

The screenshot shows a terminal window with the following content:

```
FROM node:18-alpine
WORKDIR /app
COPY package*.json .
RUN npm install
COPY . .
EXPOSE 3000
```

TERMINAL TAB

```
varsh@ubs2404vm:~$ git clone https://github.com/jagadeeshkanna97/docker_sample.git
Cloning into 'docker_sample'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (7/7), done.
varsh@ubs2404vm:~$ ls
docker_sample
varsh@ubs2404vm:~$ cd newdock/
bash: cd: newdock/: No such file or directory
varsh@ubs2404vm:~$ cd docker_sample
ls
app.js package.json README.md
varsh@ubs2404vm:~/docker_sample$ touch Dockerfile
varsh@ubs2404vm:~/docker_sample$ docker build -t docker-sample-app .
[+] Building 61.2s (10/10) FINISHED docker:default
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 158B 0.0s
=> [internal] load metadata for docker.io/library/node 6.9s
```

← → ⌂ ⚠ Not secure 192.168.117.128:3000

Hello from Docker!

```

PROBLEMS OUTPUT TERMINAL PORTS
varsh@ubs2404vm:~/docker_sample$ docker ps
ermined_engelbart

● varsh@ubs2404vm:~/docker_sample$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
Digest: sha256:341bf0f3ce6c5277d6002cf6e1fb0319fa4252add24ab6a0e262e0056d313208
Status: Image is up to date for nginx:latest
docker.io/library/nginx:latest
● varsh@ubs2404vm:~/docker_sample$ docker run -d -p 8080:80 nginx
5d2a4b0b161e3ff76b2946874ca7c3e7c083d85e40ad3a69f8cc58264adc5f97

```

← → ⌂ △ Not secure 192.168.117.128:8080

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

```

● varsh@ubs2404vm:~/docker_sample$ mkdir mynginx
● varsh@ubs2404vm:~/docker_sample$ cd mynginx
● varsh@ubs2404vm:~/docker_sample/mynginx$ nano index.html
● varsh@ubs2404vm:~/docker_sample/mynginx$ docker run -d -p 8080:80 \
-v $(pwd)/index.html:/usr/share/nginx/html/index.html \
nginx
9f993f1a69b5f7e7488c1c6f02b862b89bfb175fc4d9aa8136b85126ff62bab
docker: Error response from daemon: failed to set up container networking: driver failed programming external connectivity on endpoint hopefu
ul_keldysh (bebdd56b8d311234da4149d4d5a6eb126b5eac59fe6366d61e075de9333780d6): Bind for 0.0.0.0:8080 failed: port is already allocated
Run 'docker run --help' for more information
● varsh@ubs2404vm:~/docker_sample/mynginx$ docker run -d -p 7070:80 -v $(pwd)/index.html:/usr/share/nginx/html/index.html nginx
5be4b5f077f25424c050528978b2aab9570ad9d1a4b2309097823ac6b0338a3
● varsh@ubs2404vm:~/docker_sample/mynginx$ 

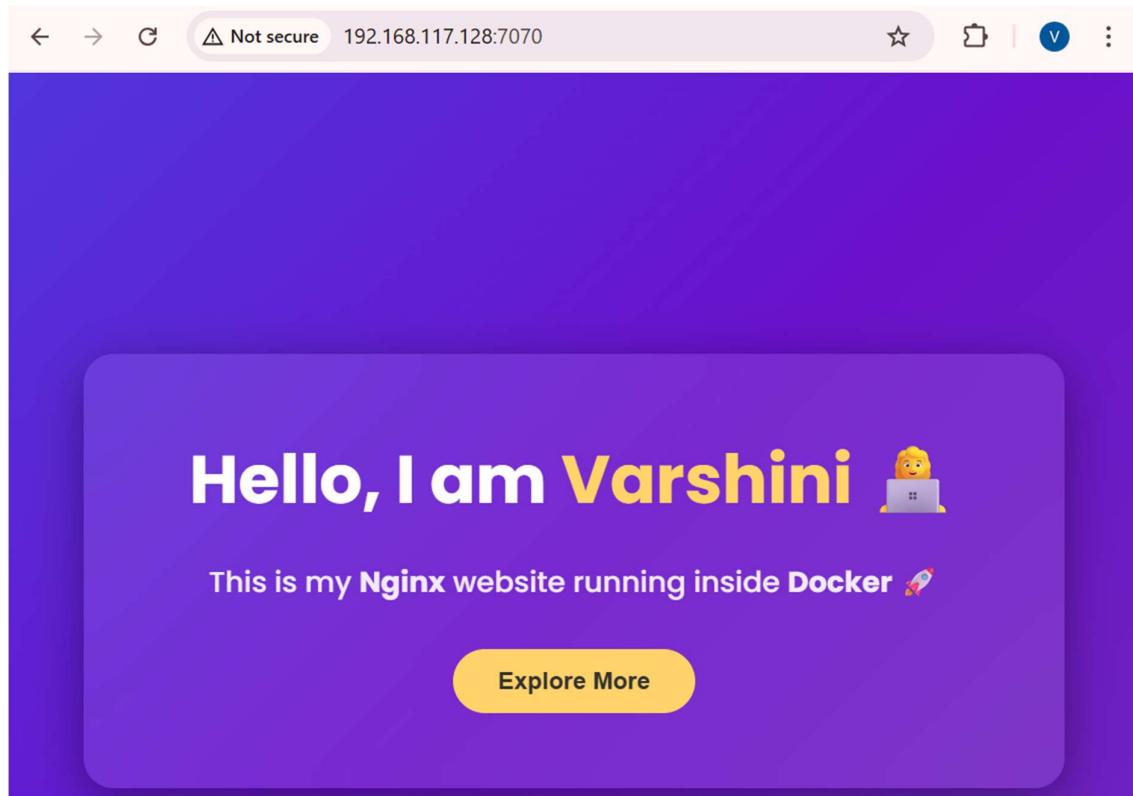
```

```

TX packets 12 bytes 1368 (1.3 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

varsh@ubs2404vm:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
5be4b5f077f2 nginx "/docker-entrypoint..." Less than a second ago Up Less than a second 0.0.0.0:7070->80/tcp, [::]::
0us_almeida 5d2a4b0b161e nginx "/docker-entrypoint..." Less than a second ago Up Less than a second 0.0.0.0:8080->80/tcp, [::]::
g_carson 354f99dbe5d5 docker-sample-app "docker-entrypoint.s..." Less than a second ago Up Less than a second 0.0.0.0:3000->3000/tcp, [::]::
ermined_engelbart
varsh@ubs2404vm:~$ 

