CoreOS for SysAdmins

Alex Crawford

Software Developer at CoreOS

alex.crawford@coreos.com

github.com/crawford

What is CoreOS?

What is CoreOS?

amazon
web services™

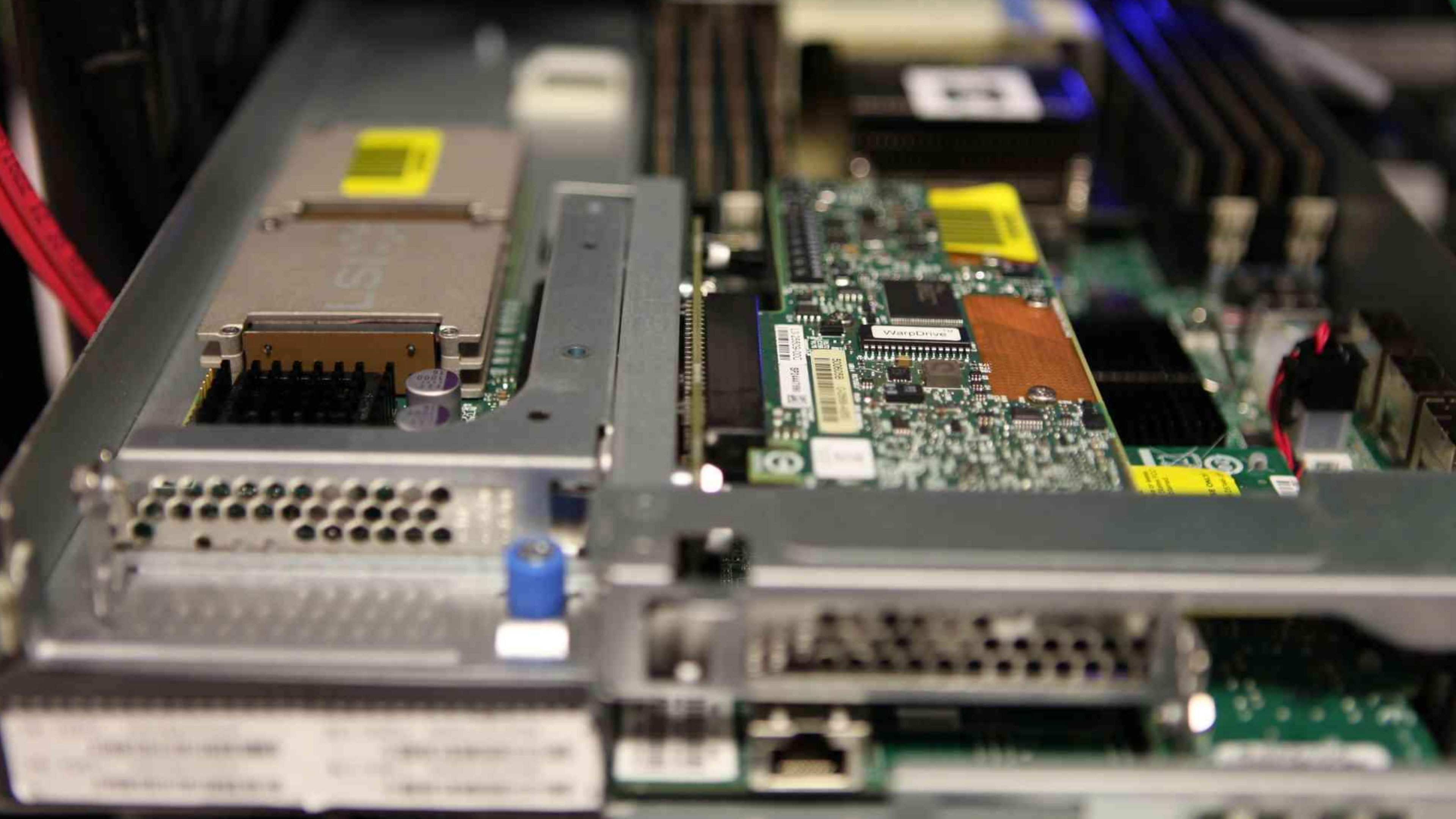
Google Cloud Platform

openstack®
CLOUD SOFTWARE

DigitalOcean

vmware®

Windows Azure


LSI
WarpDrive™

What is CoreOS?

What is CoreOS?

etcd

flannel

rkt

etcd
4,961 total commits
6 repositories
38 releases
13 CoreOS contributors
527 community contributors
Top 100 Community Contributors
Core OS
Release Frequency
Alpha
Beta
Stable
Issues
16 New Issues
18 Closed Issues
10k+ total commits
~30 repositories
324 releases
22 CoreOS contributors
605 community contributors
Top 100 Community Contributors

TECTONIC

by CoreOS

The smartest way to run your container infrastructure.

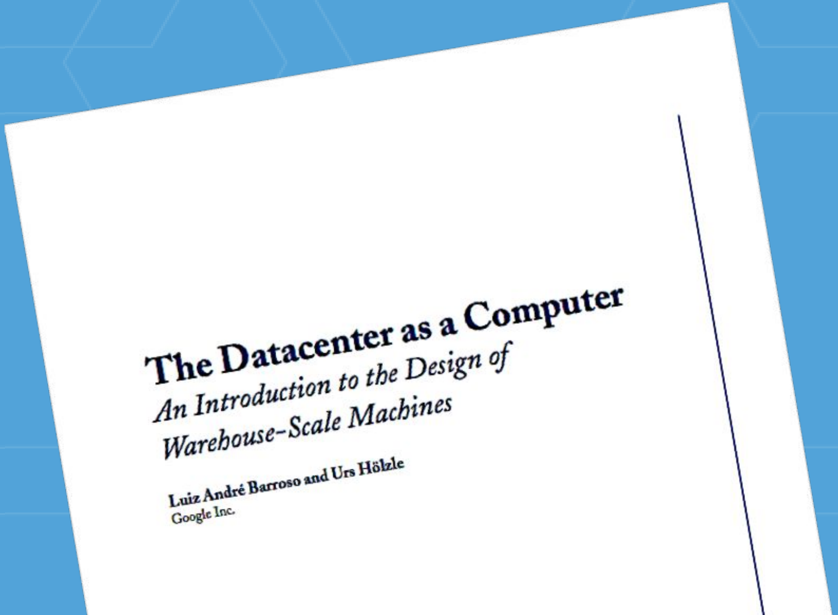
tectonic.com @tectonic

QUAY

Secure hosting for private container repositories

quay.io @quayio

Why build CoreOS?



The Datacenter as a Computer

An Introduction to the Design of Warehouse-Scale Machines

Luiz André Barroso and Urs Hölzle

Google Inc.

