

Container Orchestration Assignment

1. Install minikube/Docker on AWS ubuntu machine:

> Create EC2 instance:

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, and Kv Pairs. The main content area has a search bar at the top with filters for 'Name' (set to 'aks') and 'All states' (set to 'Running'). A single instance is listed: 'i-0f3715ec1cd9cd070 (Aks-MERN_container_orch)'. The instance details show it's 'Running' with a Public IP of '35.173.188.89'. Below the instance table, there's a detailed view for 'i-0f3715ec1cd9cd070 (Aks-MERN_container_orch)' under the 'Details' tab. It includes sections for Instance summary, Networking, and Storage.

Login using pem key file:

ssh -i "EC2-AMI-Aks-HV.pem" ubuntu@35.173.188.89

> Install Docker/kubectl/minikube/conntrack for dependencies:

```
sudo su
apt update
apt -y install docker.io
```

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s
https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/
kubectl && chmod +x ./kubectl && sudo mv ./kubectl /usr/local/bin/kubectl
```

```
curl -Lo minikube  
https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 &&  
chmod +x minikube && sudo mv minikube /usr/local/bin/
```

```
apt install conntrack  
usermod -aG docker $USER && newgrp docker
```

```
minikube start --force  
minikube status
```

```
root@ip-10-0-0-172:/home/ubuntu# minikube start --force  
😄 minikube v1.35.0 on Ubuntu 24.04 (xen/amd64)  
❗ minikube skips various validations when --force is supplied; this may lead to unexpected behavior  
💡 Automatically selected the docker driver. Other choices: ssh, none  
The "docker" driver should not be used with root privileges. If you wish to continue as root, use --force.  
If you are running minikube within a VM, consider using --driver=none:  
  https://minikube.sigs.k8s.io/docs/reference/drivers/none/  
💡 Using Docker driver with root privileges  
Starting "minikube" primary control-plane node in "minikube" cluster  
Pulling base image v0.0.46 ...  
Downloading Kubernetes v1.32.0 preload ...  
  > gcr.io/k8s-minikube/kicbase...: 500.31 MiB / 500.31 MiB 100.00% 77.58 M  
  > preloaded-images-k8s-v18-v1...: 333.57 MiB / 333.57 MiB 100.00% 40.37 M  
🔥 Creating docker container (CPUs=2, Memory=2200MB) ...  
Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...  
  ■ Generating certificates and keys ...  
  ■ Booting up control plane ...  
  ■ Configuring RBAC rules ...  
🔗 Configuring bridge CNI (Container Networking Interface) ...  
🌐 Verifying Kubernetes components...  
  ■ Using image gcr.io/k8s-minikube/storage-provisioner:v5  
⭐ Enabled addons: storage-provisioner, default-storageclass  
🔥 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default  
root@ip-10-0-0-172:/home/ubuntu#  
root@ip-10-0-0-172:/home/ubuntu# minikube status  
minikube  
type: Control Plane  
host: Running  
kubelet: Running  
apiserver: Running  
kubeconfig: Configured  
root@ip-10-0-0-172:/home/ubuntu#
```

2. Fork repositories

<https://github.com/UnpredictablePrashant/learnerReportCS frontend>
<https://github.com/UnpredictablePrashant/learnerReportCS backend/tree/main>

Forked repo below:

<https://github.com/akshaybhu/learnerReportCS frontend>
<https://github.com/akshaybhu/learnerReportCS backend>

```

root@ip-10-0-0-172:/home/ubuntu# git clone https://github.com/akshaybhu/learnerReportCS_frontend
Cloning into 'learnerReportCS_frontend'...
remote: Enumerating objects: 469, done.
remote: Counting objects: 100% (98/98), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 469 (delta 64), reused 63 (delta 54), pack-reused 371 (from 1)
Receiving objects: 100% (469/469), 982.32 KiB | 14.66 MiB/s, done.
Resolving deltas: 100% (233/233), done.
root@ip-10-0-0-172:/home/ubuntu# git clone https://github.com/akshaybhu/learnerReportCS_backend
Cloning into 'learnerReportCS_backend'...
remote: Enumerating objects: 261, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 261 (delta 28), reused 23 (delta 23), pack-reused 221 (from 1)
Receiving objects: 100% (261/261), 85.53 KiB | 17.11 MiB/s, done.
Resolving deltas: 100% (155/155), done.
root@ip-10-0-0-172:/home/ubuntu# ls
learnerReportCS_backend  learnerReportCS_frontend
root@ip-10-0-0-172:/home/ubuntu#

```

3. Run docker build for both frontend and backend.

docker build -t akshayap2901/frontend_img:latest .

>> ISSUE FACED <<

