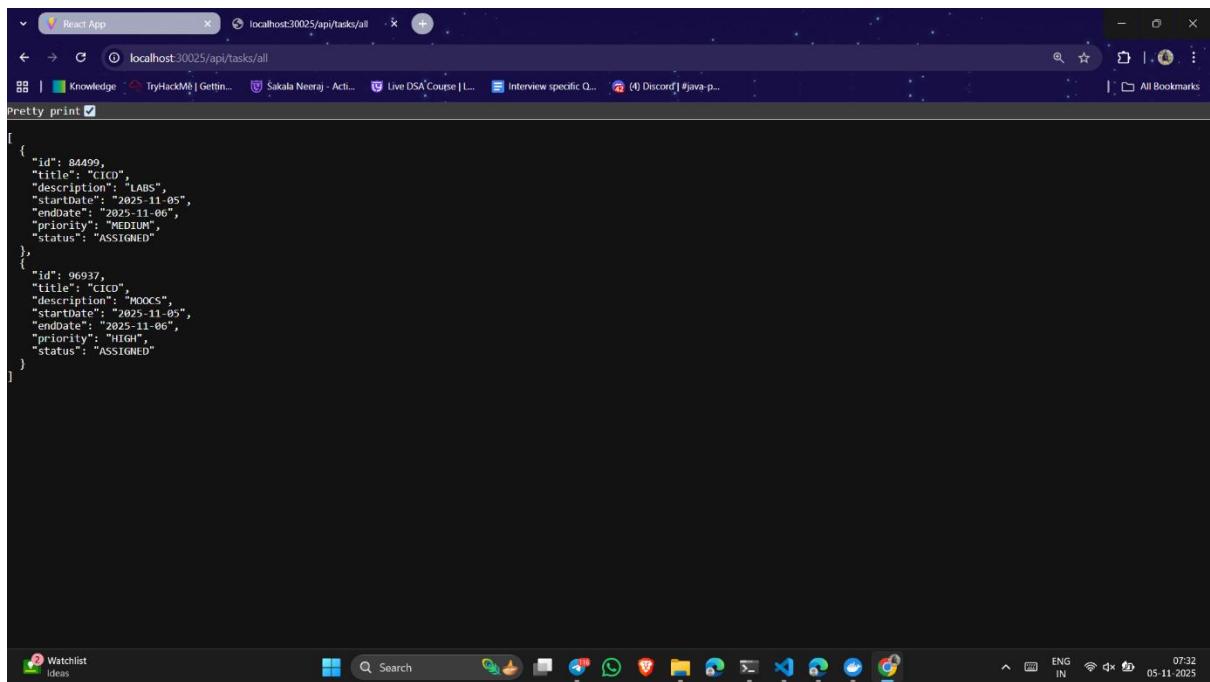
Task API Demo

Watchlist Ideas

React App | localhost:30025/api/tasks/

Search

07:32 05-11-2025

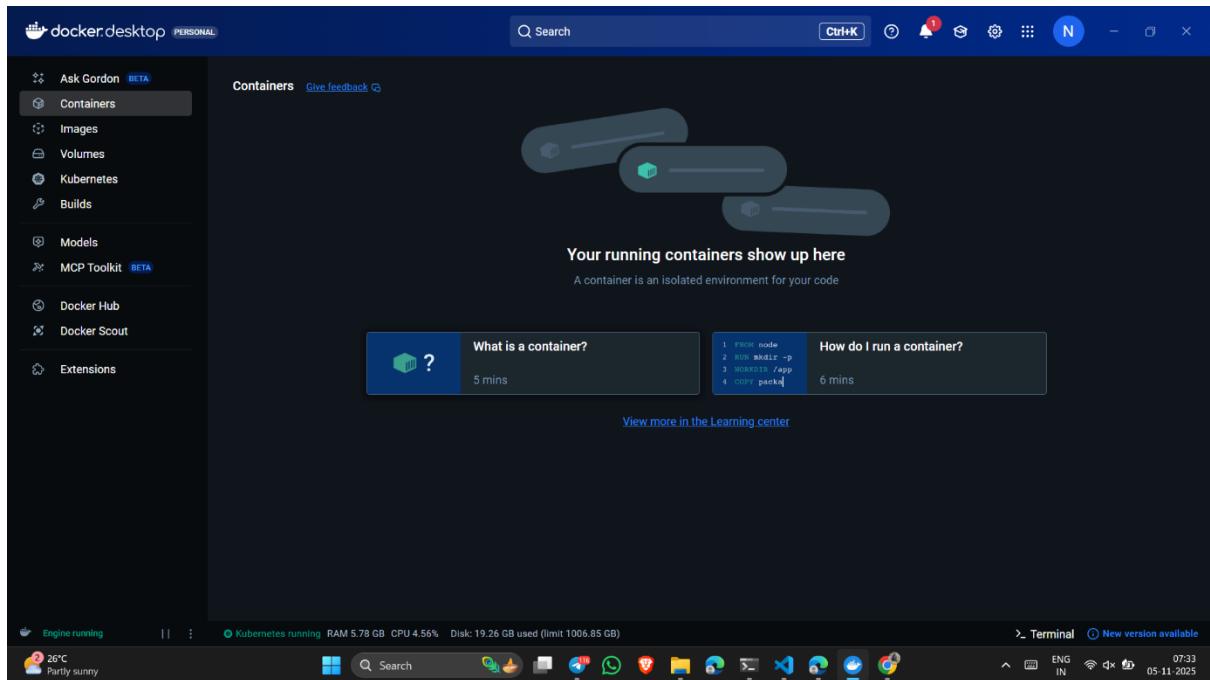


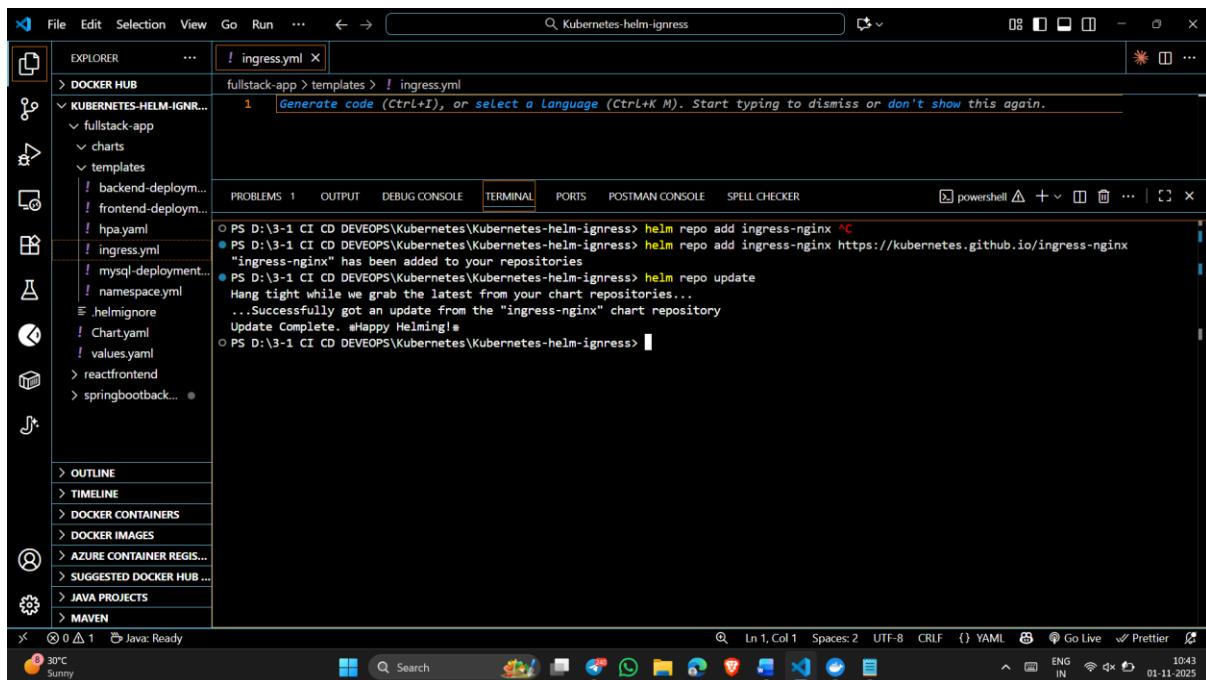
```
[  
  {  
    "id": 84499,  
    "title": "CICD",  
    "description": "LABS",  
    "startDate": "2025-11-05",  
    "endDate": "2025-11-06",  
    "priority": "MEDIUM",  
    "status": "ASSIGNED"  
  },  
  {  
    "id": 96937,  
    "title": "CICD",  
    "description": "MOOCs",  
    "startDate": "2025-11-05",  
    "endDate": "2025-11-06",  
    "priority": "HIGH",  
    "status": "ASSIGNED"  
  }  
]
```

The screenshot shows a Windows desktop environment with two main windows open:

- Browser Window (Top):** Displays the **Task Management API** documentation using Swagger UI. The URL is `localhost:30025/swagger-ui/index.html#/`. The API endpoint listed is `/api/tasks/delete/{id}`, which is highlighted in red.
- VS Code Terminal Window (Bottom):** Shows a PowerShell session running on a Windows machine. The command entered is `kubectl delete namespace fullstack-app`. The output indicates that the namespace "fullstack-app" has been deleted.

The desktop taskbar at the bottom shows various pinned icons, including a weather widget showing 26°C and Partly sunny, and system status indicators like battery level and network connection.

