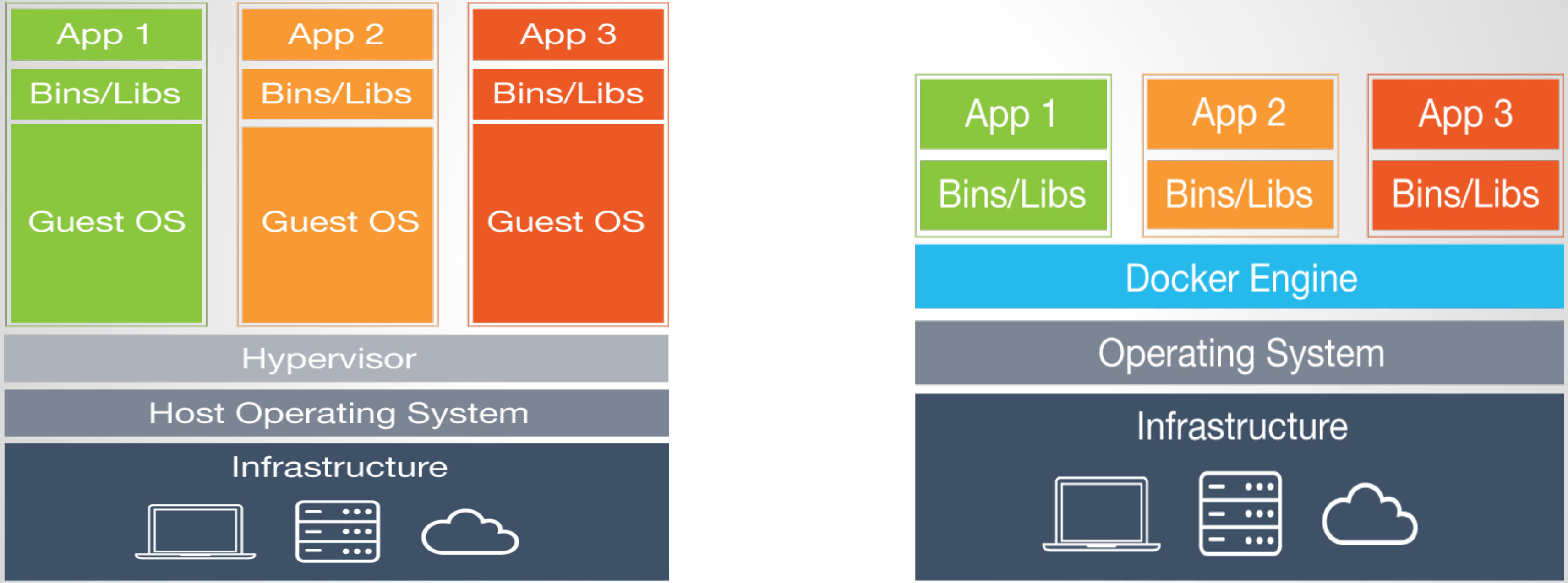
Docker

Means of packaging applications within lightweight containers

Gautham Goli
GitHub

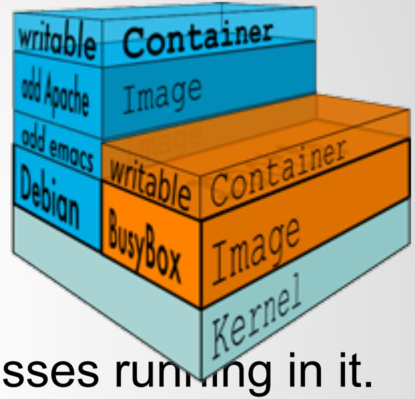
VM and Containers



Containers **share** the OS with other containers unlike Virtual Machines.

Images and Containers

Virtual Machines : Images
Docker containers : Docker Images



Docker Images are read-only templates.
A Container is loaded with a Docker Image and launched.
Technically, a container consists of isolated group of processes running in it.

