

Mohammad Hassaan Ejaz (22I - 2434)
Mohammad Awais Rafique (22I - 2511)
SE - B

SCD - Project

Kubernetes Cluster Setup with Minikube/GitHub Actions/Jenkins

In this project, you have to set up a local Kubernetes cluster using **Minikube**, deploy a **web application** to it, and explore core Kubernetes concepts including **pods**, **services**, and **deployments**.

Project Requirements:

1. Minimum/Maximum of 2 students are allowed in a group.
2. Each group must submit a **project report** detailing how each step was completed, along with clear **instructions for running the project**.

Project Steps:

1. Installation of MiniKube and Kubectl
2. **Develop a web app**
3. Containerize it with Docker
4. Push code to GitHub
5. Create Kubernetes (deployment & services) files
6. Deploy it to Minikube using Kubernetes manifests
7. Set Up Docker Hub
8. Use GitHub Actions to automatically build and push Docker images
9. Trigger local deployment

Project Overview

1. **Install Minikube and kubectl** (for using kubernetes locally)

- Install **Minikube**: <https://minikube.sigs.k8s.io/docs/start/>

```
x ➤ moz ➔ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
sudo dpkg -i minikube_latest_amd64.deb
  % Total    % Received % Xferd  Average Speed   Time   Time     Time  Current
          Dload  Upload Total Spent   Left Speed
100 37.0M 100 37.0M    0      0  638k      0  0:00:59  0:00:59 --:--:-- 1585k
Selecting previously unselected package minikube.
(Reading database ... 212243 files and directories currently installed.)
Preparing to unpack minikube_latest_amd64.deb ...
Unpacking minikube (1.35.0-0) ...
Setting up minikube (1.35.0-0) ...

[~]
```

- Install **kubectl** (Kubernetes CLI): <https://kubernetes.io/docs/tasks/tools/>

```
x ➤ moz ➔ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
  % Total    % Received % Xferd  Average Speed   Time   Time     Time  Current
          Dload  Upload Total Spent   Left Speed
100 138 100 138    0      0  329      0 --:--:-- --:--:-- --:--:-- 329
100 57.3M 100 57.3M    0      0 1620k      0  0:00:36  0:00:36 --:--:-- 2125k
[~]
moz ➔ sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
[~]
moz ➔ kubectl version --client
Client Version: v1.33.0
Kustomize Version: v5.6.0
[~]
```

minikube version

```
x ➤ moz ➔ minikube version
minikube version: v1.35.0
commit: dd5d320e41b5451cdf3c01891bc4e13d189586ed-dirty
[~]
```

kubectl version --client

```
[~]
moz ➔ kubectl version --client
Client Version: v1.33.0
Kustomize Version: v5.6.0
[~]
```

Note: Minikube usually requires a container runtime like Docker. Make sure you have installed Docker on the system.

```
[~]
moz ➔ docker -v
Docker version 28.1.1, build 4eba377
[~]
```

2. Develop a Web App

```
[~/Desktop/22I - 2434/Semester 6/scd/project]
moz ➤ main ➤ ls
RecruiteFlow

[~/Desktop/22I - 2434/Semester 6/scd/project]
moz ➤ main ➤ ls RecruiteFlow
Backend Documents Frontend README.md

[~/Desktop/22I - 2434/Semester 6/scd/project]
```

Requirement: Keep your application code organized under an **/app** folder to make it easier to manage with Docker and Kubernetes files.

```
[~/Desktop/22I - 2434/Semester 6/scd/project]
moz ➤ main ➤ mv RecruiteFlow app

[~/Desktop/22I - 2434/Semester 6/scd/project]
moz ➤ main ➤ ls
app

[~/Desktop/22I - 2434/Semester 6/scd/project]
```

Backend

Starting

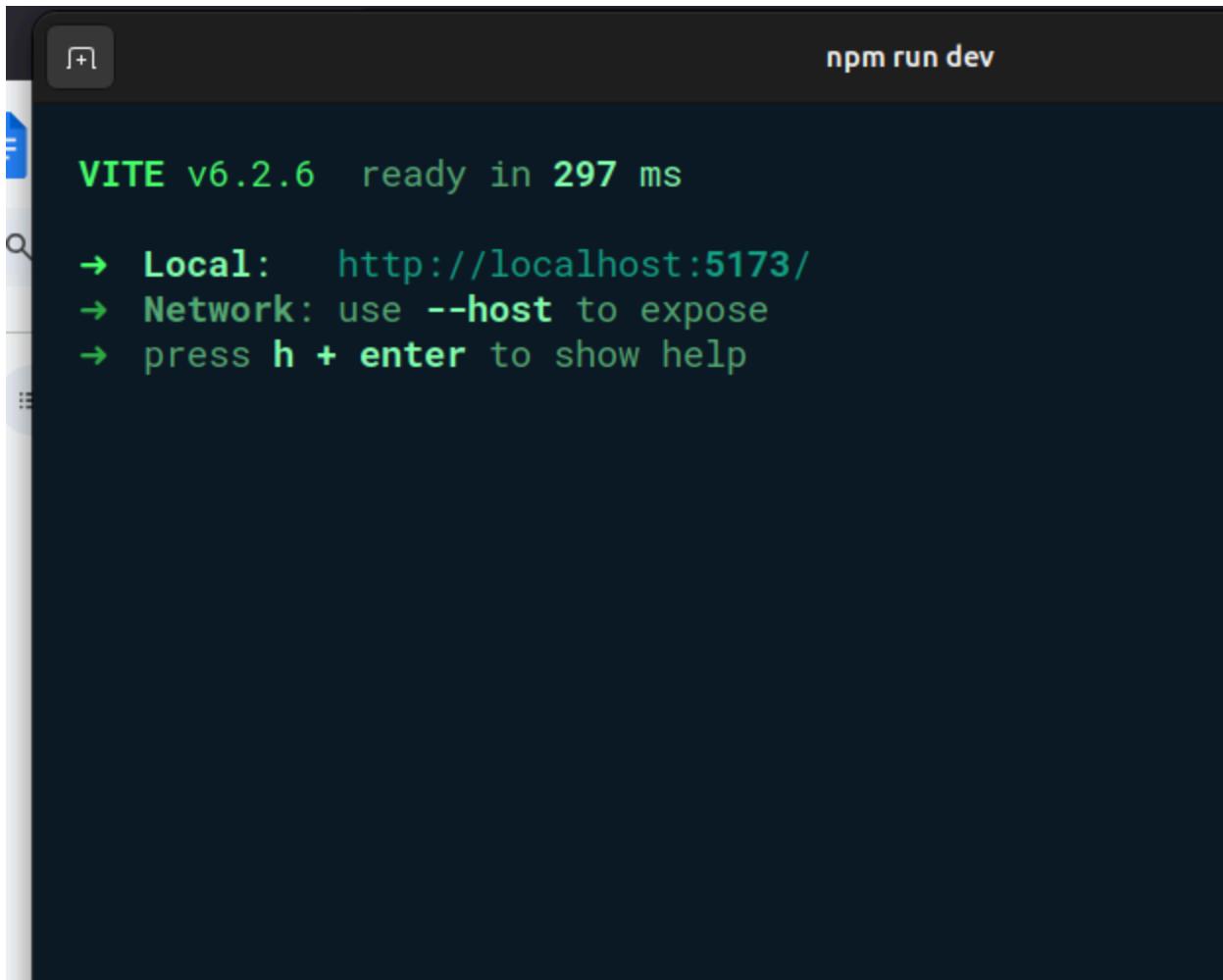
```
npm error A complete log of this run can be found in  
05-11T22_43_50_013Z-debug-0.log  
[~/Desktop/22I - 2434/Semester 6/scd/project/app/F  
x ➤ moz ➤ p main ➤ cd ../Backend  
[~/Desktop/22I - 2434/Semester 6/scd/project/app/B  
moz ➤ p main ➤ npm start  
> backend@1.0.0 start  
> nodemon src/index.js  
  
[nodemon] 3.1.9  
[nodemon] to restart at any time, enter `rs`  
[nodemon] watching path(s): *.*  
[nodemon] watching extensions: js,mjs,cjs,json  
[nodemon] starting `node src/index.js`  
Database Connected Success Host localhost  
Server is started on port 8000
```

Testing /api/auth/login

```
http://localhost:8000/api/auth/login \
-d '{"name": "Master0z786", "cnic": "3630211425713", "email": "hassanejaz400@gmail.com", "password": "Zait@1234", "userType": "Admin", "phone": "923144685510"}'

{"statusCode":200, "data": {"LoggedInUser": {"_id": "6821178c2ac56c04566983c", "name": "masteroz786", "cnic": "3630211425713", "phone": "923144685510", "email": "hassanejaz400@gmail.com", "userType": "Admin", "createdAt": "2025-05-11T21:32:51.173Z", "updatedAt": "2025-05-11T21:44:53.204Z", "__v": 0}, "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQ0I2ODIxMTc4M2MyYWM1NmMwNDU2Njk4M2MiLCJlbWFpbCI6Imhhc3NhmvqYXo0MDBAZ21haWwuY29tIiwiaWF0IjoxNzQ2OTk50DkzLCJleHAi0E3NDcwMDA3MzI9.dfnCj2i7jtKxLzmW0mUxoA4fUyp0orBd8DGcnB1tAY", "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.yJfaWQiOiI2ODIxMTc4M2MyYWM1NmMwNDU2Njk4M2MiLCJpYXQiOjE3NY50Tk40TMsiMv4cCI6MTc0NzAwMDU40X0.g575NzdFD_60VVvr7Ukr5cXLD100pJG1TGHy9UzohE"}, "message": "User Logged In Successfully", "success": true}%
```

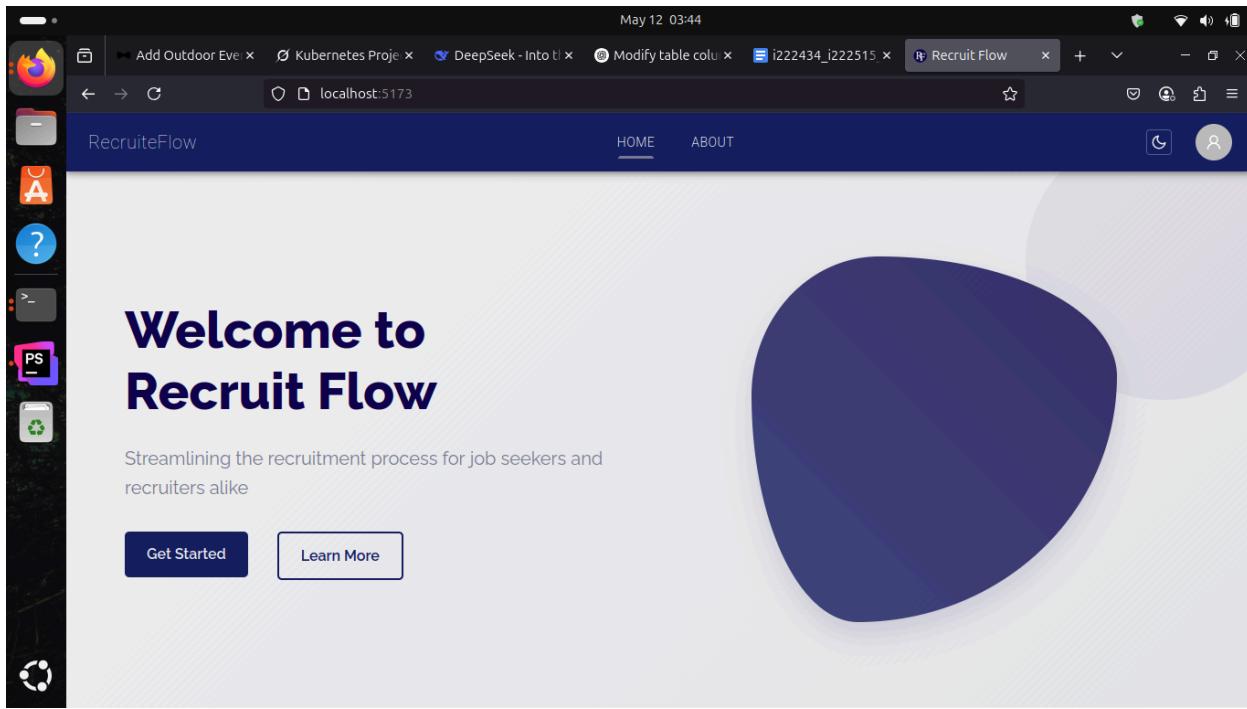
Frontend



A screenshot of a terminal window titled "npm run dev". The window displays the output of a Vite server start command. The text is white on a dark background. It shows the Vite version (v6.2.6), the time taken to start (297 ms), and the local host URL (http://localhost:5173/). It also includes instructions for exposing the network and viewing help.

```
VITE v6.2.6 ready in 297 ms

→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```



3. Containerized the Application

Build the web application Docker file. Remember to build and test the build locally using the command

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/Frontend]
moz ➤❯ main ➤ ls
dist      eslint.config.js  node_modules  package-lock.json  public      sample.env  vite.config.js
Dockerfile  index.html      package.json   postcss.config.cjs README.md  src

[~/Desktop/22I - 2434/Semester 6/scd/project/app/Frontend]
moz ➤❯ main ➤
```

```
May 12 03:46
less Dockerfile
FROM node AS build
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build
# Serve the app using a lightweight web server
FROM nginx:alpine
# Copy the build output from the build stage to the NGINX container
COPY --from=build /app/dist /usr/share/nginx/dist
# Expose the port the app will be accessible on
EXPOSE 80
# Start NGINX
CMD ["nginx", "-g", "daemon off;"]
Dockerfile (END)
```

```
docker build -t masteroz/scd-frontend .
```

