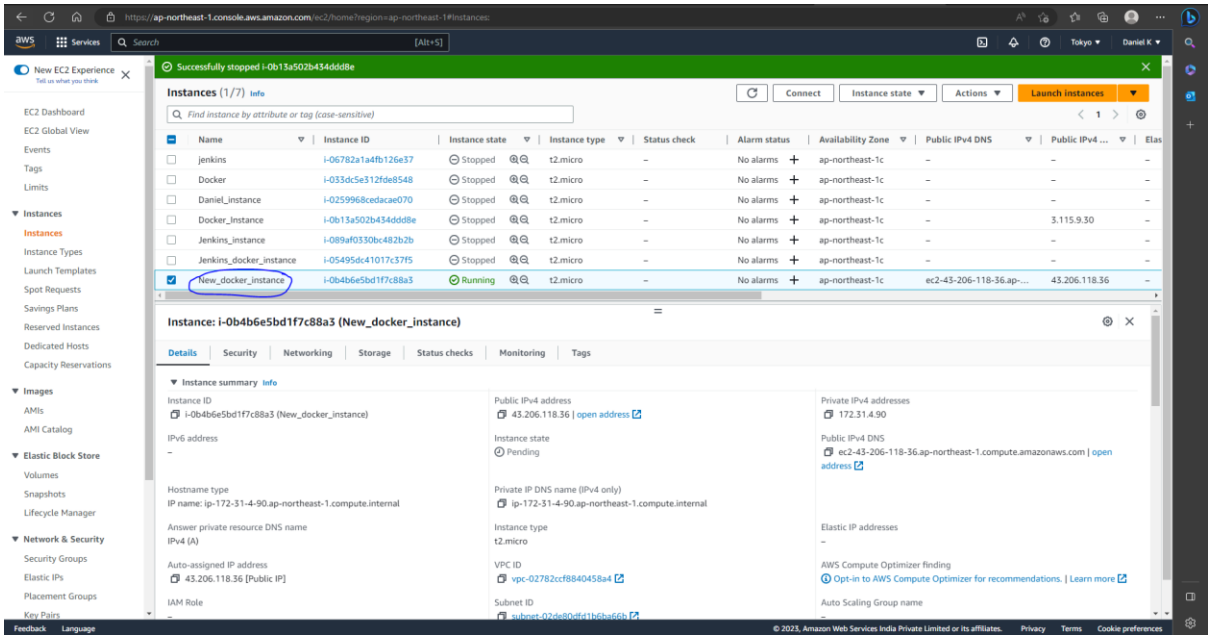
Devops Assignments - Module 6 - Advanced Docker

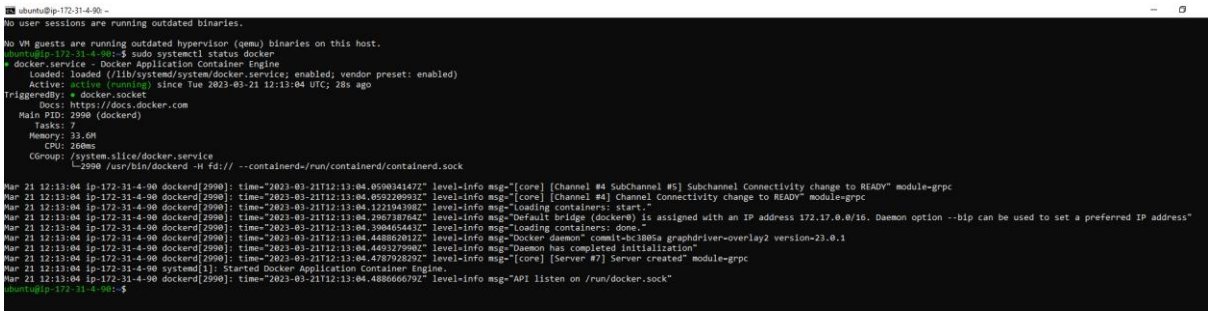
Assignment 1 - Working with Dockerfile.docx.txt

In this assignment, we need to create Apache container using Dockerfile

1. Launch an EC2 instance and Install Docker:



- Install Docker using commands
  - **sudo apt update**
  - **sudo apt install apt-transport-https ca-certificates curl software-properties-common**
  - **sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"**
  - **sudo apt install docker-ce**



2. Creating a new directory:
   - **mkdir docker_dir** → command to create a new directory

root@ip-172-31-4-90: /home/ubuntu

```
Mar 21 12:13:04 ip-172-31-4-90 dockerd[2990]: time="2023-03-21T12:13:04.059
Mar 21 12:13:04 ip-172-31-4-90 dockerd[2990]: time="2023-03-21T12:13:04.059
Mar 21 12:13:04 ip-172-31-4-90 dockerd[2990]: time="2023-03-21T12:13:04.122
Mar 21 12:13:04 ip-172-31-4-90 dockerd[2990]: time="2023-03-21T12:13:04.296
Mar 21 12:13:04 ip-172-31-4-90 dockerd[2990]: time="2023-03-21T12:13:04.390
Mar 21 12:13:04 ip-172-31-4-90 dockerd[2990]: time="2023-03-21T12:13:04.448
Mar 21 12:13:04 ip-172-31-4-90 dockerd[2990]: time="2023-03-21T12:13:04.449
Mar 21 12:13:04 ip-172-31-4-90 dockerd[2990]: time="2023-03-21T12:13:04.478
Mar 21 12:13:04 ip-172-31-4-90 systemd[1]: Started Docker Application Conta
Mar 21 12:13:04 ip-172-31-4-90 dockerd[2990]: time="2023-03-21T12:13:04.488
ubuntu@ip-172-31-4-90:~$ sudo su
root@ip-172-31-4-90:/home/ubuntu# mkdir Docker_dir
root@ip-172-31-4-90:/home/ubuntu# ls
Docker_dir
root@ip-172-31-4-90:/home/ubuntu#
```