```



```
varsh@ubs2404vm:~$ docker images

IMAGE          ID          DISK USAGE   CONTENT SIZE   EXTRA
docker-sample-app:latest  162512dd297a    196MB        47.3MB      U
hello-world:latest       05813aedc15f    25.9kB        9.52kB      U
nginx:latest            341bf0f3ce6c    240MB        65.8MB      U
ubuntu:latest           cd1dba651b30    119MB        31.7MB      U
varsh@ubs2404vm:~$ _
```

To install Ubuntu Server in VMware, configure SSH, install Docker, integrate GitHub using SSH authentication, build Docker images, and deploy containerized applications.

## REQUIREMENTS

- Windows System
- VMware Workstation Player
- Ubuntu Server ISO
- Internet Connection
- GitHub Account
- VS Code

## VMWARE SETUP

1. Install VMware Workstation Player.
2. Create a new Virtual Machine.
3. Allocate 4GB RAM and 20GB Disk.
4. Select NAT Network.
5. Install Ubuntu Server and login.

## UBUNTU INITIAL CONFIGURATION

```
sudo apt update  
sudo apt upgrade -y  
ip a  
  
sudo apt install openssh-server -y  
sudo systemctl start ssh  
sudo systemctl enable ssh  
sudo ufw allow ssh  
sudo ufw reload
```

## SSH CONNECTION

```
ssh username@VM_IP  
hostname
```

## DOCKER INSTALLATION

```
sudo apt install ca-certificates curl -y  
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  
sudo apt update  
sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin -y  
sudo systemctl start docker  
sudo systemctl enable docker  
sudo usermod -aG docker $USER  
newgrp docker  
docker --version  
docker ps
```

