The Journey to #GIFEE

Intro to the Open Source

Brandon Philips

@brandonphilips | brandon.philips@coreos.com

CoreOS

MISSION

Secure the Internet

CoreOS

STRATEGY

Separate Apps from OS

CoreOS

STRATEGY

Make Servers Consistent

CoreOS

STRATEGY

Tolerate Machine Failures

CoreOS

STRATEGY

Make Servers Easy to Upgrade

CoreOS

STRATEGY

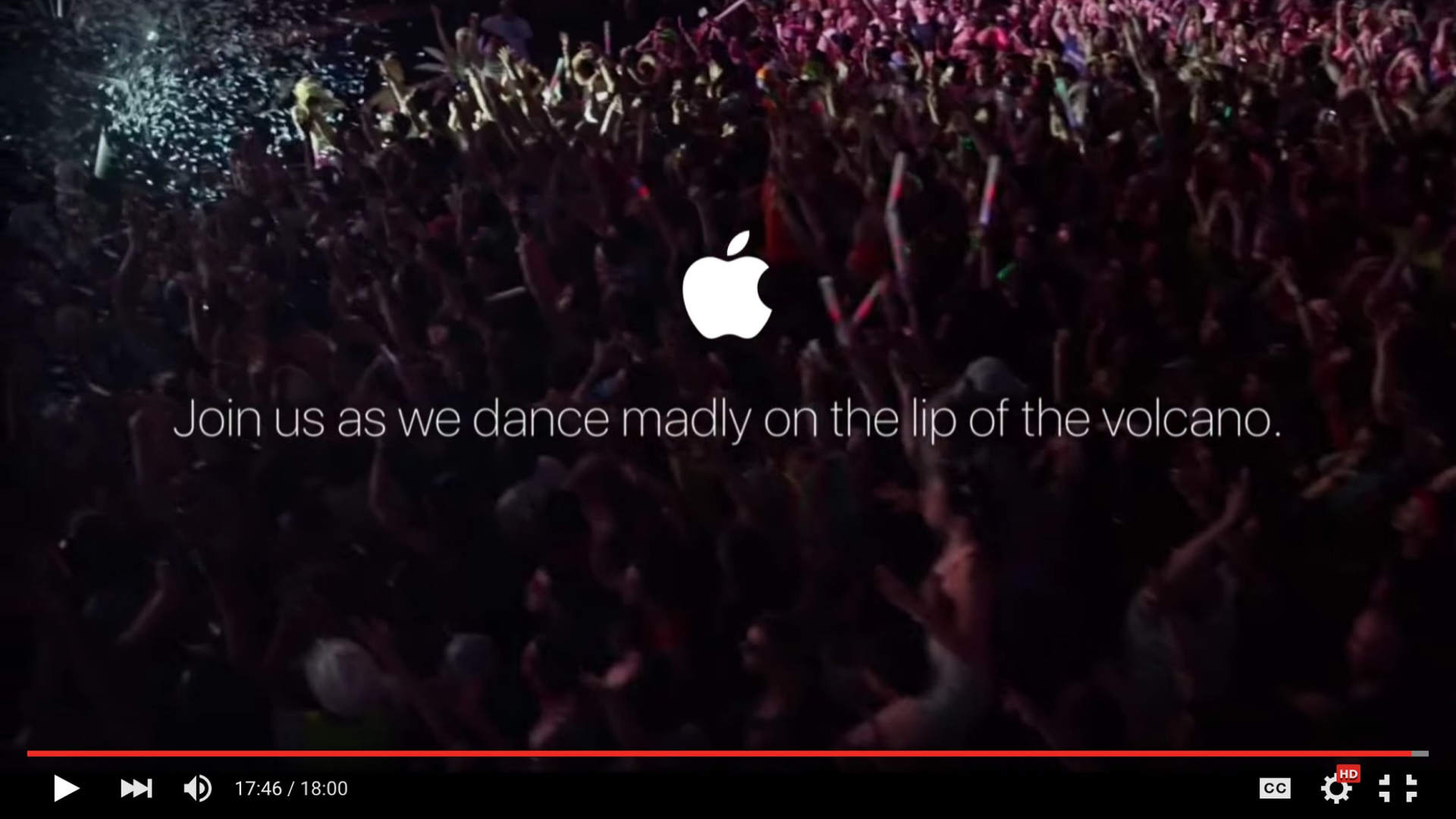
Simplify Application Upgrades


HEWLETT·PACKARD
hp 15C
ON
f
g
√x A x²
e^x B LN
10^x C LOG
y^x D %
1/x E Δ%
SST LBL BST
GTO HYP HYP-1
R/S PSE P/R
SIN DIM SIN-1
COS (i) COS-1
TAN I TAN-1
CHS MATRIX ABS
GSB Σ RTN
CLEAR
PRGM
REG
PREFIX
R↓
RND
CLx
STO FRAC INT
RCL USER MEM
ENTER RAN # LSTx
EEX RESULT π
7 FIX DEG
8 SCI RAD
9 ENG GRD
÷ SOLVE x≤y
4 x⇄ SF
5 DSE CF
6 ISG F?
× x=0
1 →R →P
2 →H.MS →H
3 →RAD →DEG
− Re⇄Im TEST
0 x! x̄
· ŷ,r s
Σ+ L.R. Σ-


ENCRYPTION
LAST WEEK TONIGHT
0:19 / 18:00
CC
HD


17:34 / 18:00
CC
HD


17:35 / 18:00
CC
HD


Join us as we dance madly on the lip of the volcano.
17:46 / 18:00
CC
HD

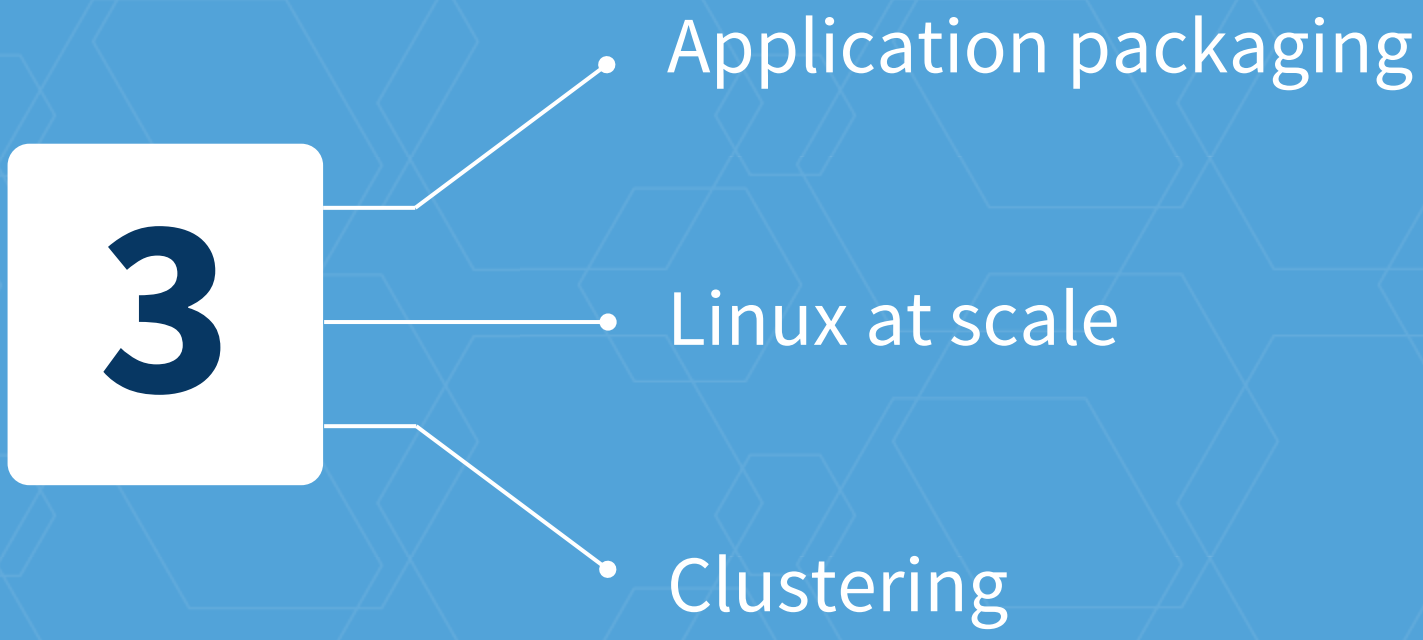

CoreOS


kubernetes


etcd


rkt


docker


3
Application packaging
Linux at scale
Clustering

CoreOS

#GIFEE

Google

Borg/Omega
Linux
Chubby

CoreOS

#GIFEE

Borg/Omega
Linux
Chubby

CoreOS

#GIFEE

Google

Borg/Omega
Linux
Chubby

Why build #GIFEE?

- Avoid single points of failure
- Design for constant updates
- Consistent environment

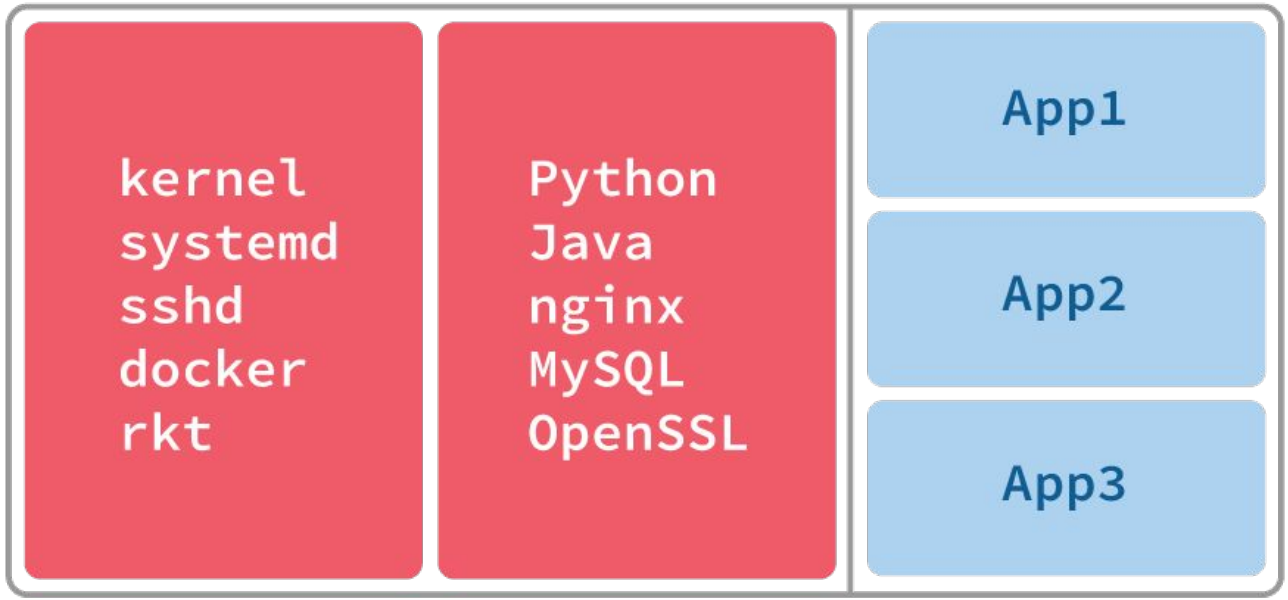
Why build #GIFEE?

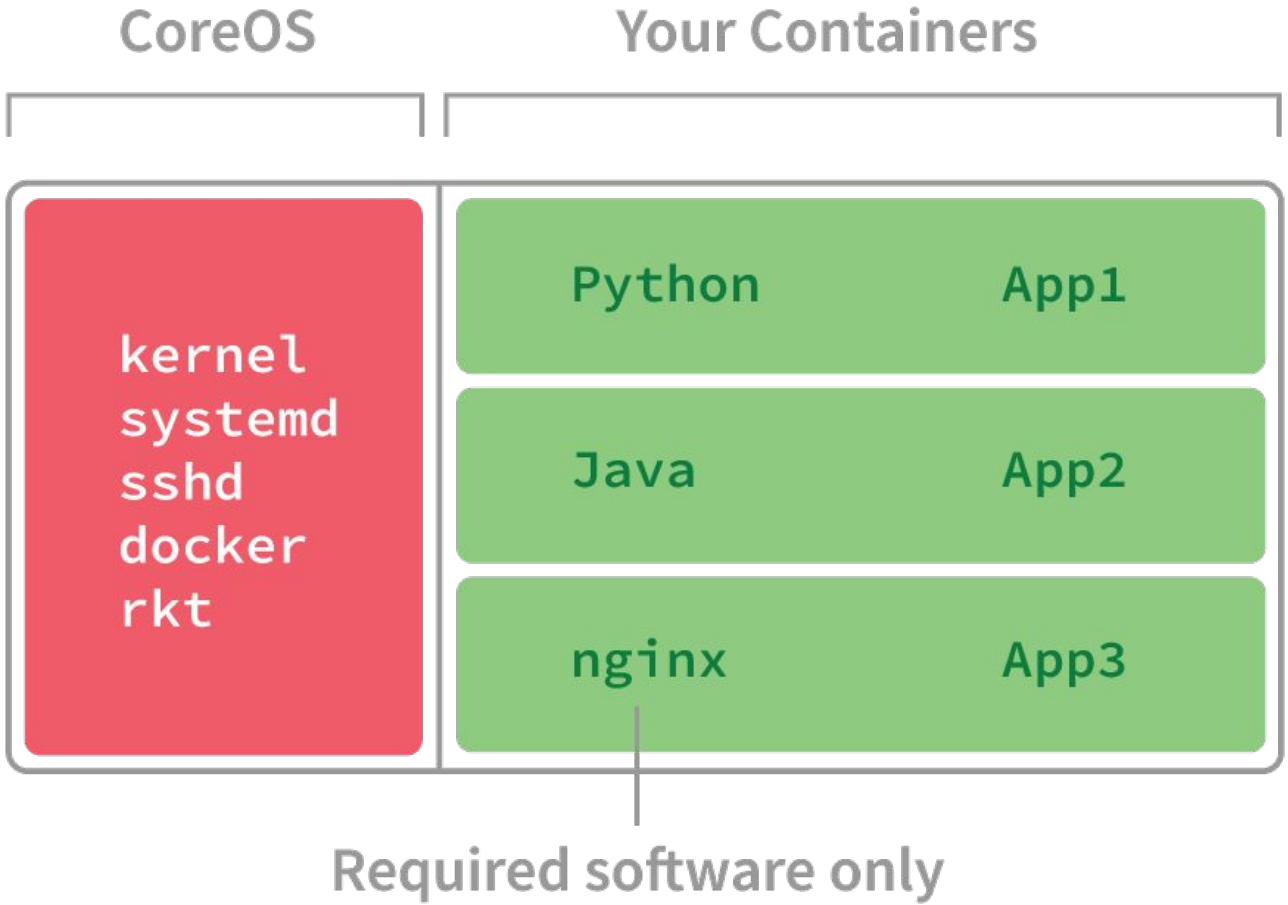
- Design for constant updates

1 Application Packaging

Abstract away app from the OS

Traditional Distro




CoreOS
Your Containers
kernel
systemd
sshd
docker
rkt
Python
App1
Java
App2
nginx
App3
Required software only

CoreOS

Base software managed by CoreOS

kernel systemd OpenSSH

Protect apps from each other

- Mixed versions of dependencies
  eg. python 3.4 & python 2.7
- Isolated network namespace
- Isolated file system namespace

rkt run

```
$ sudo rkt run coreos.com/etcd:v2.0.0

$ sudo rkt run coreos.com/etcd:v2.0.0 \
  --cpu=750m --memory=128M

$ sudo rkt run --net=host coreos.com/etcd:v2.0.0
```

Clair container security auditing

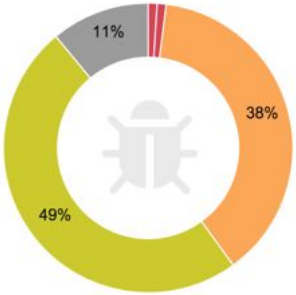
- Search container metadata
- Identify vulnerabilities
- Explain update actions

CoreOS

Clair container security auditing

After scanning millions of containers we found that over 80% still had Heartbleed

4f3f3b6e0b74



Quay Security Scanner has detected **100** vulnerabilities.

