HARSH KUMAR-21UCS086

Report on Issues Faced and Resolutions in Docker Application Deployment
Introduction

This report outlines the various issues encountered during the deployment of a Dockerized application, specifically focusing on the Dockerfiles and configuration files for both the Python and Nginx components. It details the errors identified, the corrective actions taken, and the successful outcomes achieved.

Issues Faced

1. Upon cloning the provided GitHub repository, I encountered several syntax errors within the Dockerfiles located in both the Python and Nginx folders, which impeded the successful build and execution of the application.

2. **1. Python Dockerfile Issues:**
    - **Incorrect Syntax:** The line `WORKDIR/appp` was incorrectly specified. This was corrected to `WORKDIR/app`.
    - **File Copying Errors:** The command `COPY appp.py/app` contained a typographical error; it was rectified to `COPY app.py/app`.
    - **Package Name Typo:** The package name `netiface` was corrected to `netifaces`.
    - **Port Specification:** The port `8000` was originally formatted as a string; this was corrected to an integer format.
    - **CMD Syntax Error:** The command `CMD ["python", "app.py"]` was corrected for proper syntax.

    2. Nginx Dockerfile Issues:
    - **Syntax Errors:** In the Nginx Dockerfile, the term `latests` was corrected to `latest`, and directory paths were updated accordingly.
    - **Port Exposure Syntax:** The syntax for `EXPOSE` was changed from `eighty` to `80`.
    - **CMD Syntax Correction:** The command was changed from `daemon of` to `daemon off`.

    3. Directory Structure Issue:
    - Following the Nginx build, an error indicated the absence of a directory named `html` at the specified path. In response, I created an `html` directory within the Nginx folder and added a basic HTML file that displayed the message `"Hello User!"` to resolve this issue.

    4. Nginx Configuration File Errors:
    - An error was identified in the `nginx.conf` file at `line 3`, specifically regarding the directive `worker_process`. Subsequent examination revealed several syntax errors:
        - The directive `worker_connection` was corrected to `worker_connections`.
        - The line `include /etc/nginx/mime.typess;` was updated to `include /etc/nginx/mime.types;`.
        - The directive `default_typ` was corrected to `default_type`.

    5. `docker-compose.yaml` Errors:
        - Ports in services were correctly mapped to `80`
        - `./nginx/nginx.conf:/etc/nginx/nginx.conf`, volume mapping.

HARSH KUMAR-21UCS086
Report on Issues Faced and Resolutions in Docker Application Deployment
Introduction

Resolution Steps

1. Building the Python Image:
- After rectifying the aforementioned issues in the Python Dockerfile, I executed the command `docker build -t local-python-app`, resulting in a successful image build.

2. Building the Nginx Image:
- Following the corrections made to the Nginx Dockerfile, I attempted to build the image. Upon addressing the directory structure issue and verifying the contents of the `nginx.conf`, the Nginx image was successfully built.

3. Container Management:
- After resolving all issues, I rebuilt and restarted the containers. I verified their status by executing `docker ps`.
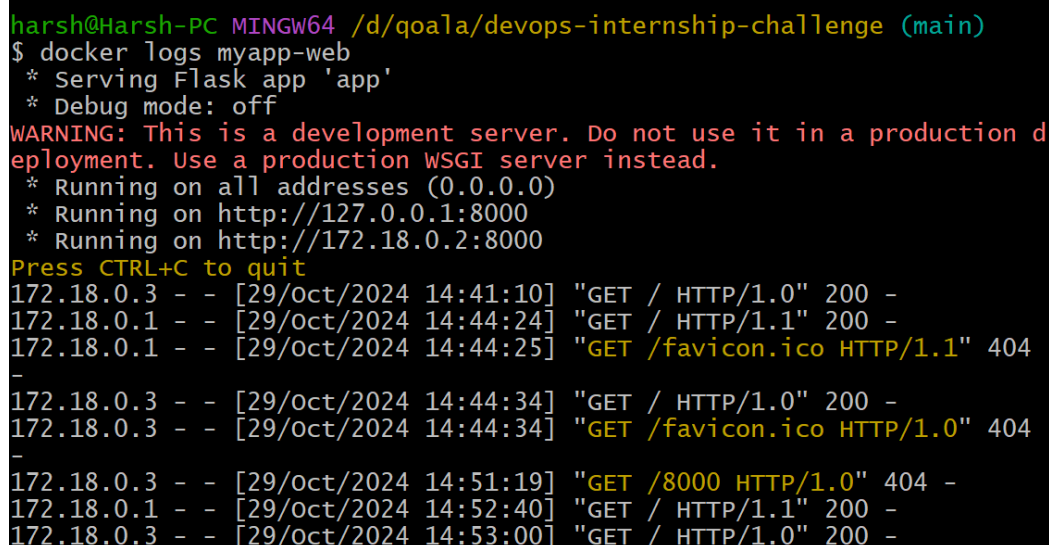
4. Testing Application Communication:
- I tested the communication between the Nginx and Python application by executing `curl http://localhost:80`. This confirmed that the Nginx server was functioning correctly and forwarding requests to the Python application.

