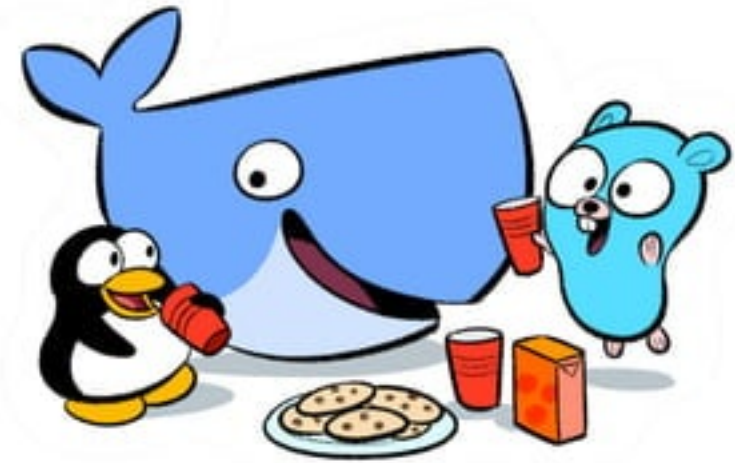


Introduction to Docker

Peng Xiao
GitHub: xiaopeng163
Network Consulting Engineer
Cisco Systems

Contents

- The Metrix from Hell
- What is Container/Docker?
- Docker Basic Usage
- Docker Networking Deep Dive
- Docker Compose
- Docker Swarm
- Ecosystem and Standardization

docker

The Challenge

Multiplicity of Stacks

Static website

nginx 1.5 + modsecurity + openssl + bootstrap 2

User DB

postgresql + pgv8 + v8

Queue

Redis + redis-sentinel

Analytics DB

hadoop + hive + thrift + OpenJDK

Background workers

Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs

Web frontend

Ruby + Rails + sass + Unicorn

API endpoint

Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client

Do services and apps interact appropriately?

Multiplicity of hardware environments

Development VM

QA server

Public Cloud

Production Cluster

Disaster recovery

Customer Data Center

Production Servers

Contributor's laptop

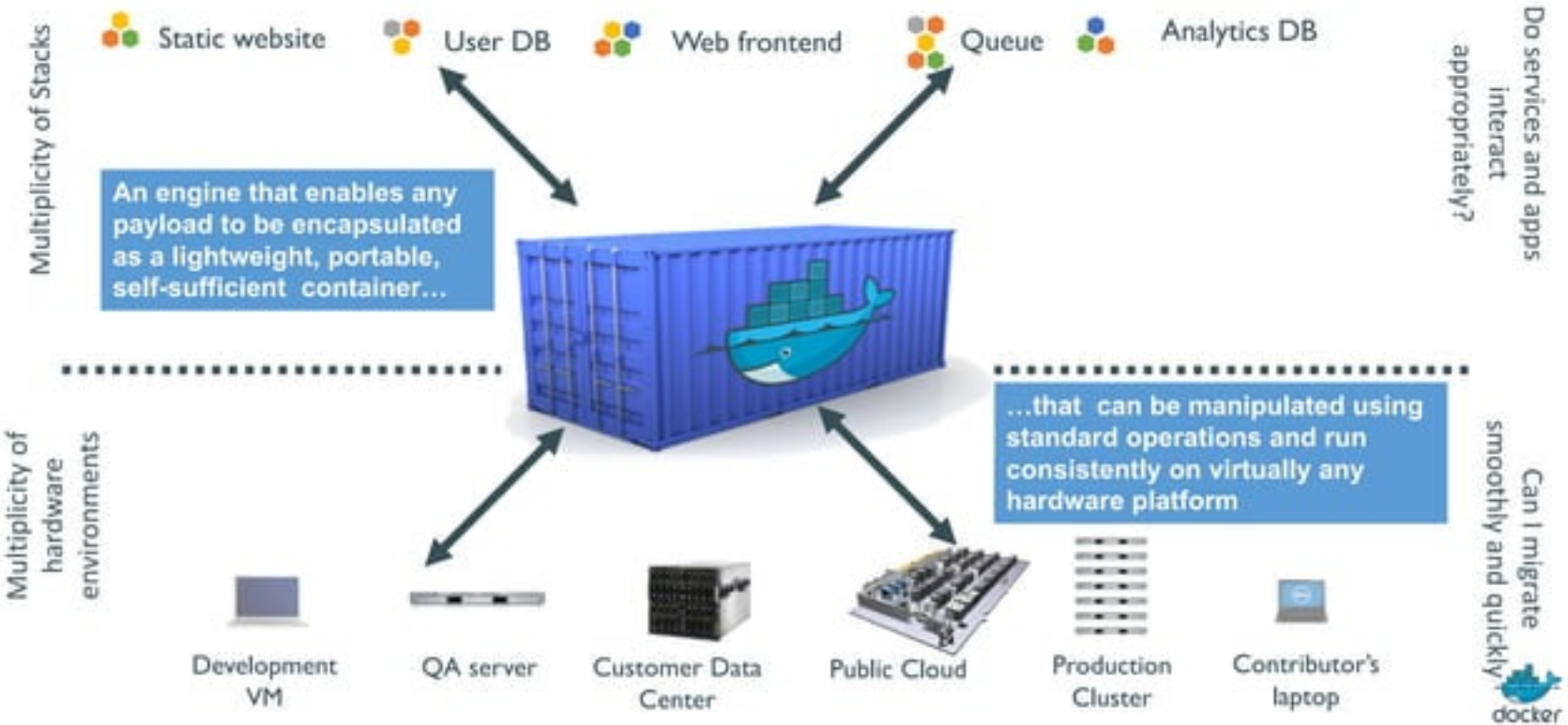
Can I migrate smoothly and quickly?

docker

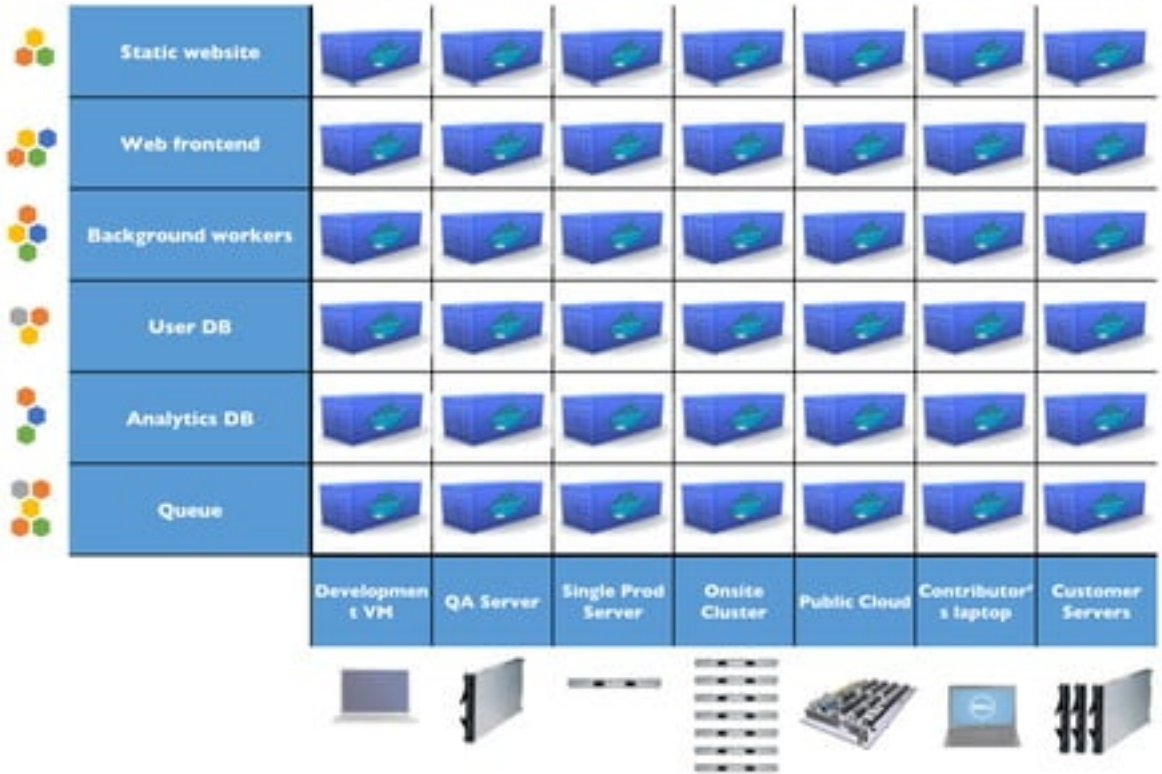
The Matrix From Hell

	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers
Static website	?	?	?	?	?	?	?
Web frontend	?	?	?	?	?	?	?
Background workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?

