Containing Infrastructure
The Internet on Kubernetes

Brandon Philips
CTO & Co-Founder, CoreOS
@BrandonPhilips
3,424,000,000

Internet Users

Source: https://www.infoq.com/news/2014/01/IDC-software-developers
29,000,000
Software Developers and IT Practioners

Source: http://www.internetlivestats.com/internet-users/
We are Outnumbered!
238,975,082
New Internet Users in 2016

Source: http://www.internetlivestats.com/internet-users/
~100,000,000
Servers Worldwide
3 Per Person
In the Software and IT Industry
100+ Per Person
At the Internet Giants
How do they do it?
Software Systems: Containers, Clustering, Monitoring

Enabling Teams To: Organize, Specialize, Take Risks
you
you as a sw engineer
with Ada.Text_IO;

procedure Hello_World is
  use Ada.Text_IO;
begin
  Put_Line("Hello, world!");
end;

#include <stdio.h>

int main()
{
    printf("Hello, world!\n");
}

package main

import "fmt"

func main() {
    fmt.Println("Hello, world!")
}
your container image
your

/bin/java
/opt/app.jar
/lib/libc
your

/bin/python
/opt/app.py
/lib/libc
your

quay.io/coreos/etcd
your

quay.io/coreos/etcd

sha256:d474e8c57737
you as an ops engineer
your
your

quay.io/coreos/etcd
x3

your
your

quay.io/coreos/etcd

x3
your

quay.io/coreos/prometheus

x1
your

quay.io/coreos/dex

x2
Containers

Application Packaging
package main

import "fmt"

func main() {
    fmt.Println("Hello, world!")
}
Building and Hosting

$ git clone git://github.com/coreos/etcd
$ docker build . -t quay.io/coreos/etcd:v3.1.0
$ docker push quay.io/coreos/etcd:v3.1.0
$ cat Dockerfile
FROM golang
ADD . /go/src/github.com/coreos/etcd
ADD cmd/vendor /go/src/github.com/coreos/etcd
RUN go install github.com/coreos/etcd
EXPOSE 2379 2380
ENTRYPOINT ["etcd"]
container image
/bin/etcd
/lib/libc
### Repository Tags

<table>
<thead>
<tr>
<th>TAG</th>
<th>LAST MODIFIED</th>
<th>SECURITY SCAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>v3.1.0</td>
<td>6 minutes ago</td>
<td>9 High + 119 others [More Info]</td>
</tr>
</tbody>
</table>
Running Containers

Application Packaging
Normal Linux Processes!

```
$ docker run -d quay.io/philips/etcd:v3.1.0 -d
$ pidof etcd
24725
$ cat /proc/24725/mounts | grep overlay
/ overlay workdir=/var/lib/docker/overlay/7134
```
Abstract away app from the OS
Containers

Required Software

- kernel
- systemd
- sshd
- docker
- rkt

- Python
- Java
- nginx

App1
App2
App3
Clustering
Clustering

Server Botnets!
100+ Per Person
At the Internet Giants
100+ Per Person
Too many for manual placement
100+ Per Person
Too many for manual placement
100+ Per Person

Too many for manual placement
$ while read host; ssh $host ... < hosts
???
$ while read host; ssh $host ... < hosts

???
$ while read host; ssh $host ... < hosts

Problems: No monitoring, no state to recover
$ kubectl run --replicas=3 quay.io/coreos/dex
$ kubectl run --replicas=3 quay.io/coreos/dex

Solution: Monitoring, and state on computers
$ kubectl run --replicas=3 quay.io/coreos/dex
$ kubectl run --replicas=3 quay.io/coreos/dex
$ kubectl run --replicas=3 quay.io/coreos/dex
Simple cluster operations

Secure and Simple API
Friendly operational tools
Clustering
Facing Failure
Simple cluster operations

Secure and Simple API
Friendly operational tools
etcd Overview

Introduced in 2013 by CoreOS
Persistent database of Kubernetes
Auto-leader election for availability
[PUT] success! "foo" : "BAZ" (at 2016-03-16 05:36:42 PST)
[PUT] success! "foo" : "bar" (at 2016-03-16 05:38:17 PST)
etcd1

