

The Cloud & Container Professionals



Docker & Cloud Native Training

Training Agenda



Introduction



Docker Overview



Docker Labs

How can you Contribute?



Use the 56K Slack Channel Twitter



Feedback



Submit PR's!



Follow-up Video Conf Calls



Brian Christner

Brian@56K.Cloud / [@idomyowntricks](https://twitter.com/idomyowntricks)

Cloud Architect & Cloud Native

- Cloud Architect
- Background in Containers, Cloud, & Engineering
- Docker Captain

Common IT Struggles

Cloud Migration
and
Transformation

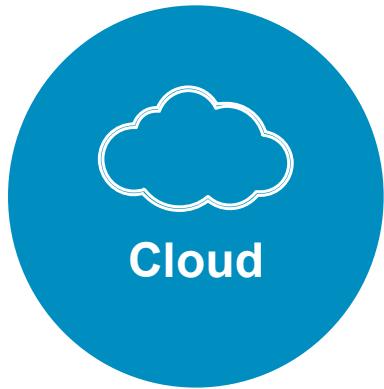
Manual
Deployment &
Operations

Legacy
Applications

Runaway Cloud
Costs

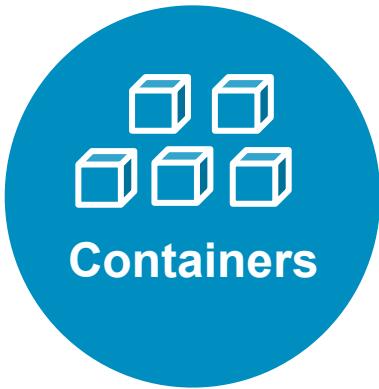


56K.Cloud Services



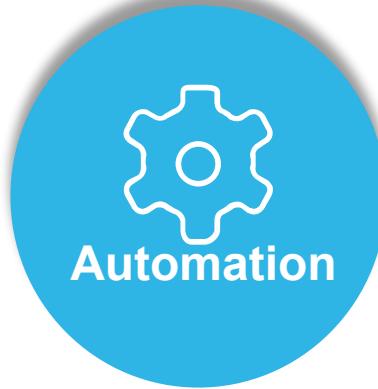
Cloud

Cloud Migration
Hybrid Cloud
Multi-Cloud



Containers

Containerization
Microservices
App Modernization



Automation

DevOps
CI/CD
Infrastructure as Code

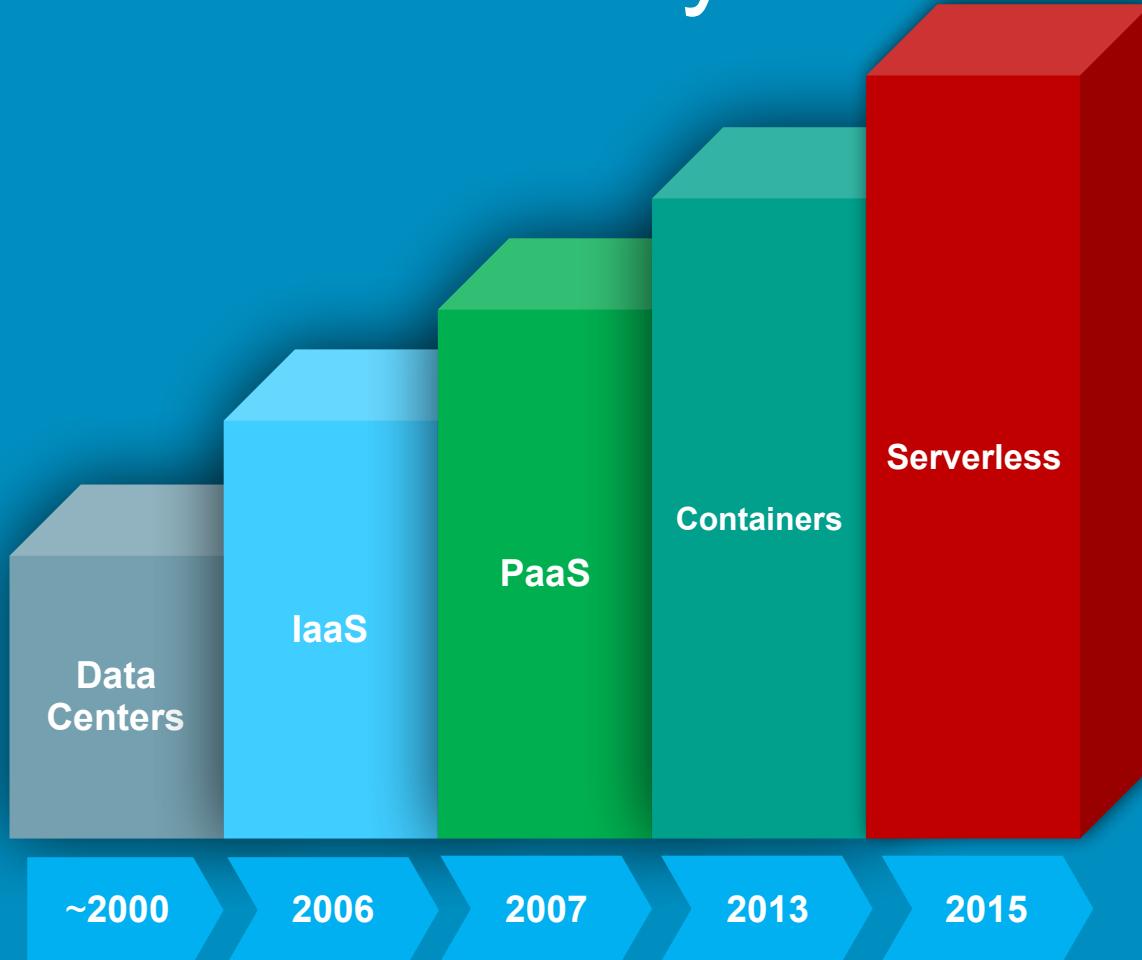


Managed Services

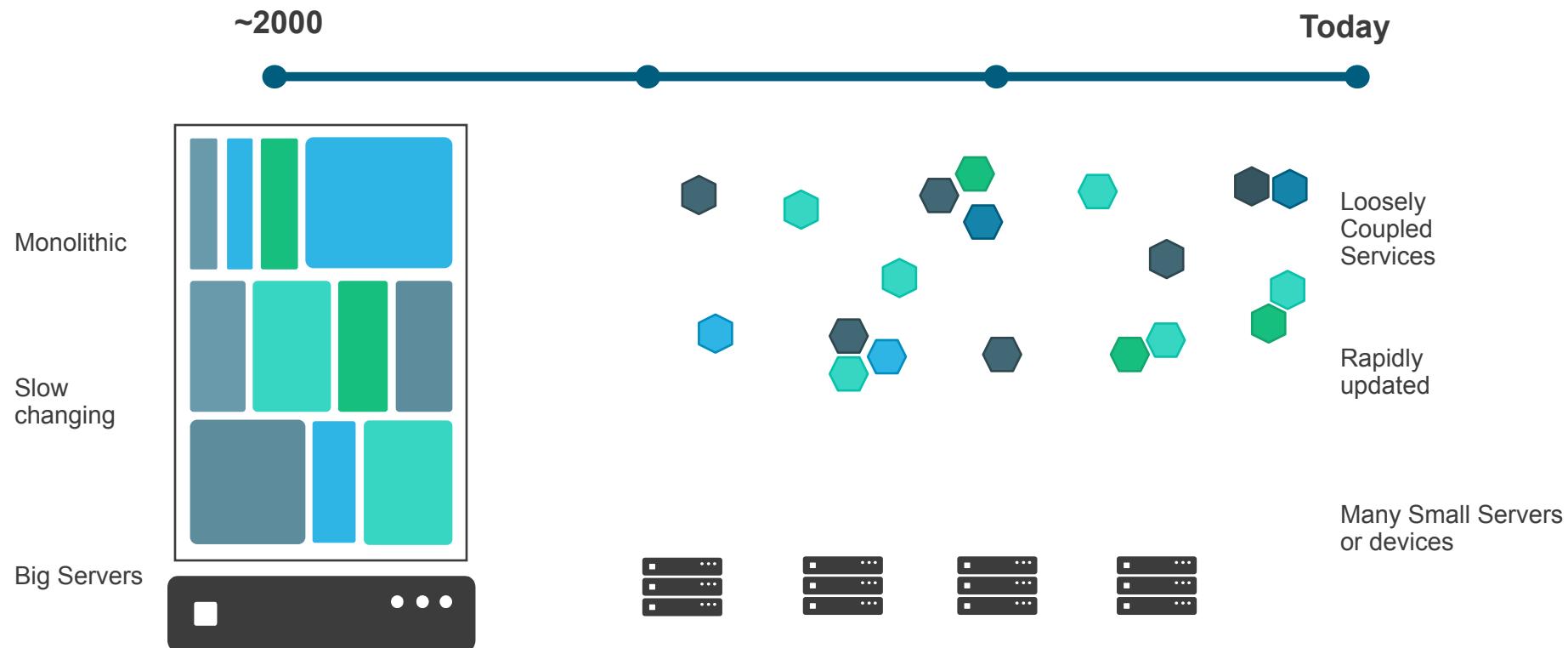
Reference Architecture
Code Library

Docker Overview

A Brief History Lesson



The application landscape is changing



THE EVOLUTION OF
SOFTWARE ARCHITECTURE

1990's

SPAGHETTI-ORIENTED
ARCHITECTURE
(aka Copy & Paste)



Cloud Native

What is Cloud Native

Cloud native computing uses an open source software stack to be:



Containerized

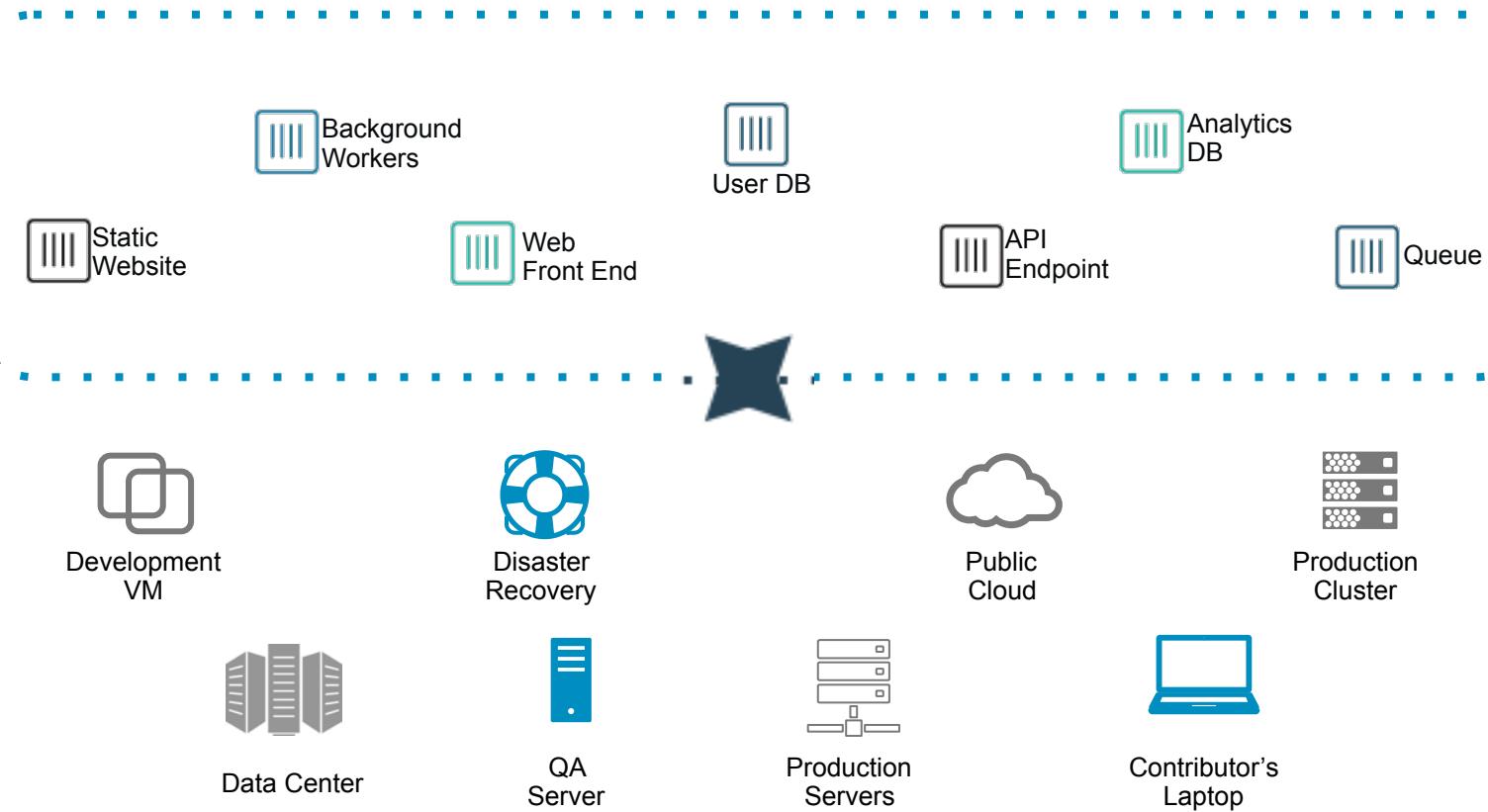


Dynamically orchestrated



Microservices oriented

Containers are the catalyst



Cloud Native Landscape

v7.1

Database & Data Analytics

Streaming

SCM

Application Definition

CI/CD

App Definition & Development

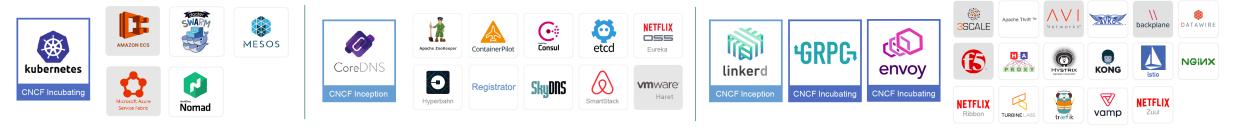


Scheduling & Orchestration

Coordination & Service Discovery

Service Management

Orchestration & Management



Cloud-Native Storage

Container Runtime

Cloud-Native Network

Runtime



Host Management / Tooling

Infrastructure Automation

Container Registries

Secure Images

Key Management

Provisioning



Public

Private

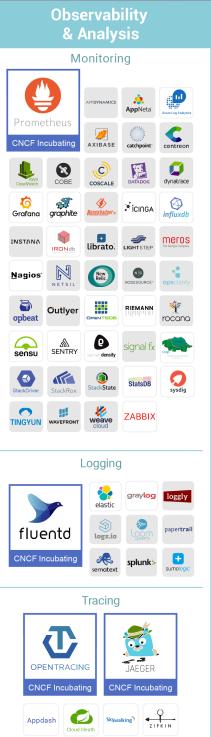
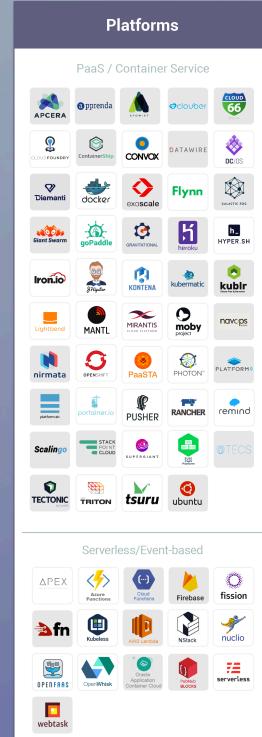
Cloud