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/Frontend]
moz ➤ main ➤ docker build -t masteroz/scd-frontend .
[+] Building 2.2s (15/15) FINISHED
  => [internal] load build definition from Dockerfile
  => => transferring dockerfile: 435B
  => [internal] load metadata for docker.io/library/nginx:alpine
  => [internal] load metadata for docker.io/library/node:latest
  => [auth] library/node:pull token for registry-1.docker.io
  => [internal] load .dockerignore
  => => transferring context: 99B
  => [build 1/6] FROM docker.io/library/node:latest@sha256:149a0b6925212aa
  => [stage-1 1/2] FROM docker.io/library/nginx:alpine
  => [internal] load build context
  => => transferring context: 3.69kB
  => CACHED [build 2/6] WORKDIR /app
  => CACHED [build 3/6] COPY package*.json ./
  => CACHED [build 4/6] RUN npm install
  => CACHED [build 5/6] COPY . .
  => CACHED [build 6/6] RUN npm run build
  => CACHED [stage-1 2/2] COPY --from=build /app/dist /usr/share/nginx/dis
  => exporting to image
  => => exporting layers
  => => writing image sha256:5f15a298293b6a91cc8e6a01376decc30f6b8192db9d8
  => => naming to docker.io/masteroz/scd-frontend
[~/Desktop/22I - 2434/Semester 6/scd/project/app/Frontend]
moz ➤ main ➤
```

```
docker run -p 3000:3000 masteroz/scd-frontend
```



```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/Frontend]
 moz ➤ main ➤ docker rm 2c83dfd82f56 -f
2c83dfd82f56

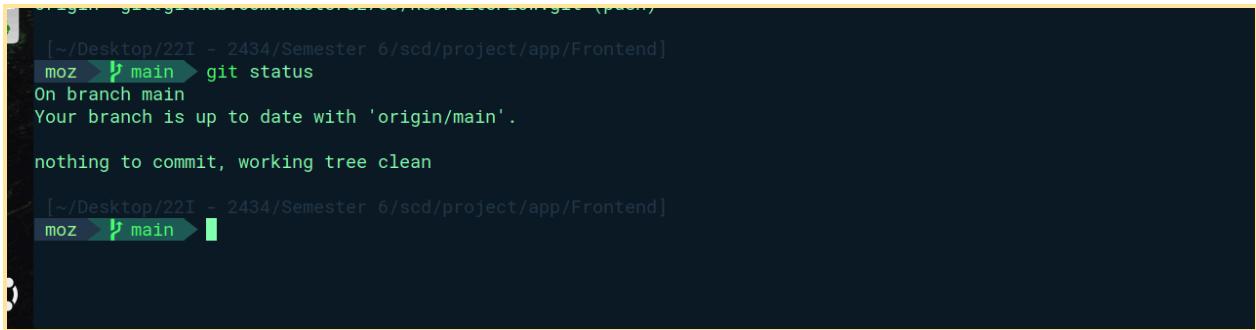
[~/Desktop/22I - 2434/Semester 6/scd/project/app/Frontend]
 moz ➤ main ➤ docker run -p 3000:3000 -d masteroz/scd-frontend
6a4cb179632183cca224a1fd9547ac3d8a4ceafde2b0cfb8c9c365c68f93bbf0

[~/Desktop/22I - 2434/Semester 6/scd/project/app/Frontend]
 moz ➤ main ➤
```

4. Push Code to GitHub

Steps to follow:

A. Initialize Git Repository (if not done already):



```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/Frontend]
 moz ➤ main ➤ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

[~/Desktop/22I - 2434/Semester 6/scd/project/app/Frontend]
 moz ➤ main ➤
```

B. Create a GitHub repository on GitHub and follow the instructions to add a remote:



```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/Frontend]
 moz ➤ main ➤ git remote -v
origin  git@github.com:MasterOz786/RecruiteFlow.git (fetch)
origin  git@github.com:MasterOz786/RecruiteFlow.git (push)
```

Important: Ensure the repository contains **all necessary files** for your app to run and build correctly.

The screenshot shows a GitHub repository page for 'RecruitFlow' (Public). The repository was forked from 'Tayyabsamii/RecruitFlow'. It has 2 branches and 0 tags. The main branch is 2 commits ahead of the original. The repository contains files like 'Backend', 'Documents', 'Frontend', and 'README.md'. The README describes RecruitFlow as a comprehensive web application for recruitment. The repository has 132 commits, with the latest being an addition of a dockerized frontend. The repository has 0 stars, 0 forks, and 0 watching. There are no releases or packages published. The languages used are JavaScript (77.3%) and SCSS (22.5%).

Issue: node_modules were being tracked in the docker images which was quite extra space and more time for the image building process. Had to use .dockerignore to prevent that from happening

The screenshot shows a terminal window with the following command history:

```
~/Desktop/22I - 2434/Semester 6/scd/project/app/Frontend/.dockerignore
"~/Desktop/22I - 2434/Semester 6/scd/project/app/Frontend/.dockerignore" 6L, 59B
```

The terminal shows the creation of a .dockerignore file in the Frontend directory of a project.

A screenshot of a terminal window on a Mac OS X system. The title bar shows "May 12 03:53". The main area displays a file tree for a project directory:

```
node_modules
build
.dockerignore
Dockerfile
npm-debug.log
```

The terminal prompt is at the bottom left, showing the current directory as `~/Desktop/22I - 2434/Semester 6/scd/project/app/Backend/`. The status bar at the bottom right shows "1,1 All".

A screenshot of a terminal window showing the output of a `git log --oneline` command:

```
f4ea7a9 (HEAD -> main, origin/main, origin/dockit, origin/HEAD, dockit) add: .dockerignore + dockerized frontend
03f593c sample .env.example + fix variable scope + dockerized backend
afa0656 Merge pull request #37 from Tayyabsami1/Tayyab
```

5. Create Kubernetes Files

Backend Deployment Manifest

The screenshot shows a terminal window titled 'vim.' running on a Linux desktop. The terminal displays a YAML file for a Kubernetes Deployment named 'backend'. The manifest includes a Deployment with one replica, a Service with port 5000, and a Container named 'backend' running the image 'masteroz/scd-backend:latest' on port 5000. It also specifies a MongoDB URI environment variable. The file path is indicated as '/Desktop/22I - 2434/Semester 6/scd/project/app/k8s/backend-deployment.yaml'.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: backend
spec:
  replicas: 1
  selector:
    matchLabels:
      app: backend
  template:
    metadata:
      labels:
        app: backend
    spec:
      containers:
        - name: backend
          image: masteroz/scd-backend:latest
          ports:
            - containerPort: 5000
          env:
            - name: MONGO_URI
              value: "mongodb://mongodb:27017/RecruitFlow"
...
apiVersion: v1
kind: Service
metadata:
  name: backend
spec:
  ports:
    - port: 5000
      targetPort: 5000
  selector:
    app: backend
  type: ClusterIP
~
~
~
~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s/backend-deployment.yaml
```

Frontend Deployment Manifest

The screenshot shows a terminal window titled 'vim.' running on a Linux desktop. The terminal displays a YAML file for a Kubernetes Deployment named 'frontend'. The manifest includes a Deployment with one replica, a Service with port 3000, and a Container named 'frontend' running the image 'masteroz/scd-frontend:latest' on port 3000. The file path is indicated as '/Desktop/22I - 2434/Semester 6/scd/project/app/k8s/frontend-deployment.yaml'.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend
spec:
  replicas: 1
  selector:
    matchLabels:
      app: frontend
  template:
    metadata:
      labels:
        app: frontend
    spec:
      containers:
        - name: frontend
          image: masteroz/scd-frontend:latest
          ports:
            - containerPort: 3000
...
apiVersion: v1
kind: Service
metadata:
  name: frontend
spec:
  ports:
    - port: 3000
      targetPort: 3000
  selector:
    app: frontend
  type: NodePort
~
~
~
~
~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s/frontend-deployment.yaml
```

Mongo Deployment Manifest



The screenshot shows a terminal window with a dark theme. On the left, there is a vertical dock containing icons for various applications: a yellow flower, a folder, a document with a red 'A', a question mark, and a purple 'PS' icon. The main area of the terminal is a vim editor displaying a YAML configuration file. The file defines a Deployment and a Service for a MongoDB application. The Deployment specifies a single pod named 'mongodb' with a selector for 'app: mongodb'. The Service maps port 27017 to the MongoDB port 27017. The vim status bar at the bottom shows the file path as '/Desktop/22T - 2434/Semester 6/scd/project/app/k8s/mongo-deployment.yaml' and indicates it has 31L and 475B.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mongodb
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mongodb
  template:
    metadata:
      labels:
        app: mongodb
    spec:
      containers:
        - name: mongodb
          image: mongo:latest
          ports:
            - containerPort: 27017
...
apiVersion: v1
kind: Service
metadata:
  name: mongodb
spec:
  ports:
    - port: 27017
      targetPort: 27017
  selector:
    app: mongodb
  type: ClusterIP
~
```



6. Start the Kubernetes Cluster

Launch your local Kubernetes cluster using Minikube. This step sets up a single-node Kubernetes environment for deploying and testing your app.

minikube start (This command sets up a single-node Kubernetes cluster locally.)

```
[~/Desktop/22] - 2434/Semester 6/scd/project/app]
moz ➤ p main minikube start
minikube v1.35.0 on Ubuntu 24.04
* Automatically selected the docker driver
* Using Docker driver with root privileges
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.46 ...
  Downloading Kubernete...
> preloaded-images-k8s-v18-v1...: 333.57 MiB / 333.57 MiB 100.00% 2.33 Mi
> gcr.io/k8s-minikube/kicbase...: 500.31 MiB / 500.31 MiB 100.00% 1.99 Mi
Creating docker container (CPUs=2, Memory=2900MB) ...
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
```

`eval $(minikube docker-env)` (If using Minikube for building images)

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app]
moz ➤ main ➤ eval $(minikube docker-env)

[~/Desktop/22I - 2434/Semester 6/scd/project/app]
moz ➤ main ➤ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS   NAMES
b0579ab16835        6e38f40d628d      "/storage-provisioner"   About an hour ago   Up About an hour
r_kube-system_75efdfc3-6e62-4997-92b2-4199af284ee_1
eac3933bc122        c69fa2e9cbf5      "/coredns -conf /etc"
system_97ab7246-c358-476d-9ca7-4154442b3fae_0
272474c181f4        registry.k8s.io/pause:3.10    "/pause"
efdf5c3-6e62-4997-92b2-4199af284ee_0
d8f373847a3b        040f9fb8acd8      "/usr/local/bin/kube_
m_869c34ef-6b30-49fa-88cd-cf38e781e00e_0
d75c4c4aa87e        registry.k8s.io/pause:3.10    "/pause"
em_97ab7246-476d-9ca7-4154442b3fae_0
03ef3bf976fc        registry.k8s.io/pause:3.10    "/pause"
4ef-6b30-49fa-88cd-cf38e781e00e_0
e30d6fffe52d        c2e17b8df4a      "kube-apiserver --ad_
_kube-system_d72da4acf4be07c9919d46b7358a5e8_0
9fe856b31a37        a389e107f4ff      "kube-scheduler --au_
_kube-system_d14ce008bee3a1fbd7cf547688f9dfe_0
7d6593f0cc77        8c8bd2a8bd0      "kube-controller-man_
r-manager-minikube_kube-system_843c74f7b3bc7d704a05c31708a6a30_0
de14f502055a        a9e766b294ba      "etcd --advertise-cl_
2a289008e0b381891e9683040_0
70c4072a6662        registry.k8s.io/pause:3.10    "/pause"
a289008e0b381891e9683040_0
034c208a1149        registry.k8s.io/pause:3.10    "/pause"
m_d14ce008bee3a1fbd7cf547688f9dfe_0
e8350a59eb8a        registry.k8s.io/pause:3.10    "/pause"
m_d72da4acf4be07c9919d46b7358a5e8_0
8d0ecfc2524f        registry.k8s.io/pause:3.10    "/pause"
ube-system_843c74f7b3bc7d704a05c31708a6a30_0

[~/Desktop/22I - 2434/Semester 6/scd/project/app]
moz ➤ main ➤
```

Applying Kubernetes manifestations

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app]
moz ➤ main ➤ ls
backend-deployment.yaml  frontend-deployment.yaml  mongo-deployment.yaml

[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
moz ➤ main ➤ kubectl apply -f mongo-deployment.yaml
deployment.apps/mongodb created
service/mongodb created

[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
moz ➤ main ➤ kubectl apply -f frontend-deployment.yaml
deployment.apps/frontend created
service/frontend created

[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
moz ➤ main ➤ kubectl apply -f backend-deployment.yaml
deployment.apps/backend created
service/backend created
```

