

DOCUMENTATION

Step 1: launch the server

Then connect in mobaextrem

And ssh it

Install docker

cat ss.sh

Add Docker's official GPG key:

sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc

Add the repository to Apt sources:

echo \

"deb [arch=\$(dpkg --print-architecture)

signed-by=/etc/apt/keyrings/docker.asc]

https://download.docker.com/linux/ubuntu \

\$(. /etc/os-release && echo "\$VERSION_CODENAME") stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

And clone both frontend and backend repos

git clone <https://github.com/Anand-1432/Techdome-frontend.git>

git clone <https://github.com/Anand-1432/Techdome-backend.git>

Step 2 : create docker file for both frontend and backend

For frontend

```
azureuser@Demo:~/Techdome-frontend$  
azureuser@Demo:~/Techdome-frontend$ ls  
Dockerfile README.md package-lock.json package.json public src ss.sh  
azureuser@Demo:~/Techdome-frontend$ cat Dockerfile  
# Stage 1: Build  
FROM node:16 AS build  
  
# Set the working directory in the container  
WORKDIR /app  
  
# Copy package.json and package-lock.json  
COPY package.json ./  
  
# Install dependencies  
RUN npm install  
  
# Copy the entire project directory to the working directory  
COPY . .  
  
# Build the React app  
RUN npm run build  
  
# Stage 2: Serve the built app using a lightweight web server  
FROM nginx:alpine  
  
# Copy the build output from the previous stage to the nginx HTML directory  
COPY --from=build /app/build /usr/share/nginx/html  
  
# Expose port 80  
EXPOSE 80  
  
# Start nginx  
CMD ["nginx", "-g", "daemon off;"]  
azureuser@Demo:~/Techdome-frontend$
```

Build and run it

sudo apt-get install docker-ce docker-ce-cli containerd.io

docker-buildx-plugin docker-compose-plugin

45 sudo docker run hello-world

46 docker build -t react-nginx .

47 sudo usermod -aG docker \$USER

48 docker build -t react-nginx .

49 sudo docker build -t react-nginx .

50 docker run -d -p 80:80 react-nginx

51 sudo docker run -d -p 80:80 react-nginx

For backend

```
azureuser@Demo:~/Techdome-backend$ ls
Dockerfile  controller  database  models  node_modules  package-lock.json  package.json  routes  server.js
azureuser@Demo:~/Techdome-backend$ cat Dockerfile
# Use the official Node.js 18 LTS image as the base
FROM node:18-alpine

# Set the working directory inside the container
WORKDIR /app

# Copy package.json and package-lock.json to the working directory
COPY package*.json ./

# Install dependencies
RUN npm install

# Copy the rest of the application code
COPY . .

# Expose the port your application will run on
EXPOSE 3000

# Define the command to start the application
CMD ["node", "server.js"]

azureuser@Demo:~/Techdome-backend$
```

Same for this

```
docker build -t backend-app1 .
```

```
78 docker ps
```

```
79 docker ps -a
```

```
80 docker rm -f 40 fa 73
```

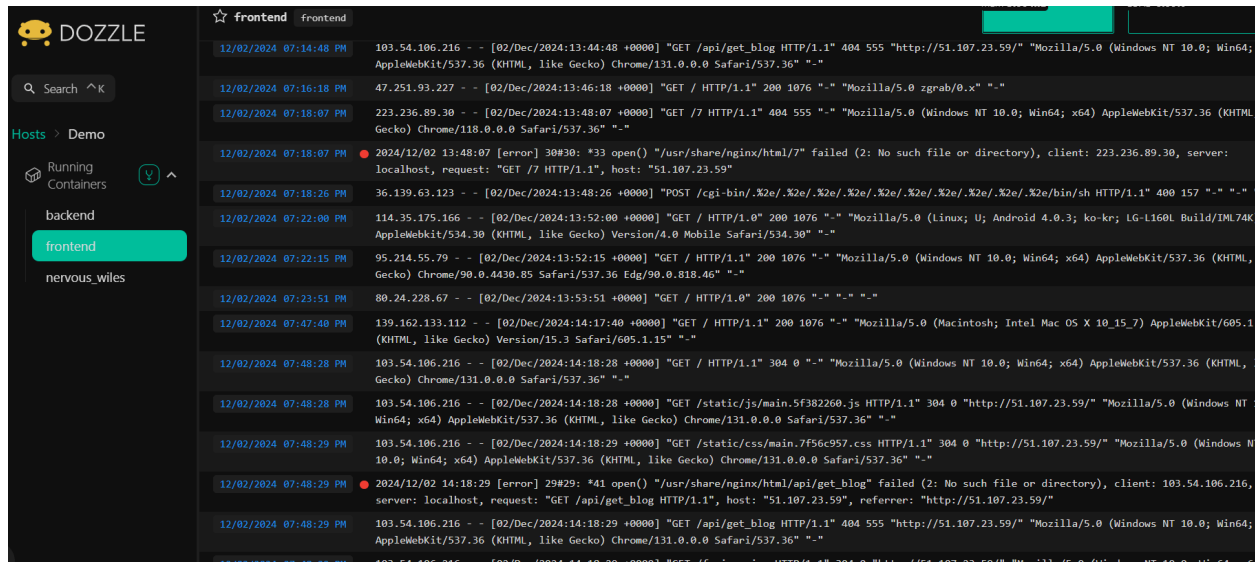
```
81 docker ps
```

```
82 docker run -p 3000:3000 backend-app
```