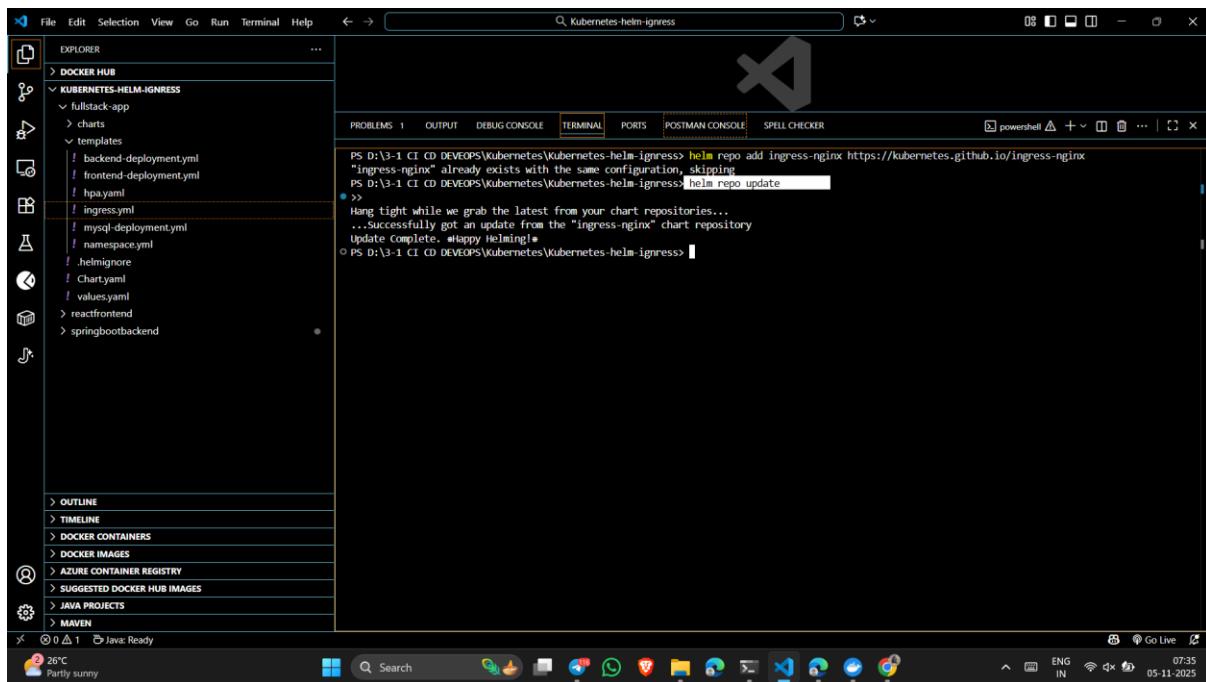




The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the following command sequence:

```
PS D:\3-1\CI\CD\DEVEOPS\Kubernetes\Kubernetes-helm-ingress> helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
"ingress-nginx" has been added to your repositories
PS D:\3-1\CI\CD\DEVEOPS\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 Helm-ing!
```

Helm with ingress

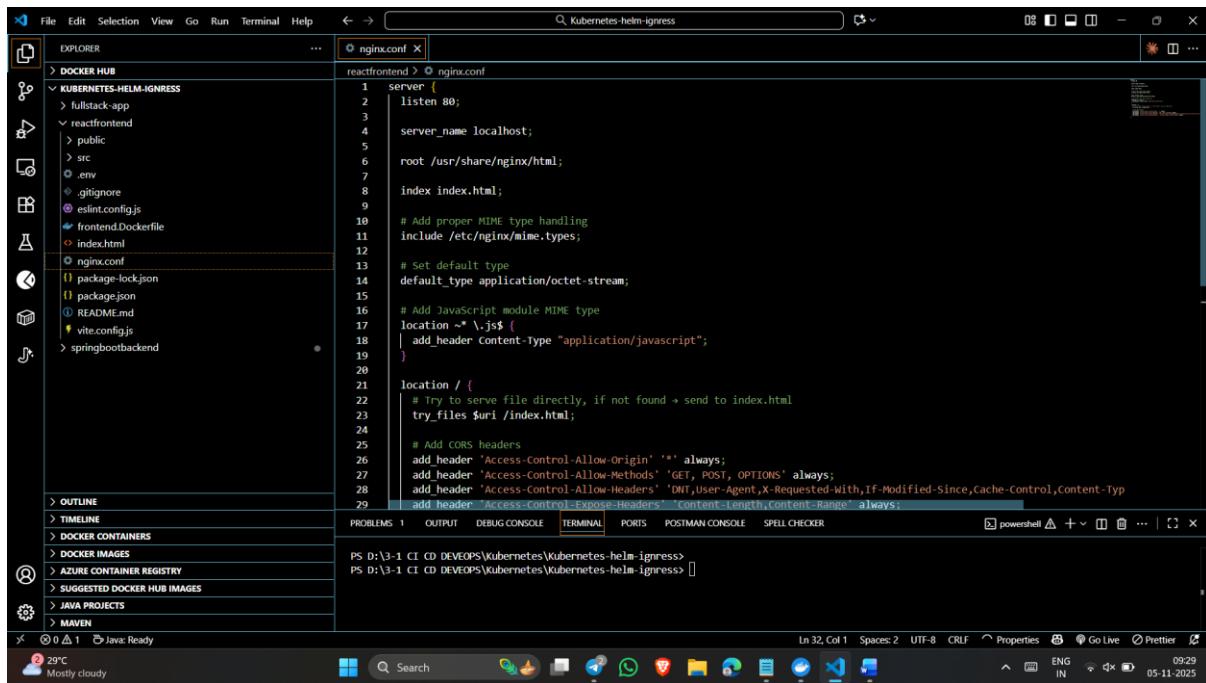
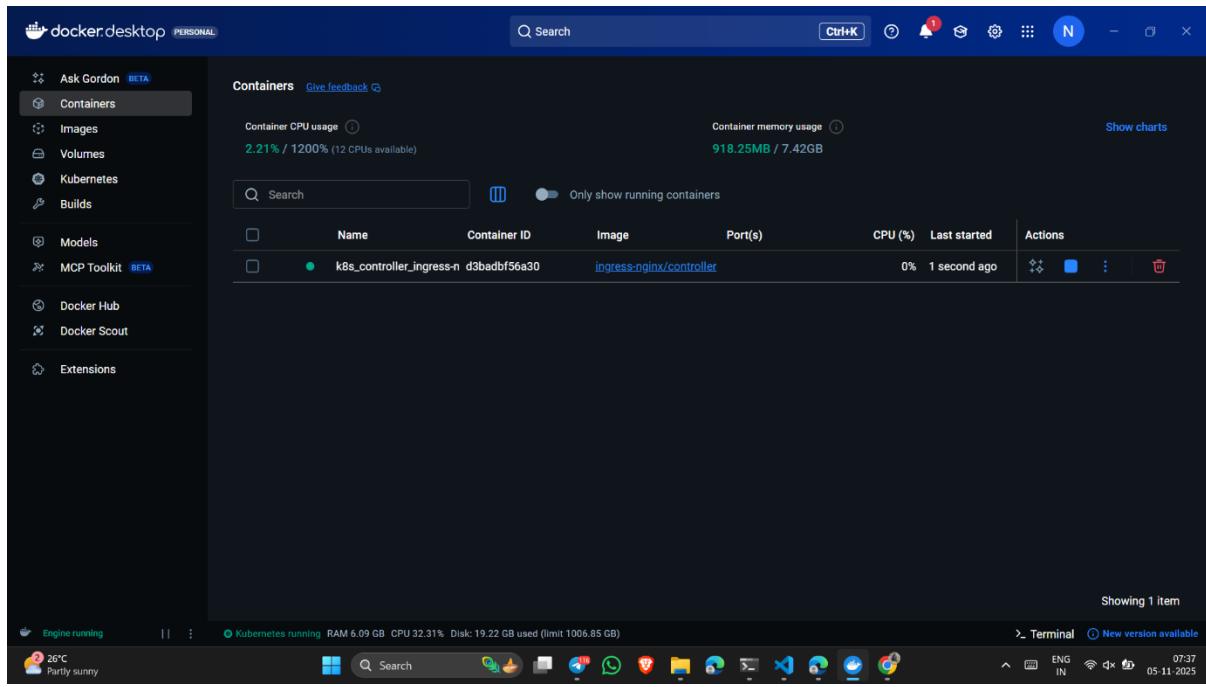


The screenshot shows the VS Code interface with the terminal tab active. The terminal window displays the following command sequence, identical to the one above but with a different timestamp:

```
PS D:\3-1\CI\CD\DEVEOPS\Kubernetes\Kubernetes-helm-ingress> helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
"ingress-nginx" already exists with the same configuration, skipping
PS D:\3-1\CI\CD\DEVEOPS\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 Helm-ing!
```

```
File Edit Selection View Go Run Terminal Help ← → 🔍 Kubernetes-helm-ingress
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SPELL CHECKER powershell + - ⌂
EXPLORER ...
DOCKER HUB
KUBERNETES-HELM-INGRESS
fullstack-app
charts
templates
backend-deployment.yml
frontend-deployment.yml
hpayaml
ingress.yml
mysql-deployment.yml
namespace.yml
.helmignore
Chart.yaml
values.yaml
reactfrontend
springbootbackend
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
AZURE CONTAINER REGISTRY
SUGGESTED DOCKER HUB IMAGES
JAVA PROJECTS
MAVEN
Java: Ready
26°C Partly sunny
powershell + - ⌂
Kubernetes-helm-ingress
PS D:\3-1 CI CD DEVEOPS\kubernetes\kubernetes-helm-ingress helm install ingress-nginx ingress-nginx -n ingress-nginx --create-namespace
● >
level=warn msg="unable to find exact version; falling back to closest available version" chart=ingress-nginx requested="" selected=4.14.0
NAME: ingress-nginx
LAST DEPLOYED: Wed Nov 5 07:36:14 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
      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:
ENG IN 07:36 05-11-2025
```

```
File Edit Selection View Go Run Terminal Help ← → 🔍 Kubernetes-helm-ingress
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SPELL CHECKER powershell + - ⌂
EXPLORER ...
DOCKER HUB
KUBERNETES-HELM-INGRESS
fullstack-app
charts
templates
backend-deployment.yml
frontend-deployment.yml
hpayaml
ingress.yml
mysql-deployment.yml
namespace.yml
.helmignore
Chart.yaml
values.yaml
reactfrontend
springbootbackend
OUTLINE
TIMELINE
DOCKER CONTAINERS
DOCKER IMAGES
AZURE CONTAINER REGISTRY
SUGGESTED DOCKER HUB IMAGES
JAVA PROJECTS
MAVEN
Java: Ready
26°C Partly sunny
powershell + - ⌂
Kubernetes-helm-ingress
PS D:\3-1 CI CD DEVEOPS\kubernetes\kubernetes-helm-ingress>
kind: Ingress
metadata:
  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
ENG IN 07:36 05-11-2025
```



