Before proceeding with the project, kindly fill out the feedback form for week 1 below:

https://forms.office.com/r/4F2xy5HvPY

Do not hesitate in sharing what you liked the most and what you think can be done to improve your learning experience.

Project 01

Objectives:

- Create and manage Docker volumes for data persistence.
- Set up a Docker network for container communication.
- Use Docker Compose to manage multi-container applications.
- View and manage Docker logs.
- Deploy the application using Docker Swarm.

Project Outline:

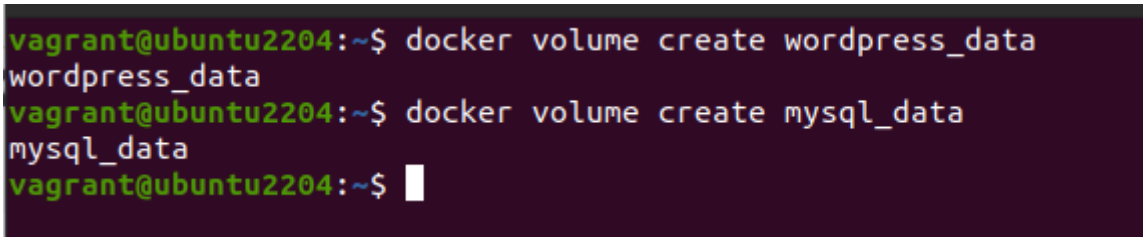
1. **Create Docker Volumes**
2. **Create a Docker Network**
3. **Write a Docker Compose File**
4. **Deploy the Application with Docker Compose**
5. **Manage Docker Logs**
6. **Deploy the Application Using Docker Swarm**

Step-by-Step Guide

1. Create Docker Volumes

Docker volumes are used to persist data generated by and used by Docker containers.

```
docker volume create wordpress_data
docker volume create mysql_data
```



2. Create a Docker Network

Create a custom network for the containers to communicate.

```
docker network create wordpress_network
```

```
vagrant@ubuntu2204:~$ docker network create wordpress_network
1c036919a95d7a357ee7aa49c6857037f2fb1ad8240a6df4c2b2afe8d4e3307a
vagrant@ubuntu2204:~$ docker network ls
NETWORK ID     NAME               DRIVER    SCOPE
7884211bef33   bridge             bridge    local
837f65cceea4   docker_gwbridge    bridge    local
0cca65868ec7   host               host      local
tfqixj2t9di5   ingress            overlay   swarm
bb73acf533a5   none               null      local
1c036919a95d   wordpress_network  bridge    local
vagrant@ubuntu2204:~$
```

3. Write a Docker Compose File

Create a docker-compose.yml file to define and manage the services.

```
version: '3.3'

services:
  db:
    image: mysql:5.7
    volumes:
      - mysql_data:/var/lib/mysql
    networks:
      - wordpress_network
    environment:
      MYSQL_ROOT_PASSWORD: example
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress

  wordpress:
    image: wordpress:latest
    volumes:
      - wordpress_data:/var/www/html
    networks:
      - wordpress_network
    ports:
      - "8000:80"
    environment:
```

```
    WORDPRESS_DB_HOST: db:3306
    WORDPRESS_DB_USER: wordpress
    WORDPRESS_DB_PASSWORD: wordpress
    WORDPRESS_DB_NAME: wordpress

volumes:
  mysql_data:
  wordpress_data:

networks:
  wordpress_network:
```