This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.

CLOUD NATIVE COMPUTING FOUNDATION

Redpoint Amplify PARTNERS



Greyed logos
are not
open source

CNCF Map



CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape cncf.io has a large number of options. This Cloud Native Trail Map is a recommended process for leveraging open source, cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

HELP ALONG THE WAY

A. Training and Certification

Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer cncf.io/training

B. Consulting Help

If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider

cncf.io/kcsps



C. Join CNCF's End User Community

For companies that don't offer cloud native services externally

cncf.io/enduser

WHAT IS CLOUD NATIVE?

Cloud-native technologies, such as containers and microservices, empower organizations to develop and deploy scalable, agile applications and services in dynamic, distributed environments. By taking into account these characteristics, such systems are designed to be resilient, elastic, and loosely coupled, via manageable abstractions and declarative APIs, thereby enabling effective, reliable automation. This allows engineers to observe the applications and to safely make impactful changes, and results in processes and workflows that fully take advantage of these environments and minimize toil.

The Cloud Native Computing Foundation seeks to drive adoption of these techniques by fostering an ecosystem of open-source, vendor-neutral projects that align with these objectives, and which are portable to public, private, and hybrid clouds. We democratize the state-of-the-art patterns and practices to ensure innovations remain open and accessible for everyone.

l.cncf.io



v20180604

5. SERVICE MESH AND DISCOVERY

- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing



CNCF Incubating



CNCF Incubating



CNCF Incubating



7. DISTRIBUTED DATABASE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding.



CNCF Incubating



9. CONTAINER RUNTIME

You can use alternative container runtimes. The most common, all of which are OCI-compliant, are containerd, rkt and CRI-O.



CNCF Incubating



CNCF Incubating

6. NETWORKING

To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net.



CNCF Incubating



8. MESSAGING

When you need higher performance than JSON-REST, consider using gRPC. NATS is publish/subscribe message-oriented middleware.



CNCF Incubating



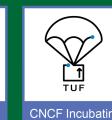
CNCF Incubating

10. SOFTWARE DISTRIBUTION

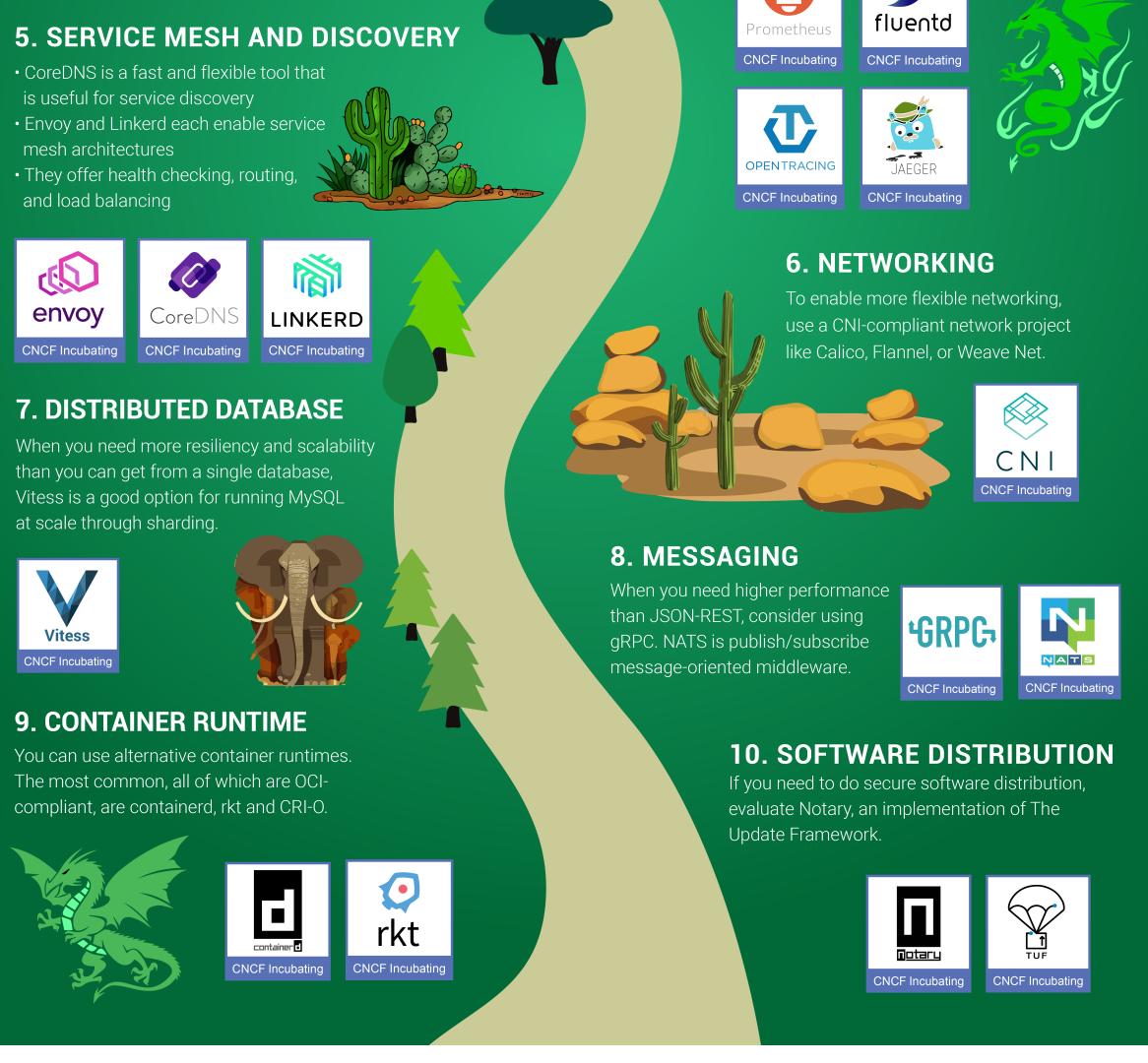
If you need to do secure software distribution, evaluate Notary, an implementation of The Update Framework.



CNCF Incubating



CNCF Incubating



Build, Ship, Run, Any App Anywhere

From Dev



To Ops



Any App



Any OS



Windows



Linux

Anywhere



Physical



Virtual



Cloud





AWESOME SAUCE!

6,000,000,000

5,750,000,000

5,500,000,000

5,250,000,000

5,000,000,000

4,750,000,000

4,500,000,000

4,250,000,000

4,000,000,000

3,750,000,000

3,500,000,000

3,250,000,000

3,000,000,000

2,750,000,000

2,500,000,000

2,250,000,000

2,000,000,000

1,750,000,000

1,500,000,000

1,250,000,000

1,000,000,000

~

10,000,000

9,000,000

8,000,000

7,000,000

6,000,000

5,000,000

4,000,000

3,000,000

2,000,000

1,000,000

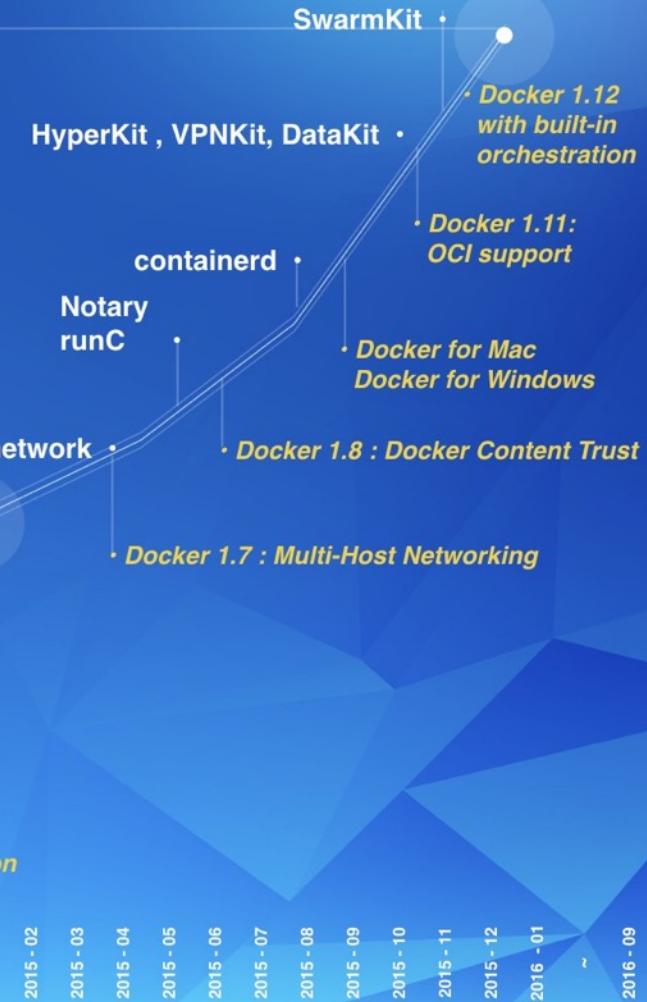
0



2019
2 Billion++
every Week

libcontainer

Docker 0.9 : Pluggable execution



Docker Momentum



450+

Docker EE
commercial
customers



37B

Container
downloads



15K

Job listings on
LinkedIn



3.5M

Dockerized
apps



200+

Active Docker
user groups

Observability @ Docker



1500

Containers

70

Microservices

985K

Time-series
metrics in
Prometheus

35K

Logs per
second

1TB

Total logs per
day

Docker delivers innovation, speed and savings



Agility

13X

More software releases

65%

Reduction in developer onboarding time



Portability

41%

Move workloads across private/
public clouds

Eliminate

“works on my machine”
issues



Control

62%

Report reduction in MTTR

10X

Cost reduction in maintaining
existing applications

Who uses



swisscom

die Mobiliar



VR

NETFLIX

One journey for all applications

1

Containerize Legacy Applications

Lift and shift for portability and efficiency



2

Transform Legacy to Microservices

Look for shared services to transform

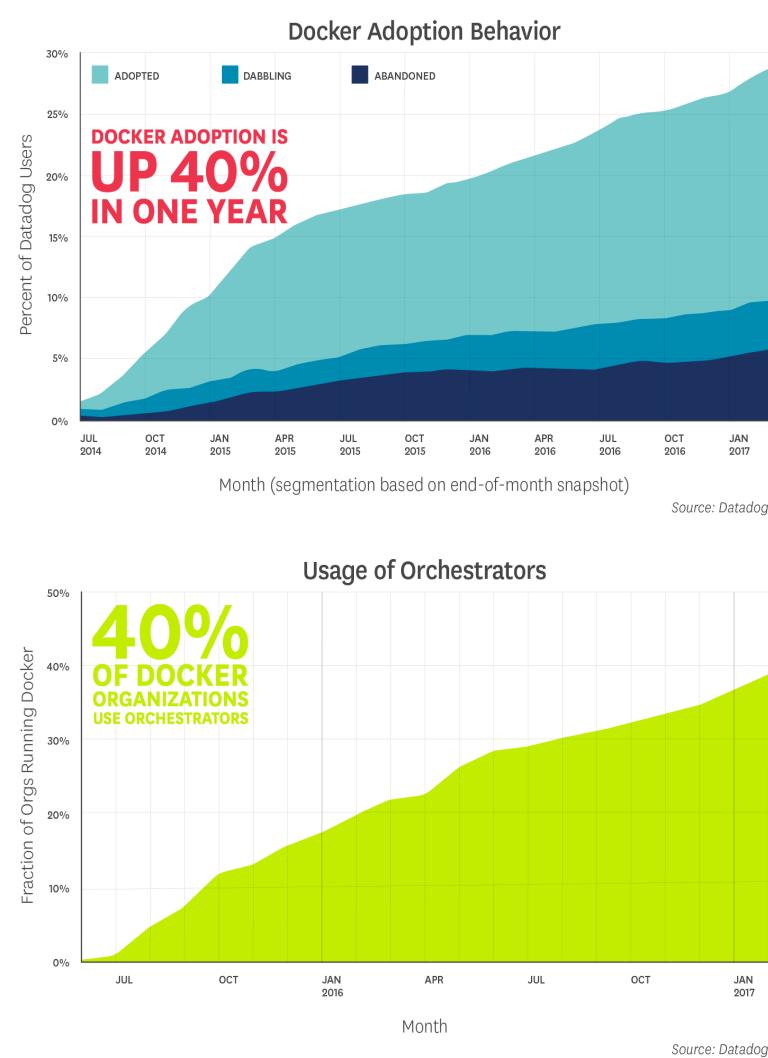


3

Accelerate New Applications

Greenfield innovation





Enterprise Adoption

- Docker Container Adoption
- Orchestration Usage
- Adoption rate nearly quintuple the number of running containers in production between their first and tenth month of usage
- <https://www.datadoghq.com/docker-adoption/>