you

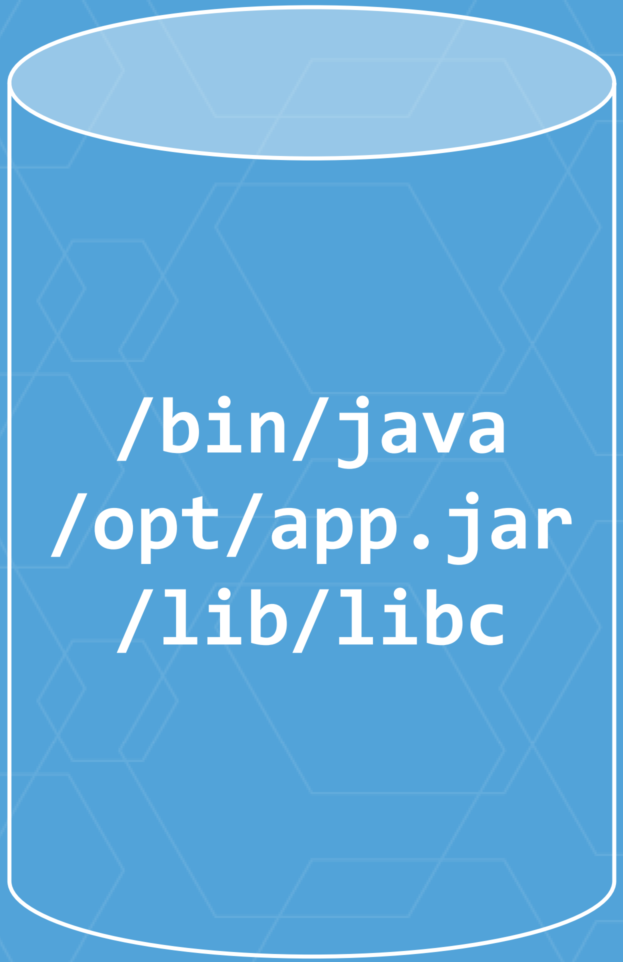
you as a sw engineer

your

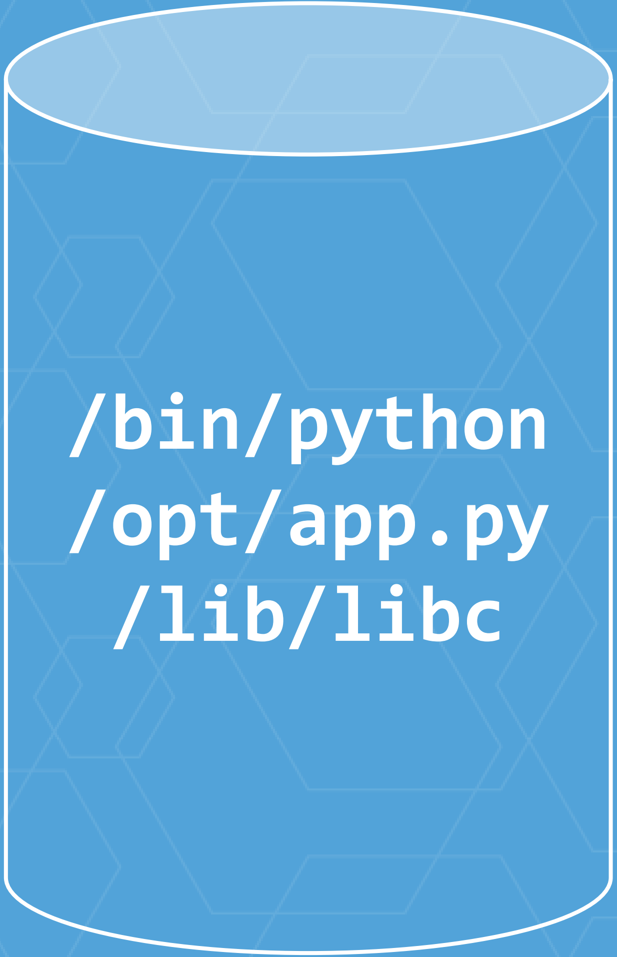
```
with Ada.Text_IO;

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

your



your



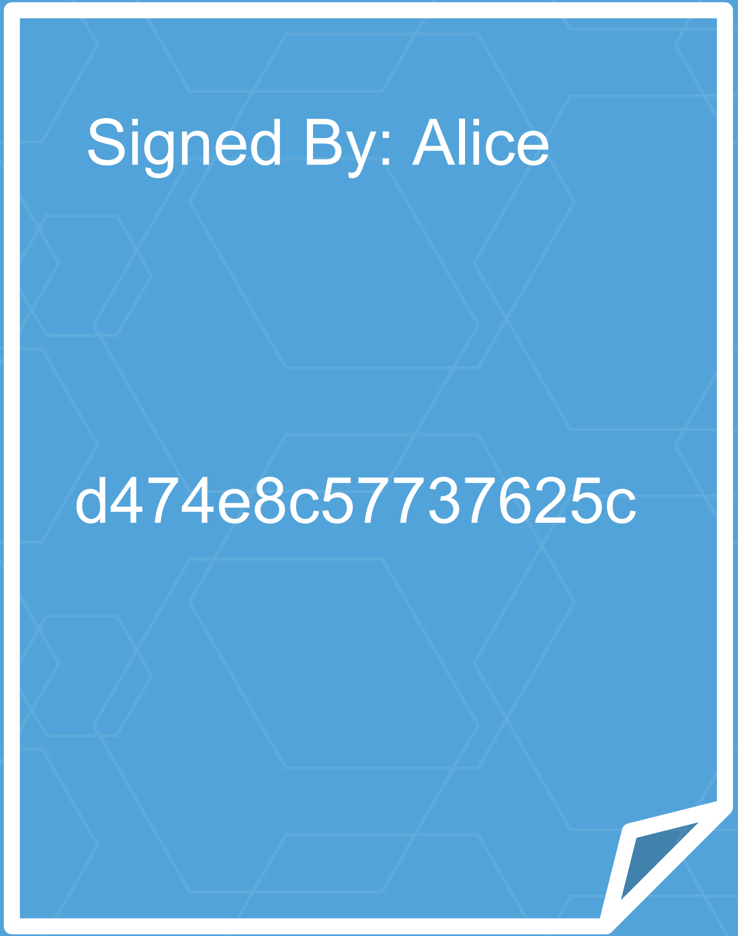
```
/bin/python
/opt/app.py
/lib/libc
```

your

com.example.app

d474e8c57737625c

your


Signed By: Alice

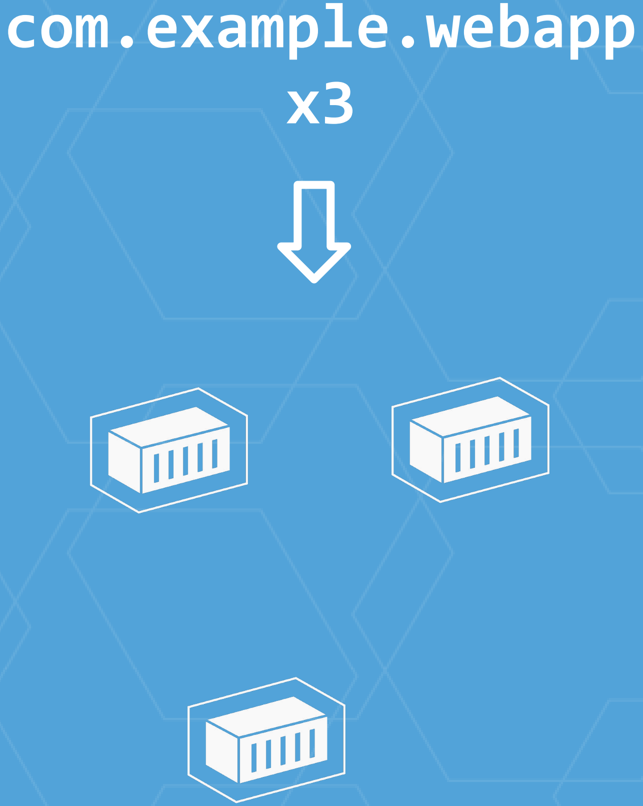
d474e8c57737625c

ops engineer

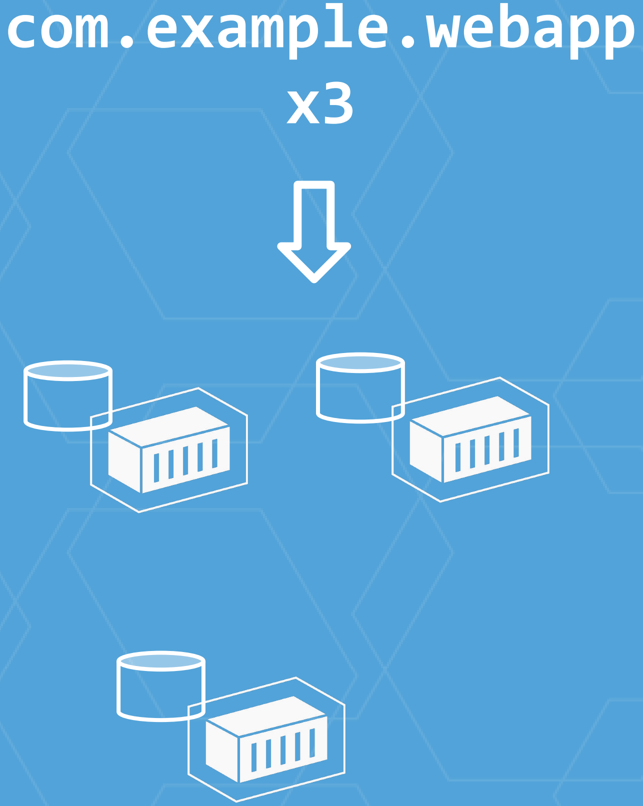
you as an ops engineer

your