4. Deploy the Application with Docker Compose

Run the following command to start the services defined in the docker-compose.yml file.

```
docker-compose up -d
```

```
vagrant@ubuntu2204:~$ docker compose up -d
WARN[0000] /home/vagrant/docker-compose.yml: `version` is obsolete
[+] Running 7/34
 ⠿ db [⠿⠿⠿⠿⠿⠿⠿⠿⠿⠿]   ] Pulling                                                    81.4s
   ⠿ 20e4dcae4c69 Downloading  4.614MB/50.5MB                                            76.8s
   ✔ 1c56c3d4ce74 Download complete                                                      26.9s
   ✔ e9f03a1c24ce Download complete                                                      39.6s
   ✔ 68c3898c2015 Download complete                                                      47.5s
   ✔ 6b95a940e7b6 Download complete                                                      41.7s
   ✔ 90986bb8de6e Download complete                                                      49.1s
   ⠿ ae71319cb779 Downloading  6.572MB/25.53MB                                           76.8s
   ✔ ffc89e9dfd88 Download complete                                                      66.9s
   ⠿ 43d05e938198 Downloading  1.081MB/56.29MB                                           76.8s
   ⠿ 064b2d298fba Waiting                                                                76.8s
   ⠿ df9a4d85569b Waiting                                                                76.8s
 ⠿ wordpress [⠿                   ] Pulling                                             81.4s
   ✔ f11c1adaa26e Already exists                                                          0.0s
   ⠿ 91c1fd48de30 Waiting                                                                76.8s
   ⠿ c3b3bda7c6d1 Waiting                                                                76.8s
   ⠿ 65a68eb681dd Waiting                                                                76.8s
   ⠿ 35406f9afc7f Waiting                                                                76.8s
   ⠿ a7d29e357509 Waiting                                                                76.8s
   ⠿ d497b137ced8 Waiting                                                                76.8s
   ⠿ dabbc6b5ab09 Waiting                                                                76.8s
   ⠿ 42ebbd004593 Waiting                                                                76.8s
   ⠿ d8437f303b6d Waiting                                                                76.8s
   ⠿ 1eb7786e3600 Waiting                                                                76.8s
   ⠿ 6f109a68b308 Waiting                                                                76.8s
   ⠿ 3fae4e1410c6 Waiting                                                                76.8s
   ⠿ 374204136091 Waiting                                                                76.8s
   ⠿ 10be860e0dea Waiting                                                                76.8s
   ⠿ d15b284f6870 Waiting                                                                76.8s
   ⠿ 4566618a287b Waiting                                                                76.8s
   ⠿ e67b58997f2b Waiting                                                                76.8s
   ⠿ a66deda1e7f1 Waiting                                                                76.8s
   ⠿ e911796db38f Waiting                                                                76.8s
   ⠿ d17017b188bc Waiting                                                                76.8s
```

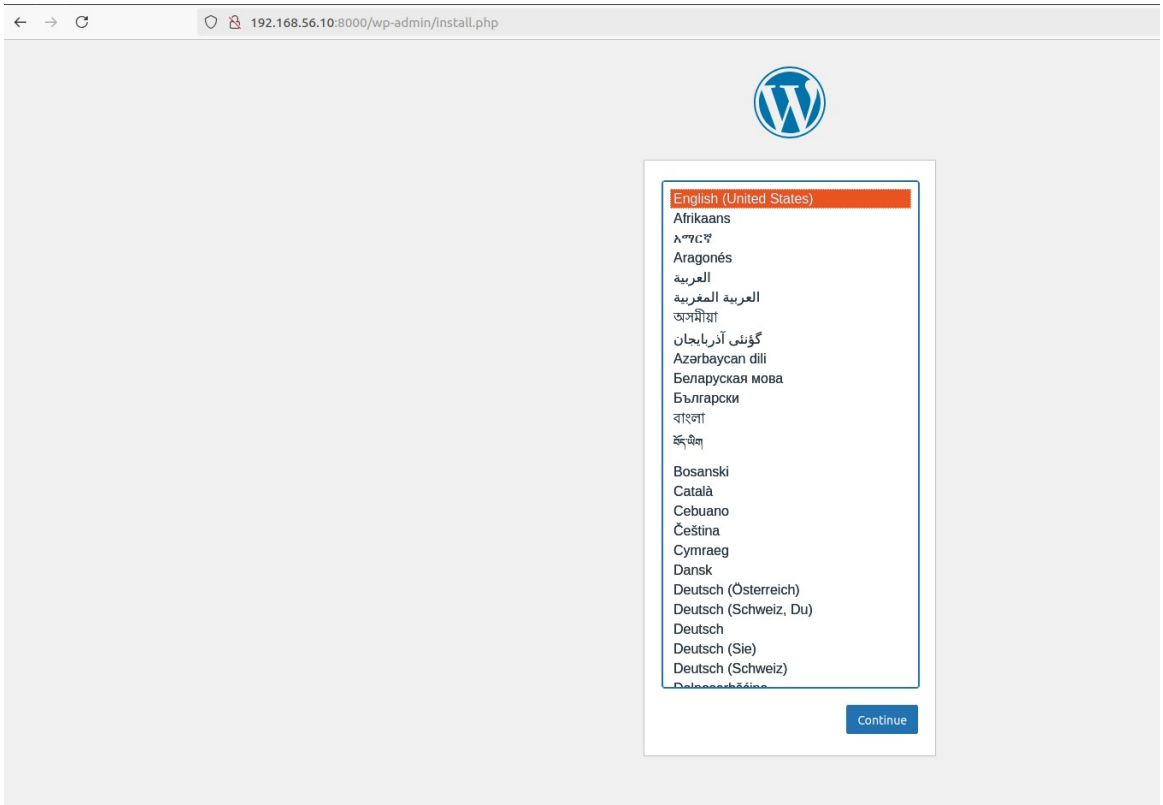
- Verify that the containers are running.

```
[+] Running 5/5
 ✔ Network vagrant_wordpress_network  Created                                    0.1s
 ✔ Volume "vagrant_wordpress_data"    Created                                    0.0s
 ✔ Volume "vagrant_mysql_data"        Created                                    0.0s
 ✔ Container vagrant-db-1             Started                                    0.5s
 ✔ Container vagrant-wordpress-1      Started                                    0.4s
vagrant@ubuntu2204:~$
```