Containers vs VM's



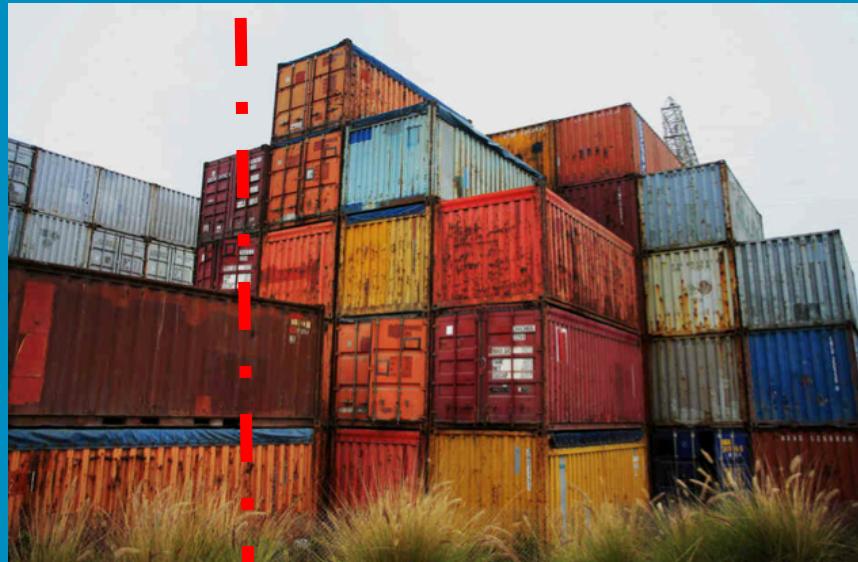
VM's



Containers

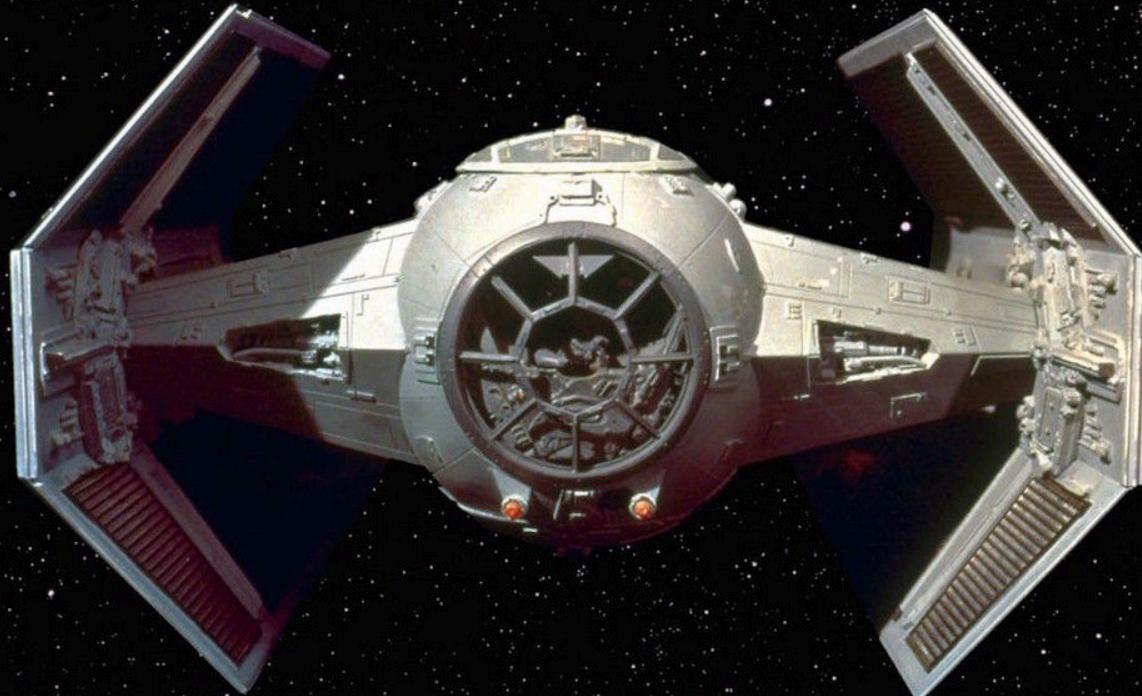
Long vs Short Lived Apps/Containers

Long Lived



Short Lived





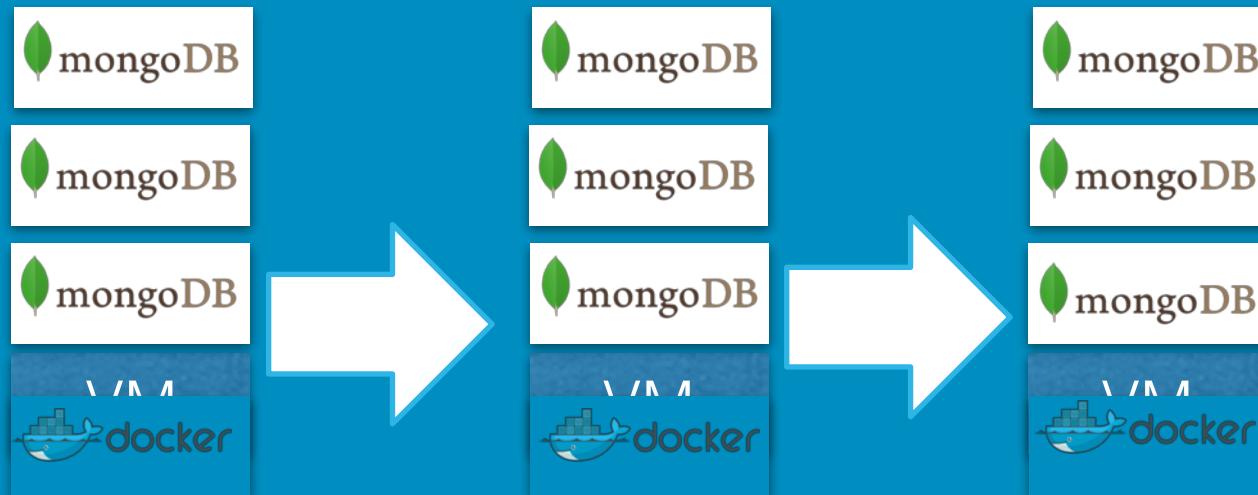
VM's



Containers



docker Use Case



DOCKER LABS

Training Agenda



Docker CLI Overview



Webapps and Docker



Write Dockerfiles



Create a Docker Build pipeline



Deploy apps to Docker Swarm



Docker Logging



Docker Monitoring

Docker Time



Let's Get Started!

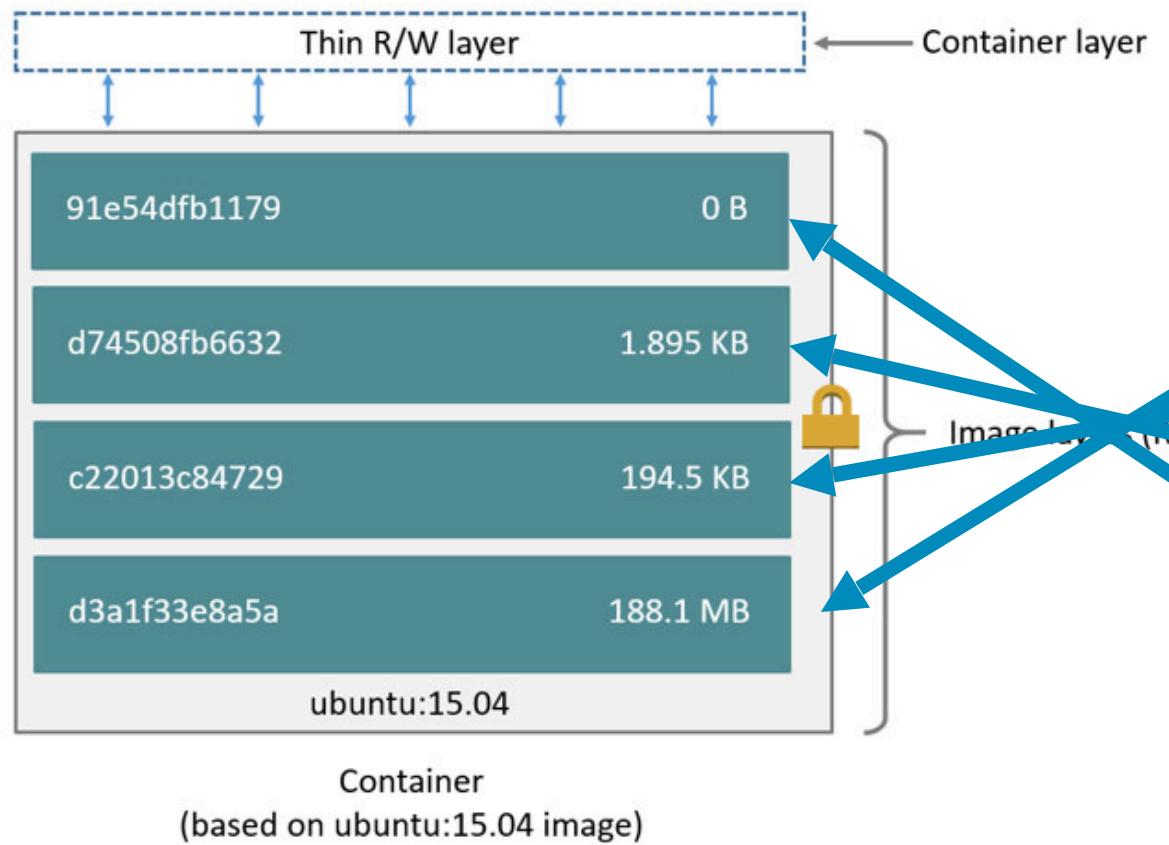
github.com/56kcloud/training

Let's Get Started!