ID: b414e782d52bc45a
Endpoint: 10.128.0.2:1278
State: Leader
Number of Keys: 5
Hash: 2219536697

[PUT] success! "foo" : "bar" (at 2016-03-16 05:38:17 PST)
[PUT] success! "foo" : "bar" (at 2016-03-16 05:38:17 PST)
[PUT] success! "foo": "bar" (at 2016-03-16 05:38:17 PST)
etcd1
(Hash: 0)

etcd2
(Hash: 2219536697)

etcd3
(Hash: 2219536697)

etcd4
(Hash: 2219536697)

etcd5
(Hash: 2219536697)

ID: unknown
Endpoint: 10.128.0.2:1278
State: unreachable
Number of Keys: 0
Hash: 0

[PUT] error grpc: timed out trying to connect (key: "foo")
[PUT] error grpc: timed out trying to connect (key: "foo")
Kubernetes Everywhere

Consistent Infrastructure Everywhere
kubectl
k8s API
Azure APIs

kubernetes v1.5.2

VirtualNet

VM
Disk
VM
Disk
VM
Disk
kubectl

k8s API

kubernetes v1.5.2

Bare Metal

SAN

Ethernet
Consistency on all major components

- **Compute**: Physical, Virtual Machine, Cloud
- **Networking**: VXLAN, BGP, IPIP, forwarding, etc
- **Storage**: EBS, NFS, GlusterFS, Cinder, etc
- **Load Balancing**: ELB, nginx, Cloud LB, etc
Federation

Handling Too Much Success
work-in-progress
Service Discovery

Labels, The Kubernetes Way
Scaling Applications

Reacting to Demand
ReplicaSet

ReplicaSet

Pod

app=web, env=prod

desired=3

count=1

scale up
ReplicaSet

```
app=web, env=prod
```

desired=3

count=3

```
app=web
env=prod
```

kubectl

start
Overview

Scaling Complex Apps
Creating a Database is Easy on Kubernetes

$ kubectl run db --image=quay.io/my/db
Managing a Distributed Database is Harder

- **Resize/Upgrade** - coordination for availability
- **Reconfigure** - tedious generation / templating
- **Backup** - requires coordination on instances
- **Healing** - observe and act for recovery
Introducing Operators
$ cat my-db-cluster.yaml
spec:
  clusterSize: 3
  readReplicas: 2
  version: v4.0.1
An Operator represents human operational knowledge in software, to reliably manage an application.
etcd Operator Resource

```
$ cat etcd-cluster.yaml
spec:
  clusterSize: 3
version: v3.1.0
```
etcd Operator

Cluster “A” has 2 running pods:
- name: A-000, version 3.0.9
- name: A-001, version 3.1.0

Differences from desired config:
- should be version 3.1.0
- should have 3 members

How to get to desired config:
- Recover 1 member
- Back up cluster
- Upgrade to 3.1.0
Monitoring

Foundation of Production
Prometheus Operator

- Operates Prometheus on k8s
- Handles common tasks:
  - Create/Destroy
  - Monitor Configuration
  - Services Targets via Labels
- Configured by resources
## Services

<table>
<thead>
<tr>
<th>SERVICE NAME</th>
<th>SERVICE LABELS</th>
<th>POD SELECTOR</th>
<th>SERVICE LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>host-info</td>
<td>app=host-info</td>
<td>q app=host-info</td>
<td>10.3.0.188:80</td>
</tr>
<tr>
<td>host-info-prometheus</td>
<td>No labels</td>
<td>q prometheus=host-info-prometheus</td>
<td>10.3.0.148:80</td>
</tr>
<tr>
<td>kubernetes</td>
<td>component=apiserver provider=kubernetes</td>
<td>No selector</td>
<td>10.3.0.1:443</td>
</tr>
<tr>
<td>prometheus-operated</td>
<td>No labels</td>
<td>q app=prometheus</td>
<td>None:9090</td>
</tr>
</tbody>
</table>
You are visitor: 3919

host-info-2478745576-m6pff

Unknown Zone

Networking

lo

- 127.0.0.1/8
- ::1/128

eth0
What's Next?

