DEVOPS ASSIGNMENT REPORT

Prajit Kaushik - 21ucs151

Project Overview:
The objective was to deploy a Python Flask application served by an Nginx server using Docker and Docker Compose. This report highlights the issues encountered during image building, container setup, and application testing, along with the steps taken to resolve them.

1. Issues Identified

- **Python App Image Building Errors**
  - **Typo with exposing port "eight thousand" and command cmd "pythn".**
  - **Work Directory Setup:** The `WORKDIR` command referenced an incorrect path (`WORKDIR /appp` instead of `WORKDIR /app`), causing issues in file referencing when the application tried to start.
  - **File Path Errors:** The `COPY` command in the Dockerfile (`COPY app.py /app`) referenced a path that was either incorrect or the file was missing locally, leading to build failures when Docker couldn't find the specified files.
  - **Python Package Installation Errors:** While attempting to install packages (`flask` and `netiface`), errors occurred:
    - `netiface` was unavailable for the specified Python version in the image, causing the `pip install` command to fail.
    - Additionally, errors were encountered when the `pip` version needed an update for compatibility with the packages.

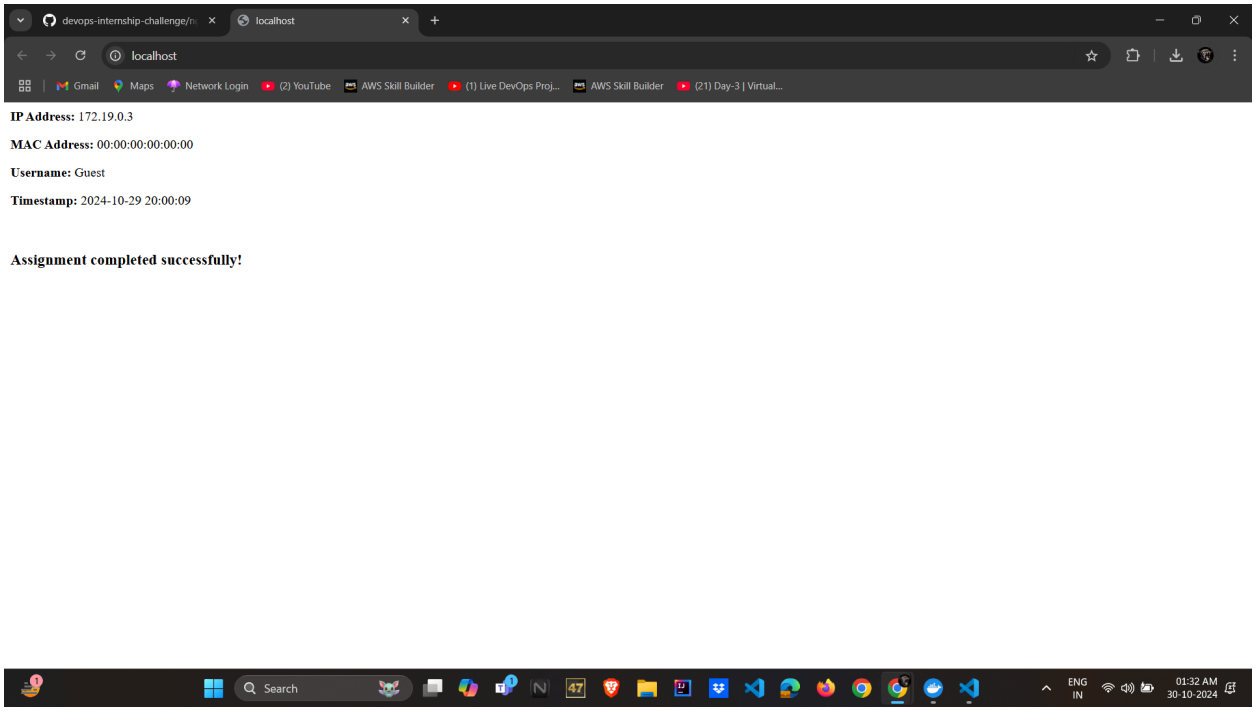
- **Nginx Image Building Errors**
  - **File Path Errors:** The `COPY` commands in the Dockerfile referenced incorrect paths, such as `COPY ./html /usr/share/nginx/html`, which caused build failures when the specified files or directories (like `html`) were missing on the local system.
  - **Nginx Configuration Errors:** Syntax errors in `nginx.conf` prevented the container from starting. For example:
    - Missing semicolon in the `worker_processes` directive (`worker_processes auto` instead of `worker_processes auto;`).
    - Incorrect path in the `include` directive (e.g., `include /etc/nginx/mime.typess;` instead of `include /etc/nginx/mime.types;`).
    - Misconfigured directives (e.g., `proxy_pass` was not correctly pointing to the Flask service).
    - Image Tag Typos: Typos in tags or image names, like `nginx:latests` instead of `nginx:latest`, caused build errors since Docker couldn't locate the specified image.