Docker is a shipping container system for code



Docker eliminates the matrix from Hell



Why Developers Care

- Build once…(finally) run anywhere*
  - A clean, safe, hygienic and portable runtime environment for your app.
  - No worries about missing dependencies, packages and other pain points during subsequent deployments.
  - Run each app in its own isolated container, so you can run various versions of libraries and other dependencies for each app without worrying
  - Automate testing, integration, packaging…anything you can script
  - Reduce/eliminate concerns about compatibility on different platforms, either your own or your customers.
  - Cheap, zero-penalty containers to deploy services? A VM without the overhead of a VM? Instant replay and reset of image snapshots? That's the power of Docker

* With the 0.7 release, we support any x86 server running a modern Linux kernel (3.2+ generally. 2.6.32+ for RHEL 6.5+, Fedora, & related)

docker

Why Devops Cares?

- Configure once…run anything
  - Make the entire lifecycle more efficient, consistent, and repeatable
  - Increase the quality of code produced by developers.
  - Eliminate inconsistencies between development, test, production, and customer environments
  - Support segregation of duties
  - Significantly improves the speed and reliability of continuous deployment and continuous integration systems
  - Because the containers are so lightweight, address significant performance, costs, deployment, and portability issues normally associated with VMs

docker

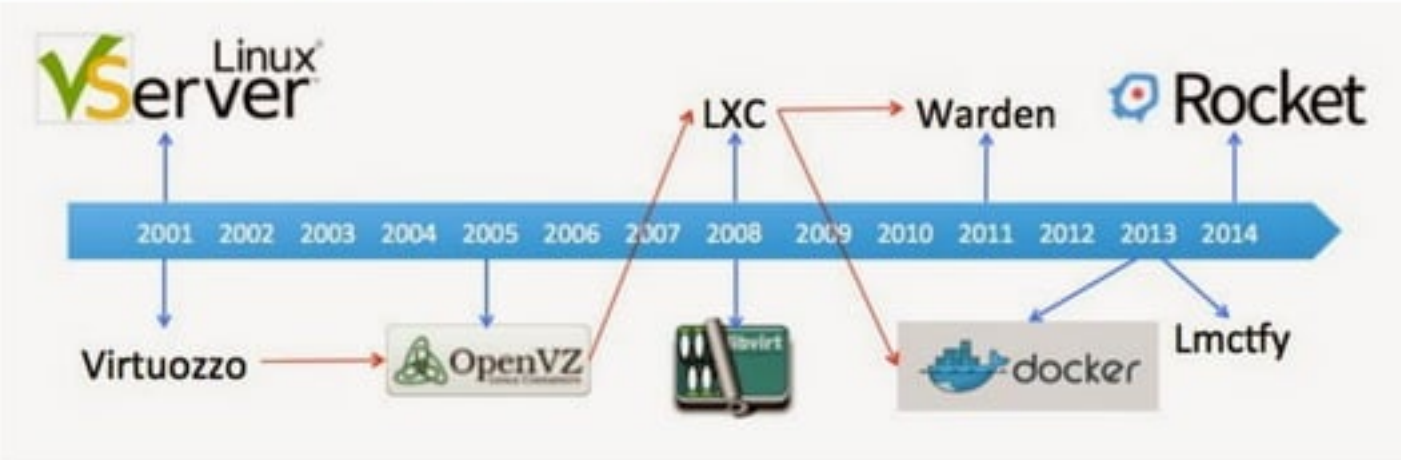
Contents

- The Metrix from Hell
- **What is Container/Docker?**
- Docker Basic Usage
- Docker Networking Deep Dive
- Docker Compose
- Docker Swarm
- Ecosystem and Standardization

docker

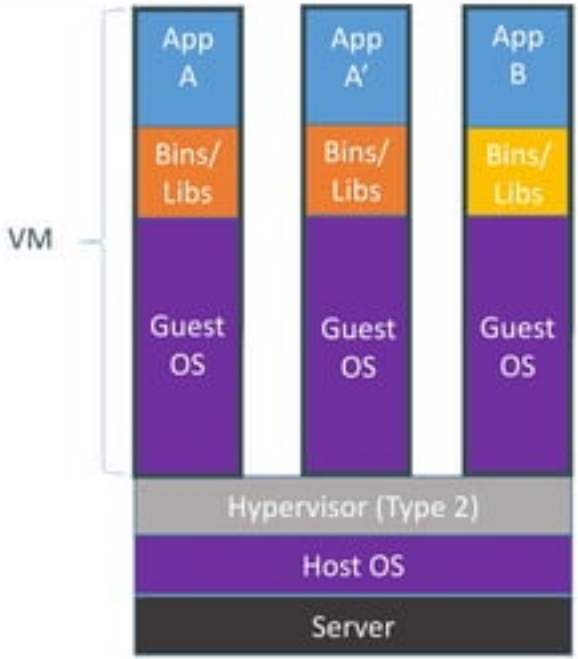
Container History

Docker is nothing new, it just show in the right time



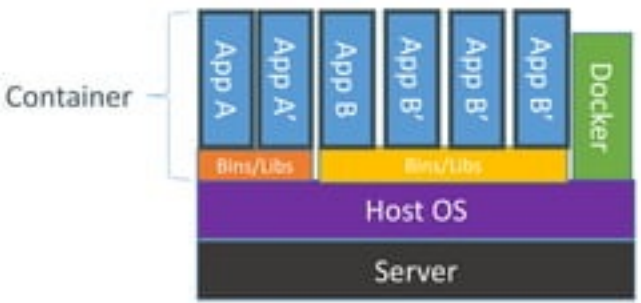
http://kiwenlau.blogspot.com/2015/01/linux-container-technology-overview.html

Containers vs. VMs



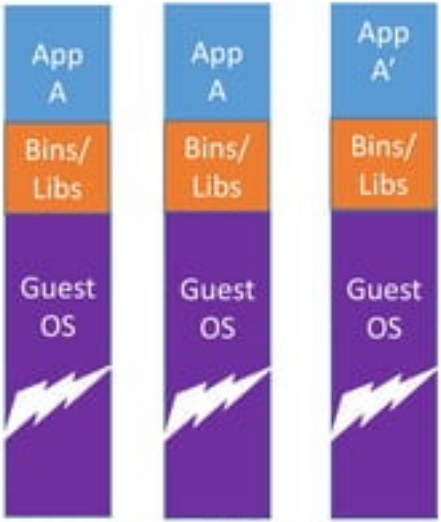
Containers are isolated, but share OS and, where appropriate, bins/libraries

...result is significantly faster deployment, much less overhead, easier migration, faster restart



Why are Containers lightweight?

VMs



VMs

Every app, every copy of an app, and every slight modification of the app requires a new virtual server

Containers

Original App
(No OS to take up space, resources, or require restart)

