Data Engineer Bootcamp (Full-Time)

<!DOCTYPE html>

WeCloudData

Docker Compose - Flask App (Beginner Level)

Data Engineering Diploma Program

Content developed by: WeCloudData Academy

Reference

> - Reference

Prerequisites

Make sure you have already installed both Docker Engine and Docker Compose. You don’t need to install Python or Redis, as both are provided by Docker images.

1. Set Up

1.1 - Create a directory for the project

```
mkdir composetest
cd composetest
```

1.2 - Create application file `app.py`

> Create a file called `app.py` in your project directory and paste this in:

```
import time

import redis
from flask import Flask

app = Flask(__name__)
cache = redis.Redis(host='redis', port=6379)


def get_hit_count():
    retries = 5
    while True:
        try:
            return cache.incr('hits')
        except redis.exceptions.ConnectionError as exc:
            if retries == 0:
                raise exc
            retries -= 1
            time.sleep(0.5)


@app.route('/')
def hello():
    count = get_hit_count()
    return 'Hello World! I have been seen {} times.\n'.format(count)
```

1.3 - Create the `requirements.txt` file

> Create another file called requirements.txt in your project directory and paste this in:

```
flask
redis
```

2. Create `Dockerfile`

In this step, you write a Dockerfile that builds a Docker image. The image contains all the dependencies the Python application requires, including Python itself.

> In your project directory, create a file named Dockerfile and paste the following:

```
FROM python:3.7-alpine
WORKDIR /code
ENV FLASK_APP app.py
ENV FLASK_RUN_HOST 0.0.0.0
RUN apk add --no-cache gcc musl-dev linux-headers
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt
COPY . .
CMD ["flask", "run"]
```

This tells Docker to:

- Build an image starting with the Python 3.7 image.
- Set the working directory to `/code` .
- Set environment variables used by the flask command.
- Install gcc so Python packages such as MarkupSafe and SQLAlchemy can compile speedups.
- Copy `requirements.txt` and install the Python dependencies.
- Copy the current directory `.` in the project to the workdir `.` in the image.
- Set the default command for the container to `flask run` .

3. Define services in a Compose file

> Create a file called `docker-compose.yml` in your project directory and paste the following:

```
version: '3'
services:
  web:
    build: .
    ports:
      - "5000:5000"
  redis:
    image: "redis:alpine"
```

✱ 4. Build and run your app with Compose

4.1 - Build the app with docker compose

> From your project directory, start up your application by running `docker-compose up`.

```
docker-compose up
```

logs

```
Creating composetest_web_1 ...
Creating composetest_redis_1 ...
Creating composetest_web_1
Creating composetest_redis_1 ... done
Attaching to composetest_web_1, composetest_redis_1
web_1    |  * Running on c (Press CTRL+C to quit)
redis_1  | 1:C 17 Aug 22:11:10.480 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
redis_1  | 1:C 17 Aug 22:11:10.480 # Redis version=4.0.1, bits=64, commit=00000000,   modified=0, pid=1, just started
redis_1  | 1:C 17 Aug 22:11:10.480 # Warning: no config file specified, using the default config. In order to specify a config
web_1    |  * Restarting with stat
redis_1  | 1:M 17 Aug 22:11:10.483 * Running mode=standalone, port=6379.
redis_1  | 1:M 17 Aug 22:11:10.483 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/som
web_1    |  * Debugger is active!
redis_1  | 1:M 17 Aug 22:11:10.483 # Server initialized
redis_1  | 1:M 17 Aug 22:11:10.483 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will c
web_1    |  * Debugger PIN: 330-787-903
redis_1  | 1:M 17 Aug 22:11:10.483 * Ready to accept connections  ```
```

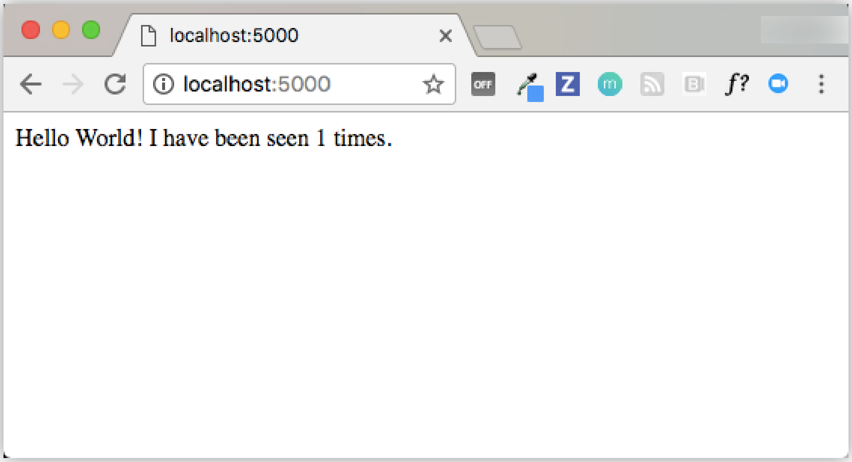
- Compose pulls a Redis image, builds an image for your code, and starts the services you defined. In this case, the code is statically copied into the image at build time.

errors

> If you run into the following error, try to delete `~/.docker/config.json`
> `ERROR: (gcloud.auth.docker-helper) There was a problem refreshing your current auth tokens: ('invalid_grant: Bad Request', u'{\n "`

4.2 - Check the app in browser

- Enter http://localhost:5000/ in a browser to see the application running.



Hello World! I have been seen 1 times.

5. Edit the Compose file to add a bind mount

5.1 - Edit the compose file

Edit `docker-compose.yml` in your project directory to add a bind mount for the web service

```
version: '3'
services:
  web:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - .:/code
    environment:
      FLASK_ENV: development
  redis:
    image: "redis:alpine"
```

- The new volumes key mounts the project directory (current directory) on the host to /code inside the container, allowing you to modify the code on the fly, without having to rebuild the image. The environment key sets the FLASK_ENV environment variable, which tells flask run to run in development mode and reload the code on change. This mode should only be used in development.

6. Re-build and run the app with Compose

6.1 - Stop the app

```
docker-compose down
```

6.2 - Rebuild the app

```
docker-compose up
```

Course Content
Enter code

All
Lecture
Recordings
Practices
Chapter
Program Information
›
Chapter
Surveys
›
Chapter
Week 00 (Virtual)- Program Preparation
›
Chapter
Week 01 - SQL
›
Chapter
Week 02 - Python
›
Chapter
Week 03 - Client Project
›
Chapter
Week 04 - Linux and AWS
›
Chapter
Week 05 - Docker and Client Project phase 2

Chapter overview
Sunday - Docker I
[Lecture Material] Docker
[Lab] Software Installation: Docker
[Lab] Account Creation Create your Dockerhub account
[Lab] Workshop Demonstrating Hello World Example
[Lab] Workshop: Work with Docker Image
[Lab] Exercise: Basic Docker Commands
[Lecture Video] Docker Sunday
Monday - Docker II
[Lab] Workshop: Install Zepplin with Docker
[Lab] Workshop: Docker Compose --Flask
[Quiz] Docker Commands Quiz
Tuesday - Real Client Project Phase 2
Wednesday - Real Client Project Phase 2
Thursday - Real Client Project Phase 2
RCP project Submission (Competition)
[Lab] Workshop: Docker Compose --Flask