**DevOps Assignment Report**

**Issues Identified:**

1. **Docker Image Not Found:** While launching the services using docker-compose.yaml, the specified images (nginxapp and python\_app) were not found locally, likely due to missing or incorrectly tagged Dockerfiles.
2. **Port Conflicts or Accessibility:** Potential issues could arise if port 80 (mapped for nginxapp) or port 8000 (exposed by python\_app) are in use or misconfigured, which might prevent access to the application.
3. **Configuration Path Mismatch:** The volume mapping ./nginx/nginx.conf:/etc/nginx/nginx.conf assumes an nginx.conf file exists in the specified directory, which may cause startup errors if the path is incorrect or the file is missing.
4. **Network Misconfiguration:** The nginx-network network configuration might cause issues if not correctly created or if there are network conflicts.

**Resolution Steps:**

1. **Image Building and Tagging:** Built Docker images for nginxapp and python\_app based on the respective Dockerfiles. Ensured correct tagging (nginxapp:latest and python\_app:latest) to match the docker-compose.yaml requirements.
2. **Port and Volume Verification:** Verified that ports 80 and 8000 are open and not used by other applications. Also checked the volume path ./nginx/nginx.conf to ensure the Nginx configuration file is available and correctly mapped.
3. **Network Setup:** Created the nginx-network before starting the containers to ensure all services could communicate. Verified network settings to avoid IP conflicts and confirmed bridge mode functionality.
4. **Testing and Verification:** After resolving the above issues, restarted the containers using docker-compose up. Accessed the application at http://localhost to confirm functionality and checked Nginx logs to verify successful requests.

This approach helped ensure the application ran smoothly, resolving initial errors and confirming container accessibility.