docker-compose ps

```
vagrant@ubuntu2204:~$ docker compose ps
WARN[0000] /home/vagrant/docker-compose.yml: `version` is obsolete
NAME                  IMAGE              COMMAND                  SERVICE     CREATED         STATUS
 PORTS
vagrant-db-1          mysql:5.7          "docker-entrypoint.s…"   db          3 minutes ago   Up 3 minutes
 3306/tcp, 33060/tcp
vagrant-wordpress-1   wordpress:latest   "docker-entrypoint.s…"   wordpress   3 minutes ago   Up 3 minutes
 0.0.0.0:8000->80/tcp
vagrant@ubuntu2204:~$
```

- Access the WordPress setup by navigating to http://localhost:8000.



5. Manage Docker Logs

- View logs for a specific service.

docker-compose logs wordpress

```
vagrant@ubuntu2204:~$ docker compose logs wordpress
WARN[0000] /home/vagrant/docker-compose.yml: `version` is obsolete
wordpress-1  | WordPress not found in /var/www/html - copying now...
wordpress-1  | Complete! WordPress has been successfully copied to /var/www/html
wordpress-1  | No 'wp-config.php' found in /var/www/html, but 'WORDPRESS_...' variables supplied; copying 'w
p-config-docker.php' (WORDPRESS_DB_HOST WORDPRESS_DB_NAME WORDPRESS_DB_PASSWORD WORDPRESS_DB_USER)
wordpress-1  | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, usin
g 172.20.0.2. Set the 'ServerName' directive globally to suppress this message
wordpress-1  | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, usin
g 172.20.0.2. Set the 'ServerName' directive globally to suppress this message
wordpress-1  | [Fri Jul 12 08:24:19.476264 2024] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.59 (Debian
) PHP/8.2.21 configured -- resuming normal operations
wordpress-1  | [Fri Jul 12 08:24:19.476300 2024] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FO
REGROUND'
wordpress-1  | 192.168.56.1 - - [12/Jul/2024:08:28:10 +0000] "GET / HTTP/1.1" 302 409 "-" "Mozilla/5.0 (X11;
 Ubuntu; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
wordpress-1  | 192.168.56.1 - - [12/Jul/2024:08:28:10 +0000] "GET /wp-admin/install.php HTTP/1.1" 200 4665 "
-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
wordpress-1  | 192.168.56.1 - - [12/Jul/2024:08:28:12 +0000] "GET /wp-includes/css/dashicons.min.css?ver=6.5
.5 HTTP/1.1" 200 36068 "http://192.168.56.10:8000/wp-admin/install.php" "Mozilla/5.0 (X11; Ubuntu; Linux x86
_64; rv:128.0) Gecko/20100101 Firefox/128.0"
wordpress-1  | 192.168.56.1 - - [12/Jul/2024:08:28:12 +0000] "GET /wp-includes/css/buttons.min.css?ver=6.5.5
 HTTP/1.1" 200 1807 "http://192.168.56.10:8000/wp-admin/install.php" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64
; rv:128.0) Gecko/20100101 Firefox/128.0"
wordpress-1  | 192.168.56.1 - - [12/Jul/2024:08:28:12 +0000] "GET /wp-admin/css/install.min.css?ver=6.5.5 HT
TP/1.1" 200 2138 "http://192.168.56.10:8000/wp-admin/install.php" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; r
v:128.0) Gecko/20100101 Firefox/128.0"
wordpress-1  | 192.168.56.1 - - [12/Jul/2024:08:28:12 +0000] "GET /wp-admin/css/forms.min.css?ver=6.5.5 HTTP
/1.1" 200 7039 "http://192.168.56.10:8000/wp-admin/install.php" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:
128.0) Gecko/20100101 Firefox/128.0"
```