<https://dockr.ly/bc>

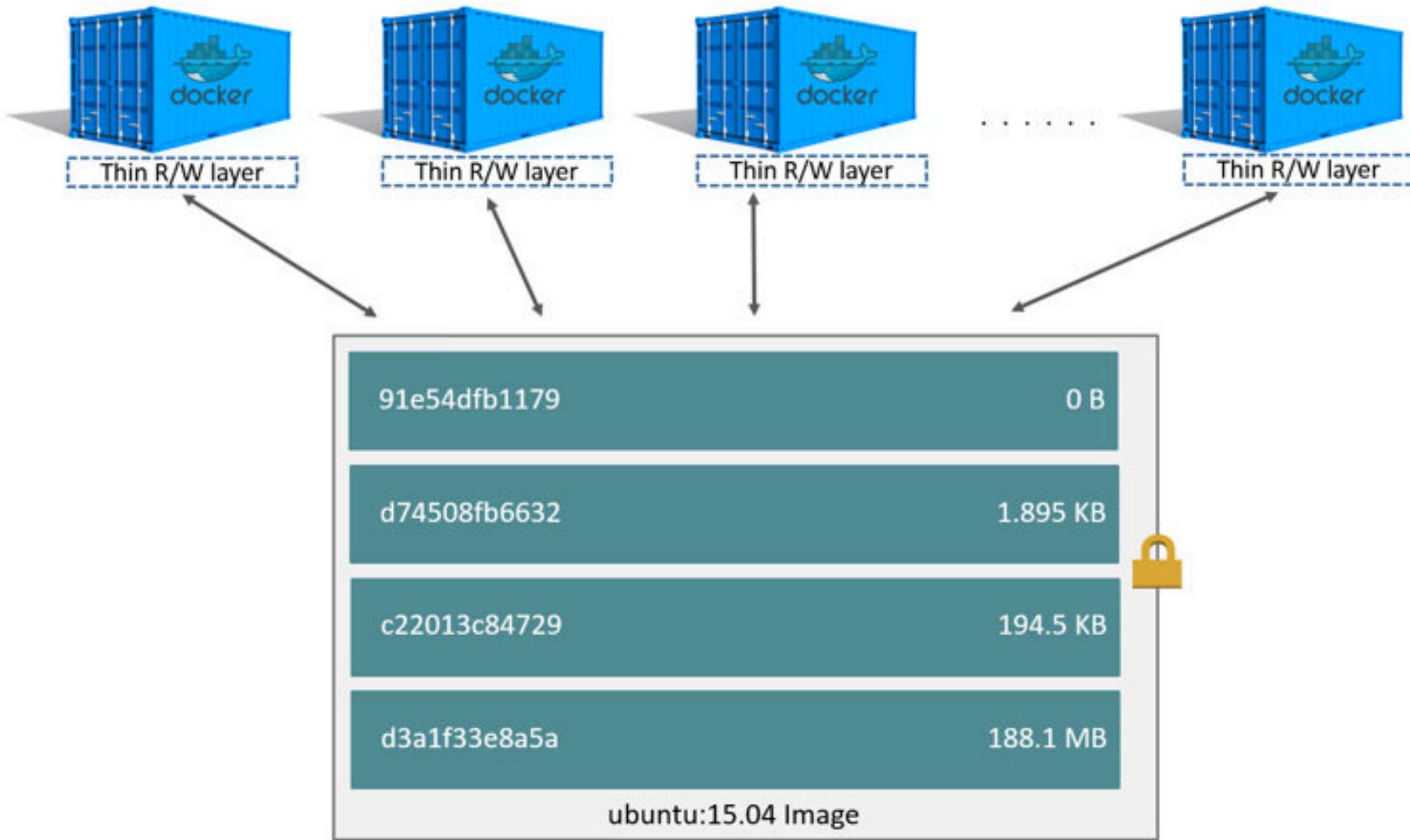
Docker Images

Images & Layers



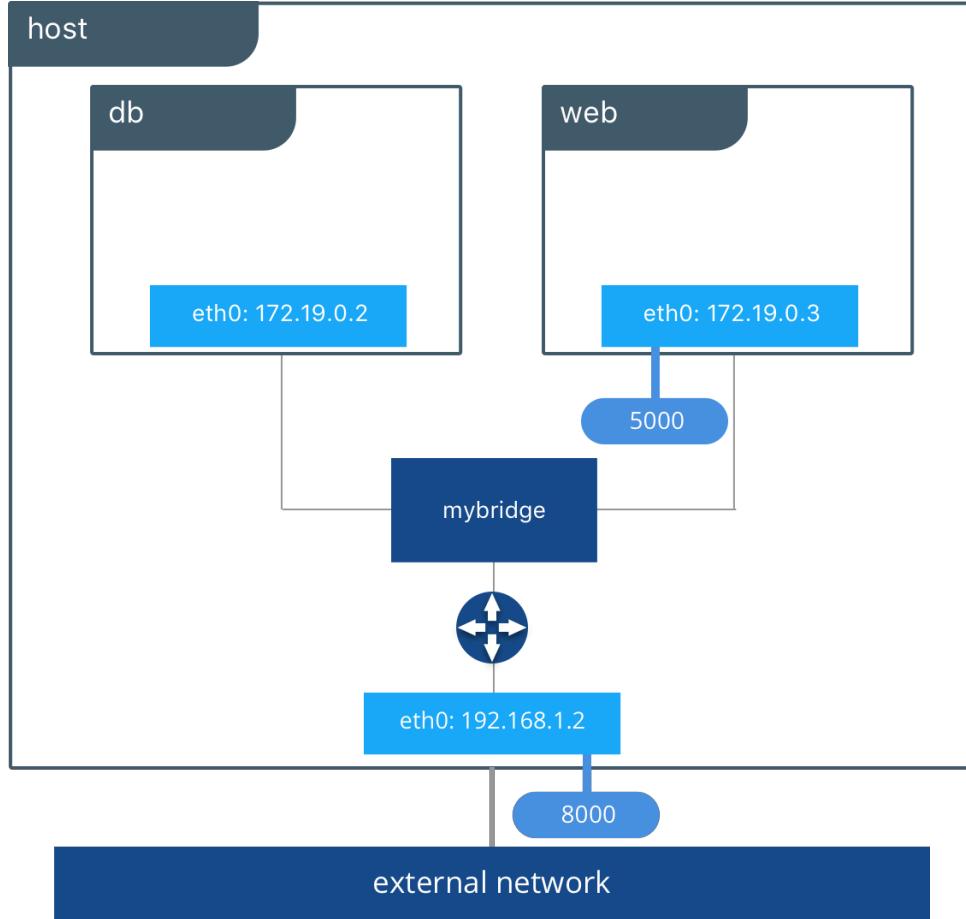
Dockerfile:

```
FROM ubuntu:15.04
COPY . /app
RUN make /app
CMD python /app/app.py
```