Checking the frontend url through minikube

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
moz ➤ p main ➤ minikube service frontend --url
http://192.168.49.2:30883

✖ Exiting due to SVC_UNREACHABLE: service not available: no running pod for service frontend found

[!] If the above advice does not help, please let us know:
👉 https://github.com/kubernetes/minikube/issues/new/choose

Please run `minikube logs --file=logs.txt` and attach logs.txt to the GitHub issue.
Please also attach the following file to the GitHub issue:
- /tmp/minikube_service_78d6e5a8da38705fdabf92576c016348f586e542_0.log
```



```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
x ➤ moz ➤ p main ➤ kubectl get pods
NAME           READY   STATUS      RESTARTS   AGE
backend-857877d7b6-st66w  0/1     ContainerCreating   0          74s
frontend-759fc6c895-8bkzz  0/1     ContainerCreating   0          78s
mongodb-745f6f5448-4z8jb  0/1     ContainerCreating   0          85s
```



```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
moz ➤ p main ➤
```

Issue: Pod isn't running. The problem may lie in the states of the PODs. Lets wait for them to be up and then re-run **minikube service frontend --url**

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
moz ➤ p main ➤ kubectl get pods
NAME           READY   STATUS      RESTARTS   AGE
backend-857877d7b6-st66w  1/1     Running    0          6m48s
frontend-759fc6c895-8bkzz 1/1     Running    0          6m52s
mongodb-745f6f5448-4z8jb  1/1     Running    0          6m59s

[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
moz ➤ p main ➤ minikube service mongodb --url
^[[A⚠️ service default/mongodb has no node port
! Services [default/mongodb] have type "ClusterIP" not meant to be exposed, however for local development minikube allows you to access this !

[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
moz ➤ p main ➤ minikube service backend --url
[!] service default/backend has no node port
! Services [default/backend] have type "ClusterIP" not meant to be exposed, however for local development minikube allows you to access this !

[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
moz ➤ p main ➤ minikube service frontend --url
http://192.168.49.2:30883

[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
moz ➤ p main ➤
```

Issue: Backend and mongo pods don't have node ports i.e. not exposed for public access. Even though minikube allows them to be accessed, but it will still be disallowed in production.

```

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
5722c75059d9 masteroz/scd-backend "docker-entrypoint.s_"
2 minutes ago Up 2 minutes k8s_backend_backend-857877d7b6-st66w_default_a26
c1599-c3d0-46d6-b420-af24e59f81f2_0
916272d419f7 masteroz/scd-frontend "/docker-entrypoint._."
5 minutes ago Up 5 minutes k8s_frontend_frontend-759fc6c895-8bkzz_default_1
b74f6d9-c34f-45f3-8e83-ad8b3d7174f1_0
29fa7-6be0-4431-9f36-ab626252960_0
8736e91b6f68 registry.k8s.io/pause:3.10 "/pause"
7 minutes ago Up 7 minutes k8s_POD_backend-857877d7b6-st66w_default_a26c159
9-c3d0-46d6-b420-af24e59f81f2_0
1972d8366188 registry.k8s.io/pause:3.10 "/pause"
8 minutes ago Up 8 minutes k8s_POD_frontend-759fc6c895-8bkzz_default_1bf4f6
d9-c34f-45f3-8e83-adbb3d7174f1_0
e7fc487900a2 registry.k8s.io/pause:3.10 "/pause"
8 minutes ago Up 8 minutes k8s_POD_mongodb-745f6f5448-4z8jb_default_293
7-6be0-4431-9f36-ab626252960_0
ca3e3a95d251 c69f2a9e9cbf5 "/coredns -conf /etc_"
10 minutes ago Up 10 minutes k8s_coredns_coredns-668d6bf9bc-6qsnw_kube-system
97ab7246-c358-476d-9ca7-4154442b3fae_1
ada7365b1f3 6e38f40d628d "/storage-provisioner"
10 minutes ago Up 10 minutes k8s_storage-provisioner_storage-provisioner_kube
-system_75efdf5c3-6e62-4997-92b2-4199a1f284ee_2
b021e3f236f1 040f9f8aa8c "/usr/local/bin/kube_"
10 minutes ago Up 10 minutes k8s_kube-proxy_kube-proxy-k96ct_kube-system_869c
34ef6b30-49fa-88cd-cf38e781e00e_1
bcc190258382 c2e17b8d0f4a "kube-apiserver --ad_"
10 minutes ago Up 10 minutes k8s_kube-apiserver_kube-apiserver-minikube_kube-
system_869c34ef6b30-49fa-88cd-cf38e781e00e_1
361674cd618c a389e107ff4f "kube-scheduler --au_"
10 minutes ago Up 10 minutes k8s_kube-scheduler_kube-scheduler-minikube_kube-
system_d14ce008beea5if3bd7cf547688f9dfe_1
3af26af684a a9e7e6b294ba "etcd --advertise-cl_"
10 minutes ago Up 10 minutes k8s_etcd_etcd-minikube_kube-system_2b4b75c2a2890
0e0b3881e91e9683048_1
93d3dafe0978a 8cab3d2a8bdd "kube-controller-man_"
10 minutes ago Up 10 minutes k8s_kube-controller-manager_kube-controller-mana
ger-minikube_kube-system_843c74f7b3bc7d7040a05c1708a6a30_1
ffb922db96c registry.k8s.io/pause:3.10 "/pause"
10 minutes ago Up 10 minutes k8s_POD_coredns-668d6bf9bc-6qsnw_kube-system_97a
b7246-c358-476d-9ca7-4154442b3fae_1
ec3b70ebbe78 registry.k8s.io/pause:3.10 "/pause"
10 minutes ago Up 10 minutes k8s_POD_storage-provisioner_kube-system_75efdf5c3
-6e62-4997-92b2-4199a1f284ee_1
318667eb7342 registry.k8s.io/pause:3.10 "/pause"
10 minutes ago Up 10 minutes k8s_POD_kube-proxy-k96ct_kube-system_869c34ef6b
30-49fa-88cd-cf38e781e00e_1
d3a47323762 registry.k8s.io/pause:3.10 "/pause"
10 minutes ago Up 10 minutes k8s_POD_kube-scheduler-minikube_kube-system_d14c
e088bee3a1f3bd7cf547688f9dfe_1
4b2e9f78c17b registry.k8s.io/pause:3.10 "/pause"
10 minutes ago Up 10 minutes k8s_POD_etcd-minikube_kube-system_2b4b75c2a2890
8e0b381e91e9683048_1
098c0f1eb82d registry.k8s.io/pause:3.10 "/pause"
10 minutes ago Up 10 minutes k8s_POD_kube-apiserver-minikube_kube-system_d72d
0a4cf4be077c9919d46b7358a5e8_1
3717e3422430 mongo "docker-entrypoint.s_"
10 minutes ago Up 10 minutes k8s_POD_mongo-745f6f5448-4z8jb_default_293

```

Issue: I couldn't seem to see my images on the **docker ps** command. It was actually piped to minikube vm through the command minikube docker-env. To revert back to default, we can use

eval \$(minikube docker-env --unset)

```

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
05fe2427a19c gcr.io/k8s-minikube/kicbase:v0.0.46 "/usr/local/bin/entr_"
2 hours ago Up 2 hours 127.0.0.1:32768->22/tcp, 127.0.0.1:32769->2376/tcp, 127.0.0.1:32770->5000/tcp, 127.0.0.1:32771->8443/tcp, 127.0.0.1:32772->32443/tcp minikube
6a4ac1796321 masteroz/scd-frontend "/docker-entrypoint._."
4 hours ago Up 4 hours 80/tcp, 0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp
03c429cfd804 dd3d2301f85a "docker-entrypoint.s_"
5 hours ago Up 5 hours 0.0.0.0:5000->8000/tcp, [::]:5000->8000/tcp
3717e3422430 mongo "docker-entrypoint.s_"
6 hours ago Up 6 hours 0.0.0.0:27017->27017/tcp, [::]:27017->27017/tcp

```

Issue: Can't create a namespace for the service files

```

moz ➤ kubectl create namespace scd-project
Unable to connect to the server: dial tcp 192.168.49.2:8443: connect: no route to host

```

Debug

Checking for minikube status

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
moz ➤ main ➤ sudo minikube status
[sudo] password for moz:
⚠ Profile "minikube" not found. Run "minikube profile list" to view all profiles.
👉 To start a cluster, run: "minikube start"
```

Found out minikube wasn't started, so proceeding forward

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
✖ ➤ moz ➤ main ➤ minikube start
❗ minikube v1.35.0 on Ubuntu 24.04
✨ Using the docker driver based on existing profile

❗ Exiting due to PROVIDER_DOCKER_NEWRP: "docker version --format <no value>--<no value>:<no value>" exit status 1: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.49/version": dial unix /var/run/docker.sock: connect: permission denied
💡 Suggestion: Add your user to the 'docker' group: 'sudo usermod -aG docker $USER && newgrp docker'
📘 Documentation: https://docs.docker.com/engine/install/linux-postinstall/

[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
```

Permissions error. Resolving by adding current user to **docker** group and also switching to the docker group instead of the current one (e.g. moz).

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
✖ ➤ moz ➤ main ➤ sudo usermod -aG docker $USER && newgrp docker
Agent pid 7489
Identity added: /home/moz/.ssh/vps (moz@15.235.184.251)
Identity added: /home/moz/.ssh/github (github)
```

It apparently started, now will try to create the namespace again

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
moz ➤ minikube start
minikube v1.35.0 on Ubuntu 24.04
Using the docker driver based on existing profile
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.46 ...
Restarting existing docker container for "minikube" ...
Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
Verifying Kubernetes components...
  ■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
  ★ Enabled addons: default-storageclass, storage-provisioner
  ★ Done! kubectl is now configured to use "minikube" cluster and "default" name
space by default
```

Creating namespace for kubernetes manifests

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
moz ➤ kubectl create namespace scd-project
namespace/scd-project created

[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
moz ➤ Creating namespace for kubernetes manifests
```

It worked, now applying manifest files

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app]
moz ➤ kubectl apply -f k8s/mongo-deployment.yaml -n scd-project
kubectl apply -f k8s/mongo-service.yaml -n scd-project
kubectl apply -f k8s/backend-deployment.yaml -n scd-project
kubectl apply -f k8s/backend-service.yaml -n scd-project
kubectl apply -f k8s/frontend-deployment.yaml -n scd-project
kubectl apply -f k8s/frontend-service.yaml -n scd-project
service/mongodb created
error: the namespace from the provided object "blog-app" does not match the name
space "scd-project". You must pass '--namespace=blog-app' to perform this operat
ion.
deployment.apps/backend created
service/backend created
service/scd-backend created
deployment.apps/frontend created
service/frontend created
service/scd-frontend created

[~/Desktop/22I - 2434/Semester 6/scd/project/app]
moz ➤
```

Verifying

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app]
moz ➤ main -> kubectl get pods -n scd-project -o wide
kubectl get services -n scd-project -o wide
kubectl get deployments -n scd-project -o wide
kubectl get nodes -o wide
NAME          READY   STATUS    RESTARTS   AGE     IP           NODE     NOMINATED-NODE   READINESS   G
ATES
backend-857877d7b6-pcnbn   1/1     Running   0          47s    10.244.0.11   minikube   <none>        <none>
frontend-759fc6c895-rwxlp  1/1     Running   0          47s    10.244.0.12   minikube   <none>        <none>
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)      AGE      SELECTOR
backend   ClusterIP  10.96.249.170  <none>        5000/TCP    47s      app=backend
frontend  NodePort   10.109.236.81  <none>        3000:30872/TCP 47s      app=frontend
mongodb  ClusterIP  10.101.242.95  <none>        27017/TCP   48s      app=mongodb
scd-backend ClusterIP  10.105.37.67  <none>        5000/TCP    47s      app=scd-backend
scd-frontend NodePort   10.103.176.208 <none>        3000:31860/TCP 47s      app=scd-frontend
NAME    READY   UP-TO-DATE   AVAILABLE   AGE      CONTAINERS   IMAGES           SELECTOR
backend  1/1     1           1           48s     backend       masteroz/scd-backend:latest  app=backend
frontend 1/1     1           1           47s     frontend      masteroz/scd-frontend:latest  app=frontend
NAME    STATUS   ROLES      AGE      VERSION   INTERNAL-IP      EXTERNAL-IP   OS-IMAGE           KERNEL-VERSION
N    CONTAINER-RUNTIME
minikube Ready   control-plane  174m    v1.32.0  192.168.49.2  <none>        Ubuntu 22.04.5 LTS  6.11.0-25-gen
eric  docker://27.4.1
```

Accessing the frontend

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app]
moz ➤ main -> minikube service frontend -n scd-project
|---|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|---|-----|-----|-----|
| scd-project | frontend | 3000 | http://192.168.49.2:30872 |
|---|-----|-----|-----|
👉 Opening service scd-project/frontend in default browser...

[~/Desktop/22I - 2434/Semester 6/scd/project/app]
moz ➤ main -> update.go:85: cannot change mount namespace according to change mount (/run/user/1000/doc/by-app/snap.firefox /run/user/1000/doc none bind,rw,x-snapd.ignore-missing 0 0): cannot inspect "/run/user/1000/doc": lstat /run/user/1000/doc: permission denied
Gtk-Message: 09:32:59.571: Not loading module "atk-bridge": The functionality is provided by GTK natively. Please try to not load it.
[13599, Main Thread] WARNING: Failed to read portal settings: GDBus.Error:org.freedesktop.DBus.Error.AccessDenied: Portal operation not allowed: Unable to open /proc/13599/root: 'glib warning', file /build/firefox/part/firefox/build/toolkit/xre/nsSigHandlers.cpp:201
(firefox_firefox:13599): Gdk-WARNING **: 09:32:59.579: Failed to read portal settings: GDBus.Error:org.freedesktop.DBus.Error.AccessDenied: Portal operation not allowed: Unable to open /proc/13599/root
```

Issue: Permissions error disallowing access for reading and not allowing changing namespace, mount location and other necessary dependencies/processes to be authorized and function

Debug

We can let the warnings persist as they are based on external dependencies which don't alter our cluster in any fatal way and still be able to access the application.

