DevOps Assignment Report

Roll-No :-> 21UCS137

Name:-> Nayan Goyal

Issues Identified:

1.) Python DockerFile Errors:

- Incorrect filenames in the Dockerfile (e.g., COPY appy.py instead of COPY app.py) caused build failures.
- The command to run the application was incorrectly specified as CMD ["pythn", "app.py"].

2.) Nginx Configuration Issues:

- The nginx.conf file contained typos like “mime.typess” and “default_typ”, which hindered Nginx from starting correctly.
- Missing upstream server configuration for proper request routing.
- Non-existent image tag(“nignx:latests” instead of “nginx:latest”)

3.) Docker Compose Configuration Problems:

- Incorrect port mappings, such as - "eighty:80" instead of - "80:80", led to connection failures.
- Misspellings in the network options affected container communication.

4.) MAC Address Retrieval: The original function for retrieving the MAC address lacked adequate error handling.

Resolution Steps:

1.) Correcting the Python Dockerfile:

- Updated the Dockerfile to use the correct filename (COPY app.py /app) and fixed the command to launch the application (CMD ["python", "app.py"]).
- Ensured the working directory was correctly set to /app and confirmed the installation of necessary packages.

2.) Fixing Nginx Configuration:

- Corrected all typographical errors in the “nginx.conf” file and added an upstream block to define the backend Python application server. (like: “nginx:latest”)
- Created the missing html directory and ensured proper files were added.

3.) Updating Docker Compose Configuration:

- Fixed port mappings in docker-compose.yaml to use correct syntax (e.g., 80:80).
- Corrected spelling mistakes in the network configuration for proper communication.

4.) Enhancing Error Handling:

- Improved the MAC address retrieval function to handle exceptions more gracefully, ensuring errors were communicated effectively.

5.) Application Verification:

- Accessed the application at http://localhost:80 and verified functionality successfully.