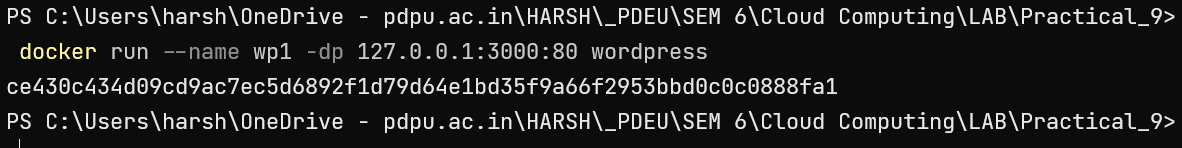
**Lab 8 Assignment**

**Working with DOCKER**

**Aim:** Re-Do the problem of Lab 5 using docker compose. Also do assign a static IP address to WordPress container using subnetting.

Create two docker containers with front-end and back-end applications and connect these containers using docker compose.

1. **Running WordPress container normally**



1. **Creating *compose.yaml* file**

services:

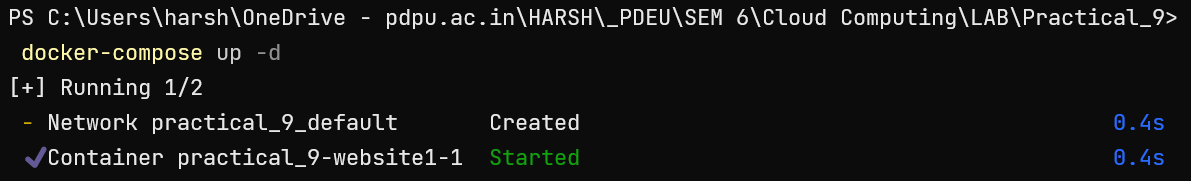
website1:

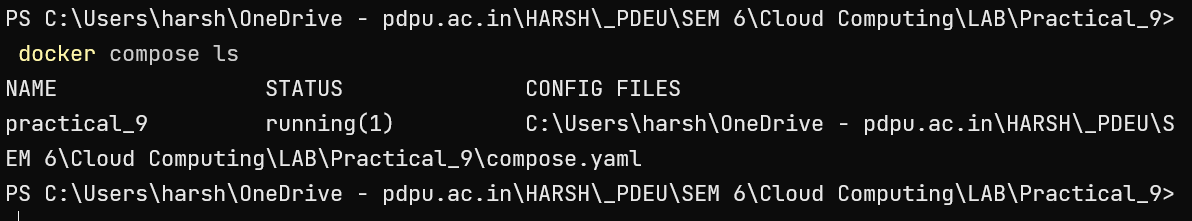
image: wordpress

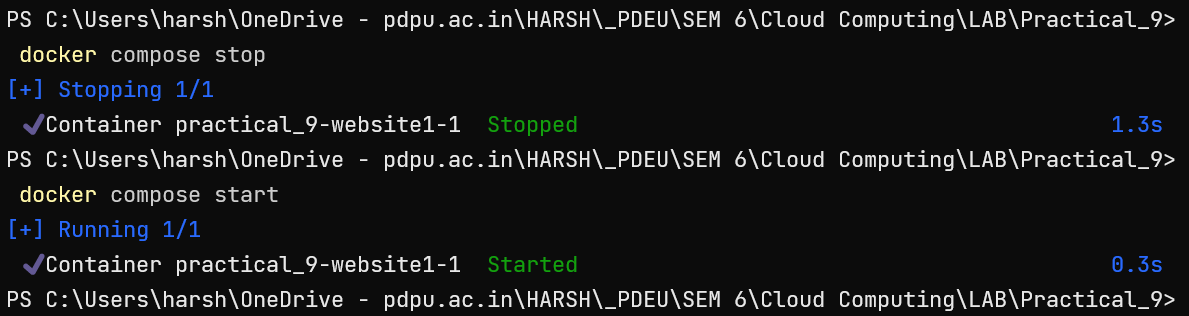
ports:

- "3000:80"

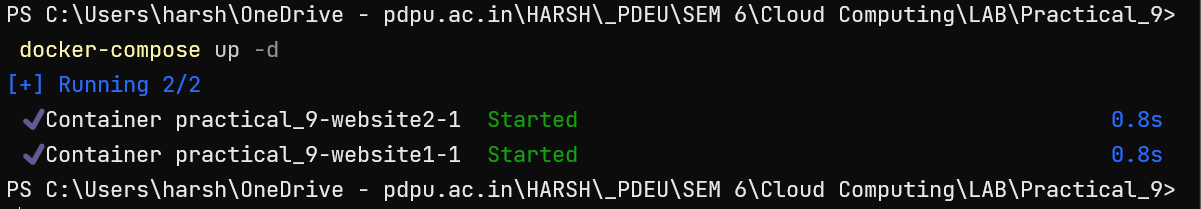
1. **Execute Docker Compose**







1. **Running multiple containers**



1. **Configuring network**

services:

  website1:

    image: wordpress

    ports:

