

AeroAspire - SDE Intern

Gokul Krishna S

Week 5 – Day 3 (October 24)

Questions/Reflections :

1. How Docker Compose manages service networks

- Docker Compose automatically creates a default network where all services (like Flask, DB, Redis, etc.) can talk to each other by name.
- For example, your Flask app can connect to the database just by using its service name (db) instead of an IP.
- You can also define custom networks if you want more control — but by default, everything in the same docker-compose.yml shares one virtual network.

2. What happens when a container crashes (how to check logs)

- If a container crashes, it just stops running — Docker doesn't delete it automatically.
- You can check what went wrong using:
 - `docker ps -a` # to see stopped containers
 - `docker logs <container_name>` # to view the error output
- If it keeps restarting, you can watch it live using:
 - `docker logs -f <container_name>`

3. What environment variables are needed to connect Flask to DB?

- Flask usually needs a few key environment variables for database connection:
- **DB_HOST** → where the database is running (like db or localhost)

- **DB_PORT** → database port (like 5432 for Postgres or 3306 for MySQL)
- **DB_USER** → database username
- **DB_PASSWORD** → password for that user
- **DB_NAME** → name of the database
- In code, Flask uses them to build the connection string, for example:
- `SQLALCHEMY_DATABASE_URI=f"postgresql://{DB_USER}:{DB_PASSWORD}@{DB_HOST}:{DB_PORT}/{DB_NAME}"`