- **1** Critical-level vulnerabilities.
- **1** High-level vulnerabilities.
- **38** Medium-level vulnerabilities.
- **49** Low-level vulnerabilities.
- **11** Negligible-level vulnerabilities.

Image Vulnerabilities

Filter Vulnerabilities...

Only display vulnerabilities with fixes

CVE	CVSS / SEVERITY	PACKAGE	CURRENT VERSION	FIXED IN VERSION	INTRODUCED IN IMAGE
CVE-2014-9488	10	less	458-2	(None)	ADD file:9b5ba3935021955492697a...
CVE-2015-8391	9	pcre3	1:8.31-2ubuntu2.1	(None)	ADD file:9b5ba3935021955492697a...
CVE-2015-8380	7.5	pcre3	1:8.31-2ubuntu2.1	(None)	ADD file:9b5ba3935021955492697a...
CVE-2015-8472	7.5	libpng	1.2.50-1ubuntu2.14.04.1	1.2.50-1ubuntu2.14.04.2	ADD file:9b5ba3935021955492697a...
CVE-2015-8390	7.5	pcre3	1:8.31-2ubuntu2.1	(None)	ADD file:9b5ba3935021955492697a...

CoreOS

In-Progress
Universal Container Format

Packaged Downloaded Verified

2 Linux at Scale

Patches to the OS and kernel are hard

APPLICATION

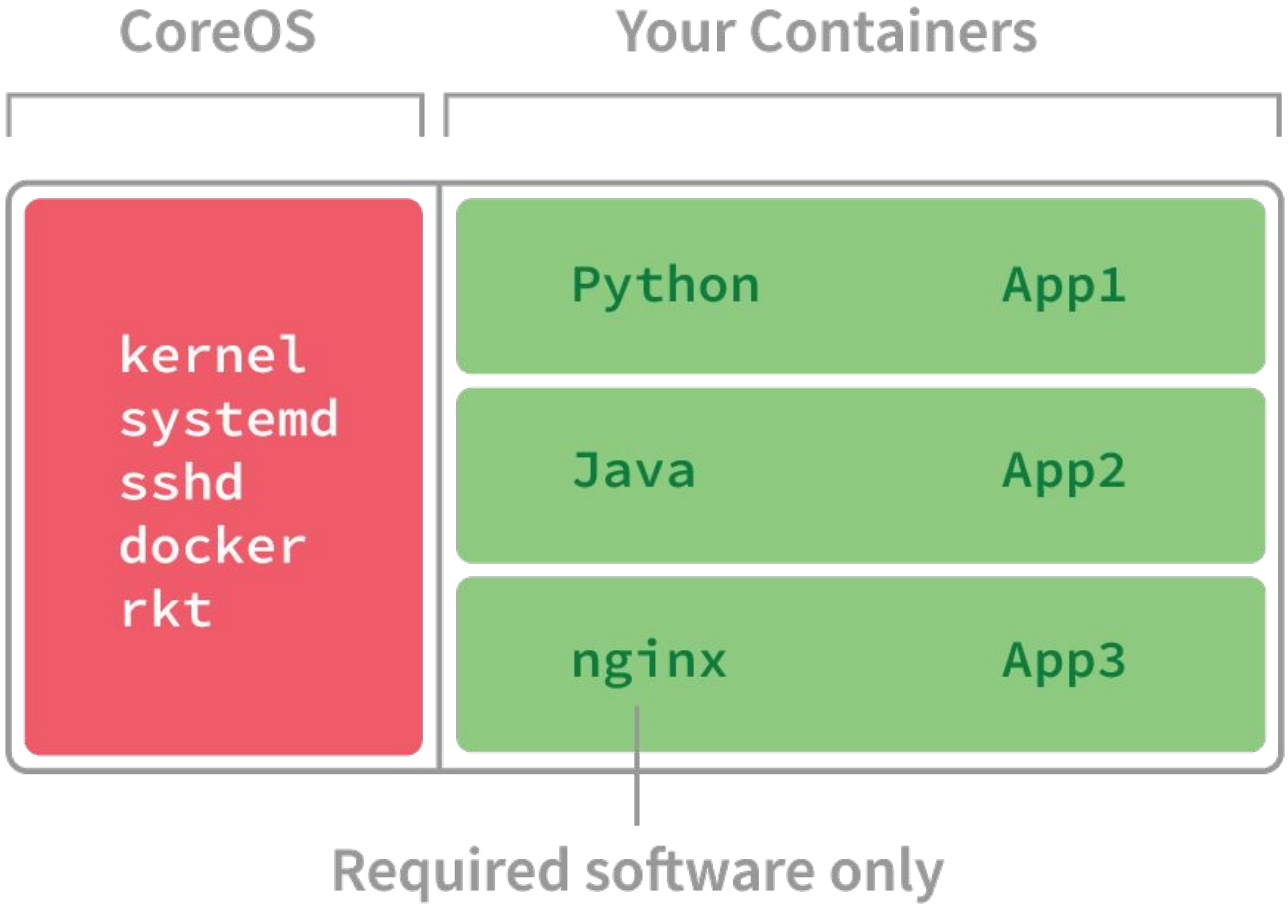
Dependency breakage

Uptime risk

SECURITY

Retest after updates

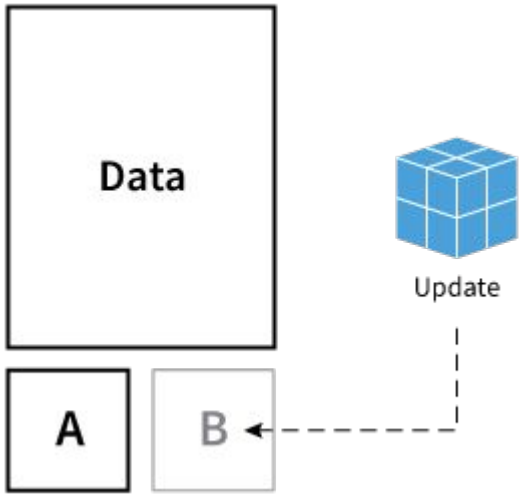
No automation


CoreOS
Your Containers
kernel
systemd
sshd
docker
rkt
Python
App1
Java
App2
nginx
App3
Required software only

