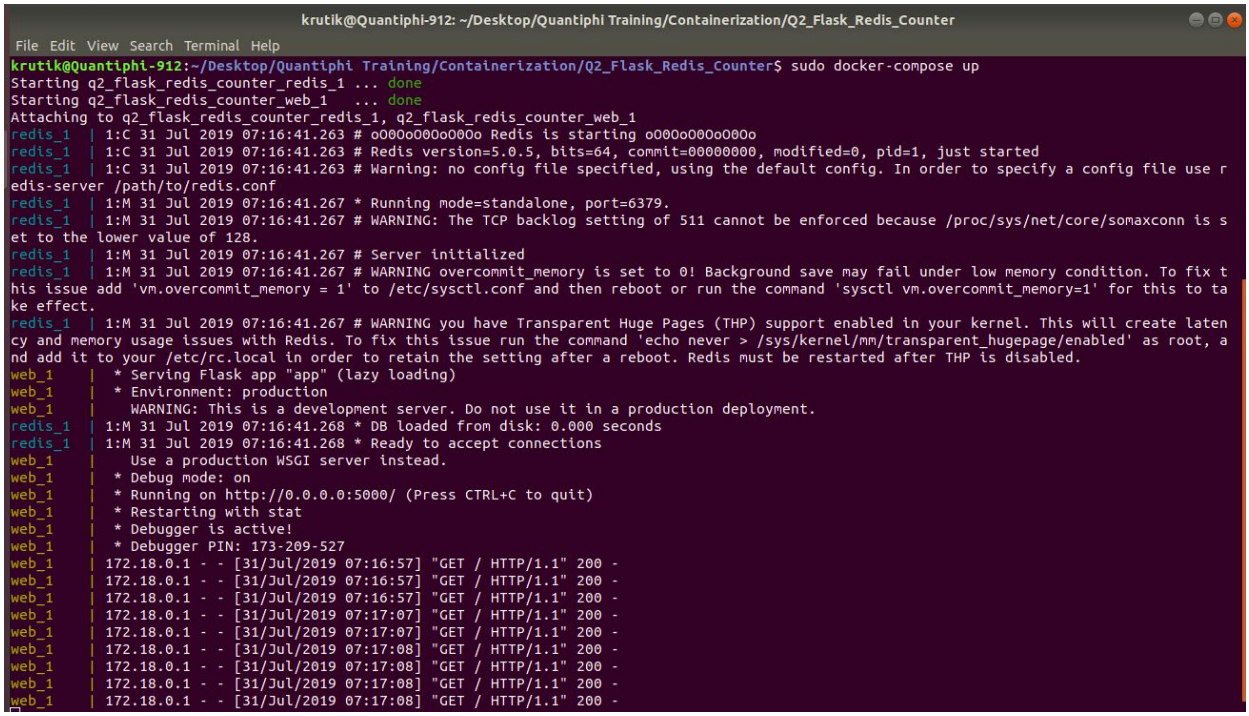
Docker compose

Part - 1

1. Create the following files inside a directory in your local machine
    a. app.py - this is for the Flask App that you want to run
    b. requirements.txt - this contains the requirements, in this case flask and redis
    c. Dockerfile - to setup the flask environment
    d. docker-compose.yml - creating two services namely web for flask application and another redis for redis image to store the counter
2. Run ‘docker-compose build’ to build the containers specified in the docker-compose file
3. Run ‘docker-compose up’ to start the container
4. Navigate to http://localhost:5000 to check the webserver



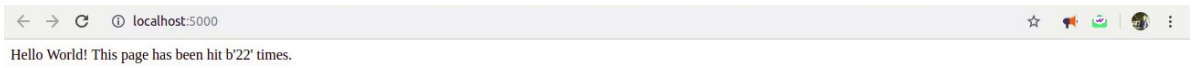
Terminal Running Containers

Web Server with hit counts

As there is no volume attached to the containers, any changes made won’t reflect unless you rebuild the docker image.