But we won't!

A solution for this is to re-install the dependent module based on compatible versions. In our case, lets try removing FireFox's snap and get it installed from APT repository

May 12 09:43

```

Pin-Priority: -1

moz > p main -> sudo add-apt-repository ppa:mozillateam/ppa
sudo apt update
PPA publishes debugsym, you may need to include 'main/debug' component
For more information see https://help.ubuntu.com/community/UbuntuPPA
URLs: https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/
Suites: noble
Components: main

Description
Mozilla Team's Firefox stable + 128 ESR and Thunderbird 128 stable builds
Support for Ubuntu 16.04 / 16.04 ESM is included for 115 ESR based releases only.
More info: https://launchpad.net/~mozillateam/+archive/ubuntu/ppa
Press [ENTER] to continue or Ctrl-C to cancel.
Hit:0 http://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu InRelease
Hit:1 http://ppa.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:2 http://ppa.archive.ubuntu.com/ubuntu noble InRelease
Hit:3 https://ci1.github.com/packages stable InRelease
Hit:4 https://packages.microsoft.com/repos/code stable InRelease
Hit:5 https://cse.ubuntu.com/apps/ubuntu/noble/main InRelease [126 kB]
Hit:6 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main InRelease
Hit:7 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ InRelease
Get:8 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ [350 kB]
Get:9 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ [24.4 kB]
Ign:10 https://ppa.jenkins.io/dubian-stable binary/ InRelease
Get:11 https://ppa.jenkins.io/dubian-stable InRelease
Get:12 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ [7,480 kB]
Get:13 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ [21.5 kB]
Get:14 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ [7,474 kB]
Get:15 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ [20.5 kB]
Get:16 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ [52.2 kB]
Get:17 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ [20.5 kB]
Get:18 http://security.ubuntu.com/ubuntu/noble-security/universe amd64 Packages [52.2 kB]
Get:19 http://security.ubuntu.com/ubuntu/noble-security/universe i386 Packages [20.5 kB]
Get:20 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ [7,473 kB]
Get:21 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main Translation-en [7,144 kB]
Fetched 11.5 MB in 97.0 kB/s
Reading package lists... Done
Hit:0 http://ppa.archive.ubuntu.com/ubuntu noble InRelease
Hit:1 http://ppa.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:2 http://ppa.archive.ubuntu.com/ubuntu noble InRelease
Hit:3 https://download.docker.com/linux/ubuntu noble InRelease
Hit:4 https://ci1.github.com/packages stable InRelease
Hit:5 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main InRelease
Hit:6 https://packages.microsoft.com/repos/code stable InRelease
Hit:7 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main InRelease
Ign:8 https://ppa.jenkins.io/dubian-stable binary/ InRelease
Hit:9 https://ppa.jenkins.io/dubian-stable InRelease
Hit:10 https://ppa.jenkins.io/dubian-stable binary/ InRelease
Hit:11 https://ppa.jenkins.io/dubian-stable InRelease
Hit:12 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ [7,480 kB]
Hit:13 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ [21.5 kB]
Hit:14 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ [7,474 kB]
Hit:15 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ [20.5 kB]
Hit:16 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ [52.2 kB]
Hit:17 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ [20.5 kB]
Hit:18 http://security.ubuntu.com/ubuntu/noble-security/universe amd64 Packages [52.2 kB]
Hit:19 http://security.ubuntu.com/ubuntu/noble-security/universe i386 Packages [20.5 kB]
Hit:20 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main/binary/ [7,473 kB]
Hit:21 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu/noble/main Translation-en [7,144 kB]
Reading state information... Done
Building dependency tree... Done
Reading state information... Done
Building dependency tree... Done
Reading state information... Done
4 packages can be upgraded. Run 'apt list --upgradable' to see them.

moz > p main -> echo 'Pin: Release o1P:PPA-mozillateam
Pin: Release o1P:PPA-mozillateam
Pin-Priority: 1000
Pin: release o1P:PPA-mozillateam
Pin-Priority: 1000
'
| sudo tee /etc/apt/preferences.d/mozilla-firefox
Pin: Priority: 1000
Pin: release o1P:PPA-mozillateam
Pin-Priority: 1000
Pin: release o1P:PPA-mozillateam
Pin-Priority: 1000
moz > p main ->

```

May 12 09:44

```

moz@Spa-Of-MasterOz:~/Desktop/22I - 2434/Semester 6/scd/project/app
Pin-Priority: 1001

moz > p main -> sudo apt-get install firefox
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  xul-ext-ubufox
Suggested packages:
  fonts-lyx
The following NEW packages will be installed:
  xul-ext-ubufox
The following packages will be DOWNGRADED:
  firefox
0 upgraded, 1 newly installed, 1 downgraded, 0 to remove and 4 not upgraded.
Need to get 75.7 MB of archives.
After this operation, 274 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://pk.archive.ubuntu.com/ubuntu noble/universe amd64 xul-ext-ubufox all 3.4-0ubuntu1.17.10.4 [2,108 B]
Get:2 https://ppa.launchpadcontent.net/mozillateam/ppa/ubuntu noble/main amd64 firefox amd64 138.0.1+build1-0ubuntu0.24.04.1~mt1 [75.7 MB]
Fetched 75.7 MB in 42s (1,814 kB/s)
dpkg: warning: downgrading firefox from :1:snapi-0ubuntu5 to 138.0.1+build1-0ubuntu0.24.04.1~mt1
(Reading database ... 212244 files and directories currently installed.)
Preparing to unpack .../firefox_138.0.1+build1-0ubuntu0.24.04.1~mt1_amd64.deb ...
Unpacking firefox (138.0.1+build1-0ubuntu0.24.04.1~mt1) over (:1:snapi-0ubuntu5) ...
Selecting previously unselected package xul-ext-ubufox.
Preparing to unpack .../xul-ext-ubufox_3.4-0ubuntu1.17.10.4_all.deb ...
Unpacking xul-ext-ubufox (3.4-0ubuntu1.17.10.4) ...
Setting up firefox (138.0.1+build1-0ubuntu0.24.04.1~mt1) ...
Please restart all running instances of firefox, or you will experience problems.
Setting up xul-ext-ubufox (3.4-0ubuntu1.17.10.4) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menu (3.36.0-1.1ubuntu3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for desktop-file-utils (0.27-2build1) ...

[~/Desktop/22I - 2434/Semester 6/scd/project/app]
moz > p main ->

```

Now since Firefox is installed from APT's repository using Mozilla's team PPA,

```

[~/Desktop/22I - 2434/Semester 6/scd/project/app]
x  moz > p main -> firefox -
[30027] Sandbox: CanCreateUserNamespace() unshare(CLONE_NEWPID): EPERM
Mozilla Firefox 138.0

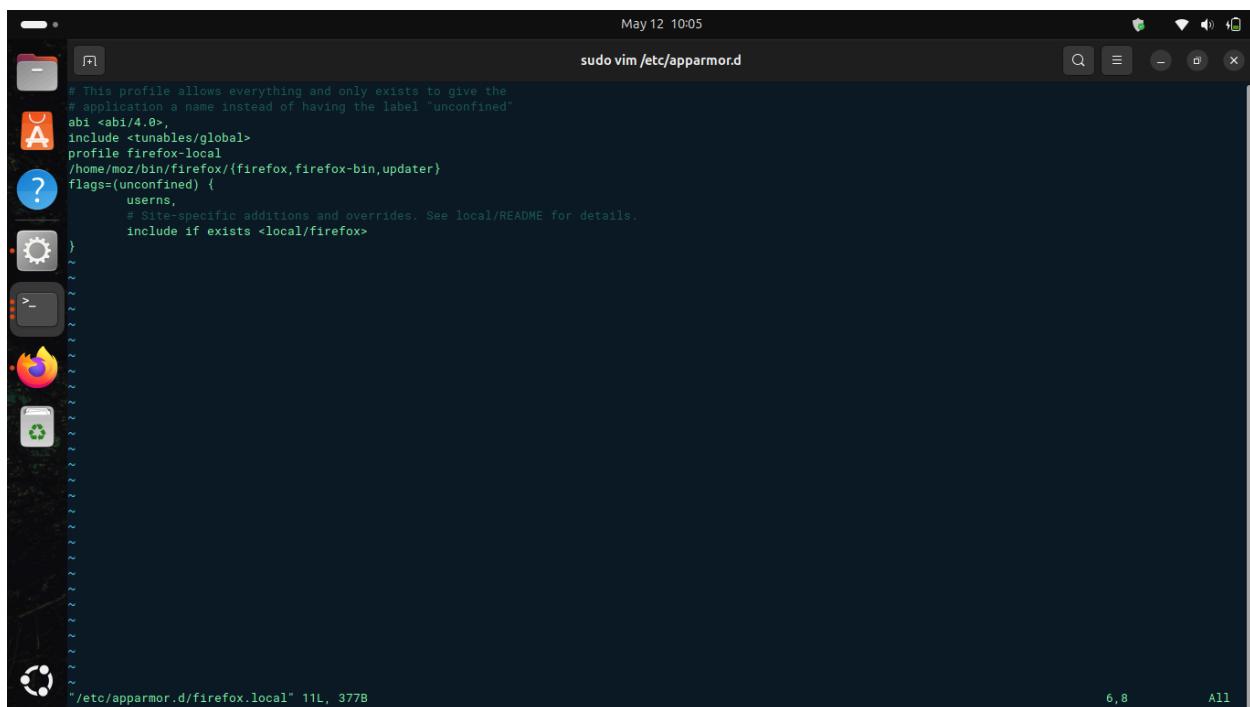
```

We shall initiate the command **minikube service frontend -n scd-project**

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app]
moz ➤ main ➤ minikube service frontend -n scd-project
|-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| scd-project | frontend | 3000 | http://192.168.49.2:30872 |
|-----|-----|-----|-----|
👉 Opening service scd-project/frontend in default browser...
[42729] Sandbox: CanCreateUserNamespace() unshare(CLONE_NEWPID): EPERM
[42722] Wayland Proxy [0x73afd4956010] Error: StartProxyServer(): bind() error :
Permission denied
```

Issue: Permission denied, hence the browser can't bind the port and it is inaccessible.

Debug It's apparently an flag raised by the Firefox security features, hence to fix it we'll use https://support.mozilla.org/en-US/kb/linux-security-warning?as=u&utm_source=inproduct as our source and create a **firefox.local** file in **/etc/apparmor.d/**



```
# This profile allows everything and only exists to give the
# application a name instead of having the label "unconfined"
abi <abi/4.0>;
include <tunables/global>
profile firefox-local
/home/moz/bin/firefox/{firefox,firefox-bin,update}
flags=(unconfined) {
    users,
    # Site-specific additions and overrides. See local/README for details.
    include if exists <local/firefox>
}
```

Then we'll restart the apparmor service for changes to take effect

```
sudo systemctl restart apparmor.service
```

Now, lets run frontend from minikube again

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app]
moz ➤ main -> minikube service frontend -n scd-project
|-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| scd-project | frontend | 3000 | http://192.168.49.2:30872 |
|-----|-----|-----|-----|
🔗 Opening service scd-project/frontend in default browser...
[42729] Sandbox: CanCreateUserNamespaces() unshare(CLONE_NEWPID): EPERM
[42722] Wayland Proxy [0x73af4956010] Error: StartProxyServer(): bind() error : Permission denied
```

Issue Continued: Same error

Debug: Listing all the services, deployments inside the cluster

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
moz ➤ main -> kubectl get all -n scd-project

NAME                               READY   STATUS    RESTARTS   AGE
pod/backend-857877d7b6-jzdth     1/1     Running   0          24s
pod/foreground-7c9f7d468d-dz9zk   1/1     Running   0          23s

NAME           TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/backend   ClusterIP  10.111.5.60 <none>        5000/TCP  24s
service/foreground   NodePort   10.103.56.2  <none>        80:31743/TCP  23s
service/mongodb   ClusterIP  10.103.144.83 <none>        27017/TCP  23s
service/scd-backend   ClusterIP  10.109.189.17 <none>        5000/TCP  23s

NAME           READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/backend   1/1       1           1           24s
deployment.apps/foreground   1/1       1           1           23s

NAME           DESIRED   CURRENT   READY   AGE
replicaset.apps/backend-857877d7b6   1         1         1         24s
replicaset.apps/foreground-7c9f7d468d   1         1         1         23s
```

Looks fine, lets try changing the port by changing the exposed port for frontend
For that, lets update the frontend Dockerfile and manifest files

May 13 21:59 vim.

