Mithun Technologies
devopstrainingblr@gmail.com

Program Agenda

- Introduction
- Installation
- Virtualization and Containerization
- Virtual Machine and Docker
- Dockerfile, Docker Image and Docker Container
- Docker Commands
- Build the Dockerfile
- Create Docker Custom images and Containers, push to Docker Hub
- Docker Compose
- Docker Swarm

Docker Introduction

Type	Containerization Tool
Vendor	Docker
Is Open Source?	Yes – Some extent
Operating system	Cross Platform
URL	https://www.docker.com/

Pre-Requisite
Download Docker and install from below url
https://www.docker.com/products/docker-toolbox

Register in https://hub.docker.com/

Mithun Technologies

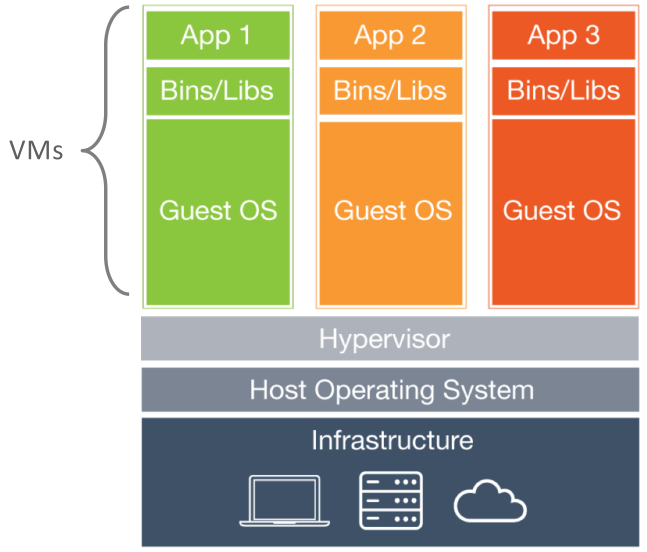
What is Docker?

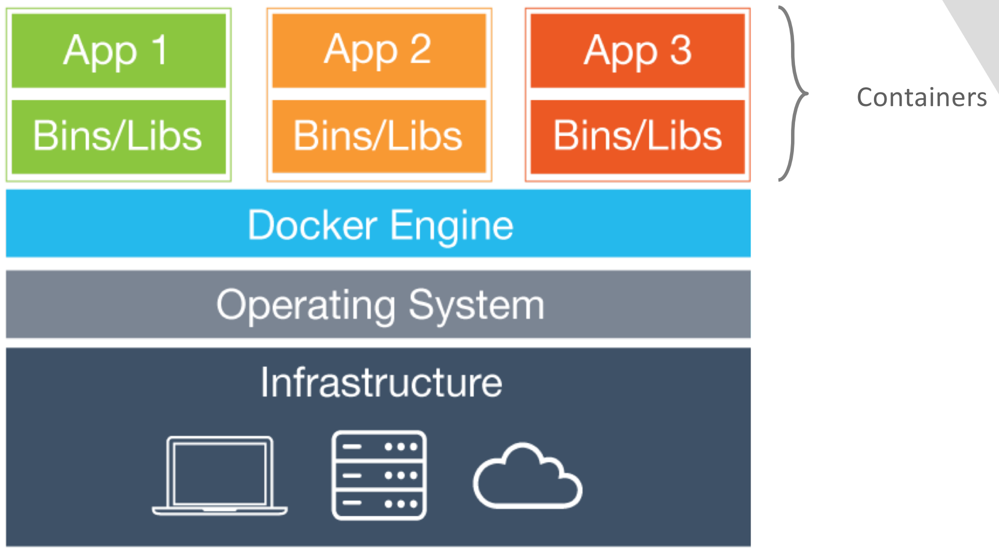
- *Docker is a containerisation platform which packages your app and its all dependencies together in the form of containers. so as to ensure that your application works seamlessly in any environment be it dev or test or prod.*
- *Docker is an open platform for developers and sysadmins to build, ship, and run distributed applications*
- *Docker is available in two editions: Community Edition (CE) and Enterprise Edition (EE).*
- *Docker Community Edition (CE) is ideal for developers and small teams looking to get started with Docker and experimenting with container-based apps. Docker CE has two update channels, stable and edge:*
- *Stable gives you reliable updates every quarter*
- *Edge gives you new features every month*
- *Docker Enterprise Edition (EE) is designed for enterprise development and IT teams who build, ship, and run business critical applications in production at scale.*

