TASK 3,4

Step 1: Clone the Repository

1. Clone your Git repository and navigate to the project folder:

git clone <your-repo-url>

cd <your-repo-folder>

Step 2: Build the Docker Image

2. Create a Docker image for your login page application:

docker build -t login-page-app .

Step 3: Start Minikube

3. Launch Minikube and check its status:

minikube start --force

minikube status

Step 4: Load the Docker Image into Minikube

4. Transfer the built Docker image to Minikube:

minikube image load login-page-app

5. Verify that the image is available:

minikube image list # Ensure "login-page-app" appears in the list

Step 5: Deploy the Application

6. Apply the Kubernetes deployment configuration:

kubectl apply -f deployment.yml

kubectl get deployments

kubectl get pods

7. If a NodePort service is required, deploy it:

kubectl apply -f Nodeport.yaml

Step 6: Resolve Image Pull Issues (If Needed)

8. If Kubernetes tries to pull the image from an external registry instead of using the local one, update the deployment:

kubectl patch deployment login-page-deployment --type='json' -p='[{"op": "replace", "path": "/spec/template/spec/containers/0/imagePullPolicy", "value": "Never"}]'

Step 7: Expose the Service & Access the Application

9. Retrieve the Minikube IP and expose the service:

minikube ip

minikube service login-page-service

Step 8: Push to GitHub

10. Initialize and push your changes to GitHub:

git init # If not already initialized

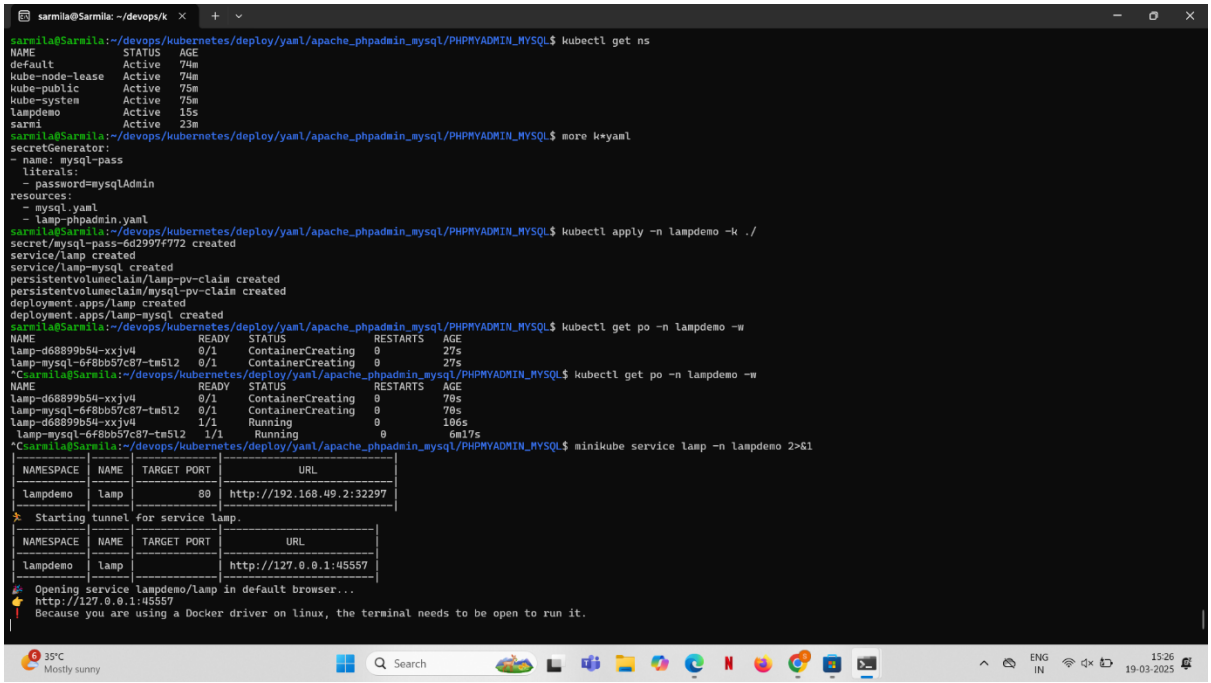
git add Dockerfile deployment.yml Nodeport.yaml