Kubernetes-helm-ingress

```

! values.yaml x
fullstack-app > t values.yaml

22 # Backend
23 # =====
24 backend: neeraj2005/k8-backend:latest
25 image: neeraj2005/k8-backend:latest
26 replicas: 2
27 port: 2000
28 nodePort: 30025
29 resources:
30   requests:
31     | cpu: "100m"
32     | memory: "128Mi"
33   limits:
34     | cpu: "500m"
35     | memory: "256Mi"
36 autoscaling:
37   enabled: true
38   minReplicas: 2
39   maxReplicas: 5
40   targetCPUUtilizationPercentage: 50
41
42 # =====
43 # Frontend Configuration
44 # =====
45 frontend:
46   image: neeraj2005/k8-frontend:latest
47 replicas: 2

```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SPELL CHECKER

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>

Java: Ready

29°C Mostly cloudy

Kubernetes-helm-ingress

```

! values.yaml x
fullstack-app > t values.yaml

22 # Backend
23 # =====
24 backend: neeraj2005/k8-backend:latest
25 image: neeraj2005/k8-backend:latest
26 replicas: 2
27 port: 2000
28 nodePort: 30025
29 resources:
30   requests:
31     | cpu: "100m"
32     | memory: "128Mi"
33   limits:
34     | cpu: "500m"
35     | memory: "256Mi"
36 autoscaling:
37   enabled: true
38   minReplicas: 2
39   maxReplicas: 5
40   targetCPUUtilizationPercentage: 50
41
42 # =====
43 # Frontend Configuration
44 # =====
45 frontend:
46   image: neeraj2005/k8-frontend:latest
47 replicas: 2

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SPELL CHECKER

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> cd ..\reactfrontend
 docker build -f ./frontend.Dockerfile -t neeraj2005/k8-frontend:latest .
 DEPRECATED: The legacy builder is deprecated and will be removed in a future release. Install the buildx component to build images with BuildKit: https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 162.8kB
 Step 1/11 : FROM node:20-alpine AS build
 --> 6178e78b972f
 Step 2/11 : WORKDIR /app
 --> Using cache
 --> 5dd1a2915de7
 Step 3/11 : COPY package*.json ./
 --> Using cache
 --> 822bc42666ea
 Step 4/11 : RUN npm install
 --> Using cache
 --> 65bf5991a2a1
 Step 5/11 : COPY . .

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SPELL CHECKER

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> cd ..\reactfrontend
 docker build -f ./frontend.Dockerfile -t neeraj2005/k8-frontend:latest .
 DEPRECATED: The legacy builder is deprecated and will be removed in a future release. Install the buildx component to build images with BuildKit: https://docs.docker.com/go/buildx/

Java: Ready

29°C Mostly cloudy

Kubernetes-helm-ingress

```

PS D:\3-1 CI CD DEVEOPS\kubernetes\helm-ingress\reactfrontend docker build -f ./frontend.Dockerfile -t neeraj2005/k8-frontend:latest .
dist/assets/index-B9NgdZZ.js 269.90 kB | gzip: 88.88 kB
✓ built in 4.71s
npm notice New major version of npm available 10.8.2 > 11.6.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.6.2
npm notice To update run: npm install -g npm@11.6.2
npm notice
-> Removed intermediate container f9b8a746de6f
-> Ffbd82d2a9135
Step 7/11 : FROM nginx:alpine
-> 61001287e546
Step 8/11 : COPY nginx.conf /etc/nginx/conf.d/default.conf
-> 623c47340ea
Step 9/11 : COPY --from=build /app/dist /usr/share/nginx/html
-> 56d12341c19f
Step 10/11 : EXPOSE 80
-> Running in 5d1d84ded38d
-> Removed intermediate container 5d1d84ded38d
-> 03d43c0e1296
Step 11/11 : CMD ["nginx", "-g", "daemon off;"]
-> Running in b55b7b00cc55
-> Removed intermediate container b55b7b00cc55
-> 2c4224223170
Successfully built 2c4224223170
Successfully tagged neeraj2005/k8-frontend:latest
SECURITY WARNING: You are building a Docker image from windows against a non-Windows Docker host. All files and directories added to build context will have '-rwxr-xr-x' permissions. It is recommended to double check and reset permissions for sensitive files and directories.
○ PS D:\3-1 CI CD DEVEOPS\kubernetes\helm-ingress\reactfrontend

```

Java: Ready

29°C Mostly cloudy

Kubernetes-helm-ingress

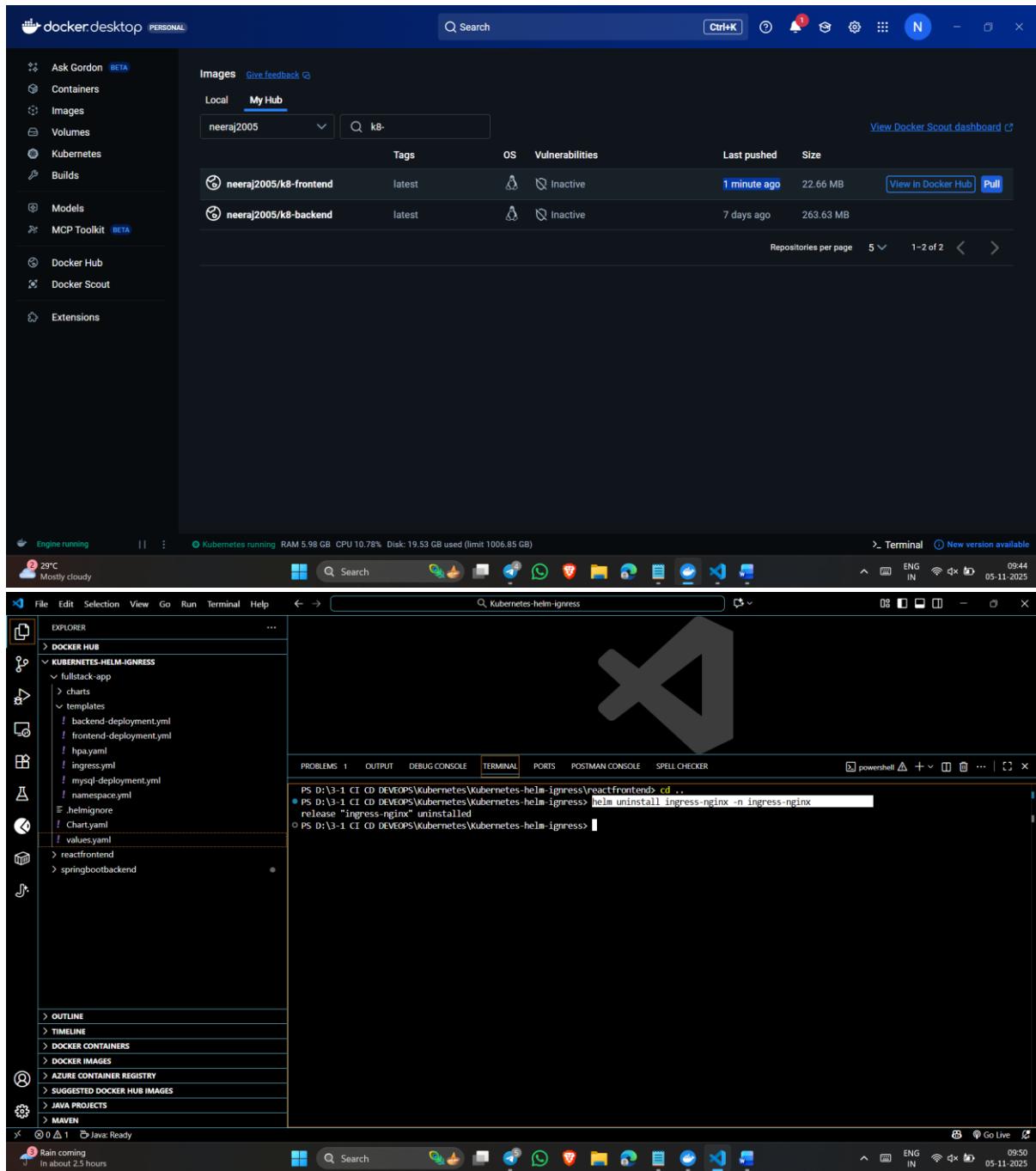
```