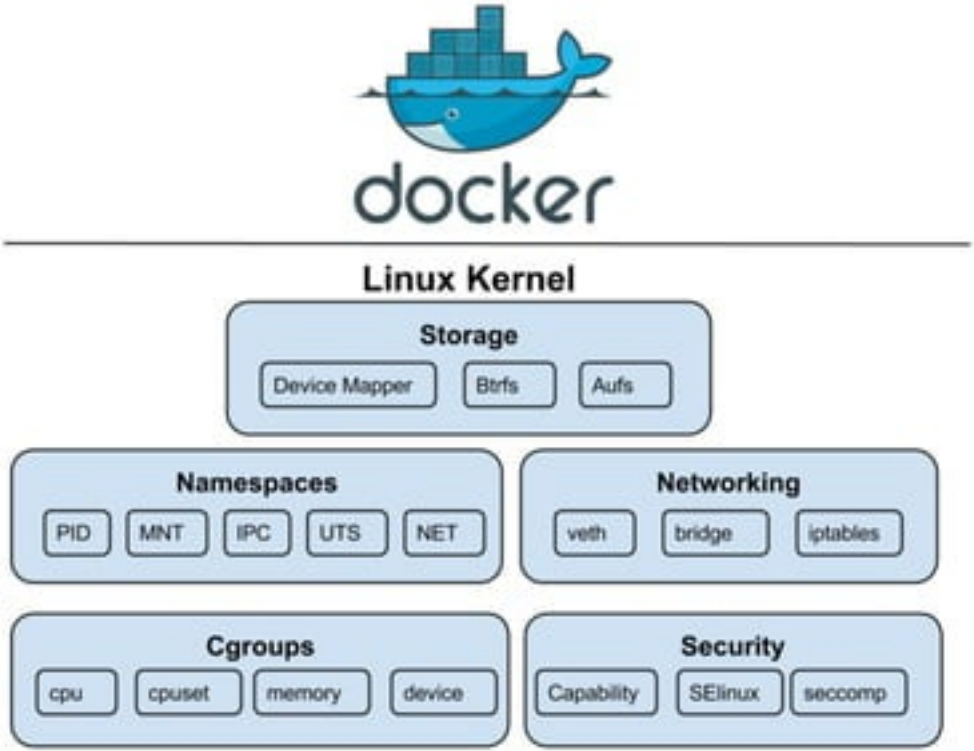
Copy of App
No OS. Can Share bins/libs

Modified App

Copy on write capabilities allow us to only save the diffs Between container A and container A'

docker

Docker Inside



Docker Inside (Deep Dive)

- A Deep Dive Into Linux Containers For Engineers Interested In The Gritty Details. http://docker-saigon.github.io/post/Docker-Internals/
- Cgroups, namespaces, and beyond: what are containers made from? (DockerCon Europe 2015) https://goo.gl/25KtpZ

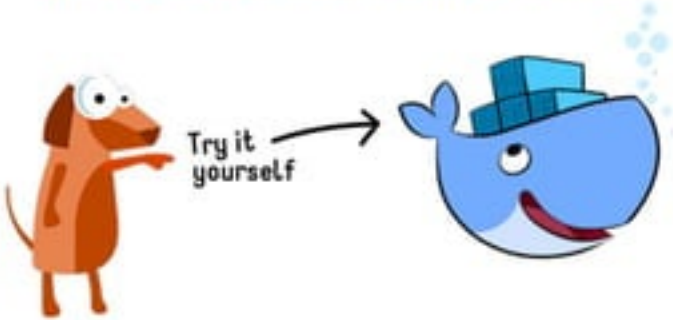
Contents

- The Metrix from Hell
- What is Container/Docker?
- **Docker Basic Usage**
- Docker Networking Deep Dive
- Docker Compose
- Docker Swarm
- Ecosystem and Standardization

docker

Docker Engine

- Docker Daemon + REST(ish) API
- Docker Client (CLI) talks to the API
- Daemon manages the Docker containers
- Start it with: docker –d
- docker version to test if docker is setup correctly
- Docker Engine (Linux, Mac, Windows)
  - https://docs.docker.com/engine/installation/

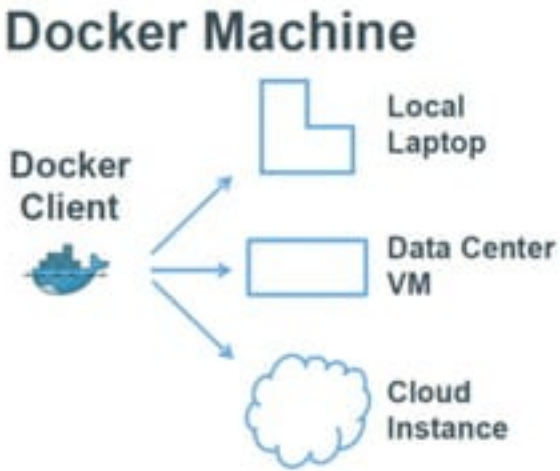


Docker Toolbox

- Docker Toolbox is an installer for quick setup and launch of a Docker environment on **older Mac and Windows systems** that do not meet the requirements of the new Docker for Mac and Docker for Windows apps.

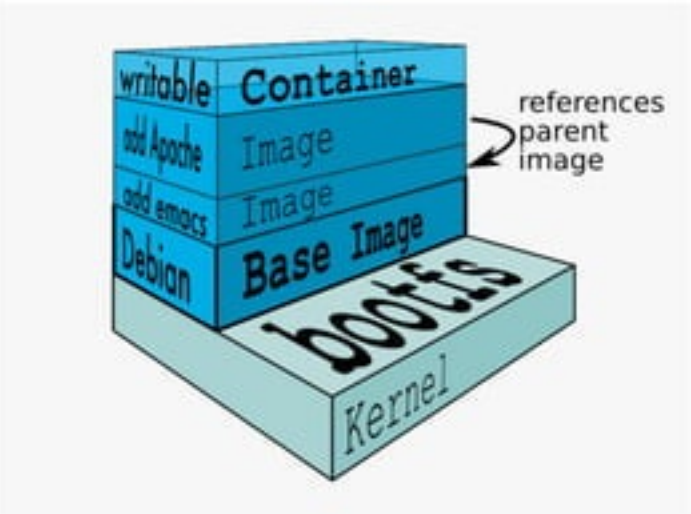
Docker Machine

- Machine 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.
- Drivers exist for:
  - AWS
  - DigitalOcean
  - Azure
  - Google Compute Engine
  - Rackspace
  - OpenStack
  - Virtualbox
  - VMWare Fusion
  - VMWare vSphere
  - Hyperv