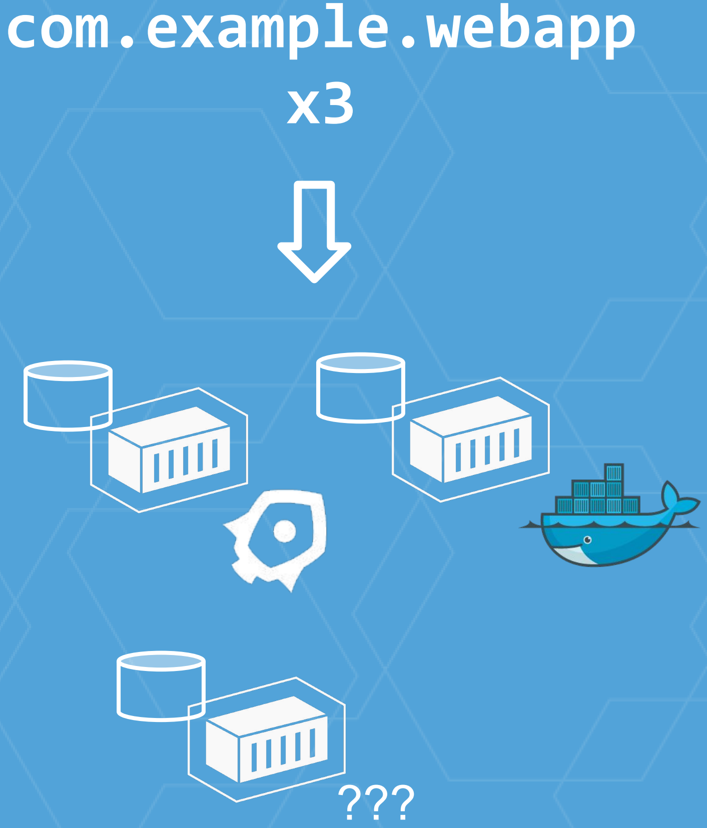
your



your



your



How do we do it?

minimal

reduce API contracts

kernel
systemd
rkt
ssh
docker

python
java
nginx
mysql
openssl

stro distro distro distro distro distro

app

kernel
systemd
rkt
ssh
docker

stro distro distro distro distro distro

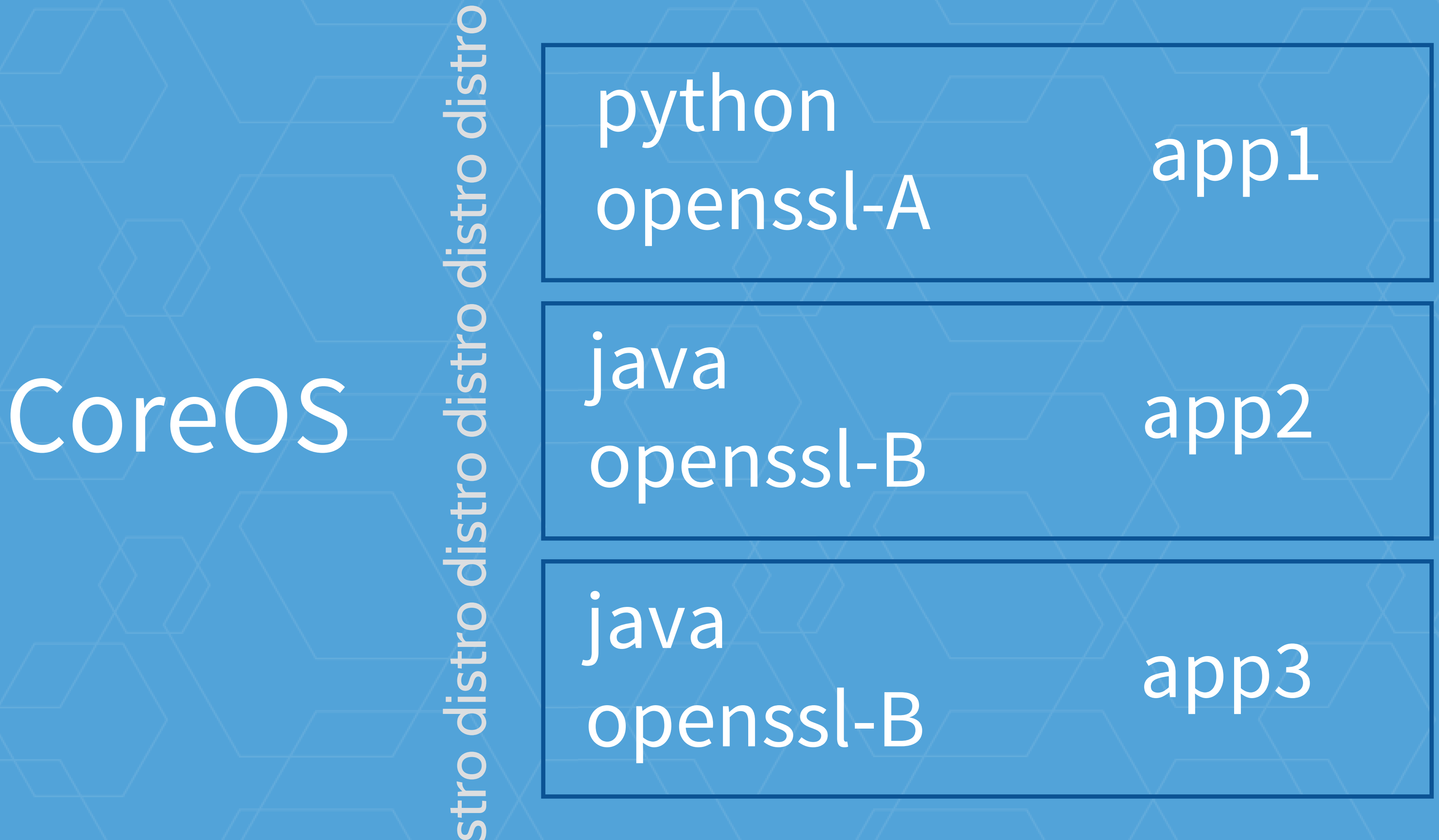
python
java
nginx
mysql
openssl

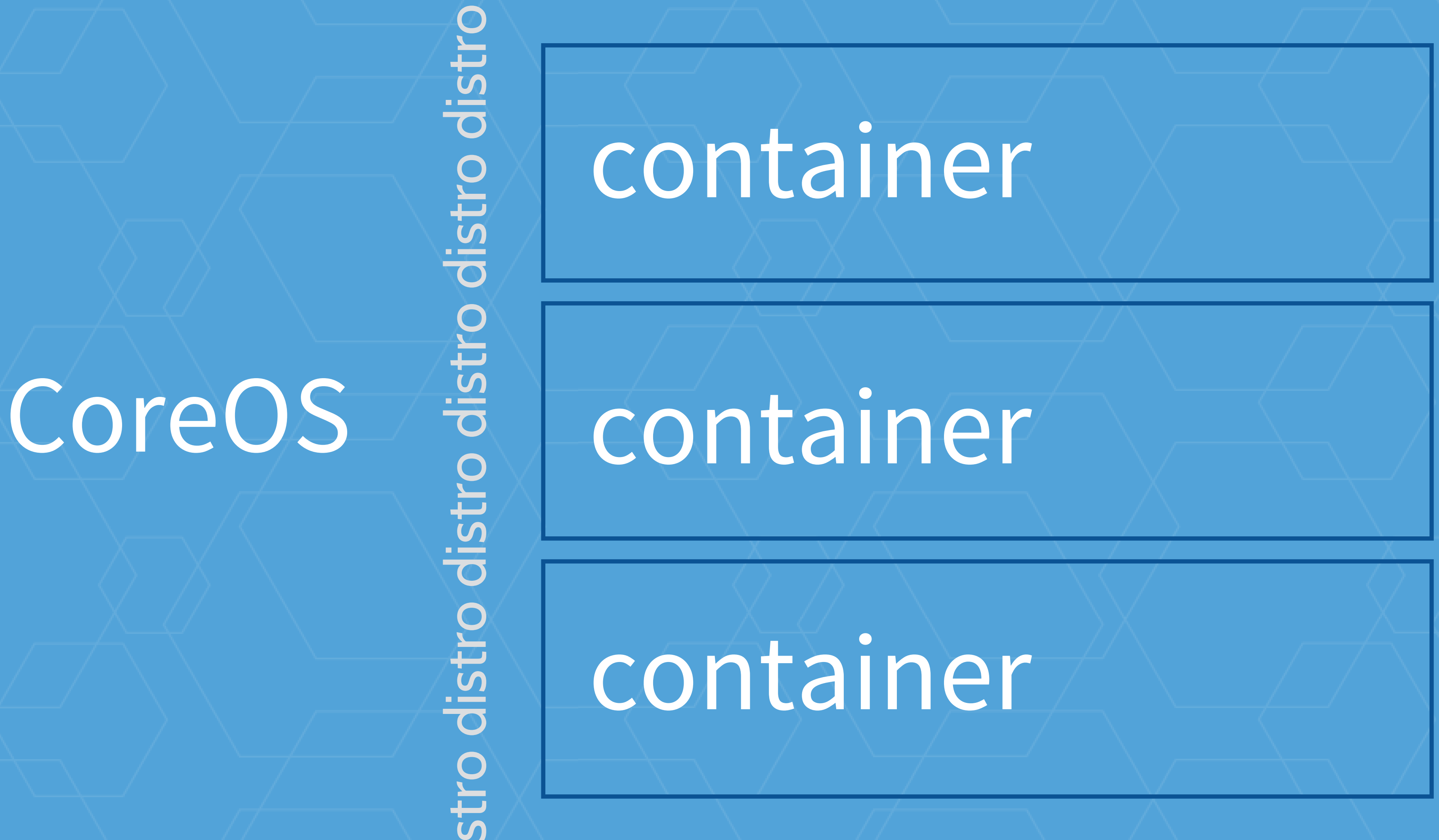
app

kernel
systemd
rkt
ssh
docker

stro distro distro distro distro distro

python openssl-A	app1