3. Inside the new directory create a new Dockerfile with the below contents:

FROM ubuntu:18.04

MAINTAINER Chaitanya

RUN apt-get update && apt-get install -y apache2 && apt-get clean && rm -rf /var/lib/apt/lists/*

ENV APACHE_RUN_USER www-data

ENV APACHE_RUN_GROUP www-data

ENV APACHE_LOG_DIR /var/log/apache2

EXPOSE 80

CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]

root@ip-172-31-4-90: /home/ubuntu/Docker_dir

```
Mar 21 12:13:04 ip-172-31-4-90 systemd[1]: Started Docker Application Container Engine.
Mar 21 12:13:04 ip-172-31-4-90 dockerd[2990]: time="2023-03-21T12:13:04.488666679Z" level=info msg="API listen on /run/docker.sock"
ubuntu@ip-172-31-4-90:~$ sudo su
root@ip-172-31-4-90:/home/ubuntu# mkdir Docker_dir
root@ip-172-31-4-90:/home/ubuntu# ls
Docker_dir
root@ip-172-31-4-90:/home/ubuntu# cd Docker_dir
root@ip-172-31-4-90:/home/ubuntu/Docker_dir# vi Dockerfile
```

root@ip-172-31-4-90: /home/ubuntu/Docker_dir

```
FROM ubuntu:43.206.118.36
MAINTAINER Chaitanya
RUN apt-get update && apt-get install -y apache2 && apt-get clean && rm -rf /var/lib/apt/lists/*

ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2

EXPOSE 80

CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
~
~
~
~
~
```

4. Build the image ad create a container from this custom image which will allow the apache application to be reachable from internet:

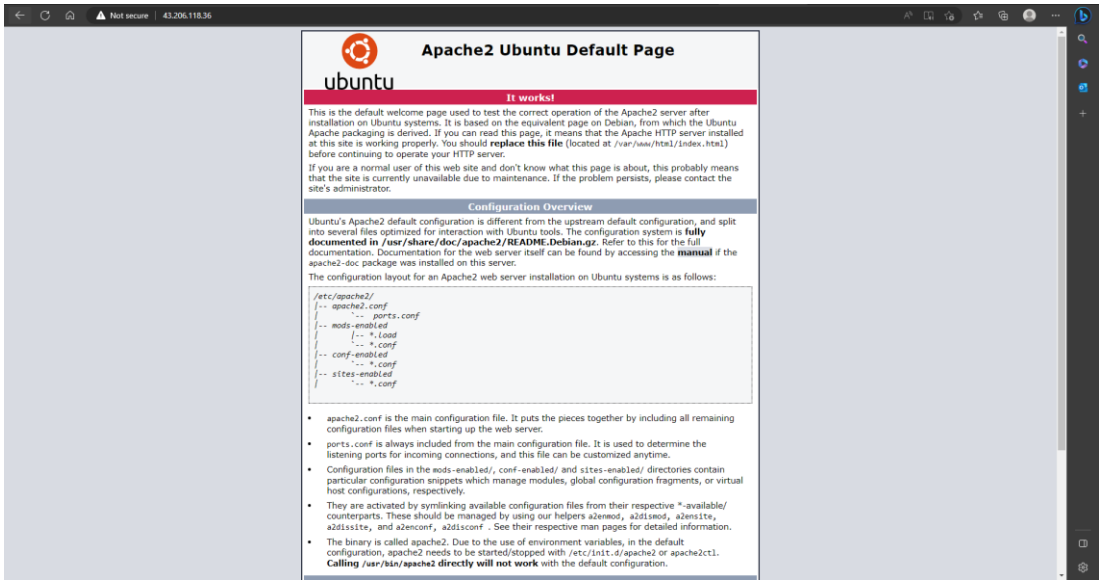
- **docker build -t myapp:v1 path_of_the_file** → command to Build an Image in Docker

```
root@ip-172-31-3-7: /home/ubuntu/Docker_dir
ubuntu@ip-172-31-3-7:~$ ls
ubuntu@ip-172-31-3-7:~$ mkdir Docker_dir
ubuntu@ip-172-31-3-7:~$ ls
Docker_dir
ubuntu@ip-172-31-3-7:~$ sudo su
root@ip-172-31-3-7:/home/ubuntu# cd Docker_dir
root@ip-172-31-3-7:/home/ubuntu/Docker_dir# vi Dockerfile
root@ip-172-31-3-7:/home/ubuntu/Docker_dir# docker build -t myapp:v4 .
[+] Building 35.0s (7/7) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 333B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/ubuntu:18.04
 => [auth] library/ubuntu:pull token for registry-1.docker.io
 => [1/2] FROM docker.io/library/ubuntu:18.04@sha256:8aa9c2798215f99544d1ce7439ea9c3a6dfd82de607da1cec3a8a2fae005931b
 => => resolve docker.io/library/ubuntu:18.04@sha256:8aa9c2798215f99544d1ce7439ea9c3a6dfd82de607da1cec3a8a2fae005931b
 => => sha256:0c5227665c11379f79e9da3d3e4f1724f9316b87d259ac0131628ca1b923a392 25.69MB / 25.69MB
 => => sha256:8aa9c2798215f99544d1ce7439ea9c3a6dfd82de607da1cec3a8a2fae005931b 1.33kB / 1.33kB
 => => sha256:0779371f96205678dbcaa3ef499be2e5f262c8b09aadc11754bf3daf9f35e03e 424B / 424B
 => => sha256:3941d3b032a8168d53508410a67baad120a563df67a7959565a30a1cb2114731 2.30kB / 2.30kB
 => => extracting sha256:0c5227665c11379f79e9da3d3e4f1724f9316b87d259ac0131628ca1b923a392
 => [2/2] RUN apt-get update && apt-get install -y apache2 && apt-get clean && rm -rf /var/lib/apt/lists/*
 => exporting to image
 => => exporting layers
 => => writing image sha256:51f6bd24e9f1a381eb4f5fc22257b0ff8d6515c92703a00d58199c771b420ad0
 => => naming to docker.io/library/myapp:v4
root@ip-172-31-3-7:/home/ubuntu/Docker_dir#
```