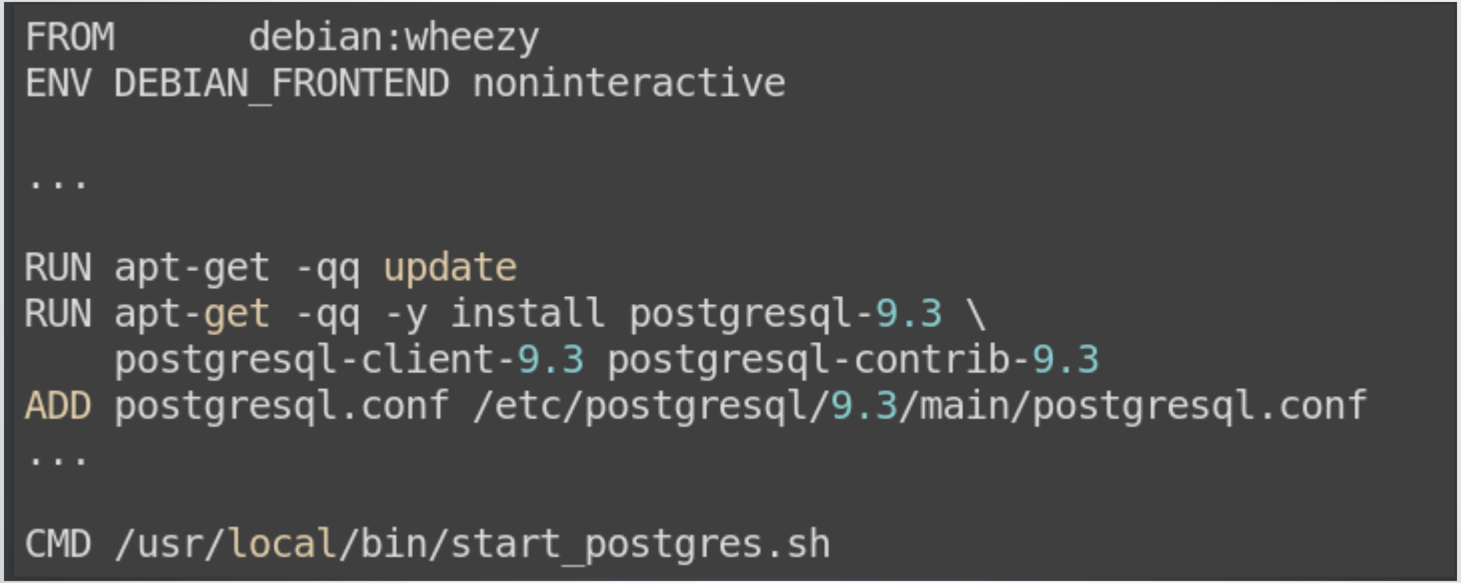
Docker Images are built from base images , as specified in a Dockerfile
Dockerfile is a text file containing sequence of instructions to build an image.

Workflow

1. Write a Dockerfile
2. Build Docker Image from the Dockerfile
3. Run the Image in a container.
4. Share the Image with others by **push**ing the Image into DockerHub
5. Others will **pull** the Image and 3.

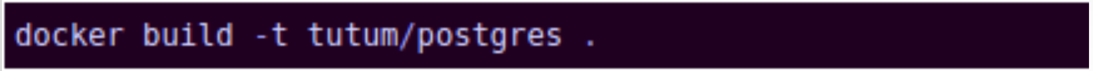
Dockerfile

Syntax:
INSTRUCTION arguments



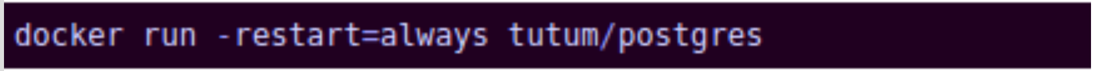
```
FROM      debian:wheezy
ENV DEBIAN_FRONTEND noninteractive

...

RUN apt-get -qq update
RUN apt-get -qq -y install postgresql-9.3 \
    postgresql-client-9.3 postgresql-contrib-9.3
ADD postgresql.conf /etc/postgresql/9.3/main/postgresql.conf
...

CMD /usr/local/bin/start_postgres.sh
```