Mithun Technologies

Basic Architecture and comparison to VM’s

Virtual Machines vs. Containers





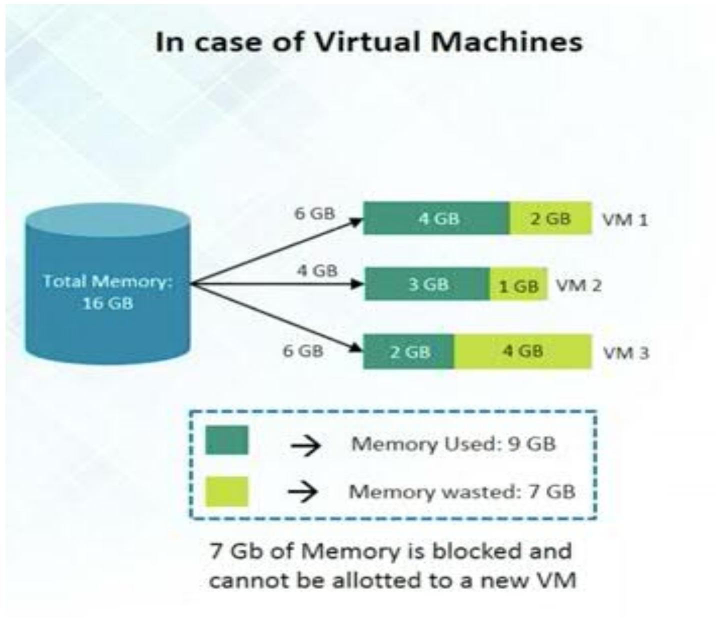
Virtual Machines

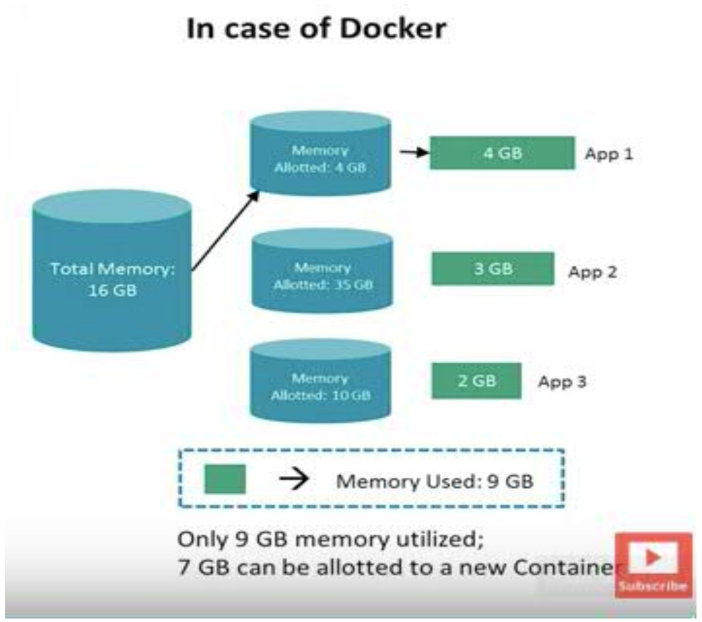
- Each virtual machine (VM) includes the app, the necessary binaries and libraries and an **entire guest operating system**