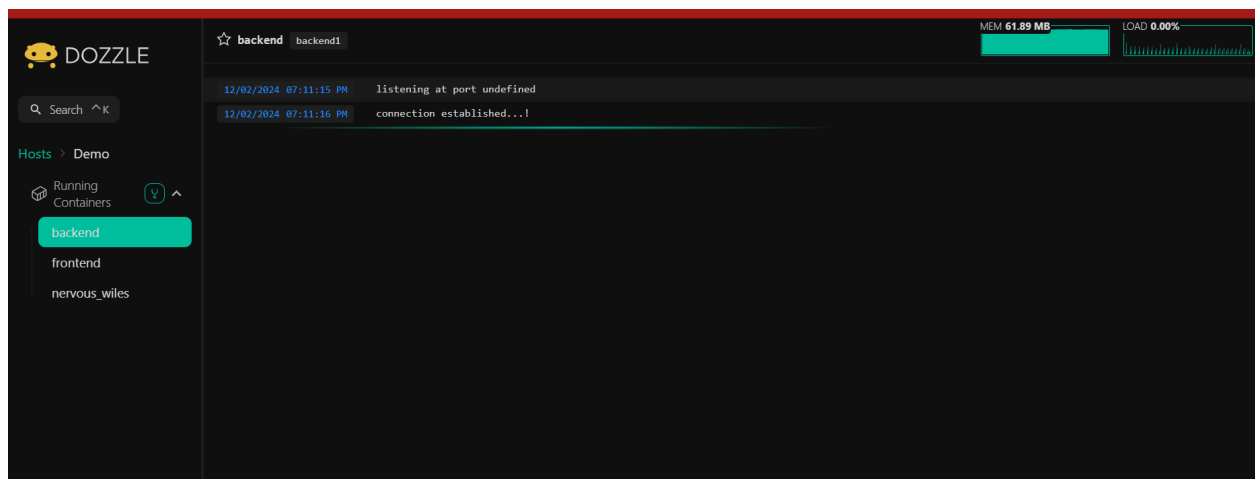
```
83 docker run -d -p 3000:3000 backend-app1
```

