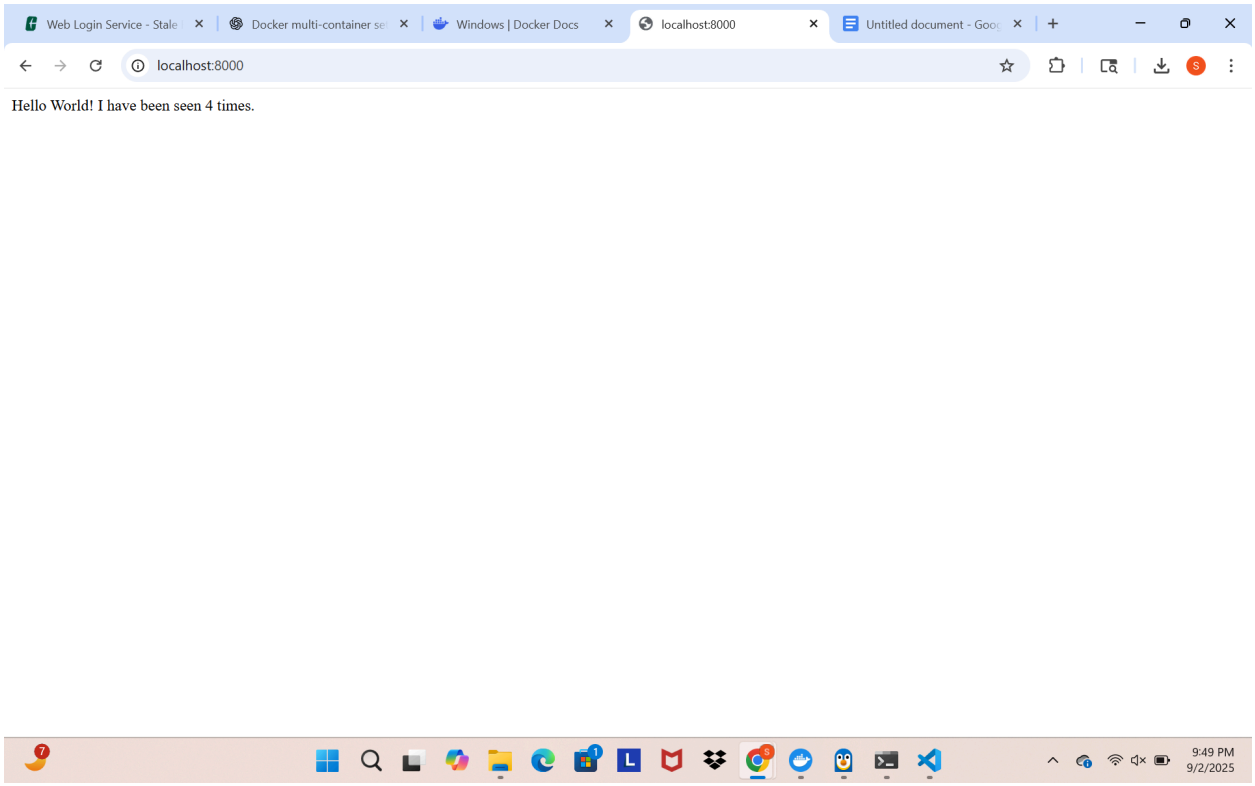
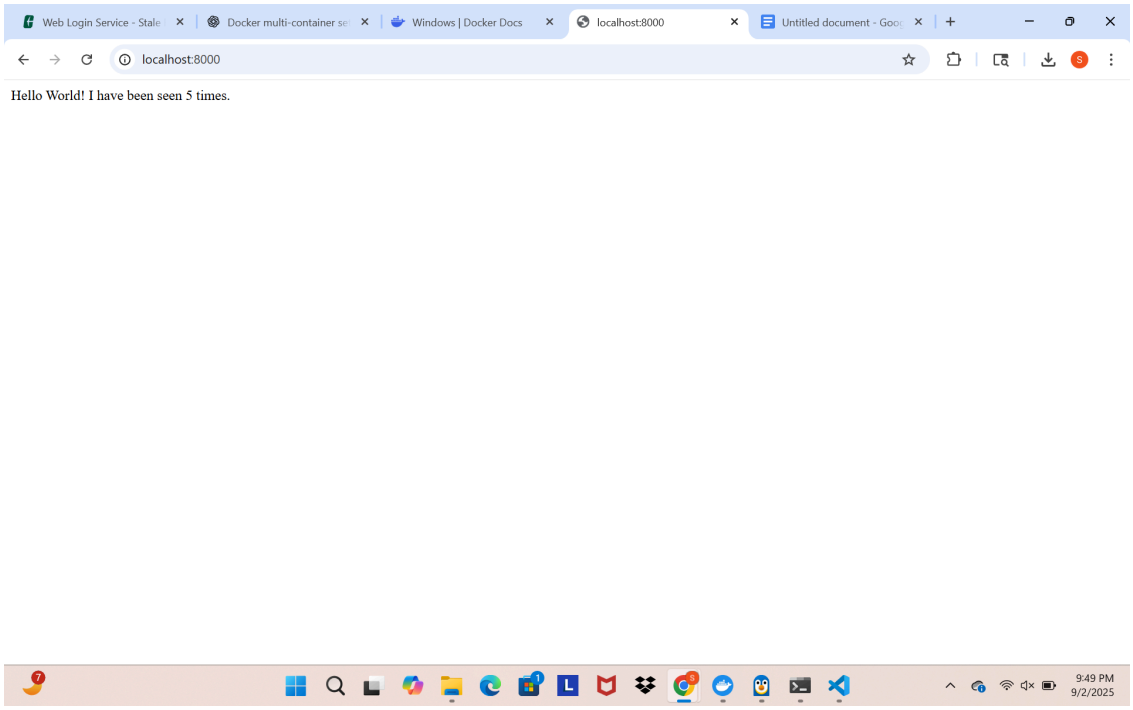
Docker L3 – Multi-Container Microservice (Flask + Redis + Postgres)