Auto-updating browsers fixed security

We got HTML5 at the same time

Atomic operating system updates



Atomic operating system updates

3 Clustering

Patches to the OS and kernel are hard

APPLICATION

Dependency breakage

Uptime risk

SECURITY

Retest after updates

No automation

Patches to the OS and kernel are hard

APPLICATION

Uptime risk

SECURITY

No automation

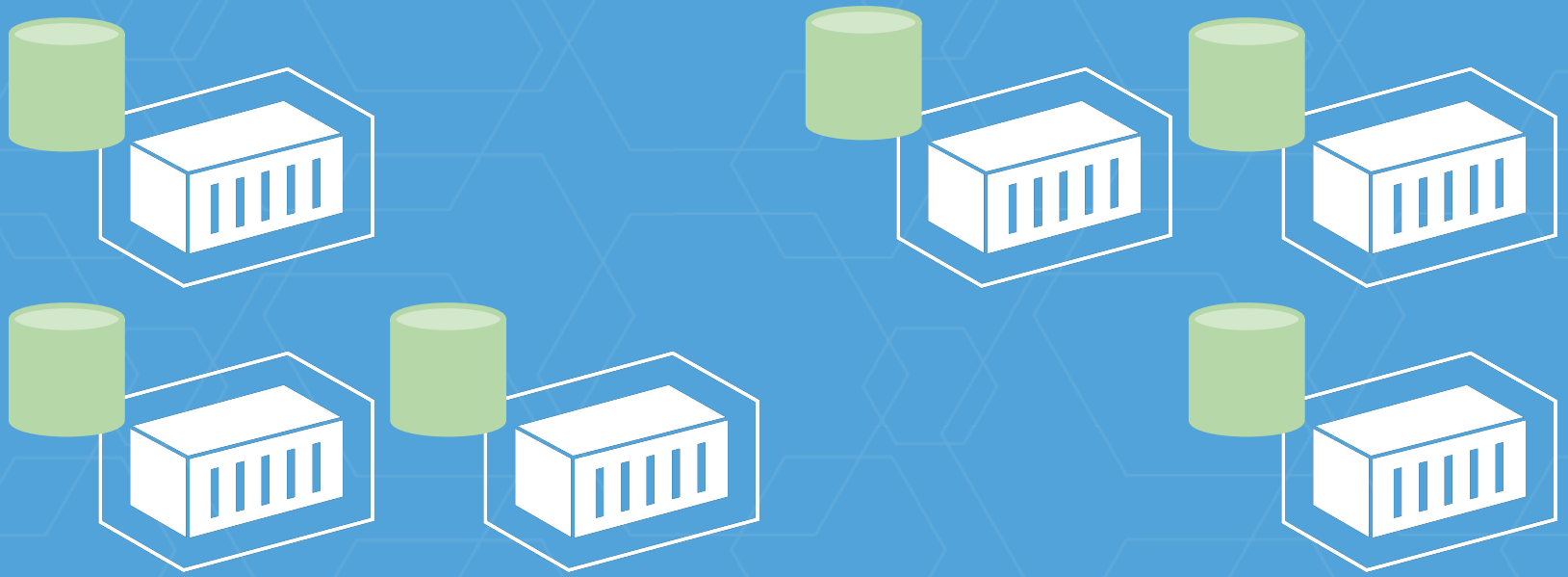
CoreOS

Operations Paradise

- Easy scale out
- Painless app upgrades
- Tolerant of machine failure

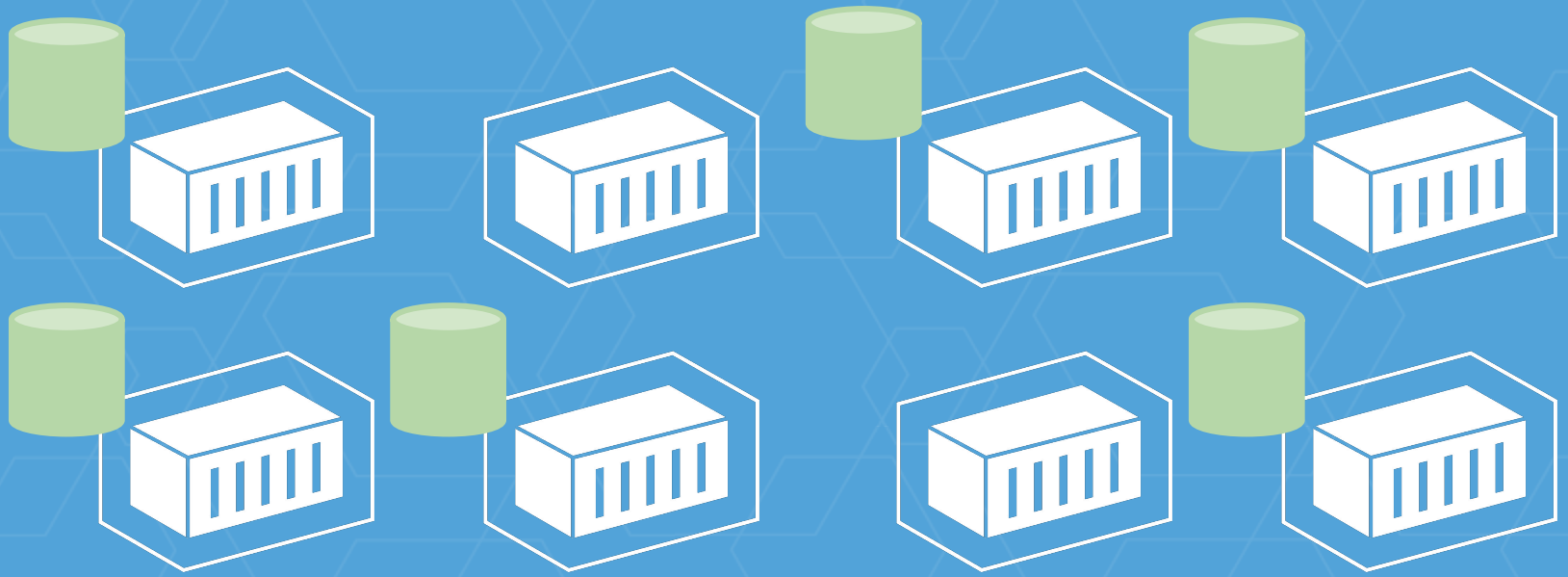
CoreOS

App Req/sec: **6,000**
App Healthy: **True**



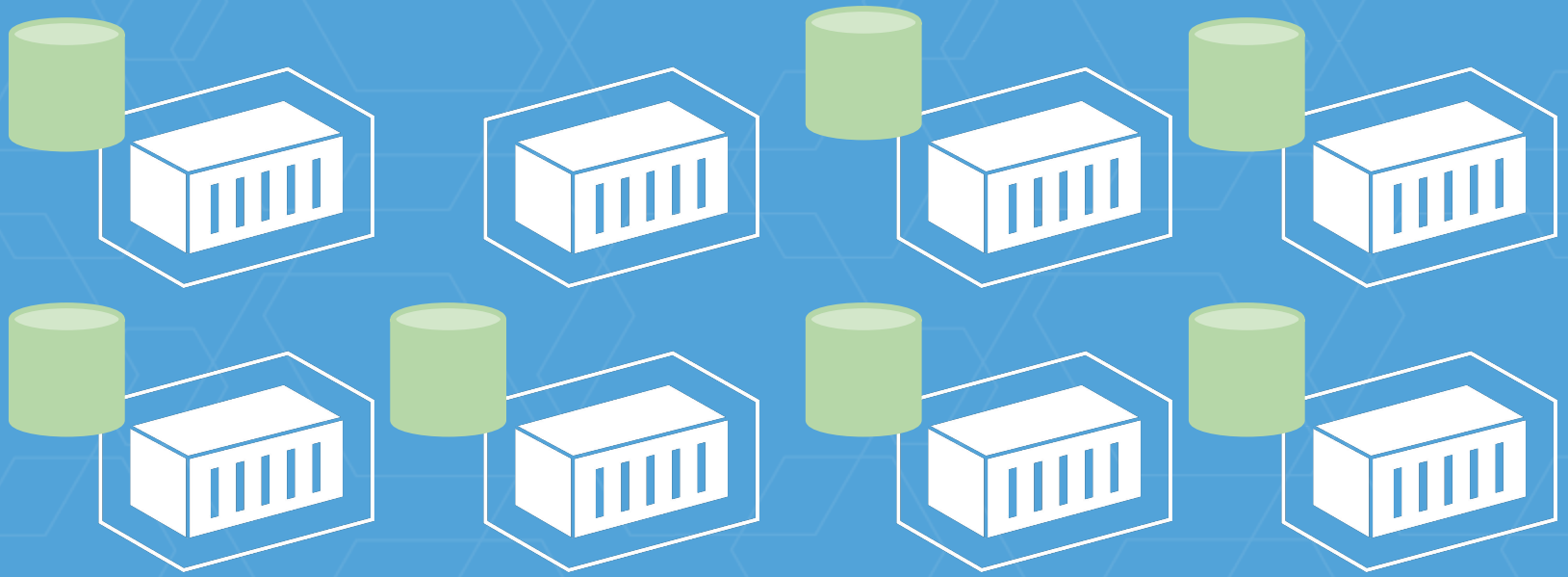
CoreOS

App Req/sec: **6,000**
App Healthy: **True**



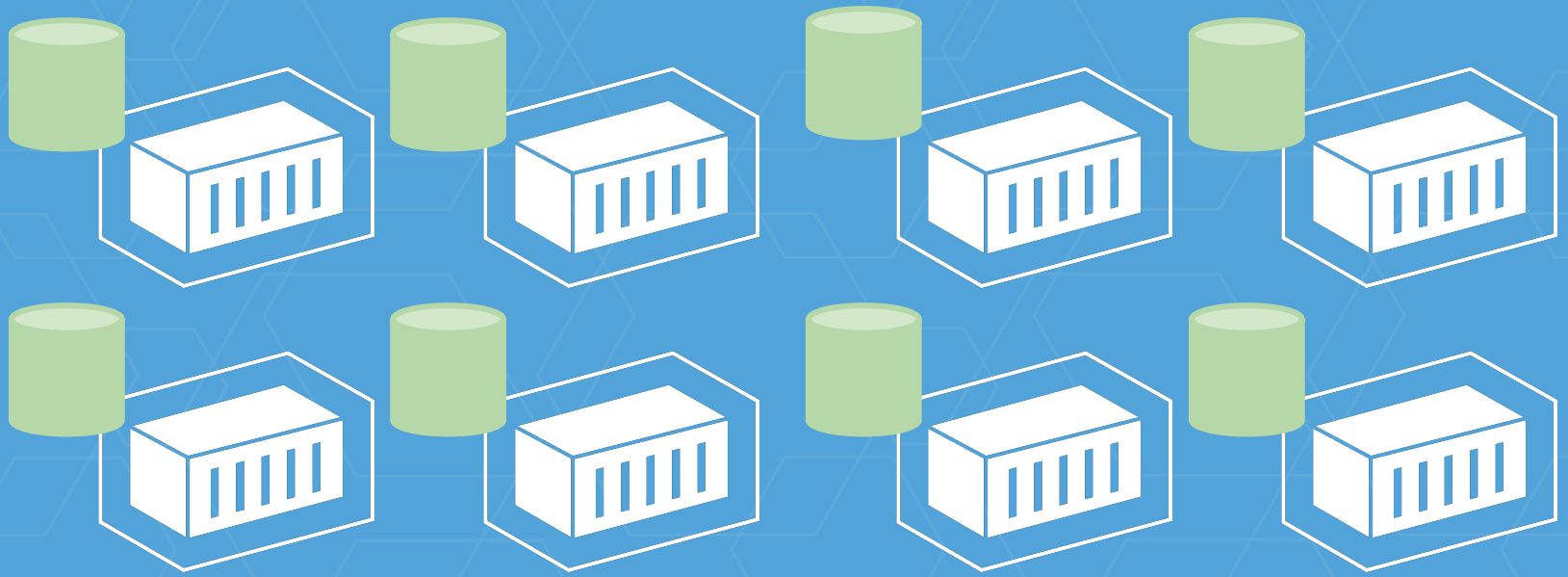
CoreOS

App Req/sec: **7,000**