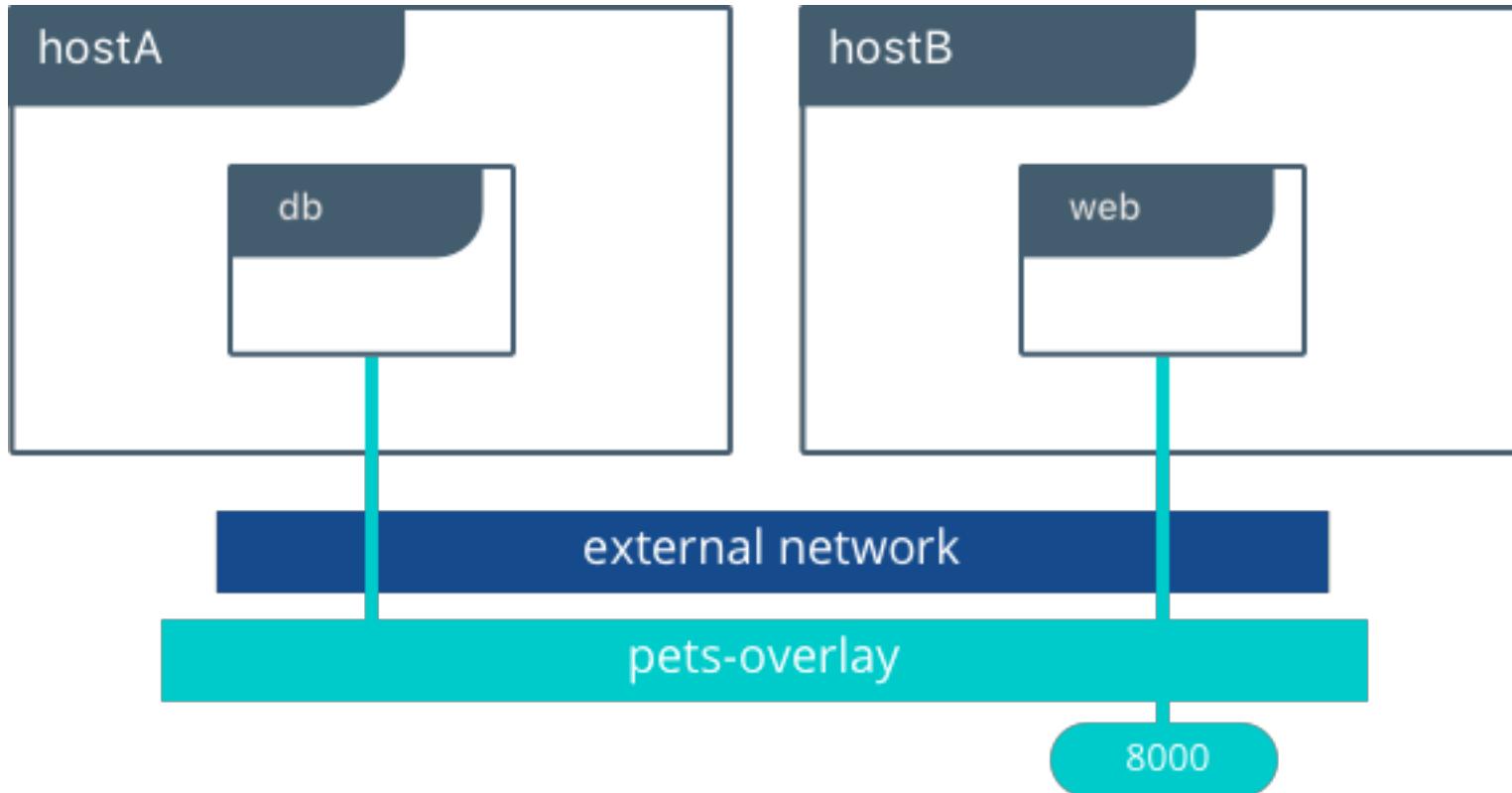


Docker Networking

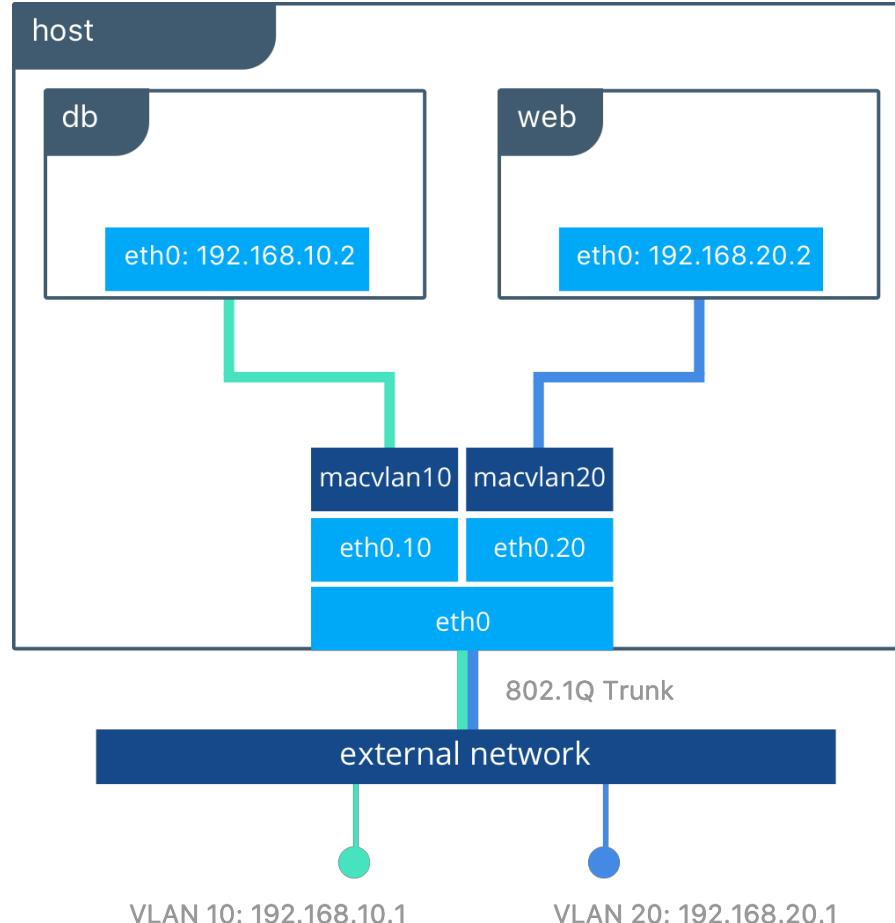
Bridged Network



Overlay Network

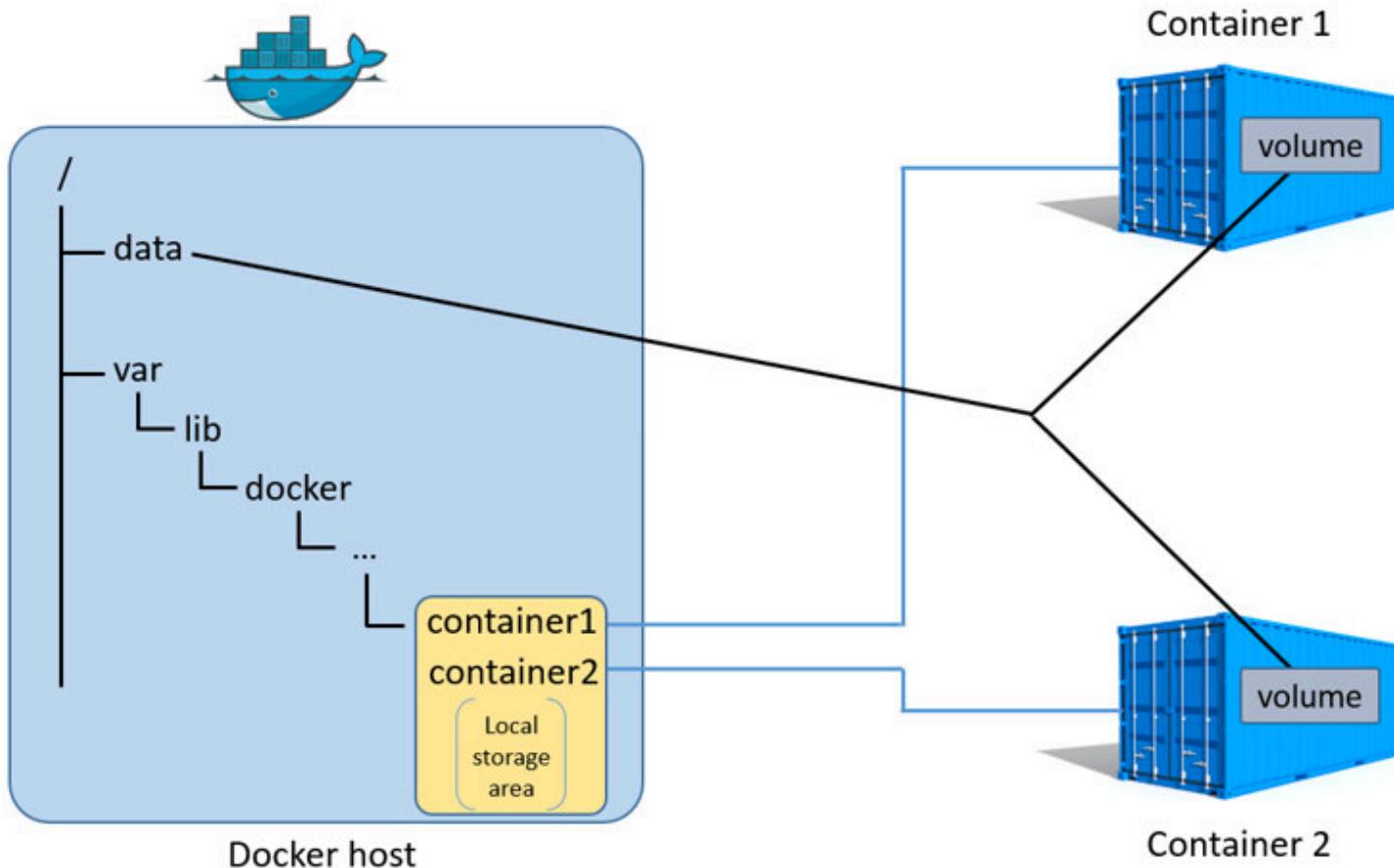


MACVLAN Network

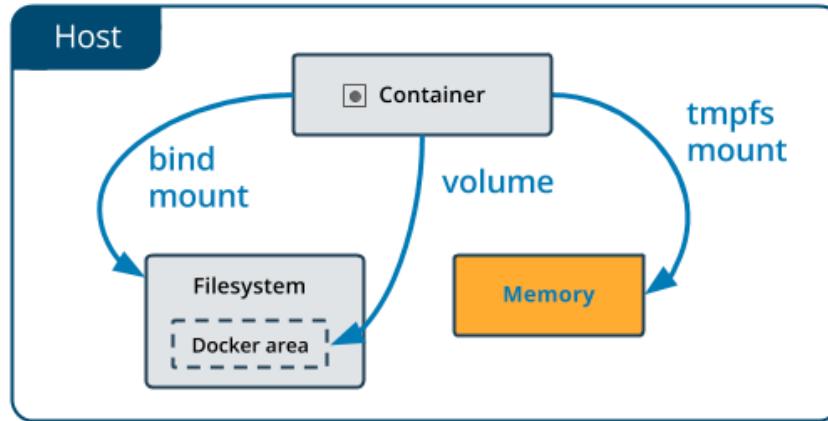
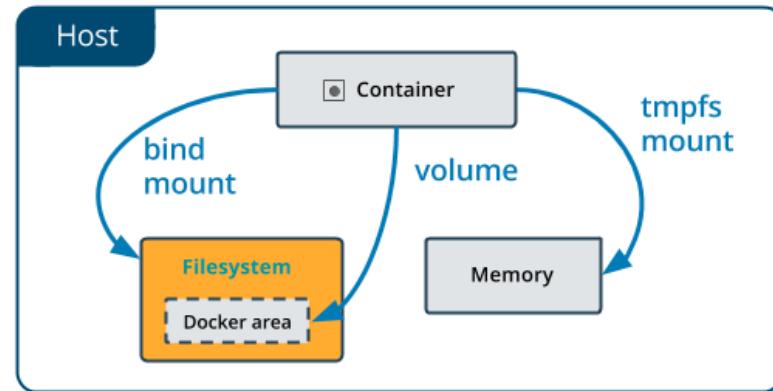
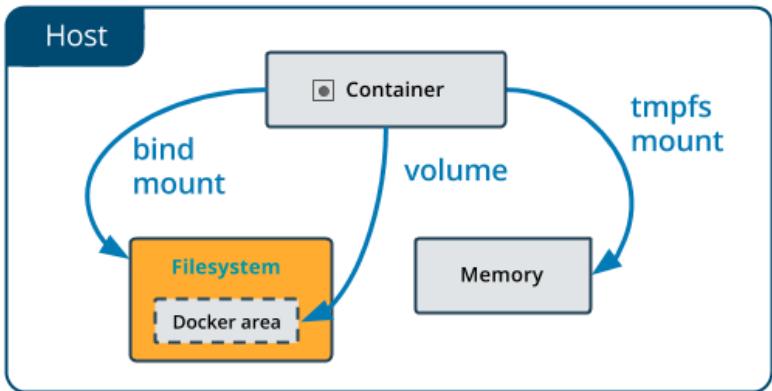


Docker Volumes

Anonymous vs Named Volumes



Bind vs Volume vs tmpfs



DevOps and Docker

DevOps tools

These tools help you manage servers and deploy happier and more often with more confidence.

22 repositories

7 languages

Last updated on May 1

Stars

Language

Search showcases



[moby / moby](#)

Moby Project - a collaborative project for the container ecosystem to assemble container-based systems

Go

45,421

13,500

Updated 43 minutes ago

Starred by 12 people you know



[kubernetes / kubernetes](#)

System for managing containerized applications across multiple hosts, providing basic mechanisms for deployment, maintenance, and scaling of applications.

Go

26,777

9,566

Updated 11 minutes ago

Starred by 10 people you know



Related showcases

Great for new contributors

These projects have a history and reputation for being welcoming to new ...

Projects that power GitHub

Here are some of the great open source projects that GitHub is using to ...

[View all](#)



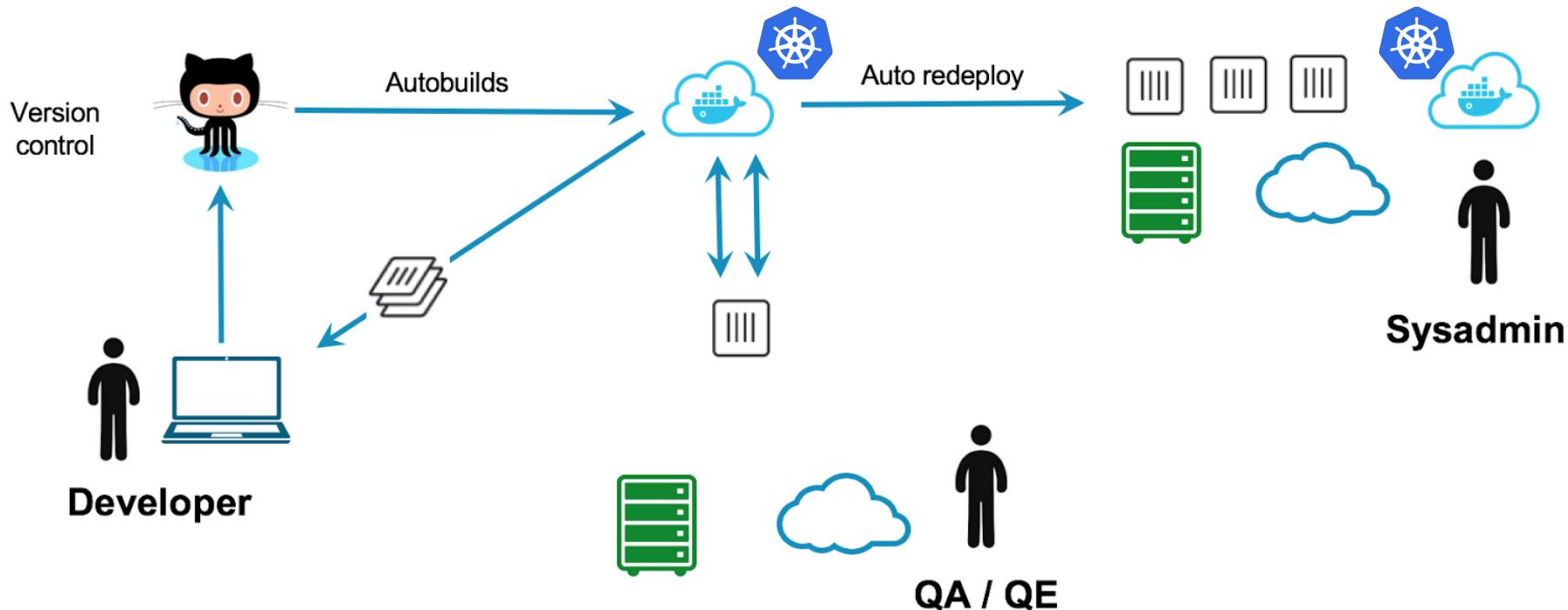
It Worked on my Dev Box

Portability: Frictionless across environments

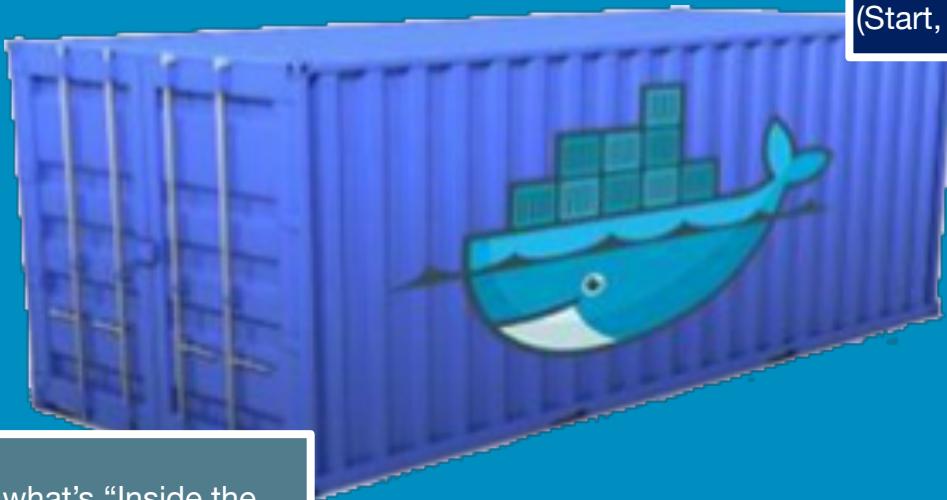
1. Development

2. Test

3. Stage / Production



Container Separation



Dan the Developer

Worries about what's "Inside the Container"

- Code
- Libraries
- Package Manager
- Data

All the servers look the same

Oscar the Ops Guy

Worries about what's
"Outside the Container"
- Logging
- Monitoring
- Networking

All containers work the same way
(Start, Stop, etc)

