TATA CONSULTANCY SERVICES
Experience certainty.

Docker

World's leading software containerization platform

Agenda

What is Docker?

Docker vs. Virtual Machine

History, Run Platforms

Features

Images and Containers

Volume Mounting,Networking

Docker Use Cases

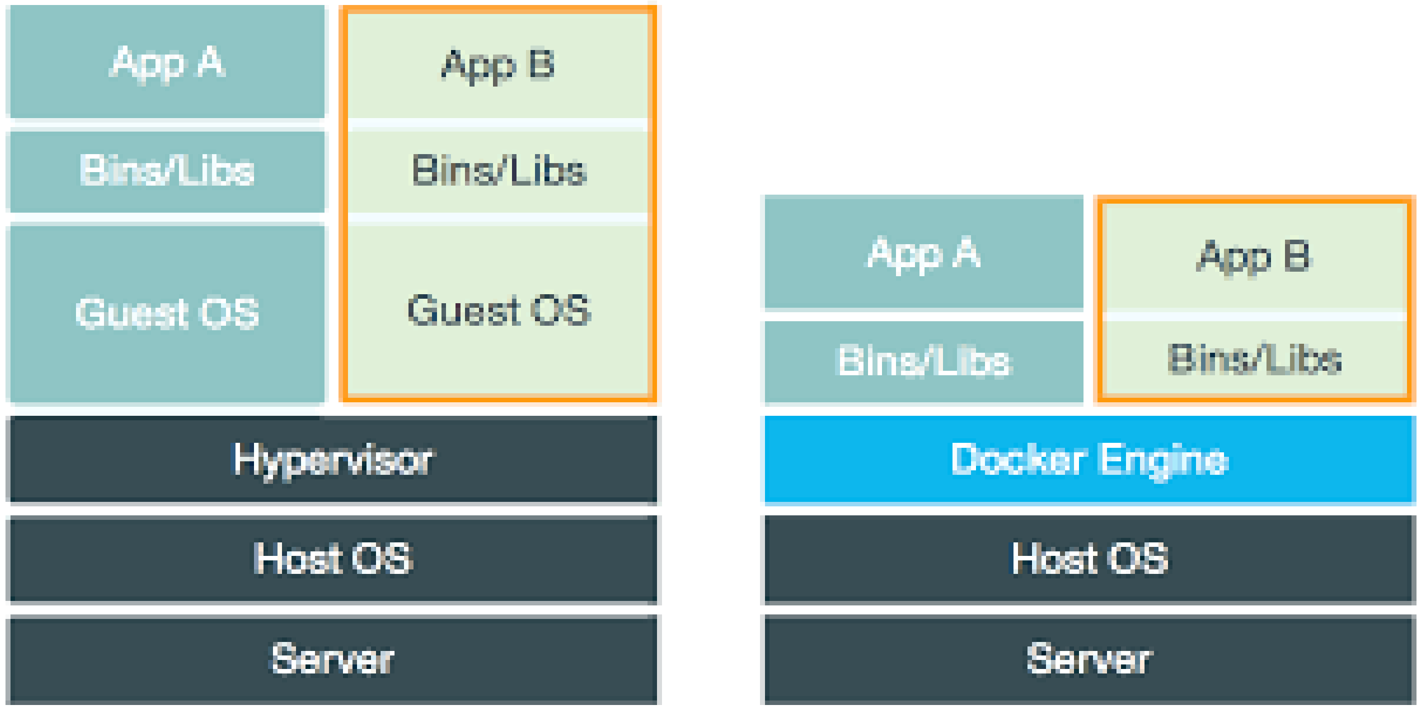
What is Docker?

Docker is an open-source project that automates the deployment of applications inside software containers, by providing an additional layer of abstraction and automation of operating system–level virtualization on Linux.

VM vs Containers

Virtual Machine: Each virtualized application includes the application,binaries, Libraries and entire guest operating system

Containers:Contains just application and its dependencies



Docker History& Platforms

History:

2013-03: Releases as Open Source

2013-09: Red Hat collaboration (Fedora, RHEL, OpenShift)

2014-03: 34th most starred GitHub project

2014-05: JAX Innovation Award (most innovative open technology)

Platforms:

Various Linux distributions (Ubuntu, Fedora, RHEL, Centos, openSUSE, ...)