App Healthy: **True**



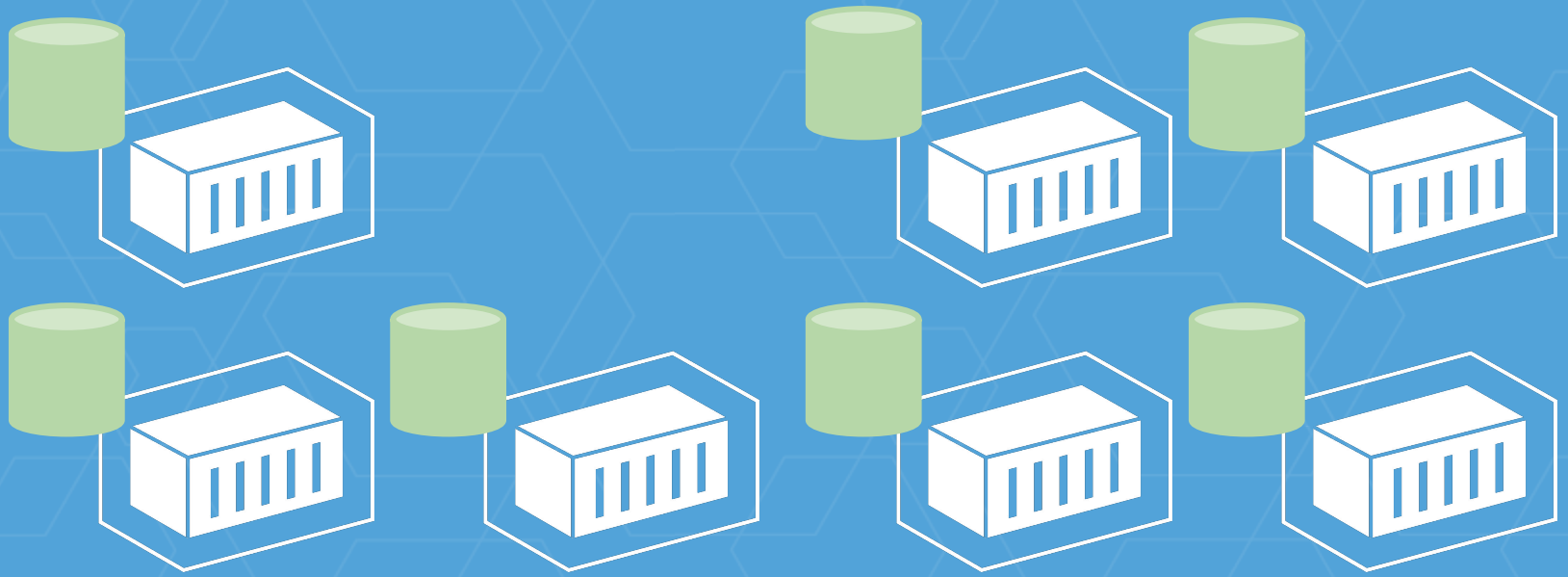
CoreOS

App Req/sec: **8,000**
App Healthy: **True**



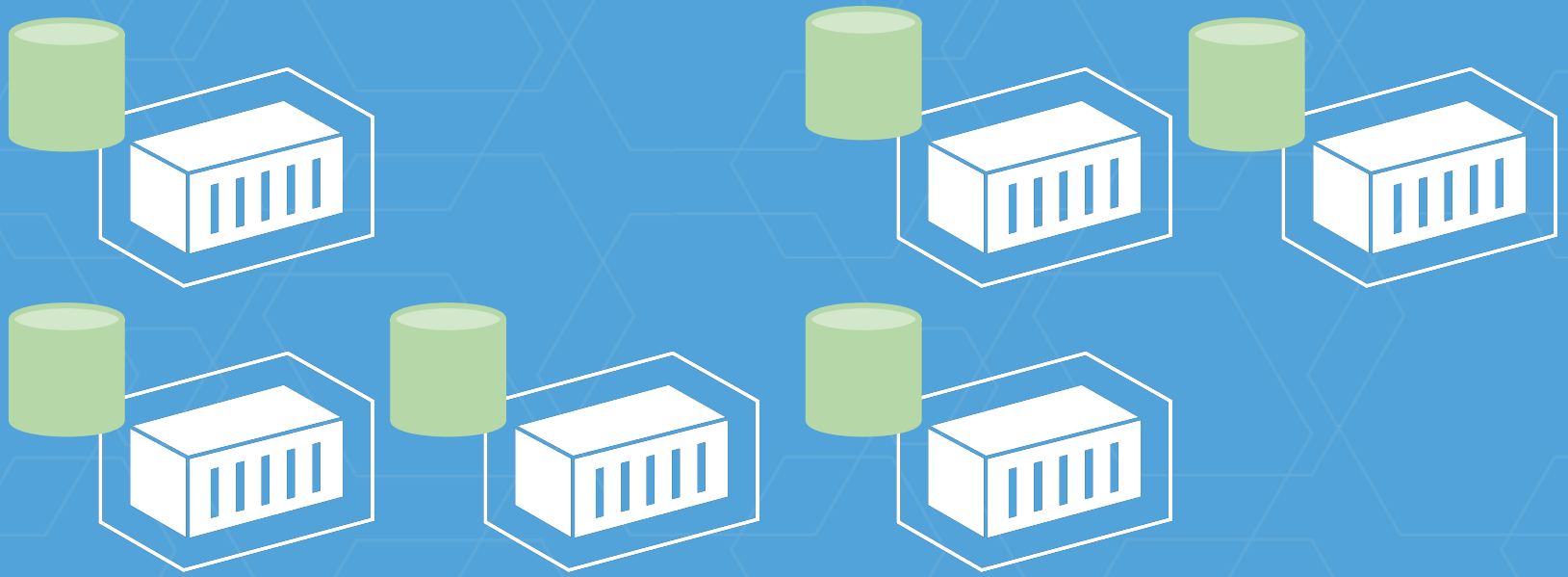
CoreOS

App Req/sec: **7,000**
App Healthy: **True**



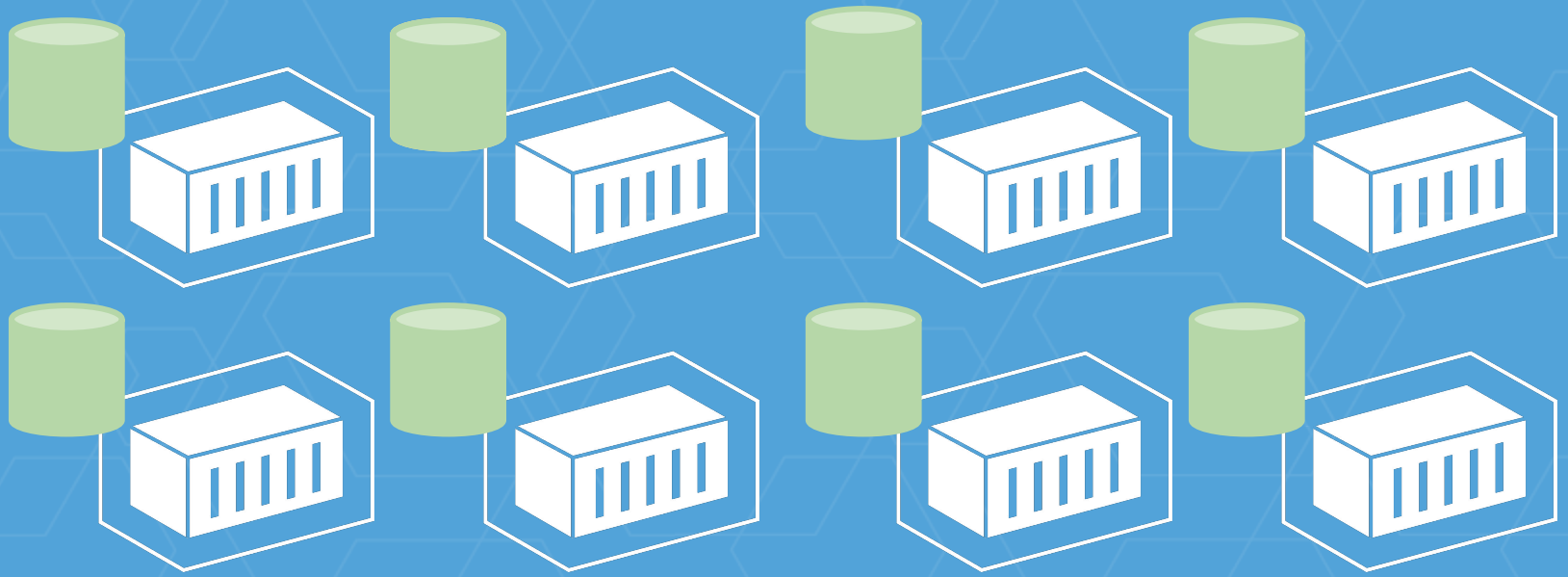
CoreOS

App Req/sec: **6,000**
App Healthy: **True**



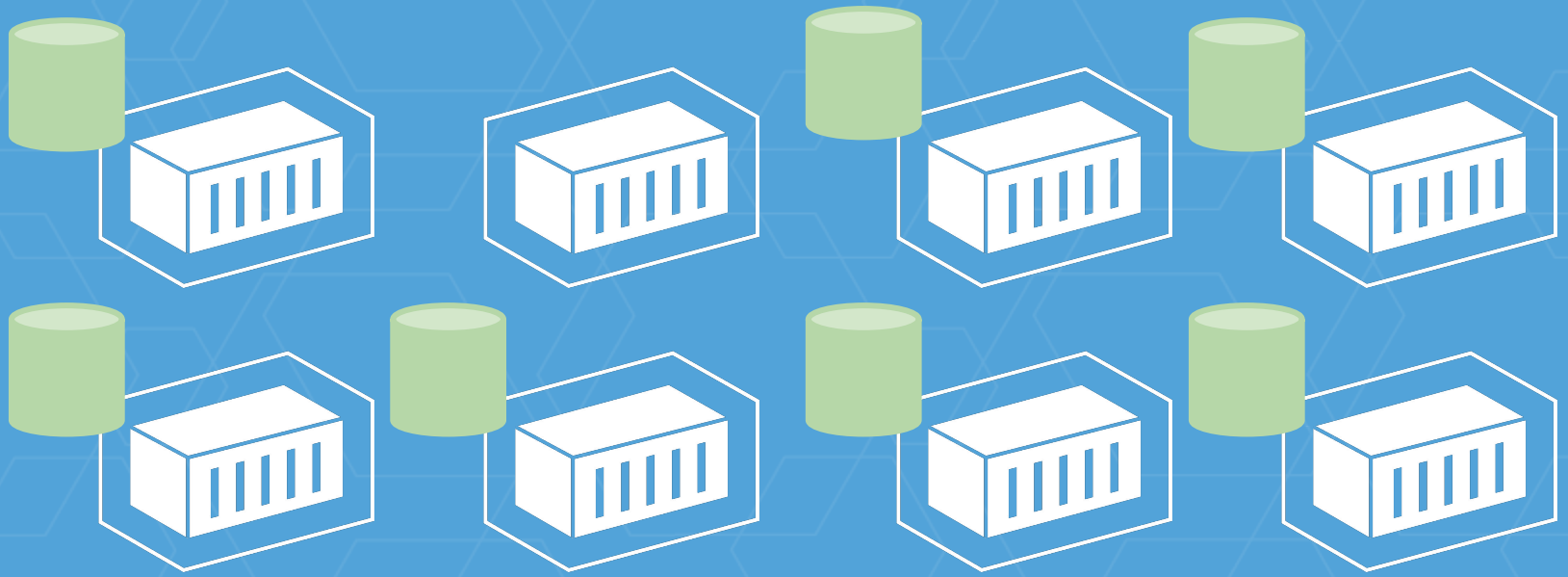
CoreOS

App Req/sec: **8,000**
App Healthy: **True**



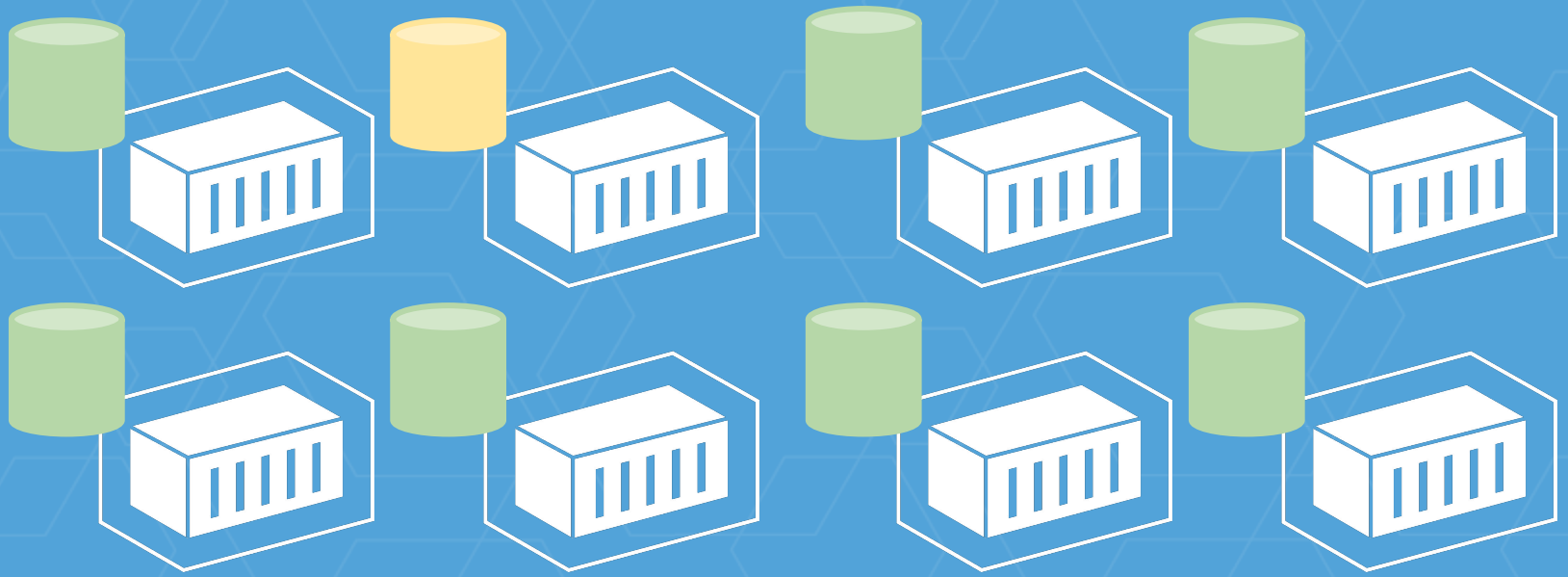
CoreOS

App Req/sec: **7,000**
App Healthy: **True**



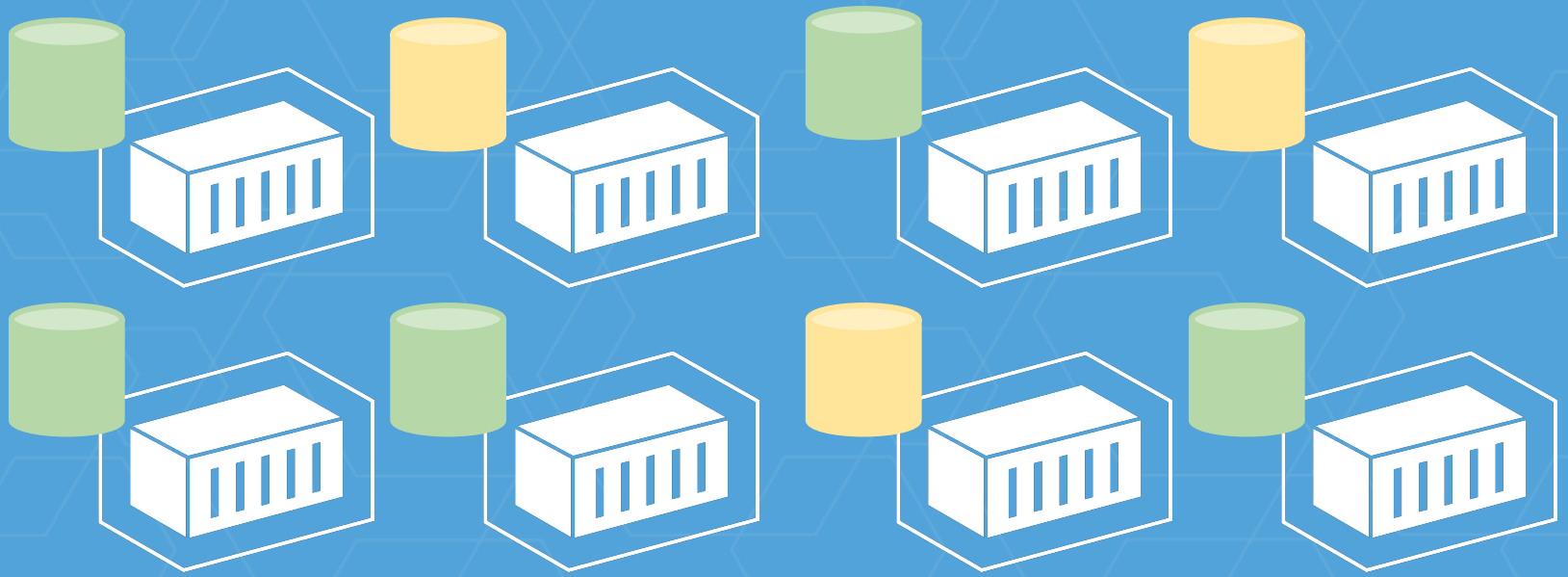
CoreOS

App Req/sec: **8,000**
App Healthy: **True**



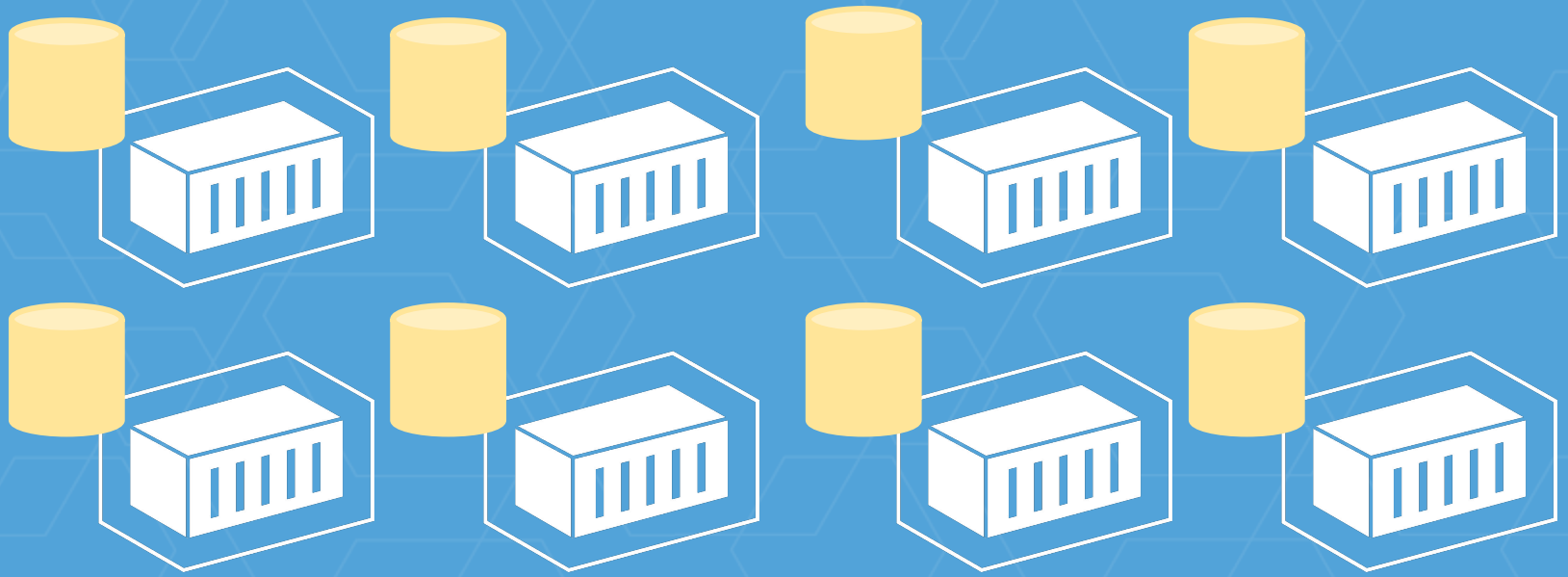
CoreOS

App Req/sec: **8,000**
App Healthy: **True**



CoreOS

App Req/sec: **8,000**
App Healthy: **True**



CoreOS

When do you need cluster coordination?

