docker

Presented by :

Rishab Ilwadi & Shivansh

About us:

We are from Internity Foundation, a nonprofit organization

Agenda

- What is Docker?
- Why Docker ?
- Docker Architecture
- Basic Docker Commands
- Behind the Scenes of Docker
- Demo

What is Docker ?

Docker : □

Docker is an open platform for developing, shipping, and running applications.

Docker allows you to package an application with all of its dependencies into a standardized unit for software development known as **Container.**

Now What the heck is Container ? 😢

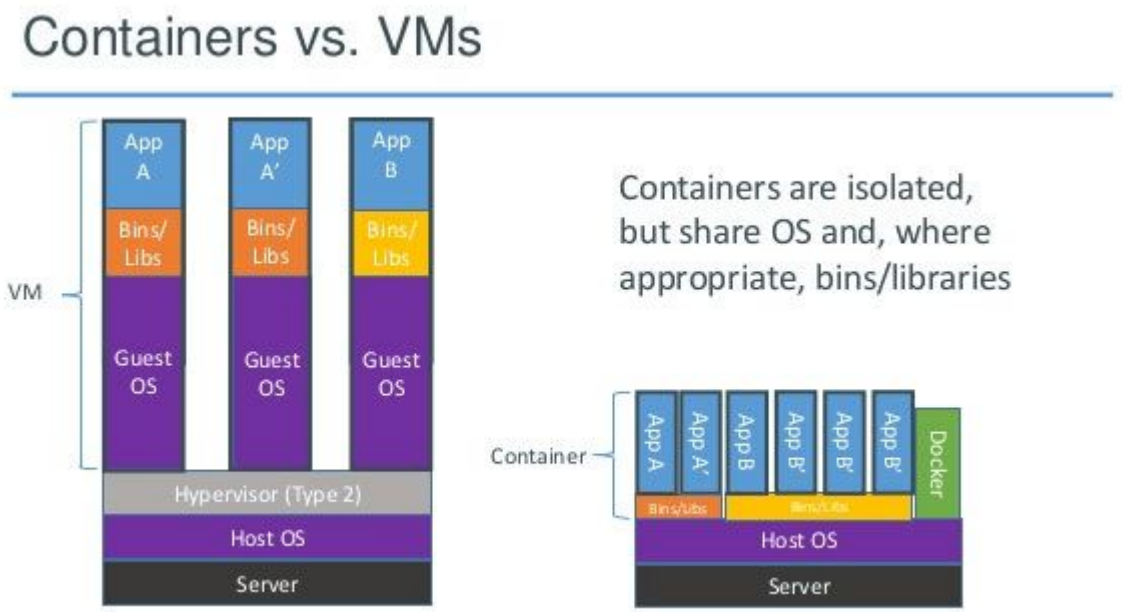
It is a standardized unit of Software.

A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings.

Available for all standard platforms.

Docker VS Virtual Machines

Do you know about Virtual Machines ?



Containers vs. VMs

Containers are isolated, but share OS and, where appropriate, bins/libraries

When it all started ?

- Solomon Hykes (@solomonstre)
- dotCloud (now Docker Inc)
- 260k public repositories on hub.docker.com
- Docker joins the "Open Container Initiative", June 2015
- Recently got converted into the Moby Project
- List of Open Source Tools provided by Docker

Why Docker ?

Docker is Aweeeee-some ! ! ! Because ...

- Fast (deployment, migration, restarts)
- Secure
- Lightweight (save disk & CPU)
- Open Source
- Portable software
- Microservices and integrations (APIs)
- Simplify DevOps
- Version control capabilities

Docker Architecture

Docker components

- (Docker) client
- daemon
- engine
- machine
- compose
- swarm

Docker client

It is the primary user interface to Docker. It accepts commands from the user and communicates back and forth with a Docker daemon.

Docker Daemon :

It runs on a host machine. The user does not directly interact with the daemon, but instead through the Docker client with the RESTful api or sockets.