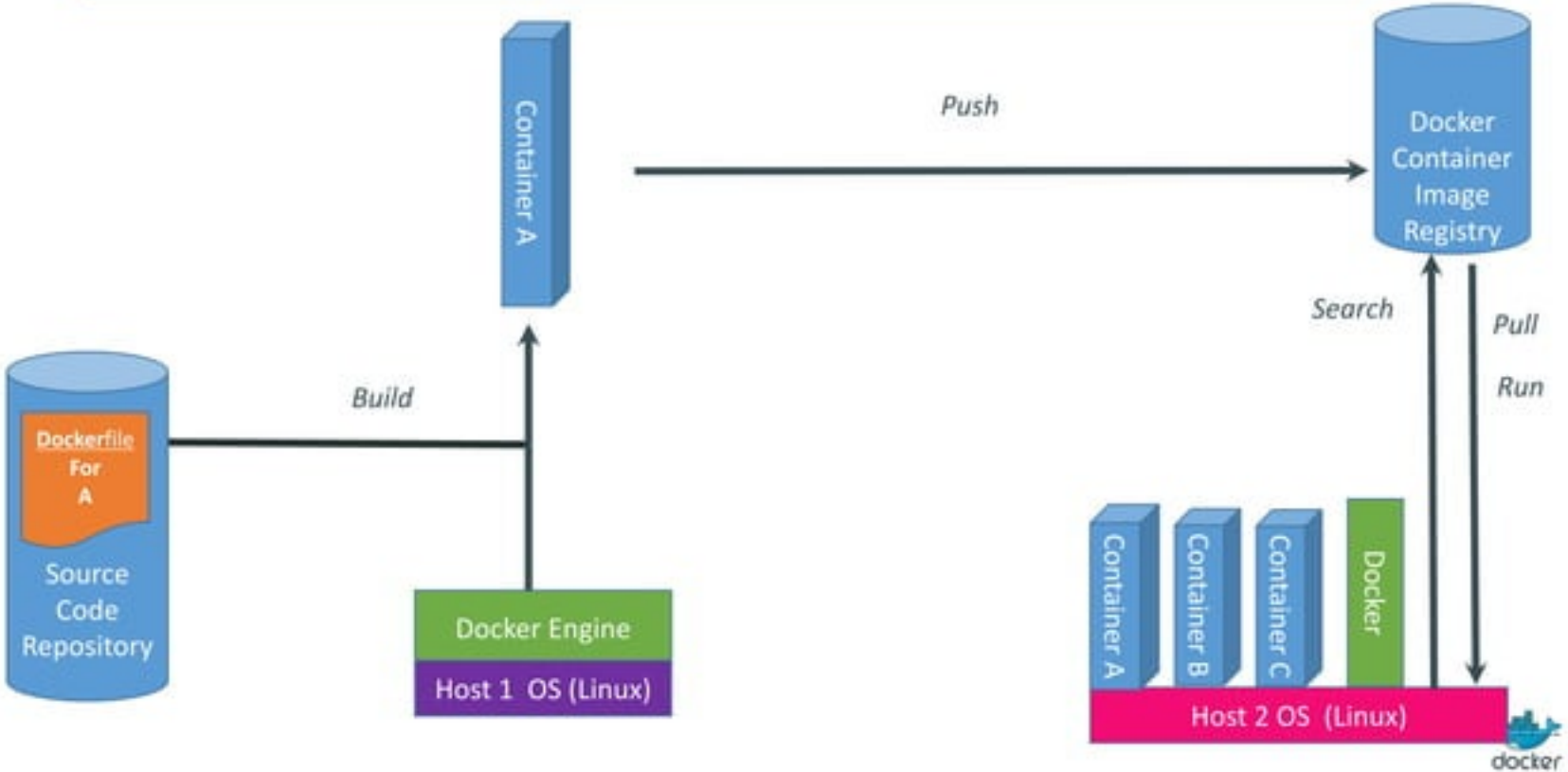
Docker Images vs Containers

- Images: About storing or moving your app
- Containers: About running your app



https://docs.docker.com/engine/userguide/storagedriver/imagesandcontainers/

Docker Workflow——Build/Pull/Push



Docker Image

- Download, list, remove images

```
$ docker pull ubuntu:14.04
14.04: Pulling from library/ubuntu

04cf3f0e25b6: Pull complete
d5b45e963ba0: Pull complete
a5c78fda4e14: Pull complete
193d4969ca79: Pull complete
d709551f9630: Pull complete
Digest: sha256:edb984703bd3e8981ff541a5b9297ca1b81fde6e6e8094d86e390a38ebc30b4d
Status: Downloaded newer image for ubuntu:14.04
```

```
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              14.04               aae2b63c4946        12 hours ago        187.9 MB
```

```
$ docker rmi aae2b63c4946
Untagged: ubuntu:14.04
Deleted: sha256:aae2b63c49461fcae4962e4a8043f66acf8e3af7e62f5ebceb70b181d8ca01e0
Deleted: sha256:50a2a0443efd0936b13eebb86f52b85551ad7883e093ba0b5bad14fec6ccf2ee
Deleted: sha256:9f0ca687b5937f9ac2c9675065b2daf1a6592e8a1e96bce9de46e94f70fbf418
Deleted: sha256:6e85e9fb34e94d299bb156252c89dfb4dcec65deca5e2471f7e8ba206eba8f8d
Deleted: sha256:cc4264e967e293d5cc16e5def86a0b3160b7a3d09e7a458f781326cd2cecedb1
Deleted: sha256:3181634137c4df95685d73bfbc029c47f6b37eb8a80e74f82e01cd746d0b4b66
```

https://hub.docker.com/

docker

Dockerfile

Create a Dockerfile

```
$ more Dockerfile
FROM        ubuntu:14.04
MAINTAINER  xiaoquwl@gmail.com
RUN         apt-get update && apt-get install -y redis-server
EXPOSE      6379
ENTRYPOINT  ["/usr/bin/redis-server"]
```

Build a image

```
$ docker build -t xiaopeng163/redis:0.1 .
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
xiaopeng163/redis   0.1                 ccbca61a8ed4        7 seconds ago       212.4 MB
ubuntu              14.04               3f755ca42730        2 days ago          187.9 MB
```

https://docs.docker.com/engine/reference/builder/

docker

Docker Container

- Create and start a container

```
$ docker run -d --name demo xiaopeng163/redis:0.1
4791db4ff0ef5a1ad9ff7c405bd7705d95779b2e9209967ffbef66cbaee80f3a
$ docker ps
CONTAINER ID        IMAGE                   COMMAND                  CREATED             STATUS              PORTS               NAMES
4791db4ff0ef        xiaopeng163/redis:0.1   "/usr/bin/redis-serve"   5 seconds ago       Up 4 seconds        6379/tcp            demo
```

- Inside of the container

```
$ docker exec demo ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 06:43 ?        00:00:00 /usr/bin/redis-server *:6379
root        10     0  0 06:59 ?        00:00:00 ps -ef
$ docker exec -it demo bash
root@4791db4ff0ef:/# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 06:43 ?        00:00:00 /usr/bin/redis-server *:6379
root        13     0  0 07:00 ?        00:00:00 bash
root        17     0  0 07:00 ?        00:00:00 bash
root        30    17  0 07:00 ?        00:00:00 ps -ef
```

Docker Container

- Stop and remove a container

```
$ docker ps
CONTAINER ID        IMAGE                   COMMAND                  CREATED             STATUS              PORTS               NAMES
4791db4ff0ef        xiaopeng163/redis:0.1   "/usr/bin/redis-serve"   20 minutes ago      Up 20 minutes       6379/tcp            demo
$ docker stop demo
demo
$ docker ps -a
CONTAINER ID        IMAGE                   COMMAND                  CREATED             STATUS                      PORTS               NAMES
4791db4ff0ef        xiaopeng163/redis:0.1   "/usr/bin/redis-serve"   20 minutes ago      Exited (0) 7 seconds ago                        demo
$ docker rm demo
demo
$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
```

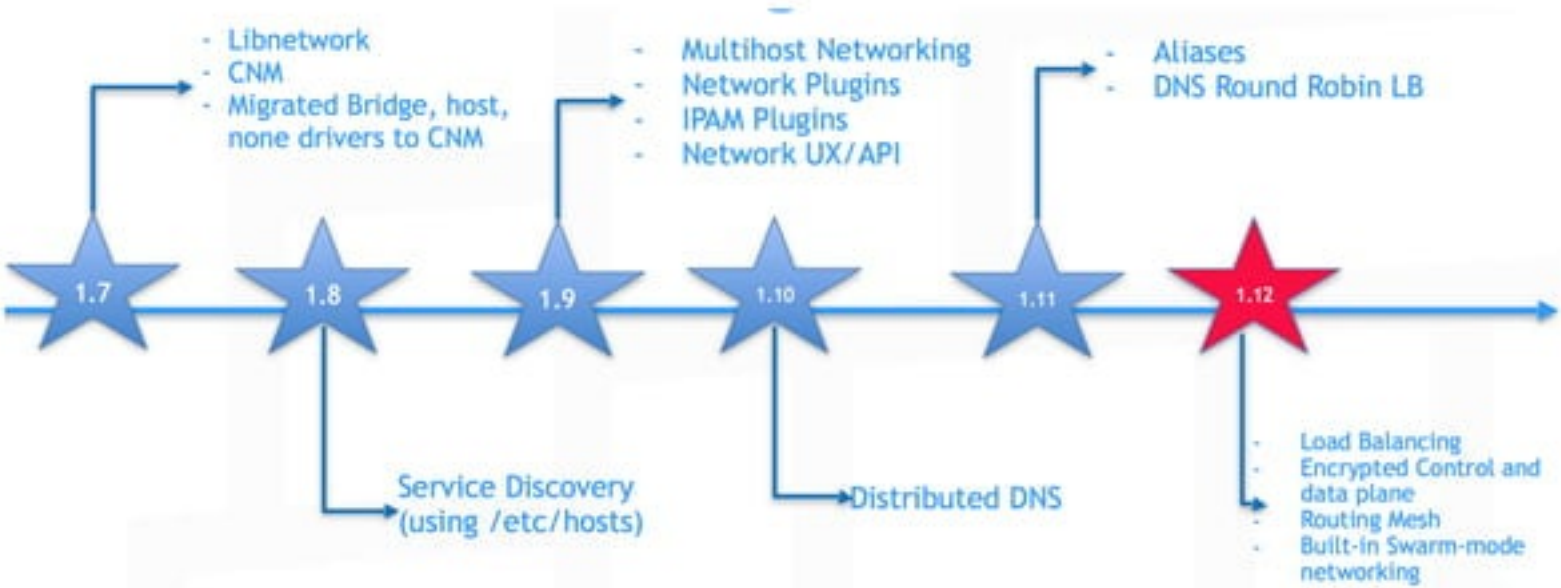
http://docker-k8s-lab.readthedocs.io/en/latest/docker/docker-cli.html