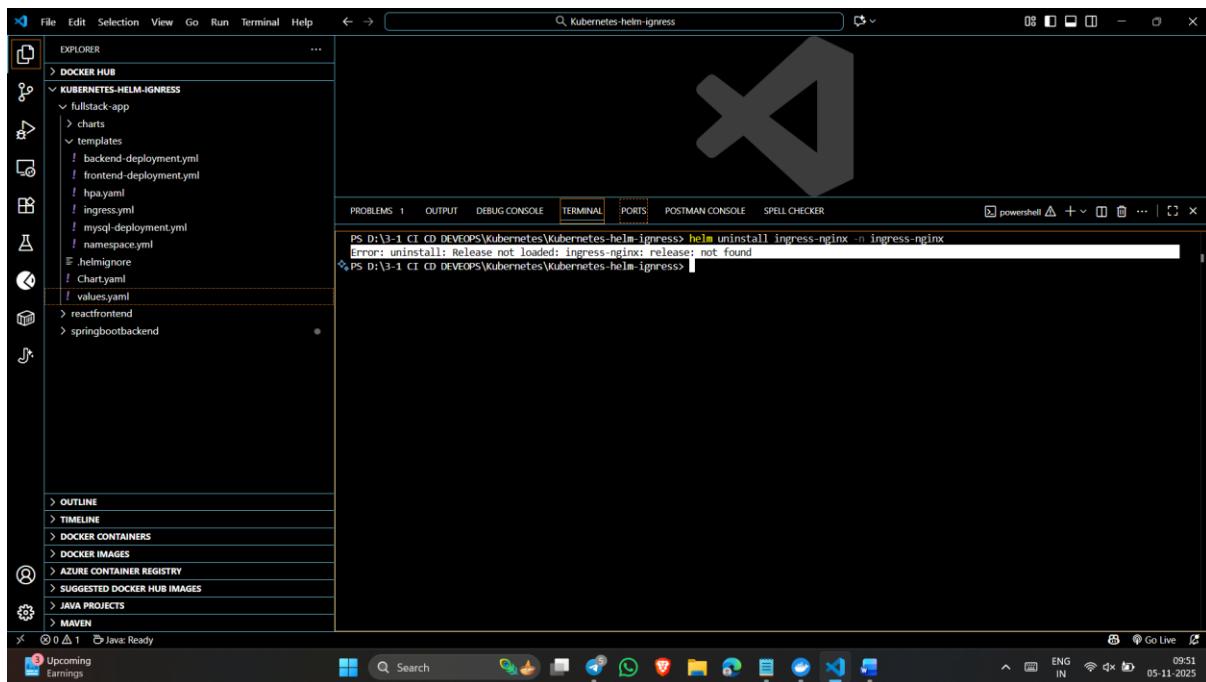
PS D:\3-1 CI CD DEVEOPS\kubernetes\helm-ingress\reactfrontend docker push neeraj2005/k8-frontend:latest
The push refers to repository [docker.io/neeraj2005/k8-frontend]
8a22924f5518: Pushed
621a5197ed7: Layer already exists
76c9bcaa163: Layer already exists
001bb0d02aae: Pushed
2d35ebd57d9: Layer already exists
7fb80cf2f8bc: Layer already exists
e2de0ad5d3690: Layer already exists
03ee3548f209: Layer already exists
f808ab05ead: Layer already exists
83c683d9960: Layer already exists
latest: digest: sha256:c2c224223170088/242d2154cb43e0168e91a5cad8994933105e0f38d77e840 size: 2483
○ PS D:\3-1 CI CD DEVEOPS\kubernetes\helm-ingress\reactfrontend

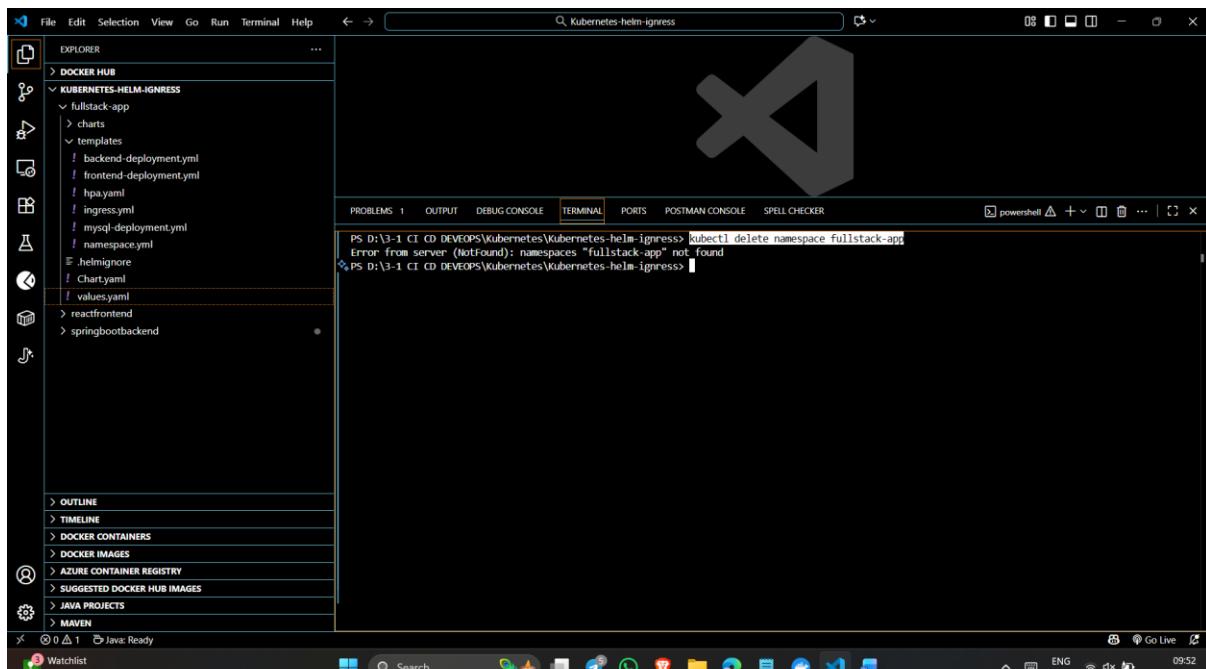
```

Java: Ready

29°C Mostly cloudy







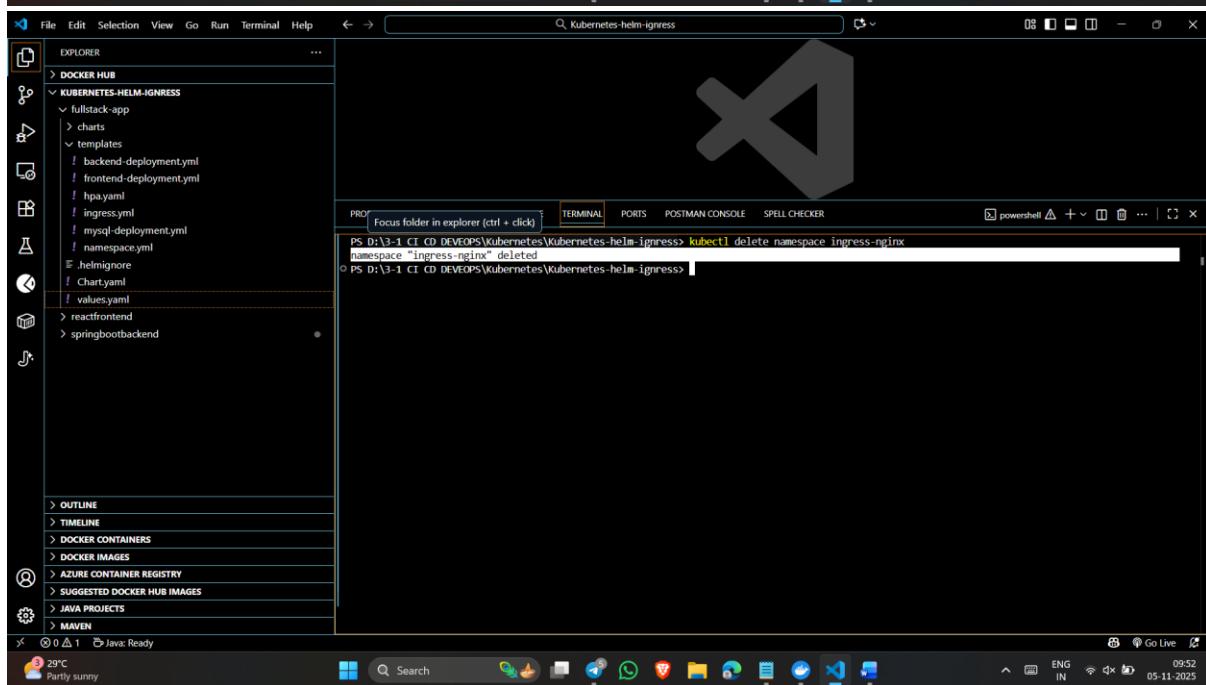
File Edit Selection View Go Run Terminal Help ⏪ ⏩ 🔍 Kubernetes-helm-ingress

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SPELL CHECKER powershell + ... x

```
PS D:\3-1\CI\CD\DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl delete namespace fullstack-app
Error from server: (NotFound): namespaces "fullstack-app" not found
PS D:\3-1\CI\CD\DEVEOPS\Kubernetes\Kubernetes-helm-ingress>
```

Watchlist Ideas



File Edit Selection View Go Run Terminal Help ⏪ ⏩ 🔍 Kubernetes-helm-ingress

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

PROBLEMS Focus folder in explorer (ctrl + click) TERMINAL PORTS POSTMAN CONSOLE SPELL CHECKER powershell + ... x

```
PS D:\3-1\CI\CD\DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl delete namespace ingress-nginx
namespace "ingress-nginx" deleted
PS D:\3-1\CI\CD\DEVEOPS\Kubernetes\Kubernetes-helm-ingress>
```

29°C Partly sunny

The screenshot shows two instances of the Visual Studio Code (VS Code) interface. Both instances have the title bar "Kubernetes-helm-ingress".

Top Instance (Terminal Output):

```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
"ingress-nginx" already exists with the same configuration, skipping
○ PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>
```

Bottom Instance (Terminal Output):