Sai Harsha Vunnava
801418991

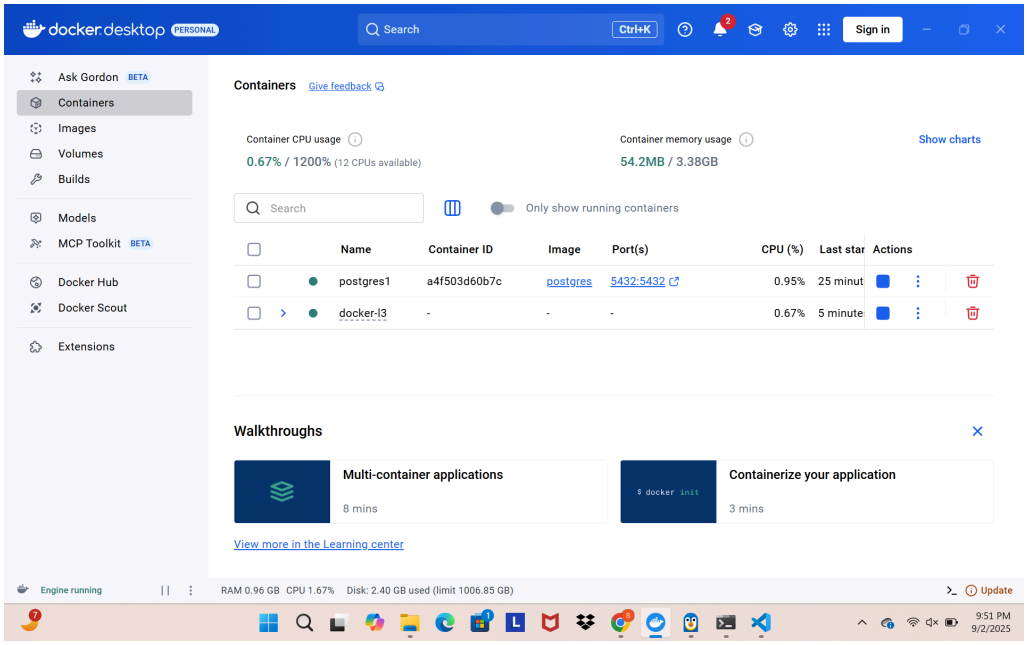
1.) Browser Output – Flask App with Redis Counter
This screenshot shows the Flask application running on http://localhost:8000.



The app is correctly connected to Redis, and the counter is increasing on each refresh.