```

root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_frontend#
root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_frontend# docker build -t akshayap2901/frontend_img: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 2.655MB
Step 1/7 : FROM node:16
16: Pulling from library/node
311da6c465ea: Pull complete
7e9bf114588c: Pull complete
ffd9397e94b7: Pull complete
513d77925604: Pull complete
ae3b95bbaa61: Pull complete
0e421f66aff4: Pull complete
ca266fd61921: Pull complete
ee7d78be1eb9: Pull complete
Digest: sha256:f77a1aef2da8d83e45ec990f45df50f1a286c5fe8bbfb8c6e4246c6389705c0b
Status: Downloaded newer image for node:16
--> 1ddc7e4055fd
Step 2/7 : WORKDIR /usr/src/app
--> Running in a0ef5cb3312d
--> Removed intermediate container a0ef5cb3312d
--> e56a7db898b2
Step 3/7 : COPY package*.json .
--> 4d4caf69439
Step 4/7 : RUN npm install --silent
--> Running in 97e897e726fd
npm notice
npm notice New major version of npm available! 8.19.4 > 11.1.0
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v11.1.0>
npm notice Run `npm install -g npm@11.1.0` to update!
npm notice
The command '/bin/sh -c npm install --silent' returned a non-zero code: 1
root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_frontend#

```

> I removed – silent to understand the error.

```
Sending build context to Docker daemon 2.655MB
Step 1/7 : FROM node:16
--> 1ddc7e4055fd
Step 2/7 : WORKDIR /usr/src/app
--> Using cache
--> e56a7db898b2
Step 3/7 : COPY package*.json .
--> Using cache
--> 4d4cafb69439
Step 4/7 : RUN npm install
--> Running in eab5c59218ea
npm notice
npm notice New major version of npm available! 8.19.4 -> 11.1.0
npm notice Changelog: <https://github.com/npm/cli/releases/tag/v11.1.0>
npm notice Run `npm install -g npm@11.1.0` to update!
npm notice
npm ERR! code ERESOLVE
npm ERR! ERESOLVE could not resolve
npm ERR!
npm ERR! While resolving: @material-ui/core@4.12.4
npm ERR! Found: react@18.2.0
npm ERR! node_modules/react
npm ERR!   react@"^18.2.0" from the root project
npm ERR!   peer react@">=18" from @chakra-ui/accordion@2.1.4
npm ERR!   node_modules/@chakra-ui/accordion
npm ERR!     @chakra-ui/accordion@"2.1.4" from @chakra-ui/react@2.4.3
npm ERR!     node_modules/@chakra-ui/react
npm ERR!       @chakra-ui/react@"^2.4.3" from the root project
npm ERR!       99 more (@chakra-ui/alert, @chakra-ui/avatar, ...)
npm ERR!
npm ERR! Could not resolve dependency:
npm ERR! peer react@"^16.8.0 || ^17.0.0" from @material-ui/core@4.12.4
npm ERR! node_modules/@material-ui/core
npm ERR!   @material-ui/core@"^4.12.4" from the root project
npm ERR!
npm ERR! Conflicting peer dependency: react@17.0.2
npm ERR! node_modules/react
npm ERR!   peer react@"^16.8.0 || ^17.0.0" from @material-ui/core@4.12.4
npm ERR!   node_modules/@material-ui/core
npm ERR!     @material-ui/core@"^4.12.4" from the root project
npm ERR!
npm ERR! Fix the upstream dependency conflict, or retry
npm ERR! this command with --force, or --legacy-peer-deps
npm ERR! to accept an incorrect (and potentially broken) dependency resolution.
npm ERR!
npm ERR! See /root/.npm/eresolve-report.txt for a full report.

npm ERR! A complete log of this run can be found in:
npm ERR!   /root/.npm/_logs/2025-02-25T20_04_05_695Z-debug-0.log
The command '/bin/sh -c npm install' returned a non-zero code: 1
root@ip-10-0-0-172:/home/ubuntu/learnerReportCS-frontend#
```

> the error message now clearly points to a dependency conflict, specifically with react and @material-ui/core.

- > This means @material-ui/core is incompatible with your React 18 version.
- > Replace @material-ui/core with @mui/material. And try again.