Docker Swarm

Docker Engine

Orchestration Components

Swarm Mode Manager

Swarm Mode Worker



TLS



Certificate Authority



Load Balancing



Service Discovery



Distributed store



Networking



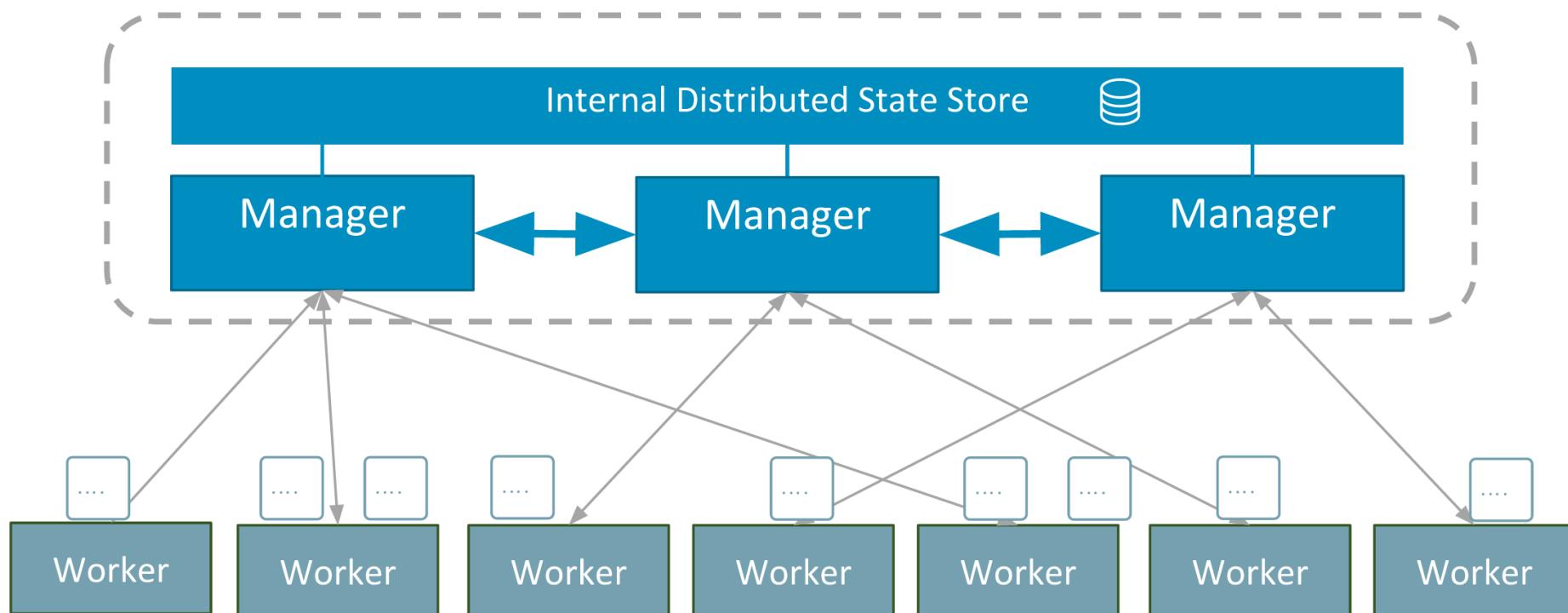
Volumes



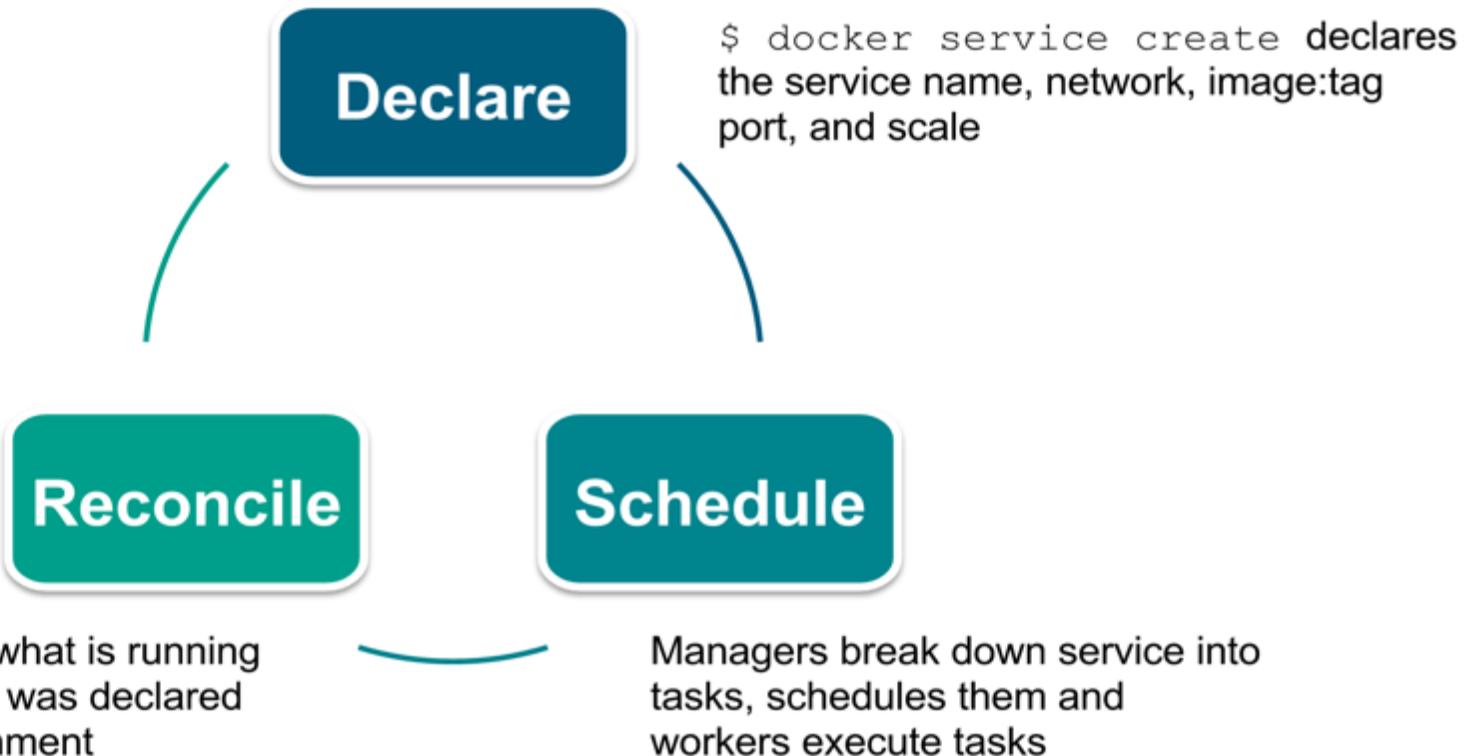
Plugins

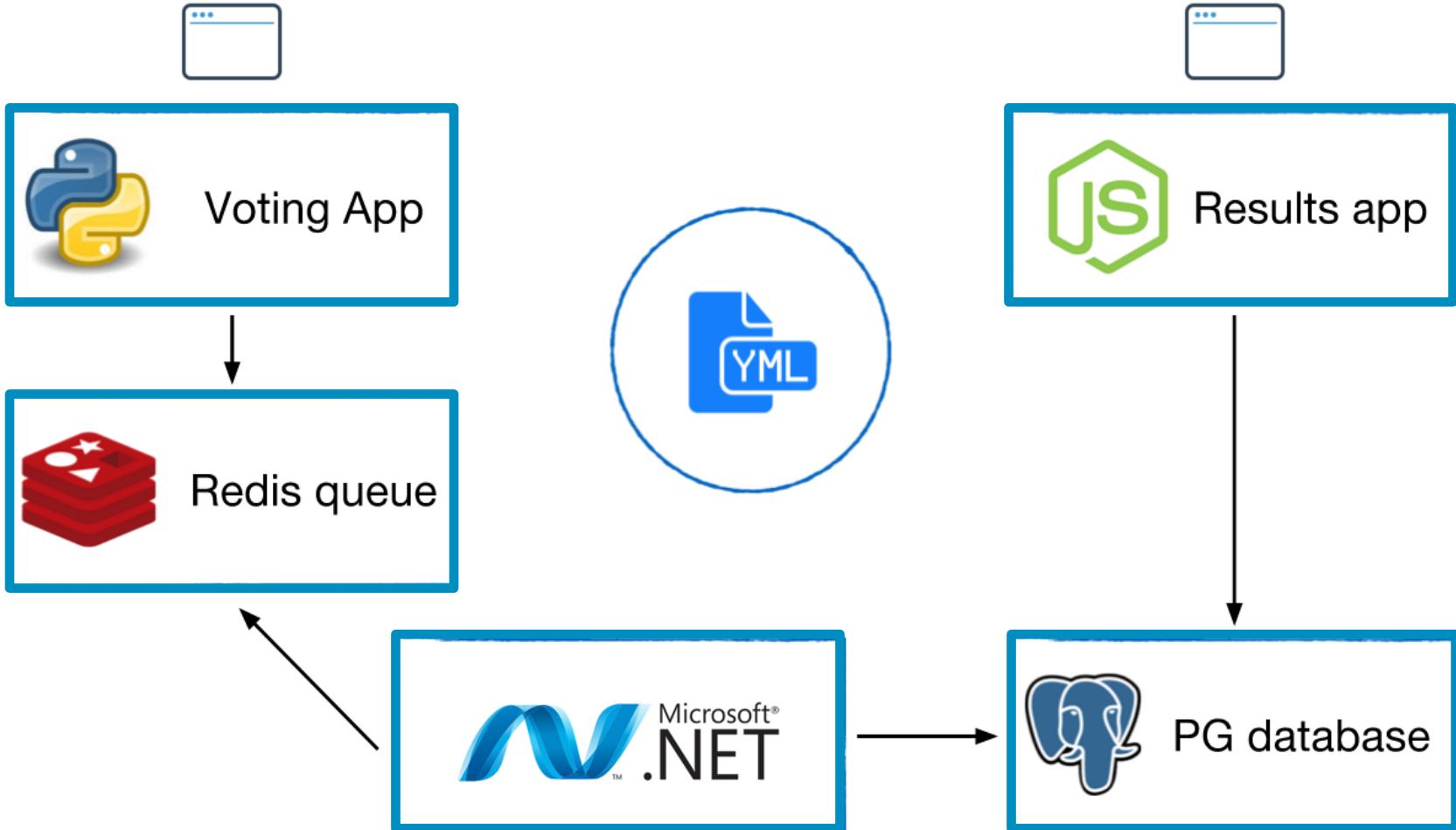
Container Runtime

Swarm Architecture



Deployment





Serverless

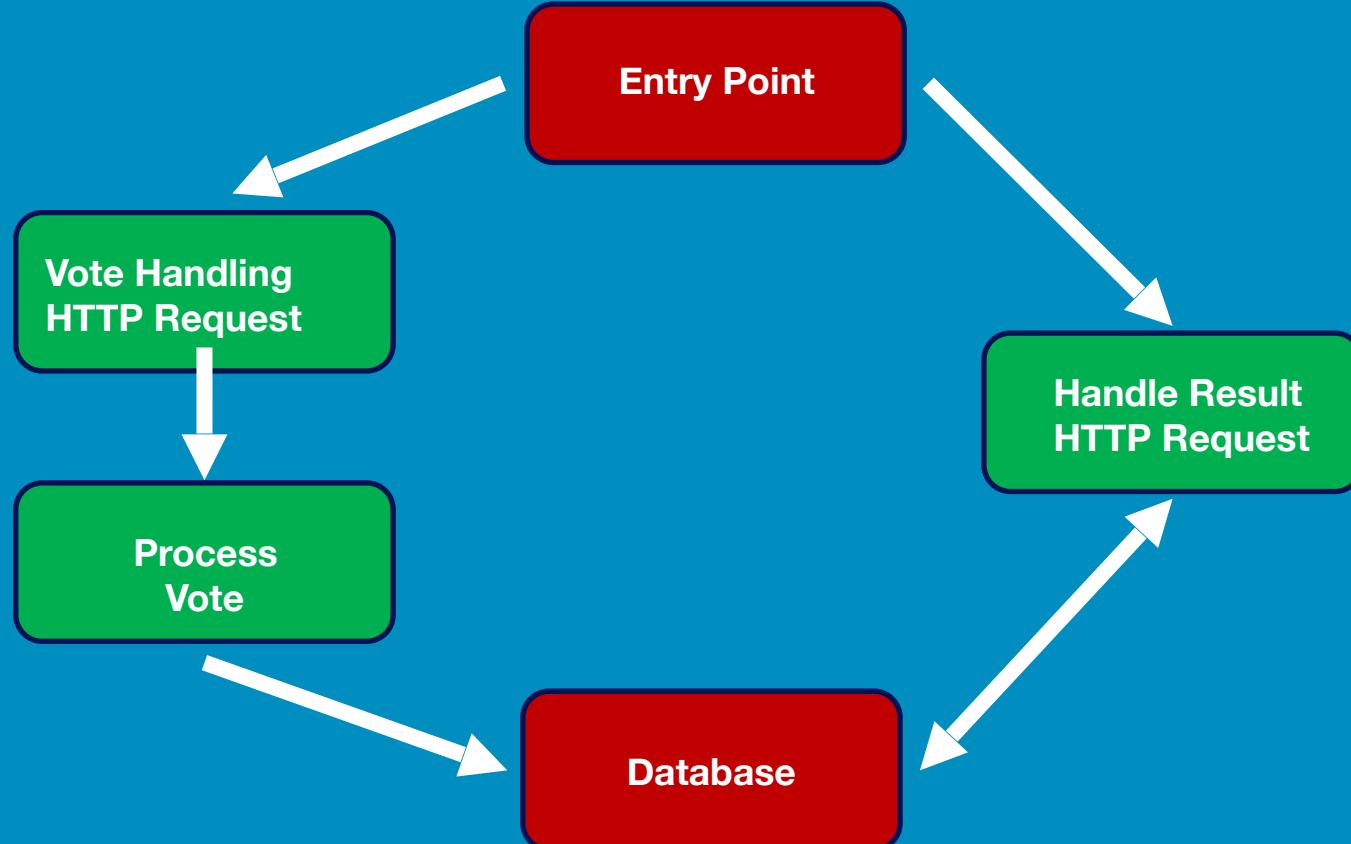
Future == Serverless



Serverless is made of Servers!



Serverless Voting App





OPEN FAAS

Functions as a Service

API Gateway

Function Watchdog



Prometheus



Swarm



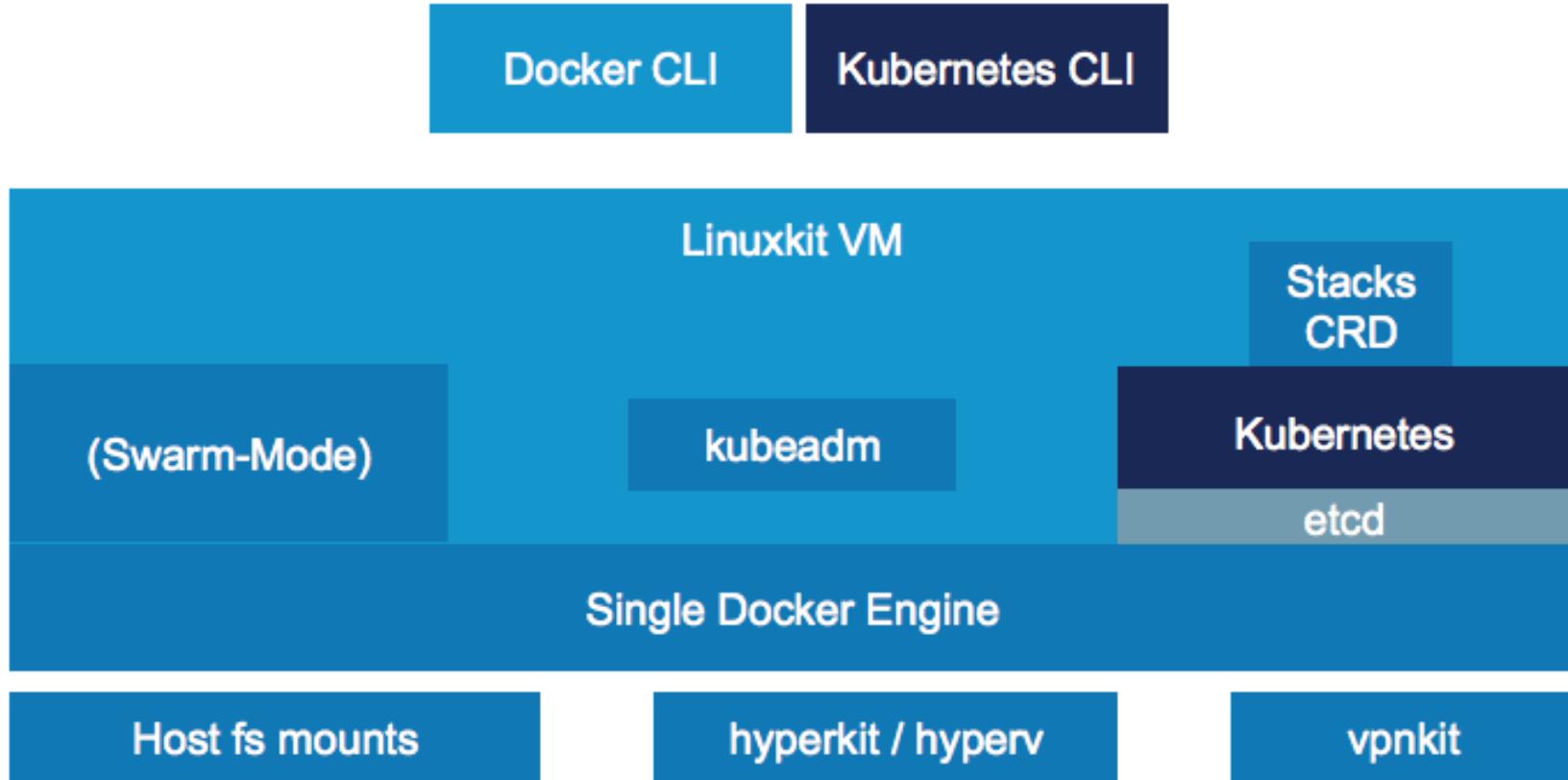
Kubernetes



<https://github.com/alexellis/faas>

Docker & Kubernetes

Docker Architecture



Introducing “Stack” API controller



Docker
Compose

Docker
Stack
Deploy



API
controller

compose.docker.com



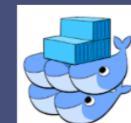
K8s objects:

Pods
Services
Deployment



kubernetes

Swarm
Services



Docker Architecture

The screenshot shows the Docker settings interface in macOS. It features two main tabs at the top: "General" and "Kubernetes".