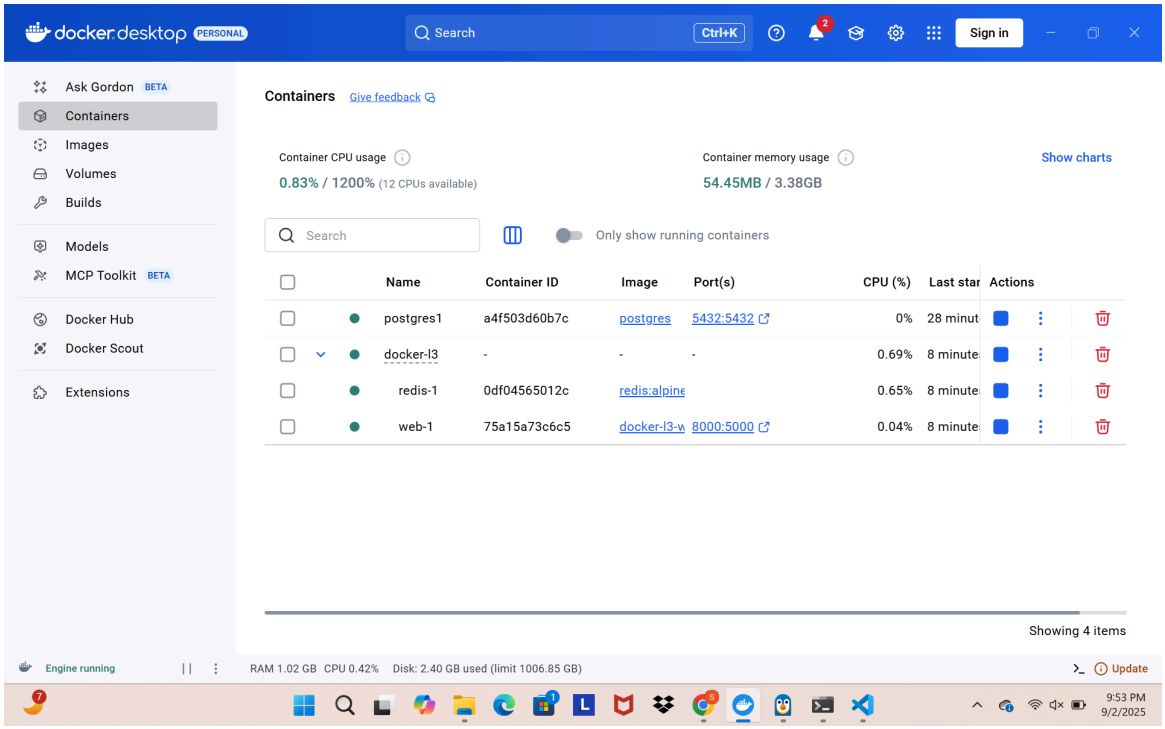


Docker Desktop



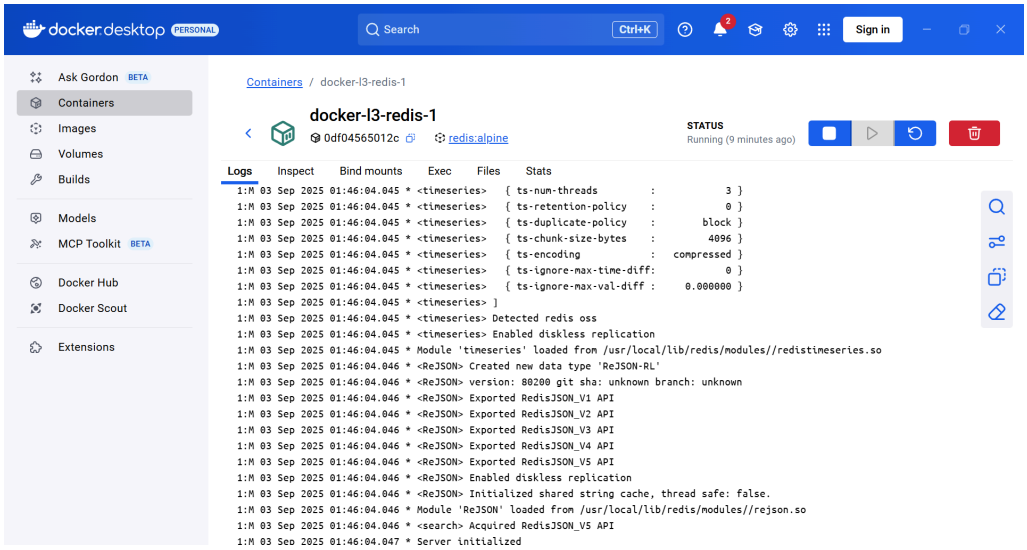
`postgres1` → PostgreSQL database container (port `5432`).

`docker-l3` → Project group containing the Flask and Redis services.