Contents

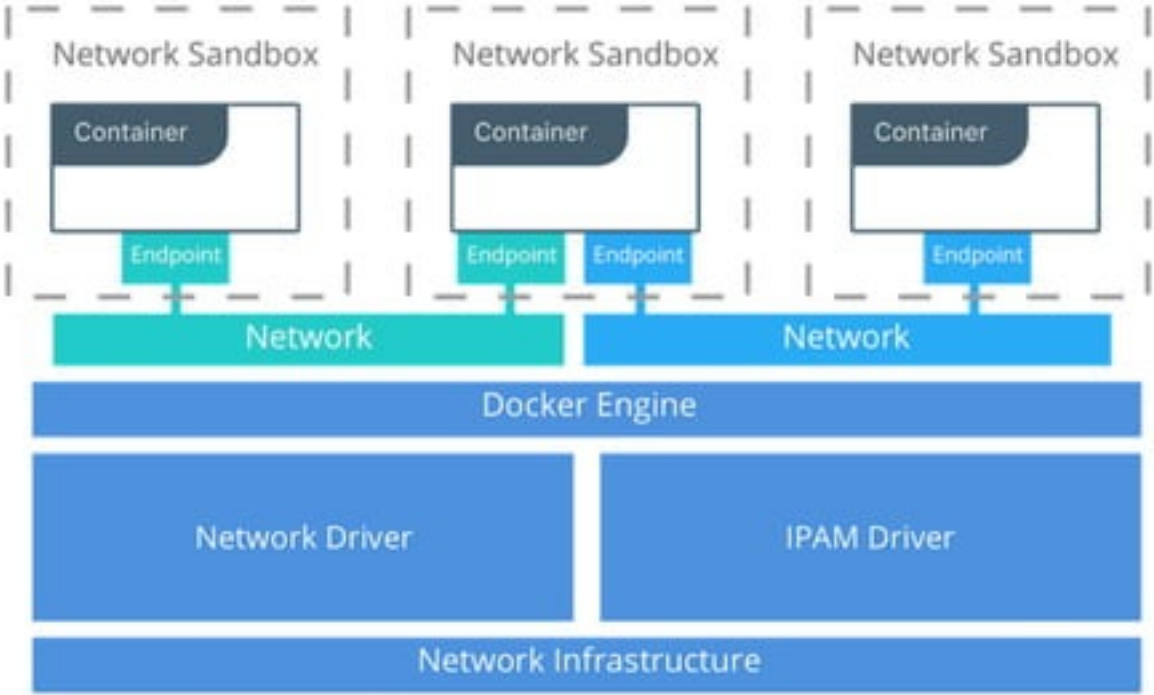
- The Metrix from Hell
- What is Container/Docker?
- Docker Basic Usage
- **Docker Networking Deep Dive**
- Docker Compose
- Docker Swarm
- Ecosystem and Standardization

docker

Docker Networking



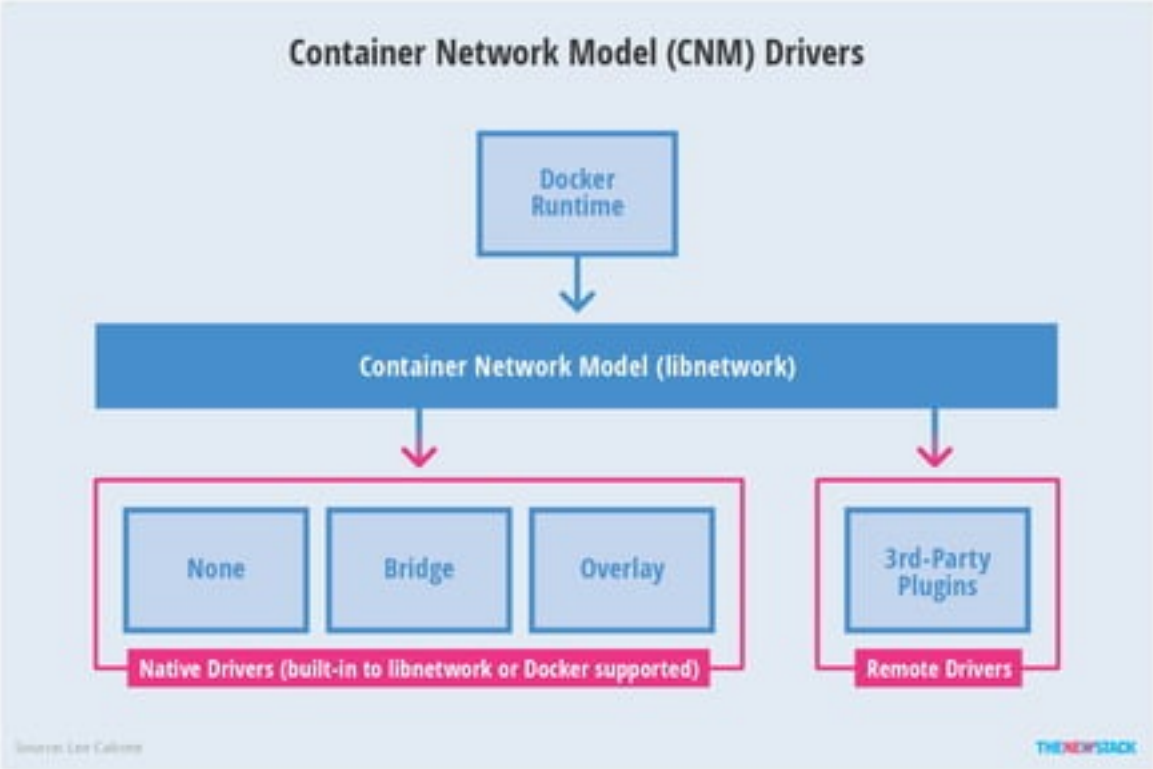
Container Network Model



https://github.com/docker/libnetwork/blob/master/docs/design.md

docker

CNM Drivers


Container Network Model (CNM) Drivers
Docker Runtime
Container Network Model (libnetwork)
None
Bridge
Overlay
3rd-Party Plugins
Native Drivers (built-in to libnetwork or Docker supported)
Remote Drivers

Demo: Docker Bridge Network

- http://docker-k8s-lab.readthedocs.io/en/latest/docker/bridged-network.html

Docker Multi-Host Networking

- Tunnel
  - Docker build-in overlay network: VXLAN
  - OVS: VXLAN or GRE
  - Flannel: VXLAN or UDP
  - Weave: VXLAN or UDP
- Routing
  - Calico: Layer 3 routing based on BGP
  - Contiv: Layer 3 routing based on BGP

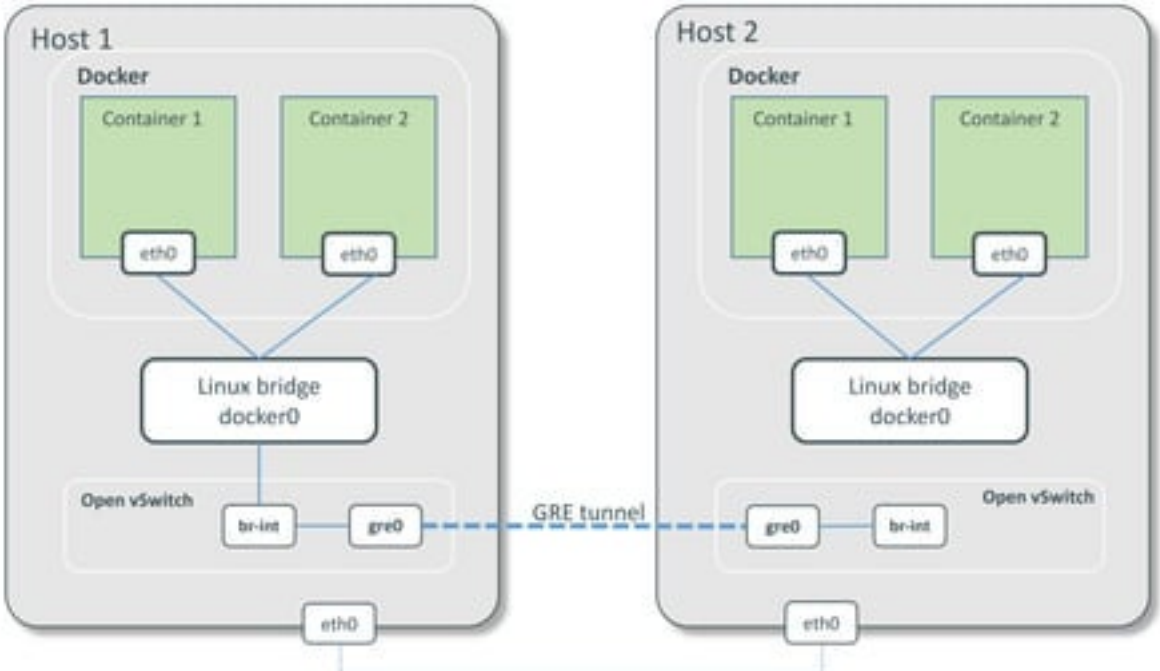
http://blog.dataman-inc.com/shurenyun-docker-133/