General Tab:

- Icons: General, File Sharing, Disk, Advanced, Proxies, Daemon, Kubernetes, Reset.
- Checkboxes:
 - Start Docker when you log in
 - Automatically check for updates
 - Include VM in Time Machine backups
 - Securely store docker logins in macOS keychain
 - Send usage statistics
- Text:

Send system version and language as well as Docker app information (e.g., starts, stops, resets).

Note: Docker always sends minimal ping with version info

To help us improve our product we automatically collect usage report. If you want to opt-out you may switch to the stable c
- Text: This is an edge version. You can switch to [stable](#).
- Status Bar: Docker is running

Kubernetes Tab:

- Icons: General, File Sharing, Disk, Advanced, Proxies, Daemon, Kubernetes, Reset.
- Checkboxes:
 - Enable Kubernetes
 - Show system containers (advanced)
- Text: Start a Kubernetes single-node cluster when starting Docker and set Kubernetes as Docker orchestrator.
- Text: Show Kubernetes internal containers when using Docker commands.
- Buttons: Apply

Status Bar: Docker is running

Monitoring

Logging Tools Overview



Logging Challenges

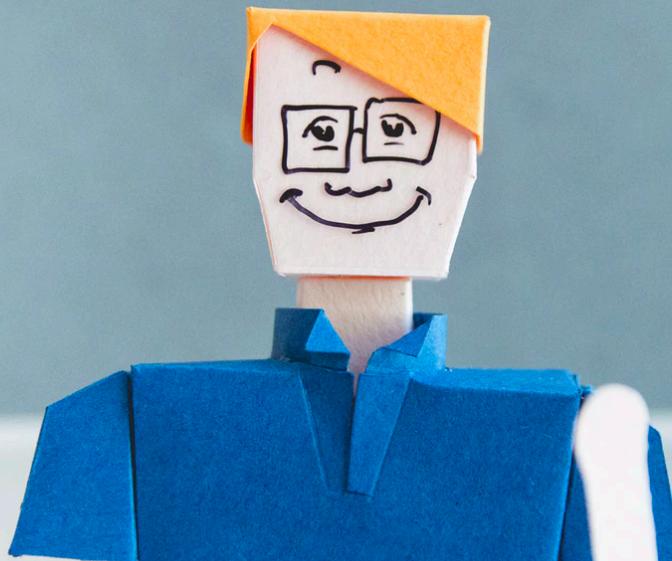
- What should we log?
- Where and how long should we store logs
- Analysis

Logging Tools

- docker logs
- docker-compose logs
- docker service logs
- log drivers

Logging Workshop

```
<?php echo  
"Hello World!"  
?>
```

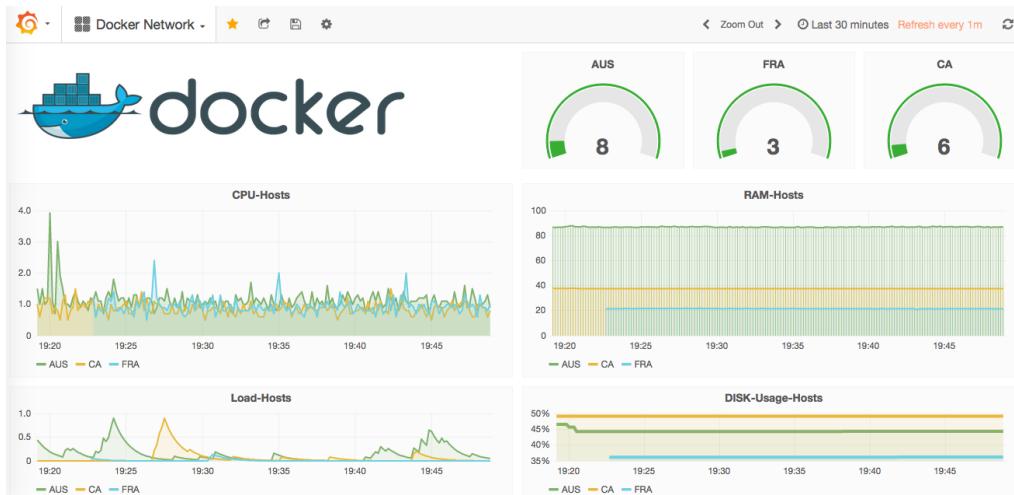


Let's Get Started!

github.com/56kcloud/training

The Basis of Monitoring

- What's Broken?
- Why is it Broken?
- How Long has it been broken?



Monitoring Tools

- Docker Stats
- Docker Top
- Docker df
- cAdvisor
- Prometheus



- Live container resources
- All containers or single
- Very basic but useful info

Docker Stats

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
35605185bf52	cadvisor	0.84%	43.18MiB / 1.952GiB	2.16%	1.31MB / 606MB	19.1MB / 0B	11
d4bd451f0b68	demo3_worker_1	0.65%	16.4MiB / 1.952GiB	0.82%	47.7kB / 54.7kB	43MB / 0B	17
1aaa67a5a5a8	redis	0.23%	1.367MiB / 1.952GiB	0.07%	26.2kB / 14.6kB	2.77MB / 0B	4
d6d556507c0f	demo3_vote_1	0.45%	31.66MiB / 1.952GiB	1.58%	2.73kB / 0B	13.5MB / 0B	3
511eb5d3a5f4	db	0.29%	9.766MiB / 1.952GiB	0.49%	34.6kB / 35.2kB	27.6MB / 90.1kB	8
7c726e569b7b	demo3_result_1	0.08%	43.22MiB / 1.952GiB	2.16%	5.35kB / 3.29kB	34.2MB / 4.1kB	20

Docker Top

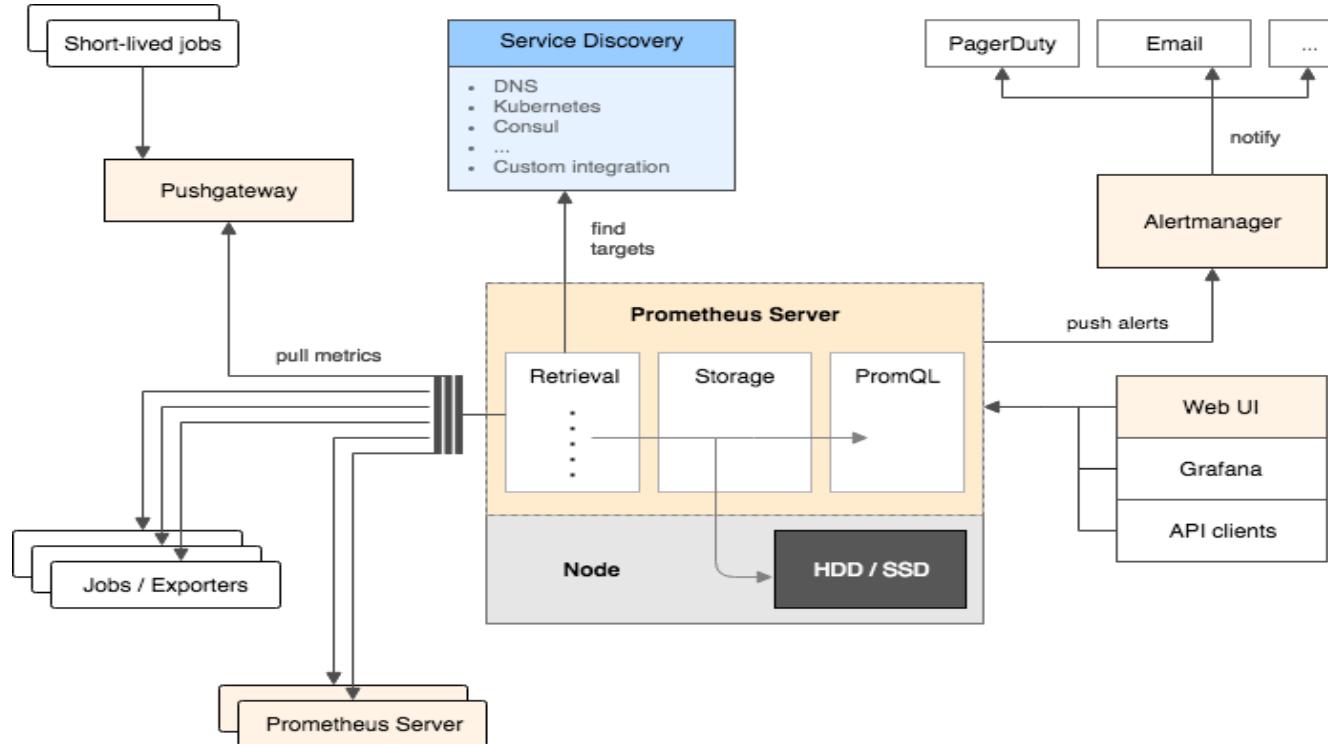
- Display running process in a container

```
vegasbrianc@Brian-56K:~$ docker top loving_hofstadter
 PID          USER          TIME        COMMAND
 97904        root         0:00        ping 8.8.8.8
vegasbrianc@Brian-56K:~$ █
```

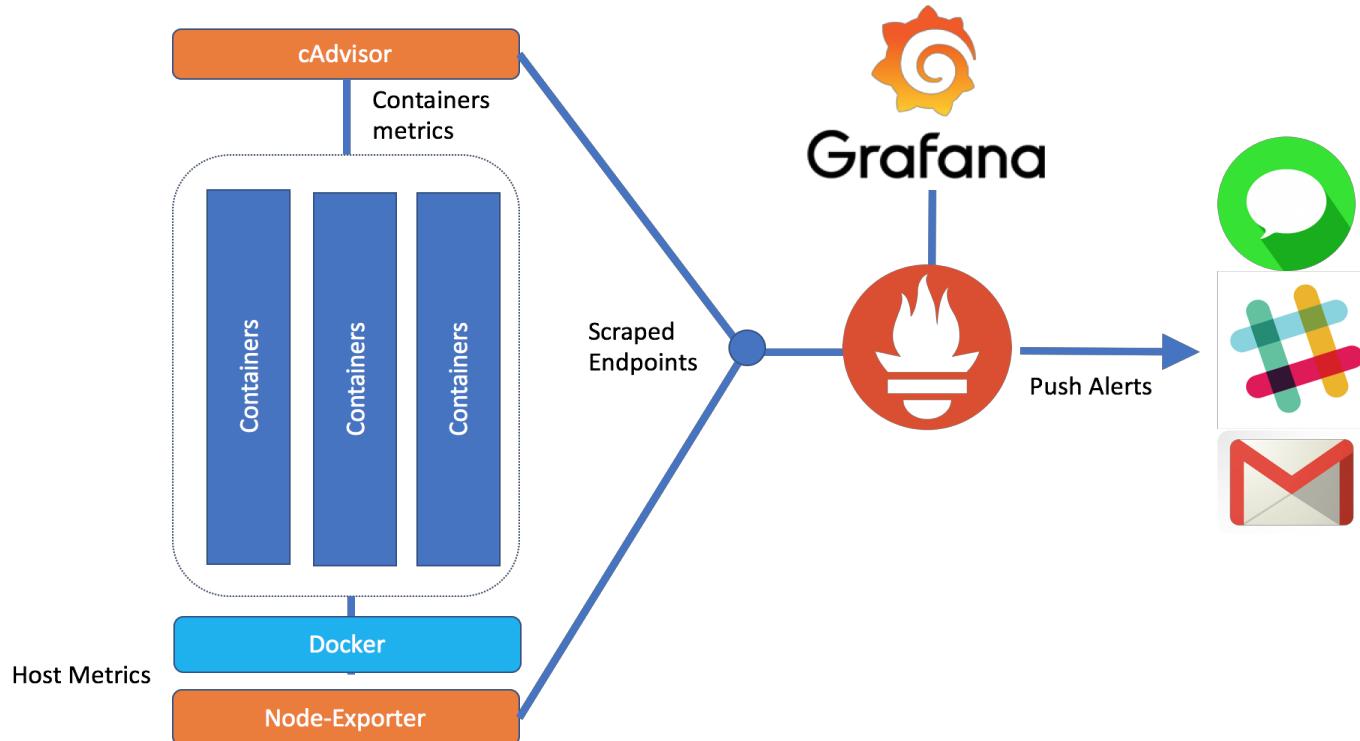


- Developed by Google
- Real-time Data
- Clean UI
- Exposes Metrics
- Integrates well