- Leader election
- Cluster-wide Semaphores
- Dynamic configuration
- Service discovery

CoreOS

Hard Computer Science Problem

CoreOS

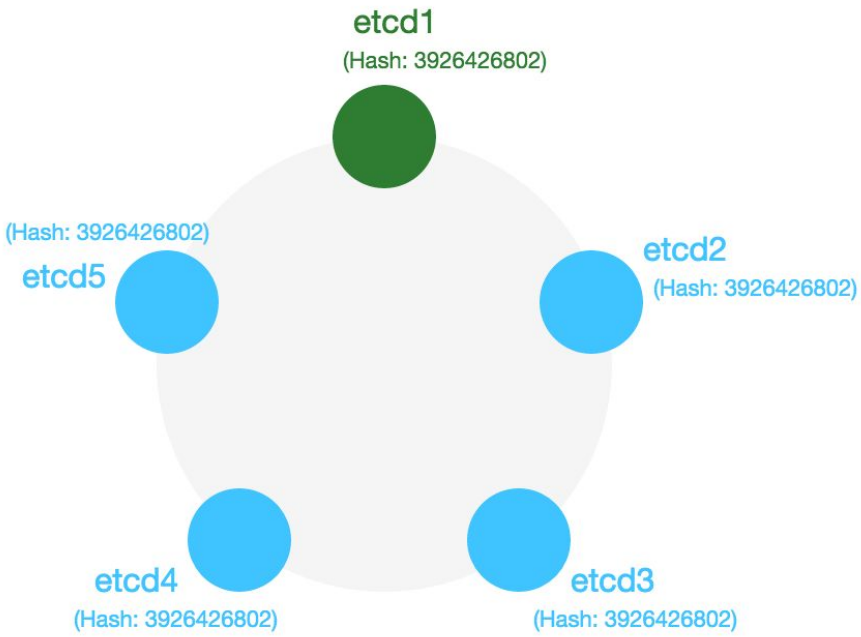
Hard Computer Science Problem

Chubby

CoreOS

A highly-available key value store for shared configuration and service discovery

Play etcd Home Code Issues



etcd1 etcd2 etcd3 etcd4 etcd5

PUT GET DELETE Submit

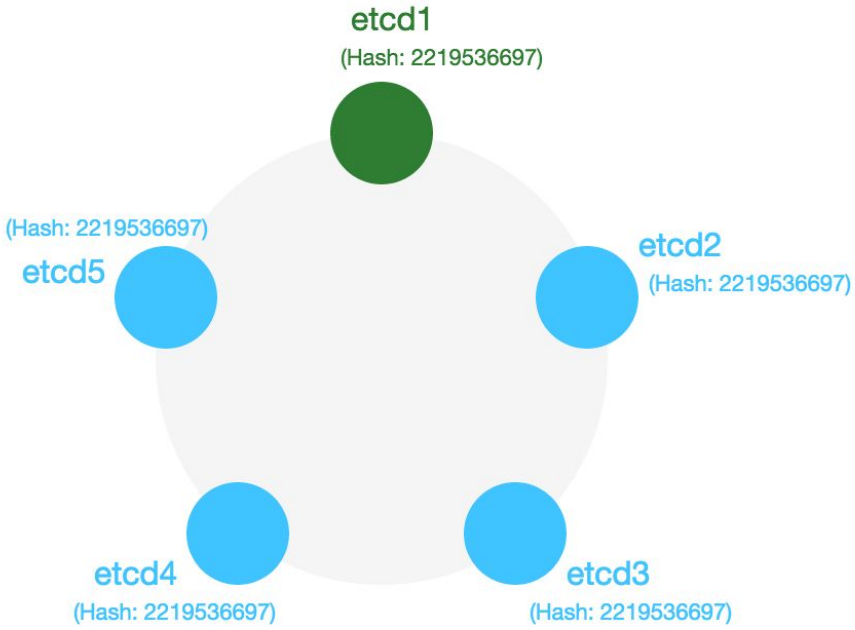
foo

BAZ

[PUT] success! "foo" : "BAZ" (at 2016-03-16 05:36:42 PST)

Log

Play etcd Home Code Issues



etcd1 etcd2 etcd3 etcd4 etcd5

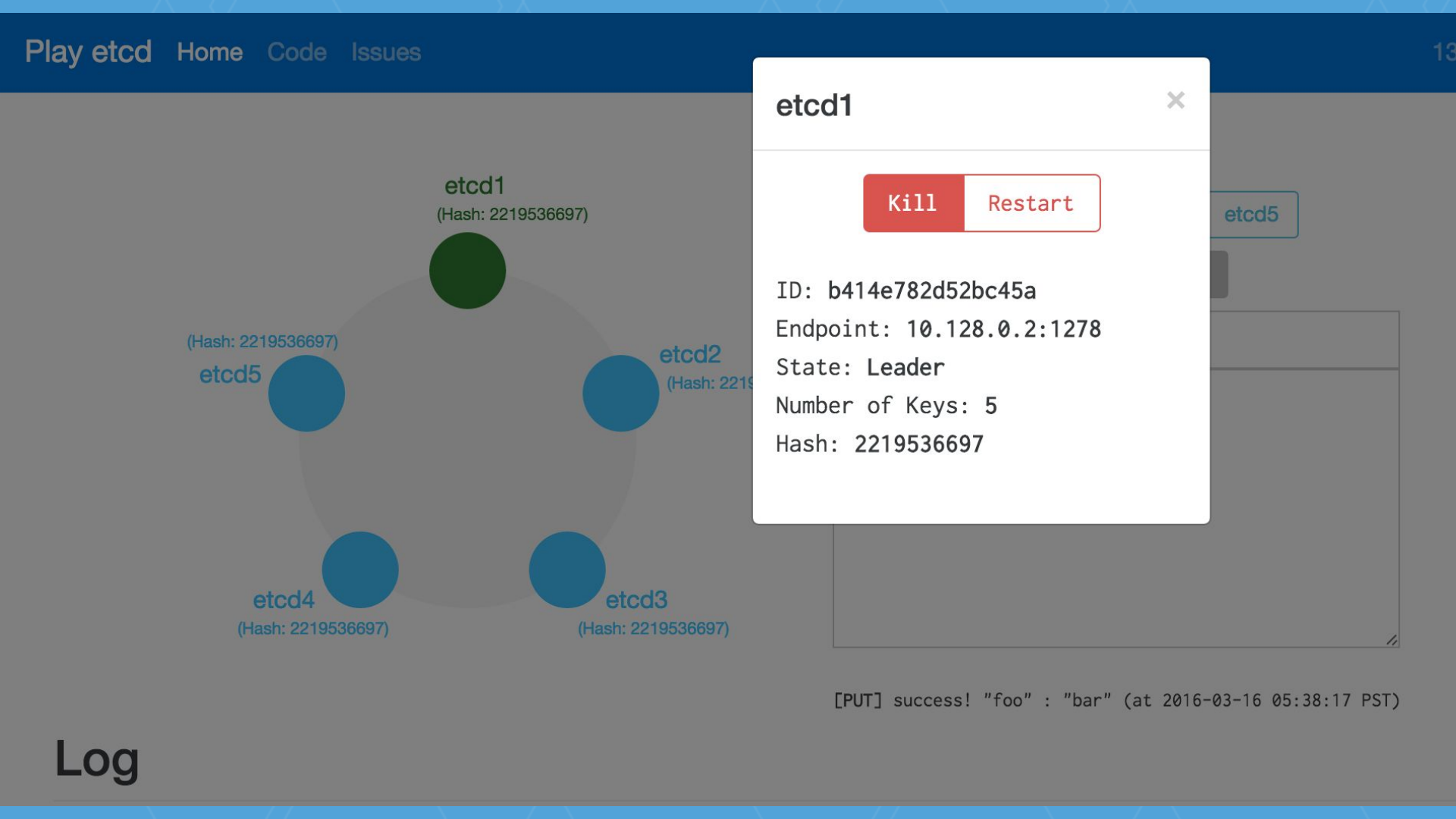
PUT GET DELETE Submit

foo

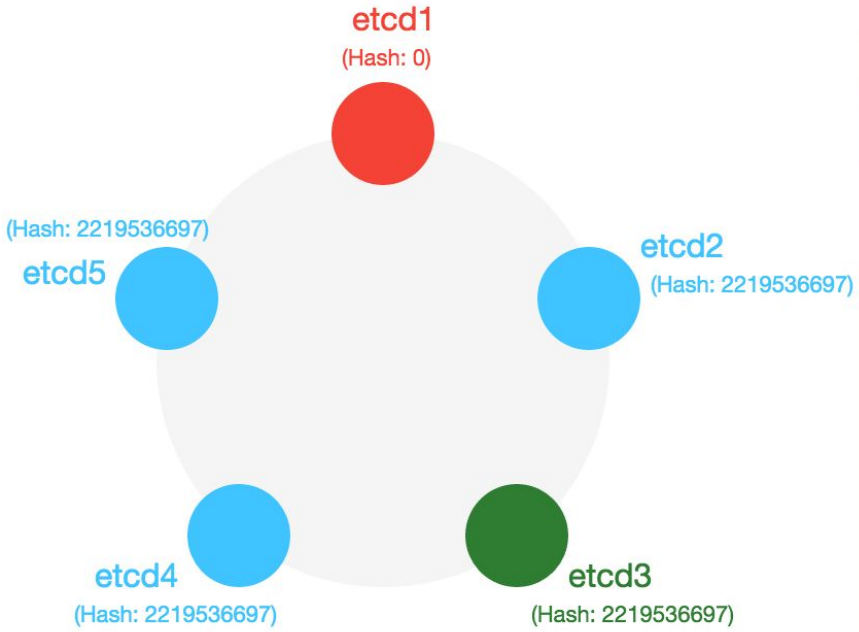
bar

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

Log


Play etcd
Home
Code
Issues
etcd1
(Hash: 2219536697)
etcd2
(Hash: 221
(Hash: 2219536697)
etcd5
etcd4
(Hash: 2219536697)
etcd3
(Hash: 2219536697)
etcd1
Kill
Restart
ID: b414e782d52bc45a
Endpoint: 10.128.0.2:1278
State: Leader
Number of Keys: 5
Hash: 2219536697
etcd5
[PUT] success! "foo" : "bar" (at 2016-03-16 05:38:17 PST)
Log

Play etcd Home Code Issues



etcd1 etcd2 etcd3 etcd4 etcd5

PUT GET DELETE Submit

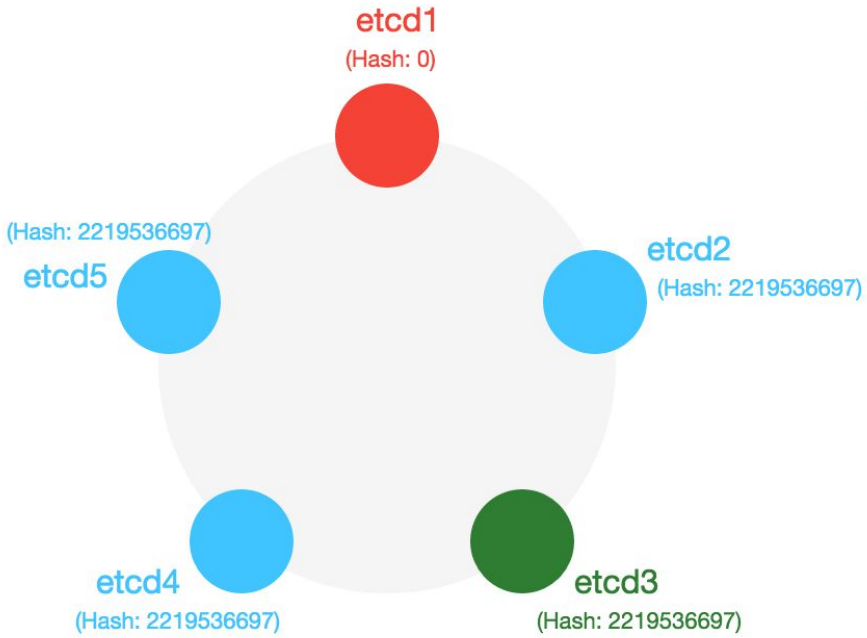
foo

bar

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

Log

Play etcd Home Code Issues



etcd1 etcd2 etcd3 etcd4 etcd5

PUT GET DELETE Submit

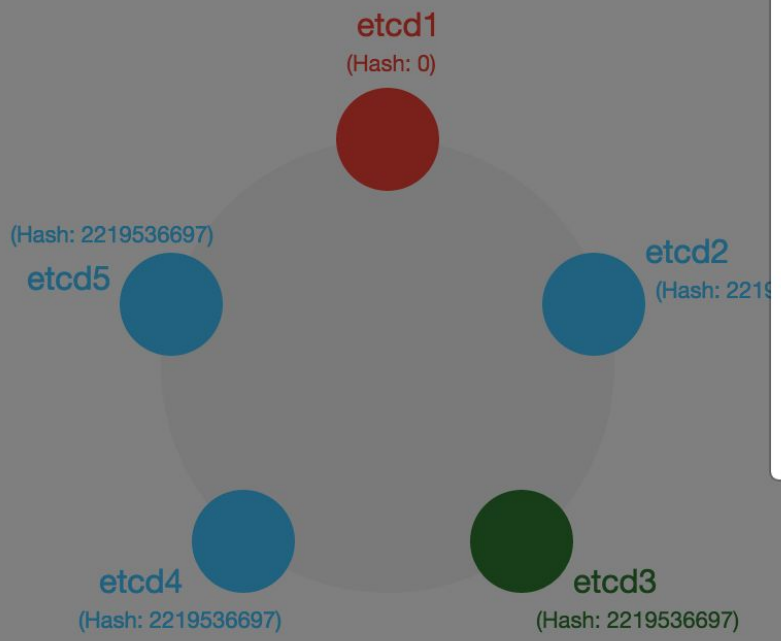
foo

BAZ

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

Log

Play etcd Home Code Issues



etcd1

Kill Restart

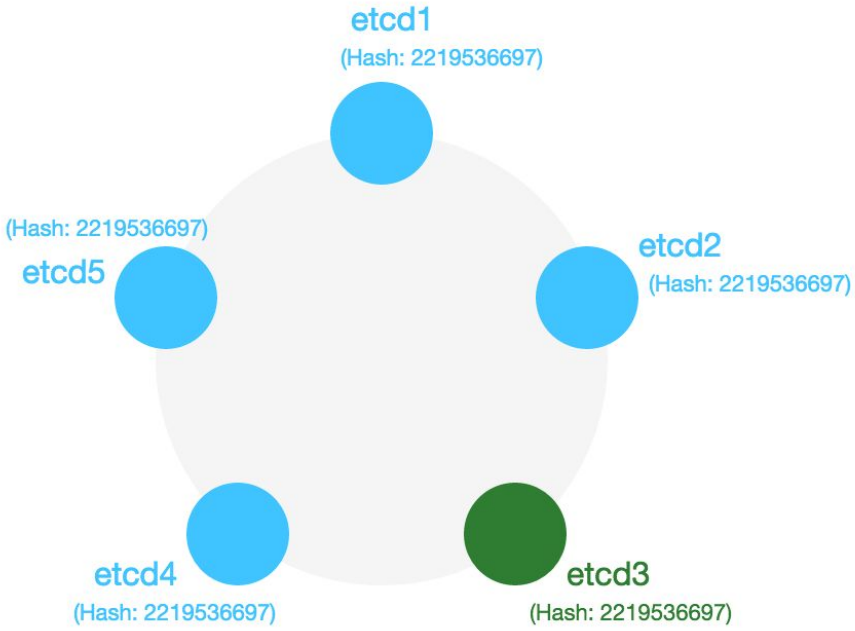
ID: unknown
Endpoint: 10.128.0.2:1278
State: unreachable
Number of Keys: 0
Hash: 0

etcd5

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

Log

Play etcd Home Code Issues



etcd1 | etcd2 | etcd3 | etcd4 | etcd5

PUT | GET | DELETE | Submit

foo

BAZ

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

Log

Why build etcd?