OvS Open vSwitch flannel Contiv

docker

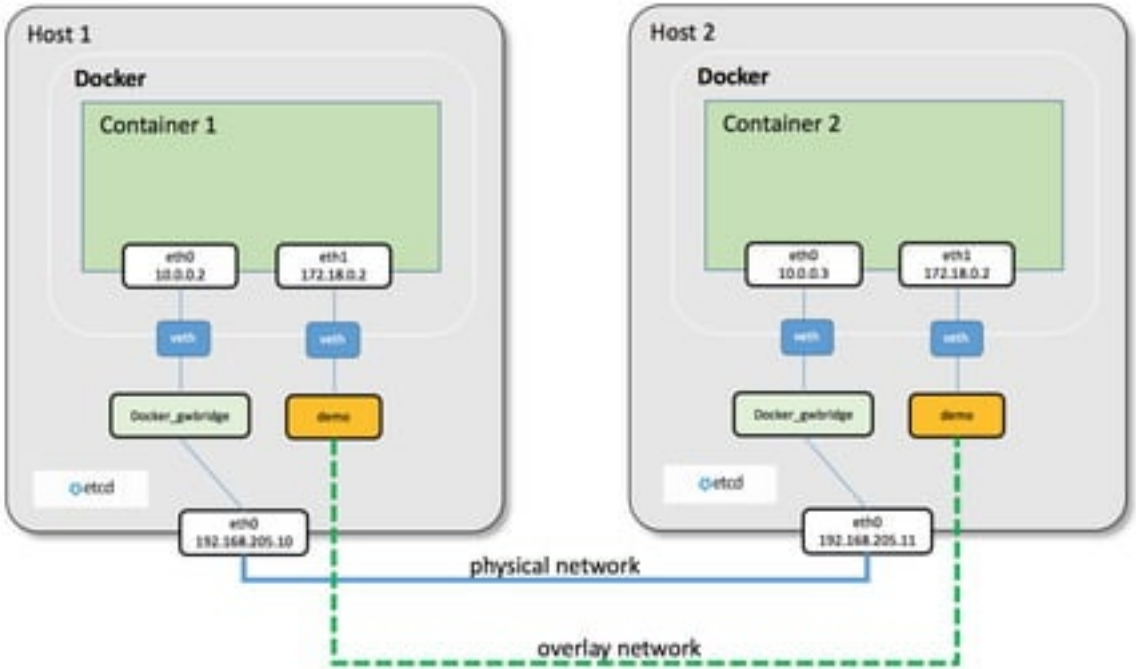
Lab A: Overlay Multi-Host Networking with OVS

- http://docker-k8s-lab.readthedocs.io/en/latest/docker/docker-ovs.html



Lab B:Multi-Host Overlay Networking with Etcd

- http://docker-k8s-lab.readthedocs.io/en/latest/docker/docker-etcd.html



Docker 1.12 Networking Model Overview

Features	Engine 1.11(and prior)	Engine 1.12
Multi-host Networking & KV Store	External KV store	No External KV Store required (Swarm Mode)
MACVLAN	Experimental	Out-of Experimental
Secure Control Plane	Insecure (Plain-text)	Secure
Secure Data Plane	VXLAN was not encrypted by default(can be secured by --opt-secure)	Encrypted VXLAN traffic(makes use of swarm certificates and key exchange)
Load Balancing	Load Balancing was featured under 1.10, based on only DNS RR	Virtual IP Load-Balancing & DNS RR both supported(LB using IPVS)
Service Discovery	Available under 1.10 but based on external service discovery backend	Service discovery now integrated into Docker Engine, Virtual IP for VIP Load-Balancing support.
Swarm Mode	Not Available	Newly Introduced (Optional Feature)
Routing Mesh	Not Available	Newly Introduced

docker

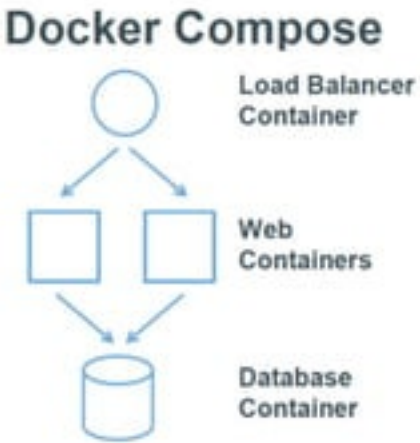
Contents

- The Metrix from Hell
- What is Container/Docker?
- Docker Basic Usage
- Docker Networking Deep Dive
- **Docker Compose**
- Docker Swarm
- Ecosystem and Standardization

Docker Compose Overview

- Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a Compose file to configure your application's services. Then, using a single command, you create and start all the services from your configuration.
- Defined in yaml

```yaml
web:
  build: .
  links:
   - db
  ports:
   - "8000:8000"
db:
  image: postgres
```



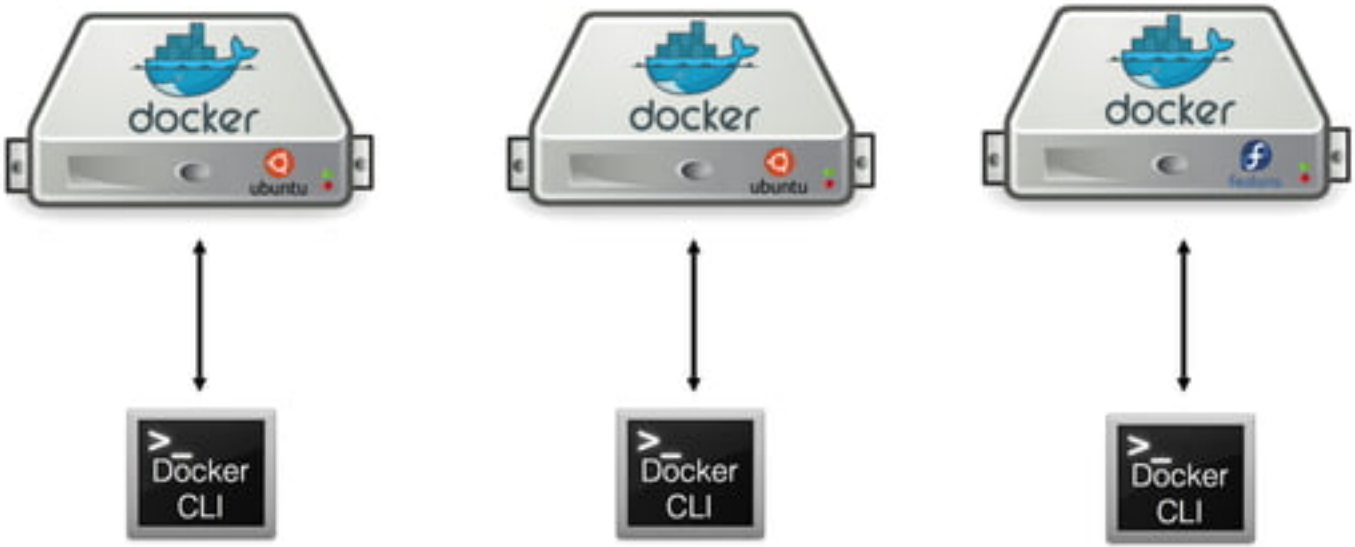
Demo: Docker Compose

- Example Voting App
  - https://github.com/DaoCloud/example-voting-app
- Guide:
  - http://docker-k8s-lab.readthedocs.io/en/latest/docker/docker-compose.html