Docker engine:

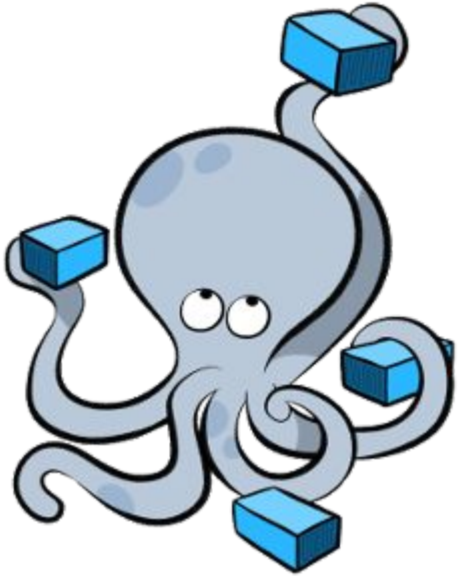
A Client with a Daemon as also as the docker-compose tool. Usually referred simply as "docker".

Docker machine :

A tool which makes it really easy to create
Docker hosts on your computer, on cloud
providers and inside your own data center.
It creates servers, installs Docker on them,
then configures the Docker client to talk to them.
Required for **Mac, Windows** users.

Docker-compose

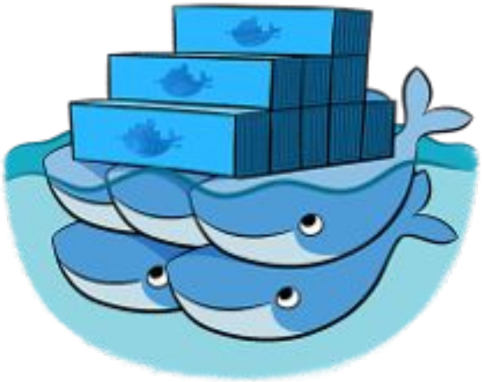
A tool for defining and running complex applications with Docker (eg a multi-container application) with a single file.



Docker-Swarm :

A native clustering tool for Docker. Swarm pools together several Docker hosts and exposes them as a single virtual Docker host.

It scale up to multiple hosts.