```
PS D:\3-1 CI CD DEVEOPS\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 DEVEOPS\Kubernetes\Kubernetes-helm-ingress>
```

Both instances show the Explorer sidebar on the left, which includes sections for Docker Hub, Kubernetes-Helm-ingress (containing charts, templates, and deployment files like backend-deployment.yaml, frontend-deployment.yaml, hpa.yaml, ingress.yaml, mysql-deployment.yaml, namespace.yaml, values.yaml, reactfrontend, and springbootbackend), and various DevOps tools like Timeline, Docker Containers, Docker Images, Azure Container Registry, Suggested Docker Hub Images, Java Projects, and Maven.

The bottom status bar of both instances displays the weather as "Partly sunny" at 29°C and the date/time as "05-11-2025".

The image shows two side-by-side instances of Microsoft Visual Studio Code (VS Code) running on a Windows operating system. Both instances have dark themes and are displaying terminal outputs.

Top Window Terminal Output:

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

Bottom Window Terminal Output:

```
PS D:\3-1 CI CD DEVEOPS\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
```

Both windows also show the system tray at the bottom, indicating a weather of 29°C and Partly sunny, along with other system icons.

The screenshot displays two terminal windows within a Visual Studio Code (VS Code) interface on a Windows operating system.

Top Terminal Window:

```
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 DEVEOPS\Kubernetes\Kubernetes-helm-ingress>

Bottom Terminal Window:

```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> helm upgrade ingress-nginx ingress-nginx/ingress-nginx --namespace ingress-nginx
level=WARN msg="unable to find exact version; falling back to closest available version" chart=ingress-nginx requested="" selected=4.14.0
Release "ingress-nginx" has been upgraded. Happy Helming!
NAME: ingress-nginx
LAST DEPLOYED: Wed Nov  5 10:04:50 2025
NAMESPACE: ingress-nginx
STATUS: deployed
REVISION: 2
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:
                  number: 80
```

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>

```
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
```

```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl get pods -n ingress-nginx
NAME           READY   STATUS    RESTARTS   AGE
ingress-nginx-controller-668c4fc947-9v8n4   1/1     Running   0          4m48s
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>
```

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> 0 1005 05-11-2025

Docker Desktop Personal

Containers Give feedback

Container CPU usage: 0.28% / 1200% (12 CPUs available)

Container memory usage: 135.4MB / 7.42GB

Search Only show running containers

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
k8s_controller-ingress-n f3008f651cbb	e41270b5d031			0.17%	5 minutes ago	

Showing 1 item

Engine running | Kubernetes running RAM 6.14 GB CPU 5.67% Disk: 19.53 GB used (limit 1006.85 GB) Terminal New version available

29°C Partly sunny

File Edit Selection View Go Run ... File Edit Selection View Go Run ... File Edit Selection View Go Run ...

Kubernetes-helm-ingress

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SPELL CHECKER kubectl + -

```

ingress-nginx-controller-668c4fc947-9v8n4 1/1 Running 0 4m48s
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl logs -f ingress-nginx-controller-668c4fc947-9v8n4 -n ingress-nginx
-----
```

NGINX Ingress controller

```

Release: v1.14.0
Build: 52c0a83ac9bc72e9ce1b9fe4f2d6dcc8854516a8
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 04:31:07.344651    7 main.go:205] "Creating API client" host="https://10.96.0.1:443"
I1105 04:31:07.351515    7 main.go:248] "Running in Kubernetes cluster" major="1" minor="32" git="v1.32.2" state="clean" commit="67a30c0adcf52bd3f56ff0893ce19966b
e12991f" platform="linux/amd64"
I1105 04:31:07.434728    7 main.go:101] "SSL fake certificate created" file="/etc/ingress-controller/ssl/default-fake-certificate.pem"
I1105 04:31:07.446287    7 ssl.go:535] "Loading tls certificate" path="/usr/local/certificates/cert" key="/usr/local/certificates/key"
I1105 04:31:07.453894    7 nginx.go:273] "Starting NGINX Ingress controller"
I1105 04:31:07.461396    7 event.go:377] Event{v1.ObjectReference{Kind:"ConfigMap", Namespace:"ingress-nginx", Name:"ingress-nginx-controller", UID:"c252536e-e99a
-4413-9ef2-c102fd4c8f5", APIVersion:"v1", ResourceVersion:"158868", FieldPath:""}: type: 'Normal' reason: 'CREATE' ConfigMap ingress-nginx/ingress-nginx-controller
I1105 04:31:09.570720    7 nginx.go:319] "Starting NGINX process"
I1105 04:31:09.571032    7 leadelection.go:257] attempting to acquire leader lease ingress-nginx/ingress-nginx-leader...
I1105 04:31:09.571236    7 nginx.go:339] "Starting validation webhook" address=:8443 certPath=/usr/local/certificates/cert keyPath=/usr/local/certificates/ke
y"
I1105 04:31:09.571754    7 controller.go:214] "Configuration changes detected, backend reload required"
I1105 04:31:09.580643    7 leadelection.go:271] successfully acquired lease ingress-nginx/ingress-nginx-leader
I1105 04:31:09.580735    7 status.go:85] "New leader elected" identity="ingress-nginx-controller-668c4fc947-9v8n4"
I1105 04:31:09.598452    7 controller.go:228] "Backend successfully reloaded"
I1105 04:31:09.598607    7 controller.go:240] "Initial sync, sleeping for 1 second"
I1105 04:31:09.598679    7 event.go:377] Event{v1.ObjectReference{Kind:"Pod", Namespace:"ingress-nginx", Name:"ingress-nginx-controller-668c4fc947-9v8n4", UID:"f5
d18f42-61ad-4a20-acbc-99723edae75f", APIVersion:"v1", ResourceVersion:"158895", FieldPath:""}: type: 'Normal' reason: 'RELOAD' NGINX reload triggered due to a chang
e in configuration
```

Java: Ready

29°C Partly sunny

Search

ENG IN 1007 05-11-2025

```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> helm install fullstack-app ./fullstack-app --create-namespace --namespace fullstack-app
NAME: fullstack-app
LAST DEPLOYED: Wed Nov  5 10:08:38 2025
NAMESPACE: fullstack-app
STATUS: deployed
REVISION: 1
DESCRIPTION: Install complete
TEST SUITE: None
○ PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>
```



```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> helm install fullstack-app ./fullstack-app --create-namespace --namespace fullstack-app
NAME: fullstack-app
LAST DEPLOYED: Wed Nov  5 10:08:38 2025
NAMESPACE: fullstack-app
STATUS: deployed
REVISION: 1
DESCRIPTION: Install complete
TEST SUITE: None
● PS D:\3-1 CI CD DEVEOPS\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 DEVEOPS\Kubernetes\Kubernetes-helm-ingress>
```

File Edit Selection View Go Run ... ← → 🔍 Kubernetes-helm-ingress

TERMINAL PROBLEMS 1 OUTPUT DEBUG CONSOLE PORTS POSTMAN CONSOLE SPELL CHECKER

```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl get pods -n fullstack-app
NAME           READY   STATUS    RESTARTS   AGE
backend-deployment-69d98c7c7b-6ddfk  1/1    Running   0          2m18s
backend-deployment-69d98c7c7b-sfvzg  1/1    Running   0          2m18s
frontend-deployment-5587966678-4lqn  1/1    Running   0          2m18s
frontend-deployment-5587966678-wnbjv 1/1    Running   0          2m18s
mysql-deployment-78c4d56997-w269k   1/1    Running   0          2m18s
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>
```

Java: Ready

29°C Partly sunny

docker:desktop PERSONAL

Containers Give feedback ⓘ

Container CPU usage ⓘ 1.14% / 1200% (12 CPUs available)

Container memory usage ⓘ 1.02GB / 7.42GB

Show charts

	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
1	k8s_controller_ingress-n	f3008f651cbb	e4127065d031		0.09%	11 minutes ago	⋮ ⚡ 📈 🗑️
2	k8s_mysql_mysql-deploy	b9bdec2fad7a	[37951fc3753]		0.73%	4 minutes ago	⋮ ⚡ 📈 🗑️
3	k8s_backend_backend-d	afb63e8154bd	neeraj2005/k8-backend		0.17%	3 minutes ago	⋮ ⚡ 📈 🗑️
4	k8s_backend_backend-d	59f6f486b906	neeraj2005/k8-backend		0.15%	3 minutes ago	⋮ ⚡ 📈 🗑️
5	k8s_frontend_frontend-d	9539c91d542d	neeraj2005/k8-frontend		0%	3 minutes ago	⋮ ⚡ 📈 🗑️
6	k8s_frontend_frontend-d	d8b5b6617a0b	neeraj2005/k8-frontend		0%	3 minutes ago	⋮ ⚡ 📈 🗑️

Showing 6 items

Engine running || | Kubernetes running RAM 7.22 GB CPU 7.04% Disk: 19.73 GB used (limit 1006.85 GB)

Terminal New version available

29°C Partly sunny

The screenshot shows two terminal windows side-by-side, both titled "Kubernetes-helm-ingress".