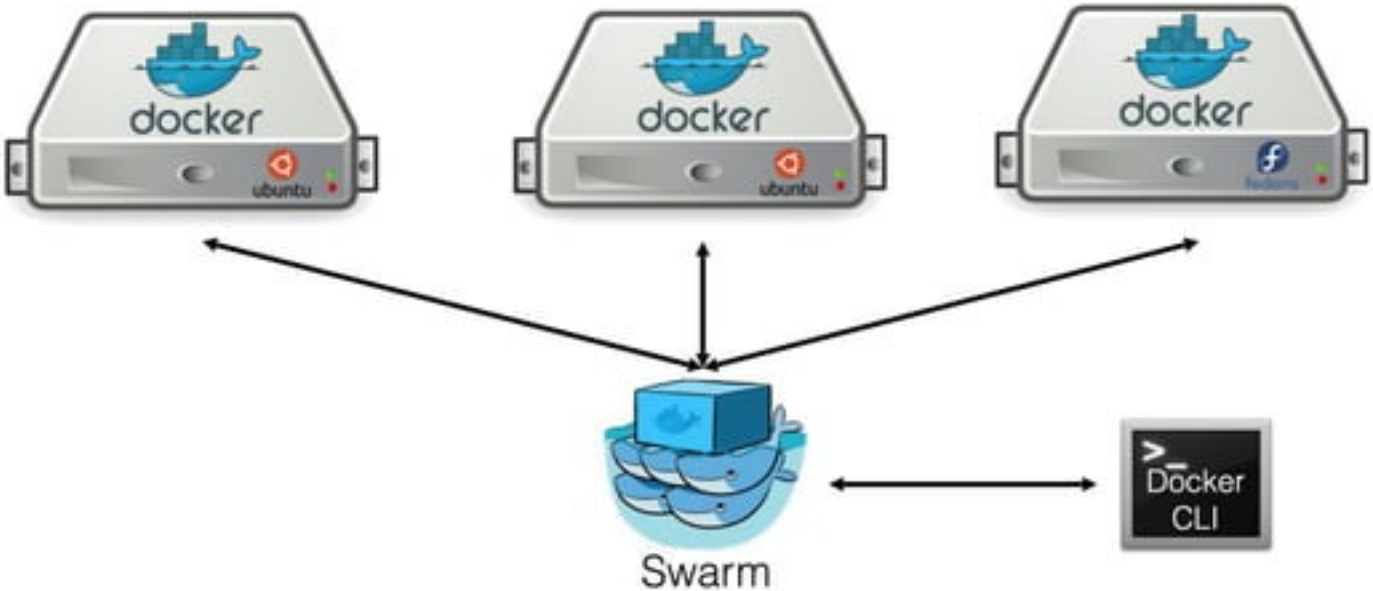
Contents

- The Metrix from Hell
- What is Container/Docker?
- Docker Basic Usage
- Docker Networking Deep Dive
- Docker Compose
- **Docker Swarm**
- Ecosystem and Standardization

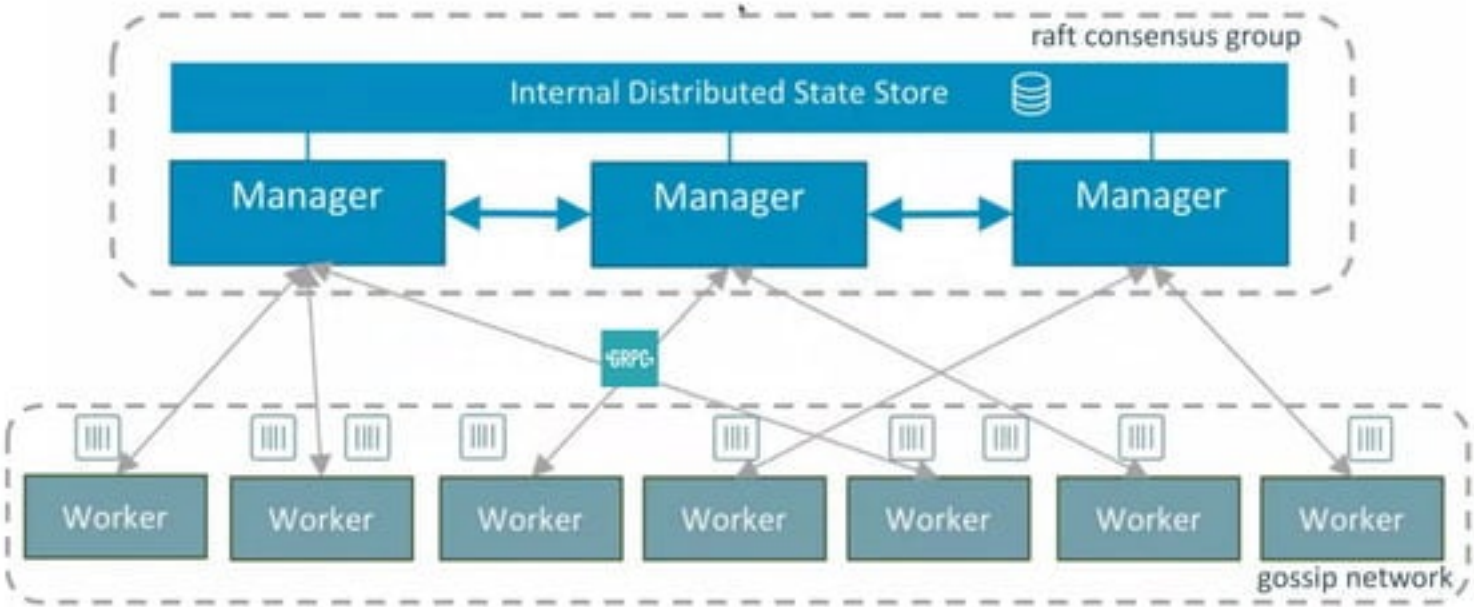
Before Docker Swarm



With Docker Swarm



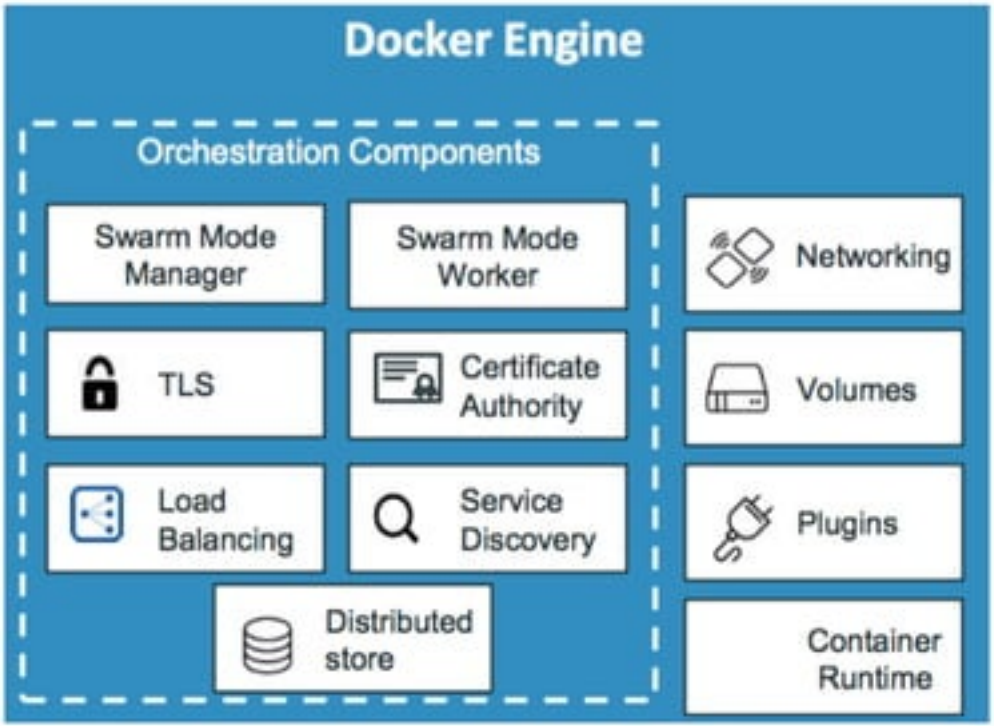
Swarm mode cluster architecture



http://www.grpc.io/
http://thesecretlivesofdata.com/raft/
https://en.wikipedia.org/wiki/Gossip_protocol

