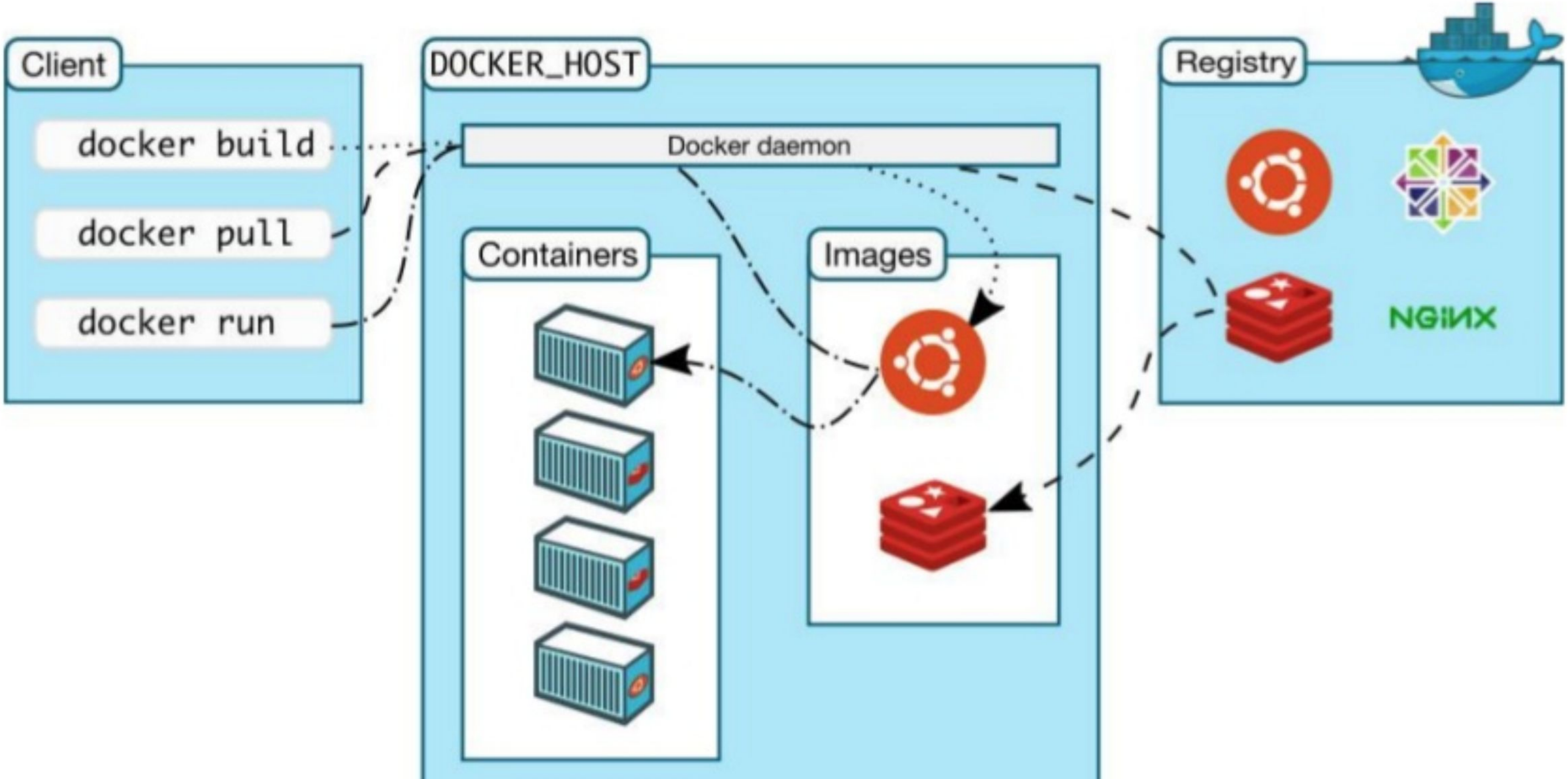
Introduction to Docker Compose

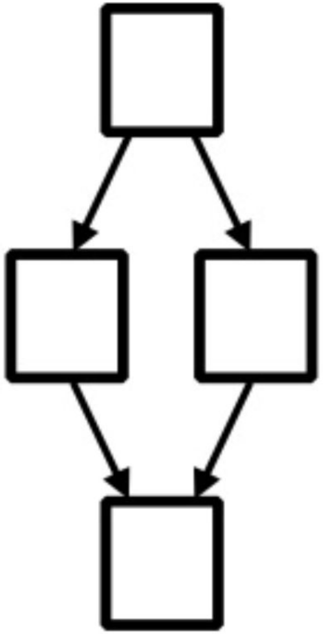
Remember Docker



Multi-container apps are a hassle.

