

Report on Docker Configuration Corrections

Introduction

This report outlines the issues encountered during the setup of the Docker environment for the application and the subsequent corrections made to resolve them. The goal was to ensure that both the Python application and the Nginx server run smoothly within Docker containers.

Issues Identified

Throughout the setup process, several issues were discovered:

1. Python Dockerfile Issues:

- The working directory was incorrectly set to /appp, instead of the correct /app.
- The file referenced in the COPY command was incorrectly named appy.py; it should be app.py.
- The package name for installation was mistyped as netiface instead of netifaces.
- The EXPOSE command had an incorrect format, where it was set to "eight thousand" instead of 8000.
- There was a typo in the command to run the application, using "pythn" instead of "python".

2. Nginx Dockerfile Issues:

- The image tag was misspelled as latests; the correct tag is latest.
- The configuration file was incorrectly named nginx.conf in the COPY command, which should be nginx.conf.
- The destination path for copying the HTML files was incorrectly specified as /usr/share/nginx/html instead of /usr/share/nginx/html.
- The EXPOSE command also had an incorrect format, using "eighty" instead of 80.
- The command to run Nginx had a formatting error: it was set as "daemon of;", which should be "daemon off;".

3. Nginx Configuration File Issues:

- The directive worker_process was missing the pluralization and should be worker_processes.
- The directive worker_connection should be worker_connections.
- The path in the include statement was incorrectly specified as /etc/nginx/mime.typeess instead of /etc/nginx/mime.types.
- The directive for default_typ was misspelled and should be default_type.

4. Docker Compose Configuration Issues:

- The Nginx service incorrectly referenced an image instead of a build context.
- Port mapping was incorrectly specified as "eighty:80" rather than "80:80".
- The volume path for the Nginx configuration had a typo (nginx.confi instead of nginx.conf).
- The exposed port format was again incorrectly written as "eight thousand" instead of 8000.
- The network driver was misspelled as bridg, which should be bridge.
- An irrelevant option key compelex_option was present and needed removal.

5. HTML Directory and Index File:

- An HTML directory was missing, which is essential for the Nginx service.

- There was no index.html file to serve as the landing page for the application.

Corrections Made

To address these issues, the following corrections were implemented:

1. Python Dockerfile

```
FROM python:3.9

WORKDIR /app

COPY app.py /app

RUN pip install flask netifaces

EXPOSE 8000

CMD ["python", "app.py"]
```

2. Nginx Dockerfile

```
FROM nginx:latest
COPY nginx.conf /etc/nginx/nginx.conf
COPY ./html /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

3. Nginx Configuration File

```
worker_processes auto;

events {
    worker_connections 1024;
}

http {
    include      /etc/nginx/mime.types;
    default_type application/octet-stream;

    server {
        listen 80;
        server_name localhost;

        location / {
            proxy_pass http://python_app:8000;
            proxy_set_header Host $host;
```

```

    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
  }
}
}

```

4. Docker Compose Configuration

```

version: '3.8'

services:
  nginx:
    build:
      context: ./nginx
    ports:
      - "80:80"
    depends_on:
      - python-app
    networks:
      - app-network

  python-app:
    build:
      context: ./Python
    container_name: python_app
    expose:
      - "8000"
    networks:
      - app-network

networks:
  app-network:
    driver: bridge

```

5. HTML Directory and Index File

- **Created an HTML Directory:** A directory named html was created to store static HTML files for the application.
- **Added an index.html File:** An index.html file was added within the html directory. This file includes dynamic content generated by the Flask application, such as user IP address, MAC address, username, and timestamp.

Conclusion

These corrections ensured that the Docker environment was properly configured for the application to build and run without issues. The addition of the HTML directory and the

index.html file provided essential content for the Nginx server to serve, allowing the application to function as intended. This process highlights the importance of attention to detail in configuration files and demonstrates effective troubleshooting steps taken to resolve the identified problems.