      - "3100:80"

  website2:

    image: wordpress

    ports:

      - "3200:80"

networks:

      wordpress\_net:

        ipv4\_address: 192.168.99.20

networks:

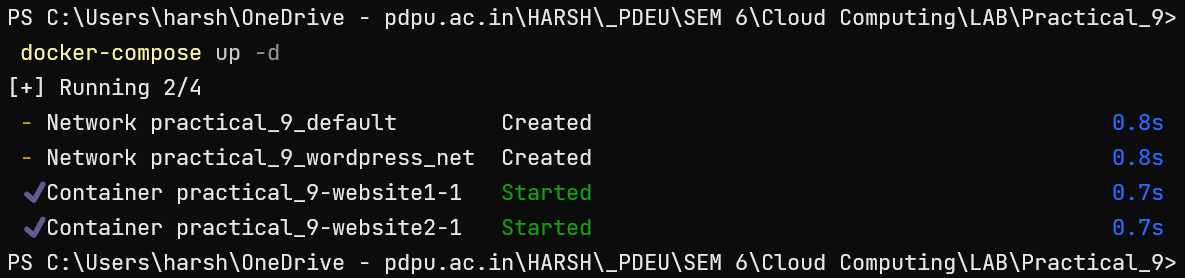
  wordpress\_net:

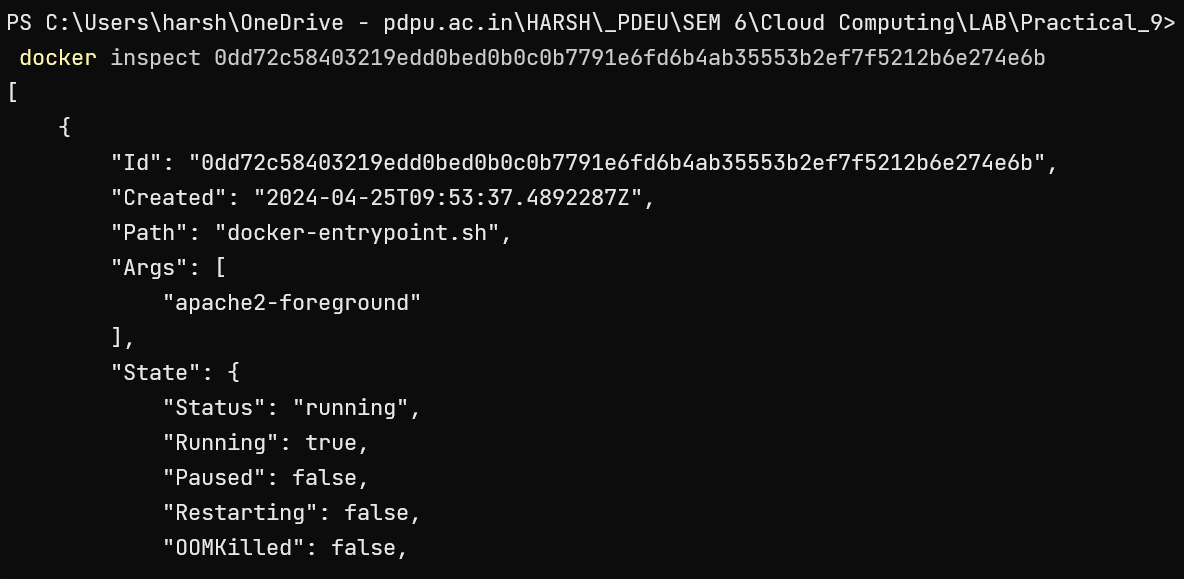
    ipam:

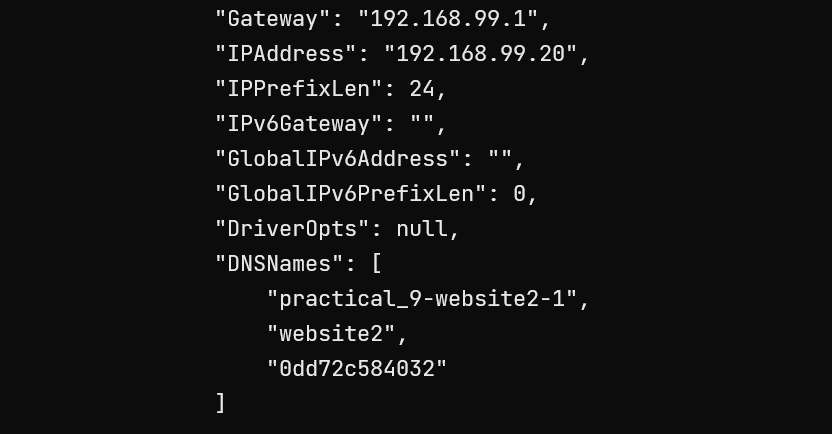
      driver: default

      config:

        - subnet: "192.168.99.0/24"







1. **Connection Database with Container**

version: "3"

services:

  mariadb\_db:

    container\_name: mariadb\_container

    image: mariadb

    restart: always

    environment:

      MARIADB\_ROOT\_PASSWORD: password

      MARIADB\_DATABASE: testdb

      MARIADB\_USER: harsh

      MARIADB\_PASSWORD: wp\_password

    volumes:

    - mariadb\_vol:/var/lib/mysql

  wordpress:

    depends\_on:

      - mariadb\_db

    image: wordpress:latest

    restart: always

    ports:

      - "8000:80"

    environment:

      WORDPRESS\_DB\_HOST: mariadb\_db:3306

      WORDPRESS\_DB\_NAME: testdb

      WORDPRESS\_DB\_USER: harsh

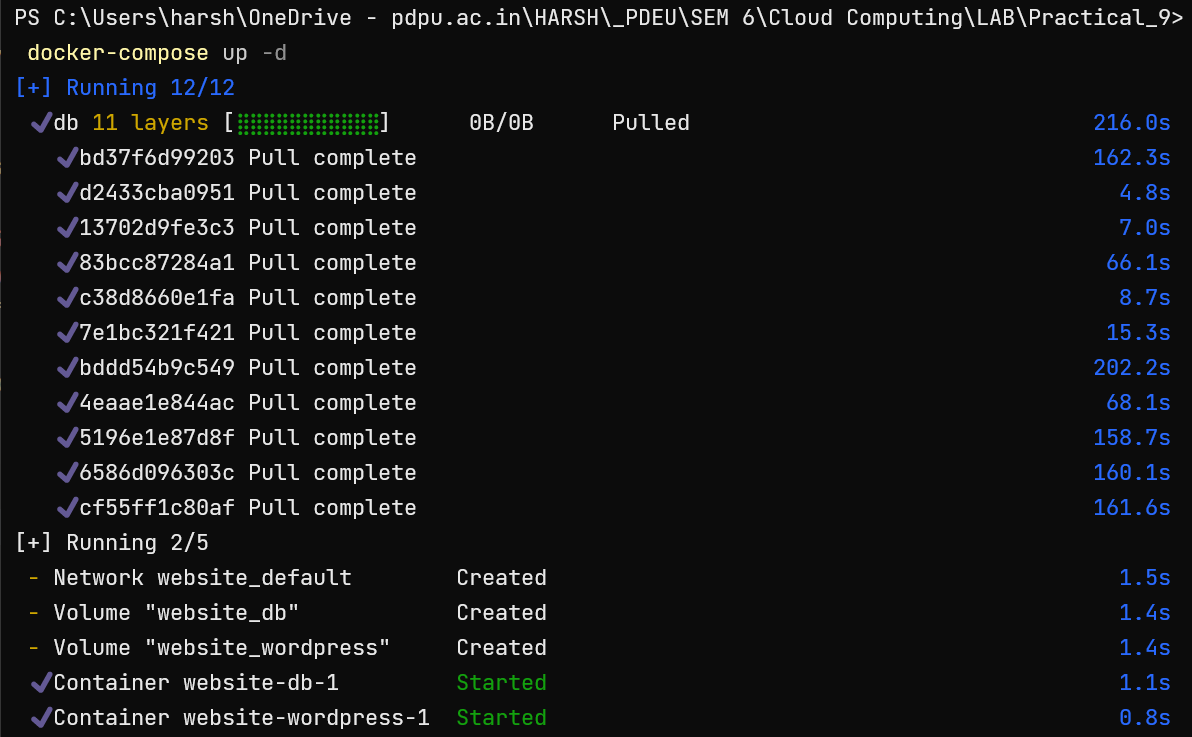
      WORDPRESS\_DB\_PASSWORD: wp\_password