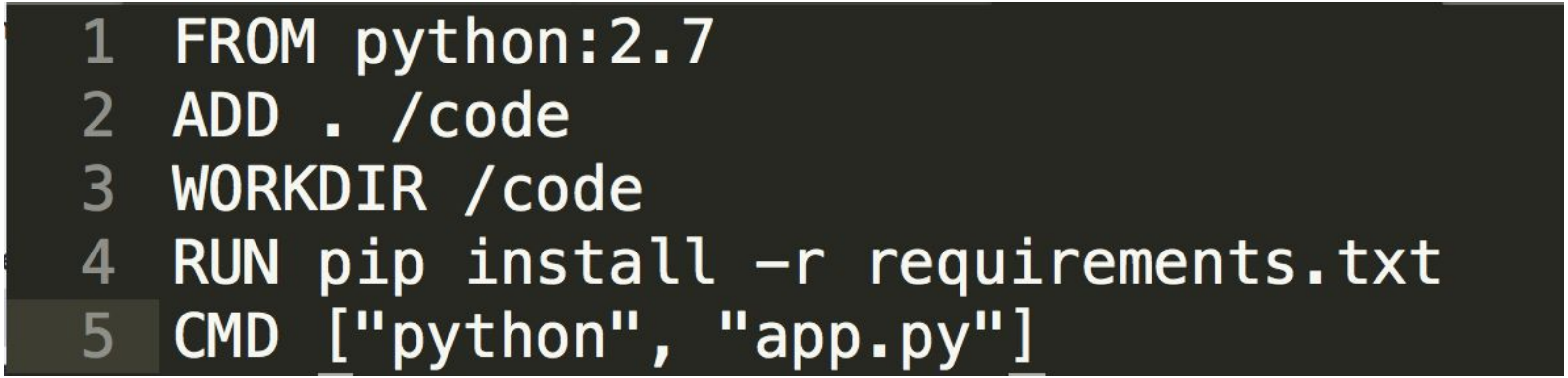
- Build images from Dockerfiles
- Pull images from the Hub or a private registry
- Configure and create containers
- Start and stop containers
- Stream their logs

Multi-container apps are a hassle.



```
$ docker pull ...
$ docker pull ...
$ docker build ...
$ docker build ...

$ docker run ...
$ docker run ...
$ docker run ...
$ docker run ...
```