## DOCKER PRACTICAL EXECUTION

```
git clone https://github.com/jagadeeshkanna97/docker_sample.git  
cd docker_sample  
touch Dockerfile  
docker build -t docker-sample-app .  
docker run -d -p 3000:3000 docker-sample-app  
docker pull nginx  
docker run -d -p 8080:80 nginx
```

Custom Nginx:

```
mkdir mynginx  
cd mynginx  
nano index.html  
docker run -d -p 7070:80 -v $(pwd)/index.html:/usr/share/nginx/html/index.html nginx
```

## GIT & GITHUB CONFIGURATION

```
git --version  
git config --global user.name "VARSHINI1805"  
git config --global user.email "varshini.sureshkumar224@gmail.com"  
git config --list  
ssh-keygen -t ed25519 -C "varshini.sureshkumar224@gmail.com"  
cat ~/.ssh/id_ed25519.pub  
ssh -T git@github.com
```

## BRANCH MANAGEMENT

```
git clone git@github.com:VARSHINI1805/devops.git  
cd devops  
git branch  
git checkout -b day3  
touch day3.txt  
git add .  
git commit -m "Added day3 file"  
git push -u origin day3
```

## MERGE PROCESS

```
git checkout main  
git pull origin main  
git merge day3  
git push
```

## ADD DOCKER PROJECT TO REPOSITORY

```
cp -r ~/docker_sample .  
rm -rf docker_sample/.git  
git add .  
git commit -m "Added docker_sample project"  
git push
```

```
varsh@ubs2404vm:~$ history  
1 hostname  
2 ssh varsh@192.168.117.128  
3 ifconfig  
4 docker ps  
5 git clone https://github.com/jagadeeshkanna97/docker_sample.git  
6 ls  
7 cd newdock/  
8 cd docker_sample  
9 ls  
10 touch Dockerfile  
11 docker build -t docker-sample-app .  
12 docker ps  
13 docker stop <container_id>  
14 docker ps  
15 docker run -d -p 3000:3000 docker-sample-app  
16 docker ps  
17 docker pull nginx  
18 docker run -d -p 8080:80 nginx  
19 mkdir mynginx  
20 cd mynginx  
21 nano index.html  
22 docker run -d -p 8080:80 -v $(pwd)/index.html:/usr/share/nginx/html/index.html nginx  
23 docker run -d -p 7070:80 -v $(pwd)/index.html:/usr/share/nginx/html/index.html nginx  
24 git --version  
25 history
```

```
varsh@ubs2404vm:~/devops$ history
```

```
1 hostname
2 ssh varsh@192.168.117.128
3 ifconfig
4 docker ps
5 git clone https://github.com/jagadeeshkanna97/docker_sample.git
6 ls
7 cd newdock/
8 cd docker_sample
9 ls
10 touch Dockerfile
11 docker build -t docker-sample-app .
12 docker ps
13 docker stop <container_id>
14 docker ps
15 docker run -d -p 3000:3000 docker-sample-app
16 docker ps
17 docker pull nginx
18 docker run -d -p 8080:80 nginx
19 mkdir mynginx
20 cd mynginx
21 nano index.html
22 docker run -d -p 8080:80 -v $(pwd)/index.html:/usr/share/nginx/html/index.html nginx
23 docker run -d -p 7070:80 -v $(pwd)/index.html:/usr/share/nginx/html/index.html nginx
24 git --version
25 history
26 git config --global user.name "VARSHINI1805"
27 git config --global user.email "varshinisureshkumar224@gmail.com"
28 git config --list
29 ssh-keygen -t ed25519 -C "varshinisureshkumar224@gmail.com"
30 cat ~/.ssh/id_ed25519.pub
31 ssh -T git@github.com
```

```
32 git clone git@github.com:VARSHINI1805/devops.git
33 cd devops
34 ls
35 ls -a
36 git branch
37 git checkout -b day3
38 git branch
39 touch day3.txt
40 nano day3.txt
41 history
42 git add .
43 git status
44 git pull origin main
45 git push
46 git push -u origin day3
47 git checkout main
48 git pull origin main
49 git merge day3
50 git push
51 cd ~/devops
52 cp -r ~/docker_sample .
53 ls
54 git status
55 git add .
56 rm -rf docker_sample/.git
57 git add .
58 git status
59 git commit -m "Added docker_sample project"
60 git push
61 history
```