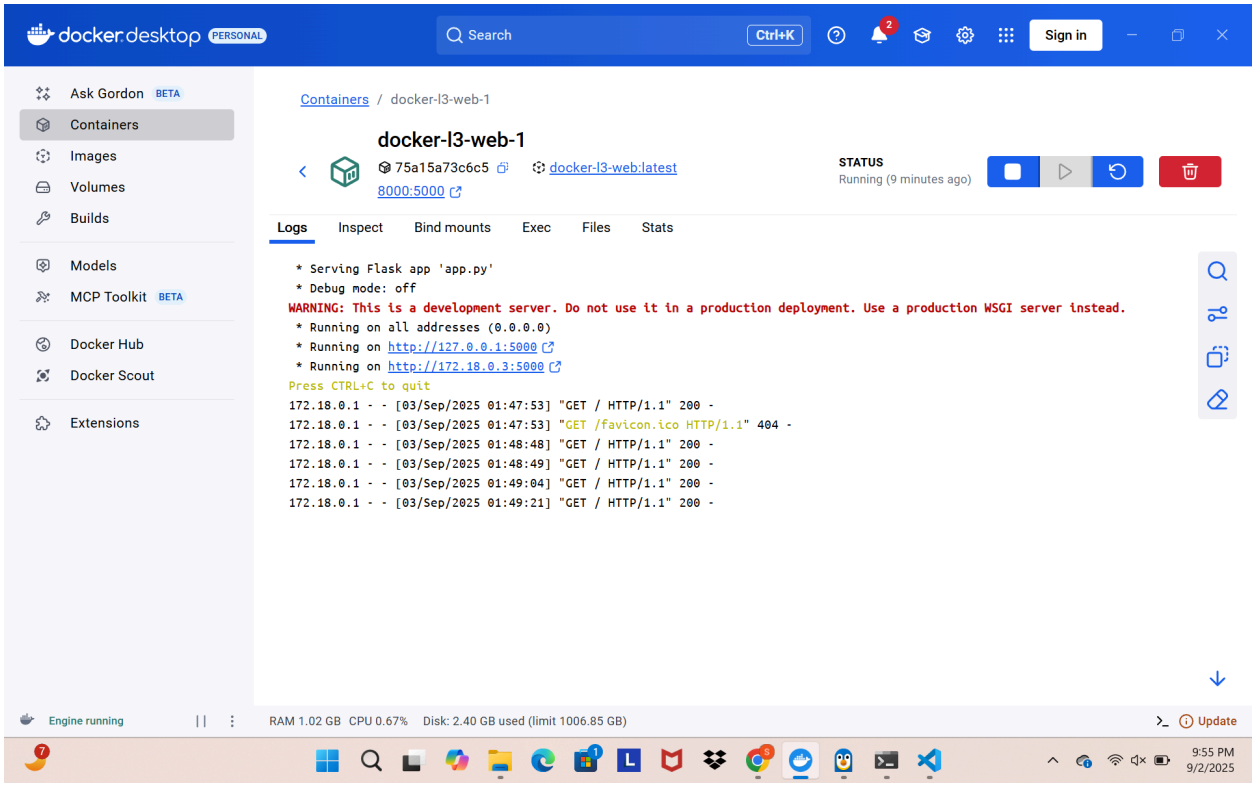
This screenshot confirms that all containers are running together:

- **postgres1** – PostgreSQL database (port `5432`)
- **redis-1** – Redis cache (default port `6379`)
- **web-1** – Flask web application (mapped to port `8000:5000`)



The following screenshot shows the logs for the **docker-l3 → redis** container.
It confirms that:

- Redis modules (`timeseries`, `ReJSON`) were loaded successfully.
- Diskless replication was enabled.
- The Redis server initialized and is ready to accept connections.



The Flask application (`app.py`) started successfully.

The app is listening on all addresses (`0.0.0.0`) and mapped to port `5000` internally.

External requests to `http://localhost:8000` are reaching the container and being served (`GET / HTTP/1.1" 200`).

```
C:\Users\saiha\docker-l3>docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED          STATUS          PORTS
  NAMES
75a15a73c6c5   docker-l3-web   "flask run"              14 minutes ago   Up 13 seconds   0.0.0.0:8000->5000/tcp, [::]:8000->5000/tcp
  docker-l3-web-1
0df04565012c   redis:alpine    "docker-entrypoint.s…"   14 minutes ago   Up 13 seconds   6379/tcp
  docker-l3-redis-1
a4f503d60b7c   postgres        "docker-entrypoint.s…"   34 minutes ago   Up 34 minutes   0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp
  postgres1
```

The following screenshot shows the output of the `docker ps` command.

- **docker-l3-web** → Flask application container, mapped to port **8000** → **5000**
- **docker-l3-redis** → Redis service, running on port **6379**
- **postgres1** → PostgreSQL database, mapped to port **5432** → **5432**

This confirms that **all three services (Web, Redis, PostgreSQL)** are up and running simultaneously, which validates the successful execution of the multi-container setup using Docker Compose.