Step 3 : create dozzle for monitoring for frontend and backend

For frontend



For backend



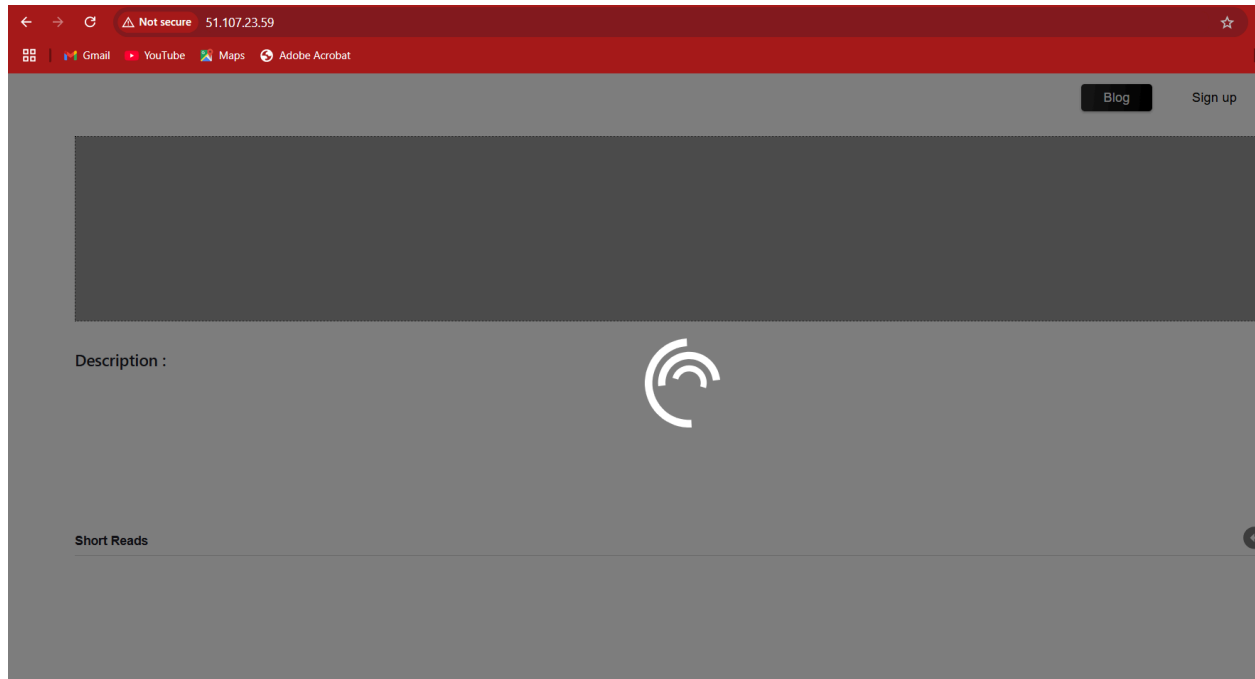
Command for this

sudo docker images 108 docker ps

109 docker run -d -v /var/run/docker.sock:/var/run/docker.sock -p 8080:8080 amir20/dozzle:latest

110 sudo docker ps

Step 4 : check the deployment



Step 5 : now push this all to github

git remote add origin

https://github.com/mathematicsgk/Tech_dome-.git

188 git remote -v

189 git init

190 git add .

191 git push

192 git push origin main

193 git pull origin main

194 git add .

195 git commit -m "add"

196 git push origin main

197 git pull origin main --rebase

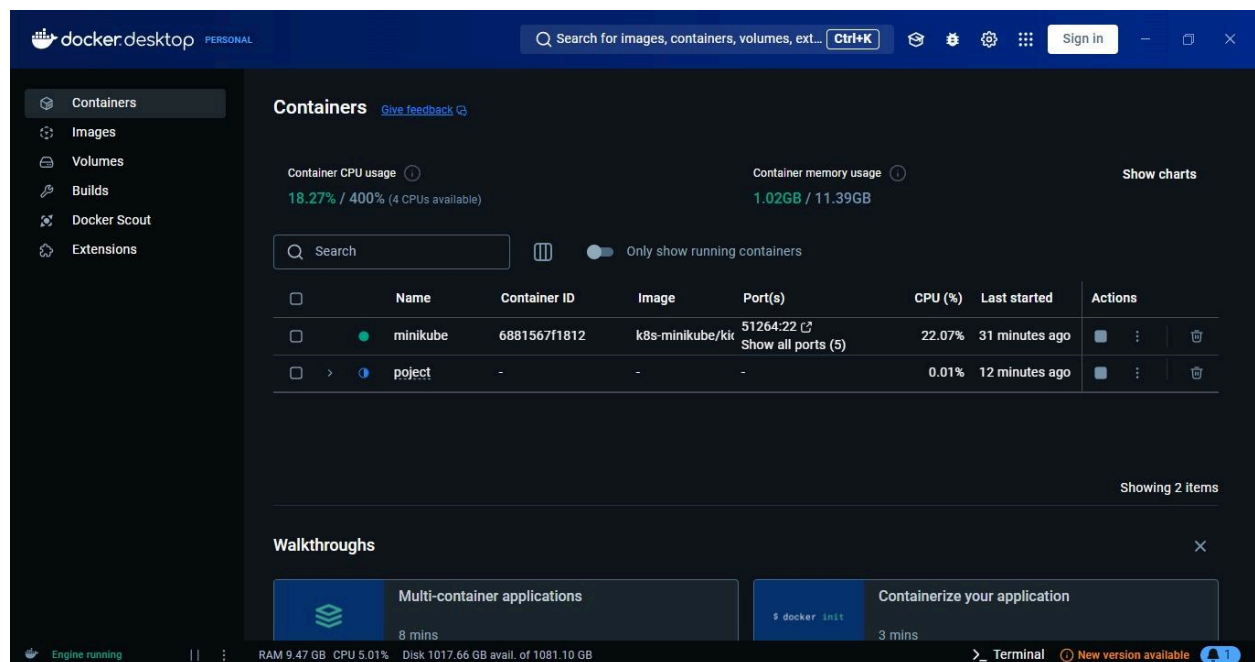
198 git branch

```
199 git checkout main
200 git add README.md
201 git rebase --continue
202 git push origin main --force
```