java openssl-B	app2
java openssl-B	app3


CoreOS
stro distro distro distro distro distro
python
openssl-A
app1
java
openssl-B
app2
java
openssl-B
app3


CoreOS
stro distro distro distro distro distro
container
container
container

rkt

OS operations

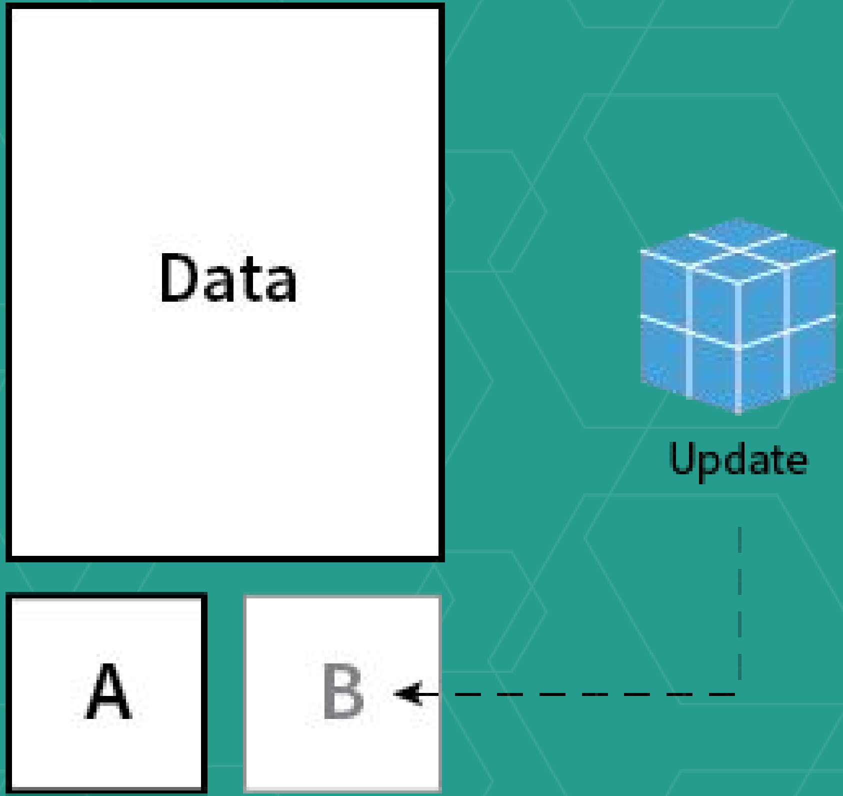
OS operations

updates

manual updates

automatic updates

automatic updates


Data
A
B
Update

Data
A
B

CoreOS Updates

atomic update with rollback

OS operations

machine configuration

machine config

get into the cluster