Kubernetes Next Steps
Kubernetes Next Steps

- Healthy growth of the open source community
- Better metrics and monitoring
- Ever improving security defaults
- Support for more cloud platforms
- More pre-packaged applications
- https://github.com/kubernetes/features
CoreOS Mission

Secure the Internet
Documents

Commerce

Communications
~100,000,000
Servers Worldwide
Self Driving Infrastructure
What is self-driving infrastructure

Applying updates to our infrastructure software much like the way updates are applied to our phones.
Why self driving infrastructure

Patching Software is important
Updating Container Linux

VM / Bare Metal

A

B

OS Update
Updating Container Linux

VM / Bare Metal

OS Update
Updating Container Linux

VM / Bare Metal
Updating Kubernetes
Updating Kubernetes

Deployment

Kubernetes

API Server Container
Updating Kubernetes
Azure APIs

kubectl

k8s API

Azure APIs

VirtualNet

VM

VM

VM

PD

PD

PD

kubernetes v1.5.1
Free Stuff

Get Kubernetes For Yourself
Minikube - Kubernetes on your laptop

All-in-one Virtual Machine
Single file download
Latest Kubernetes

github.com/kubernetes/minikube
Tectonic Free Tier

Pure upstream Kubernetes

Production configuration

Bare metal and AWS installer
(more coming soon)

coreos.com/tectonic
Join Us

Build great code
Container Focused Operating System

Hack on Operating Systems
Build distributed systems
Geek out on internet identity protocols

Clustered Database

Identity & Federation

10+ other major projects
Most active project on GitHub

Dozens of special interest groups

Not just code! Docs, PM, and more
Work With Us

Community is taking off
What container and PaaS tools are used to manage OpenStack applications?

<table>
<thead>
<tr>
<th>Tool</th>
<th>0%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kubernetes</td>
<td>27%</td>
<td>8%</td>
<td>7%</td>
<td></td>
<td>42%</td>
</tr>
<tr>
<td>Cloud Foundry</td>
<td>16%</td>
<td>3%</td>
<td>5%</td>
<td></td>
<td>24%</td>
</tr>
<tr>
<td>OpenShift</td>
<td>10%</td>
<td>9%</td>
<td>5%</td>
<td></td>
<td>24%</td>
</tr>
<tr>
<td>Mesos</td>
<td>11%</td>
<td>6%</td>
<td>2%</td>
<td>19%</td>
<td></td>
</tr>
<tr>
<td>Cloudify</td>
<td>3%</td>
<td>2%</td>
<td></td>
<td></td>
<td>9%</td>
</tr>
<tr>
<td>Docker Swarm</td>
<td>2%</td>
<td>4%</td>
<td>2%</td>
<td>7%</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td>18%</td>
<td>3%</td>
<td>3%</td>
<td></td>
<td>24%</td>
</tr>
</tbody>
</table>


Developer engagement

CoreOS is running the world’s containers

We’re hiring: careers@coreos.com

<table>
<thead>
<tr>
<th>OPEN SOURCE</th>
<th>ENTERPRISE</th>
</tr>
</thead>
<tbody>
<tr>
<td>90+ Projects on GitHub, 1,000+ Contributors</td>
<td>Support plans, training and more</td>
</tr>
<tr>
<td>container</td>
<td>TECTONIC</td>
</tr>
<tr>
<td>linux</td>
<td>QUAY</td>
</tr>
<tr>
<td>dex</td>
<td><a href="mailto:sales@coreos.com">sales@coreos.com</a></td>
</tr>
<tr>
<td>etcd</td>
<td></td>
</tr>
</tbody>
</table>

coreos.com
Thanks!

QUESTIONS?
brandon.philips@coreos.com
@BrandonPhilips
linkedin.com/in/brandonphilips

LONGER CHAT?
Let’s talk!
IRC
More events: coreos.com/community

We’re hiring: coreos.com/careers
Preview

Self-Driving Kubernetes