```
COPY package*.json ./  
RUN npm install  
  
COPY . .  
RUN npm run build  
  
# Serve the app using a lightweight web server  
FROM nginx:alpine  
  
# Copy the build output from the build stage to the NGINX container  
COPY --from=build /app/dist /usr/share/nginx/html  
  
# Expose the port the app will be accessible on  
EXPOSE 80  
  
# Start NGINX  
CMD ["nginx", "-g", "daemon off;"]
```

~/Desktop/22I - 2434/Semester 6/scd/project/app/Frontend/Dockerfile 1,1 All

May 13 22:00 vim.

```
apiVersion: v1  
kind: Service  
metadata:  
  name: frontend  
spec:  
  ports:  
    - port: 80  
      targetPort: 80  
    selector:  
      app: frontend  
    type: NodePort
```

~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s/frontend-service.yaml 1,1 All ^[

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend
spec:
  replicas: 1
  selector:
    matchLabels:
      app: frontend
  template:
    metadata:
      labels:
        app: frontend
    spec:
      containers:
        - name: frontend
          image: masteroz/scd-frontend:latest
          ports:
            - containerPort: 80
...
~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s/frontend-deployment.yaml 1,1 All
~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s/frontend-deployment.yaml" 21L, 340B
```

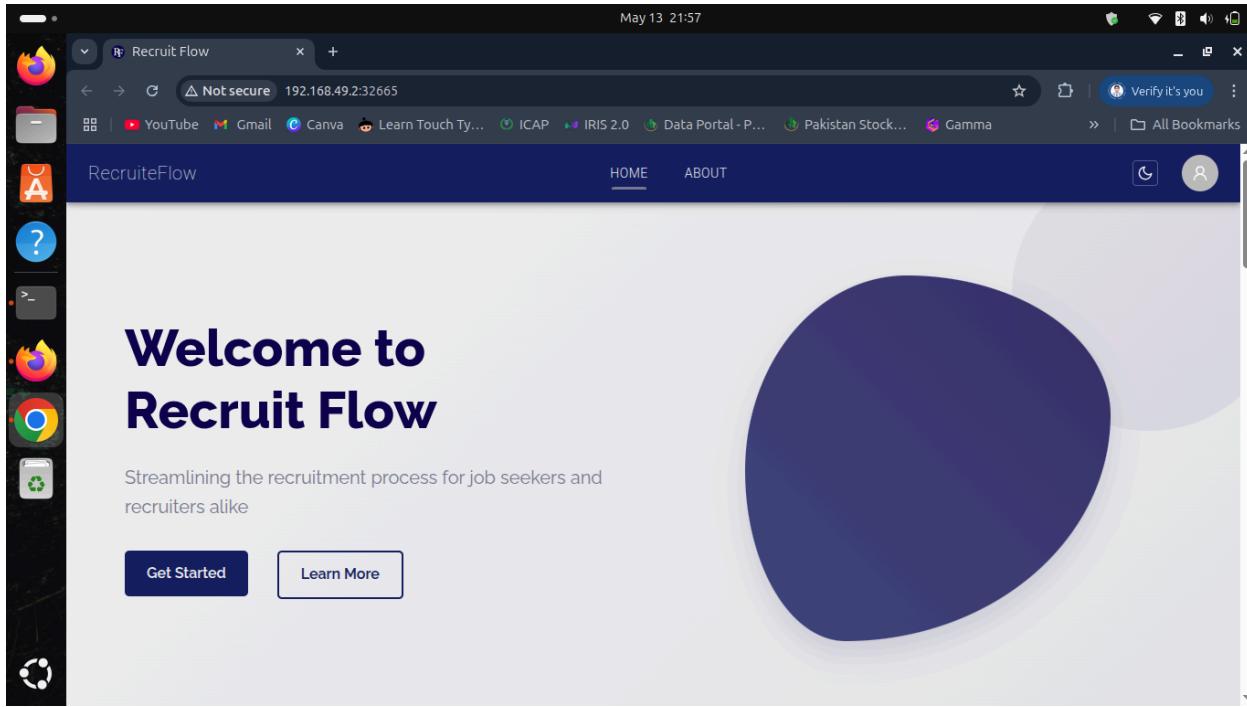
Crux: Changed port 3000 to 80 for the frontend dockerfile then re-ran the deployments and service files

Now re-running the service through minikube

```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/k8s]
x ➤ moz ➤ main ➤ minikube service frontend -n scd-project

|-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| scd-project | frontend | 80 | http://192.168.49.2:32665 |
|-----|-----|-----|-----|
💡 Opening service scd-project/frontend in default browser...
Opening in existing browser session.
```

The issue got solved!



7. Setup Docker Hub

In this step, you'll create a Docker Hub repository and upload your Docker image so it can be pulled by Kubernetes during deployment.

Issue: Copying from PhpStorm terminal didn't work; you have to bypass the pseudo-terminal to override the SIGINT before passing the command to shell through CTRL + SHIFT + V.

It couldn't recognize the PAT when copying using CTRL + SHIFT + C

```
[~/Desktop/221 - 2434/Semester 6/scd/project/app]
x moz ➤ y scavhunt ➤ cat ~/Desktop/tokay/docker | cut -b 20-32
SKLU3h_po0-D3

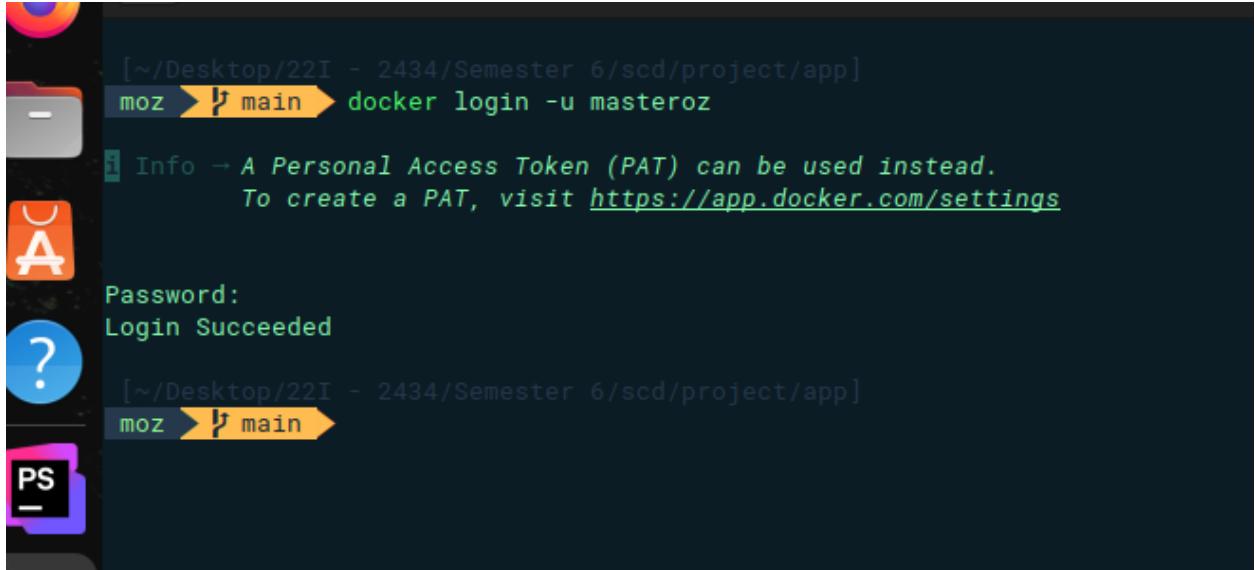
Using cut to expose only some part (12+1 bytes) of the token, not the entire token.

[~/Desktop/221 - 2434/Semester 6/scd/project/app]
x moz ➤ y main ➤ docker login -u masteroz

Info → A Personal Access Token (PAT) can be used instead.
To create a PAT, visit https://app.docker.com/settings

Password:
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password
```

Fixing by using CTRL+C on PhpStorm shell and then pasting it here



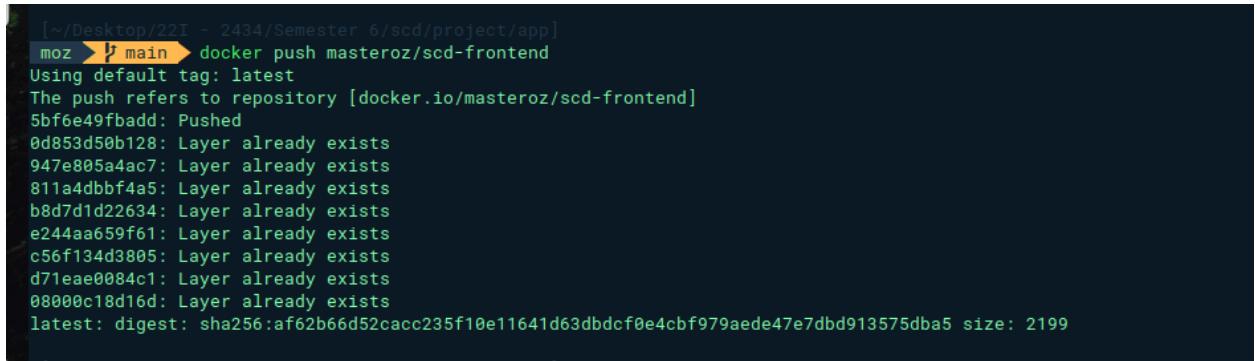
```
[~/Desktop/22I - 2434/Semester 6/scd/project/app]
moz ➤ docker login -u masteroz

i Info → A Personal Access Token (PAT) can be used instead.
To create a PAT, visit https://app.docker.com/settings

Password:
Login Succeeded

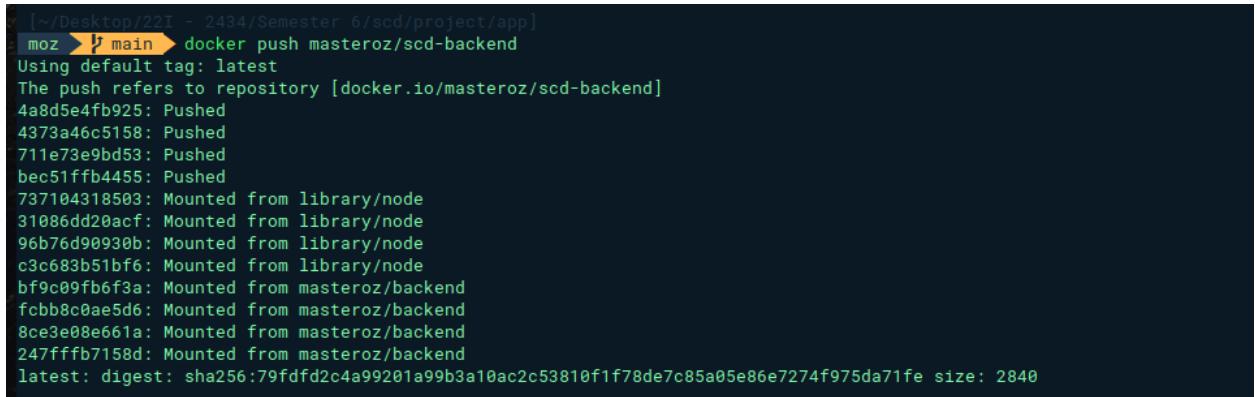
[~/Desktop/22I - 2434/Semester 6/scd/project/app]
moz ➤
```

Pushing scd-frontend



```
[~/Desktop/22I - 2434/Semester 6/scd/project/app]
moz ➤ docker push masteroz/scd-frontend
Using default tag: latest
The push refers to repository [docker.io/masteroz/scd-frontend]
5bf6e49fbadd: Pushed
0d853d50b128: Layer already exists
947e805a4acf: Layer already exists
811a4dbbf4a5: Layer already exists
b8d7d1d22634: Layer already exists
e244aa659f61: Layer already exists
c56f134d3805: Layer already exists
d71eae0084c1: Layer already exists
08000c18d16d: Layer already exists
latest: digest: sha256:af62b66d52acc235f10e11641d63dbdcf0e4cbf979aede47e7dbd913575dba5 size: 2199
```