```
[Service]
ExecStart=/usr/bin/kubelet   --
api_servers=https://172.17.4.101   --
register-node=true  --hostname-
override=172.17.4.201   --cluster_dns=10.
3.0.10   --cluster_domain=cluster.local
--tls-cert-file=worker.pem   --tls-
private-key-file=worker-key.pem
```

```
[Service]
ExecStart=/usr/bin/kubelet    --
api_servers=https://172.17.4.101   --
register-node=true  --hostname-
override=172.17.4.201   --cluster_dns=10.
3.0.10   --cluster_domain=cluster.local
--tls-cert-file=worker.pem   --tls-
private-key-file=worker-key.pem
```

```
[Service]
ExecStart=/usr/bin/kubelet    --
api_servers=https://172.17.4.101    --
register-node=true  --hostname-
override=172.17.4.201    --cluster_dns=10.
3.0.10   --cluster_domain=cluster.local
--tls-cert-file=worker.pem   --tls-
private-key-file=worker-key.pem
```

cluster operations

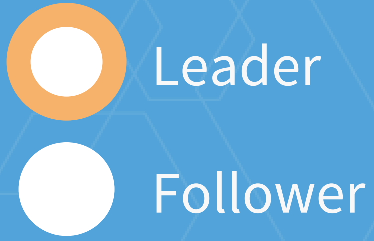
distributed configuration

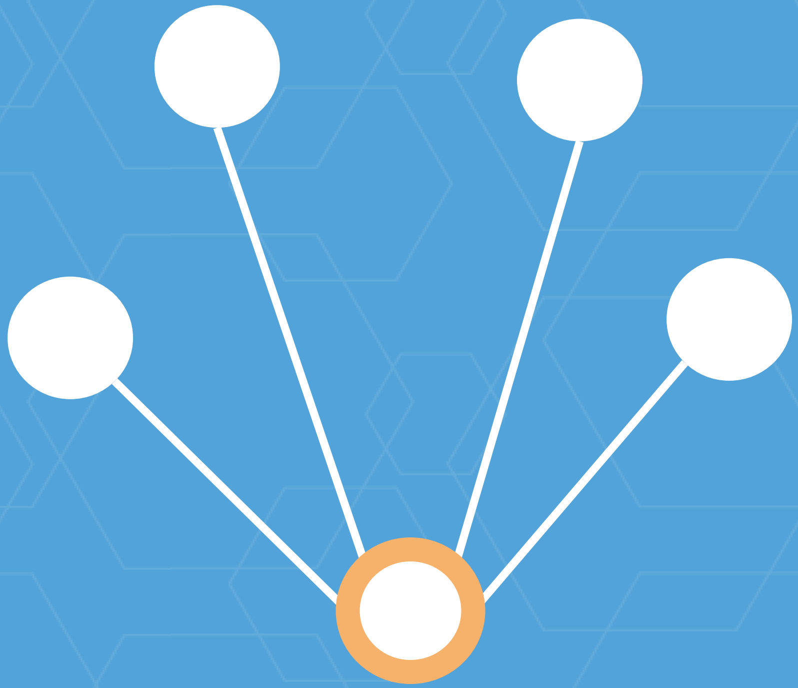
etcd

/etc

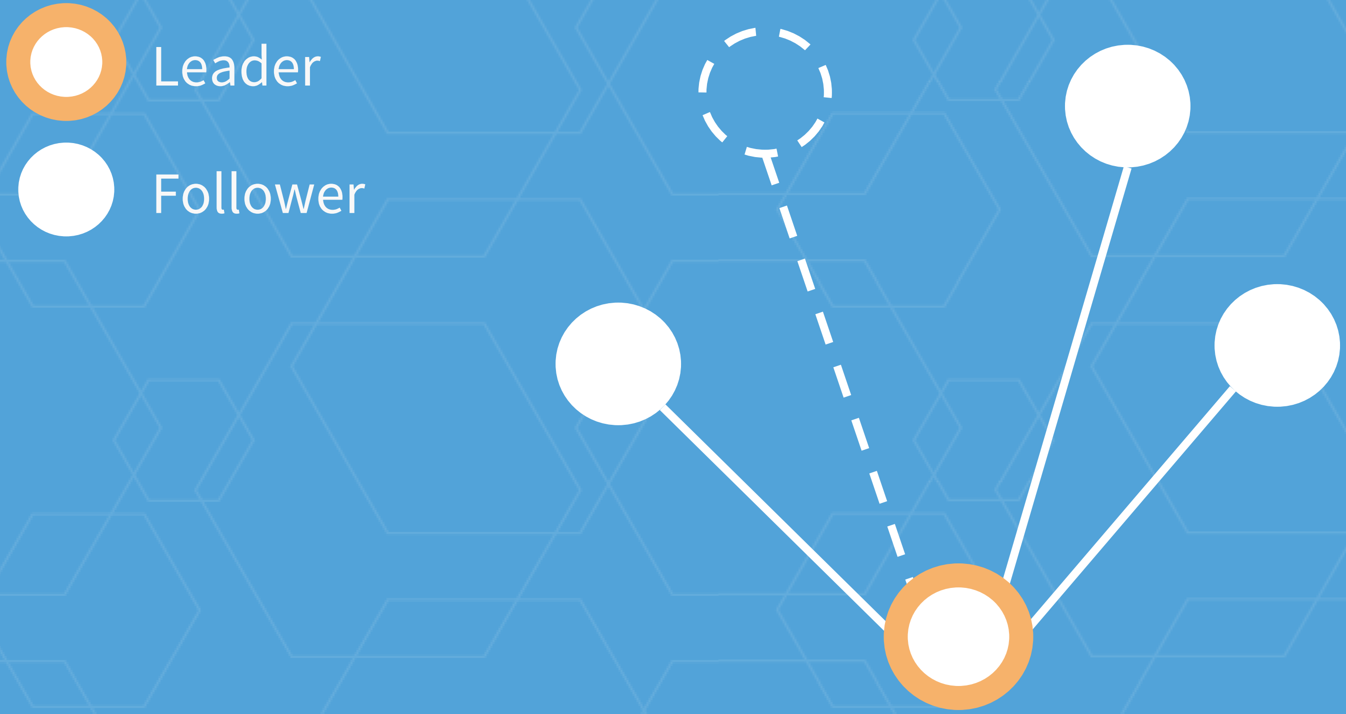
distributed

Available