Dockerfile



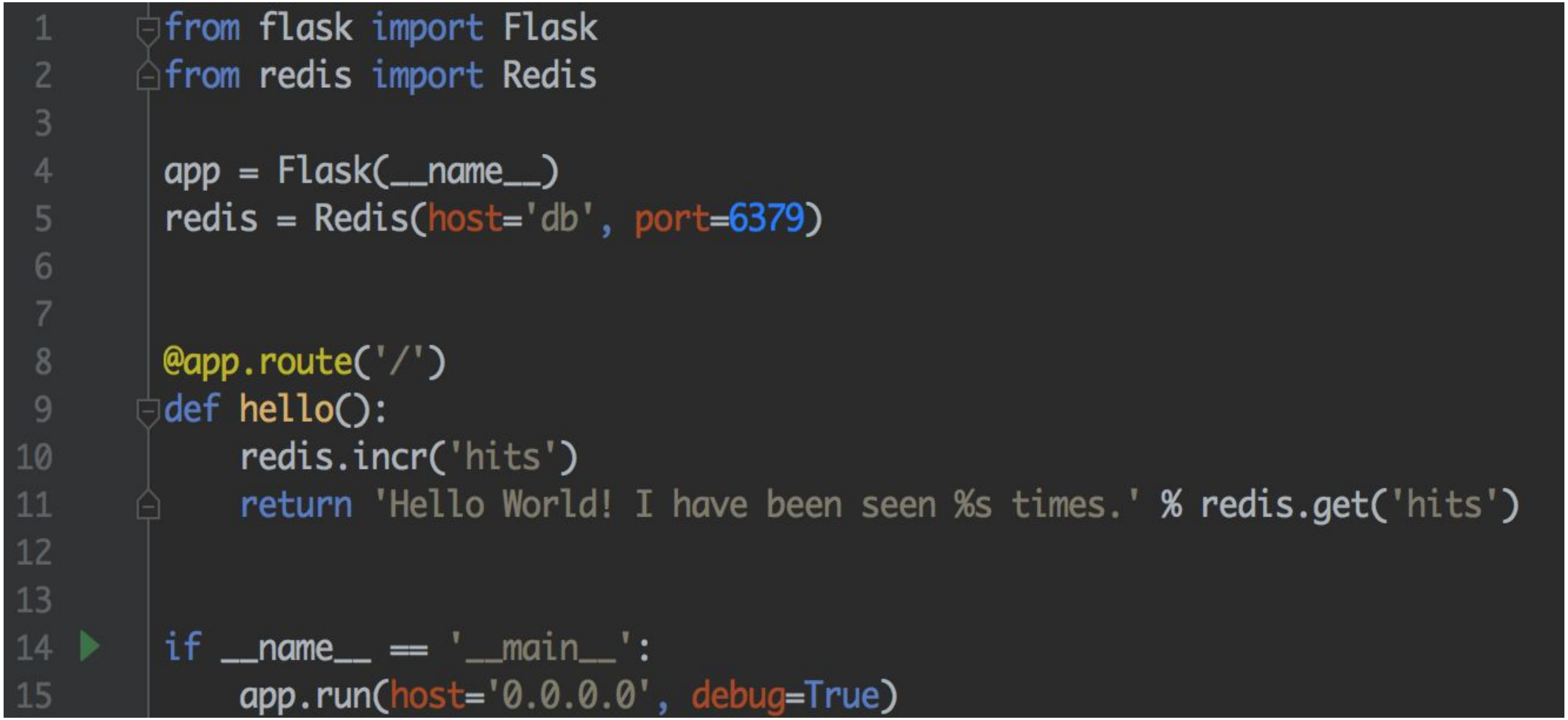
```
FROM python:2.7
ADD . /code
WORKDIR /code
RUN pip install -r requirements.txt
CMD ["python", "app.py"]
```

full source code

$ open https://goo.gl/4wL7FL

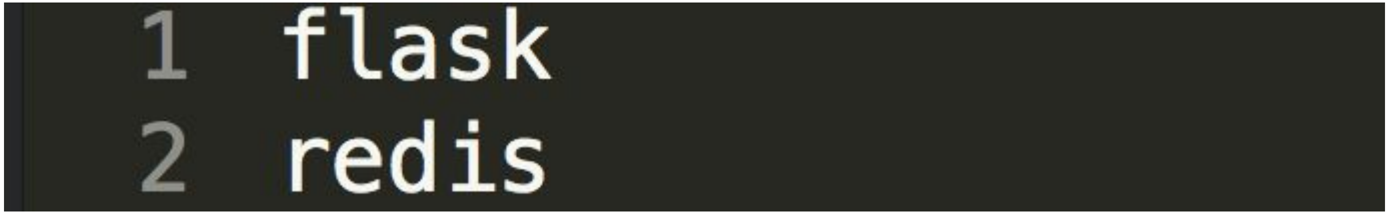
$ git clone https://github.com/soportefacil/presentation_dockercompose101.git