- **docker run –name myapache -d -p 80:80 myapp:v4** → command to run docker image

```
root@ip-172-31-4-90: /home/ubuntu/Docker_dir
ubuntu@ip-172-31-4-90:~$ sudo su
root@ip-172-31-4-90:/home/ubuntu# ls
Docker_dir
root@ip-172-31-4-90:/home/ubuntu# cd Docker_dir
root@ip-172-31-4-90:/home/ubuntu/Docker_dir# ls
Dockerfile
root@ip-172-31-4-90:/home/ubuntu/Docker_dir# docker run --name myapache -d -p 80:80 myapp:v4
4f90be354c14080d0117d317633d2189fa50a59e5e009e13b975479a5b18054d
root@ip-172-31-4-90:/home/ubuntu/Docker_dir# ls
Dockerfile
root@ip-172-31-4-90:/home/ubuntu/Docker_dir#
```

5. Here Apache container has been created successfully using Dockerfile:



Assignment 2 - Docker Compose and WordPress.docx.txt

In this Assignment, we need to configure Docker Compose to easily run WordPress in an isolated environment built with Docker containers

1. Install docker compose on docker host and confirm the docker-compose version:

Command to Install Docker Compose

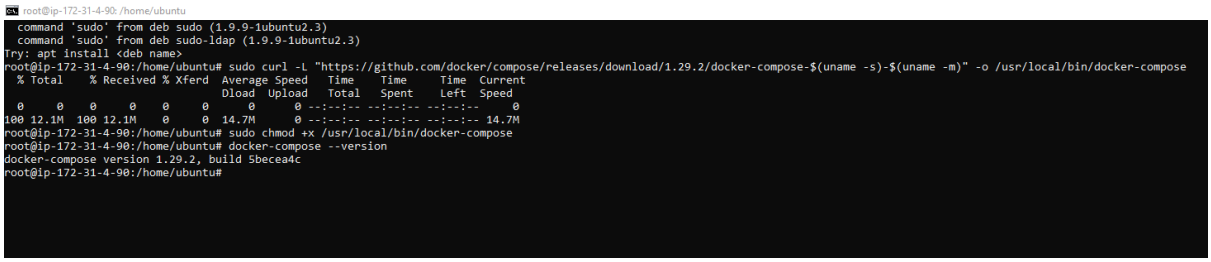
- **sudo curl -L https://github.com/docker/compose/releases/download/1.29.2/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose**

Set the permissions

- **sudo chmod +x /usr/local/bin/docker-compose**

Checking the version of Docker Compose

- **docker-compose --version**



2. Create a new folder and write docker-compose.yaml with the below contents:

```
version: â€œ3.9â€
services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: somewordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
```

ports:

- "8000:80"

restart: always

environment:

WORDPRESS_DB_HOST: db:3306

WORDPRESS_DB_USER: wordpress

WORDPRESS_DB_PASSWORD: wordpress

WORDPRESS_DB_NAME: wordpress

volumes:

db_data: {}

root@ip-172-31-4-90: /home/ubuntu/docker-compose

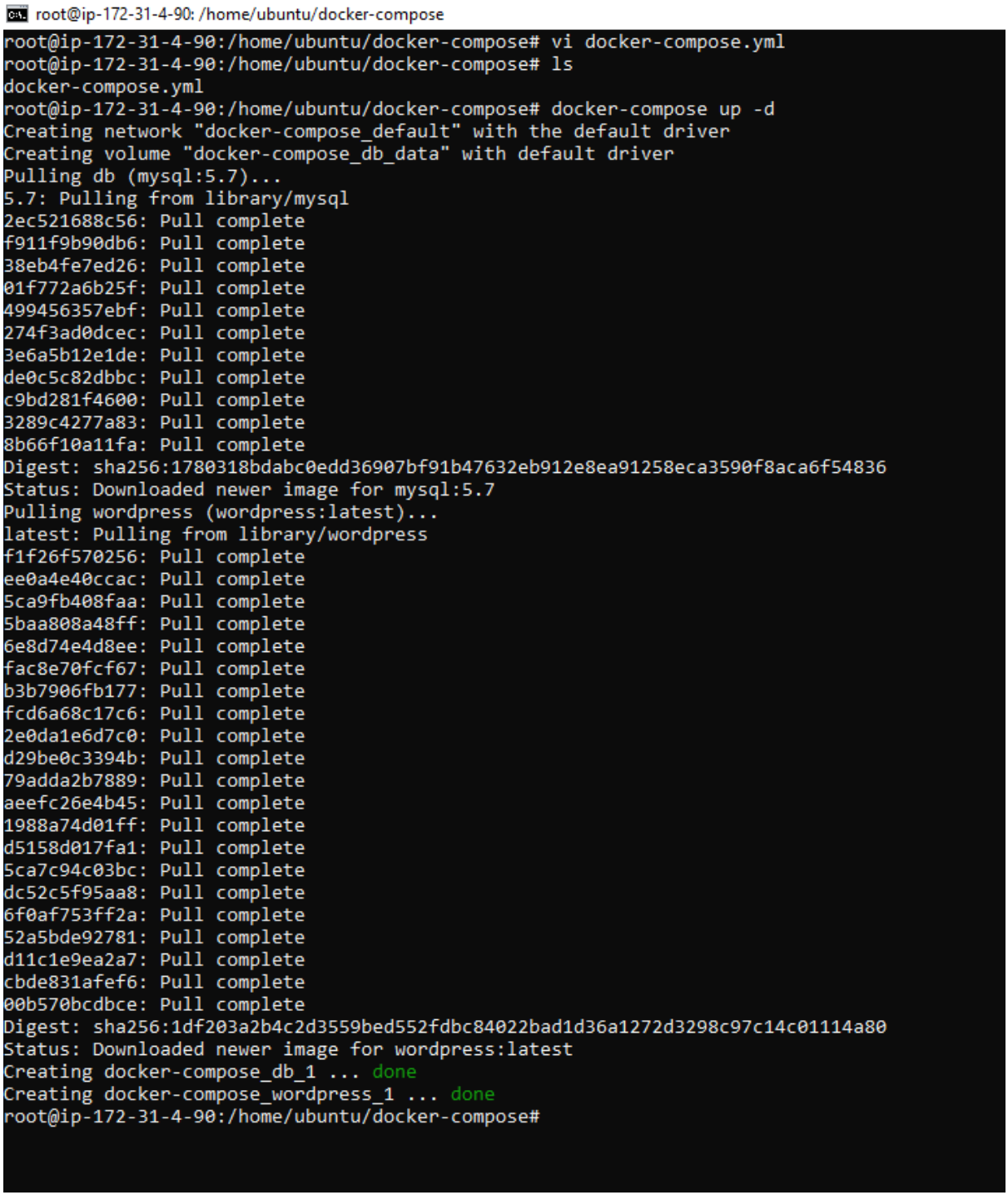
```
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 12.1M  100 12.1M    0     0  14.7M      0 --:--:-- --:--:-- --:--:-- 14.7M
root@ip-172-31-4-90:/home/ubuntu# sudo chmod +x /usr/local/bin/docker-compose
root@ip-172-31-4-90:/home/ubuntu# docker-compose --version
docker-compose version 1.29.2, build 5becea4c
root@ip-172-31-4-90:/home/ubuntu# mkdir docker-compose
root@ip-172-31-4-90:/home/ubuntu# ls
Docker_dir  docker-compose
root@ip-172-31-4-90:/home/ubuntu# cd docker-compose/
root@ip-172-31-4-90:/home/ubuntu/docker-compose# vi docker-compose.yml
root@ip-172-31-4-90:/home/ubuntu/docker-compose#
```

root@ip-172-31-4-90: /home/ubuntu/docker-compose

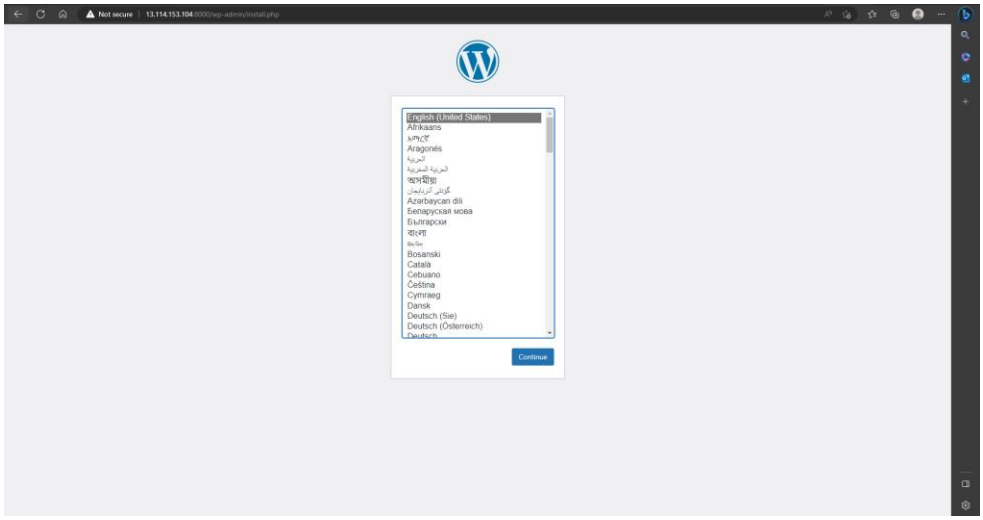
```
version: "3.9"
services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: somewordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress


  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress
volumes:
  db_data: {}
~
~
~
~
```