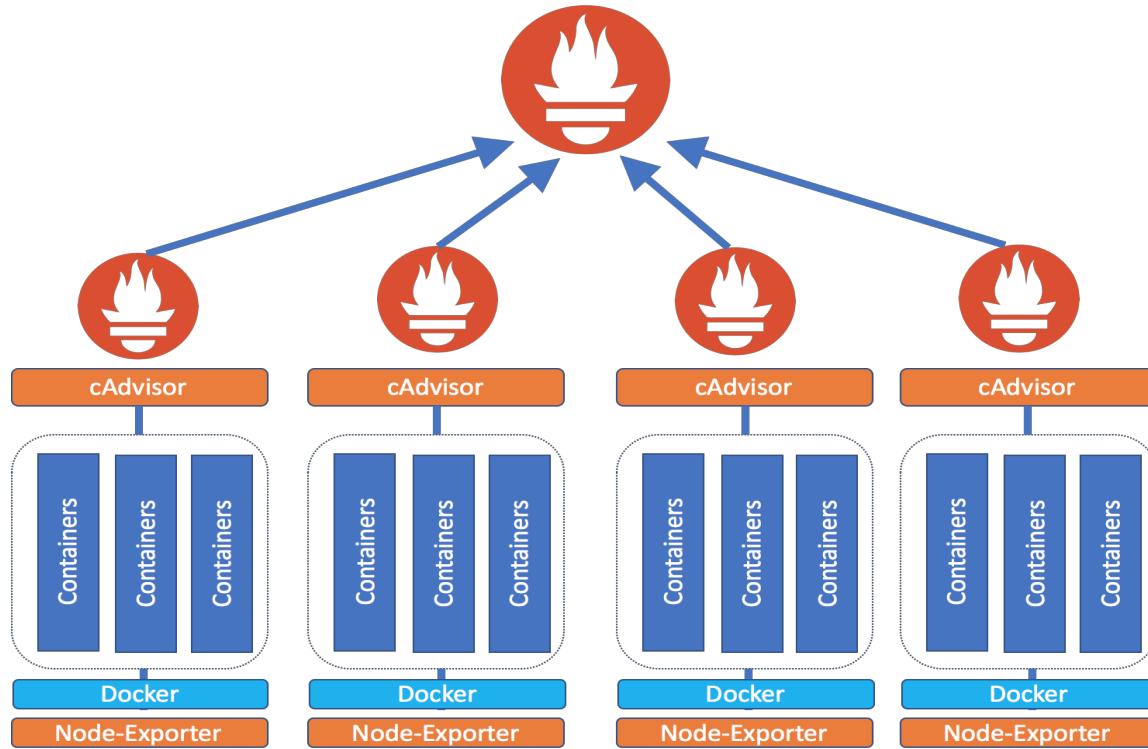
Prometheus



Prometheus Stack



Federated Configuration

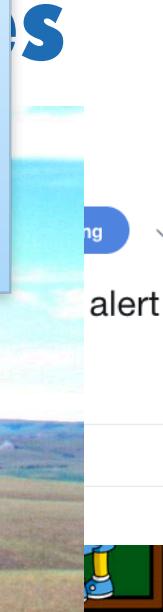


Break out the best practices

Pic
eve
5:36 F

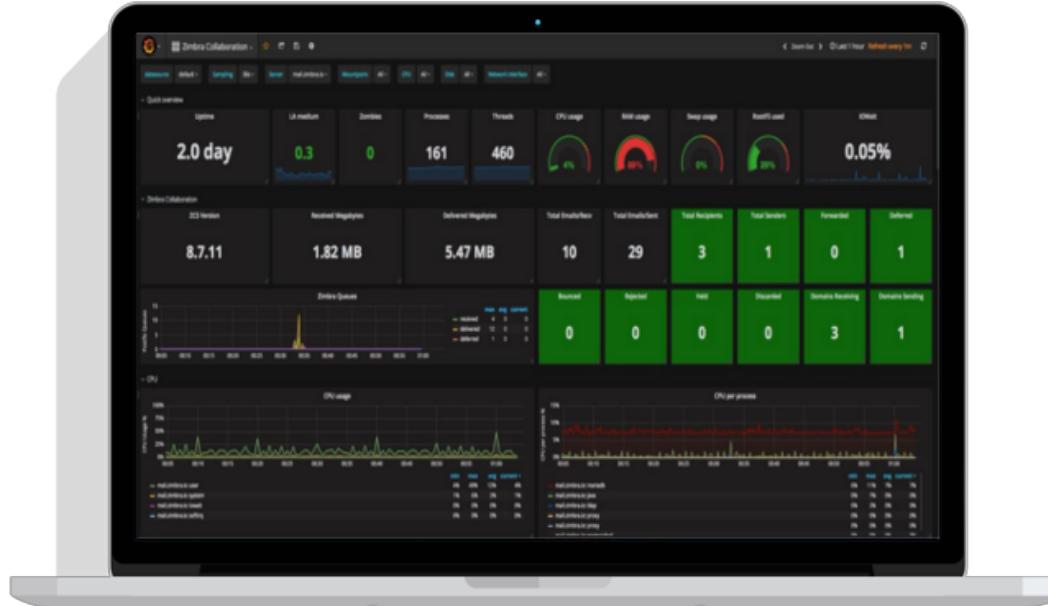
964 R

Q 2



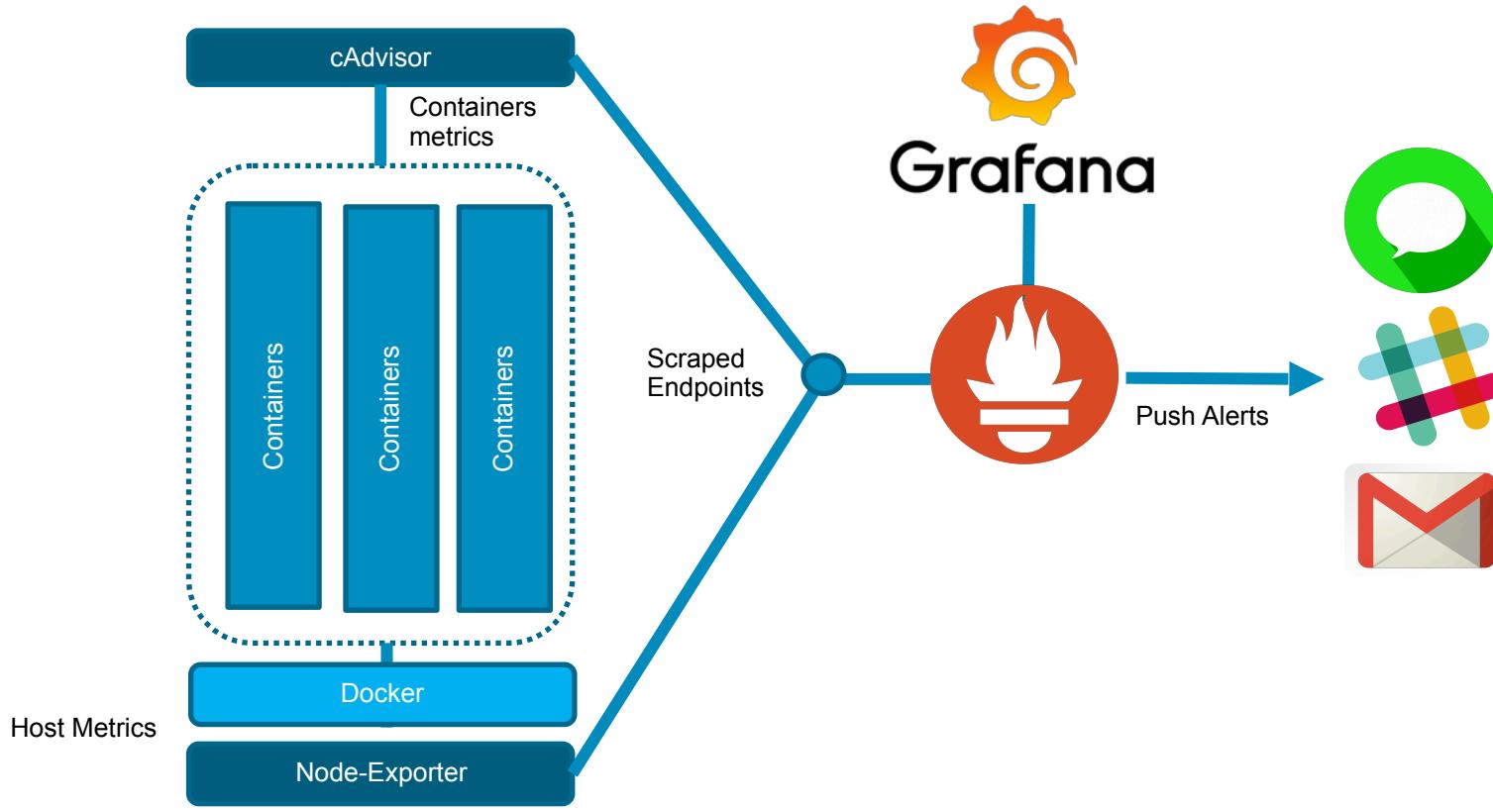
- Start small & increment
- Don't Overlert yourself
- Set Resource Limits
- Aim for actionable Information
- Run separate from Workload
- Test for Failures
- Know your Failure Models

Custom Monitoring

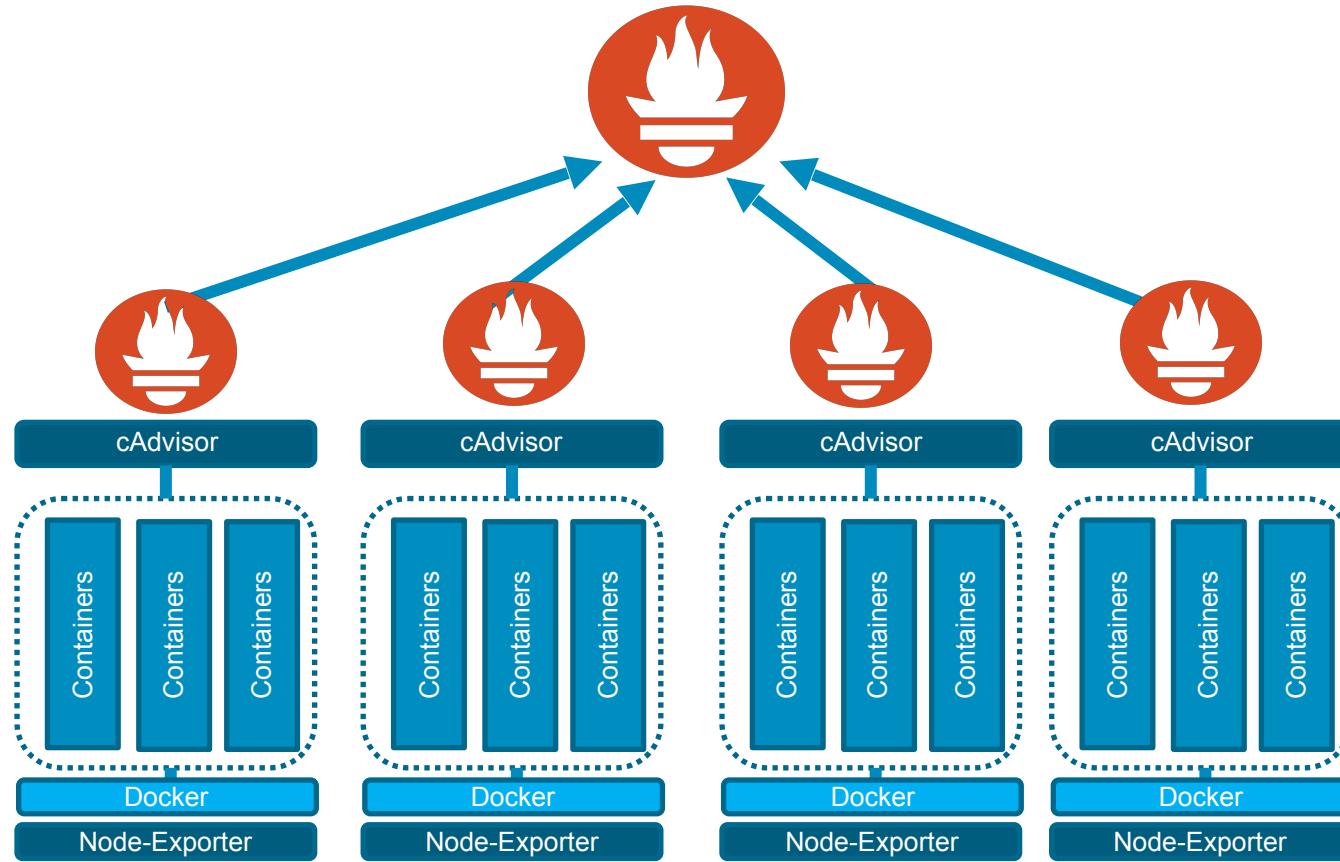


<https://monitor.webwren.ch/>

Monitoring Single Tenant



Monitoring multiple tenants



Docker Resources

Resources

- 56K.Cloud - <https://56K.Cloud>
- Prometheus - <https://github.com/vegasbrianc/prometheus>
- ELK - <https://github.com/deviantony/docker-elk>
- Labs – github.com/56kcloud/Training/tree/master/DockerCon
- Docker Resource Link - <https://awesome-docker.netlify.com>

Even More Resources

- GitLab Dashboards - <https://monitor.gitlab.net>
- Grafana Dashboards - <https://grafana.com/dashboards>



Thank You!!