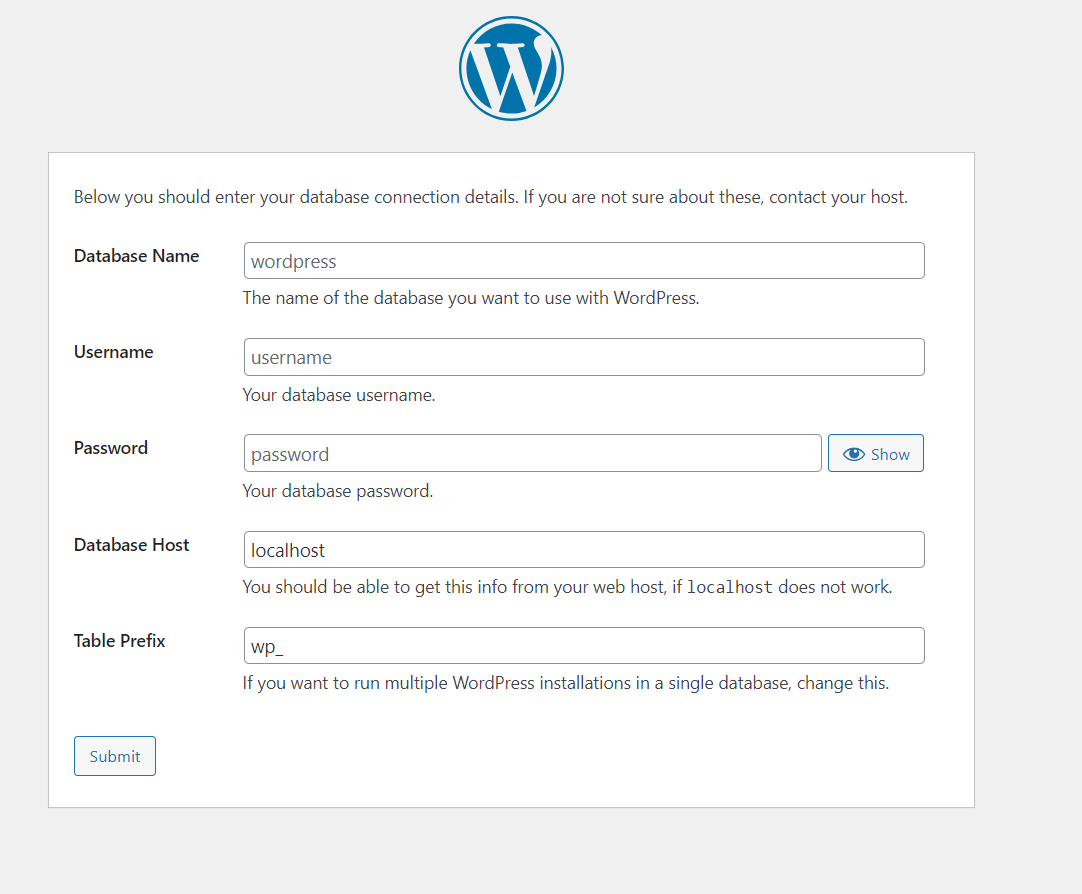
    volumes:

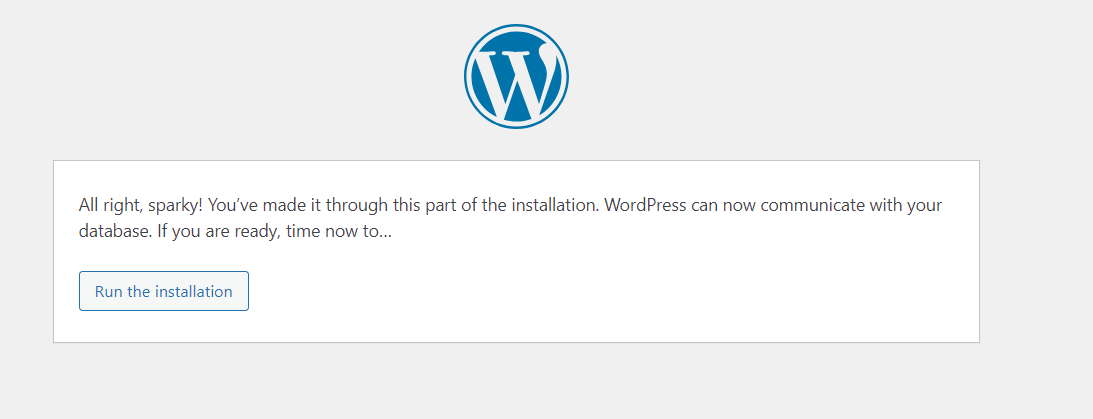
      ["./:/var/www/html"]

