Docker compose

what is Docker compose

compose is a tool for defining and running multi-container Docker applications.with compose use a yaml file to configure your application services.with a singe command you create and start all the services from configuration Run "docker-compose up" and compose starts "docker-compose down" stop.

installing docker compose?

run the command current stable release of docker compose

curl -L "https://github.com/docker/compose/releases/download/1.10.0-rc2/dockercompose-$(uname -s) -$(uname -m)" -o /home/docker-compose



chmod +x /home/docker-compose



$ docker-compose --version



Compose use a YAML or yml scripts it looks like this

version: "3.7"

services:

webapp:

build: ./dir

docker-compose config

it will check validity of the command

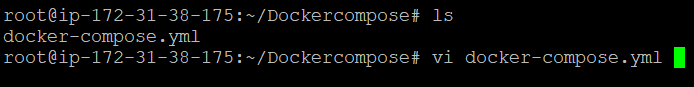
docker-compose up -d

it will run in deatch mode

docker-compose down

it will down the application

\*For creating a simple app using compose file



docker-compose.nginx.yml

version: "2"

services:

webapp:

image: nginx

ports:

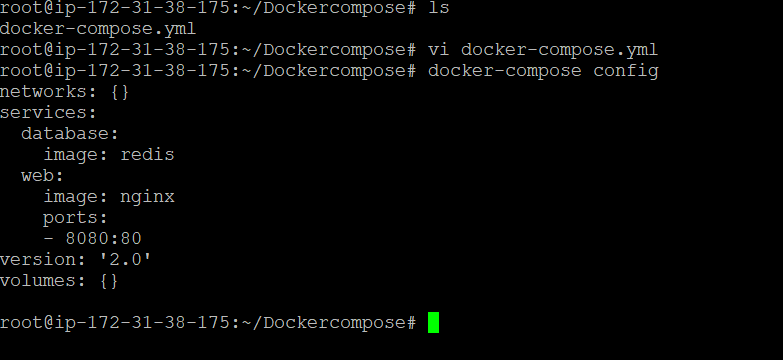
- 8080:80

database:

image: redis

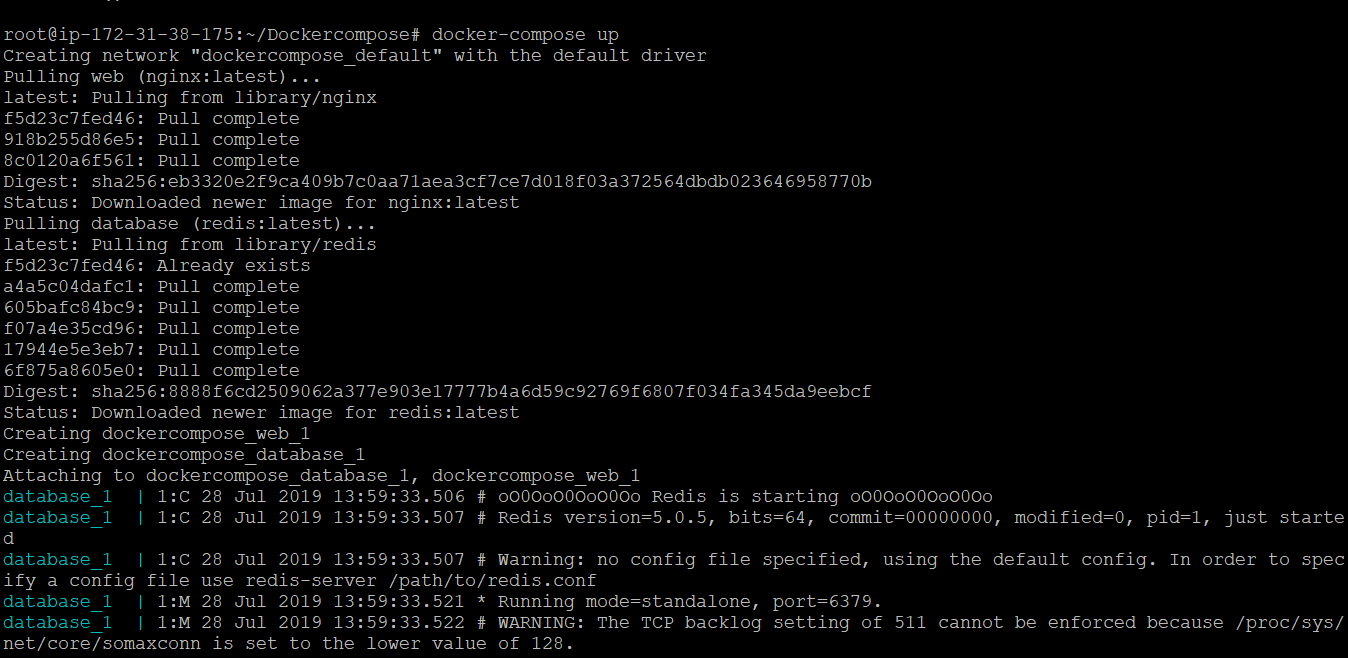


docker-compose config



it will check the syntax

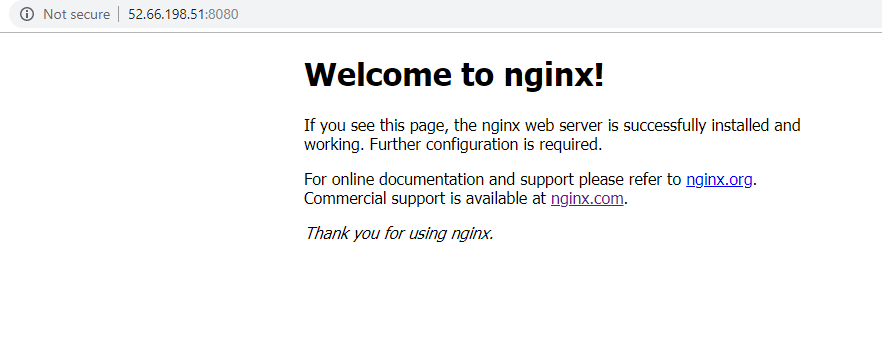
docker-compose up



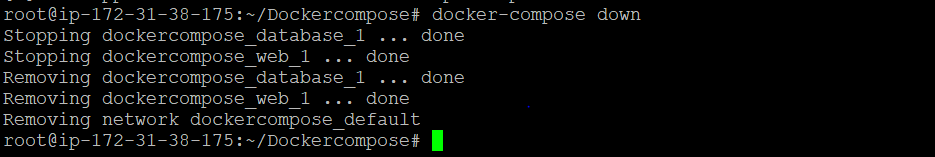
it will pull the images from docker hub

and start the applications

see



docker-compose down



it will stop the applications

run the command

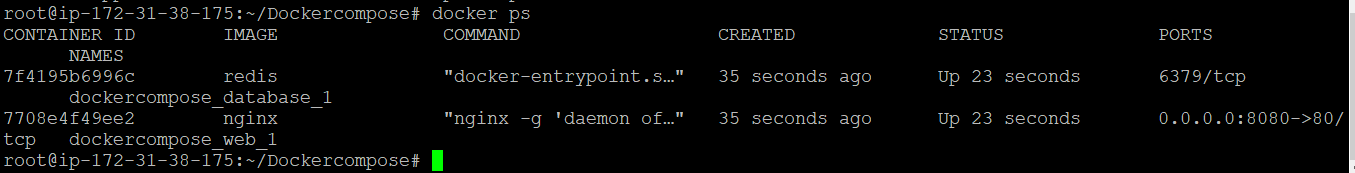
docker-compose up

if you want scale the services

docker-compose up -d --scale web=4

to check the services

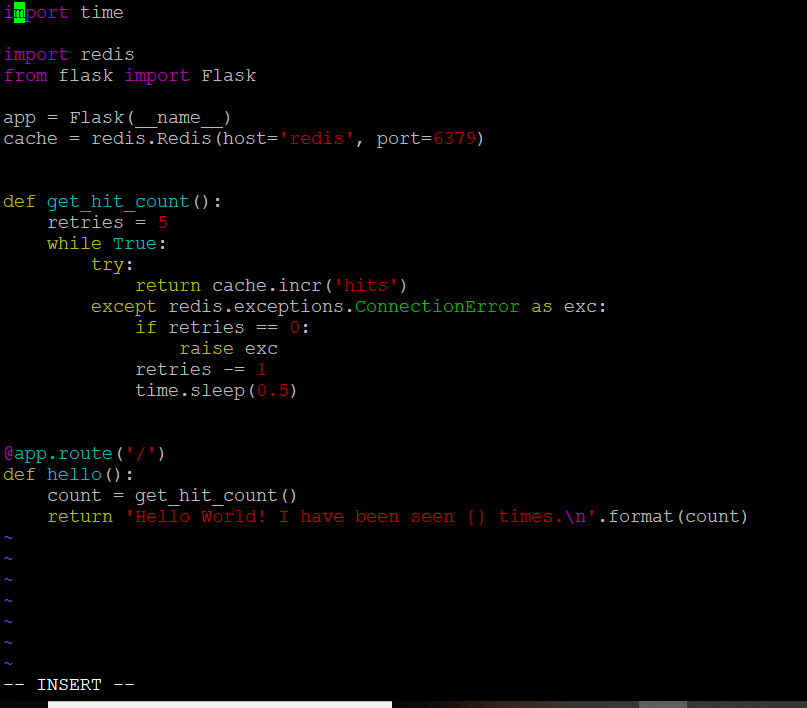
docker ps



to a sample app

i am creating a flask project "app.py"

7.PNG



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():