3. Build the application using docker-compose application:

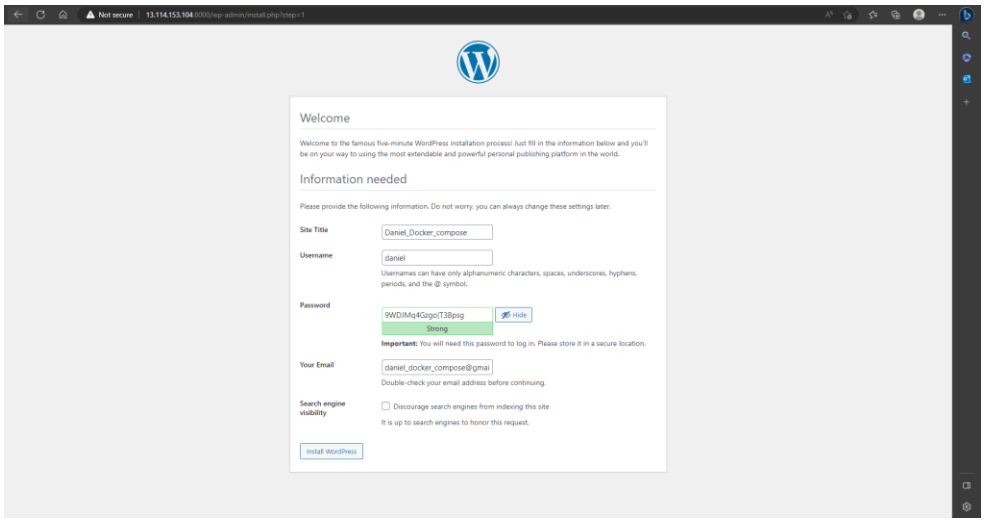
- **docker-compose up -d** → command to Build and Run Docker compose

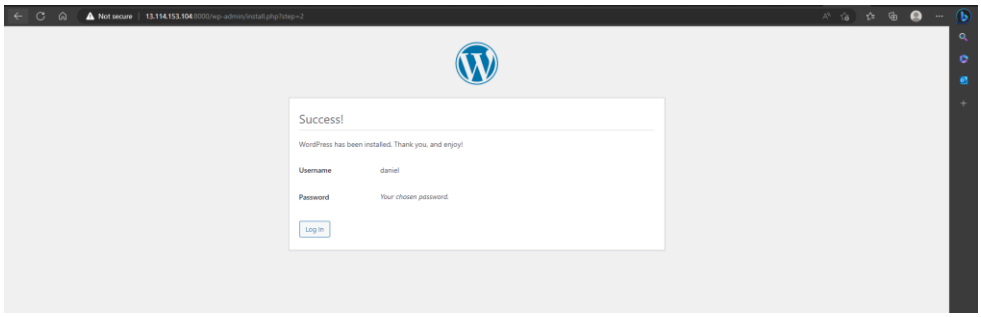


4. Access the application under port 8000:

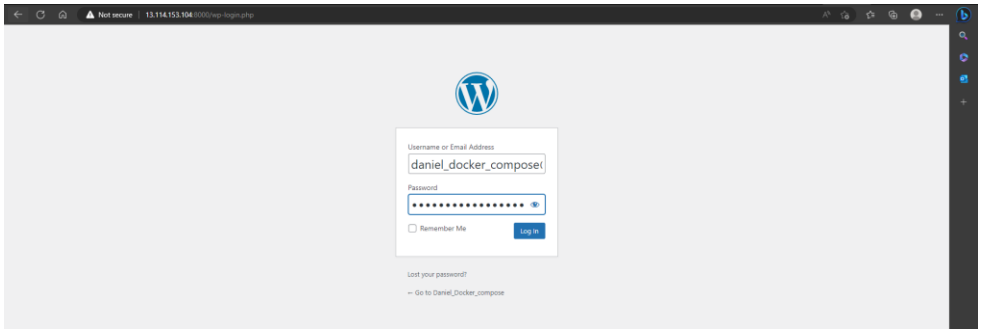


- Providing Username and Login Credentials to Login and Install Wordpress:

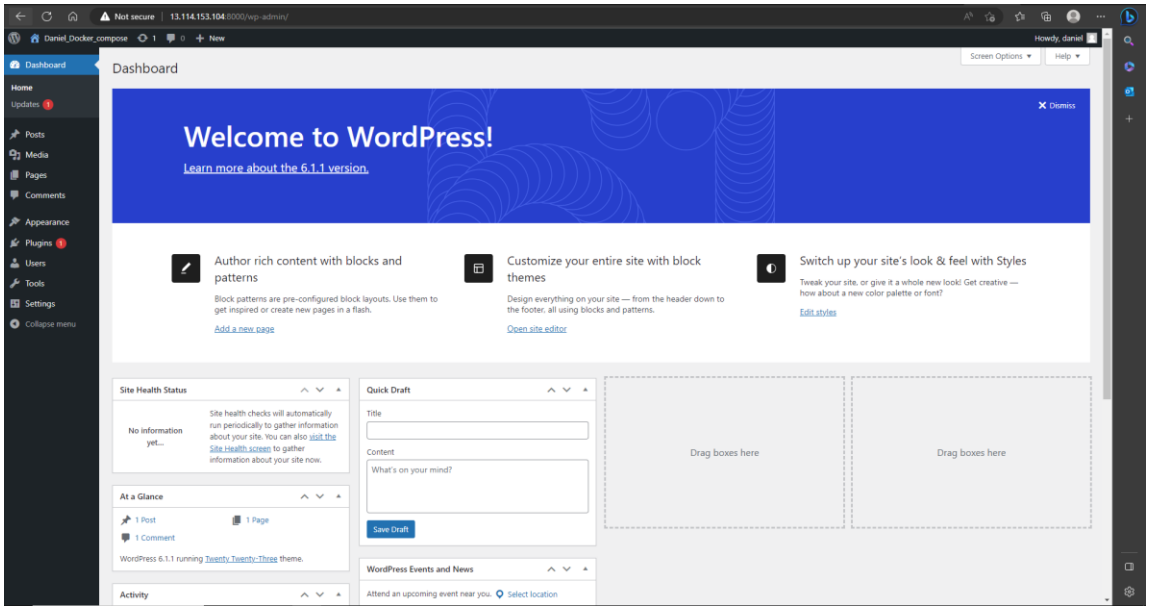




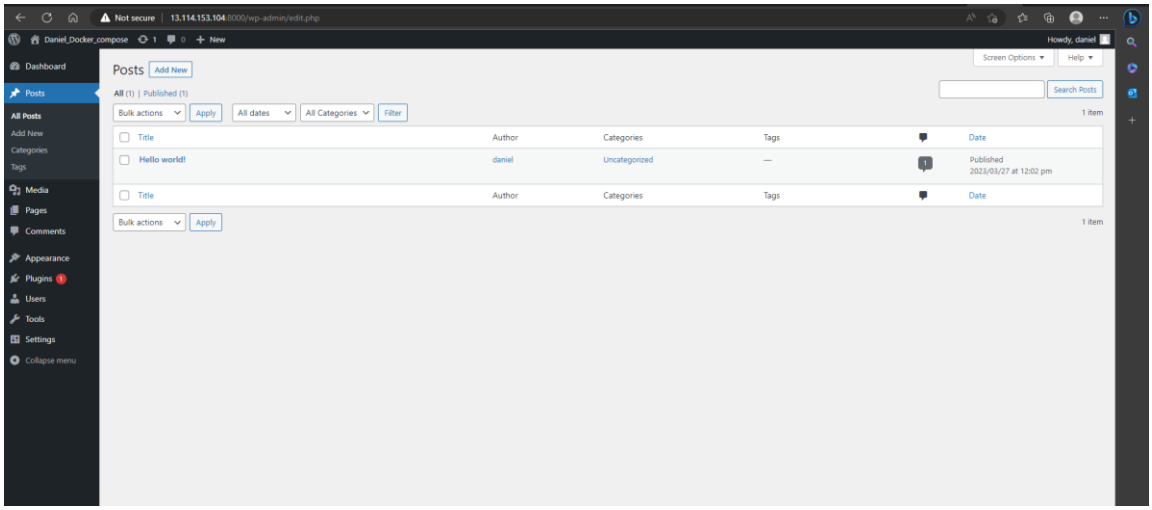
- Logging in:



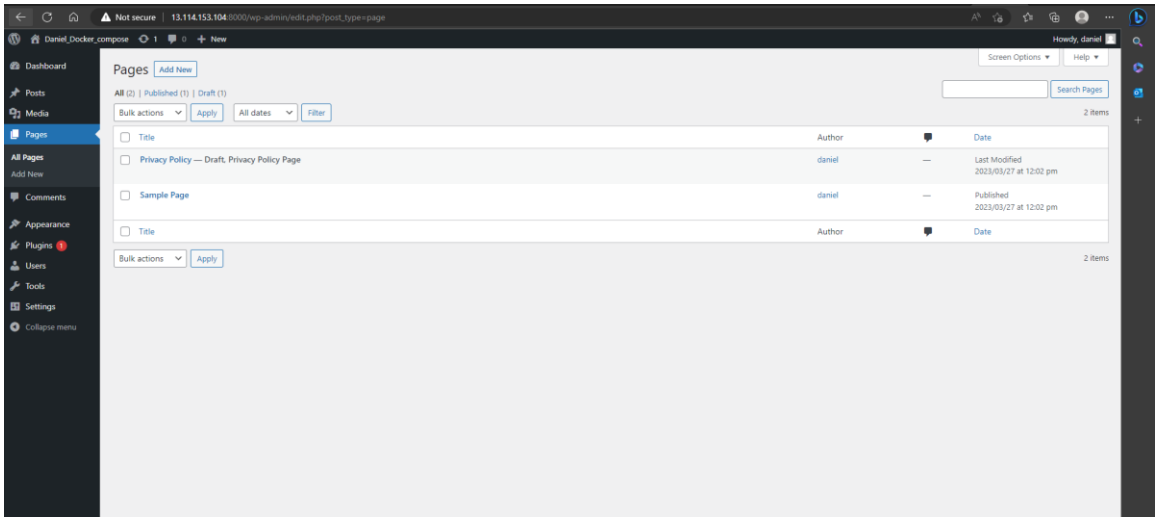
- Logged In Successfully:



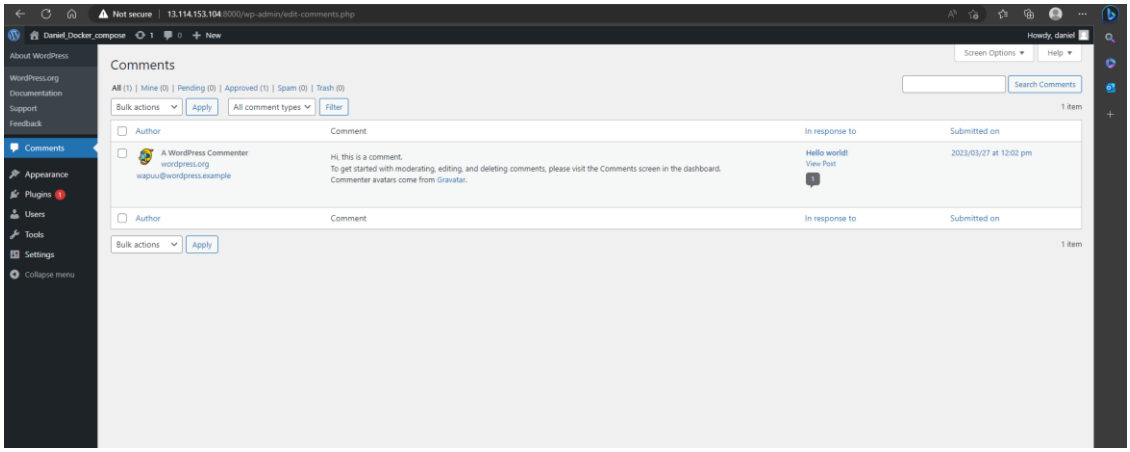
- Posts Tab:



- Pages Tab:



- Comments Tab



5. Observe the mapping of volumes between host and mysql containers:

- Here container wordpress has been mapped to port 8000
- And container mysql has been mapped to port 3306, which is default port of mysql

➢ **docker ps** → command to observe the Mapping of volumes between host and mysql

➢ **docker inspect mysql_container** → views the full details mysql container

```
root@ip-172-31-36-95: /home/ubuntu
root@ip-172-31-36-95:/home/ubuntu# docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS                                   NAMES
785723cd673e   mysql:5.7          "docker-entrypoint.s…"   27 minutes ago   Up 27 minutes   3306/tcp, 33060/tcp                     docker-compose_db_1
095ea5be8ad0   wordpress:latest   "docker-entrypoint.s…"   39 minutes ago   Up 39 minutes   0.0.0.0:8000->80/tcp, :::8000->80/tcp   docker-compose_wordpress_1
root@ip-172-31-36-95:/home/ubuntu# docker inspect 785723cd673e
[
    {
        "Id": "785723cd673ea4ad0932c7c7827cc2b0cff87edaefc328f324617e99198cc931",
        "Created": "2023-07-02T13:02:35.891518622Z",
        "Path": "docker-entrypoint.sh",
        "Args": [
            "mysqld"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 2379,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2023-07-02T13:02:36.238099941Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:2be84dd575ee2ecdb186dc43a9cd951890a764d2cefbd31a72cdf4410c43a2d0",
        "ResolvConfPath": "/var/lib/docker/containers/785723cd673ea4ad0932c7c7827cc2b0cff87edaefc328f324617e99198cc931/resolv.conf",
        "HostnamePath": "/var/lib/docker/containers/785723cd673ea4ad0932c7c7827cc2b0cff87edaefc328f324617e99198cc931/hostname",
        "HostsPath": "/var/lib/docker/containers/785723cd673ea4ad0932c7c7827cc2b0cff87edaefc328f324617e99198cc931/hosts",
        "LogPath": "/var/lib/docker/containers/785723cd673ea4ad0932c7c7827cc2b0cff87edaefc328f324617e99198cc931/785723cd673ea4ad0932c7c7827cc2b0cff87edaefc328f324617e99198cc931-json.log",
        "Name": "/docker-compose_db_1",
        "RestartCount": 0,
        "Driver": "overlay2",
        "Platform": "linux",
        "MountLabel": "",
        "ProcessLabel": "",
        "AppArmorProfile": "docker-default",
        "ExecIDs": null,
        "HostConfig": {
            "Binds": [
                "docker-compose_db_data:/var/lib/mysql:rw"
            ],
            "ContainerIDFile": "",
            "LogConfig": {
                "Type": "json-file",
                "Config": {}
            },
            "NetworkMode": "docker-compose_default",
            "PortBindings": {},
            "RestartPolicy": {
                "Name": "always",
                "MaximumRetryCount": 0
            },
            "AutoRemove": false,
            "VolumeDriver": "",
            "VolumesFrom": [],
            "ConsoleSize": [
                0,
                0
            ],
            "CapAdd": null,
            "CapDrop": null,
```

- Look for the "Volumes" section in the output of the previous command. This section will list all the volumes that are mounted to the container.

  "Volumes": {
  "/var/lib/mysql": {}
  }

- Find the location of the volume on the Docker host by looking at the "Source" field in the same "Volumes" section