- ❖ `Docker-compose up` failed to run the container due to syntax errors and incorrect configurations in Dockerfiles (e.g., missing or incorrect paths, typos,incorrect mapping like eighty:80 or eight thousand).

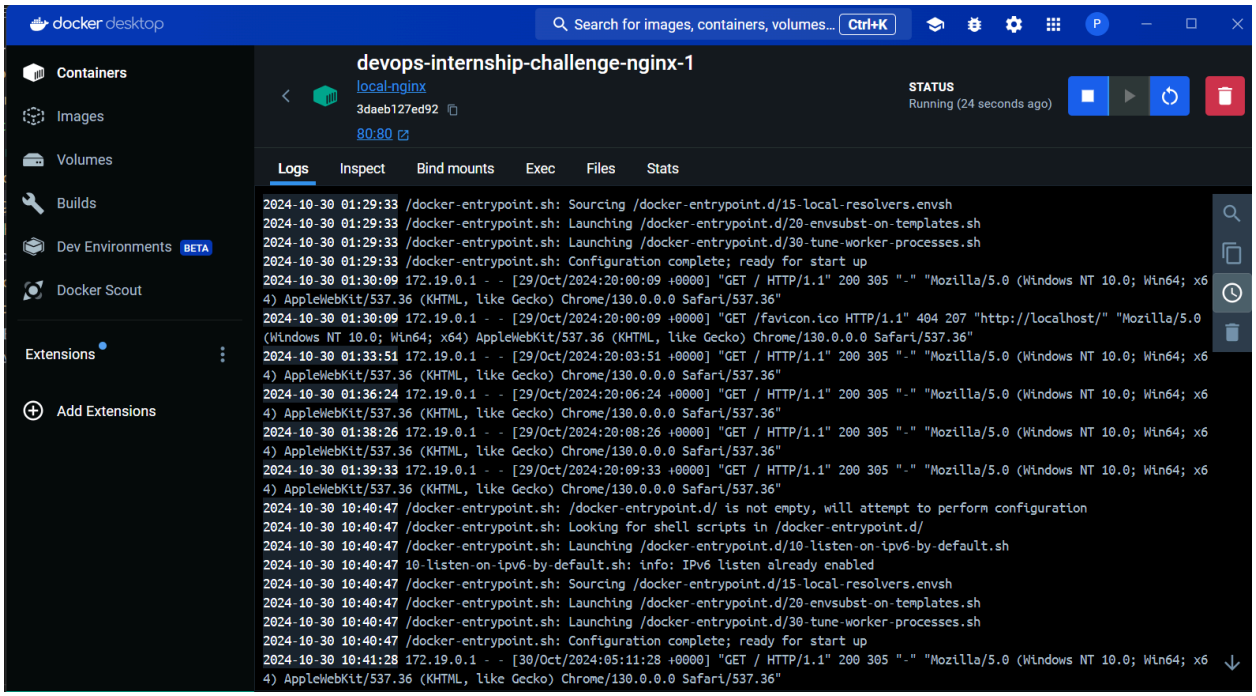
2.Resolutions Steps:

- Verified and corrected paths for `COPY` directives and ensured necessary files were present.
- Fixed typos like “Eighty:80” and “eigth thousand: 8000” in docker compose and docker files.
- Removed options from networks in docker compose
- Fixed configuration syntax errors in `nginx.conf`, tested with `nginx -t` for validation.
- Used correct image tags and names to avoid mismatches during build.
- Verified file paths and adjusted `COPY` commands to reference accurate directories, ensuring `app.py` was copied to the correct location.
- Fixed typo of `netiface` package and updated the `pip` version for package compatibility.
- Corrected the `WORKDIR` path to `/app` for consistency and ensured the application ran from the intended directory.
- Corrected syntax in Dockerfiles, including accurate file paths and command structures.
- To support a static webpage, an HTML file was incorporated into the project for the Nginx container to serve as a front-end interface

RUNNING SERVER SCREENSHOT



NGINX LOGS SCREENSHOT



APPLICATION HOSTING

DEPLOYED APPLICATION ON : **AWS**
HOSTED IP : 13.60.31.109