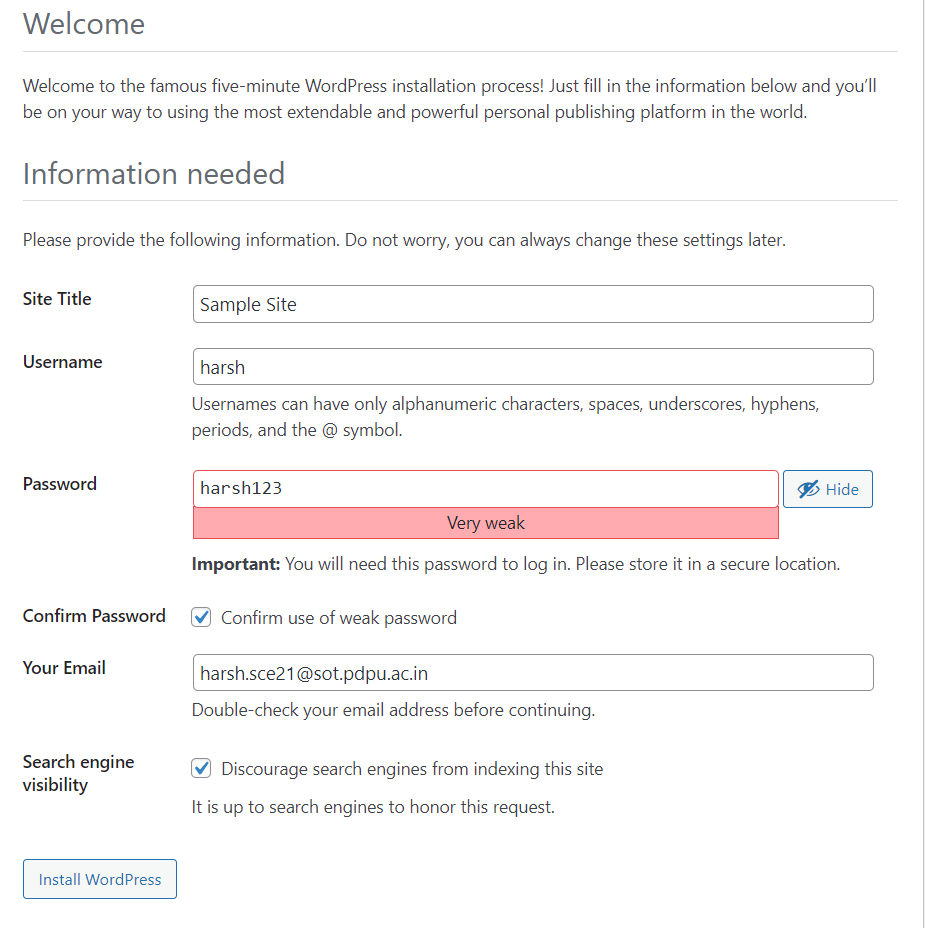
volumes:

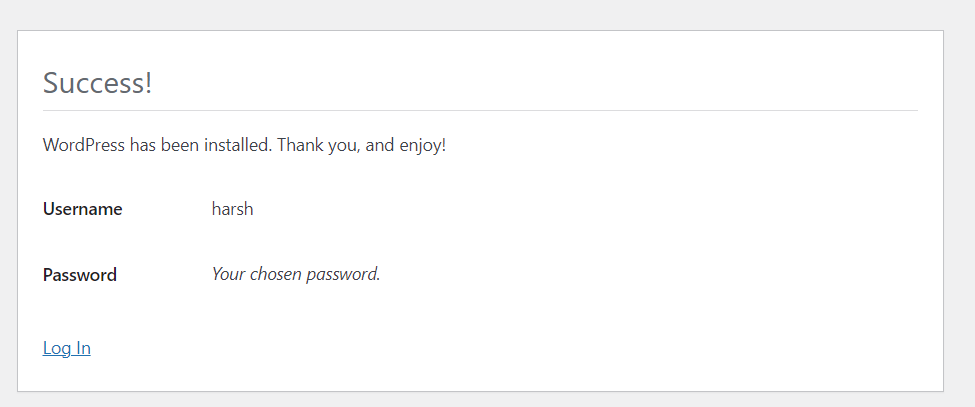
 mariadb\_vol: {}











1. **Access Website on HostIP:PORT**

<http://127.0.0.1:3300/>