Dockerize your Python Application

Dockerfile

First, start with a fresh empty directory. As an example, you can call this *my_new_docker_project* – but feel free to use whatever name you like. This directory defines the context of your build, meaning it contains `all of the things you need` to do.

Create a new text file in *my_new_docker_project* called **Dockerfile** (note no extension; on Windows, you may need to save the file as "All types" and put the filename in quotes to avoid automatically appending an extension); use whatever text file editor you already know (you might use Sublime, Notepad++, emacs, nano, or even vim).

In our example, we use the basic Python 3 image as our launching point. Add the following line to your Dockerfile:

```
FROM python:3
```

We want to run a basic Python script which we'll call *app*. First, we need to `add` the script to the Dockerfile:

```
ADD . /app
WORKDIR /app
```

Our script depends on the some requirements (in our example flask and redis), so we need to make sure we install that before we run app.py. Add this line to your Dockerfile to install:

```
RUN pip3 install -r requirements.txt
```

Add this line to your Dockerfile to execute the script:

```
CMD python3 app.py
```

Your Dockerfile should look like this:

```
FROM python:3
ADD . /app
WORKDIR /app
RUN pip3 install -r requirements.txt
CMD python3 app.py
```

requirements.txt

After creating your Dockerfile, You should create `requirements.txt` file at the same path. As we mentioned above there are two required python libraries (flask and redis). So inside of the requirements.txt should be like that:

```
flask
redis
```

Docker-compose.yml

Create a docker compose.yml at your project directory. In our example, we create it at the *my_new_docker_project*. We will use version 3 and two services. We still use port *5000* at *localhost*. We all declared inside of the docker-compose.yml file.

```
version: '3'
services:
    app:
        build: .
        ports:
            - "5000:5000"
        volumes:
            - .:/app
        depends_on:
            - redis
    redis:
        image: redis
```

app.py File

Do not have to understand every line of code but this code renders a web page. When you change URL path. It gives you to Bonus and web page changes according to your path's name.

```
# compose_flask/app.py

# import Flask and Redis libraries
from flask import Flask
from redis import Redis
import random

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

# declare main route
@app.route('/')
def main():
    return 'Hi. In order to earn bonus points enter your name in the url. eg:
        /John'

# declare a route which gets the visitor's name. Every name has its own bonus
    points.
@app.route('/')
def greet(name):
    bonus = random.randrange(1, 100)
    redis.incrby(name, bonus)
    return 'Hello %s. You have earned %d bonus points. Your total point is %s.' %
        (name, bonus, redis.get(name))

# run the flask application on the local development server.
if __name__ == "__main__":
    app.run(host="0.0.0.0", debug=True)
```

docker-compose up

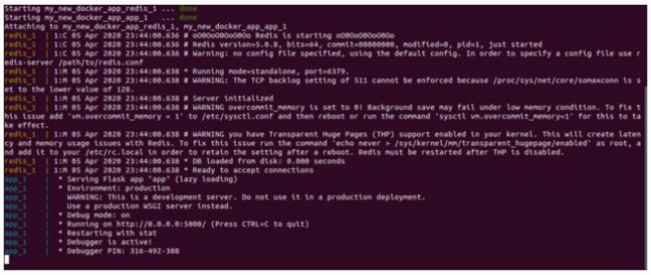
💡 *Now time to set on fire our first docker application.*

As you understand from the title you just need to write `docker-compose up` at the terminal. The project directory should contain:

- docker-compose.yml
- Dockerfile
- requirements.txt
- app.py (Your python codes name)

Be sure to this command is running at the project directory. Otherwise, it doesn't work.

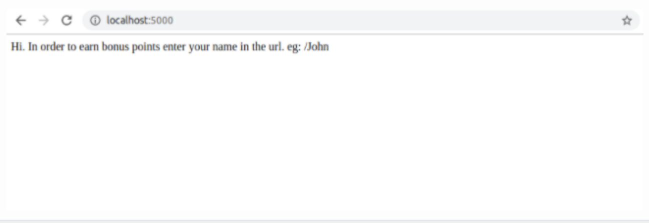
```
clarus-linux@professor:~/my_new_docker_app$ docker-compose up
```



```
Starting my_new_docker_app_redis_1 ... done
Starting my_new_docker_app_app_1   ... done
Attaching to my_new_docker_app_redis_1, my_new_docker_app_app_1
redis_1  | 1:C 05 Apr 2020 23:44:00.636 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
redis_1  | 1:C 05 Apr 2020 23:44:00.636 # Redis version=5.0.8, bits=64, commit=00000000, modified=0, pid=1, just started
redis_1  | 1:C 05 Apr 2020 23:44:00.636 # Warning: no config file specified, using the default config. In order to specify a config file use r
edis-server /path/to/redis.conf
redis_1  | 1:M 05 Apr 2020 23:44:00.638 * Running mode=standalone, port=6379.
redis_1  | 1:M 05 Apr 2020 23:44:00.638 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/somaxconn is s
et to the lower value of 128.
redis_1  | 1:M 05 Apr 2020 23:44:00.638 # Server initialized
redis_1  | 1:M 05 Apr 2020 23:44:00.638 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix t
his issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to ta
ke effect.
redis_1  | 1:M 05 Apr 2020 23:44:00.638 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create laten
cy and memory usage issues with Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root, a
nd add it to your /etc/rc.local in order to retain the setting after a reboot. Redis must be restarted after THP is disabled.
redis_1  | 1:M 05 Apr 2020 23:44:00.638 * DB loaded from disk: 0.000 seconds
redis_1  | 1:M 05 Apr 2020 23:44:00.638 * Ready to accept connections
app_1    |  * Serving Flask app "app" (lazy loading)
app_1    |  * Environment: production
app_1    |    WARNING: This is a development server. Do not use it in a production deployment.
app_1    |    Use a production WSGI server instead.
app_1    |  * Debug mode: on
app_1    |  * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
app_1    |  * Restarting with stat
app_1    |  * Debugger is active!
app_1    |  * Debugger PIN: 316-492-308
```

Your output should seem like this output

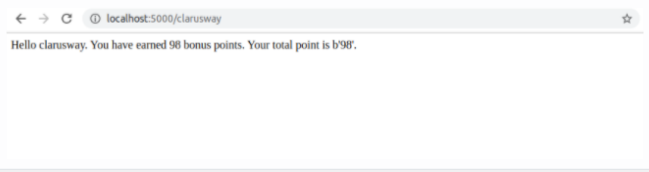
Then open your web browser and type `localhost/5000`.



Hi. In order to earn bonus points enter your name in the url. eg: /John

First web page will render like this

Please write your name and see the results. For example when i type `localhost:5000/clarusway`:



Hello clarusway. You have earned 98 bonus points. Your total point is b'98'.

After you type your name this web page should change like above