Cloud (Amazon EC2, Google Compute Engine, Rackspace)

2014-10: Microsoft announces plans to integrate Docker with next release of Windows Server

Docker Features

Light-Weight

Minimal overhead (cpu/io/network)

Based on Linux containers

Uses layered filesystem to save space (AUFS/LVM)

Uses a copy-on-write filesystem to track changes

Portable

Can run on any Linux system that supports LXC (today).

0.7 release includes support for RedHat/Fedora family.

Future plans to support other container tools (lmctfy, etc.)

Possible future support for other operating systems (Solaris, OSX, Windows?)

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.

Docker componets

Docker Deamon-Runs on host machine

Docker client-Primary User interface to Docker

Docker images-Read-only templates

Docker registries-Hold images

Docker Containers-Hold everything needed for the application to run

Docker Image

Persisted snapshot that can be run

images: List all local images

run: Create a container from an image and execute a command in it

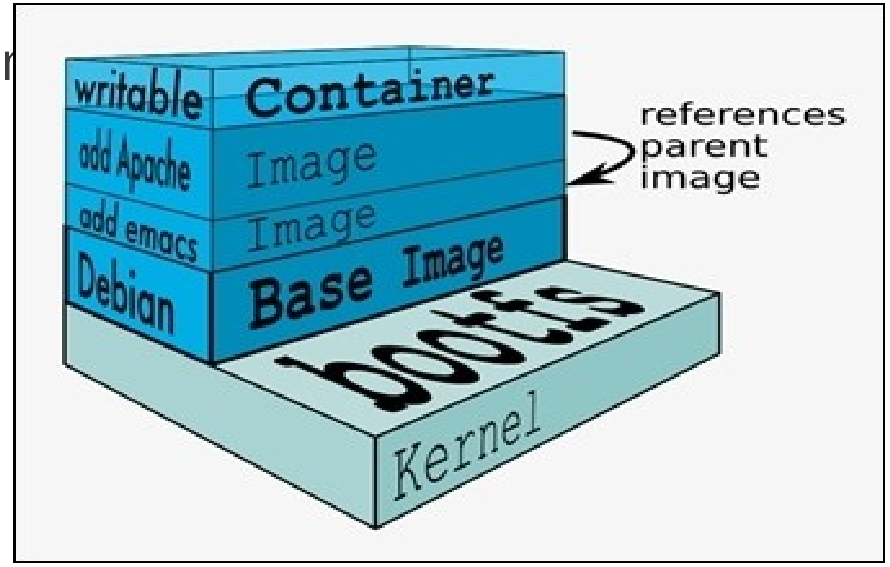
tag: Tag an image

pull: Download image from repository

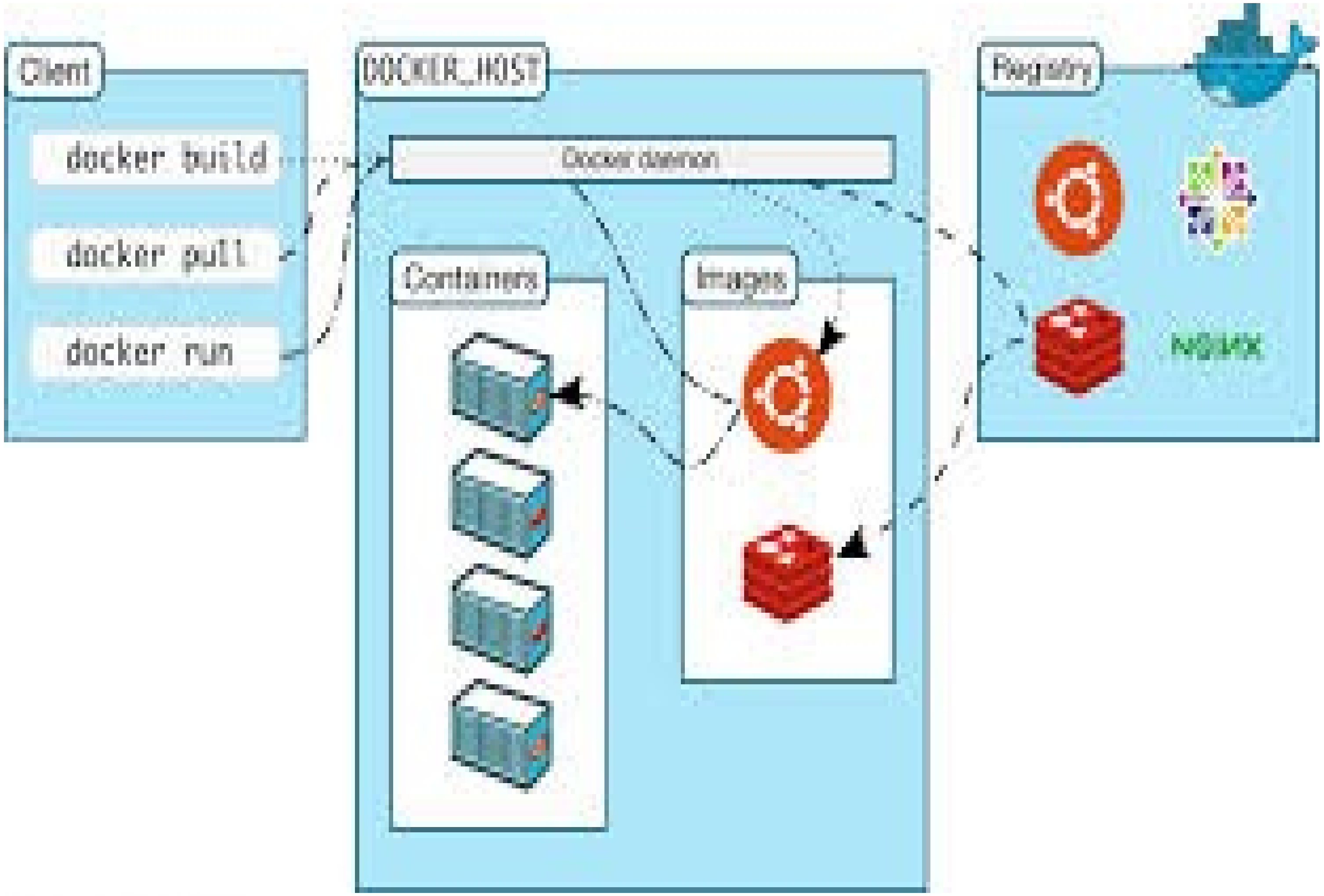
rmi: Delete a local image

This will also remove intermediate i
longer used

Layered File System