Pushing scd-backend

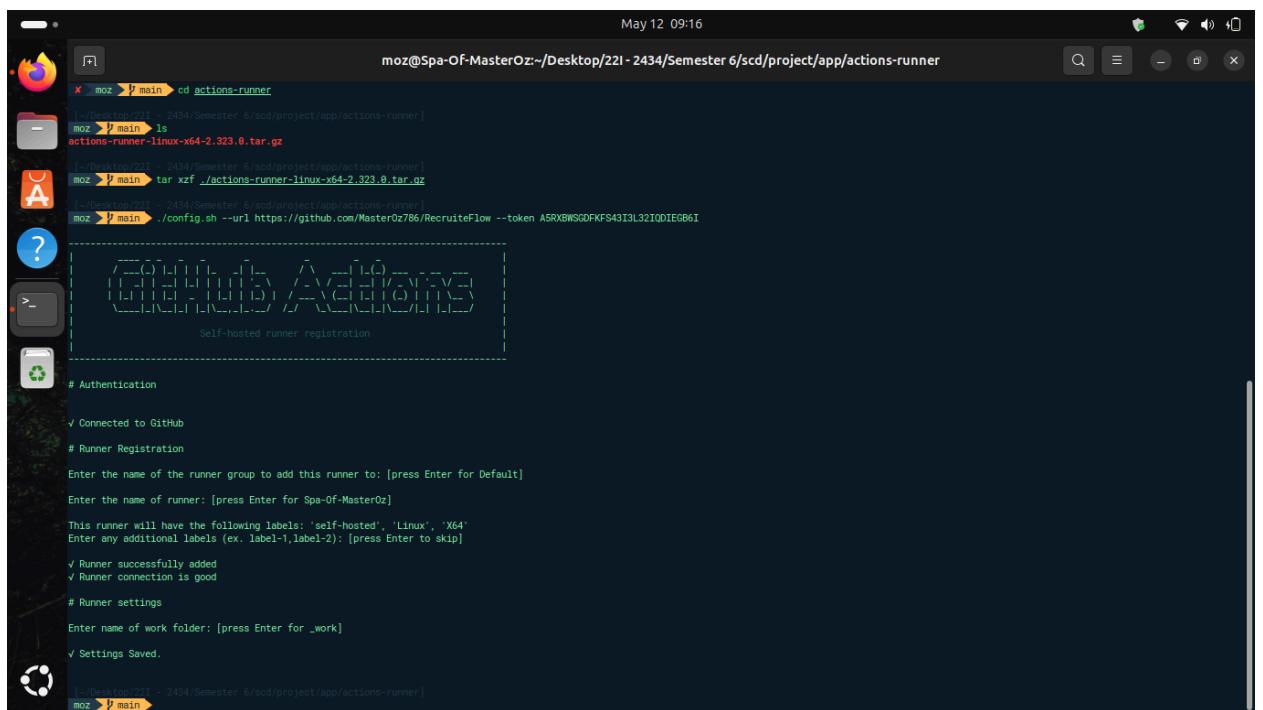
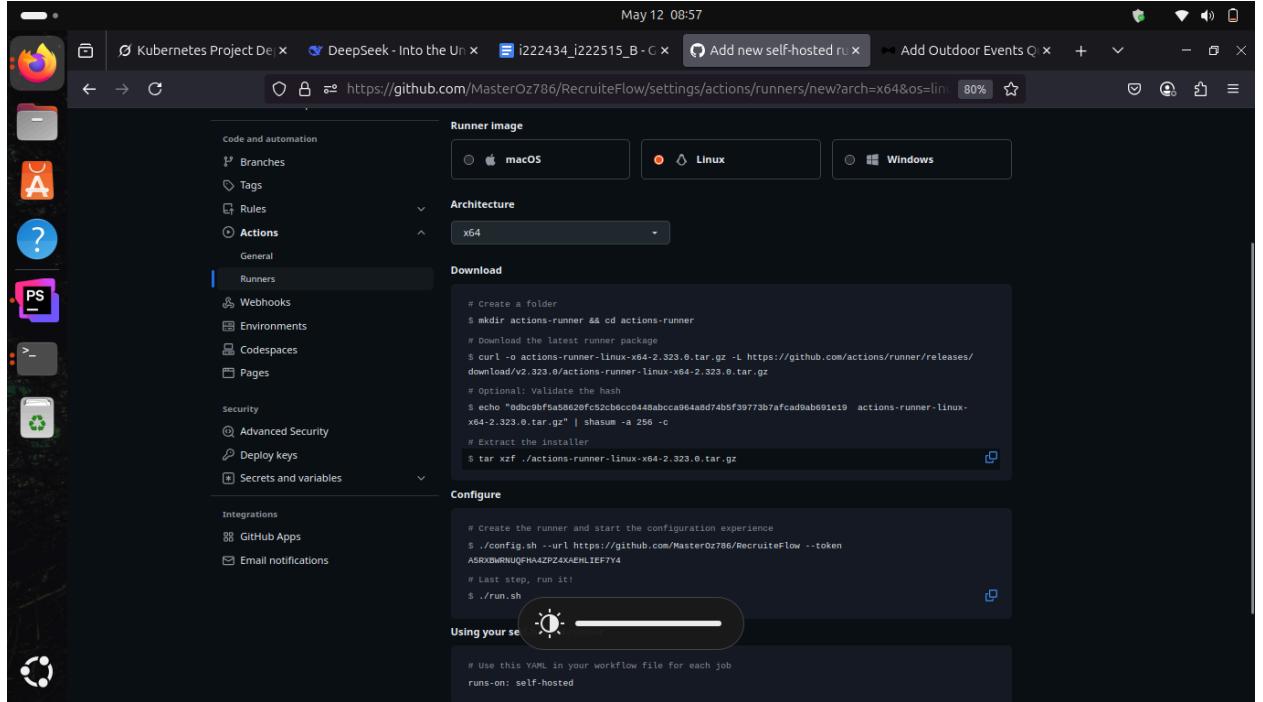


```
[~/Desktop/22T - 2434/Semester 6/scd/project/app]
moz ➤ docker push masteroz/scd-backend
Using default tag: latest
The push refers to repository [docker.io/masteroz/scd-backend]
4a8d5e4fb925: Pushed
4373a46c5158: Pushed
711e73e9bd53: Pushed
bec51ffb4455: Pushed
737104318503: Mounted from library/node
31086dd20acf: Mounted from library/node
96b76d90930b: Mounted from library/node
c3c683b51bf6: Mounted from library/node
bf9c09fb6f3a: Mounted from masteroz/backend
fcbb8c0ae5d6: Mounted from masteroz/backend
8ce3e08e661a: Mounted from masteroz/backend
247fffb7158d: Mounted from masteroz/backend
latest: digest: sha256:79fdfd2c4a99201a99b3a10ac2c53810f1f78de7c85a05e86e7274f975da71fe size: 2840
```

8. Set Up GitHub Actions Workflow

As you are using **Minikube** running on your local machine. So, you will be using GitHub Actions on a **self-hosted runner** on your machine that has access to Minikube and Docker.

1. A self-hosted GitHub Actions runner



A screenshot of a terminal window titled "vim." showing a directory listing of Kubernetes configuration files. The files listed are:

```
Netrw Directory Listing
Sorted by      name
Sort sequence: [ \ ]$ \<core\%(\.\d+\+)\>=\>, \.h$, \.c$, \.cpp$, \~\=*\$, *, \.o$, \
Quick Help: <F1>:help -:go up dir D:delete R:rename s:sort-by x:special
=====
./
backend-deployment.yaml
backend-service.yaml
frontend-deployment.yaml
frontend-service.yaml
mongo-deployment.yaml
mongo-service.yaml
~
```

The terminal status bar at the bottom shows "k8s [RO] :Rex" and "13,1 All".

2. Create a deploy.yml file in your repo

.github/workflows/deploy.yml file

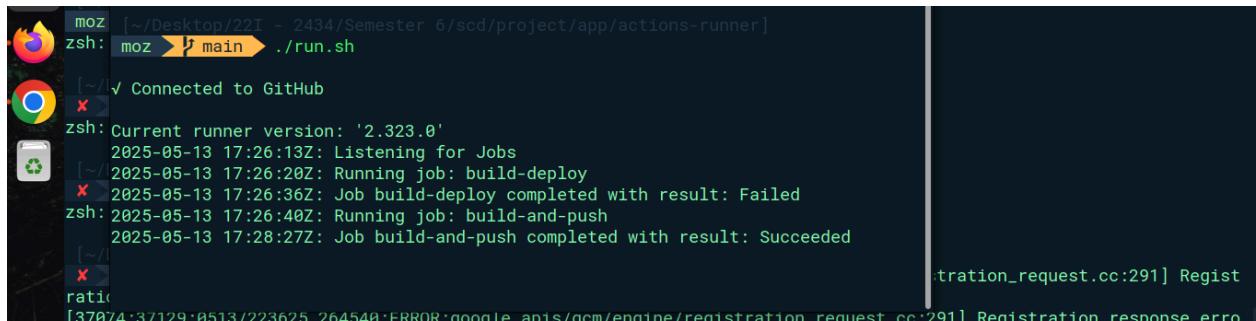
A screenshot of a terminal window titled "vim." showing a GitHub Actions workflow file named "deploy.yml". The file defines a workflow named "Build and Deploy to Minikube" with the following steps:

```
name: Build and Deploy to Minikube
on:
  push:
    branches:
      - main
jobs:
  build-deploy:
    runs-on: self-hosted
    steps:
      - uses: actions/checkout@v3
      - run: echo "Verifying environment"
      - run: echo "Minikube version: $(minikube version)"
      - run: echo "Minikube status: $(minikube status)"
      - run: echo "Kubectl version: $(kubectl version --client)"
      - run: echo "Configuring Docker to use Minikube's environment"
      - run: eval $(minikube docker-env)
      - uses: actions/docker@v2
      - name: Build backend Docker image
        runs:
          - uses: actions/docker@v2
            with:
              image: "msteroz/scd-backend:latest"
              command: "docker build -t msteroz/scd-backend:latest ./backend"
              success: "Backend Docker image built successfully"
      - name: Build frontend Docker image
        runs:
          - uses: actions/docker@v2
            with:
              image: "msteroz/scd-frontend:latest"
              command: "docker build -t msteroz/scd-frontend:latest ./frontend"
              success: "Frontend Docker image built successfully"
      - name: Login to Docker Hub
        runs:
          - run: echo "Logging into Docker Hub..."
          - run: echo "${{ secrets.DOCKER_PASSWORD }}" | docker login -u "${{ secrets.DOCKER_USERNAME }}" --password-stdin
      - name: Push Docker Images
        runs:
          - run: docker push msteroz/scd-backend:latest
          - run: echo "Backend image pushed successfully"
          - run: docker push msteroz/scd-frontend:latest
          - run: echo "Frontend image pushed successfully"
      - name: Deploy to Minikube
        runs:
          - run: kubectl apply -f k8s/mongodb-deployment.yaml -n scd-project
          - run: kubectl apply -f k8s/backend-deployment.yaml -n scd-project
          - run: kubectl apply -f k8s/backend-service.yaml -n scd-project
          - run: kubectl apply -f k8s/frontend-deployment.yaml -n scd-project
          - run: kubectl apply -f k8s/frontend-service.yaml -n scd-project
          - name: Verify deployment
            runs:
              - run: echo "Checking Kubernetes resources..."
              - run: kubectl get pods -n scd-project -o wide
              - run: echo "Services"
              - run: kubectl get services -n scd-project -o wide
              - run: echo "Deployments"
              - run: kubectl get deployments -n scd-project -o wide
              - run: echo "Nodes"
              - run: kubectl get nodes -o wide
```

The terminal status bar at the bottom shows "/Desktop/2II_2024/Semester 0/scd/project/app/.github/workflows/deploy.yml" and "1,1 All".

