

## Individual Assignment 3

### Goal

In this assignment, you will containerize your project from the updated version of Individual Assignment 2 into three separate components:

- FastAPI server (backend)
- Streamlit app (frontend)
- Crontab job (scheduler)

You will test this setup in two environments:

1. Locally (using Docker Compose) → 3 containers (FastAPI, Streamlit, Crontab).
2. On Google Cloud → 2 containers (FastAPI + Streamlit on Cloud Run), and Cloud Scheduler (instead of crontab).
  - We will build and push the image to Docker Hub, and then run the container on Google Cloud Run. The cron job will be executed through Cloud Scheduler.

The crontab/scheduler will regularly call your FastAPI server's `'/search_and_save/jobs'` endpoint, which triggers Google Custom Search and stores the output. The Streamlit app will display the collected information.

For this assignment, you won't change any Python codes and use the provided ones.

### Testing Setup

We will test your solution in two steps:


1. Local test : Run with Docker Compose.

```
$ docker compose build
```

```
$ docker compose up
```

This should start all three containers: FastAPI, Streamlit, and Crontab.

2. Cloud test : Push your FastAPI and Streamlit Docker images to Docker Hub and then deploy them on Google Cloud Run. The scheduled job will be handled by Cloud Scheduler (instead of your local crontab). We will use your **api-cron** file to set the schedule.

 Note: Docker Compose is not officially supported on GCP, so Cloud Run will only use your FastAPI and Streamlit containers.

### Details & Requirements

#### 1. Project Structure

Your repo should include three folders as provided:

```
fastapi/  
streamlit/  
crontab/  
docker-compose.yml
```

Each folder must have its own Dockerfile.

Use the provided environment.yml files to install dependencies.

Define environment variables in a .env file. (Testing will be done with my .env files)

### 2. Local Containers (0.75 pt)

Each folder includes an extended version of HW2 solution, where now we use ' / search\_and\_save/jobs' that merges /search/jobs and /save\_to\_gcs in Individual Assignment 2. Without modifying any Python codes,

- A. Complete all **three Dockerfiles** (for FastAPI, Streamlit, Crontab).
- B. Complete **docker-compose.yml** to orchestrate them. While streamlit is independent, crontab should call fastapi's endpoint regularly.
- C. **api-cron** : This file should include crontab to call your FastAPI /search\_and\_save/jobs route every midnight.

### 3. Cloud Deployment (0.75 pt)

We will use the same Dockerfiles for FastAPI and Streamlit as Local Container in the Step2, but build them with the '--platform linux/amd64' option. We will build Dockerfile in /fastapi and /streamlit and push them to Docker Hub. The two containers will be deployed to Google Cloud Run.

For crontab, instead of running locally, you will use Cloud Scheduler.

- A. Cloud Scheduler should call your FastAPI server's /search\_and\_save/jobs route **every midnight**.

### 4. API Cron Job (0.5 pt)

Complete the api\_cron file in the crontab folder. (For Cloud Deployment, we will check the syntax and translate it to Cloud Run - where API\_SERVER is .)

The file should schedule a POST request to your FastAPI route (**every midnight**) :


Ex. `curl -X POST -H "Content-Type: application/json" -d '{"no_days": 5, "job_title": "engineer", "company_dict": {"Meta": "https://www.metacareers.com/jobs", "Microsoft": "https://careers.microsoft.com/", "Google": "https://www.google.com/about/careers/applications/jobs"}}' $API_SERVER/search_and_save/jobs`

 Note: API\_SERVER should be a complete URL with a port (if it uses a non-default port).

### 5. Files to Submit (total: 2.75 pts)

Submit exactly 5 files:

- A. api\_cron (0.5 pt)
- B. Dockerfile\_fastapi (0.5 pt)
- C. Dockerfile\_streamlit (0.5 pt)
- D. Dockerfile\_crontab (0.5 pt)
- E. docker-compose.yml (0.5 pt)

 Note: Rename each Dockerfile with a suffix (\_fastapi, \_streamlit, \_crontab). During testing, we will rename them back to Dockerfile in the correct folders.

### 6. Testing Notes

We will place your submitted files into the correct folders:

- A. api\_cron → crontab/
  - B. Dockerfiles → fastapi/, streamlit/, crontab/
  - C. docker-compose.yml → top level.
- We will then run local and cloud tests as described above.

**Please see the provided URLs as a reference.**

**Streamlit :** <https://dwoodbridge-hw3-streamlit-477009951698.europe-west1.run.app>

**FastAPI:** <https://dwoodbridg-hw3-fastapi-477009951698.europe-west1.run.app>