root@ip-172-31-36-95: /home/ubuntu

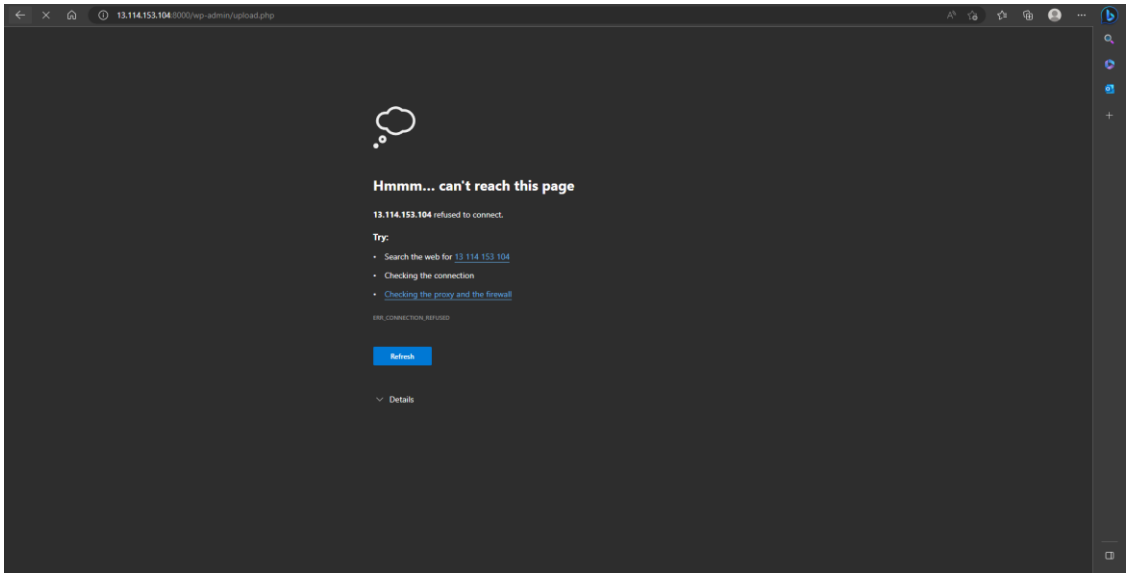

```
            },
            "Name": "overlay2"
        },
        "Mounts": [
            {
                "Type": "volume",
                "Name": "docker-compose_db_data",
                "Source": "/var/lib/docker/volumes/docker-compose_db_data/_data",
                "Destination": "/var/lib/mysql",
                "Driver": "local",
                "Mode": "rw",
                "RW": true,
                "Propagation": ""
            }
        ],
        "Config": {
            "Hostname": "785723cd673e",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "ExposedPorts": {
                "3306/tcp": {},
                "33060/tcp": {}
            },
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "MYSQL_ROOT_PASSWORD=somewordpress",
                "MYSQL_DATABASE=wordpress",
                "MYSQL_USER=wordpress",
                "MYSQL_PASSWORD=wordpress",
                "affinity:container==5b65c7e93aff18d7a292c9d7f502eafdaf4f1cca84a2adf69459b7e926d7a232",
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
                "GOSU_VERSION=1.16",
                "MYSQL_MAJOR=5.7",
                "MYSQL_VERSION=5.7.42-1.el7",
                "MYSQL_SHELL_VERSION=8.0.33-1.el7"
            ],
            "Cmd": [
                "mysqld"
            ],
            "Image": "mysql:5.7",
            "Volumes": {
                "/var/lib/mysql": {}
            },
            "WorkingDir": "",
            "Entrypoint": [
                "docker-entrypoint.sh"
            ],
            "OnBuild": null,
            "Labels": {
                "com.docker.compose.config-hash": "1cfb7a8334e5348cb9ea66433cce96053166f47a4d8f96c83cc1eb6dde579de7",
                "com.docker.compose.container-number": "1",
                "com.docker.compose.oneoff": "False",
                "com.docker.compose.project": "docker-compose",
                "com.docker.compose.project.config_files": "docker-compose.yaml",
                "com.docker.compose.project.working_dir": "/home/ubuntu/docker-compose",
                "com.docker.compose.service": "db",
                "com.docker.compose.version": "1.29.2"
            }
```


6. Make the application down and remove the volumes using docker compose commands:

➢ **docker compose down** → command to make the application down

root@ip-172-31-4-90: /home/ubuntu/docker-compose

```
                    (default: 10)
root@ip-172-31-4-90:/home/ubuntu/docker-compose# docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS                                            NAMES
b082aafbdad8   wordpress:latest   "docker-entrypoint.s…"   28 minutes ago   Up 28 minutes   0.0.0.0:8000->80/tcp, :::8000->80/tcp            docker-compose_wordpress_1
04194719a208   mysql:5.7          "docker-entrypoint.s…"   28 minutes ago   Up 28 minutes   3306/tcp, 33060/tcp                              docker-compose_db_1
root@ip-172-31-4-90:/home/ubuntu/docker-compose# docker-compose down
Stopping docker-compose_wordpress_1 ... done
Stopping docker-compose_db_1        ... done
Removing docker-compose_wordpress_1 ... done
Removing docker-compose_db_1        ... done
Removing network docker-compose_default
root@ip-172-31-4-90:/home/ubuntu/docker-compose# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@ip-172-31-4-90:/home/ubuntu/docker-compose#
```



- Removing the volumes:
  - **docker volume ls** → command to view the volumes

```
root@ip-172-31-4-90: /home/ubuntu/docker-compose
Stopping docker-compose_wordpress_1 ... done
Stopping docker-compose_db_1        ... done
Removing docker-compose_wordpress_1 ... done
Removing docker-compose_db_1        ... done
Removing network docker-compose_default
root@ip-172-31-4-90:/home/ubuntu/docker-compose# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@ip-172-31-4-90:/home/ubuntu/docker-compose# docker volume ls
DRIVER    VOLUME NAME
local     8e3cdabbd00c0399599c2f0b1f8d705985a550addcd080ffd76c91335013fcbf
local     docker-compose_db_data
root@ip-172-31-4-90:/home/ubuntu/docker-compose#
```

  - **docker volume rm volume_name** → command to remove volumes

Here docker-compose_db_data volume has been removed.

```
root@ip-172-31-4-90: /home/ubuntu/docker-compose
Stopping docker-compose_wordpress_1 ... done
Stopping docker-compose_db_1        ... done
Removing docker-compose_wordpress_1 ... done
Removing docker-compose_db_1        ... done
Removing network docker-compose_default
root@ip-172-31-4-90:/home/ubuntu/docker-compose# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@ip-172-31-4-90:/home/ubuntu/docker-compose# docker volume ls
DRIVER    VOLUME NAME
local     8e3cdabbd00c0399599c2f0b1f8d705985a550addcd080ffd76c91335013fcbf
local     docker-compose_db_data
root@ip-172-31-4-90:/home/ubuntu/docker-compose# docker volume rm docker-compose_db_data
docker-compose_db_data
root@ip-172-31-4-90:/home/ubuntu/docker-compose# docker volume ls
DRIVER    VOLUME NAME
local     8e3cdabbd00c0399599c2f0b1f8d705985a550addcd080ffd76c91335013fcbf
root@ip-172-31-4-90:/home/ubuntu/docker-compose#
```