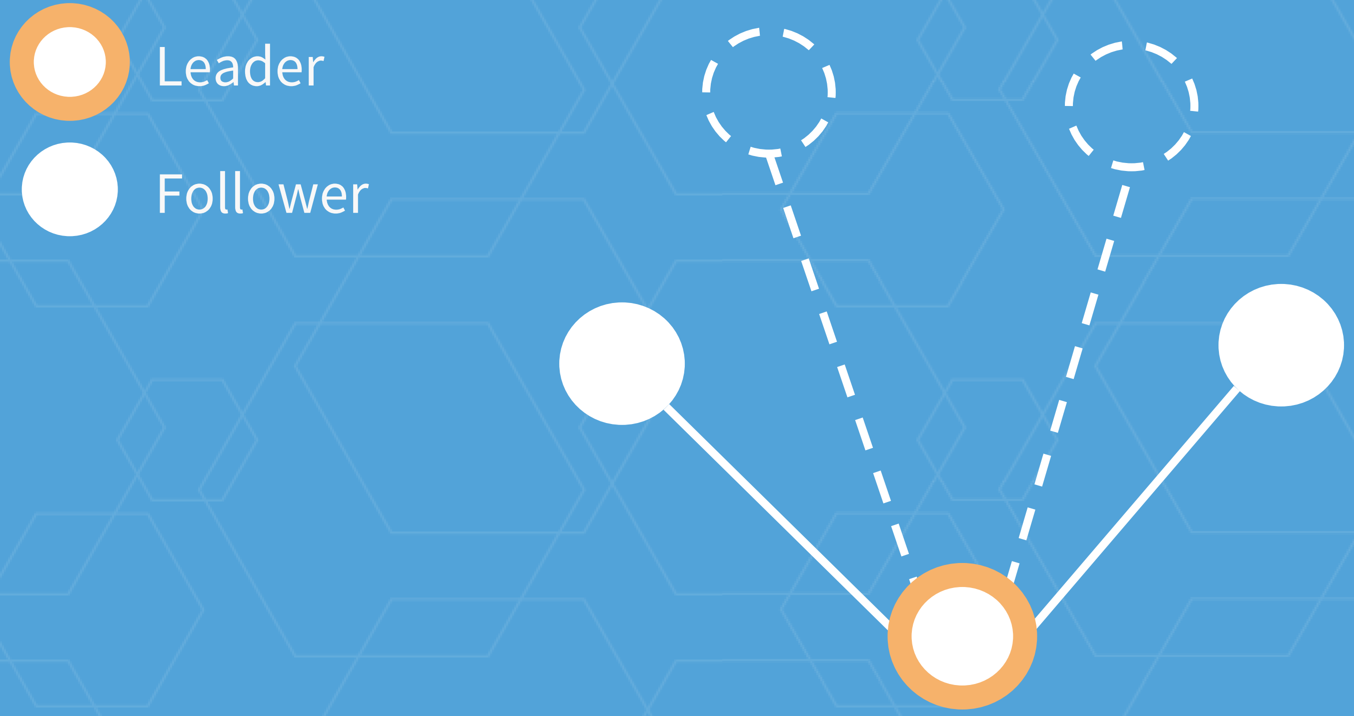




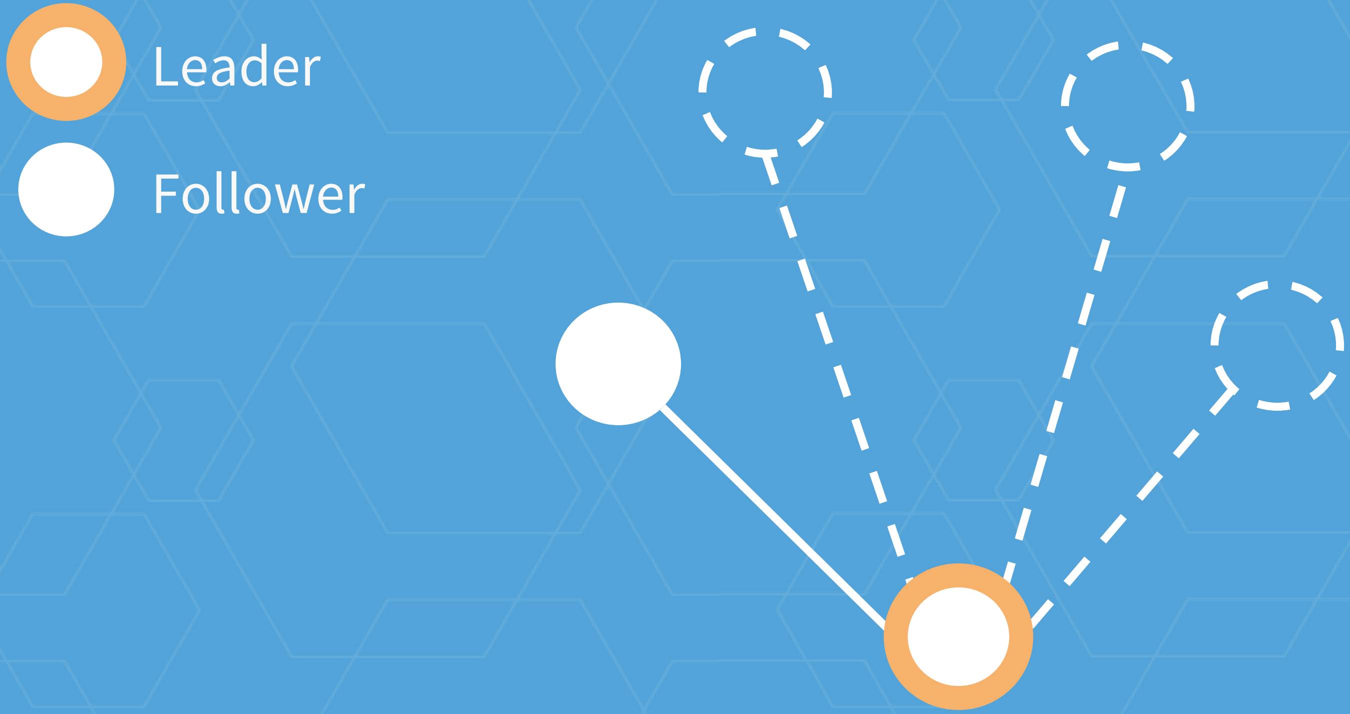
Available



Available

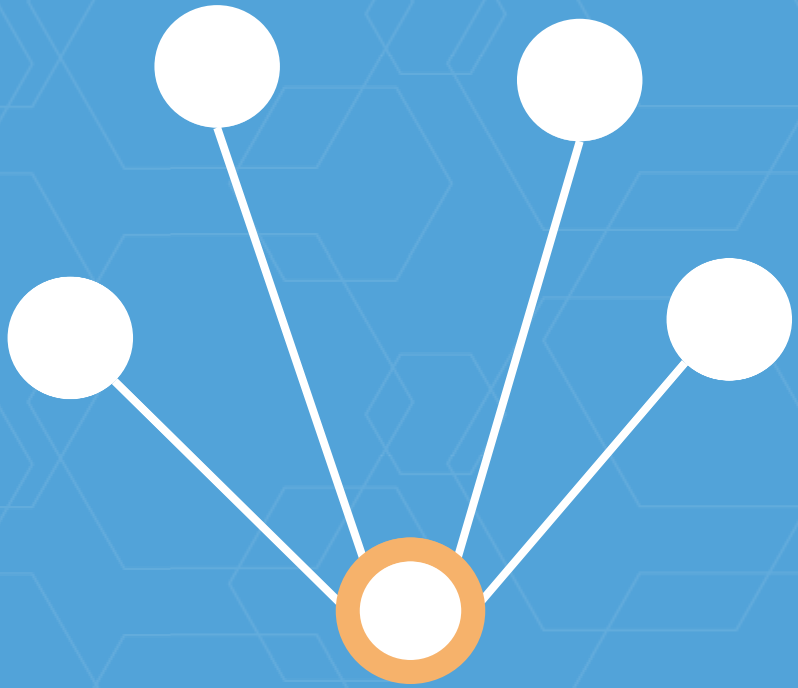


Unavailable

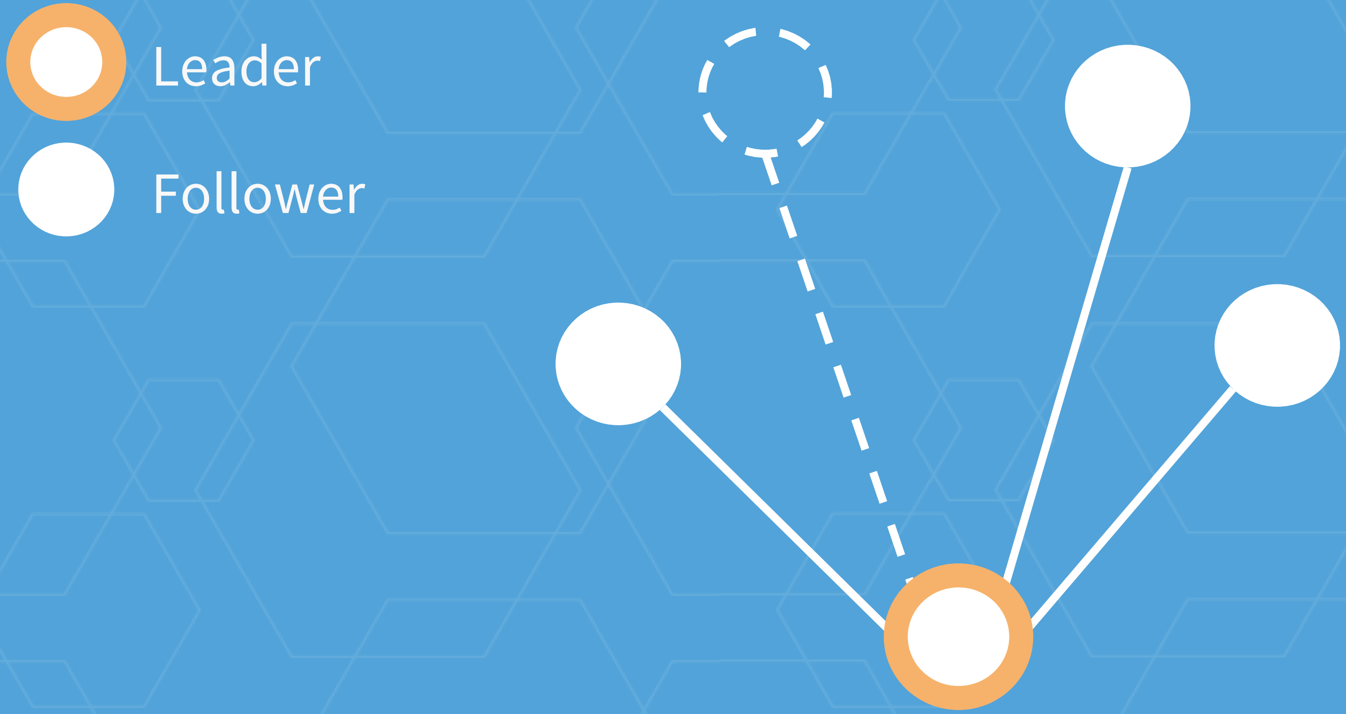


Available

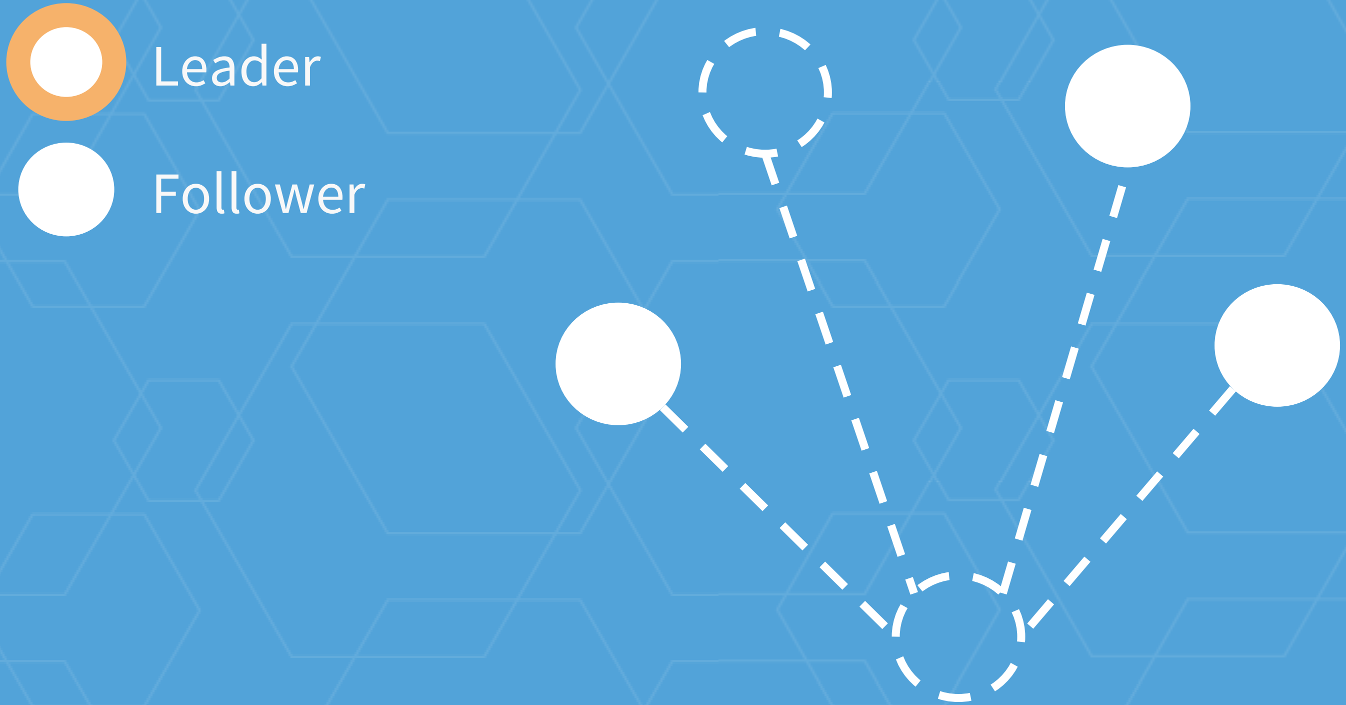
Leader
Follower



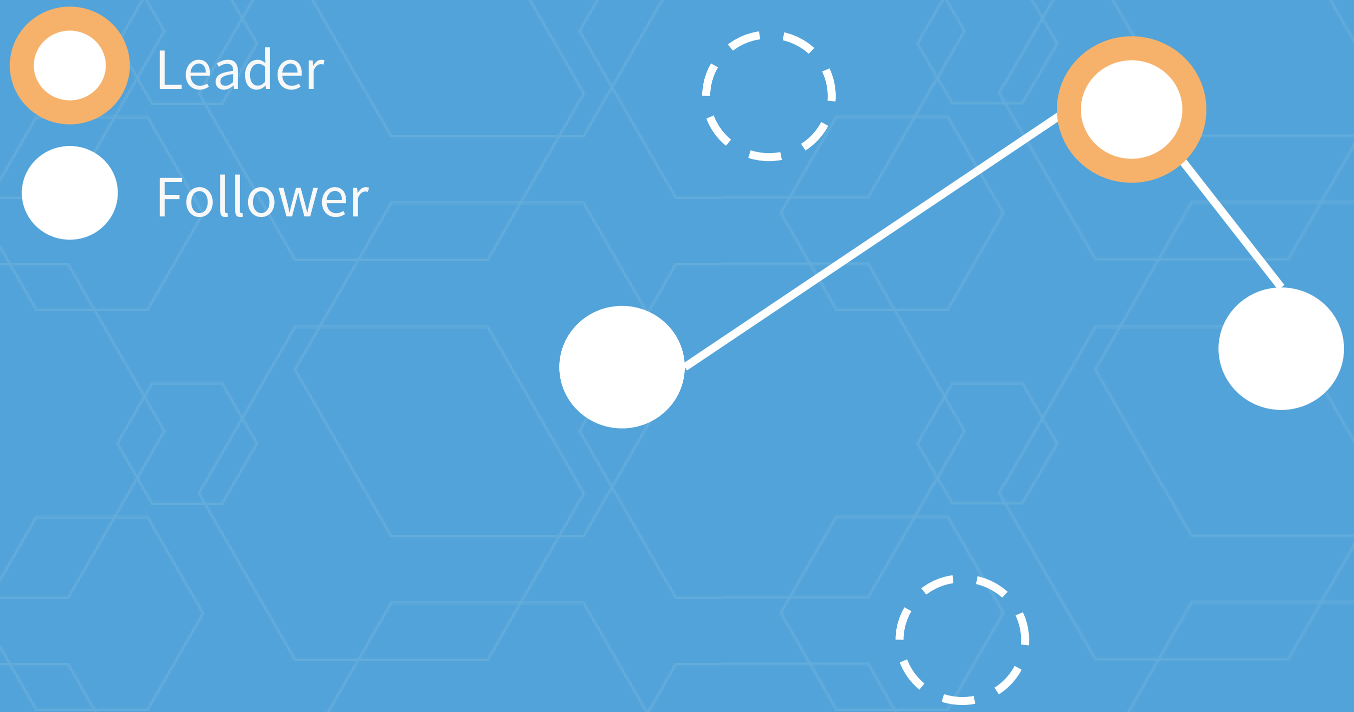
Available



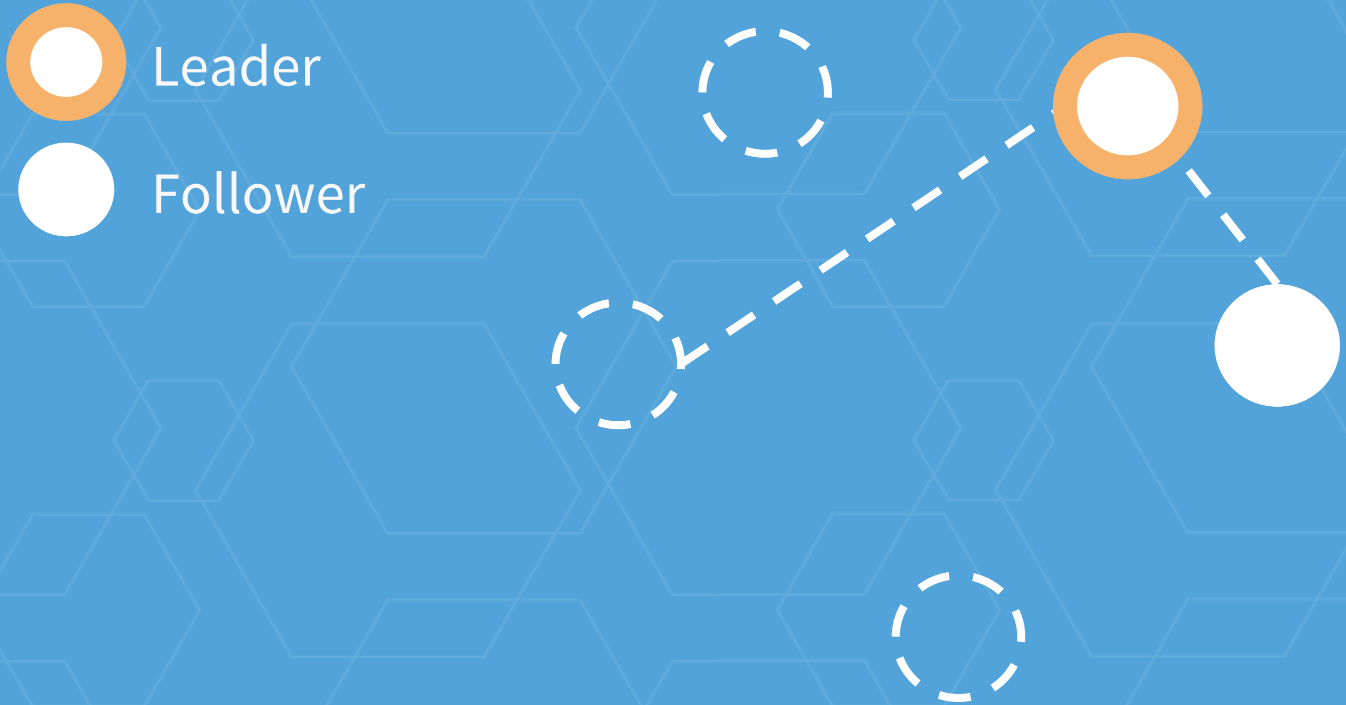
Temporarily Unavailable



Available



Unavailable



cluster operations

what should run

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 lowest-loadaverage