app.py



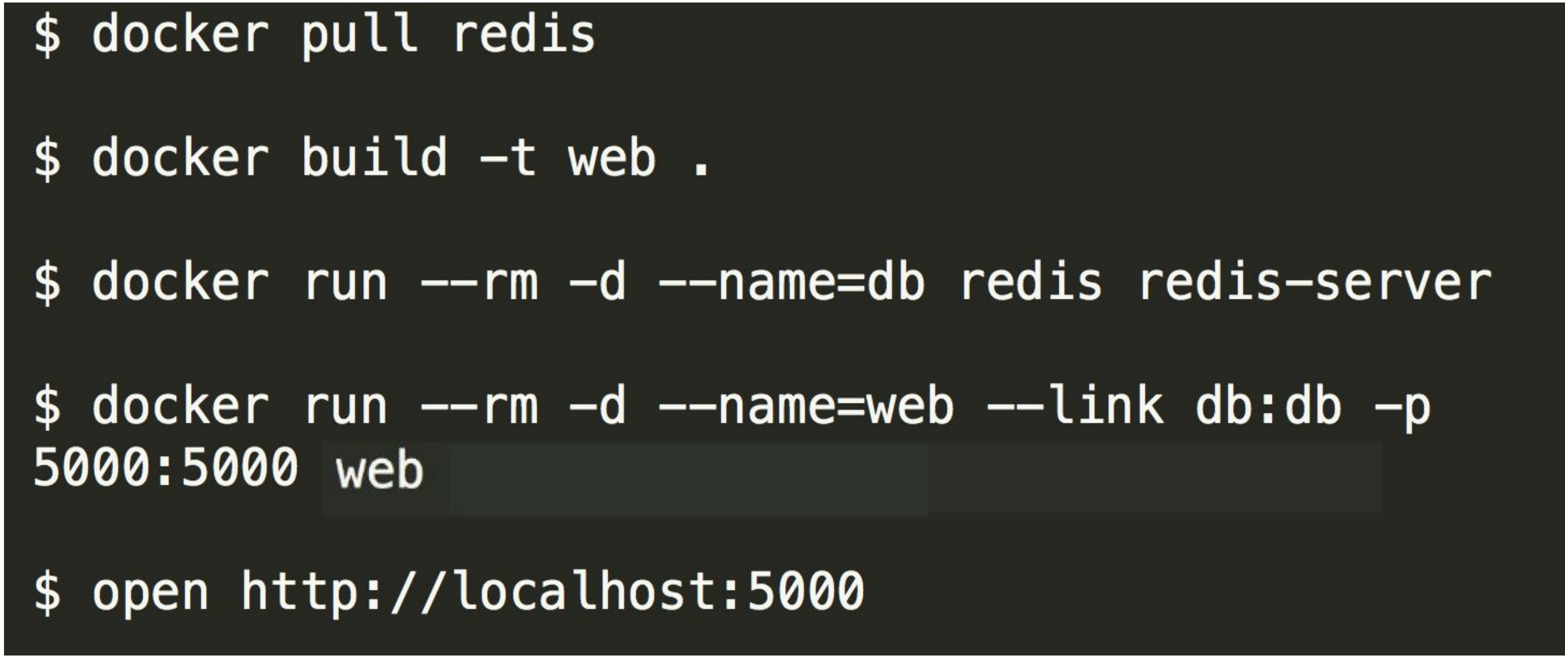
```python
from flask import Flask
from redis import Redis

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


@app.route('/')
def hello():
    redis.incr('hits')
    return 'Hello World! I have been seen %s times.' % redis.get('hits')


if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=True)
```

requirements.txt



```
flask
redis
```

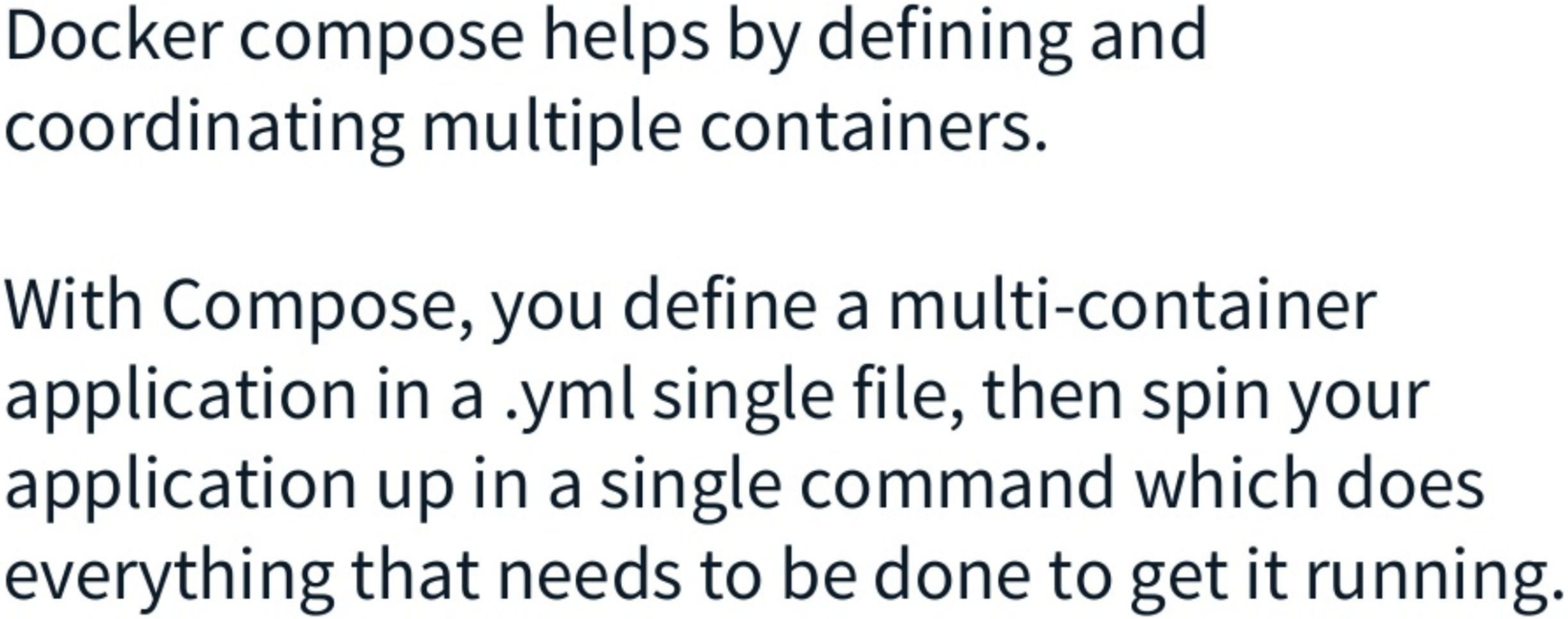
Multi-container apps are a hassle.



```
$ docker pull redis

$ docker build -t web .

$ docker run --rm -d --name=db redis redis-server

$ docker run --rm -d --name=web --link db:db -p
5000:5000 web

$ open http://localhost:5000
```