- Follow logs for real-time updates.

docker-compose logs -f wordpress

```
vagrant@ubuntu2204:~$ docker compose logs -f wordpress
WARN[0000] /home/vagrant/docker-compose.yml: `version` is obsolete
wordpress-1  | WordPress not found in /var/www/html - copying now...
wordpress-1  | Complete! WordPress has been successfully copied to /var/www/html
wordpress-1  | No 'wp-config.php' found in /var/www/html, but 'WORDPRESS_...' variables supplied; copying 'w
p-config-docker.php' (WORDPRESS_DB_HOST WORDPRESS_DB_NAME WORDPRESS_DB_PASSWORD WORDPRESS_DB_USER)
wordpress-1  | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, usin
g 172.20.0.2. Set the 'ServerName' directive globally to suppress this message
wordpress-1  | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, usin
g 172.20.0.2. Set the 'ServerName' directive globally to suppress this message
wordpress-1  | [Fri Jul 12 08:24:19.476264 2024] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.59 (Debian
) PHP/8.2.21 configured -- resuming normal operations
wordpress-1  | [Fri Jul 12 08:24:19.476300 2024] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FO
REGROUND'
wordpress-1  | 192.168.56.1 - - [12/Jul/2024:08:28:10 +0000] "GET / HTTP/1.1" 302 409 "-" "Mozilla/5.0 (X11;
 Ubuntu; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
wordpress-1  | 192.168.56.1 - - [12/Jul/2024:08:28:10 +0000] "GET /wp-admin/install.php HTTP/1.1" 200 4665 "
-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
wordpress-1  | 192.168.56.1 - - [12/Jul/2024:08:28:12 +0000] "GET /wp-includes/css/dashicons.min.css?ver=6.5
.5 HTTP/1.1" 200 36068 "http://192.168.56.10:8000/wp-admin/install.php" "Mozilla/5.0 (X11; Ubuntu; Linux x86
_64; rv:128.0) Gecko/20100101 Firefox/128.0"
wordpress-1  | 192.168.56.1 - - [12/Jul/2024:08:28:12 +0000] "GET /wp-includes/css/buttons.min.css?ver=6.5.5
 HTTP/1.1" 200 1807 "http://192.168.56.10:8000/wp-admin/install.php" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64
; rv:128.0) Gecko/20100101 Firefox/128.0"
wordpress-1  | 192.168.56.1 - - [12/Jul/2024:08:28:12 +0000] "GET /wp-admin/css/install.min.css?ver=6.5.5 HT
TP/1.1" 200 2138 "http://192.168.56.10:8000/wp-admin/install.php" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; r
v:128.0) Gecko/20100101 Firefox/128.0"
```

6. Deploy the Application Using Docker Swarm

Docker Swarm is a native clustering and orchestration tool for Docker.