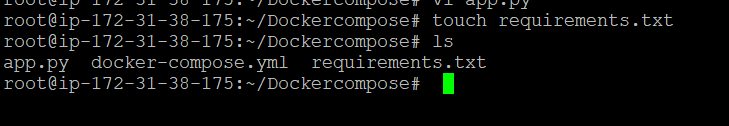
count = get\_hit\_count()

return 'Hello World! I have been seen {} time\n'.format(count)

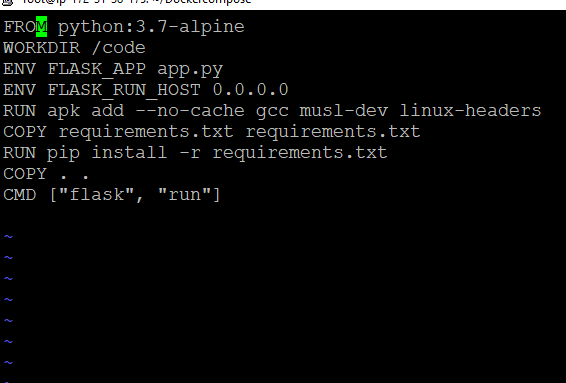
create requirements.txt file

flask

redis



create a Docker file



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 gccmusl-dev linux-headers

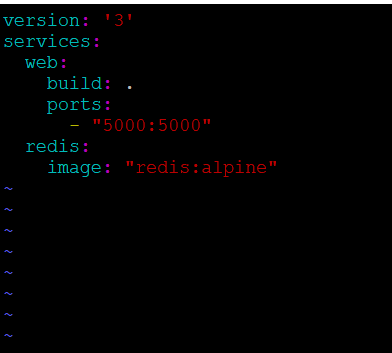
COPY requirements.txt requirements.txt

RUN pip install -r requirements.txt

COPY . .

CMD ["flask", "run"]

create a docker-compose.yml



version: '3'

services:

web:

build: .

ports:

- "5000:5000"

redis:

image: "redis:alpine"

to build up a application

run

docker-compose up