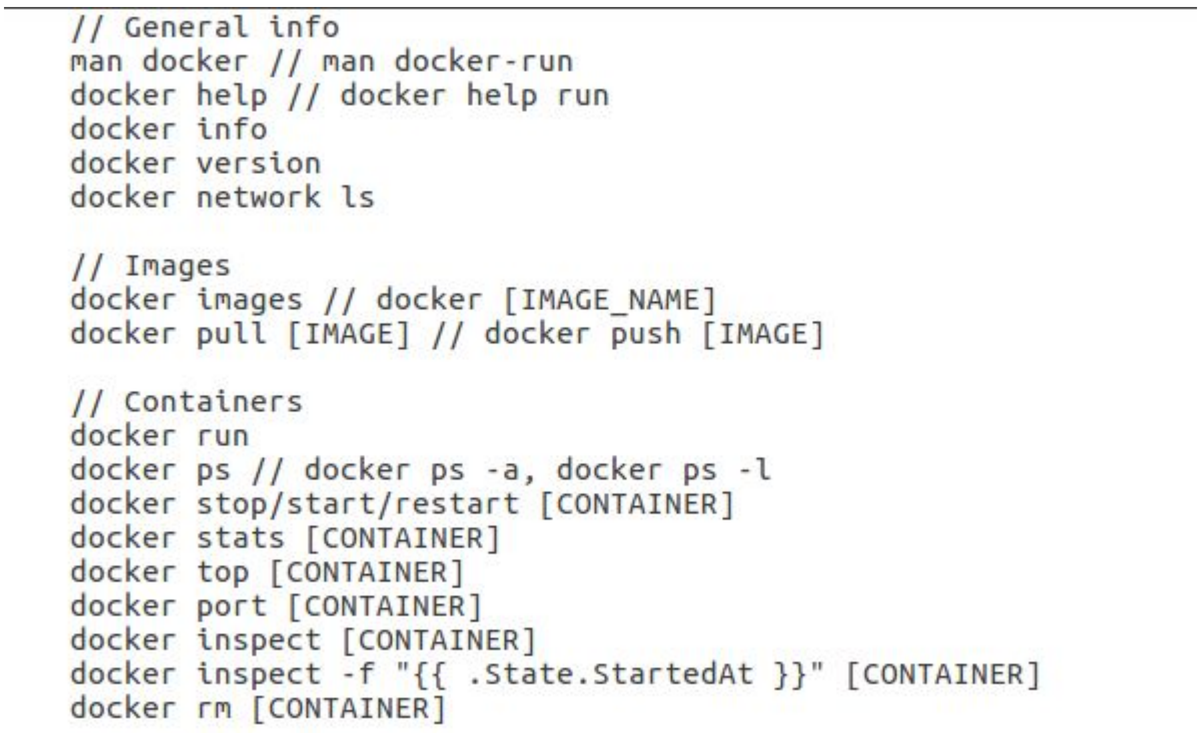


Behind the Scenes

Don’t you feel like Samuel Right now ? 😁

Basic Docker Commands

Some Basic Docker Commands:



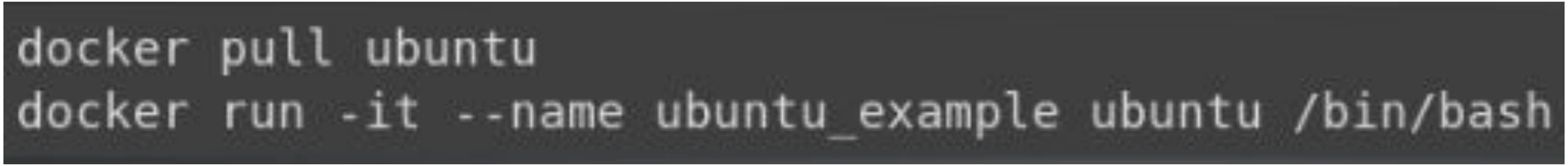
```
// General info
man docker // man docker-run
docker help // docker help run
docker info
docker version
docker network ls

// Images
docker images // docker [IMAGE_NAME]
docker pull [IMAGE] // docker push [IMAGE]

// Containers
docker run
docker ps // docker ps -a, docker ps -l
docker stop/start/restart [CONTAINER]
docker stats [CONTAINER]
docker top [CONTAINER]
docker port [CONTAINER]
docker inspect [CONTAINER]
docker inspect -f "{{ .State.StartedAt }}" [CONTAINER]
docker rm [CONTAINER]
```

DEMO

Docker examples

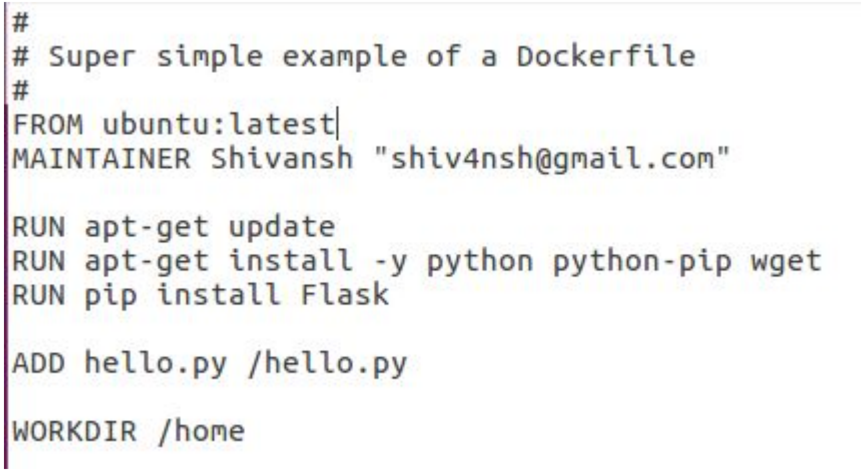
- SSH into a container
- Build an image
- Running the Container
- Pushing an image

SSH into a container



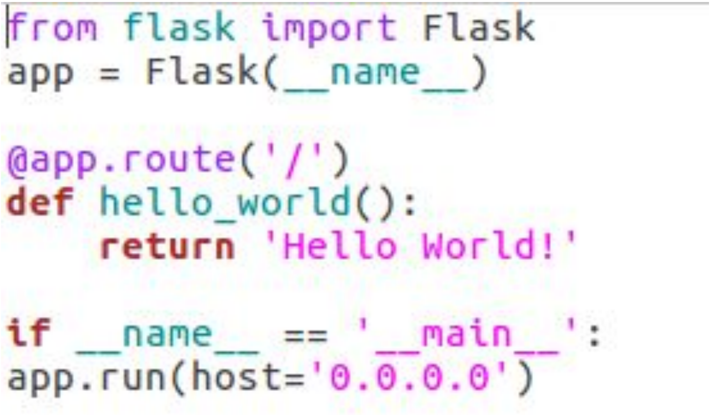
```
docker pull ubuntu
docker run -it --name ubuntu_example ubuntu /bin/bash
```