Build an image named tutum/postgres



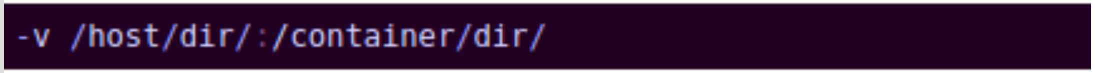
```
docker build -t tutum/postgres .
```

Running the docker image in a container



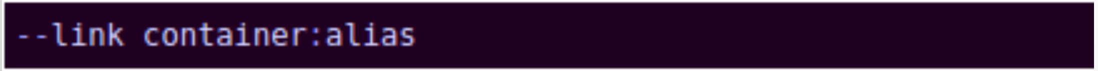
```
docker run -restart=always tutum/postgres
```

Mounting data from host to container

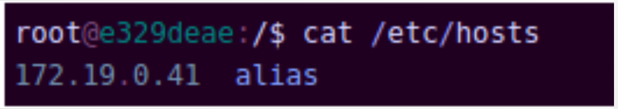


```
-v /host/dir/:/container/dir/
```

Linking containers



```
--link container:alias
```

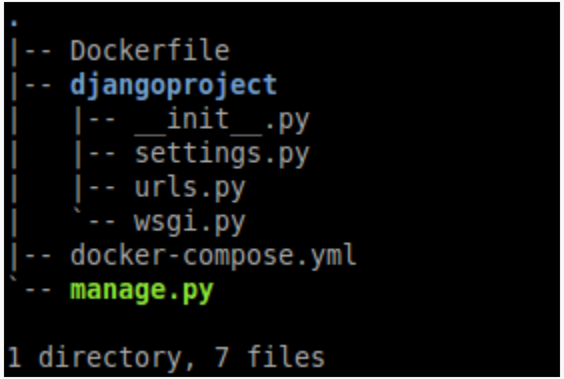


```
root@e329deae:/$ cat /etc/hosts
172.19.0.41   alias
```

Docker Compose

`docker-compose.yml`

```
web:
  build: .
  ports:
   - "5000:5000"
  volumes:
   - .:/code
  links:
   - redis
redis:
  image: redis
```

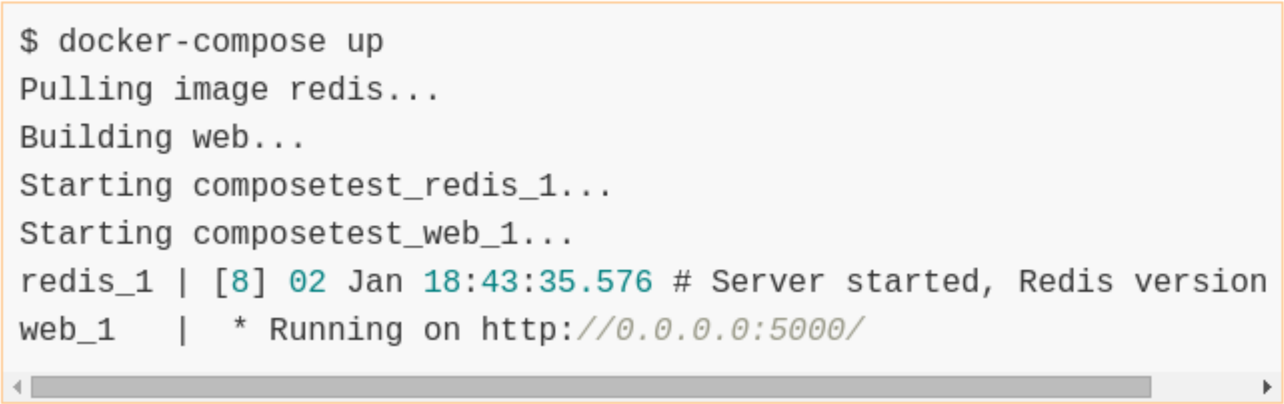


```
.
|-- Dockerfile
|-- djangoproject
|   |-- __init__.py
|   |-- settings.py
|   |-- urls.py
|   `-- wsgi.py
|-- docker-compose.yml
`-- manage.py

1 directory, 7 files
```

Run multiple containers , define link them, mount volumes, forward ports specify commands and more.