Docker compose

#Define and run multi-container applications with Docker.



Docker compose helps by defining and coordinating multiple containers.

With Compose, you define a multi-container application in a .yml single file, then spin your application up in a single command which does everything that needs to be done to get it running.

Native Docker API

The same API used by **other Docker commands**.

The same API used by **other Docker tools**.

The same API **exposed by Docker Swarm**.

Multi-container apps on multi-host clusters.

Install Docker Compose

You can run Compose on macOS, Windows, and 64-bit Linux.

Prerequisites

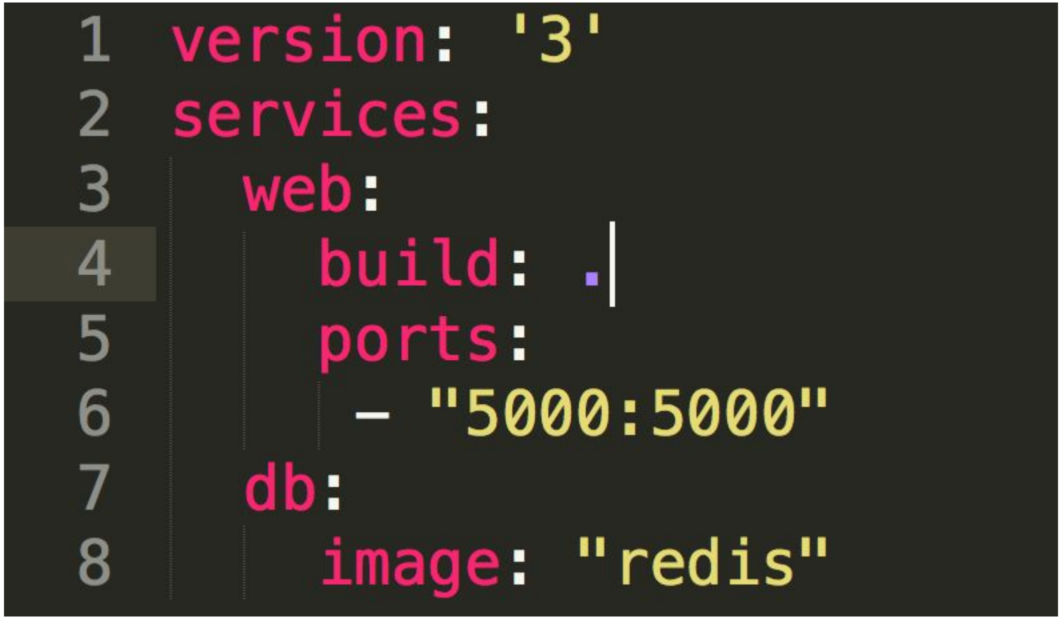
Docker Compose relies on Docker Engine for any meaningful work, so make sure you have Docker Engine installed either locally or remote, depending on your setup.
https://docs.docker.com/compose/install/