Part - 2

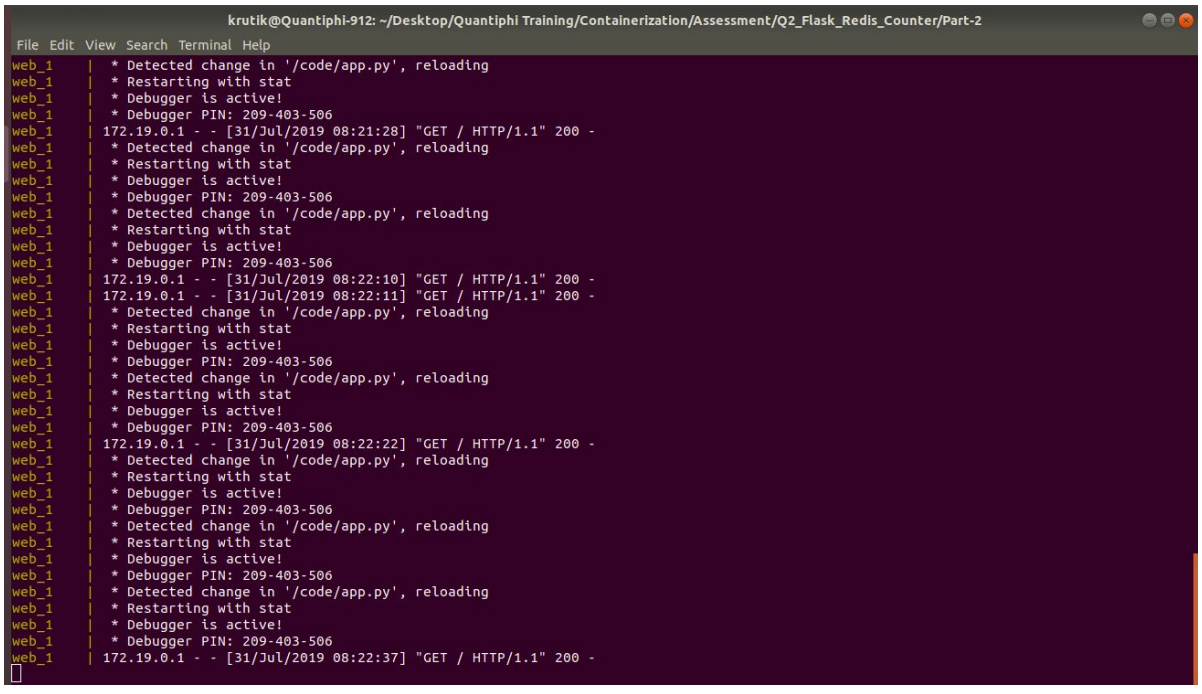
1. To attach a volume to the container, add volumes in the docker-compose.yml file with the mapping of directories
2. Now run ‘docker-compose build’ to rebuild the images with the added volumes
3. To start the containers in daemon (to later edit files in same terminal) run ‘docker-compose up -d’ or run ‘docker-compose up’ to run the containers in normal mode
4. Now edit the file in the container using ‘docker exec -it [container-id] bash’ and then using any in terminal text editor as nano or vim
5. Or you can edit in the files that are locally stored in your machine as the volumes are attached and would reflect the changes even from the outside



Initial build page

Updated page without rebuilding image

Another update



Terminal showing changes detected in working code as volume is attached

This is a host volume type of volume attached to the container to reflect changes from local source to inside of the container.
Another method to do the same is by attaching an external volume. For the same you need to first create a volume and then attach the same.
You can do this by adding this volumes specification in your required service -

```
volumes:
      - [name-of-volume]:/your/local/path
```

And then add this at the end of the docker-compose file to mention the source of the mounted volume

```
volumes:
   [name-of-volume]:
      external: false
```

external true if you provide your volume from an external source, not directly from the docker-compose spec