```
[root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_frontend# grep -r material-ui/core .  
.package.json:      "@material-ui/core": "^4.12.4",  
.package-lock.json:      "@material-ui/core": "^4.12.4",  
.package-lock.json:      "node_modules/@material-ui/core": {  
.package-lock.json:          "resolved": "https://registry.npmjs.org/@material-ui/core/-/core-4.12.4.tgz",  
/usr/share/nginx/html/StaticPage/build/StaticPage.js:import React from 'react'; import { Container, Grid } from '@material-ui/core';
```

Also modified, NODE from 16 to [node:23-alpine3.20](#)

```
[root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_frontend# cat Dockerfile
FROM node:23-alpine3.20

# Working directory be app
WORKDIR /usr/src/app

COPY package*.json ./

### Installing dependencies

RUN npm install --silent

# copy local files to app folder
COPY . .

EXPOSE 3000

CMD ["npm", "start"]
root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_frontend#
```

```

root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_frontend# docker build -t akshayap2901/frontend_img: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 3.473MB
Step 1/7 : FROM node:23-alpine3.20
23-alpine3.20: Pulling from library/node
0a9a5dfd008f: Pull complete
8ebf0d079dc4: Pull complete
5145f4f53ee4: Pull complete
fb829c2a5c3e: Pull complete
Digest: sha256:1b4d82a2bee3b5ea9a9048bd3b326220ac77065a84cbef339a8f41c698f0ec66
Status: Downloaded newer image for node:23-alpine3.20
--> 376057dfd4ca
Step 2/7 : WORKDIR /usr/src/app
--> Running in 0649509be997
--> Removed intermediate container 0649509be997
--> 45b5eb48b339
Step 3/7 : COPY package*.json ./
--> b7f146a47436
Step 4/7 : RUN npm install --silent
--> Running in 05f7631e3cd2
--> Removed intermediate container 05f7631e3cd2
--> 21d931366677
Step 5/7 : COPY . .
--> 6c1befc25dd9
Step 6/7 : EXPOSE 3000
--> Running in 137cb0a8f091
--> Removed intermediate container 137cb0a8f091
--> e08a8db36974
Step 7/7 : CMD [ "npm", "start" ]
--> Running in 6d4aa5808d0c
--> Removed intermediate container 6d4aa5808d0c
--> f9d1b87990e7
Successfully built f9d1b87990e7
Successfully tagged akshayap2901/frontend_img:latest
root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_frontend#

```

docker build -t akshayap2901/backend_img:latest .

Modified the node version to latest with alpine image.

```

root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_frontend#
root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_frontend# cat ../learnerReportCS_backend/Dockerfile
FROM node:23-alpine3.20

# Create app directory
WORKDIR /usr/src/app

# Install app dependencies
# A wildcard is used to ensure both package.json AND package-lock.json are copied
# where available (npm@5+)
COPY package*.json ./
RUN npm install --silent
# For production RUN npm ci --only=production

# Bundle app source
COPY . .

EXPOSE 3000

#Define the command
CMD [ "node", "index.js" ]
root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_frontend#

```

```

root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_frontend# cd ../../learnerReportCS_backend/
root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_backend# docker build -t akshayap2901/backend_img: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 393.2kB
Step 1/7 : FROM node:23-alpine3.20
--> 376057dfd4ca
Step 2/7 : WORKDIR /usr/src/app
--> Using cache
--> 45b5eb48b339
Step 3/7 : COPY package*.json .
--> de3af074568a
Step 4/7 : RUN npm install --silent
--> Running in 1a0b9d59750f
--> Removed intermediate container 1a0b9d59750f
--> c1e60f55b9fc
Step 5/7 : COPY . .
--> 99b8c4bb0494
Step 6/7 : EXPOSE 3000
--> Running in 1438f9b3e9d7
--> Removed intermediate container 1438f9b3e9d7
--> 0bcd4db2c050
Step 7/7 : CMD [ "node", "index.js" ]
--> Running in 11fe06586af8
--> Removed intermediate container 11fe06586af8
--> 2e6a98817ae1
Successfully built 2e6a98817ae1
Successfully tagged akshayap2901/backend_img:latest
root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_backend#
root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_backend# 

```

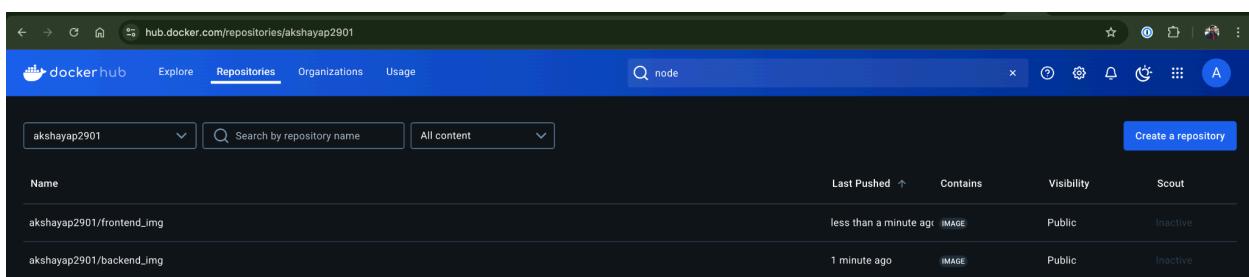
```