Terminal 1 (Left):

```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl logs -f backend-deployment-69d98c7c7b-6ddfk -n fullstack-app
:: Spring Boot ::          (v3.5.6)

2025-11-05T04:38:50.533Z INFO 1 --- [springbootbackend] [           main] c.klef.dev.SpringbootbackendApplication : Starting SpringbootbackendApplication v0.0.1-SNAPSHOT using Java 21.0.8 with PID 1 (/app/app.jar started by root in /app)
2025-11-05T04:38:50.605Z INFO 1 --- [springbootbackend] [           main] c.klef.dev.SpringbootbackendApplication : No active profile set, falling back to 1 default: "default"
2025-11-05T04:38:55.192Z INFO 1 --- [springbootbackend] [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2025-11-05T04:38:55.401Z INFO 1 --- [springbootbackend] [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 198 ms. Found 1 JPA repository interface.
2025-11-05T04:38:57.912Z INFO 1 --- [springbootbackend] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 2000 (http)
2025-11-05T04:38:57.980Z INFO 1 --- [springbootbackend] [           main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-11-05T04:38:57.980Z INFO 1 --- [springbootbackend] [           main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.46]
2025-11-05T04:38:58.175Z INFO 1 --- [springbootbackend] [           main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicationContext
2025-11-05T04:38:58.680Z INFO 1 --- [springbootbackend] [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 7853 ms
2025-11-05T04:39:01.392Z INFO 1 --- [springbootbackend] [           main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name : default]
2025-11-05T04:39:01.983Z INFO 1 --- [springbootbackend] [           main] org.hibernate.Version : HHH000412: Hibernate ORM core version 6.6.29.Final
2025-11-05T04:39:02.390Z INFO 1 --- [springbootbackend] [           main] o.h.c.internal.RegionFactoryInitiator : HHH00026: Second-level cache disabled
```

Terminal 2 (Right):

```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl logs -f backend-deployment-69d98c7c7b-6ddfk -n fullstack-app
at org.springframework.boot.SpringApplication.run(SpringApplication.java:1361) ~[spring-boot-3.5.6.jar!/3.5.6]
at org.springframework.boot.SpringApplication.run(SpringApplication.java:1350) ~[spring-boot-3.5.6.jar!/3.5.6]
at com.klef.dev.SpringbootbackendApplication.main(SpringbootbackendApplication.java:11) ~[!/0.0.1-SNAPSHOT]
at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:103) ~[na:na]
at java.base/java.lang.reflect.Method.invoke(Method.java:588) ~[na:na]
at org.springframework.boot.loader.launch.Launcher.launch(Launcher.java:102) ~[app.jar:0.0.1-SNAPSHOT]
at org.springframework.boot.loader.launch.Launcher.launch(Launcher.java:64) ~[app.jar:0.0.1-SNAPSHOT]
at org.springframework.boot.loader.JarLauncher.main(JarLauncher.java:40) ~[app.jar:0.0.1-SNAPSHOT]
Caused by: java.sql.SQLSyntaxErrorException: Table 'task_table' already exists
at com.mysql.cj.jdbc.exceptions.SQLExceptions.createSQLException(SQLException.java:112) ~[mysql-connector-j-9.4.0.jar!/9.4.0]
at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:114) ~[mysql-connector-j-9.4.0.jar!/9.4.0]
at com.mysql.cj.jdbc.StatementImpl.executeInternal(StatementImpl.java:837) ~[mysql-connector-j-9.4.0.jar!/9.4.0]
at com.mysql.cj.jdbc.StatementImpl.execute(StatementImpl.java:685) ~[mysql-connector-j-9.4.0.jar!/9.4.0]
at com.zaxxer.hikari.pool.ProxyStatement.execute(ProxyStatement.java:95) ~[HikariCP-6.3.3.jar!/na]
at com.zaxxer.hikari.pool.HikariProxyStatement.execute(HikariProxyStatement.java) ~[HikariCP-6.3.3.jar!/na]
at org.hibernate.tool.schema.internal.exec.GenerationTargetToDatabase.accept(GenerationTargetToDatabase.java:80) ~[hibernate-core-6.6.29.Final.jar!/6.6.29.Final]
... 42 common frames omitted

2025-11-05T04:39:10.595Z INFO 1 --- [springbootbackend] [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2025-11-05T04:39:12.095Z WARN 1 --- [springbootbackend] [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2025-11-05T04:39:12.296Z INFO 1 --- [springbootbackend] [           main] o.s.v.b.OptionalValidatorFactoryBean : Failed to set up a Bean Validation provider: jakarta.validation.NoProviderFoundException: Unable to create a Configuration, because no Jakarta Bean Validation provider could be found. Add a provider like Hibernate Validator (RI) to your classpath.
2025-11-05T04:39:14.093Z INFO 1 --- [springbootbackend] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 2000 (http) with context path '/'
2025-11-05T04:39:14.194Z INFO 1 --- [springbootbackend] [           main] c.klef.dev.SpringbootbackendApplication : Started SpringbootbackendApplication in 27.369 seconds (process running for 38.574)
Project Backend is Running Successfully ...
```

The image shows three vertically stacked terminal windows from a development environment, likely Visual Studio Code, running on a Windows operating system.

Top Terminal:

```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> helm list -n fullstack-app
NAME      NAMESPACE   REVISION  UPDATED             STATUS      CHART
fullstack-app  fullstack-app  2        2025-11-05 10:09:44.5437061 +0530 IST  deployed  fullstack-app-1.0.0
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>
```

Middle Terminal:

```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> helm list -n fullstack-app
NAME      NAMESPACE   REVISION  UPDATED             STATUS      CHART
fullstack-app  fullstack-app  2        2025-11-05 10:09:44.5437061 +0530 IST  deployed  fullstack-app-1.0.0
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl get svc -n fullstack-app
NAME           TYPE      CLUSTER-IP    EXTERNAL-IP   PORT(S)          AGE
backend-deployment-service  NodePort  10.99.183.110 <none>       2000:30025/TCP  9m16s
frontend-deployment-service  NodePort  10.103.242.133 <none>       80:30080/TCP   9m16s
mysql-service   ClusterIP  10.99.3.3    <none>       3306/TCP       9m16s
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>
```

Bottom Terminal:

```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl get svc -n fullstack-app
NAME           TYPE      CLUSTER-IP    EXTERNAL-IP   PORT(S)          AGE
backend-deployment-service  NodePort  10.99.183.110 <none>       2000:30025/TCP  9m16s
frontend-deployment-service  NodePort  10.103.242.133 <none>       80:30080/TCP   9m16s
mysql-service   ClusterIP  10.99.3.3    <none>       3306/TCP       9m16s
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>
```

The terminal windows show the execution of Helm commands to list applications and Kubernetes commands to inspect services. The environment includes a Java application ready, news notifications, and system status indicators at the bottom.

The screenshot shows a Windows desktop environment with a terminal window open in Visual Studio Code. The terminal is running a PowerShell session titled "Kubernetes-helm-ingress". The command executed was "helm status fullstack-app -n fullstack-app". The output displays the status of the "fullstack-app" deployment across three namespaces: v1/Namespaces, v1/Service, v1/Deployment, v1/Pod, and v2/HorizontalPodAutoscaler.

```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> helm status fullstack-app -n fullstack-app
NAME: fullstack-app
LAST DEPLOYED: Wed Nov  5 10:09:44 2025
NAMESPACE: fullstack-app
STATUS: deployed
REVISION: 2
DESCRIPTION: Upgrade complete
RESOURCES:
==> v1/Namespaces
NAME STATUS AGE
fullstack-app Active 10m

==> v1/Service
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
backend-deployment-service NodePort 10.99.183.110 <none> 2000:30025/TCP 10m
frontend-deployment-service NodePort 10.103.242.133 <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 2/2 2 2 10m
mysql-deployment 1/1 1 1 10m

==> v1/Pod
NAME READY STATUS RESTARTS AGE
backend-deployment-69d98c7c7b-6ddfk 1/1 Running 0 10m
backend-deployment-69d98c7c7b-sfvzg 1/1 Running 0 10m
frontend-deployment-5587966678-4lqnq 1/1 Running 0 10m
frontend-deployment-5587966678-wnbjv 1/1 Running 0 10m
mysql-deployment-78c4d56997-w269k 1/1 Running 0 10m

==> v2/HorizontalPodAutoscaler
```

Kubernetes-helm-ingress

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE SPELL CHECKER powershell

```

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  10.99.183.110  <none>        2000:30025/TCP  10m
frontend-deployment-service  NodePort  10.103.242.133  <none>        80:30088/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  2/2     2           2          10m
mysql-deployment  1/1     1           1          10m

==> v1/Pod(related)
NAME      READY  STATUS  RESTARTS  AGE
backend-deployment-69d98c7c7b-6ddfk  1/1   Running  0          10m
backend-deployment-69d98c7c7b-sfvzg  1/1   Running  0          10m
frontend-deployment-5587966678-4lqnq  1/1   Running  0          10m
frontend-deployment-5587966678-wnbjv  1/1   Running  0          10m
mysql-deployment-78c4d56997-w269k  1/1   Running  0          10m

==> v2/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

```

PAK - SA Game score Java: Ready

File Edit Selection View Go Run ... K Kubernetes-helm-ingress powershell

```

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> helm get manifest fullstack-app -n fullstack-app
...
# Source: fullstack-app/templates/namespace.yaml
apiVersion: v1
kind: Namespace
metadata:
  name: fullstack-app
...
# Source: fullstack-app/templates/backend-deployment.yaml
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.yaml
apiVersion: v1
kind: Service
metadata:
  name: frontend-deployment-service
  namespace: fullstack-app
spec:
  selector:
    app: frontend-deployment

```

Gold +0.66% Java: Ready

File Edit Selection View Go Run ... K Kubernetes-helm-ingress powershell

```
PS D:\3-1\CI\CD\DEVEOPS\Kubernetes\Kubernetes-helm-ingress> helm history fullstack-app -n fullstack-app
REVISION UPDATED STATUS CHART APP VERSION DESCRIPTION
1 Wed Nov 5 10:08:38 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\CI\CD\DEVEOPS\Kubernetes\Kubernetes-helm-ingress>
```



```
PS D:\3-1\CI\CD\DEVEOPS\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\CI\CD\DEVEOPS\Kubernetes\Kubernetes-helm-ingress>
```