- ✖ No existing “cloud native” solutions
- ✚ Simple HTTP + JSON APIs
- 🔄 Dynamic reconfiguration

Simple key/value

- “Distributed etc”
- Keys are versioned
- Changes can be watch

CoreOS

- Cluster-wide reboot lock - *locksmith*
- Service discovery - *vulcand, skydns*
- Cluster orchestration - *k8s, cloud foundry*

CoreOS

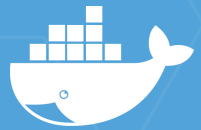
Industry Adoption

kubernetes

CLOUD FOUNDRY

MESOS

DEIS



docker

500+ projects on Github

CoreOS

When do you need cluster coordination?

- Leader election
- Cluster-wide Semaphores
- Dynamic configuration
- Service discovery

CoreOS

Hard Computer Science Problem

CoreOS

Hard Computer Science Problem

Chubby

CoreOS

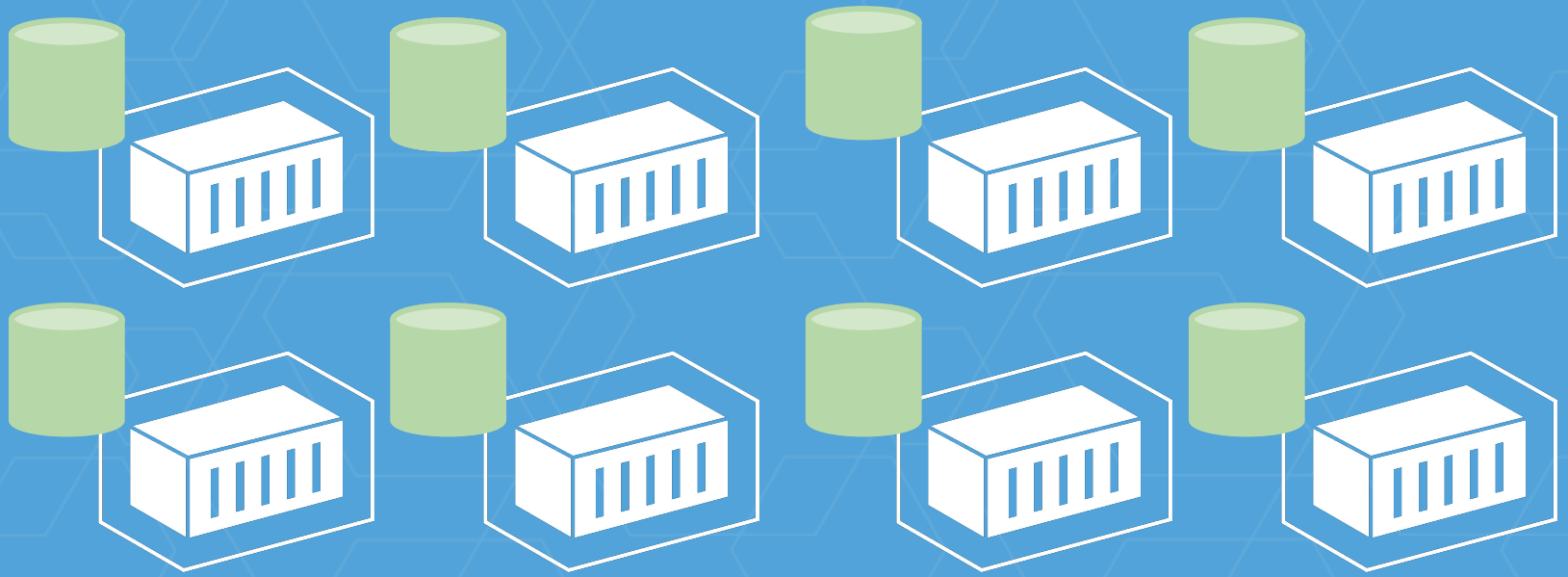
A highly-available key value store for shared configuration and service discovery

kubernetes

Kubernetes is our choice for orchestration platform

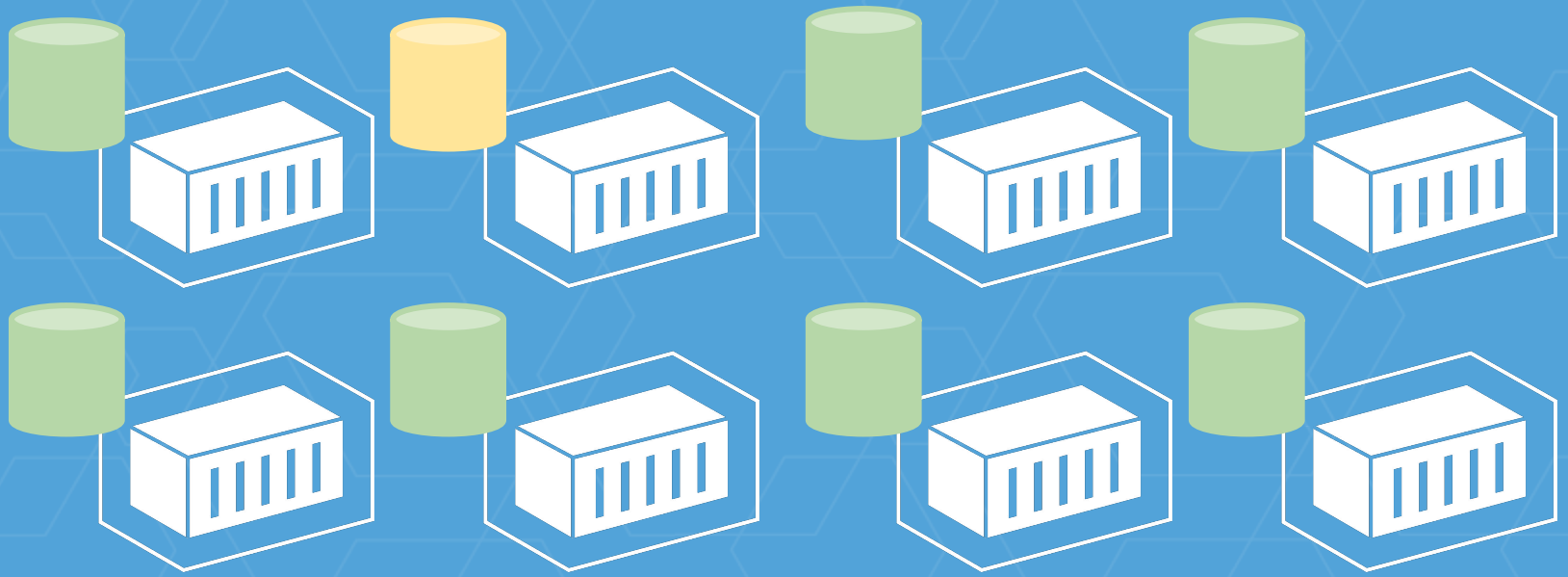
CoreOS

App Req/sec: **8,000**
App Healthy: **True**



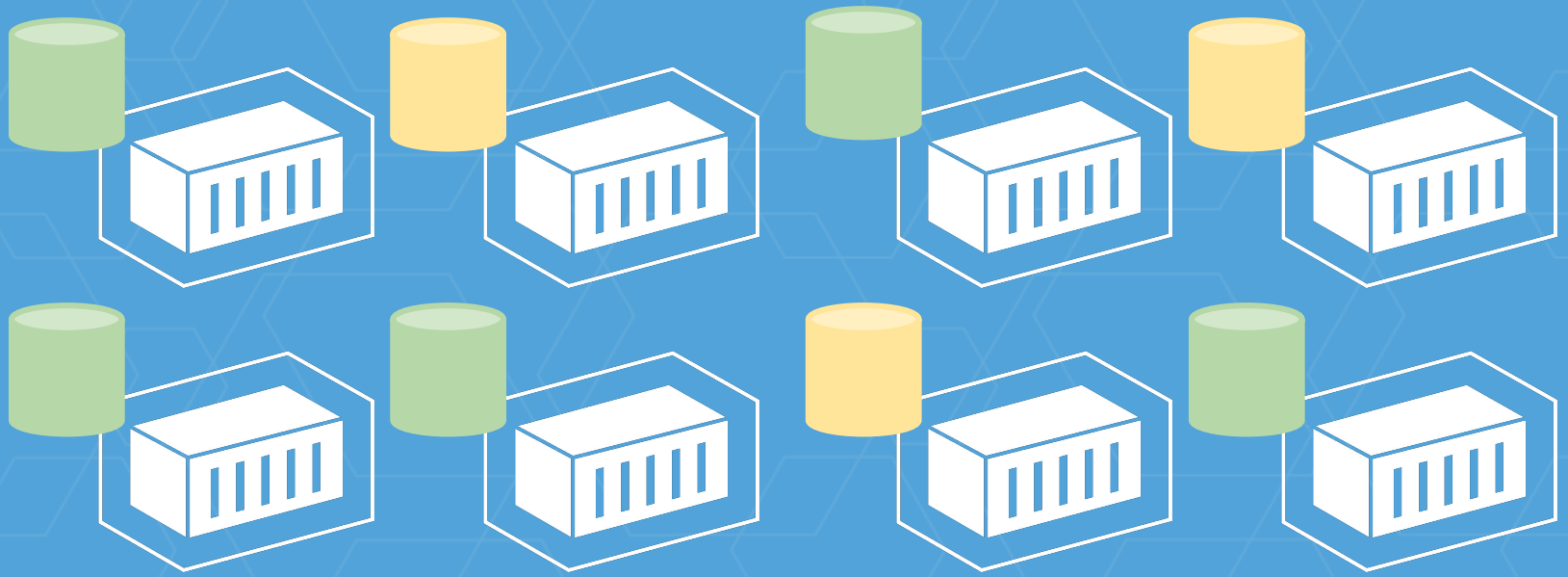
CoreOS

App Req/sec: **8,000**
App Healthy: **True**



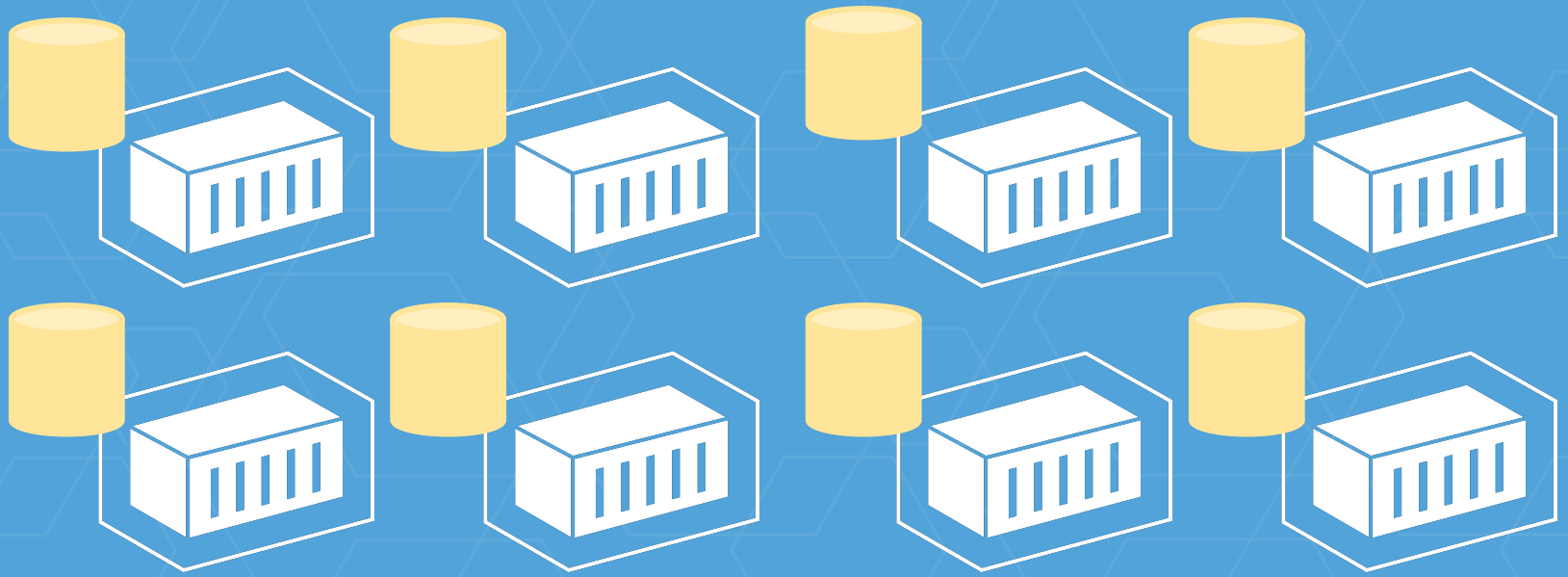
CoreOS

App Req/sec: **8,000**
App Healthy: **True**



CoreOS

App Req/sec: **8,000**
App Healthy: **True**



Guides & Tools

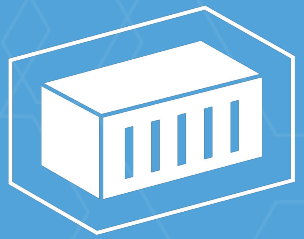
- coreos.com/kubernetes
- kube-aws
- Cloud-configs

kubernetes

Scheduler gets work to the servers

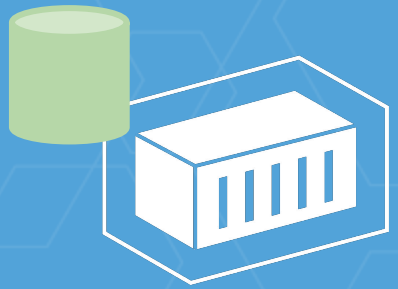
CoreOS

```
$ scp app host:/opt
$ ssh host systemd-run /opt/app
```