- Initialize Docker Swarm.

docker swarm init

- Convert the Docker Compose file to a Docker Stack file, docker-stack.yml.

```yaml
version: '3.3'

services:
  db:
    image: mysql:5.7
    volumes:
      - mysql_data:/var/lib/mysql
    networks:
      - wordpress_network
    environment:
      MYSQL_ROOT_PASSWORD: example
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress
    deploy:
      replicas: 1

  wordpress:
    image: wordpress:latest
    volumes:
      - wordpress_data:/var/www/html
    networks:
      - wordpress_network
    ports:
      - "8000:80"
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress
    deploy:
      replicas: 1

volumes:
  mysql_data:
  wordpress_data:

networks:
  wordpress_network:
```

- Deploy the stack using Docker Swarm.

```
docker stack deploy -c docker-stack.yml wordpress_stack
```

- Verify the stack is running.

docker stack services wordpress_stack

```
vagrant@ubuntu2204:~$ vim docker-stack.yml
vagrant@ubuntu2204:~$ docker stack deploy -c docker-stack.yml wordpress_stack
Since --detach=false was not specified, tasks will be created in the background.
In a future release, --detach=false will become the default.
Creating network wordpress_stack_wordpress_network
Creating service wordpress_stack_db
Creating service wordpress_stack_wordpress
vagrant@ubuntu2204:~$ docker stack services wordpress_stack
ID             NAME                      MODE         REPLICAS   IMAGE              PORTS
zqxot31tc5uc   wordpress_stack_db        replicated   1/1        mysql:5.7
kdkuyepg0csg   wordpress_stack_wordpress  replicated   0/1        wordpress:latest   *:8000->80/t
cp
vagrant@ubuntu2204:~$
```

Project 02:

Objectives:

- Deploy an application across multiple Docker Swarm worker nodes.
- Place specific components on designated nodes.
- Monitor and troubleshoot using Docker logs.
- Modify and redeploy the application.

Project Outline:

1. **Initialize Docker Swarm and Join Worker Nodes**
2. **Label Nodes for Specific Component Placement**
3. **Create a Docker Stack File**
4. **Deploy the Application**
5. **Monitor and Troubleshoot Using Docker Logs**
6. **Modify and Redeploy the Application**

Step-by-Step Guide

1. Initialize Docker Swarm and Join Worker Nodes

On the manager node, initialize Docker Swarm:

```
docker swarm init --advertise-addr <MANAGER-IP>
```

Join the worker nodes to the swarm. On each worker node, run the command provided by the `docker swarm init` output:

```
docker swarm join --token <SWARM-TOKEN> <MANAGER-IP>:2377
```

Verify the nodes have joined:

```
docker node ls
```

```
vagrant@ubuntu2204:~$ docker node ls
ID                            HOSTNAME                STATUS    AVAILABILITY   MANAGER STATUS   E
NGINE VERSION
yfuhn3za1dhhapxnh5qcrgdck     localhost.localdomain   Ready     Active                          2
7.0.3
pq87umlgb5kg6xlnszz1jn6po *   ubuntu2204.localdomain  Ready     Active         Leader           2
7.0.3
vagrant@ubuntu2204:~$
```

2. Label Nodes for Specific Component Placement

Label nodes to specify where certain components should run. For example, label a node for the database service:

```
docker node update --label-add db=true <NODE-ID>
```

Label another node for the application service:

```
docker node update --label-add app=true <NODE-ID>
```

Verify the labels:

```
docker node inspect <NODE-ID>
```

```
vagrant@ubuntu2204:~$ docker node update --label-add db=true yfuhn3za1dhhapxnh5qcrgdck
yfuhn3za1dhhapxnh5qcrgdck
vagrant@ubuntu2204:~$ docker node update --label-add app=true pq87umlgb5kg6xlnszz1jn6po
pq87umlgb5kg6xlnszz1jn6po
vagrant@ubuntu2204:~$ docker node inspect yfuhn3za1dhhapxnh5qcrgdck
[
    {
        "ID": "yfuhn3za1dhhapxnh5qcrgdck",
        "Version": {
            "Index": 1470
        },
        "CreatedAt": "2024-07-12T06:15:57.093782742Z",
        "UpdatedAt": "2024-07-12T08:45:49.161740081Z",
        "Spec": {
            "Labels": {
                "db": "true"
            },
            "Role": "worker",
            "Availability": "active"
        },
        "Description": {
            "Hostname": "localhost.localdomain",
            "Platform": {
                "Architecture": "x86_64",
                "OS": "linux"
```