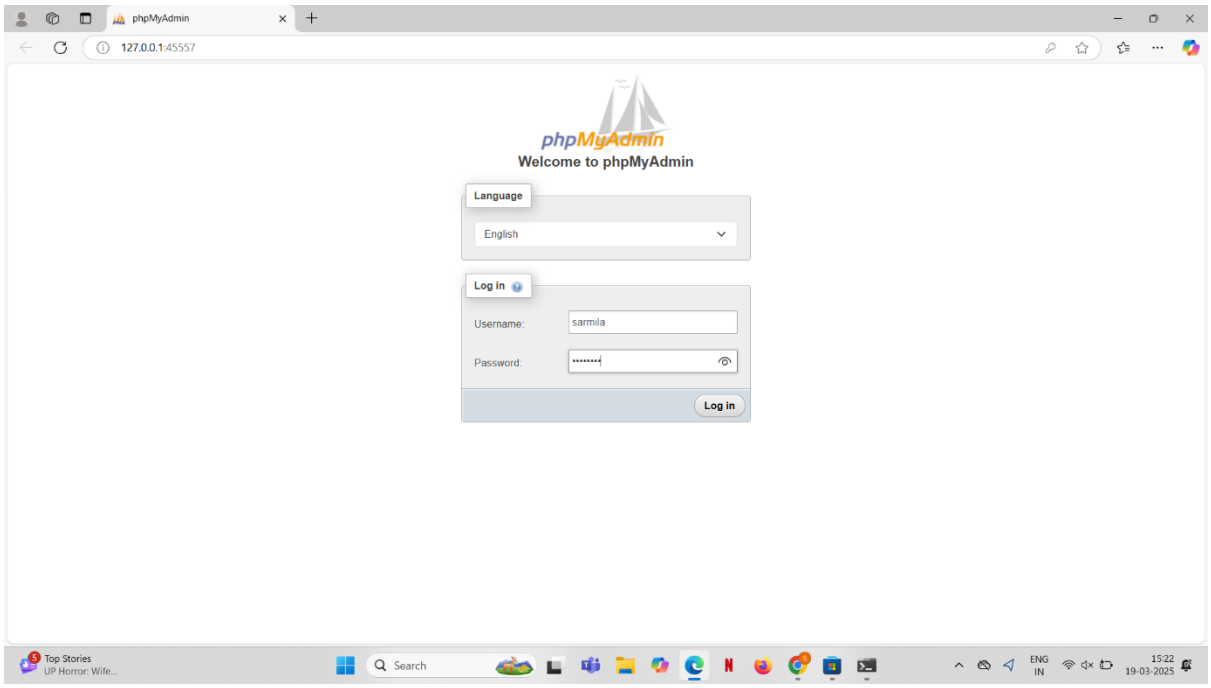
git commit -m "Kubernetes deployment for login page"