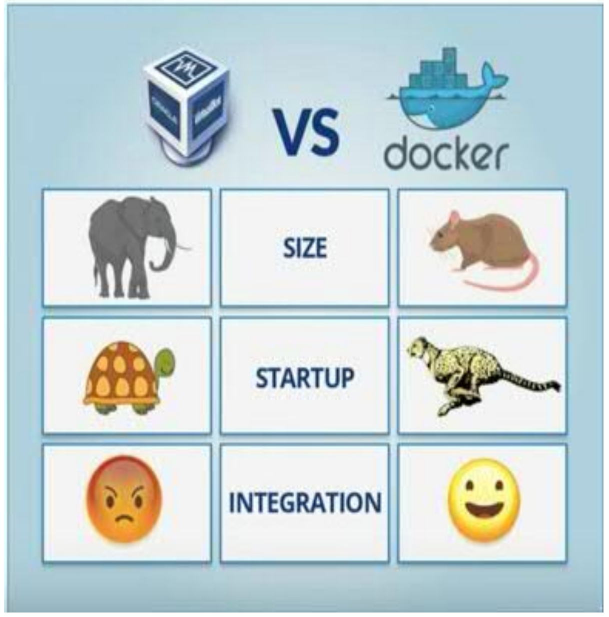
Containers

- Containers include the app & all of its dependencies, but **share the kernel** with other containers.
- Run as an isolated process in userspace on the host OS
- Not tied to any specific infrastructure – containers run on any computer, infrastructure and cloud.

Virtual Machines vs. Containers





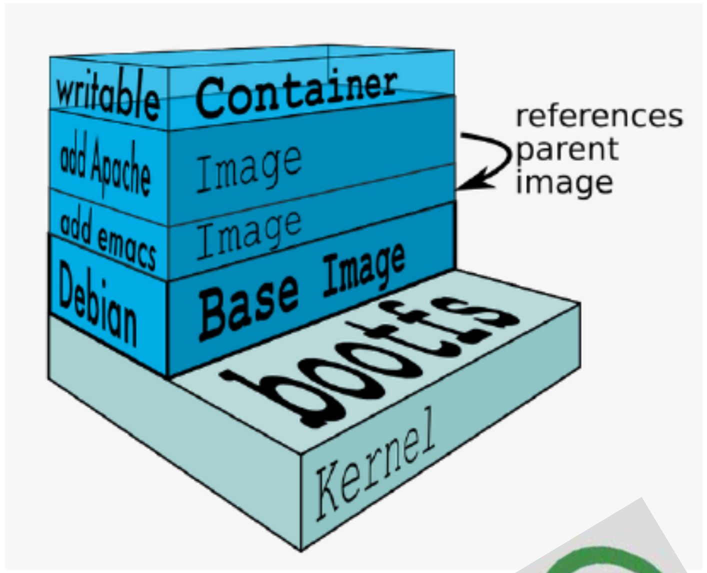


Dockerfile

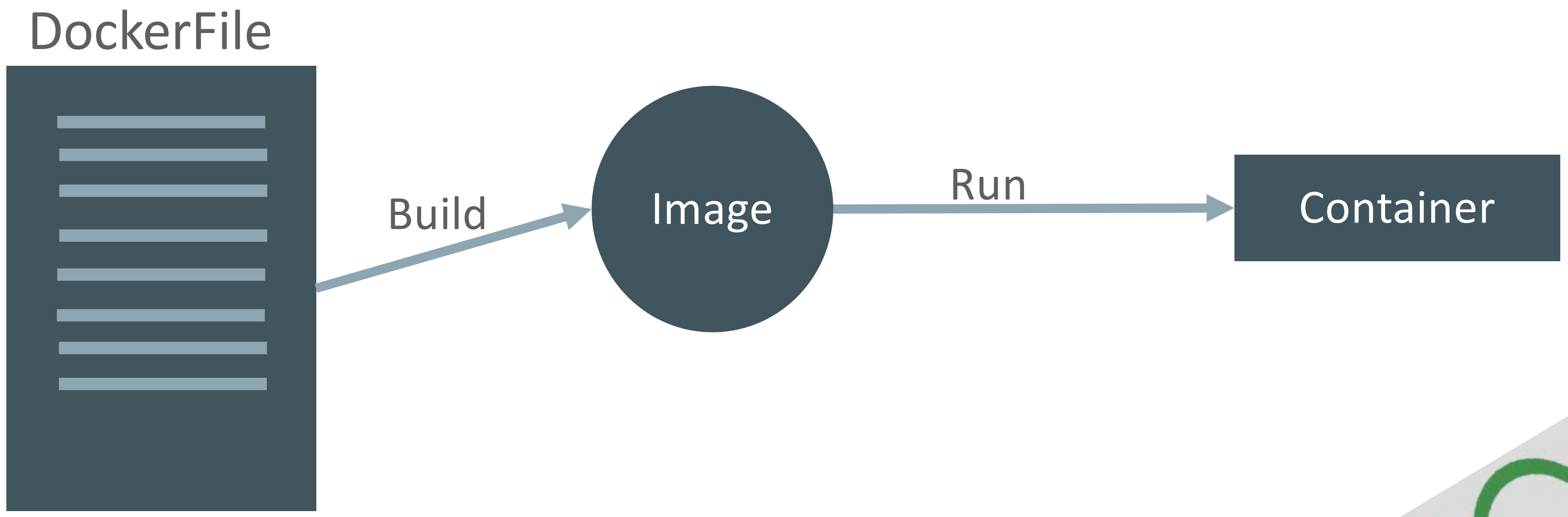
- Dockerfile is a file, it will describe the steps in order to create an image quickly.
- The Dockerfile uses a basic DSL (Domain Specific Language) with instructions for building Docker images.
- The Docker daemon runs the instructions in the Dockerfile are processed from the top down, so you should order them accordingly.
- The Dockerfile is the source code of the Docker Image.