3. Create a Docker Stack File

Create a docker-stack.yml file to define the services and node placement constraints:

```yaml
version: '3.8'

services:
  db:
    image: mysql:5.7
    volumes:
      - mysql_data:/var/lib/mysql
    networks:
      - app_network
    environment:
      MYSQL_ROOT_PASSWORD: example
      MYSQL_DATABASE: appdb
      MYSQL_USER: user
      MYSQL_PASSWORD: password
    deploy:
      placement:
        constraints:
          - node.labels.db == true

  app:
    image: your-app-image
    networks:
      - app_network
    ports:
      - "8000:80"
    environment:
      DB_HOST: db
```

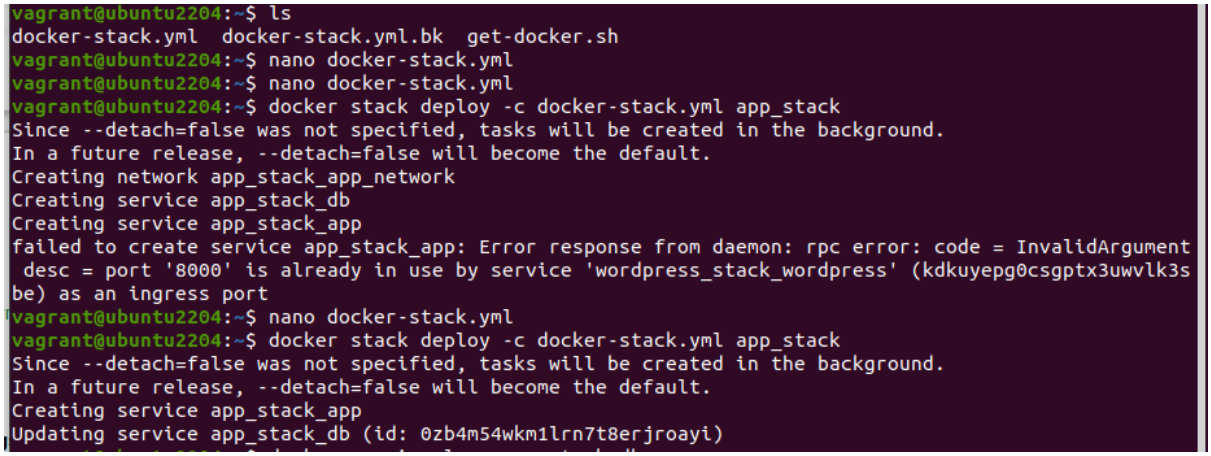
```
    deploy:
      replicas: 2
      placement:
        constraints:
          - node.labels.app == true
volumes:
  mysql_data:
networks:
  app_network:
```

4. Deploy the Application

Deploy the stack using Docker Swarm:

```
docker stack deploy -c docker-stack.yml app_stack
```

```
docker stack services app_stack
```



```
vagrant@ubuntu2204:~$ ls
docker-stack.yml  docker-stack.yml.bk  get-docker.sh
vagrant@ubuntu2204:~$ nano docker-stack.yml
vagrant@ubuntu2204:~$ nano docker-stack.yml
vagrant@ubuntu2204:~$ docker stack deploy -c docker-stack.yml app_stack
Since --detach=false was not specified, tasks will be created in the background.
In a future release, --detach=false will become the default.
Creating network app_stack_app_network
Creating service app_stack_db
Creating service app_stack_app
failed to create service app_stack_app: Error response from daemon: rpc error: code = InvalidArgument
 desc = port '8000' is already in use by service 'wordpress_stack_wordpress' (kdkuyepg0csgptx3uwvlk3s
be) as an ingress port
vagrant@ubuntu2204:~$ nano docker-stack.yml
vagrant@ubuntu2204:~$ docker stack deploy -c docker-stack.yml app_stack
Since --detach=false was not specified, tasks will be created in the background.
In a future release, --detach=false will become the default.
Creating service app_stack_app
Updating service app_stack_db (id: 0zb4m54wkm1lrn7t8erjroayi)
```