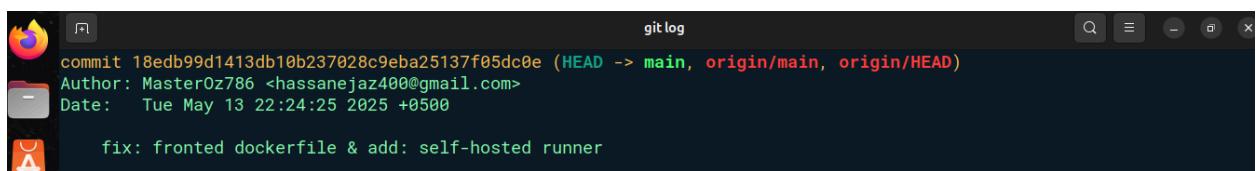
Self-Hosted Runner Results

Making sure that we have the self-hosted runner script initiated



```
[~/Desktop/22I - 2434/Semester 6/scd/project/app/actions-runner]
zsh: ~/Desktop/22I - 2434/Semester 6/scd/project/app/actions-runner]
zsh: moz > main ./run.sh
zsh: ✓ Connected to GitHub
zsh: Current runner version: '2.323.0'
2025-05-13 17:26:13Z: Listening for Jobs
2025-05-13 17:26:20Z: Running job: build-deploy
2025-05-13 17:26:36Z: Job build-deploy completed with result: Failed
zsh: 2025-05-13 17:26:40Z: Running job: build-and-push
2025-05-13 17:28:27Z: Job build-and-push completed with result: Succeeded
[~/Desktop/22I - 2434/Semester 6/scd/project/app/actions-runner]
zsh: ./run.sh
zsh: [37074-3]129-0513/223625-264548:ERROR:google_apis/gcm/engine/registration_request.cc:2911 Registration_response_error
zsh: [37074-3]129-0513/223625-264548:ERROR:google_apis/gcm/engine/registration_request.cc:2911 Registration_response_error
```

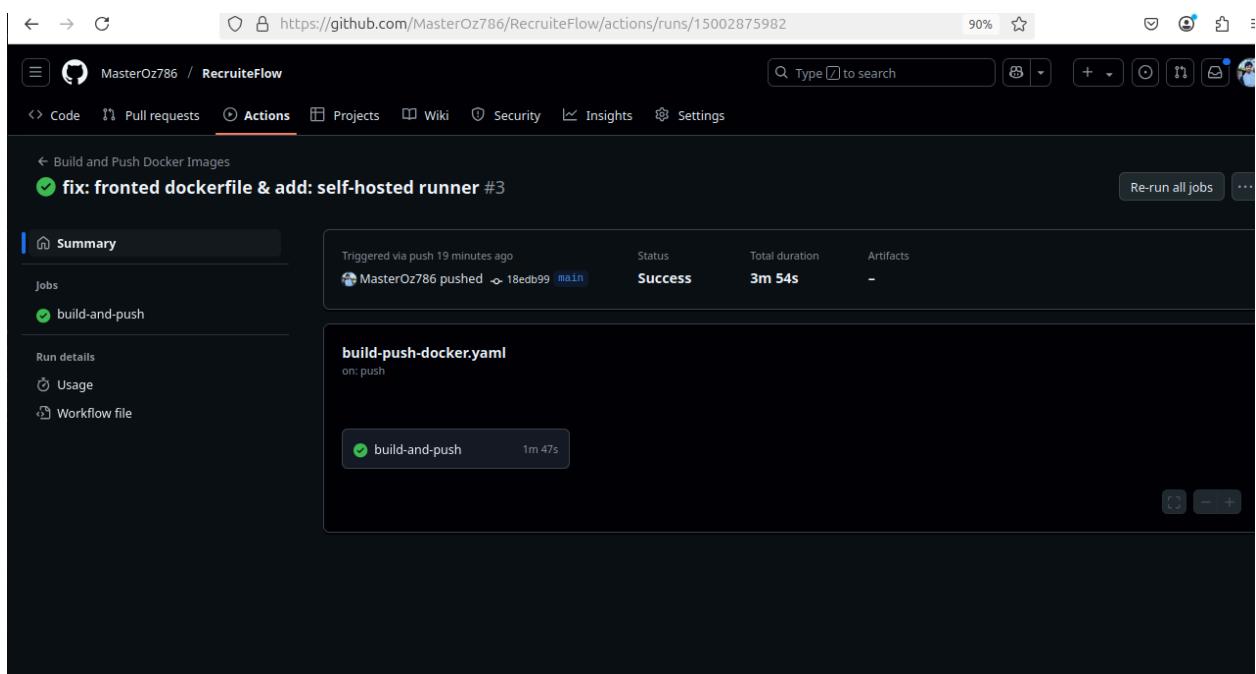
Committing and pushing the code



```
git log
commit 18edb99d1413db10b237028c9eba25137f05dc0e (HEAD -> main, origin/main, origin/HEAD)
Author: MasterOz786 <hassaneja2400@gmail.com>
Date: Tue May 13 22:24:25 2025 +0500

fix: fronted dockerfile & add: self-hosted runner
```

GitHub Actions depicting the self-hosted runner's magic



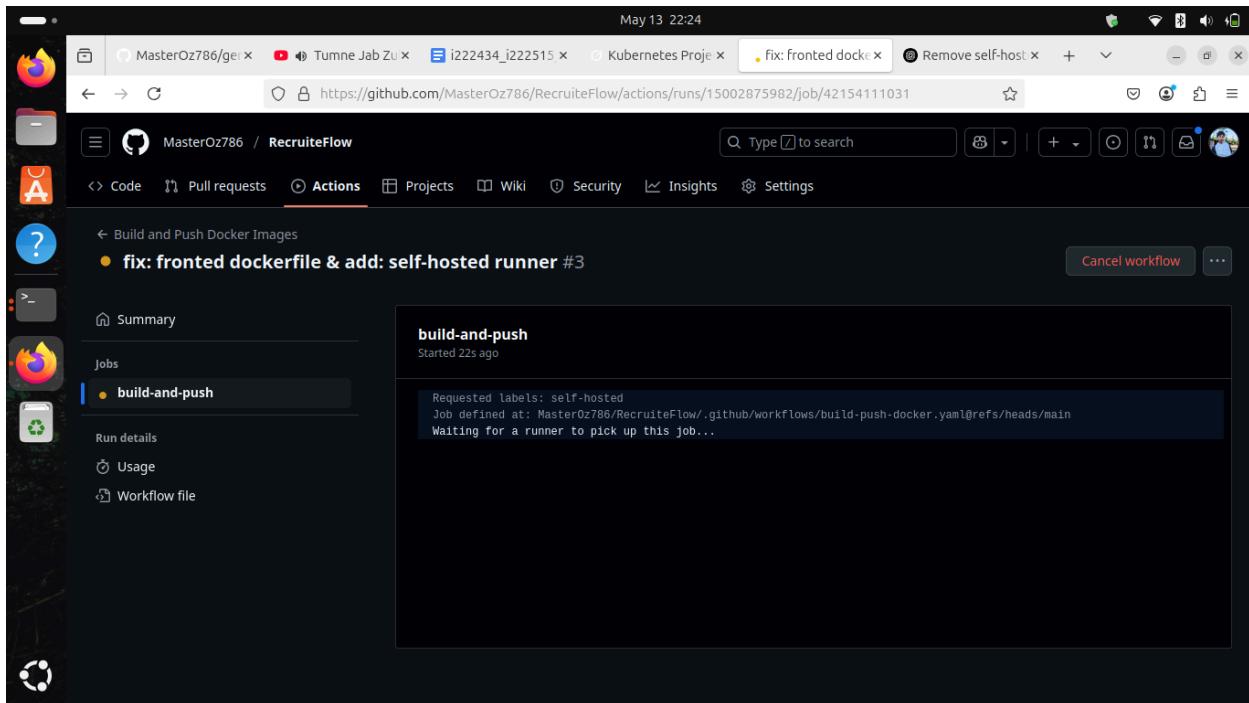
The screenshot shows the GitHub Actions interface for a workflow named "fix: fronted dockerfile & add: self-hosted runner #3". The workflow triggered via push 19 minutes ago by MasterOz786. The status is Success, total duration is 3m 54s, and there are no artifacts.

Summary

- Jobs:
 - build-and-push
- Run details
- Usage
- Workflow file

build-push-docker.yaml
on: push

Job	Status	Duration
build-and-push	Success	1m 47s



fix: fronted dockerfile & add: self-hosted runner #3

[Cancel workflow](#) ...

build-and-push

Started 52s ago

Search logs ⚙

Set up job 8s

```
1 Current runner version: '2.323.0'
2 Runner name: 'Spa-Of-MasterOz'
3 Machine name: 'Spa-Of-MasterOz'
4 ▶ GITHUB_TOKEN Permissions
20 Secret source: Actions
21 Prepare workflow directory
22 Prepare all required actions
23 Getting action download info
24 Download action repository 'actions/checkout@v3' (SHA:f43a0e5ff2bd294095638e18286ca9a3d1956744)
25 Download action repository 'docker/login-action@v2' (SHA:465a07811f14bebb1938fbed4728c6a1ff8901fc)
26 Download action repository 'docker/build-push-action@v4' (SHA:0a97817b6ade9f46837855d676c4cca3a2471fc9)
27 Complete job name: build-and-push
```

Checkout code 3s

```
53 remote: Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
54
55 ▶ Determining the checkout info
```

Started 7m 55s ago

- Checkout code
 - remote: Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
 - Determining the checkout info
 - Checking out the ref
 - /usr/bin/git log -1 --format='%H'
 - '18edb99d1413db10b237028c9eba25137f05dc0e'
- Log in to Docker Hub
 - Run docker/login-action@v2
 - Logging into Docker Hub...
 - Login Succeeded!
- Build and push backend
 - Run docker/build-push-action@v4
 - GitHub Actions runtime token ACs
 - Docker info
 - Proxy configuration
 - Buildx version
 - /usr/bin/docker buildx build --iidfile /tmp/docker-actions-toolkit-cQbar1/iidfile --tag ***/scd-backend --metadata-file /tmp/docker-actions-toolkit-cQbar1/metadata-file --push ./Backend
 - #0 building with "default" instance using docker driver
 - #1 [internal] load build definition from Dockerfile
 - #1 transferring dockerfile: 147B done
 - #1 DONE 0.0s
 - #2 [internal] load metadata for docker.io/library/node:latest
 - #2 ...
 - #3 [auth] library/node:pull token for registry-1.docker.io

Started 7m 7s ago

- Build and push frontend
 - Run docker/build-push-action@v4
 - GitHub Actions runtime token ACs
 - Docker info
 - Proxy configuration
 - Buildx version
 - /usr/bin/docker buildx build --iidfile /tmp/docker-actions-toolkit-kKYFEj/iidfile --tag ***/scd-frontend --metadata-file /tmp/docker-actions-toolkit-kKYFEj/metadata-file --push ./Frontend
 - #0 building with "default" instance using docker driver
 - #1 [internal] load build definition from Dockerfile
 - #1 transferring dockerfile: 435B done
 - #1 DONE 0.0s
 - #2 [internal] load metadata for docker.io/library/nginx:alpine
 - #2 DONE 0.0s
 - #3 [internal] load metadata for docker.io/library/node:latest
 - #3 DONE 0.8s
 - #4 [internal] load .dockerrcignore
 - #4 transferring context: 99B done
 - #4 DONE 0.0s
 - #5 [build 1/6] FROM docker.io/library/node:latest@sha256:149a0b6925212aa032160fe556ea5c10963ccfbe51f4af154ce50e39783bde00
 - #5 DONE 0.0s
 - #6 [stage-1 1/2] FROM docker.io/library/nginx:alpine
 - #6 DONE 0.0s
 - #7 [internal] load build context
 - #7 transferring context: 722.47kB 0.0s done
 - #7 DONE 0.0s
 - #8 [build 2/6] WORKDIR /app
 - #8 CACHED
 - #9 [build 3/6] COPY package*.json ./
 - #9 CACHED

```
Succeeded (Run in 7m 47s)

Build and push frontend
207 #16 pushing layer c56f134d3805 18.5s done
208 #16 pushing layer d71eae0084c1 18.5s done
209 #16 DONE 19.3s
210 ▶ ImageID
212 ▶ Digest
214 ▶ Metadata

Post Build and push frontend
0s
1 Post job cleanup.
2 ▶ Removing temp folder /tmp/docker-actions-toolkit-kKYFEj

Post Build and push backend
0s
1 Post job cleanup.
2 ▶ Removing temp folder /tmp/docker-actions-toolkit-cQbar1

Post Log in to Docker Hub
1s
1 Post job cleanup.
2 /usr/bin/docker logout
3 Removing login credentials for https://index.docker.io/v1/

Post Checkout code
0s
1 Post job cleanup.
2 /usr/bin/git version
3 git version 2.43.0
4 Copying '/home/moz/.gitconfig' to '/home/moz/Desktop/22I - 2434/Semester 6/scd/project/app/actions-runner/_work/_temp/5ebd34f3-bbc7-45a3-9cf3-f57c21abe02a/.gitconfig'
```

```
Succeeded (Run in 7m 47s)

Post Log in to Docker Hub
1s
1 Post job cleanup.
2 /usr/bin/docker logout
3 Removing login credentials for https://index.docker.io/v1/

Post Checkout code
0s
1 Post job cleanup.
2 /usr/bin/git version
3 git version 2.43.0
4 Copying '/home/moz/.gitconfig' to '/home/moz/Desktop/22I - 2434/Semester 6/scd/project/app/actions-runner/_work/_temp/5ebd34f3-bbc7-45a3-9cf3-f57c21abe02a/.gitconfig'
5 Temporarily overriding HOME='/home/moz/Desktop/22I - 2434/Semester 6/scd/project/app/actions-runner/_work/_temp/5ebd34f3-bbc7-45a3-9cf3-f57c21abe02a' before making global git config changes
6 Adding repository directory to the temporary git global config as a safe directory
7 /usr/bin/git config --global --add safe.directory /home/moz/Desktop/22I - 2434/Semester 6/scd/project/app/actions-runner/_work/_RecruitFlow/RecruitFlow
8 /usr/bin/git config --local --name-only --get-regexp core\\.sshCommand
9 /usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'core\\.sshCommand' && git config --local --unset-all 'core.sshCommand' || :"
10 /usr/bin/git config --local --name-only --get-regexp http\\.https:\\/\\/github\\.com\\/\\.extraheader
11 http\\.https://github\\.com/.extraheader
12 /usr/bin/git config --local --unset-all http\\.https://github\\.com/.extraheader
13 /usr/bin/git submodule foreach --recursive sh -c "git config --local --name-only --get-regexp 'http\\.https:\\/\\/github\\.com\\/\\.extraheader' && git config --local --unset-all 'http\\.https://github\\.com/.extraheader' || :"

Complete job
0s
```