Horizontal pod autoscaling

```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl describe hpa backend -n fullstack-app
Name:          backend-deployment-hpa
Namespace:     fullstack-app
meta.helm.sh/release-namespace: fullstack-app
CreationTimestamp: Wed, 05 Nov 2025 10:08:38 +0530
Reference:    Deployment/backend-deployment
              ( current / target )
Metrics:      resource cpu on pods  (as a percentage of request): <unknown> / 50%
Min replicas: 2
Max replicas: 5
Deployment pods: 2 current / 0 desired
Conditions:
  Type      Status  Reason
  ----      ----  -----
  AbleToScale  True   SucceededGetScale  the HPA controller was able to get the target's current scale
  ScalingActive False  FailedGetResourceMetric  the HPA was unable to compute the replica count: failed to get cpu utilization: unable to get metrics for resource
cpu: unable to fetch metrics from resource metrics API: the server could not find the requested resource (get pods.metrics.k8s.io)
Events:
  Type      Reason           Age   From            Message
  ----      ----           --   --            --
  Warning  FailedComputeMetricsReplicas  3m51s (x13 over 16m)  horizontal-pod-autoscaler  invalid metrics (1 invalid out of 1), first error is: failed to get cpu res
ource metric value: failed to get cpu utilization: unable to get metrics for resource cpu: unable to fetch metrics from resource metrics API: the server could not fi
nd the requested resource (get pods.metrics.k8s.io)
  Warning  FailedGetResourceMetric       46s (x16 over 16m)   horizontal-pod-autoscaler  failed to get cpu utilization: unable to get metrics for resource cpu: unab
le to fetch metrics from resource metrics API: the server could not find the requested resource (get pods.metrics.k8s.io)

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl describe hpa frontend -n fullstack-app
Name:          frontend-deployment-hpa
Namespace:     fullstack-app
Labels:        app.kubernetes.io/managed-by=Helm
Annotations:   meta.helm.sh/release-name: fullstack-app
               meta.helm.sh/release-namespace: fullstack-app
CreationTimestamp: Wed, 05 Nov 2025 10:08:38 +0530
Reference:    Deployment/frontend-deployment
              ( current / target )
Metrics:      resource cpu on pods  (as a percentage of request): <unknown> / 50%
Min replicas: 2
Max replicas: 5
Deployment pods: 2 current / 0 desired
Conditions:
  Type      Status  Reason
  ----      ----  -----
  AbleToScale  True   SucceededGetScale  the HPA controller was able to get the target's current scale
  ScalingActive False  FailedGetResourceMetric  the HPA was unable to compute the replica count: failed to get cpu utilization: unable to get metrics for resource
cpu: unable to fetch metrics from resource metrics API: the server could not find the requested resource (get pods.metrics.k8s.io)
Events:
  Type      Reason           Age   From            Message
  ----      ----           --   --            --
  Warning  FailedComputeMetricsReplicas  5m25s (x13 over 17m)  horizontal-pod-autoscaler  invalid metrics (1 invalid out of 1), first error is: failed to get cpu res
ource metric value: failed to get cpu utilization: unable to get metrics for resource cpu: unable to fetch metrics from resource metrics API: the server could not fi
nd the requested resource (get pods.metrics.k8s.io)
  Warning  FailedGetResourceMetric       2m20s (x16 over 17m)   horizontal-pod-autoscaler  failed to get cpu utilization: unable to get metrics for resource cpu: unab
le to fetch metrics from resource metrics API: the server could not find the requested resource (get pods.metrics.k8s.io)

PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>
```

```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl get hpa -n fullstack-app -w
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 DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl get pods -n fullstack-app
NAME                           READY   STATUS    RESTARTS   AGE
backend-deployment-69d98c7cb-6ddf8k  1/1    Running   0          28m
backend-deployment-69d98c7cb-sfvzg  1/1    Running   0          28m
frontend-deployment-5587966678-4lqn8h 1/1    Running   0          28m
frontend-deployment-5587966678-wnbjv  1/1    Running   0          28m
mysql-deployment-78c4d56997-w269k   1/1    Running   0          20m
```

The image shows two nearly identical screenshots of a terminal window within a development environment. The terminal is titled 'Kubernetes-helm-ingress' and is running on a Windows operating system, as indicated by the taskbar icons at the bottom.

The terminal displays the following command-line session:

```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl get hpa -n fullstack-app -w
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

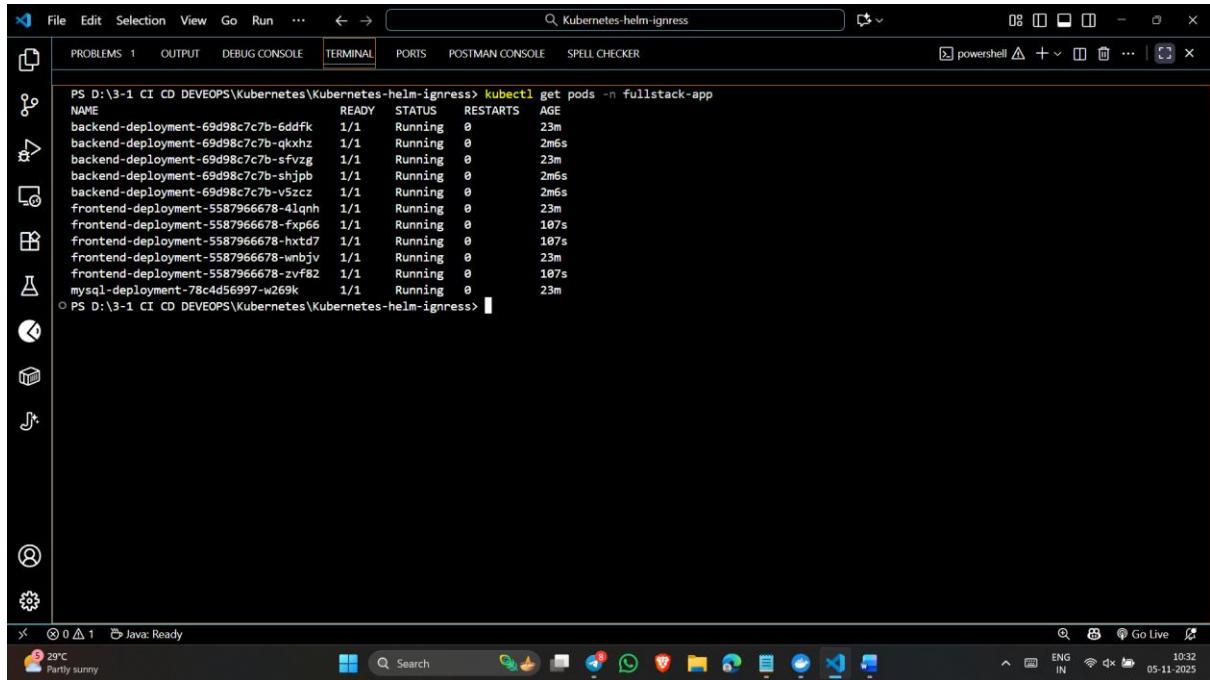
● PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl get pods -n fullstack-app
NAME           READY   STATUS    RESTARTS   AGE
backend-deployment-69d98c7c7b-6ddf8k  1/1    Running   0          28m
backend-deployment-69d98c7c7b-sfvzg   1/1    Running   0          28m
frontend-deployment-5587966678-4lqn8h 1/1    Running   0          28m
frontend-deployment-5587966678-wnbjv   1/1    Running   0          28m
mysql-deployment-78c4d56997-w269k     1/1    Running   0          28m

● PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl scale deployment backend-deployment --replicas=5 -n fullstack-app
deployment.apps/backend-deployment scaled

○ PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>
```

The terminal window has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, PORTS, POSTMAN CONSOLE, and SPELL CHECKER. The DEBUG CONSOLE tab is currently selected. The title bar also includes a search field for 'Kubernetes-helm-ingress' and a powershell icon.

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



The screenshot shows a terminal window titled "Kubernetes-helm-ingress" in a code editor interface. The terminal tab is active, displaying the command "kubectl get pods -n fullstack-app". The output lists several pods with their names, readiness, status, restarts, and age. All pods are shown as "Running" with 0 restarts and less than 3 minutes of age. The terminal window has a dark theme with light-colored text. The status bar at the bottom shows "Java: Ready" and system icons like weather and battery level.

```
PS D:\3-1 CI CD DEVEOPS\Kubernetes-helm-ingress> kubectl get pods -n fullstack-app
NAME           READY   STATUS    RESTARTS   AGE
backend-deployment-69d98c7c7b-6ddf4k   1/1     Running   0          23m
backend-deployment-69d98c7c7b-qkxhz   1/1     Running   0          2m6s
backend-deployment-69d98c7c7b-sfvzg   1/1     Running   0          23m
backend-deployment-69d98c7c7b-shjpb   1/1     Running   0          2m6s
backend-deployment-69d98c7c7b-v5zcz   1/1     Running   0          2m6s
frontend-deployment-5587966678-4lqnq  1/1     Running   0          23m
frontend-deployment-5587966678-fxp66  1/1     Running   0          107s
frontend-deployment-5587966678-hxtd7  1/1     Running   0          107s
frontend-deployment-5587966678-wnbjv  1/1     Running   0          23m
frontend-deployment-5587966678-zvf82  1/1     Running   0          107s
mysql-deployment-78c4d56997-w269k    1/1     Running   0          23m
PS D:\3-1 CI CD DEVEOPS\Kubernetes-helm-ingress>
```

Screenshot of a Task Manager application running on localhost:30080.

The application has a dark-themed header with "Task Manager" and navigation links for "Dashboard" and "Task Board".

The main interface shows an "Add Task" form with fields for Title, Description, Start Date, End Date, and Priority, along with a search bar and an "Add Task" button.

The "All Tasks" table displays the following data:

ID	TITLE	DESCRIPTION	START	END	PRIORITY	STATUS	ACTIONS
							No matching tasks found

After adding a task, a success message "Task added successfully!" is displayed, and the table shows two entries:

ID	TITLE	DESCRIPTION	START	END	PRIORITY	STATUS	ACTIONS
5932	Ingress LAB	LAB 14	2025-11-05	2025-11-07	HIGH	ASSIGNED	<button>Edit</button> <button>Delete</button>
75113	demo	demo	2025-11-05	2025-11-20	MEDIUM	ASSIGNED	<button>Edit</button> <button>Delete</button>

React App

localhost:30080/board

Knowledge TryHackMe Sakala Neeraj - Acti... Live DSA Course | L... Interview specific Q... (4) Discord #java-p...

