

Step 1

Install Compose

 $\begin{array}{l} \textbf{sudo} \ \textbf{curl} \ \textbf{-L} \ \textbf{"https://github.com/docker/compose/releases/download/1.25.4/docker-compose-\$(uname \ \textbf{-s})-\$(uname \ \textbf{-m})" \ \textbf{-o} \ \textbf{/usr/local/bin/docker-compose} \end{array}$

sudo chmod +x /usr/local/bin/docker-compose

sudo yum install epel-release sudo yum install -y python-pip sudo pip install docker-compose sudo yum upgrade python*

Verify Docker Compose Installation

\$ docker-compose --version

Luiss-MacBook-Air:Dockerfiles luiscastro\$ docker-compose --version docker-compose version 1.21.1, build 5a3f1a3

Step 2

Deploy an app with Docker Compose

\$ git clone https://github.com/nigelpoulton/counter-app.git

\$ cd counter-app

\$ ls

app.py

docker-compose.yml

Dockerfile

requirements.txt



Step 3

Analize docker-compose.yml

\$ cat docker-compose.yml

```
[ec2-user@ip-172-31-50-64 counter-app]$ cat docker-compose.yml
version: "3.5"
services:
 web-fe:
   build: .
   command: python app.py
   ports:
    - target: 5000
       published: 5000
   networks:
     - counter-net
   volumes:
     - type: volume
       source: counter-vol
       target: /code
                        RS-DOCKER-LAB #5 - Docker Com
   image: "redis:alpine"
   networks:
     counter-net:
networks:
 counter-net:
volumes:
[ec2-user@ip-172-31-50-64 counter-app]$
```

Step 4

Start the app

\$ docker-compose up &

```
[1] 1635
Creating network "counterapp_counter-net" with the default driver
Creating volume "counterapp_counter-vol" with default driver
Pulling redis (redis:alpine)...
alpine: Pulling from library/redis
1160f4abea84: Pull complete
a8c53d69ca3a: Pull complete
<Snip>
web-fe 1 | * Debugger PIN: 313-791-729
```

CONTAINERS-DOCKER-LAB #5 - Docker Compose

\$ docker image Is

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
counterapp_web-fe	latest	966ff9e	3 minutes ago	95 .9 MB
python	3.4-alpine	0117a02	2 weeks ago	85.5MB
redis	alpine	edc83de	5 weeks ago	26.9MB

Step 5

Analyze Dockerfile

\$ vim Dockerfile

FROM python: 3.4-alpine < < Base image

ADD . /code < < Copy app into image

WORKDIR /code < < Set working directory

RUN pip install -r requirements.txt < < install requirements

CMD [" python", "app.py"] < < Set the default app

Step 6

Verify Docker components

\$ docker container Is

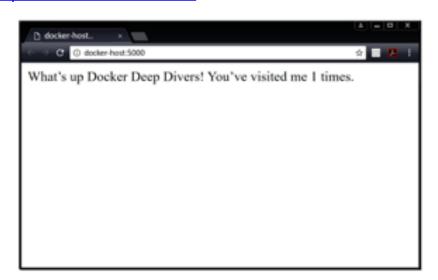
\$ docker network Is

\$ docker volume Is

Step 7

Validate app is running

curl http://Private-IP-Address:5000





Step 8

Stop the app

\$ docker-compose down