9. Trigger Local Deployment

Check the containers in the minikube as the docker environment is linked with it so running `docker ps` gives us the k8s containers attached

```
[~/Desktop/223 - 2434/Semester 6/scd-project/app/actions-runner] moz ➜ main ➔ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
46715cb1881d        masteroz/scd-frontend   "/docker-entrypoint_."   33 minutes ago    Up 33 minutes      "k8s_frontend_frontend-7c9f7d468d-4vxjj_scd-project_fba5d19e-e93-4bd-8d10-10cf7a3e0ef5_0
a63a666cb7d4        masteroz/scd-backend    "/docker-entrypoint_."   33 minutes ago    Up 33 minutes      "k8s_backend_backend-857877d7b6-d1561_scd-project_c885e095-5ebd-4307-98c9-30b68363b39a_0
8e487306cb786       6e3fb40d628d          "/storage-provisioner"   36 minutes ago    Up 36 minutes      "k8s_storage-provisioner_storage-provisioner_kube-system_15cbf896-a4aa-4973-8302-f2f22fb9feed_3
21fe1f2678bc        registry.k8s.io/pause:3.10  "/pause"           37 minutes ago    Up 37 minutes      "k8s_POD_frontend-7c9f7d468d-4vxjj_scd-project_fba5d19e-e93-4bd-8d10-10cf7a3e0ef5_0
6f5c639e21bf        registry.k8s.io/pause:3.10  "/pause"           37 minutes ago    Up 37 minutes      "k8s_POD_backend-857877d7b6-d1561_scd-project_c885e095-5ebd-4307-98c9-30b68363b39a_0
fe744fd3a35         c69fa2e9cbf5          "/coredns -conf /etc_."  37 minutes ago    Up 37 minutes      "k8s_coredns_coredns-6869bf9bc-4b7m7_kube-system_e6cebbda-74fe-4d59-b369-152f2b0d4e80_1
4d4398d4d927        840ff1fb8ac8c          "/usr/local/bin/kube_."  37 minutes ago    Up 37 minutes      "k8s_kube-proxy_kube-proxy-7npb_kube-system_a5d52098-82f1-4eb9-9059-3068dbf92ea3_1
de090976e09b        registry.k8s.io/pause:3.10  "/pause"           37 minutes ago    Up 37 minutes      "k8s_POD_coredns-6609888-444f_kube-system_d14ce008be3a1f3bd7f547688f9dfe_1
f0cc3ec308f         registry.k8s.io/pause:3.10  "/pause"           37 minutes ago    Up 37 minutes      "k8s_POD_kube-scheduler-minikube_kube-system_15cbf896-a4aa-4973-8302-f2f22fb9feed_1
aacc5d4e639         registry.k8s.io/pause:3.10  "/pause"           37 minutes ago    Up 37 minutes      "k8s_POD_storage-provisioner_kube-system_15cbf896-a4aa-4973-8302-f2f22fb9feed_1
94bed2b6e92         a389e107f4ff          "kube-scheduler --au_."  37 minutes ago    Up 37 minutes      "k8s_kube-scheduler_kube-scheduler-minikube_kube-system_d14ce008be3a1f3bd7f547688f9dfe_1
c5adda3f3fb8        8cab3d2a8b0d8          "kube-controller-man_."  37 minutes ago    Up 37 minutes      "k8s_kube-controller-manager_kube-controller-manager-minikube_kube-system_843c74f7b3bc7d704a05c31708a6
a30_1
77780cc5b1cb        a97e6b294ba          "etcd - advertise-cl_."  37 minutes ago    Up 37 minutes      "k8s_etcd_etcd-minikube_kube-system_24b075c2a289008eb381891e9683840_1
18ef76890c00        c2e17b0d0f4a          "kube-apiserver --ad_."  37 minutes ago    Up 37 minutes      "k8s_kube-apiserver_kube-apiserver-minikube_kube-system_d7209a4cf4be077c9919d46b7358a5e8_1
9e57b111635d        registry.k8s.io/pause:3.10  "/pause"           37 minutes ago    Up 37 minutes      "k8s_POD_etcd-minikube_kube-system_2b4b75c2a289008eb381891e9683840_1
ab7a7475ea1d        registry.k8s.io/pause:3.10  "/pause"           37 minutes ago    Up 37 minutes      "k8s_POD_kube-scheduler-minikube_kube-system_d14ce008be3a1f3bd7f547688f9dfe_1
27ba990fd9ad        registry.k8s.io/pause:3.10  "/pause"           37 minutes ago    Up 37 minutes      "k8s_POD_kube-apiserver-minikube_kube-system_d7209a4cf4be077c9919d46b7358a5e8_1
ef97b6d7c62e        registry.k8s.io/pause:3.10  "/pause"           37 minutes ago    Up 37 minutes      "k8s_POD_kube-controller-manager-minikube_kube-system_843c74f7b3bc7d704a05c31708a6_1
[~/Desktop/223 - 2434/Semester 6/scd-project/app/actions-runner] moz ➜ main
```

Verify the Deployment

Use kubectl to get and display all the resources information here for eg; pods, services and deployment.

```
[~/Desktop/223 - 2434/Semester 6/scd-project/app/actions-runner] moz ➜ main ➔ kubectl get pods -n scd-project
NAME            READY   STATUS    RESTARTS   AGE
backend-857877d7b6-d1561   1/1     Running   0          38m
frontend-7c9f7d468d-4vxjj  1/1     Running   0          38m

[~/Desktop/223 - 2434/Semester 6/scd-project/app/actions-runner] moz ➜ main ➔ kubectl get services -n scd-project
NAME            TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
backend          ClusterIP  10.110.241.233  <none>        5000/TCP   38m
Frontend         NodePort   10.103.74.151   <none>        80:32665/TCP   38m
mongodb          ClusterIP  10.100.132.229  <none>        27017/TCP   38m
scd-backend      ClusterIP  10.108.66.87   <none>        5000/TCP   38m

[~/Desktop/223 - 2434/Semester 6/scd-project/app/actions-runner] moz ➜ main ➔ kubectl get deployments -n scd-project
NAME            READY   UP-TO-DATE   AVAILABLE   AGE
backend          1/1     1           1           38m
Frontend         1/1     1           1           38m

[~/Desktop/223 - 2434/Semester 6/scd-project/app/actions-runner] moz ➜ main ➔ kubectl get namespaces -n scd-project
NAME          STATUS   AGE
default        Active   3h26m
kube-node-lease Active   3h26m
kube-public    Active   3h26m
kube-system    Active   3h26m
scd-project   Active   155m

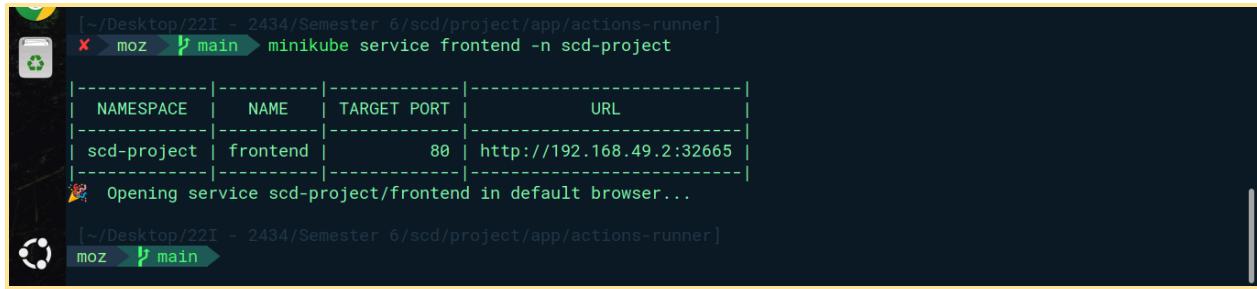
[~/Desktop/223 - 2434/Semester 6/scd-project/app/actions-runner] moz ➜ main ➔ kubectl get -n scd-project
```

While displaying the resources, we made sure that our namespace is `scd-project` and not `default`. Create a new namespace to manage your resources.

Access the Application

Using Minikube to access the service:

```
minikube service frontend -n scd-project
```



NAMESPACE	NAME	TARGET PORT	URL
scd-project	frontend	80	http://192.168.49.2:32665

Opening service scd-project/frontend in default browser...

This opened the application in our default browser as it's a NodePort servie using port .

Project Structure

```
moz > main > tree -L 3
```

```
. └── app
    ├── actions-runner
    │   ├── actions-runner-linux-x64-2.323.0.tar.gz
    │   ├── bin
    │   ├── config.sh
    │   ├── _diag
    │   ├── env.sh
    │   ├── externals
    │   ├── run-helper.cmd.template
    │   ├── run-helper.sh
    │   ├── run-helper.sh.template
    │   ├── run.sh
    │   ├── safe_sleep.sh
    │   ├── svc.sh
    │   └── _work
    ├── Backend
    │   ├── Dockerfile
    │   ├── node_modules
    │   ├── package.json
    │   ├── package-lock.json
    │   ├── src
    │   └── uploads
    ├── Documents
    │   └── Recruit_Flow_Project_Proposal.pdf
    ├── Frontend
    │   ├── dist
    │   ├── Dockerfile
    │   ├── eslint.config.js
    │   ├── index.html
    │   ├── node_modules
    │   ├── package.json
    │   ├── package-lock.json
    │   ├── postcss.config.cjs
    │   ├── public
    │   ├── README.md
    │   ├── sample.env
    │   ├── src
    │   └── vite.config.js
    └── k8s
        ├── backend-deployment.yaml
        ├── backend-service.yaml
        ├── frontend-deployment.yaml
        ├── frontend-service.yaml
        ├── mongo-deployment.yaml
        └── mongo-service.yaml
    README.md
```

Operating System Selection Requirement – Important

- OS and setup.

```
[~/Desktop/22I - 2434/Semester 6/scd/project]
moz ➤ main ➤ neofetch --ascii
      .-/+oossssoo+/-.
      `:+ssssssssssssssssssss+:`+
      -+ssssssssssssssssssyyssss+-+
      .osssssssssssssssssdMMMNyssso.
      /sssssssssssshdmmNNmmyNMMMHhssssss/
      +ssssssssssshyddMMMMMMMdffffyssssssss+
      =/sssssssshhNMMyhyyyyhmNMMMNhssssssss/
      .ssssssssdMMMNhsssssssssshNMMMdssssssss.
      +sssshhhyNMMNyssssssssssssyNMMMyssssssss+
      osyNMMMNyMMhssssssssssssshmmmhssssssso
      osyNMMMNyMMhssssssssssssssshmmmhssssssso
      +sssshhhyNMMNyssssssssssssyNMMMyssssssss+
      .ssssssssdMMMNhsssssssssshNMMMdssssssss.
      =/sssssssshhNMMyhyyyyhdNMMMNhssssssss/
      +ssssssssdmydMMMMMMMdffffyssssssss+
      /sssssssssssshdmmNNNmyNMMMHhssssssss/
      .osssssssssssssssssdMMMNyssso.
      -+ssssssssssssssssyyssss+-+
      `:+ssssssssssssssssss+:`+
      .-/+oossssoo+/-.

[~/Desktop/22I - 2434/Semester 6/scd/project]
moz ➤ main ➤ ==
```

moz@Spa-Of-Master0z

OS: Ubuntu 24.04.1 LTS x86_64
Host: 80TV Lenovo ideapad 310-15IKB
Kernel: 6.11.0-25-generic
Uptime: 56 mins
Packages: 1829 (dpkg), 12 (snap)
Shell: zsh 5.9
Resolution: 1366x768
DE: GNOME 46.0
WM: Mutter
WM Theme: Adwaita
Theme: Yaru [GTK2/3]
Icons: Yaru [GTK2/3]
Terminal: gnome-terminal
CPU: Intel i5-7200U (4) @ 3.100GHz
GPU: Intel HD Graphics 620
Memory: 7490MiB / 11861MiB



Versions

- **Git:** 2.43.0
 - **Actions Runner:** 2.323.0
 - **Docker:** 28.1.1 Build 4eba377
 - **Kubernetes:**
 - Client: v1.33.0
 - Kustomize: v5.6.0
 - Server: v1.32.0
 - **Minikube:** v1.35.0
-
- **Advantages and Challenges**

Got to learn about minikube, had previous grip on Docker and on Github so it was easier to setup runners and other tokens easily.

For the challenges, there was a bit of a permissions error with Firefox dependencies, so we couldn't get it working for the time being. A better fix was to switch the browser so using **Chrome** worked like a charm as it somehow was a trusted source and not using overly-sensitive dependencies.

- **Affect on Development and Deployment**

We got to experience the networking (hiding services (backend and mongo) from public access and exposing services (e.g. front-end) using NodePort

JazakAllah Sir!
Geo Hazaro Saal!