Fault Tolerant Infrastructure

Building Systems with etcd

@coreoslinux
@brandonphilips
Brandon Philips

CTO, CoreOS

github.com/philips
What is CoreOS?
What is CoreOS?
What is CoreOS?
The smartest way to run your container infrastructure.

tectonic.com  @tectonic
QUAY

Secure hosting for private Docker repositories

quay.io   @quayio
Why build CoreOS?
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

com.example.app
d474e8c57737625c
you as an ops engineer
your
your

com.example.webapp
x3

↓

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
your

com.example.webapp

x3

???
How do we do it?
cluster operations
architecture in practice
OS operations
machine configuration
cluster operations
distributed configuration
github.com/philips/hacks/tree/master/etcddemos
etcd
/etc distributed
etcd

open source software
failure tolerant
durable
watchable
exposed via HTTP
runtime reconfigurable
Data Store API

-X GET
    Get Wait

-X PUT
    Put Create CAS

-X DELETE
    Delete CAD
etcd basics
clusters
Typical Cluster

- Leader
- Follower
etcd basics API
etcd basics

fault tolerance
Available

Leader

Follower
Available

Leader

Follower
etcd basics
leader fault tolerance
Available

Leader

Follower
Available

Leader

Follower
Unavailable

Leader

Follower
etcd durability
wal, snapshots, backups
etcd bootstrap
discovery, static
$ curl discovery.etcd.io/new?size=5
discovery.etcd.io/6eadeac2
discovery
discovery
discovery

Leader

Follower
etcd reconfig
live addition and removal
etcd apps
Cluster Wide Reboot Lock

- Need to reboot? Decrement the semaphore key atomically with etcd.
- manager.Reboot() and wait...
- After reboot increment the semaphore key in etcd atomically.
etcd apps
skydns
etcd apps
vulcand
etcd apps

confd
scheduling
k8s/mesos/etc scheduler
scheduling

getting work to servers
$ scp app host:/opt
$ ssh host systemd-run /opt/app
$ scp app host:/opt
$ ssh host systemd-run /opt/app
$ fab deploy:app
$ fab deploy:app
$ fab deploy:app
$ fab deploy:collector-app
$ fab deploy:collector-app
$ fab deploy:collector-app
$ fab deploy deploy:collector-app
$ fab lowest-loadaverage
$ fab lowest-loadaverage
host1
$ fab lowest-loadaverage
host1
$ fab -H host1 deploy:job
Scheduler API

Scheduler

Machine(s)
while true {
    todo = diff(desState, curState)
    schedule(todo)
}
while true {
    todo = diff(desState, curState)
    schedule(todo)
}
while true {
    todo = diff(desState, curState)
    schedule(todo)
}
while true {
    todo = diff(desState, curState)
    schedule(todo)
}
services
dns, LBs, k8s labels
k8s labels
flexible service discovery
service test.example.com
select(env=dev,app=web)

pod
env=dev
app=web

service beta.example.com
select(env=test,app=web)
OR
select(env=prod,app=web)

pod
env=test
app=web

pod
env=prod
app=web

service example.com
select(env=prod,app=web)
github.com/coreos/coreos-kubernetes
scheduler & API

worker kubelet

worker kubelet
works on 1 node too
thank you

@coreoslinux
@tectonicstack
@brandonphilips