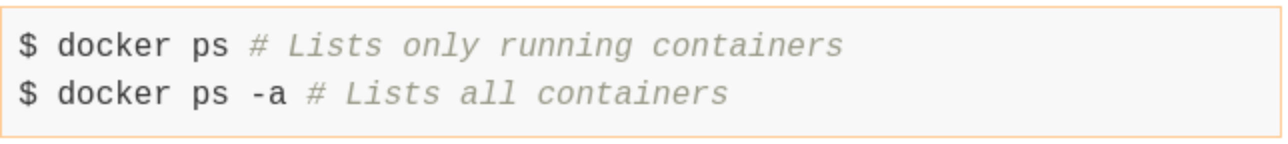
Reference: https://docs.docker.com/compose/yml/

Running the app with compose

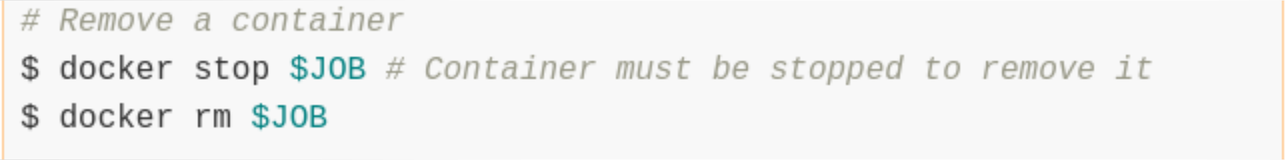


```
$ docker-compose up
Pulling image redis...
Building web...
Starting composetest_redis_1...
Starting composetest_web_1...
redis_1 | [8] 02 Jan 18:43:35.576 # Server started, Redis version
web_1   |  * Running on http://0.0.0.0:5000/
```

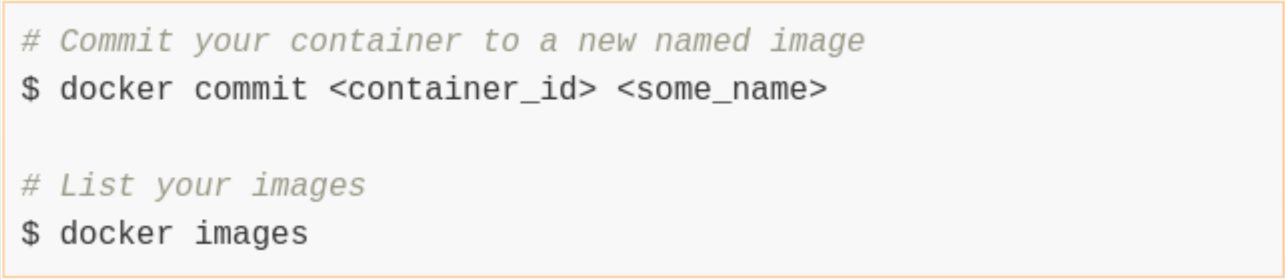
Managing containers



```
$ docker ps # Lists only running containers
$ docker ps -a # Lists all containers
```



```
# Remove a container
$ docker stop $JOB # Container must be stopped to remove it
$ docker rm $JOB
```



```
# Commit your container to a new named image
$ docker commit <container_id> <some_name>

# List your images
$ docker images
```

Useful references

- Docker Basics: https://docs.docker.com/articles/basics/
- Dockerfile reference: https://docs.docker.com/reference/builder/

- Packaging django applications in Docker: http://michal.karzynski.pl/blog/2015/04/19/packaging-django-applications-as-docker-container-images/