root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_backend#
root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_backend# docker images | grep akshay
akshayap2901/backend_img      latest          2e6a98817ae1   17 seconds ago  202MB
akshayap2901/frontend_img     latest          f9d1b87990e7   3 minutes ago   603MB
root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_backend#
root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_backend#
root@ip-10-0-0-172:/home/ubuntu/learnerReportCS_backend# 

```

Push images to Docker HUB

docker push akshayap2901/backend_img:latest
 docker push akshayap2901/frontend_img:latest



Create K8S deployment/Service/configmap files apply them:

```
kubectl apply -f frontend-configmap.yaml
```

```
kubectl apply -f frontend-deploy.yaml
```

```
kubectl apply -f frontend-svc.yaml
```

```
Events:
Type Reason Age From Message
Normal Scheduled 44s default-scheduler Successfully assigned default/frontend-deployment-59bf5f79cd-mwjz2 to minikube
Normal Pulling 44s kubelet Pulling image "akshayap2901/frontend_img:latest"
Normal Pulled 3s kubelet Successfully pulled image "akshayap2901/frontend_img:latest" in 41.118s (41.118s including waiting). Image size: 602582147 bytes.
Normal Created 2s kubelet Created container: frontend
Normal Started 2s kubelet Started container frontend
ubuntu@ip-10-0-0-172:~/kubernetes_files$ kubectl get pods
NAME READY STATUS RESTARTS AGE
frontend-deployment-59bf5f79cd-mwjz2 1/1 Running 0 62s
frontend-deployment-59bf5f79cd-rwf2c 1/1 Running 0 62s
ubuntu@ip-10-0-0-172:~/kubernetes_files$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
frontend-service LoadBalancer 10.100.196.196 <pending> 80:30210/TCP 64s
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 115s
ubuntu@ip-10-0-0-172:~/kubernetes_files$
```

```
kubectl apply -f backend-configmap.yaml
```

```
kubectl apply -f backend-deploy.yaml
```

```
kubectl apply -f backend-secret.yaml
```

```
kubectl apply -f backend-svc.yaml
```

```
[ubuntu@ip-10-0-0-172:~/kubernetes_files]$ kubectl apply -f backend-configmap.yaml
configmap/backend-config created
[ubuntu@ip-10-0-0-172:~/kubernetes_files]$ kubectl apply -f backend-deploy.yaml
deployment.apps/backend-deployment created
[ubuntu@ip-10-0-0-172:~/kubernetes_files]$ kubectl apply -f backend-secret.yaml
secret/backend-secret created
[ubuntu@ip-10-0-0-172:~/kubernetes_files]$ kubectl apply -f backend-svc.yaml
service/backend-service created
[ubuntu@ip-10-0-0-172:~/kubernetes_files$]
[ubuntu@ip-10-0-0-172:~/kubernetes_files$]
[ubuntu@ip-10-0-0-172:~/kubernetes_files$] kubectl get pods
NAME READY STATUS RESTARTS AGE
backend-deployment-94dbfd85b-8pj55 0/1 Error 1 (12s ago) 15s
backend-deployment-94dbfd85b-gp9lg 0/1 Error 1 (12s ago) 15s
frontend-deployment-59bf5f79cd-mwjz2 1/1 Running 1 (14m ago) 18m
frontend-deployment-59bf5f79cd-rwf2c 1/1 Running 0 18m
[ubuntu@ip-10-0-0-172:~/kubernetes_files$] kubectl get pods
NAME READY STATUS RESTARTS AGE
backend-deployment-94dbfd85b-8pj55 1/1 Running 2 (13s ago) 19s
backend-deployment-94dbfd85b-gp9lg 0/1 Error 1 (16s ago) 19s
frontend-deployment-59bf5f79cd-mwjz2 1/1 Running 1 (14m ago) 18m
frontend-deployment-59bf5f79cd-rwf2c 1/1 Running 0 18m
[ubuntu@ip-10-0-0-172:~/kubernetes_files$]
```