New commands

- `docker-compose up` #Builds, (re)creates, starts, and attaches to containers for a service.
- `docker-compose ps` #Lists containers.
- `docker-compose logs -f` #Displays log output from services.
- `docker-compose stop, top, rm, kill,…`

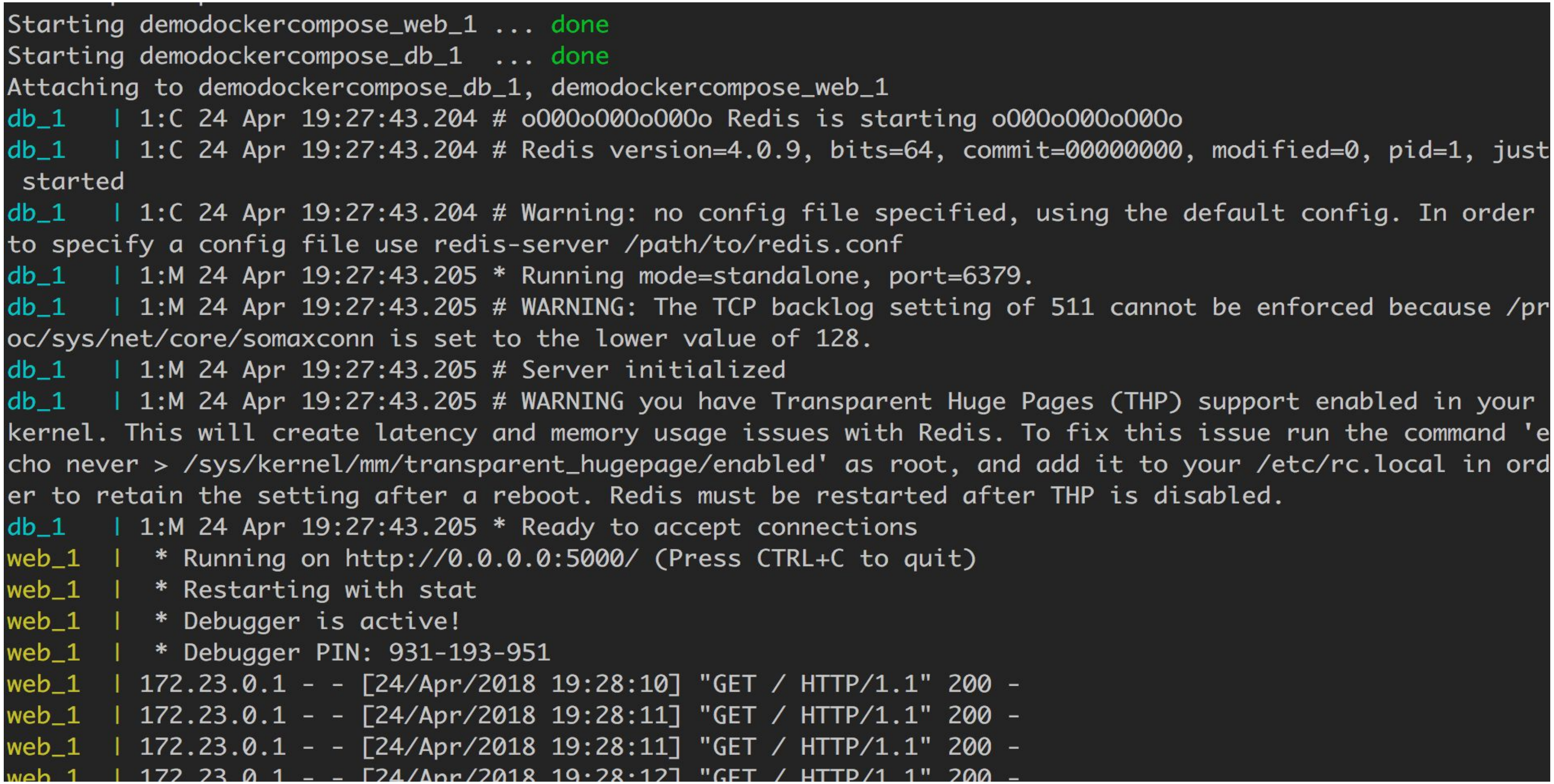
More info: https://docs.docker.com/compose/reference/