5. Monitor and Troubleshoot Using Docker Logs

Check the logs for the services:

```
docker service logs app_stack_db
docker service logs app_stack_app
```

```
vagrant@ubuntu2204:~$ docker stack services app_stack
ID             NAME            MODE         REPLICAS   IMAGE              PORTS
jtaw3ghkwo3b   app_stack_app   replicated   2/2        wordpress:latest   *:8001->80/tcp
0zb4m54wkm1l   app_stack_db    replicated   0/1        mysql:5.7
vagrant@ubuntu2204:~$ docker service logs app_stack_db
vagrant@ubuntu2204:~$ docker service logs app_stack_app
app_stack_app.1.4s2dtpg7186k@ubuntu2204.localdomain    | WordPress not found in /var/www/html - copyi
ng now...
app_stack_app.1.4s2dtpg7186k@ubuntu2204.localdomain    | Complete! WordPress has been successfully co
pied to /var/www/html
app_stack_app.1.4s2dtpg7186k@ubuntu2204.localdomain    | AH00558: apache2: Could not reliably determi
ne the server's fully qualified domain name, using 10.0.2.7. Set the 'ServerName' directive globally
to suppress this message
app_stack_app.1.4s2dtpg7186k@ubuntu2204.localdomain    | AH00558: apache2: Could not reliably determi
ne the server's fully qualified domain name, using 10.0.2.7. Set the 'ServerName' directive globally
to suppress this message
app_stack_app.1.4s2dtpg7186k@ubuntu2204.localdomain    | [Fri Jul 12 09:10:35.718615 2024] [mpm_prefo
rk:notice] [pid 1] AH00163: Apache/2.4.59 (Debian) PHP/8.2.21 configured -- resuming normal operation
s
app_stack_app.1.4s2dtpg7186k@ubuntu2204.localdomain    | [Fri Jul 12 09:10:35.718798 2024] [core:noti
ce] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
app_stack_app.1.4s2dtpg7186k@ubuntu2204.localdomain    | 10.0.0.2 - - [12/Jul/2024:09:11:55 +0000] "G
ET /wp-admin/install.php HTTP/1.1" 302 295 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:128.0) Gec
ko/20100101 Firefox/128.0"
app_stack_app.2.q02xma0oe37g@ubuntu2204.localdomain    | WordPress not found in /var/www/html - copyi
ng now...
```

Follow the logs in real-time to monitor issues:

```
docker service logs -f app_stack_app
```

6. Modify and Redeploy the Application

Make modifications to the application or the stack file as needed. For example, change the number of replicas:

```
services:
  app:
    deploy:
      replicas: 3
```

Update the stack with the new configuration:

```
docker stack deploy -c docker-stack.yml app_stack
```

Verify the changes:

```
docker stack services app_stack
```

```
vagrant@ubuntu2204:~$ nano docker-stack.yml
vagrant@ubuntu2204:~$ docker stack deploy -c docker-stack.yml app_stack
Since --detach=false was not specified, tasks will be created in the background.
In a future release, --detach=false will become the default.
Updating service app_stack_db (id: 0zb4m54wkm1lrn7t8erjroayi)
Updating service app_stack_app (id: jtaw3ghkwo3b2kt63ri6cft7d)
vagrant@ubuntu2204:~$ docker stack services app_stack
ID             NAME            MODE         REPLICAS   IMAGE              PORTS
jtaw3ghkwo3b   app_stack_app   replicated   3/3        wordpress:latest   *:8001->80/tcp
0zb4m54wkm1l   app_stack_db    replicated   0/1        mysql:5.7
vagrant@ubuntu2204:~$
```