Docker Images

- An **image** is a lightweight, stand-alone, executable package that includes everything needed to run a piece of software, including the code, a runtime, libraries, environment variables, and config files.



Docker Flow:



Docker Flow:

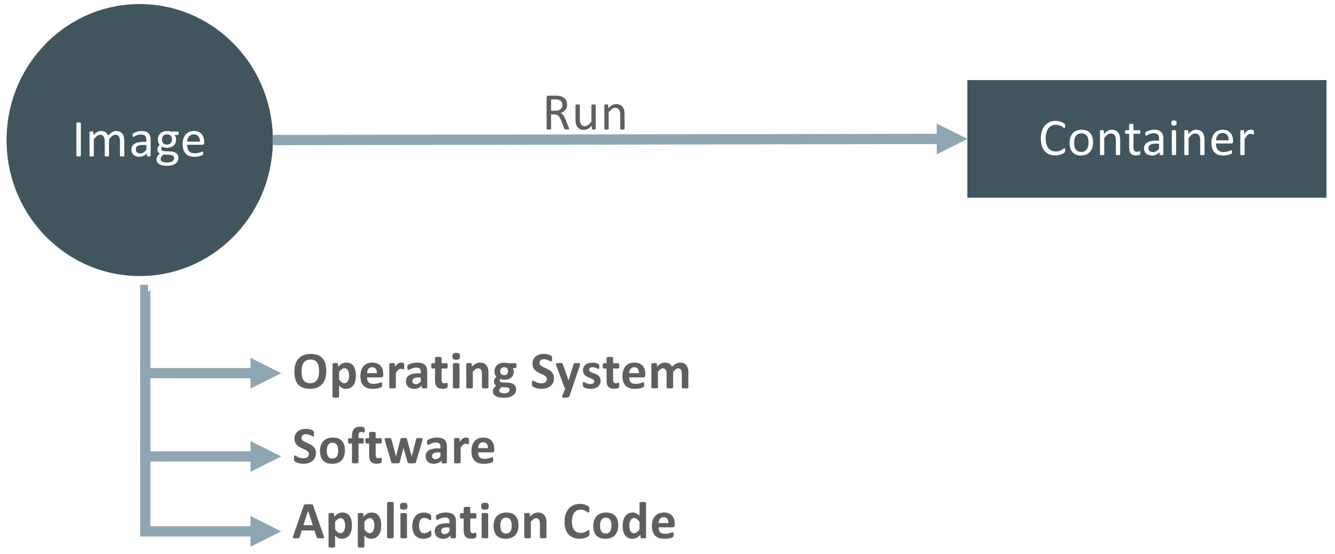


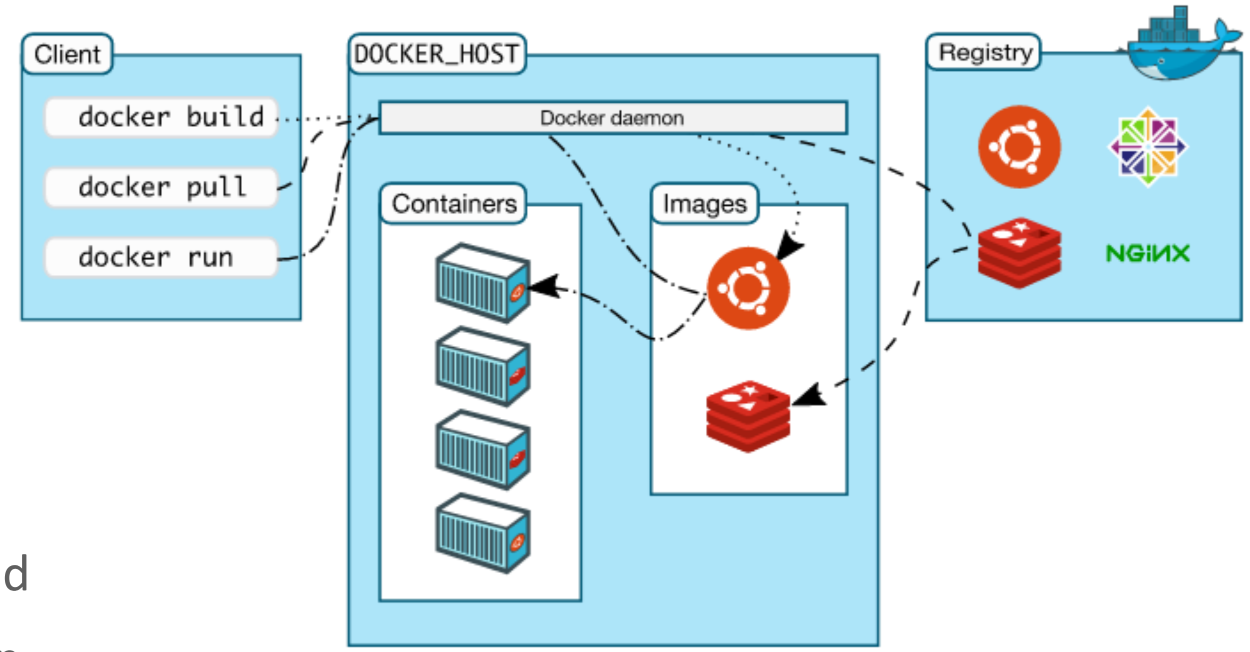
Image
Run
Container
Operating System
Software
Application Code

Docker Container Lifecycle

- Conception
  - BUILD an Image from a Dockerfile
- Birth
  - RUN (create+start) a container
- Reproduction
  - COMMIT (persist) a container to a new image
  - RUN a new container from an image
- Sleep
  - KILL a running container
- Wake
  - START a stopped container
- Death
  - RM (delete) a stopped container
- Extinction
  - RMI a container image (delete image)

Docker Architecture

- Docker client – Command Line Interface (CLI) for interfacing with the Docker
- Dockerfile – Text file of Docker instructions used to assemble a Docker Image
- Image – Hierarchies of files built from a Dockerfile, the file used as input to the docker build command
- Container – Running instance of an Image using the docker run command
- Registry – Image repository



Docker Features….

- Light-Weight
  - Minimal overhead (cpu/io/network)
  - Based on Linux containers
  - Decrease storage consumption
  - Uses layered filesystem to save space (AUFS/LVM)
- Portable
  - Run it Everywhere! - Linux, Mac OS or Windows operating system that has Docker installed.
  - Raspberry pi support.
  - Move from one environment to another by using the same Docker technology.
- Self-sufficient
  - A Docker container contains everything it needs to run
  - Minimal Base OS
  - Libraries and frameworks
  - Application code
  - A Docker container should be able to run anywhere that Docker can run.