Docker Registry



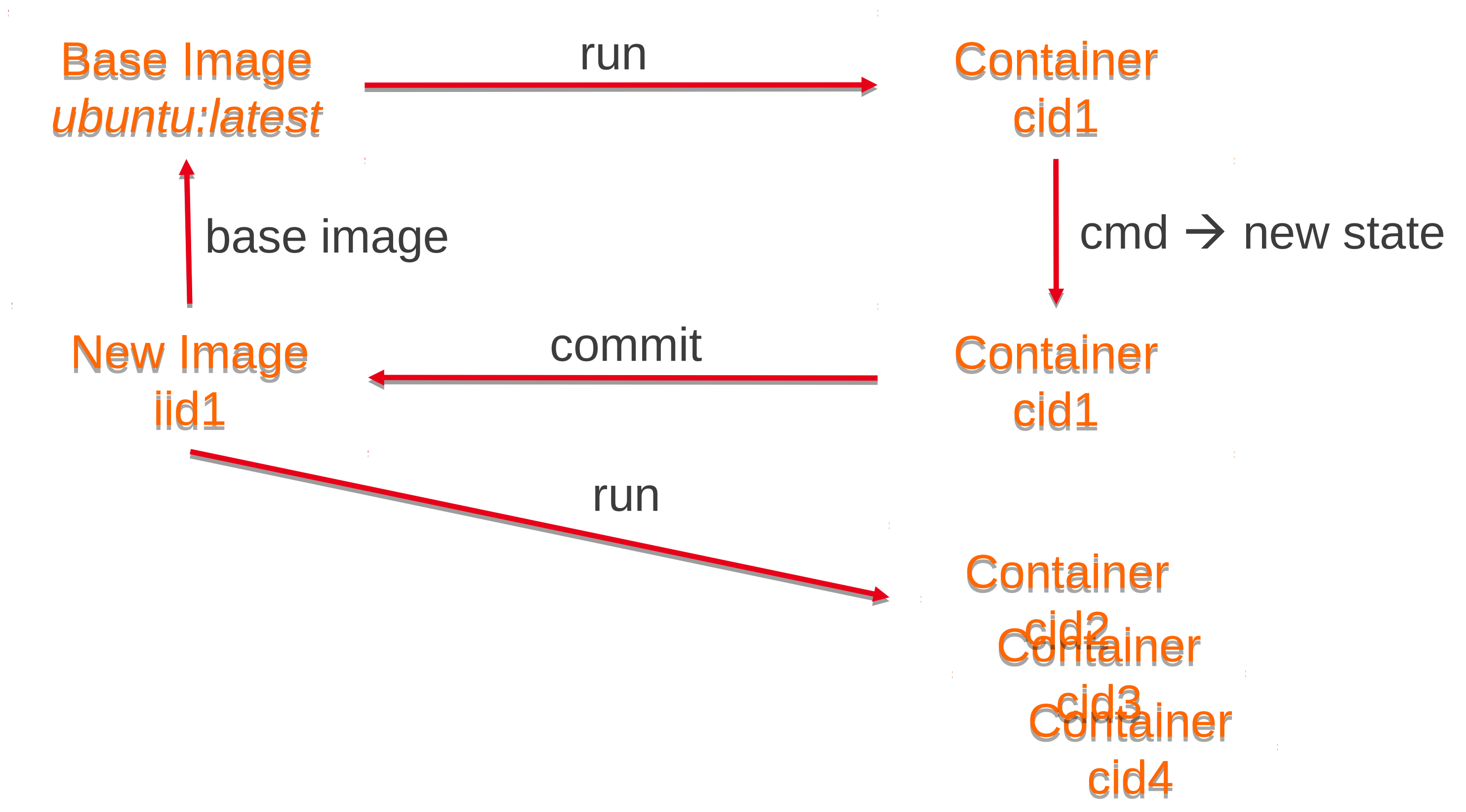
Docker Container Lifecycle

The Life of a Container

- 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)

Image vs. Container



Docker File

Create images automatically using a build script: «Dockerfile»

Can be versioned in a version control system like Git or SVN, along with all dependencies

Docker Hub can automatically build images based on dockerfiles on Github

Dockerfile:

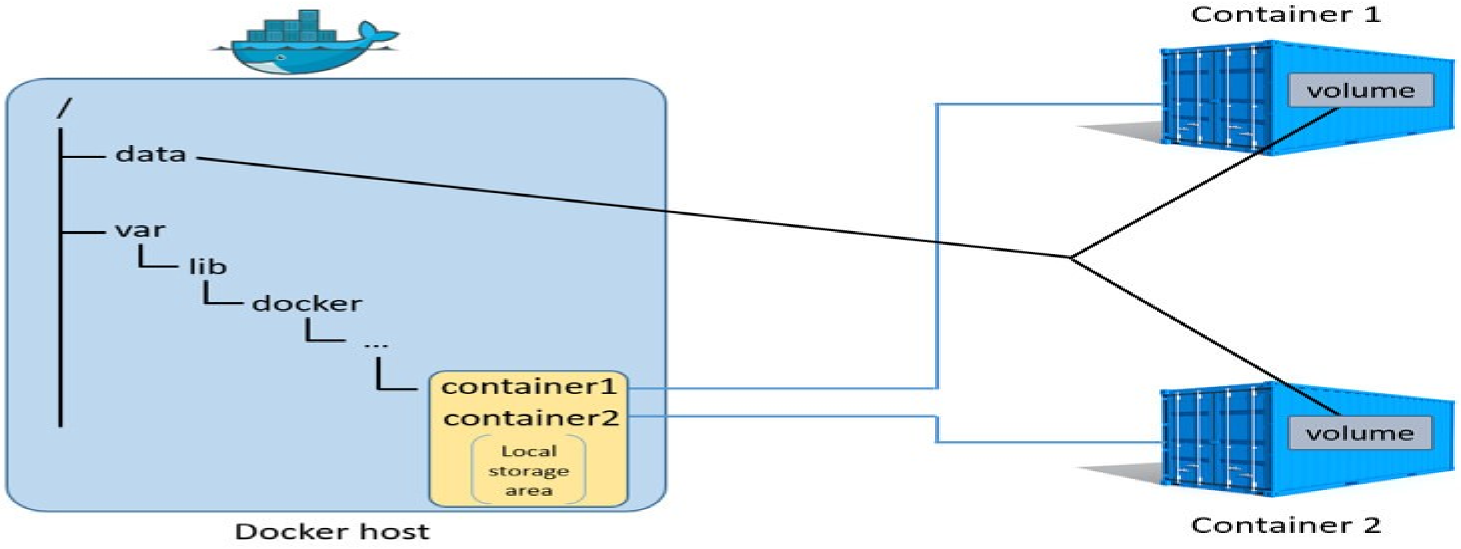
```
FROM ubuntu
ENV DOCK_MESSAGE Hello My World
ADD dir /files
CMD ["bash", "someScript"]
docker build [DockerFileDir]
docker inspect [imageId]
```