docker-compose.yml

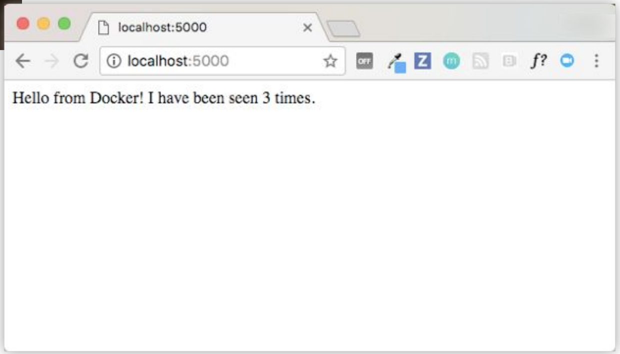


```yaml
version: '3'
services:
  web:
    build: .
    ports:
     - "5000:5000"
  db:
    image: "redis"
```

$ docker-compose up



```
Starting demodockercompose_web_1 ... done
Starting demodockercompose_db_1  ... done
Attaching to demodockercompose_db_1, demodockercompose_web_1
db_1    | 1:C 24 Apr 19:27:43.204 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
db_1    | 1:C 24 Apr 19:27:43.204 # Redis version=4.0.9, bits=64, commit=00000000, modified=0, pid=1, just
 started
db_1    | 1:C 24 Apr 19:27:43.204 # Warning: no config file specified, using the default config. In order
to specify a config file use redis-server /path/to/redis.conf
db_1    | 1:M 24 Apr 19:27:43.205 * Running mode=standalone, port=6379.
db_1    | 1:M 24 Apr 19:27:43.205 # WARNING: The TCP backlog setting of 511 cannot be enforced because /pr
oc/sys/net/core/somaxconn is set to the lower value of 128.
db_1    | 1:M 24 Apr 19:27:43.205 # Server initialized
db_1    | 1:M 24 Apr 19:27:43.205 # WARNING you have Transparent Huge Pages (THP) support enabled in your
kernel. This will create latency and memory usage issues with Redis. To fix this issue run the command 'e
cho never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.local in ord
er to retain the setting after a reboot. Redis must be restarted after THP is disabled.
db_1    | 1:M 24 Apr 19:27:43.205 * Ready to accept connections
web_1   |  * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
web_1   |  * Restarting with stat
web_1   |  * Debugger is active!
web_1   |  * Debugger PIN: 931-193-951
web_1   | 172.23.0.1 - - [24/Apr/2018 19:28:10] "GET / HTTP/1.1" 200 -
web_1   | 172.23.0.1 - - [24/Apr/2018 19:28:11] "GET / HTTP/1.1" 200 -
web_1   | 172.23.0.1 - - [24/Apr/2018 19:28:11] "GET / HTTP/1.1" 200 -
web_1   | 172.23.0.1 - - [24/Apr/2018 19:28:12] "GET / HTTP/1.1" 200 -
```



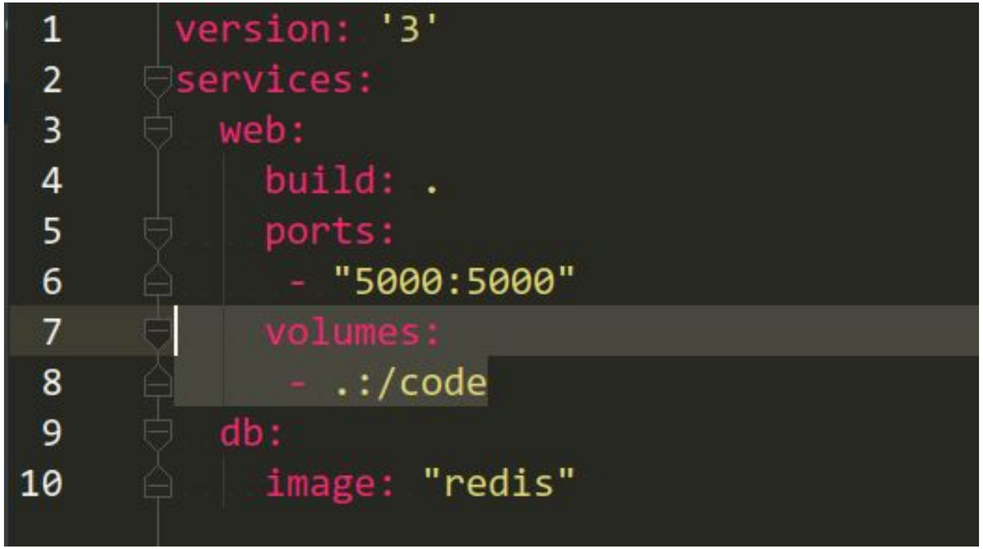
Hello from Docker! I have been seen 3 times.