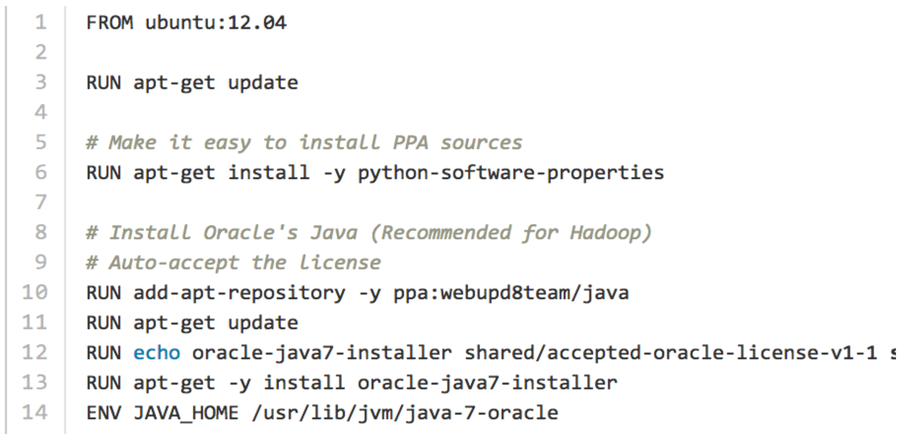
Mithun Technologies

Common Terms and commands:

- Docker Engine – Docker Deamon, Docker Registry, CLI.
- Image – operating systems kernels supplied for a specific instance type / application.
- Container – an application running from an image.
- DockerFile – a text file with a list of steps to perform to create an image.
- Docker Hub – Docker Registry and Repository used for download and share images.

Dockerfile ……

- Like a Makefile (shell script with keywords)
- Extends from a Base Image
- Results in a new Docker Image
- Imperative, not Declarative



```
FROM ubuntu:12.04

RUN apt-get update

# Make it easy to install PPA sources
RUN apt-get install -y python-software-properties

# Install Oracle's Java (Recommended for Hadoop)
# Auto-accept the license
RUN add-apt-repository -y ppa:webupd8team/java
RUN apt-get update
RUN echo oracle-java7-installer shared/accepted-oracle-license-v1-1 s
RUN apt-get -y install oracle-java7-installer
ENV JAVA_HOME /usr/lib/jvm/java-7-oracle
```

- A Docker file lists the steps needed to build an images
- docker build is used to run a Docker file
- Can define default command for docker run, ports to expose, etc

Dockerfile – Text file (recipe) used to create Docker images

Example Hello World Dockerfile

```
FROM nginx:1.10.1-alpine
Add index.html /usr/share/nginx/html/index.html
# Override the nginx start from the base container
COPY start.sh /start.sh
RUN chmod +x /start.sh
ENTRYPOINT ["/start.sh"]
```