```
kubectl apply -f mongo-deployment.yaml  
kubectl apply -f mongo-secret.yaml
```

```
[ubuntu@ip-10-0-0-172:~/kubernetes_files]$ kubectl apply -f mongo-deployment.yaml  
deployment.apps/mongodb created  
[ubuntu@ip-10-0-0-172:~/kubernetes_files]$ kubectl apply -f mongo-secret.yaml  
secret/mongo-secret created  
[ubuntu@ip-10-0-0-172:~/kubernetes_files]$  
[ubuntu@ip-10-0-0-172:~/kubernetes_files]$  
[ubuntu@ip-10-0-0-172:~/kubernetes_files]$  
[ubuntu@ip-10-0-0-172:~/kubernetes_files]$ kubectl get pods  
NAME READY STATUS RESTARTS AGE  
backend-deployment-94dbfd85b-8pj55 0/1 CrashLoopBackOff 4 (75s ago) 2m41s  
backend-deployment-94dbfd85b-gp9lg 0/1 CrashLoopBackOff 4 (66s ago) 2m41s  
frontend-deployment-59bf5f79cd-mwjz2 1/1 Running 1 (16m ago) 20m  
frontend-deployment-59bf5f79cd-rwf2c 1/1 Running 0 20m  
mongodb-7b97b4886f-rzgz4 0/1 Pending 0 12s  
[ubuntu@ip-10-0-0-172:~/kubernetes_files]$  
[ubuntu@ip-10-0-0-172:~/kubernetes_files]$
```

Install Helm

```
curl https://baltocdn.com/helm/signing.asc | gpg --dearmor | sudo tee  
/usr/share/keyrings/helm.gpg > /dev/null
```

```
sudo apt-get install apt-transport-https --yes
```

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/helm.gpg]  
https://baltocdn.com/helm/stable/debian/ all main" | sudo tee  
/etc/apt/sources.list.d/helm-stable-debian.list
```

```
sudo apt-get update
```

```
sudo apt-get install helm
```

```
ubuntu@ip-10-0-0-172:~/kubernetes_files$ 
[ubuntu@ip-10-0-0-172:~/kubernetes_files$ sudo apt-get install helm
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  helm
0 upgraded, 1 newly installed, 0 to remove and 72 not upgraded.
Need to get 17.5 MB of archives.
After this operation, 58.2 MB of additional disk space will be used.
Get:1 https://baltocdn.com/helm/stable/debian all/main amd64 helm amd64 3.17.1-1 [17.5 MB]
Fetched 17.5 MB in 11s (1618 kB/s)
Selecting previously unselected package helm.
(Reading database ... 70987 files and directories currently installed.)
Preparing to unpack .../helm_3.17.1-1_amd64.deb ...
Unpacking helm (3.17.1-1) ...
Setting up helm (3.17.1-1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

Restarting services...

Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart docker.service
systemctl restart getty@tty1.service
systemctl restart networkd-dispatcher.service
systemctl restart serial-getty@ttyS0.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

No user sessions are running outdated binaries.
```

helm package .

```
[ubuntu@ip-10-0-0-172:~/kubernetes_files$ cd templates/
[ubuntu@ip-10-0-0-172:~/kubernetes_files/templates$ helm package .
Successfully packaged chart and saved it to: /home/ubuntu/kubernetes_files/templates/kubernetes_files-0.1.0.tgz
[ubuntu@ip-10-0-0-172:~/kubernetes_files/templates$
```