GAME TIME

- `docker-compose up -d` #detached
- `docker-compose run web env` #run one-off commands
- `docker-compose stop` #docker stop
- `docker-compose down` #docker stop & docker rm

- **`docker-compose --help`**
- https://docs.docker.com/compose/django/

one thing more



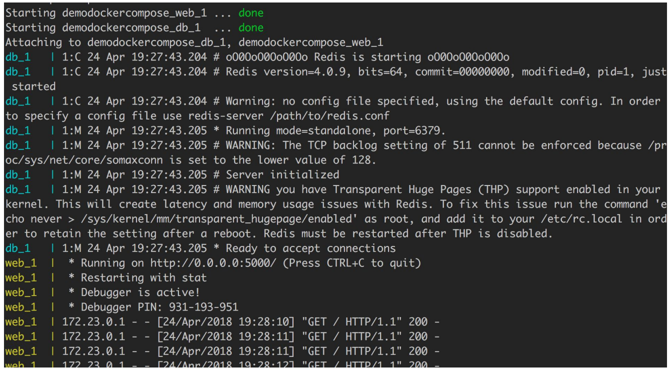
```yaml
version: '3'
services:
  web:
    build: .
    ports:
     - "5000:5000"
    volumes:
     - .:/code
  db:
    image: "redis"
```

a) **Edit the Compose file to add a bind mount** : 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.

one thing more

b) Re-build and run the app with Compose.

```
$ docker-compose up
```



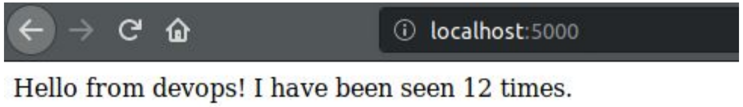
```
Starting demodockercompose_web_1 ... done
Starting demodockercompose_db_1  ... done
Attaching to demodockercompose_db_1, demodockercompose_web_1
db_1    | 1:C 24 Apr 19:27:43.204 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
db_1    | 1:C 24 Apr 19:27:43.204 # Redis version=4.0.9, bits=64, commit=00000000, modified=0, pid=1, just
 started
db_1    | 1:C 24 Apr 19:27:43.204 # Warning: no config file specified, using the default config. In order
to specify a config file use redis-server /path/to/redis.conf
db_1    | 1:M 24 Apr 19:27:43.205 * Running mode=standalone, port=6379.
db_1    | 1:M 24 Apr 19:27:43.205 # WARNING: The TCP backlog setting of 511 cannot be enforced because /pr
oc/sys/net/core/somaxconn is set to the lower value of 128.
db_1    | 1:M 24 Apr 19:27:43.205 # Server initialized
db_1    | 1:M 24 Apr 19:27:43.205 # WARNING you have Transparent Huge Pages (THP) support enabled in your
kernel. This will create latency and memory usage issues with Redis. To fix this issue run the command 'e
cho never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.local in ord
er to retain the setting after a reboot. Redis must be restarted after THP is disabled.
db_1    | 1:M 24 Apr 19:27:43.205 * Ready to accept connections
web_1   |  * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
web_1   |  * Restarting with stat
web_1   |  * Debugger is active!
web_1   |  * Debugger PIN: 931-193-951
web_1   | 172.23.0.1 - - [24/Apr/2018 19:28:10] "GET / HTTP/1.1" 200 -
web_1   | 172.23.0.1 - - [24/Apr/2018 19:28:11] "GET / HTTP/1.1" 200 -
web_1   | 172.23.0.1 - - [24/Apr/2018 19:28:11] "GET / HTTP/1.1" 200 -
web_1   | 172.23.0.1 - - [24/Apr/2018 19:28:12] "GET / HTTP/1.1" 200 -
```

c) Update the application code “`Hello from devops`” …..

d) You get the change directly without rebuild.

#Check the `Hello World` message in a web browser again,

and refresh to see the count increment.



Hello from devops! I have been seen 12 times.

QUESTIONS

THANK YOU

Sources:

https://docs.docker.com/compose/install/

https://docs.docker.com/compose/gettingstarted/

https://es.slideshare.net/then4way/introduction-to-docker-and-docker-compose