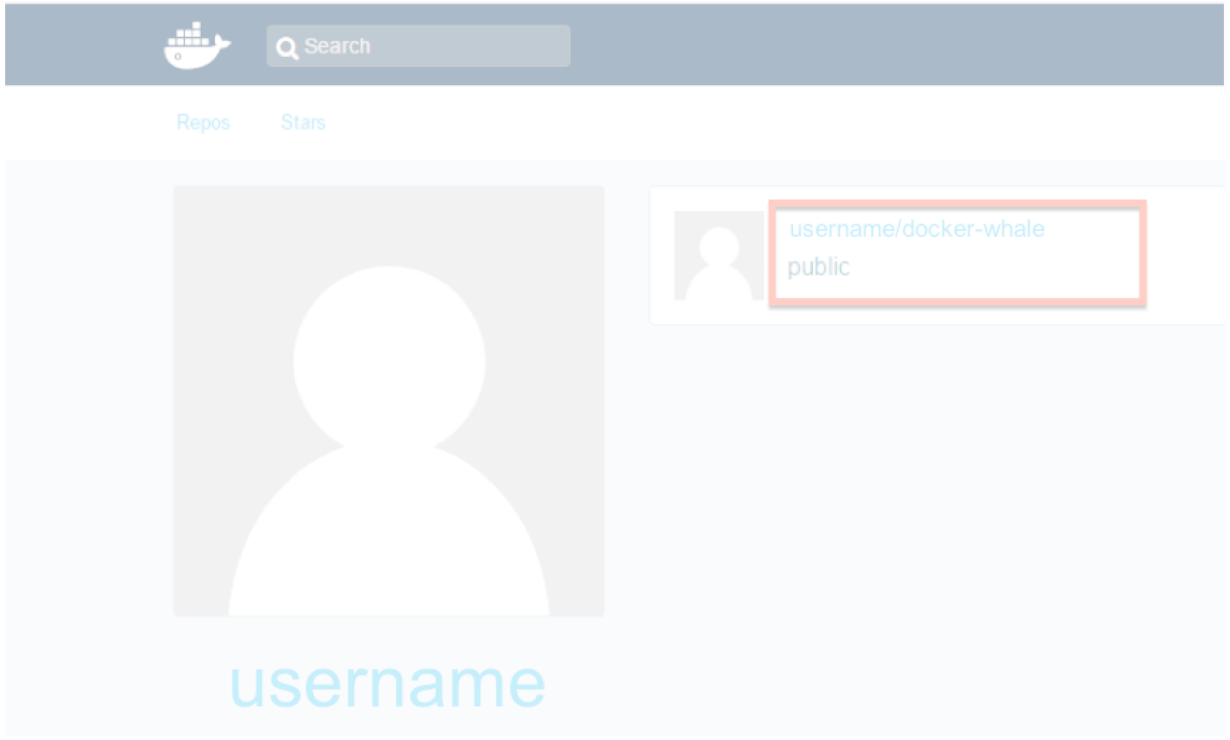
Docker build image CLI example

```
$ docker build -t helloworld:1.0  .
```

NOTE: The "." references Dockerfile in local directory

Docker Hub

- Docker Inc.
  - Repository
  - public and private images



- Enables images to be shared and moved off the laptop

- Example usage:
  - $ docker tag docker-whale:latest username/docker-whale:latest
  - $ docker push username/docker-whale:latest
  - $ docker pull username/docker-whale:latest

Mithun Technologies

Docker CLI – Common / useful commands

- docker build : build docker image from Dockerfile
- docker run : run docker image
- docker logs : show log data for a running or stopped container
- docker ps : list running docker containers (analogous to ps)
- docker ps –a : list all containers including not running
- docker images : list all images on the local volume
- docker rm : remove/delete a container | docker rmi : remove/delete an image
- docker tag : name a docker image
- docker login : login to registry
- docker push/pull : push or pull volumes to/from Docker Registries
- docker inspect : return container run time configuration parameter metadata

Mithun Technologies

Docker Run

Pulls the image and runs it as a container

- Examples:
  - Simple:

```
$ docker run hello-world
```

  - Complex:

```
$ docker run -d  --restart=always  -p=443:5000/tcp
-e="REGISTRY_HTTP_TLS_CERTIFICATE=/certs/domain.crt"
-e="REGISTRY_HTTP_TLS_KEY=/certs/registry.example.com.key"
-e="REGISTRY_AUTH=htpasswd"
-e="REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd"
-e="REGISTRY_AUTH_HTPASSWD_REALM=Our Test Registry"
-v=/home/opc/certs:/certs   -v=/home/opc/auth:/auth
-v /home/opc/registry:/var/lib/registry  "registry:2"
```

Why Containers?

Developers care because:

- Quickly create ready-to-run packaged applications, low cost deployment and replay
- Automate testing, integration, packaging
- Reduce / eliminate platform compatibility issues (“It works in dev!”)
- Support next gen applications (microservices)

IT cares because:

- Improve **speed** and frequency of releases, reliability of deployments
- Makes app lifecycle efficient, consistent and repeatable – configure once, run many times
- Eliminate environment inconsistencies between development, test, production
- Improve production application resiliency and scale out / in on demand

Common Terms and commands:

- Tag – give a name to an image.
- Compose – create multi container application tier for running a service / stack. Done by running a YAML file.
- Exec – starts another process in an existing container (commonly used for debugging, DB administration, etc..
- Logs – keeps the output of a containers.
- Swarm – Docker Cluster features and Orchestration.

Questions ?

Mithun Technologies

Thank You

Mithun Technologies