helm install kubernetes-files kubernetes-files/templates/kubernetes_files-0.1.0.tgz

```
[ubuntu@ip-10-0-0-172:~$ helm install kubernetes-files kubernetes-files/templates/kubernetes_files-0.1.0.tgz
NAME: kubernetes-files
LAST DEPLOYED: Wed Feb 26 15:33:16 2025
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
ubuntu@ip-10-0-0-172:~$
```

```
[ubuntu@ip-10-0-0-172:~$ 
[ubuntu@ip-10-0-0-172:~$ kubectl get pods
NAME                      READY   STATUS        RESTARTS   AGE
backend-deployment-94dbfd85b-8pj55   0/1    CrashLoopBackOff  11 (3m31s ago)  34m
backend-deployment-94dbfd85b-gp9lg    0/1    CrashLoopBackOff  11 (3m14s ago)  34m
frontend-deployment-59bf5f79cd-mwjzj2  1/1    Running       1 (48m ago)   52m
frontend-deployment-59bf5f79cd-rwf2c   1/1    Running       0           52m
mongodb-7b97b4886f-rzgz4            0/1    Pending        0           32m
[ubuntu@ip-10-0-0-172:~$ kubectl get svc
NAME          TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE
backend-service ClusterIP  10.99.213.119  <none>        5000/TCP    34m
frontend-service LoadBalancer  10.100.196.196  <pending>     80:30210/TCP  52m
kubernetes     ClusterIP  10.96.0.1      <none>        443/TCP     53m
[ubuntu@ip-10-0-0-172:~$ ]
```

minikube service frontend-service

```
[ubuntu@ip-10-0-0-172:~$ 
[ubuntu@ip-10-0-0-172:~$ minikube service frontend-service
|-----|-----|-----|-----|
|  NAMESPACE  |     NAME     | TARGET PORT |      URL      |
|-----|-----|-----|-----|
|  default    | frontend-service |      80      | http://192.168.49.2:30210 |
|-----|-----|-----|-----|
👉 Opening service default/frontend-service in default browser...
👉 http://192.168.49.2:30210
[ubuntu@ip-10-0-0-172:~$ 
[ubuntu@ip-10-0-0-172:~$ 
[ubuntu@ip-10-0-0-172:~$ ]
```

```
ubuntu@ip-10-0-0-172:~$ [ubuntu@ip-10-0-0-172:~$ curl http://192.168.49.2:30210
<!-- <!DOCTYPE html> -->
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <!-- <link rel="icon" href="/favicon.ico" /> -->
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />

    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/
    -->
    <!-- <link rel="manifest" href="/manifest.json" /> -->
    <!--
      Notice the use of in the tags above.
      It will be replaced with the URL of the `public` folder during the build.
      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "/favicon.ico" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
    <title>React App</title>
    <script defer src="/static/js/bundle.js"></script></head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
    <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.
      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run `npm start` or `yarn start`.
      To create a production bundle, use `npm run build` or `yarn build`.
    -->
  </body>
</html>
ubuntu@ip-10-0-0-172:~$ █
```

Automate CICD using Jenkins

1. Install Git, Docker & Kubernetes plugins in Jenkins
2. Add Dockerhub credentials to the Jenkins
 - Go to Jenkins Dashboard → Manage Jenkins → Manage Credentials. Add new credentials:
 - Kind: Username with password
 - Scope: Global
 - Username: Your Docker Hub username
 - Password: Your Docker Hub password
 - ID: dockerhub-credentials
3. Jenkins pipeline Create a new pipeline:
 - New Item → Pipeline
 - Name: MERN_CI_CD_Pipeline
 - Pipeline: Scripted Pipeline [JenkinsFile](#)

```
pipeline {  
    agent any  
    environment {  
        DOCKER_CREDENTIALS_ID = 'dockerhub-credentials'  
        DOCKER_REPO = 'akshayap2901'  
    }  
    stages {  
        stage('Checkout Code') {  
            steps {  
                git url: 'https://github.com/akshaybhu/learnerReportCS_frontend.git', branch: 'main'  
                git url: 'https://github.com/akshaybhu/learnerReportCS_backend.git', branch: 'main'  
            }  
        }  
  
        stage('Build') {  
            steps {  
                sh 'docker build -t $DOCKER_REPO/frontend-image:latest ./learnerReportCS_frontend'  
                sh 'docker build -t $DOCKER_REPO/backend-image:latest ./learnerReportCS_backend'  
            }  
        }  
        stage('Push to DockerHub') {  
            steps {  
                withCredentials([usernamePassword(credentialsId: "$DOCKER_CREDENTIALS_ID",  
usernameVariable: 'USERNAME', passwordVariable: 'PASSWORD')]) {
```

```
    sh 'echo "$PASSWORD" | docker login -u "$USERNAME" --password-stdin'
    sh 'docker push $DOCKER_REPO/frontend-image:latest'
    sh 'docker push $DOCKER_REPO/backend-image:latest'
}
}
stage('Deploy to Kubernetes') {
steps {
    sh 'helm upgrade --install kubernetes-files ./mern-stack'
}
}
}
}
```

Save and run the pipeline, Access the application using

kubectl get svc frontend-service

<http://<minikube-ip>:<node-port>>