Docker Volumes

docker run –ti –v /hostLog:/log ubuntu

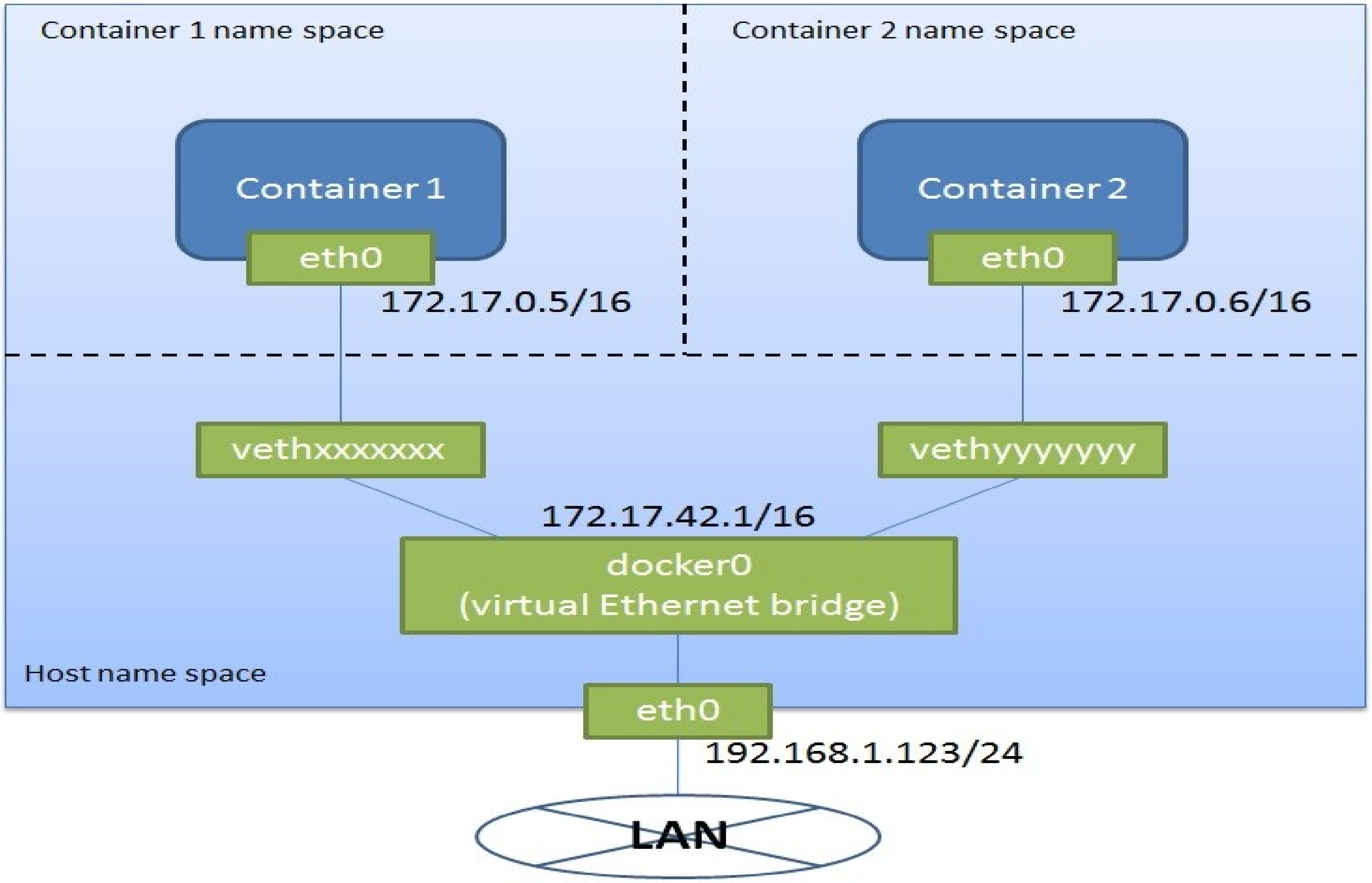
Run second container: Volume can be shared

docker run –ti --volumes-from firstContainerName ubuntu



Docker Networking

By default Docker containers are connected only to a virtual network .



Docker usecases

DevOps

Development Environment

Environments for Integration Tests

Quick evaluation of software

- Microservices

Unified execution environment (dev → test → prod (local, VM, cloud, ...)

Thank You