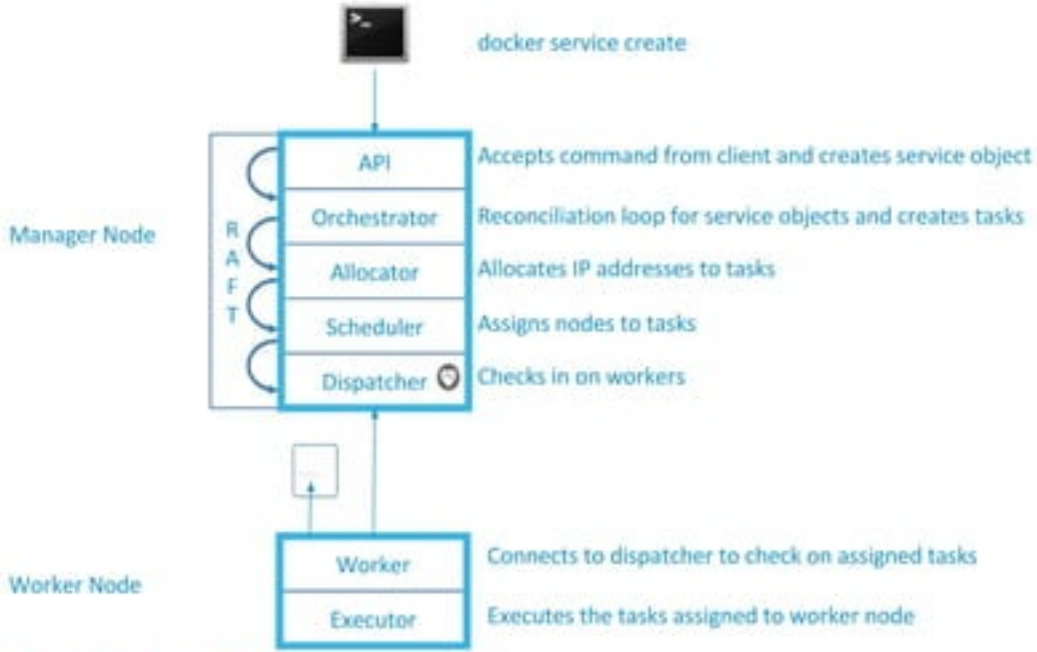
docker

Docker Orchestration Components



http://www.slideshare.net/ajeetraina/service-discovery-loadbalancing-under-docker-1120-docker-meetup-22

Create Docker Service



http://collabnix.com/archives/1445

https://www.youtube.com/watch?v=_F6PSP-qhdA

Demo: Docker Swarm

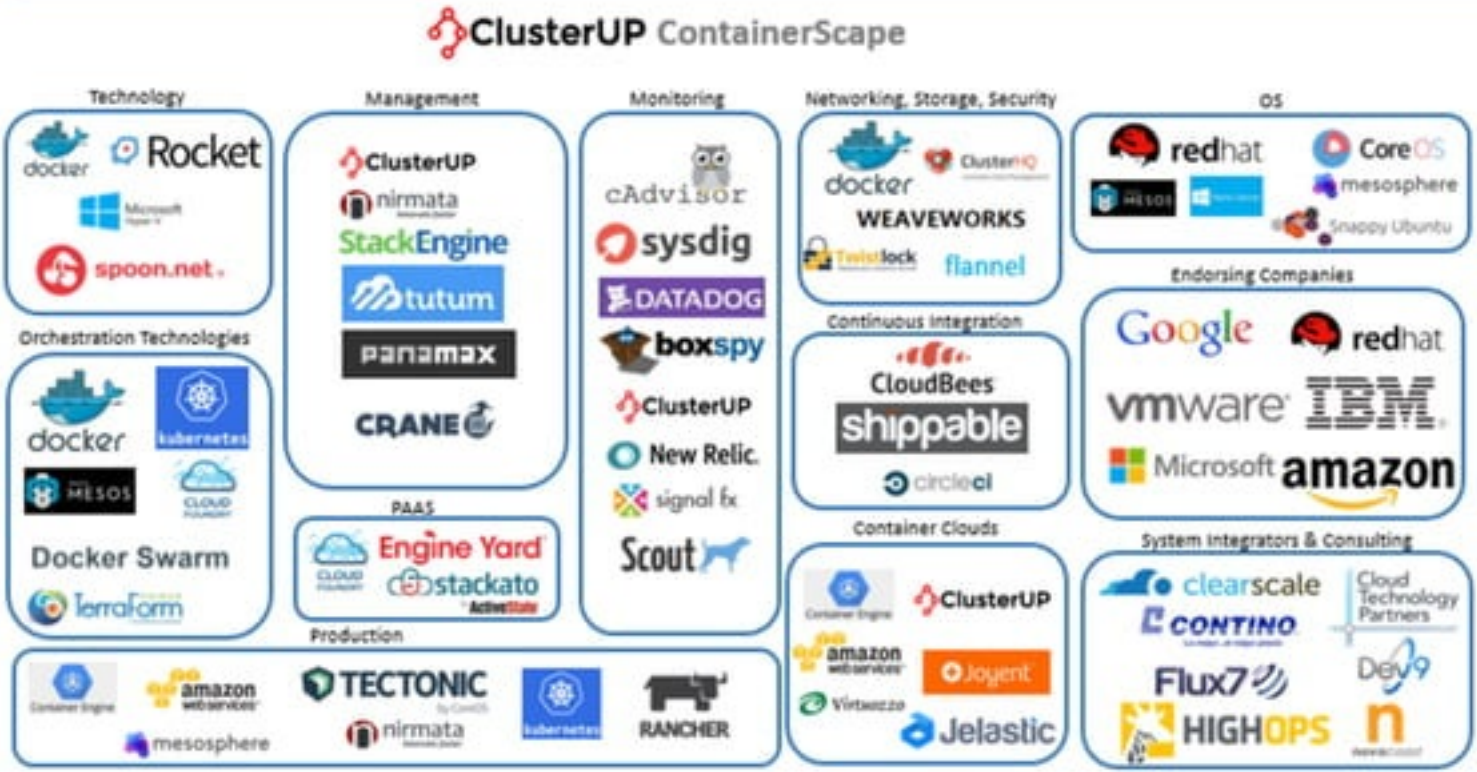
- Docker Swarm with Load Balancing and Scaling
  - http://docker-k8s-lab.readthedocs.io/en/latest/docker/docker-swarm-lb-scale.html
- Multi-Services issue
  - https://github.com/docker/compose/issues/3656

docker

Contents

- The Metrix from Hell
- What is Container/Docker?
- Docker Basic Usage
- Docker Networking Deep Dive
- Docker Compose
- Docker Swarm
- **Ecosystem and Standardization**

Container Ecosystem



Open Container Initiative (OCI)

- A **Linux Foundation** Collaborative Project
- Free from control by any particular vendor's specfic cloud stack or ecosystem
- Includes:
  - Container runtime specification -> runc
  - Image format specification

http://www.slideshare.net/PhilEstes/runc-the-little-engine-that-could-run-docker-containers
https://www.opencontainers.org/
https://github.com/opencontainers

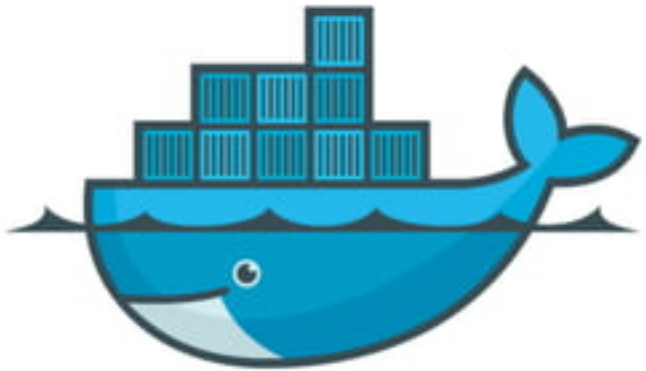
docker

Container Networking

- Container Network Model (CNM) by Docker
  - Adopted by Docker libnetwork, Cisco Contiv, Kuryr, OVN, Project Calico, Vmware, Vwave
- Container Network Interface(CNI) by CoreOS
  - Adopted by kuberntes, Kurma, rkt, Apache Mesos, Clould Foundry, Cisco Contiv, Project Calico and Weave.

https://github.com/containernetworking
http://thenewstack.io/container-networking-landscape-cni-coreos-cnm-docker/

docker



docker

?