All Bookmarks

Task Board

ASSIGNED	PROGRESS	COMPLETED	REJECTED
<p>Ingress LAB</p> <p>LAB 14</p> <p>Priority: HIGH</p> <p>Start: 2025-11-05</p> <p>End: 2025-11-07</p> <p>Start</p> <p>demo</p> <p>demo</p> <p>Priority: MEDIUM</p> <p>Start: 2025-11-05</p> <p>End: 2025-11-20</p> <p>Start</p>			

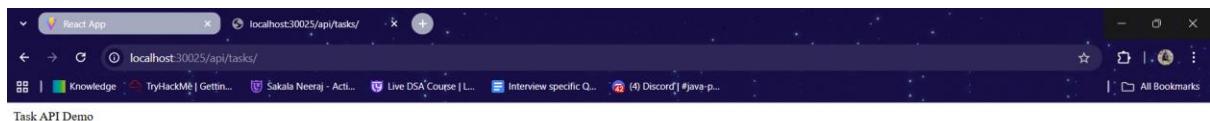
29°C Partly sunny

Search

10:39

ENG IN

05-11-2025



```
29°C Partly sunny
```

```
React App localhost:30025/api/tasks/ localhost:30025/api/tasks/all
```

```
localhost:30025/api/tasks/all
```

```
Pretty print 
```

```
[  
  {  
    "id": 592,  
    "title": "Ingress LAB",  
    "description": "LAB 1A",  
    "startDate": "2025-11-05",  
    "endDate": "2025-11-07",  
    "priority": "HIGH",  
    "status": "ASSIGNED"  
  },  
  {  
    "id": 75113,  
    "title": "demo",  
    "description": "demo",  
    "startDate": "2025-11-05",  
    "endDate": "2025-11-20",  
    "priority": "MEDIUM",  
    "status": "ASSIGNED"  
  }]
```

```
29°C Partly sunny
```



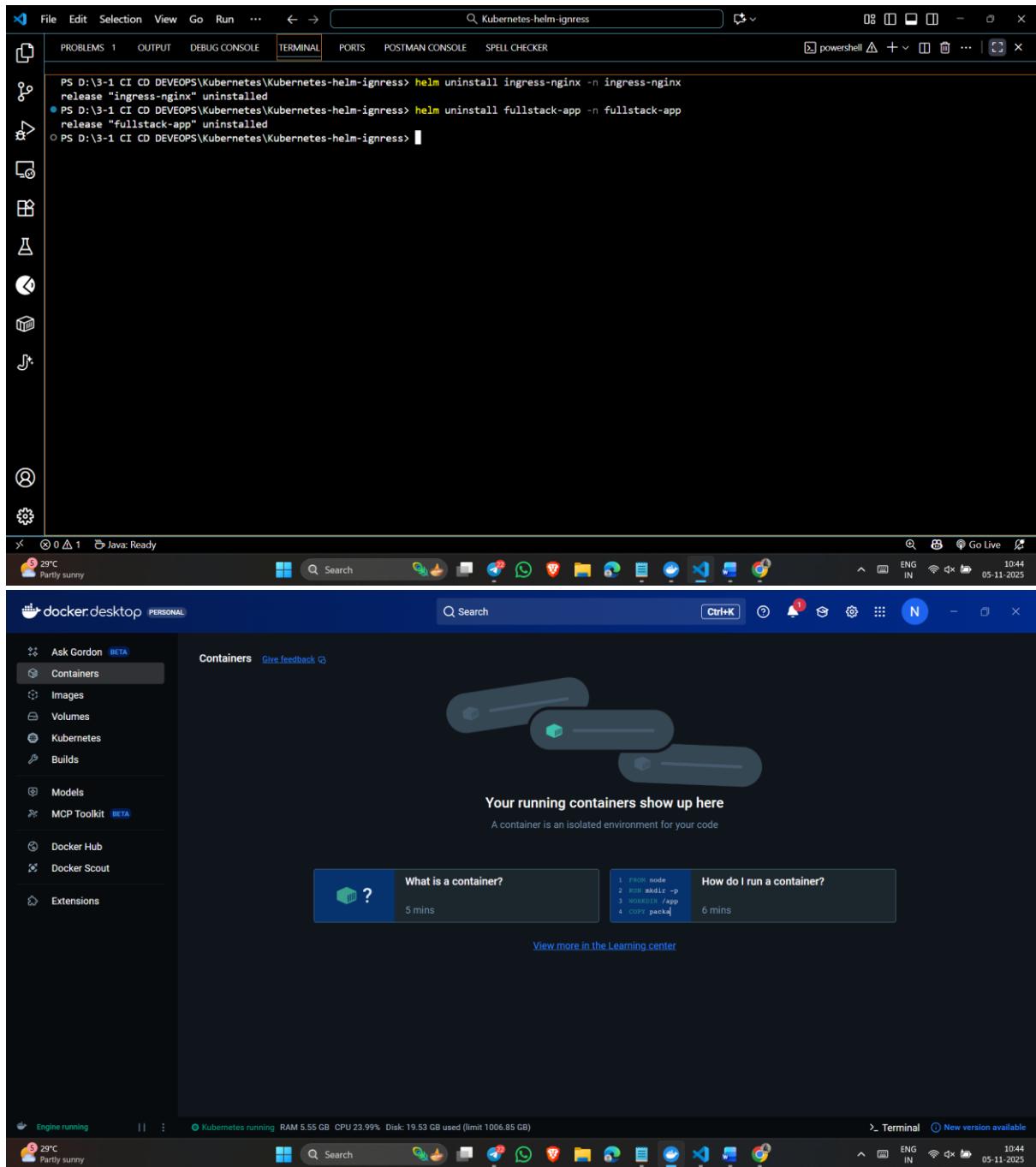
The screenshot shows a Windows desktop environment with a dark theme. At the top, there is a taskbar with several pinned icons and a search bar. Below the taskbar, a browser window is open with five tabs, each showing a different URL related to a local API endpoint. The tabs are titled: "React App", "localhost:30025/api/tasks/", "localhost:30025/api/tasks/all", "Swagger UI", and "localhost:30025/api/tasks/get/5932". The content of the "localhost:30025/api/tasks/get/5932" tab is a JSON response:

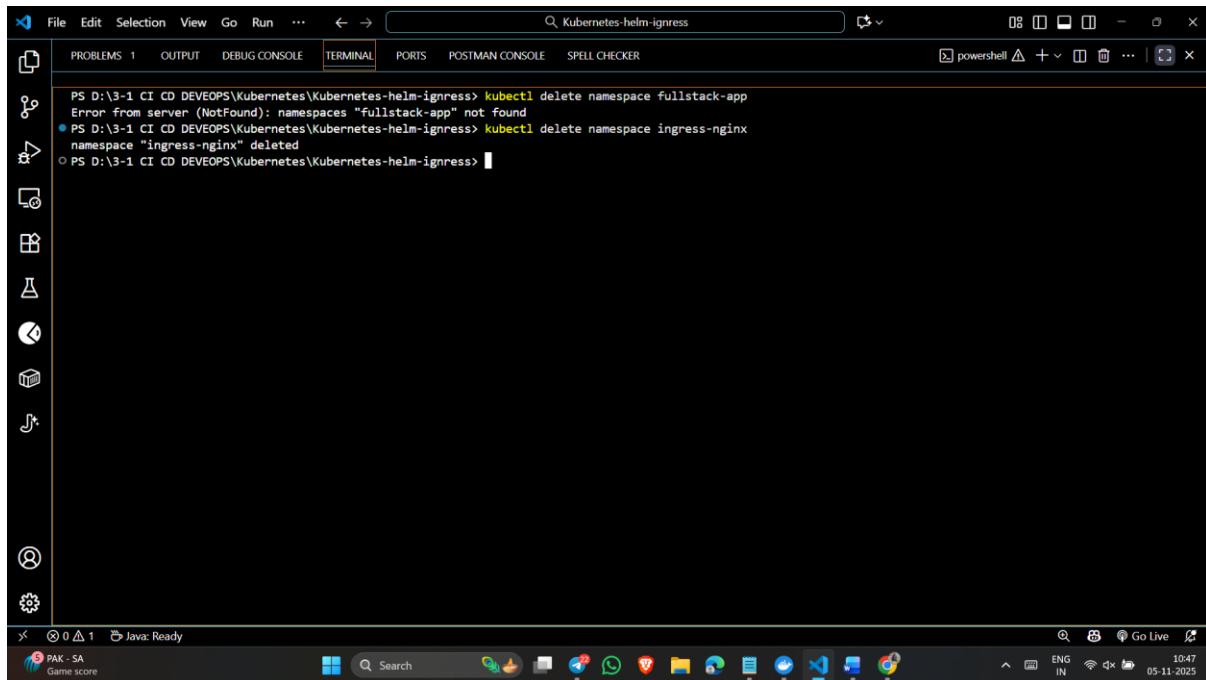
```
{  
    "id": 5932,  
    "title": "Ingress LAB",  
    "description": "LAB 14",  
    "startDate": "2025-11-05",  
    "endDate": "2025-11-07",  
    "priority": "HIGH",  
    "status": "ASSIGNED"  
}
```

Below the browser is a large window for the Visual Studio Code (VS Code) editor. The title bar of the VS Code window says "Kubernetes-helm-ingress". The main area of the editor shows a terminal window with the following command history:

```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> helm uninstall ingress-nginx -n ingress-nginx  
release "ingress-nginx" uninstalled  
○ PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> █
```

The VS Code interface includes a sidebar with various icons, a top navigation bar with tabs like PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, and a bottom status bar indicating "Java: Ready".





```
PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl delete namespace fullstack-app
Error from server (NotFound): namespaces "fullstack-app" not found
● PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress> kubectl delete namespace ingress-nginx
namespace "ingress-nginx" deleted
○ PS D:\3-1 CI CD DEVEOPS\Kubernetes\Kubernetes-helm-ingress>
```

■ 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
```