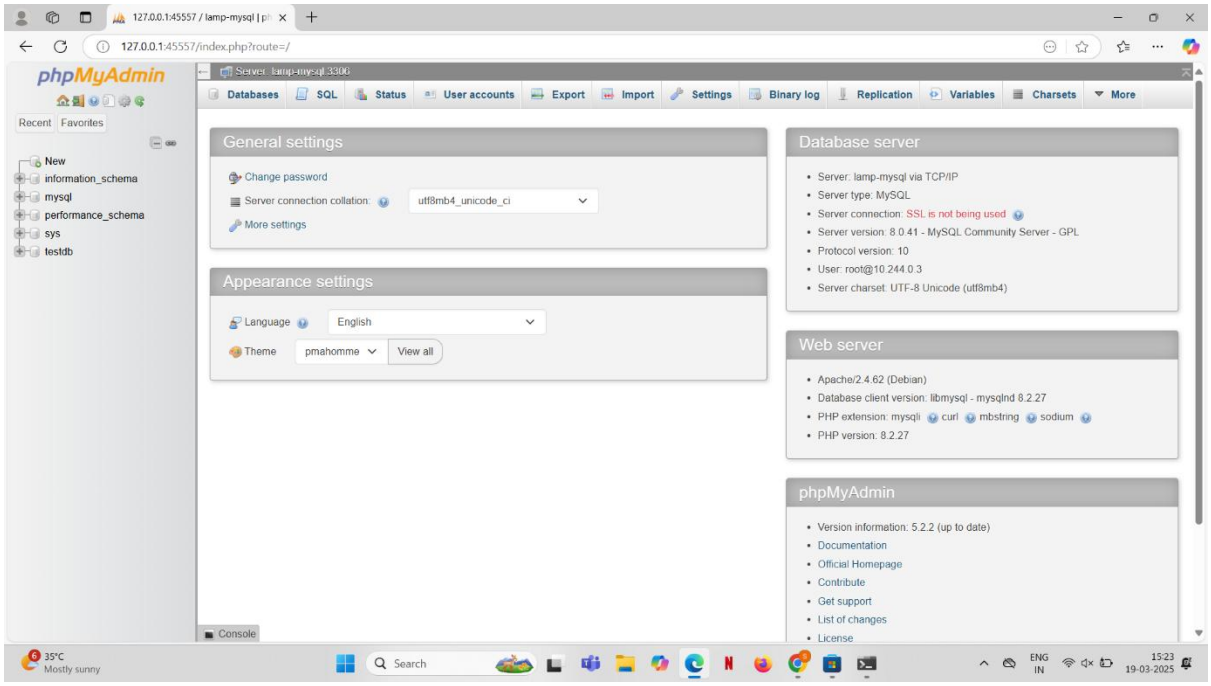
git remote add origin <your-repo-url>

git branch -M main

git push -u origin main


sarmila@Sarmila: ~/devops/k
sarmila@Sarmila:~/devops/kubernetes/deploy/yaml/apache_phpadmin_mysql/PHPMYADMIN_MYSQL$ kubectl get ns
NAME STATUS AGE
default Active 74m
kube-node-lease Active 74m
kube-public Active 75m
kube-system Active 75m
lampdemo Active 15s
sarmi Active 23m
sarmila@Sarmila:~/devops/kubernetes/deploy/yaml/apache_phpadmin_mysql/PHPMYADMIN_MYSQL$ more k*yaml
secretGenerator:
- name: mysql-pass
literals:
- password=mysqlAdmin
resources:
- mysql.yaml
- lamp-phpadmin.yaml
sarmila@Sarmila:~/devops/kubernetes/deploy/yaml/apache_phpadmin_mysql/PHPMYADMIN_MYSQL$ kubectl apply -n lampdemo -k ./
secret/mysql-pass-6d2997f772 created
service/lamp created
service/lamp-mysql created
persistentvolumeclaim/lamp-pv-claim created
persistentvolumeclaim/mysql-pv-claim created
deployment.apps/lamp created
deployment.apps/lamp-mysql created
sarmila@Sarmila:~/devops/kubernetes/deploy/yaml/apache_phpadmin_mysql/PHPMYADMIN_MYSQL$ kubectl get po -n lampdemo -w
NAME READY STATUS RESTARTS AGE
lamp-d68899b54-xxjv4 0/1 ContainerCreating 0 27s
lamp-mysql-6f8bb57c87-tm5l2 0/1 ContainerCreating 0 27s
^Csarmila@Sarmila:~/devops/kubernetes/deploy/yaml/apache_phpadmin_mysql/PHPMYADMIN_MYSQL$ kubectl get po -n lampdemo -w
NAME READY STATUS RESTARTS AGE
lamp-d68899b54-xxjv4 0/1 ContainerCreating 0 70s
lamp-mysql-6f8bb57c87-tm5l2 0/1 ContainerCreating 0 70s
lamp-d68899b54-xxjv4 1/1 Running 0 106s
lamp-mysql-6f8bb57c87-tm5l2 1/1 Running 0 6m17s
^Csarmila@Sarmila:~/devops/kubernetes/deploy/yaml/apache_phpadmin_mysql/PHPMYADMIN_MYSQL$ minikube service lamp -n lampdemo 2>&1
NAMESPACE NAME TARGET PORT URL
lampdemo lamp 80 http://192.168.49.2:32297
Starting tunnel for service lamp.
NAMESPACE NAME TARGET PORT URL
lampdemo lamp http://127.0.0.1:45557
Opening service lampdemo/lamp in default browser...
http://127.0.0.1:45557
Because you are using a Docker driver on linux, the terminal needs to be open to run it.
35°C
Mostly sunny
Search
ENG
IN
15:26
19-03-2025


phpMyAdmin
127.0.0.1:45557
phpMyAdmin
Welcome to phpMyAdmin
Language
English
Log in
Username:
sarmila
Password:
Log in
Top Stories
UP Horror: Wife...
Search
ENG
IN
15:22
19-03-2025


127.0.0.1:45557 / lamp-mysql | ph
127.0.0.1:45557/index.php?route=/
phpMyAdmin
Recent
Favorites
New
information_schema
mysql
performance_schema
sys
testdb
Server: lamp-mysql:3306
Databases
SQL
Status
User accounts
Export
Import
Settings
Binary log
Replication
Variables
Charsets
More
General settings
Change password
Server connection collation:
utf8mb4_unicode_ci
More settings
Appearance settings
Language
English
Theme
pmahomme
View all
Database server
Server: lamp-mysql via TCP/IP
Server type: MySQL
Server connection: SSL is not being used
Server version: 8.0.41 - MySQL Community Server - GPL
Protocol version: 10
User: root@10.244.0.3
Server charset: UTF-8 Unicode (utf8mb4)
Web server
Apache/2.4.62 (Debian)
Database client version: libmysql - mysqlnd 8.2.27
PHP extension: mysqli curl mbstring sodium
PHP version: 8.2.27
phpMyAdmin
Version information: 5.2.2 (up to date)
Documentation
Official Homepage
Contribute
Get support
List of changes
License
Console
35°C
Mostly sunny
Search
ENG
IN
15:23
19-03-2025