CoreOS

```
$ scp app host:/opt
$ ssh host systemd-run /opt/app
```



CoreOS

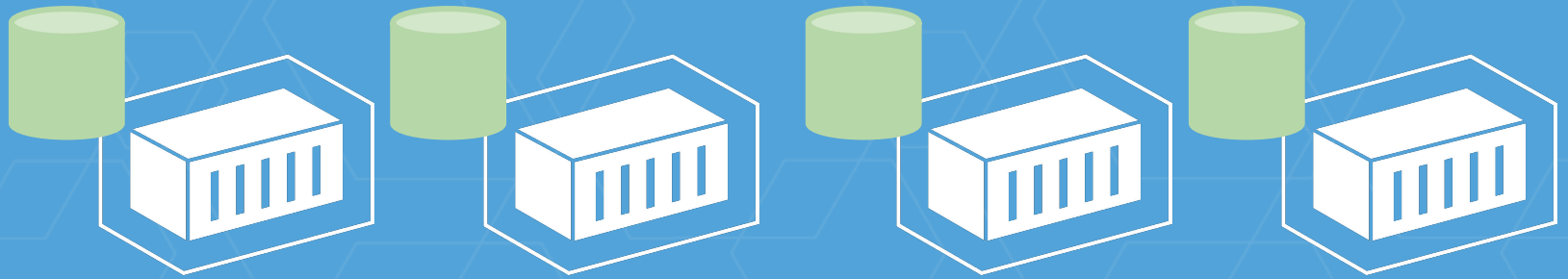
$ fab deploy:app

CoreOS

$ fab deploy:app

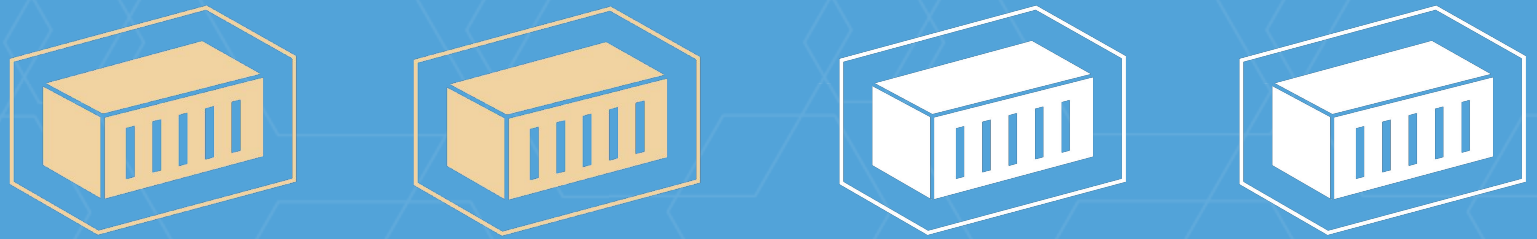
CoreOS

$ fab deploy:app



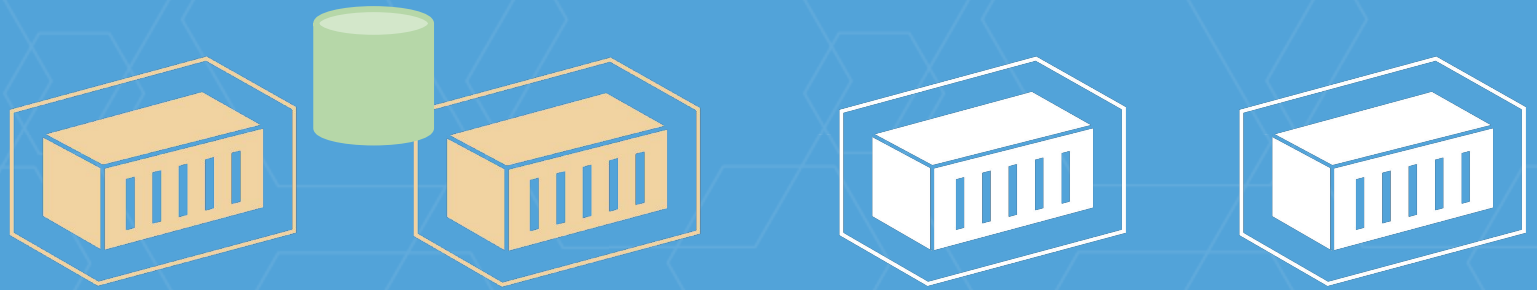
CoreOS

$ fab deploy:gpu-app



CoreOS

$ fab deploy:gpu-app



CoreOS

$ fab deploy:gpu-app

kubernetes

Replication controllers drive the cluster to a desired state

rc web-prod
select(env=prod,app=web)
count=1

pod
env=prod
app=web

rc web-prod
select(env=prod,app=web)
count=1

pod
env=prod
app=web

rc web-prod
select(env=prod,app=web)
count=1

pod
env=prod
app=web

rc web-prod
select(env=prod,app=web)
count=5

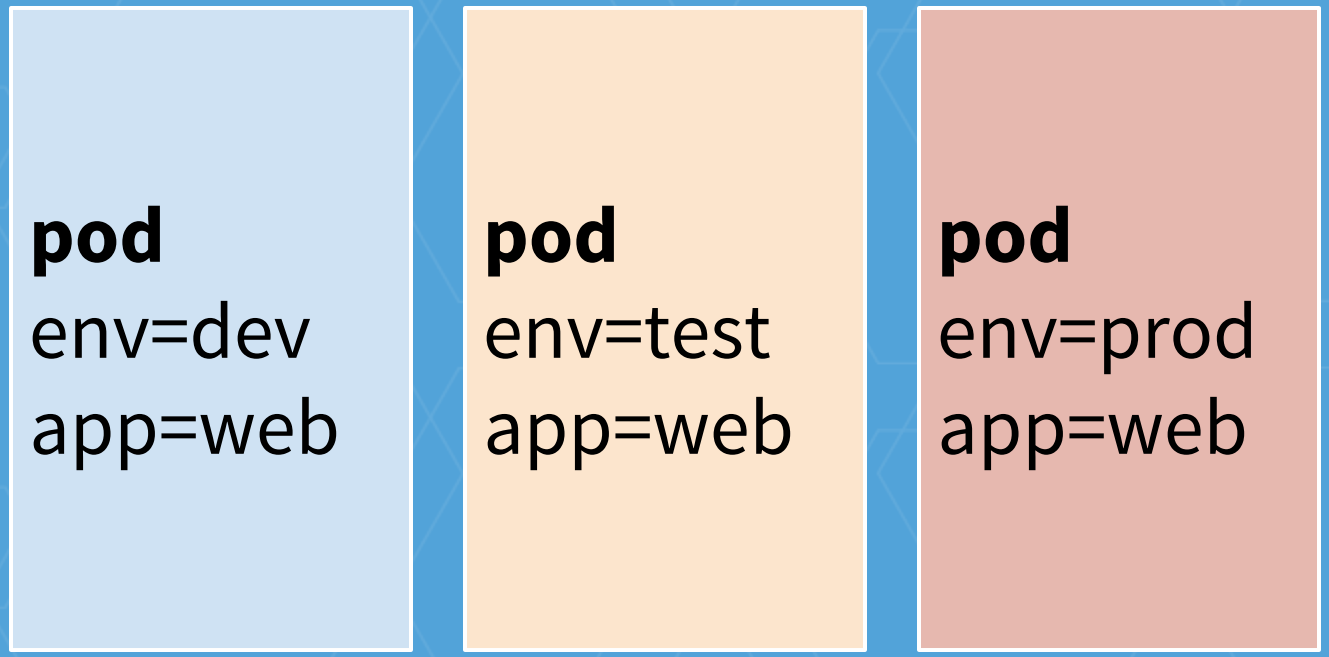
pod
env=prod
app=web

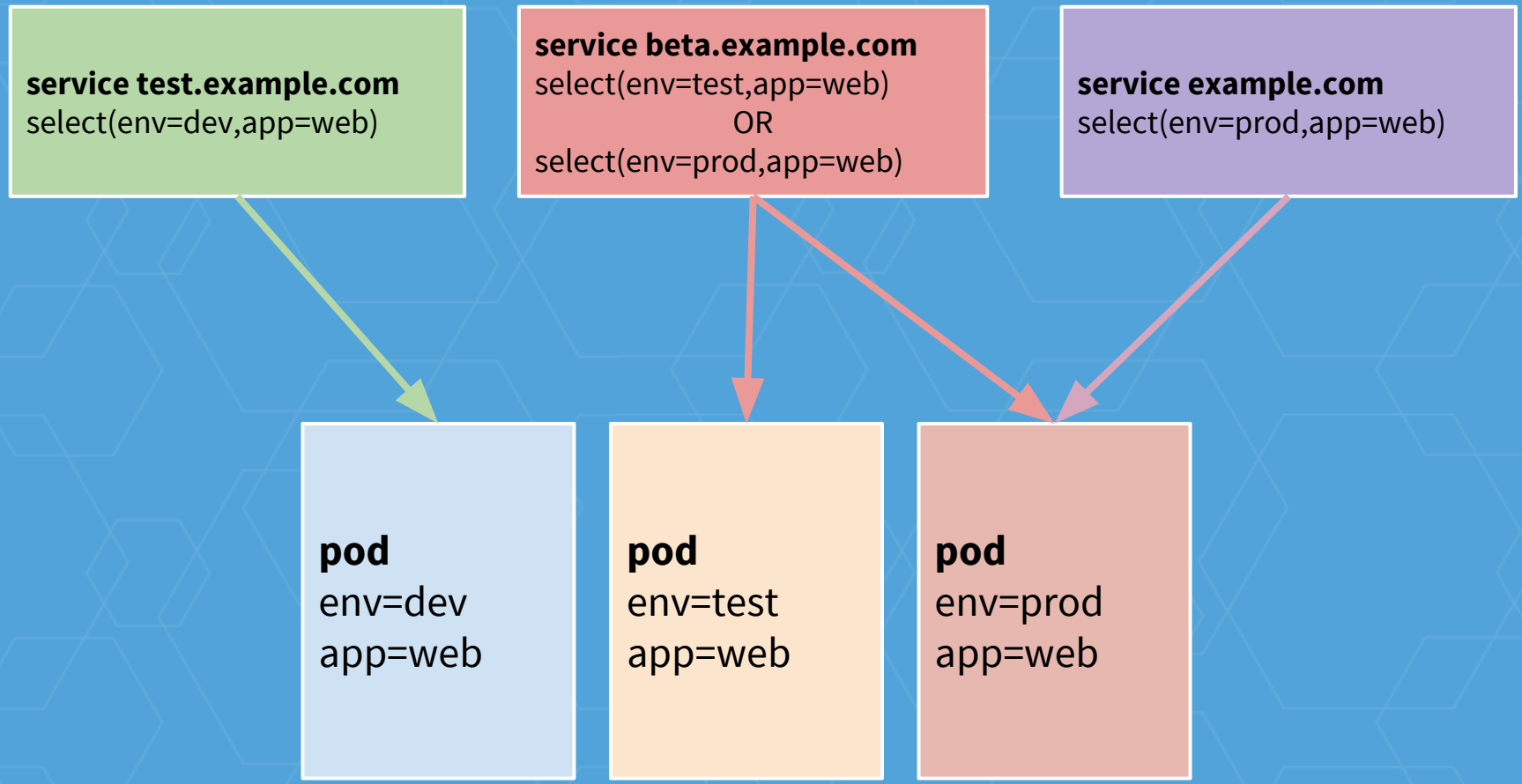
rc web-prod
select(env=prod,app=web)
count=5