Build an Image



```
#
# Super simple example of a Dockerfile
#
FROM ubuntu:latest
MAINTAINER Shivansh "shiv4nsh@gmail.com"

RUN apt-get update
RUN apt-get install -y python python-pip wget
RUN pip install Flask

ADD hello.py /hello.py

WORKDIR /home
```

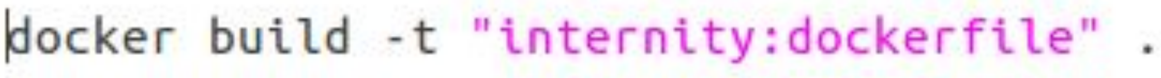
Dockerfile



```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello World!'

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

hello.py

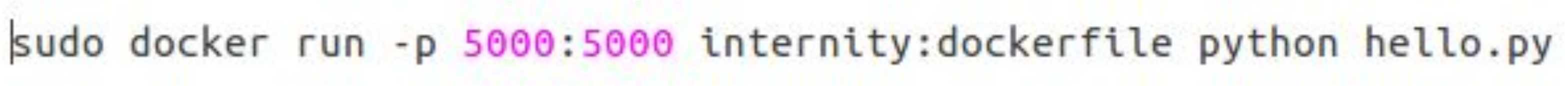
Using Command :



```
docker build -t "internity:dockerfile" .
```

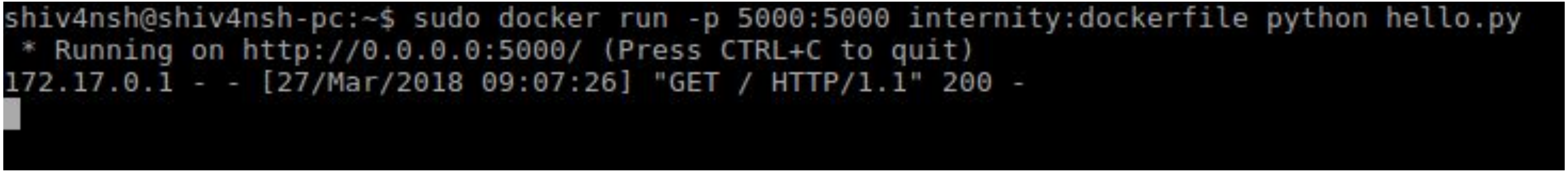
Running the container

Command :



```
sudo docker run -p 5000:5000 internity:dockerfile python hello.py
```

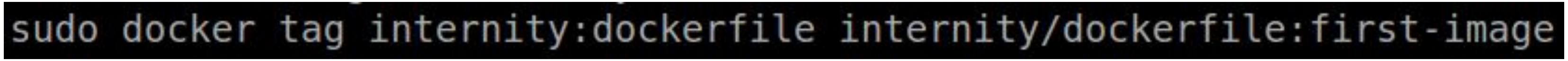
Result :