5. Logging Verification:
- Finally, I reviewed the logs for both the Nginx container using `docker logs local-nginx` and for the Python application using `docker logs local-python-app` to ensure that there were no further errors and that the applications were running as expected.

SCREENSHOTS

```
harsh@Harsh-PC MINGW64 /d/qoala/devops-internship-challenge (main)
$ docker logs myapp-web
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production d
eployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8000
 * Running on http://172.18.0.2:8000
Press CTRL+C to quit
172.18.0.3 - - [29/Oct/2024 14:41:10] "GET / HTTP/1.0" 200 -
172.18.0.1 - - [29/Oct/2024 14:44:24] "GET / HTTP/1.1" 200 -
172.18.0.1 - - [29/Oct/2024 14:44:25] "GET /favicon.ico HTTP/1.1" 404
-
172.18.0.3 - - [29/Oct/2024 14:44:34] "GET / HTTP/1.0" 200 -
172.18.0.3 - - [29/Oct/2024 14:44:34] "GET /favicon.ico HTTP/1.0" 404
-
172.18.0.3 - - [29/Oct/2024 14:51:19] "GET /8000 HTTP/1.0" 404 -
172.18.0.1 - - [29/Oct/2024 14:52:40] "GET / HTTP/1.1" 200 -
172.18.0.3 - - [29/Oct/2024 14:53:00] "GET / HTTP/1.0" 200 -
```
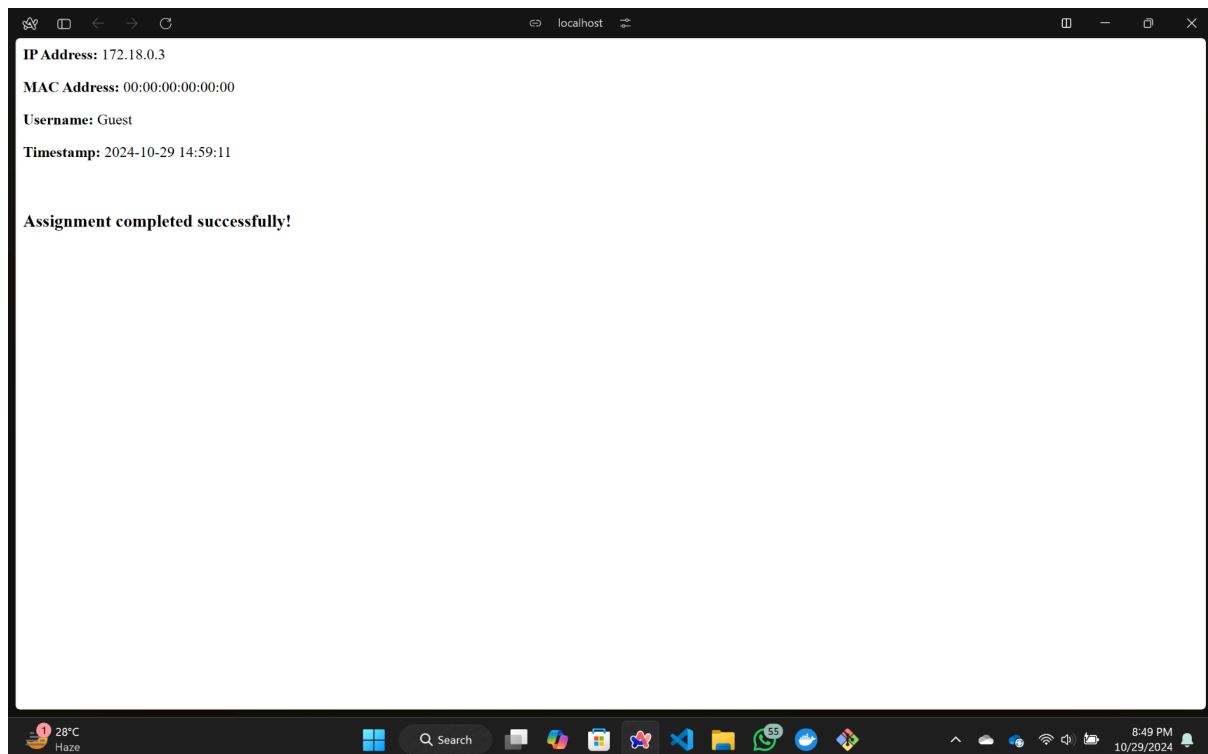
*LOGS OF local-python-app

Report on Issues Faced and Resolutions in Docker Application Deployment
Introduction

*Building the docker-compose file.

*APPLICATION RUNNING IN THE BROWSER