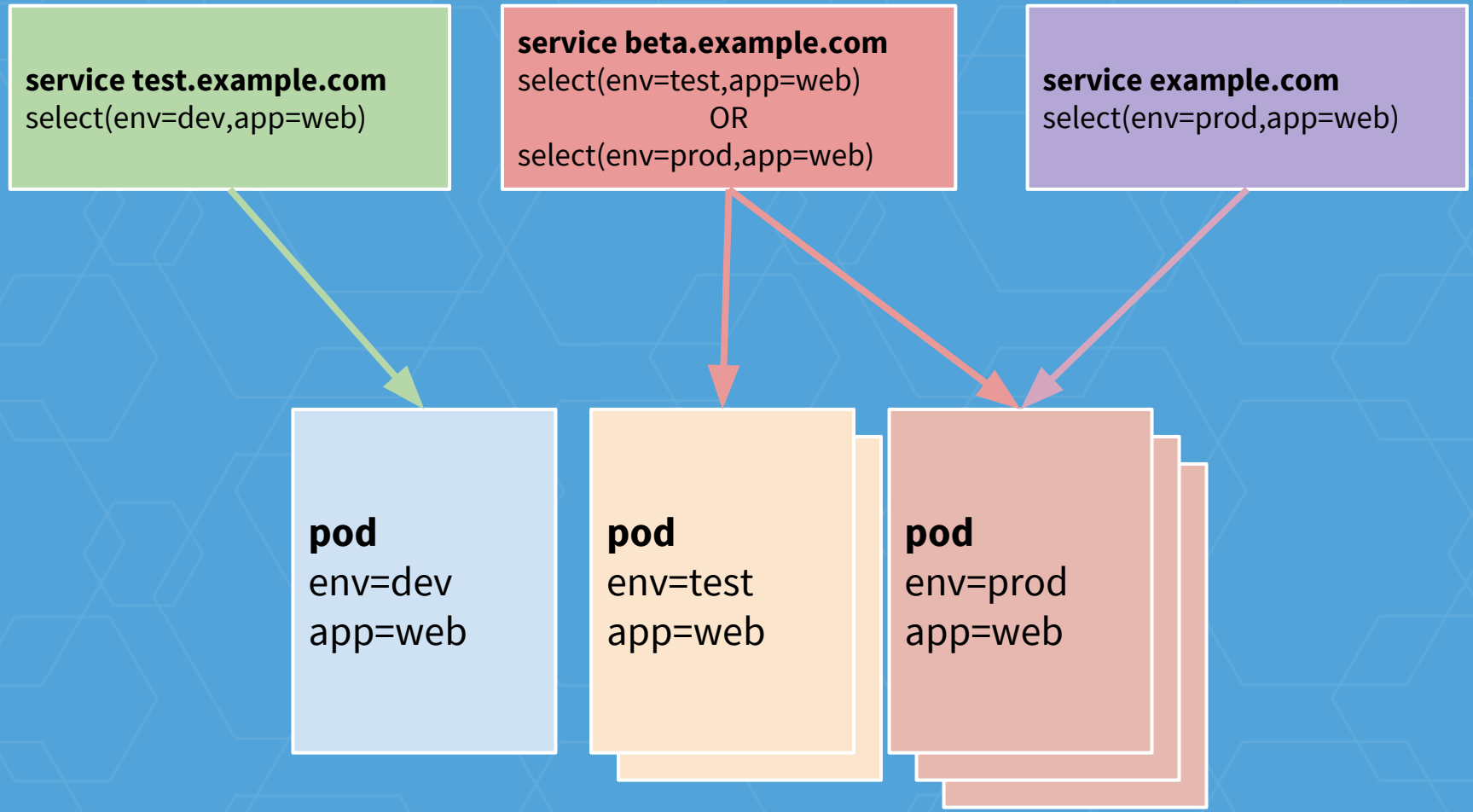
pod
env=prod
app=web

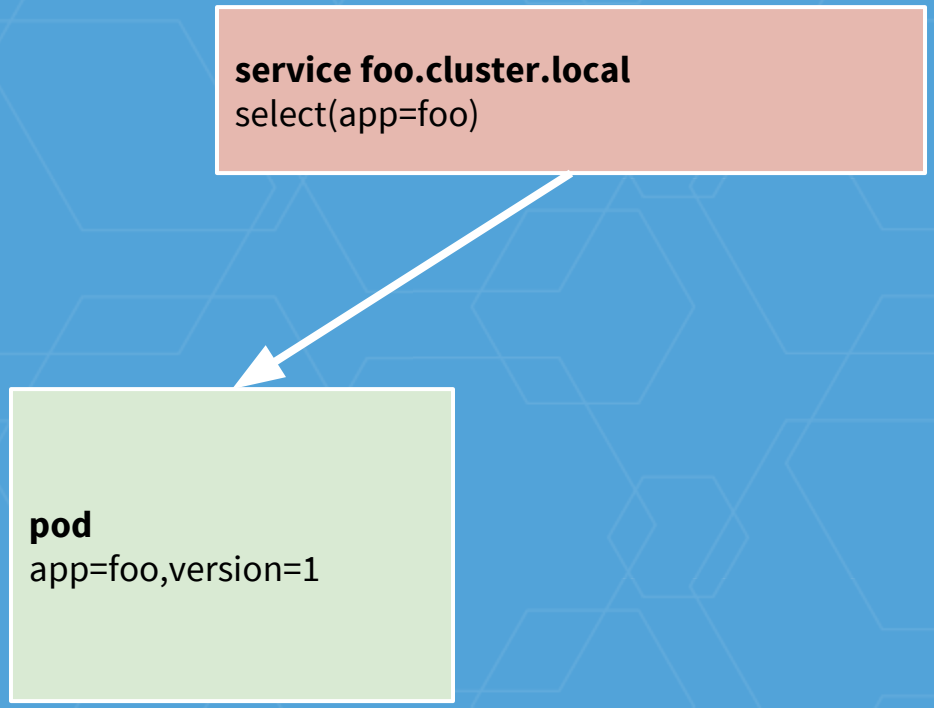
kubernetes

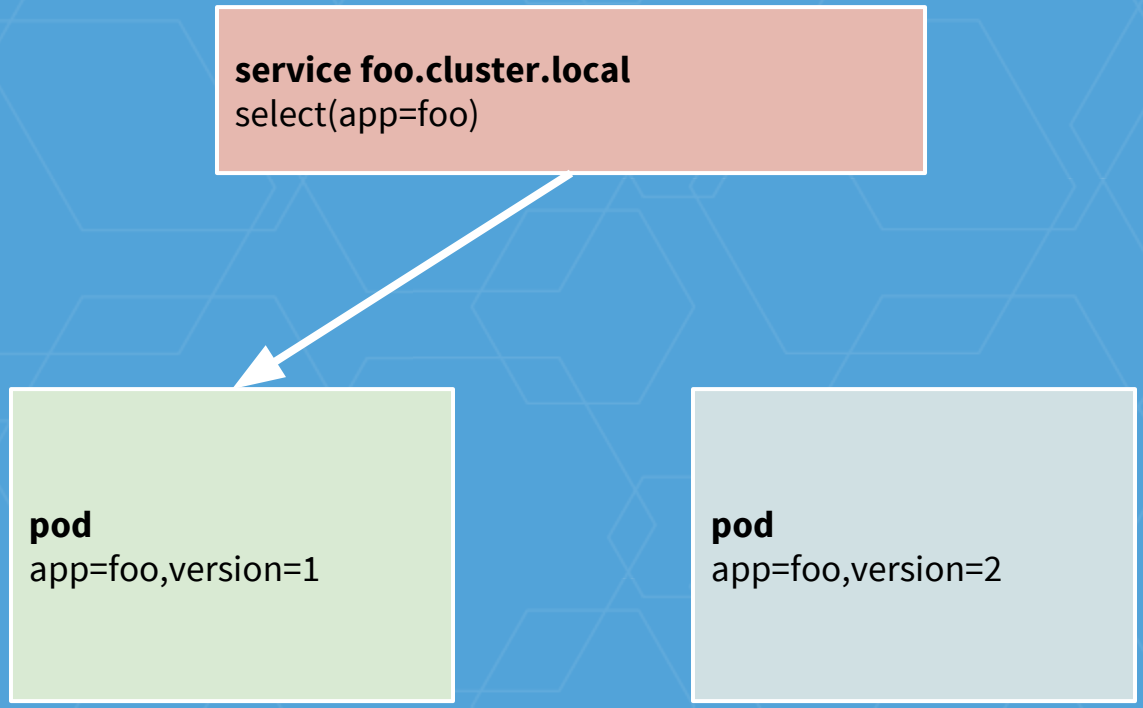
Services find scheduled apps (pods) and load balance them

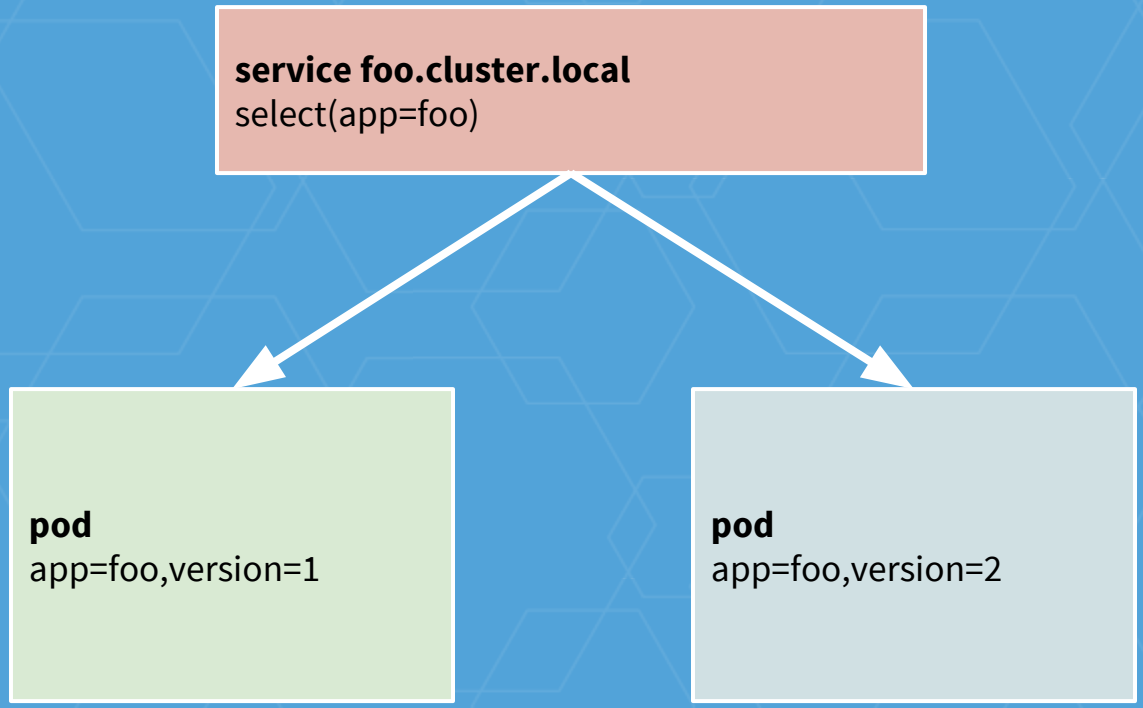

pod
env=dev
app=web
pod
env=test
app=web
pod
env=prod
app=web


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


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


service foo.cluster.local
select(app=foo)
pod
app=foo,version=1


service foo.cluster.local
select(app=foo)
pod
app=foo,version=1
pod
app=foo,version=2

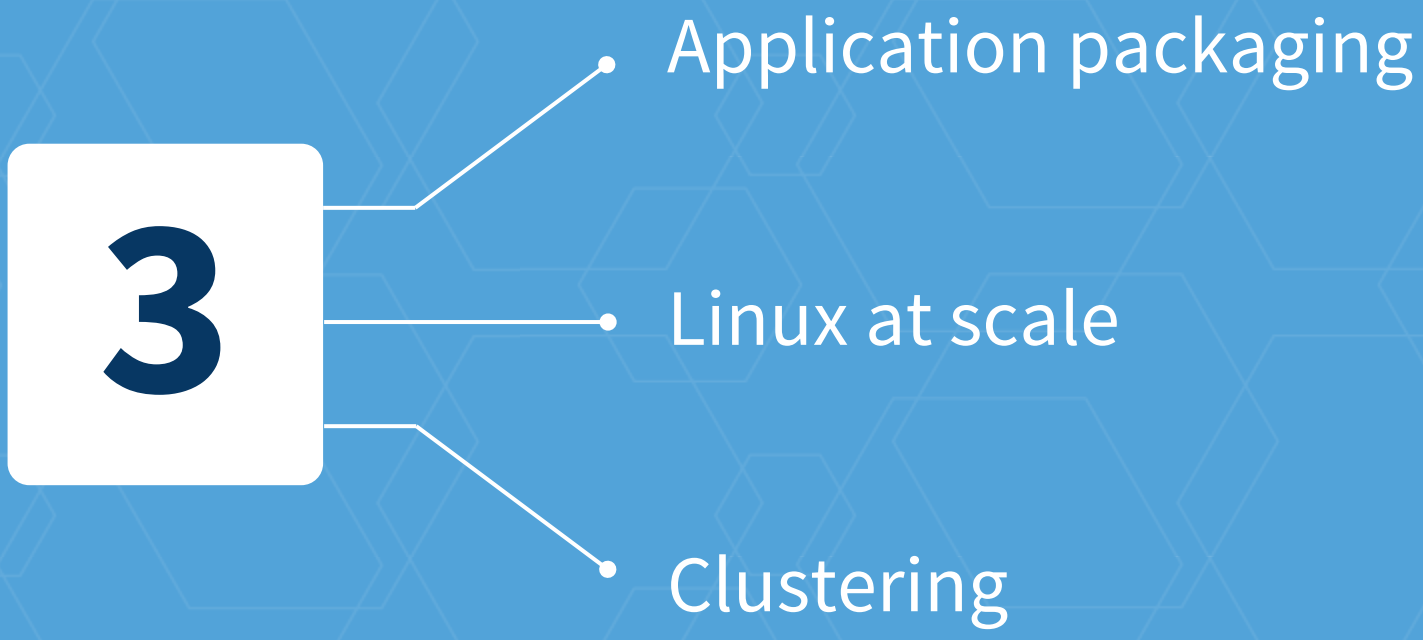

service foo.cluster.local
select(app=foo)
pod
app=foo,version=1
pod
app=foo,version=2

Upstream

- rktnetes
- Auth/OIDC
- Node self-signed TLS

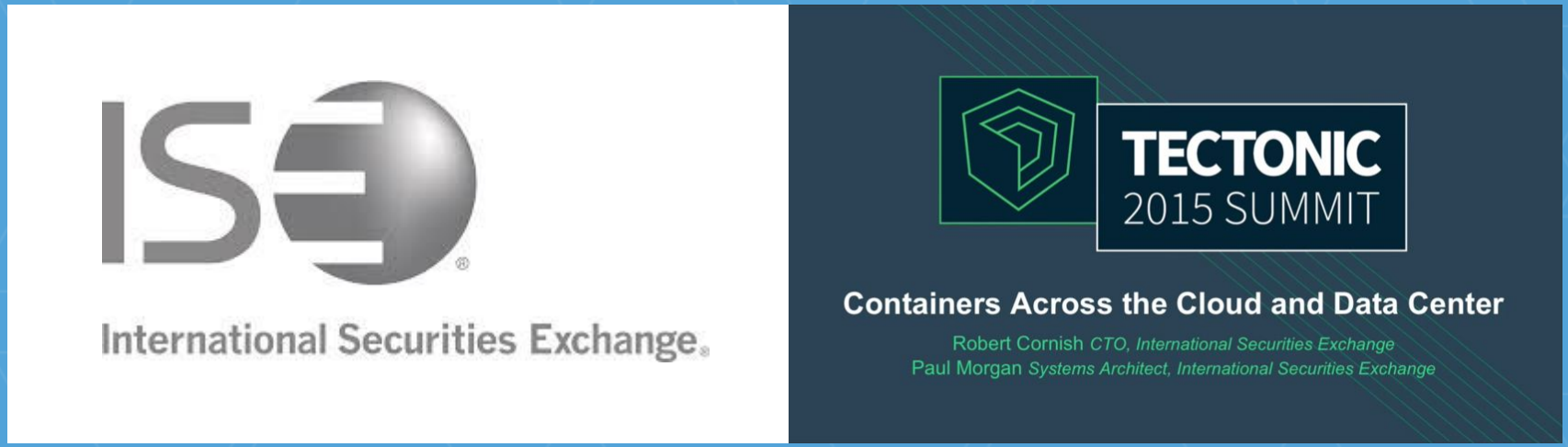
Scaling

- 15x scheduler performance
- 30k pods on 1k nodes
- SIG-scale


3
Application packaging
Linux at scale
Clustering

Sounds good, but...

Is anyone successful with all this in prod?



Publically traded options exchange

CoreOS

Containers on CoreOS are powering ISE's high-throughput, low-latency financial exchange

- Running in production
- Bare metal & AWS
- Billions of transactions a day
- 150 million req/sec


viacom


ca
technologies


salesforce


verizon

CoreOS

MISSION

Secure the Internet

CoreOS

STRATEGY

Separate Apps from OS

CoreOS

STRATEGY

Make Servers Consistent

CoreOS

STRATEGY

Tolerate Machine Failures

CoreOS

STRATEGY

Make Servers Easy to Upgrade

CoreOS

STRATEGY

Simplify Application Upgrades


17:34 / 18:00
CC
HD

May 9 & 10, 2016 - Berlin, Germany

coreos.com/fest - @coreosfest


TECTONIC
by CoreOS

Build, Store and Distribute your Containers

quay.io

Brandon Philips

@brandonphilips | brandon@coreos.com | coreos.com

Thank you!

We're hiring in all departments! Email: careers@coreos.com Positions: coreos.com/ careers