```
shiv4nsh@shiv4nsh-pc:~$ sudo docker run -p 5000:5000 internity:dockerfile python hello.py
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
172.17.0.1 - - [27/Mar/2018 09:07:26] "GET / HTTP/1.1" 200 -
```

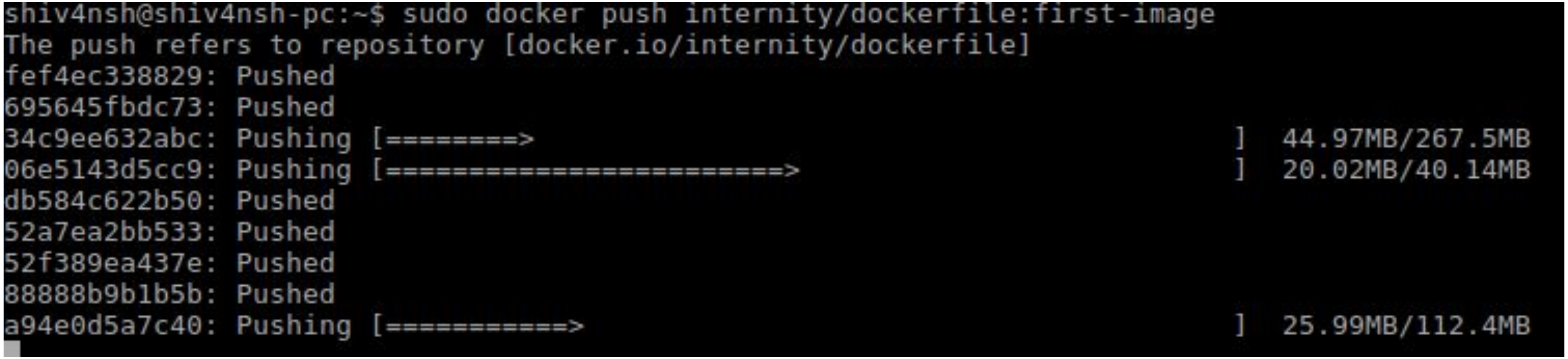
Pushing the Docker Image

Tag the image:



```
sudo docker tag internity:dockerfile internity/dockerfile:first-image
```

Push the image:

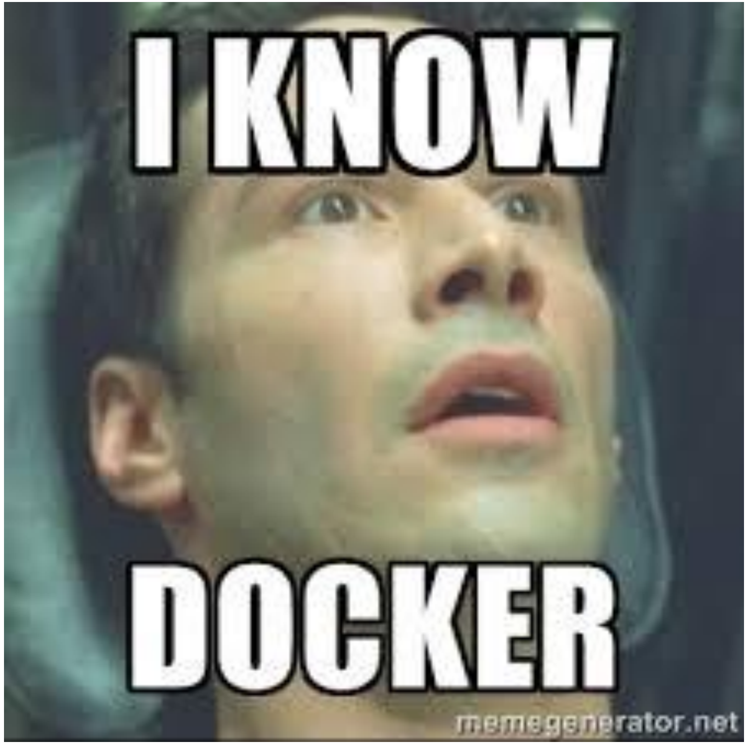


```
shiv4nsh@shiv4nsh-pc:~$ sudo docker push internity/dockerfile:first-image
The push refers to repository [docker.io/internity/dockerfile]
fef4ec338829: Pushed
695645fbdc73: Pushed
34c9ee632abc: Pushing [========>                                          ]  44.97MB/267.5MB
06e5143d5cc9: Pushing [=========================>                         ]  20.02MB/40.14MB
db584c622b50: Pushed
52a7ea2bb533: Pushed
52f389ea437e: Pushed
88888b9b1b5b: Pushed
a94e0d5a7c40: Pushing [===========>                                       ]  25.99MB/112.4MB
```

Resources :

- Awesome Docker (list of Docker resources & projects)
- Docker cheat sheet
- Docker in Practice, The Docker Book (books)
- Docker aliases/shortcuts
- Docker case studies

Do you feel like Neo now ?



ARE THERE
ANY QUESTIONS?
memegenerator.net

Refrences:

1. Docker Documentation
2. Theodorosploumis presentation.