Issues faced

Method1: In System:windows

Firstly I tried docker software and minikube software in my system so out of 3 containers two container is created but one is running and another is completed and got error in port service.



```
Windows PowerShell
Container frontend Created 0.0s
Container database Created 0.0s
Attaching to backend, database, frontend
database | PostgreSQL Database directory appears to contain a database; Skipping initialization
database |
database | python: can't open file 'app.py': [Errno 2] No such file or directory
backend | 2024-11-30 17:45:02.251 UTC [1] LOG: starting PostgreSQL 13.18 (Debian 13.18-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
database | 2024-11-30 17:45:02.251 UTC [1] LOG: listening on IPv4 address "0.0.0.0", port 5432
database | 2024-11-30 17:45:02.251 UTC [1] LOG: listening on IPv6 address ":::", port 5432
database | 2024-11-30 17:45:02.271 UTC [1] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
database | 2024-11-30 17:45:02.284 UTC [27] LOG: database system was shut down at 2024-11-30 17:44:56 UTC
database | 2024-11-30 17:45:02.294 UTC [1] LOG: database system is ready to accept connections
frontend | > blog_website@0.1.0 start /app
frontend | > react-scripts start
frontend |
frontend | sh: 1: react-scripts: not found
frontend | npm ERR! code ELIFECYCLE
frontend | npm ERR! syscall spawn
frontend | npm ERR! file sh
frontend | npm ERR! errno ENOENT
frontend | npm ERR! blog_website@0.1.0 start: 'react-scripts start'
frontend | npm ERR! spawn ENOENT
frontend | npm ERR!
frontend | npm ERR! Failed at the blog_website@0.1.0 start script.
frontend | npm ERR! This is probably not a problem with npm. There is likely additional logging output above.
frontend | npm WARN Local package.json exists, but node_modules missing, did you mean to install?
frontend |
frontend | npm ERR! A complete log of this run can be found in:
frontend | npm ERR! /root/.npm/_logs/2024-11-30T17_45_02_476Z-debug.log
backend exited with code 2
frontend exited with code 1
```

```
Windows PowerShell
frontend-deployment-5b9d8c5dfc-sztnk 0/1 Completed 5 (95s ago) 4m
PS C:\Users\ABCD\Downloads\project\k8s>

PS C:\Users\ABCD\Downloads\project\k8s> kubectl logs frontend-deployment-5b9d8c5dfc-fmgw7
PS C:\Users\ABCD\Downloads\project\k8s> kubectl logs backend-deployment-5d975f5f49-4dplkf
PS C:\Users\ABCD\Downloads\project\k8s> kubectl describe frontend-deployment-5b9d8c5dfc-fmgw7
error: unknown command "describe" for "kubectl"

Did you mean this?
describe
PS C:\Users\ABCD\Downloads\project\k8s> kubectl describe frontend-deployment-5b9d8c5dfc-fmgw7
error: the server doesn't have a resource type "frontend-deployment-5b9d8c5dfc-fmgw7"
PS C:\Users\ABCD\Downloads\project\k8s> kubectl get pods
NAME                                READY    STATUS              RESTARTS   AGE
backend-deployment-5d975f5f49-4dplkf 0/1      CrashLoopBackOff    6           21s
backend-deployment-5d975f5f49-86j9v 0/1      Completed            6           3m2s
database-deployment-5cb646466d-x2s2v 1/1      Running              0           6m53s
frontend-deployment-5b9d8c5dfc-fmgw7 0/1      CrashLoopBackOff    6           72s
frontend-deployment-5b9d8c5dfc-sztnk 0/1      CrashLoopBackOff    6           87s

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ABCD\Downloads\project> docker-compose up --build
time="2024-11-30T23:15:01+05:30" level=warning msg="C:\\Users\\ABCD\\Downloads\\project\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 3/0
Container backend Created 0.0s
Container frontend Created 0.0s
Container database Created 0.0s
Attaching to backend, database, frontend
database | PostgreSQL Database directory appears to contain a database; Skipping initialization
database |
database | python: can't open file 'app.py': [Errno 2] No such file or directory
backend | 2024-11-30 17:45:02.251 UTC [1] LOG: starting PostgreSQL 13.18 (Debian 13.18-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
database | 2024-11-30 17:45:02.251 UTC [1] LOG: listening on IPv4 address "0.0.0.0", port 5432
database | 2024-11-30 17:45:02.251 UTC [1] LOG: listening on IPv6 address ":::", port 5432
database | 2024-11-30 17:45:02.271 UTC [1] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
database | 2024-11-30 17:45:02.284 UTC [27] LOG: database system was s
```

Method2: Via Azure Virtual Machine

Frontend not connecting backend due to database connectivity issue.

Remark:

Source of learning and deployment done via google and online. And I am new learner in devops but I know about Azure services and Ethical hacking ,iot and cybersecurity and I try my best .And my semester exam running so that's why I am late.

Thanksyou