```
$ fab lowest-loadaverage host1
```

```
$ fab lowest-loadaverage
host1
$ fab -H host1 deploy:job
```

You

You

⬇

Scheduler API

$$\begin{tikzcd}
\text{You} \arrow[d] \\
\text{Scheduler API} \arrow[d] \\
\text{Scheduler}
\end{tikzcd}$$

$$\begin{tikzcd}
\text{You} \arrow[d] \\
\text{Scheduler API} \arrow[d] \\
\text{Scheduler} \arrow[d] \\
\text{Machine(s)}
\end{tikzcd}$$

```
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)
}
```

```
$ kubectl run example
--image=quay.io/crawford/example
--replicas=1

$ kubectl get pods
POD                IP
example-97wt8  10.2.29.4
```

```
$ kubectl run example
--image=quay.io/crawford/example
--replicas=1

$ kubectl get pods
POD                IP
example-97wt8  10.2.29.4
```

```
$ kubectl run example
--image=quay.io/crawford/example
--replicas=1

$ kubectl get pods
POD                IP
example-97wt8  10.2.29.4
```

```
$ kubectl run example
--image=quay.io/crawford/example
--replicas=1

$ kubectl get pods
POD                IP
example-97wt8  10.2.29.4
```

```
$ kubectl run example
--image=quay.io/crawford/example
--replicas=1

$ kubectl get pods
POD                IP
example-97wt8  10.2.29.4
```

```
$ kubectl run example
--image=quay.io/crawford/example
--replicas=1

$ kubectl get pods
POD                 IP
example-97wt8  10.2.29.4
```

```
$ kubectl scale rc example
--replicas=2

$ kubectl get pods
POD                 IP
example-97wt8  10.2.29.4
example-f839d  10.2.29.8
```

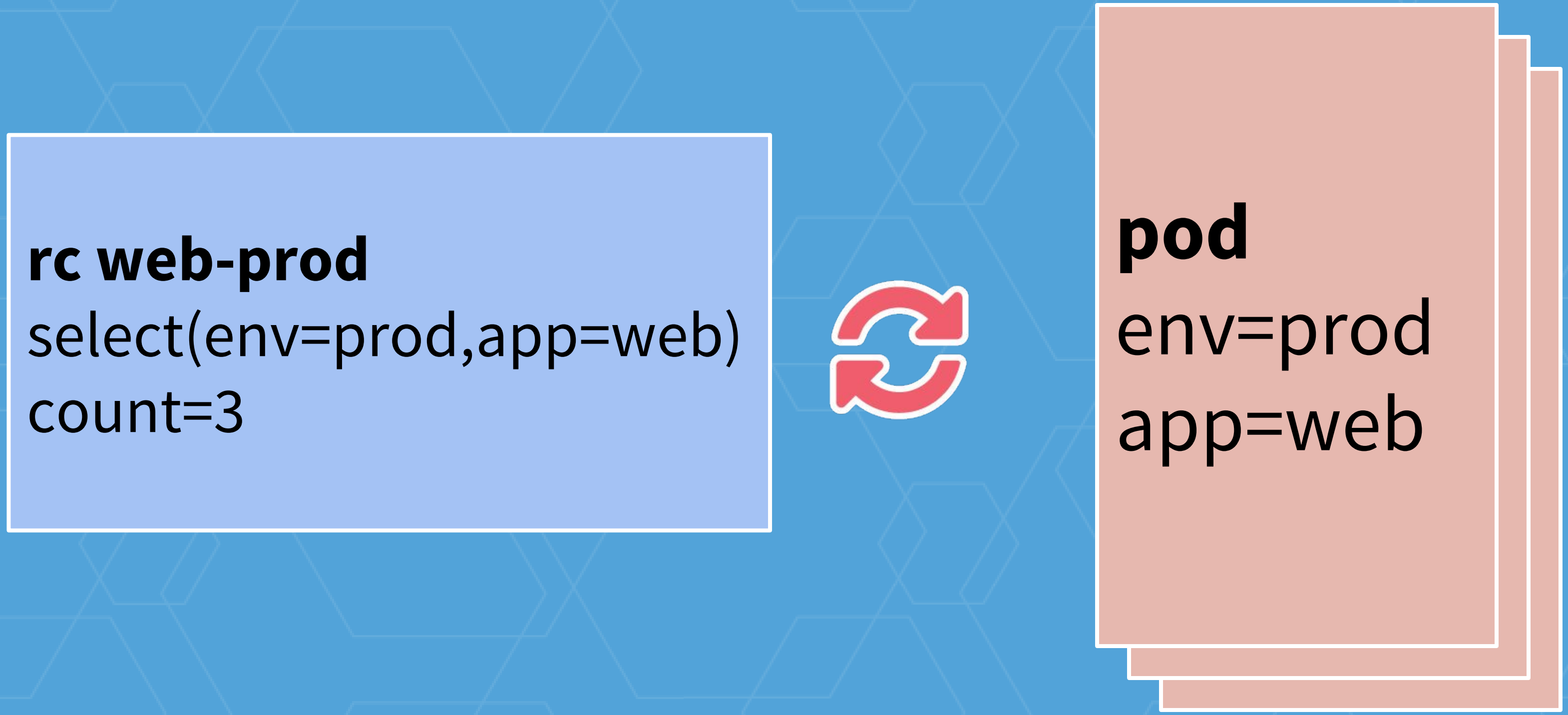
```
$ kubectl scale rc example
--replicas=2

$ kubectl get pods
POD                IP
example-97wt8  10.2.29.4
example-f839d  10.2.29.8
```

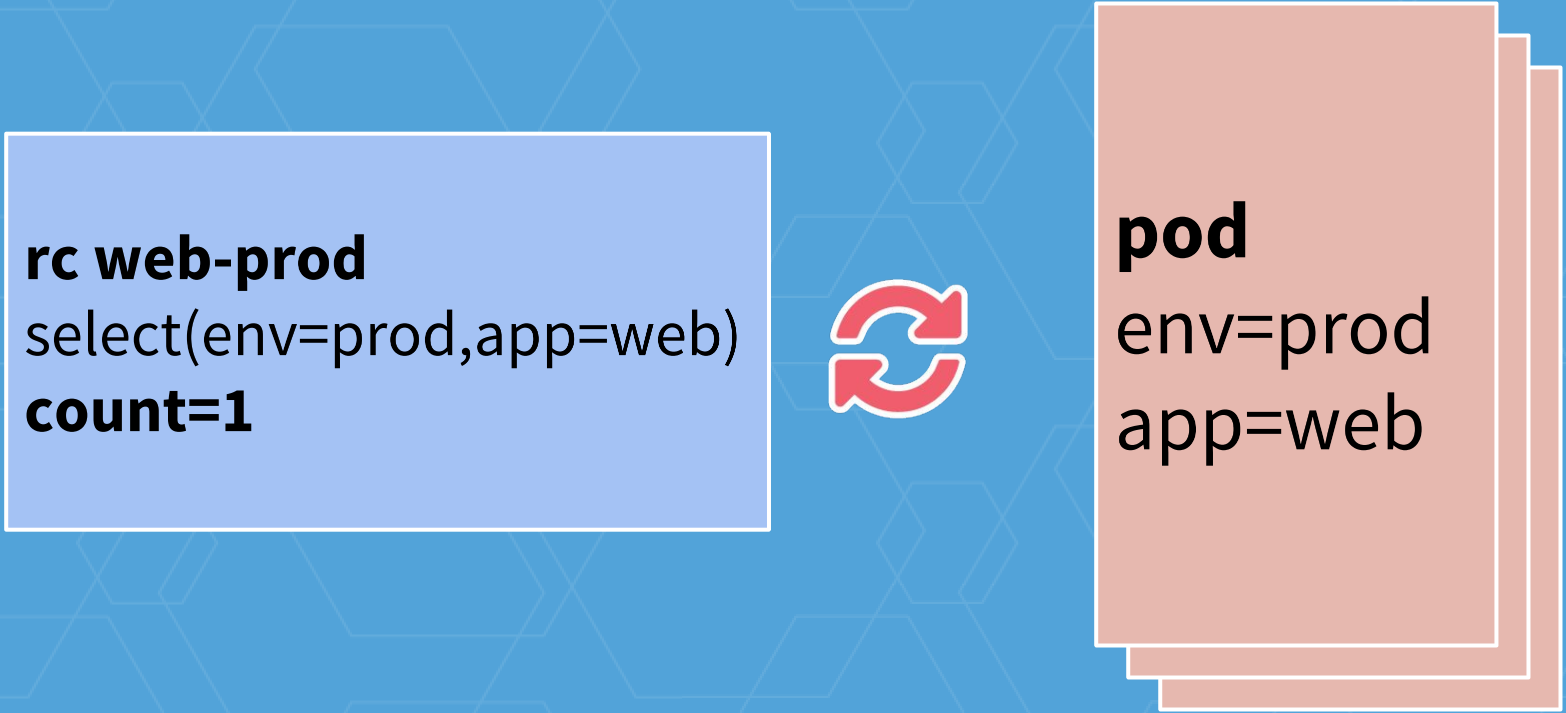
```
$ kubectl scale rc example
--replicas=2

$ kubectl get pods
POD               IP
example-97wt8  10.2.29.4
example-f839d  10.2.29.8
```

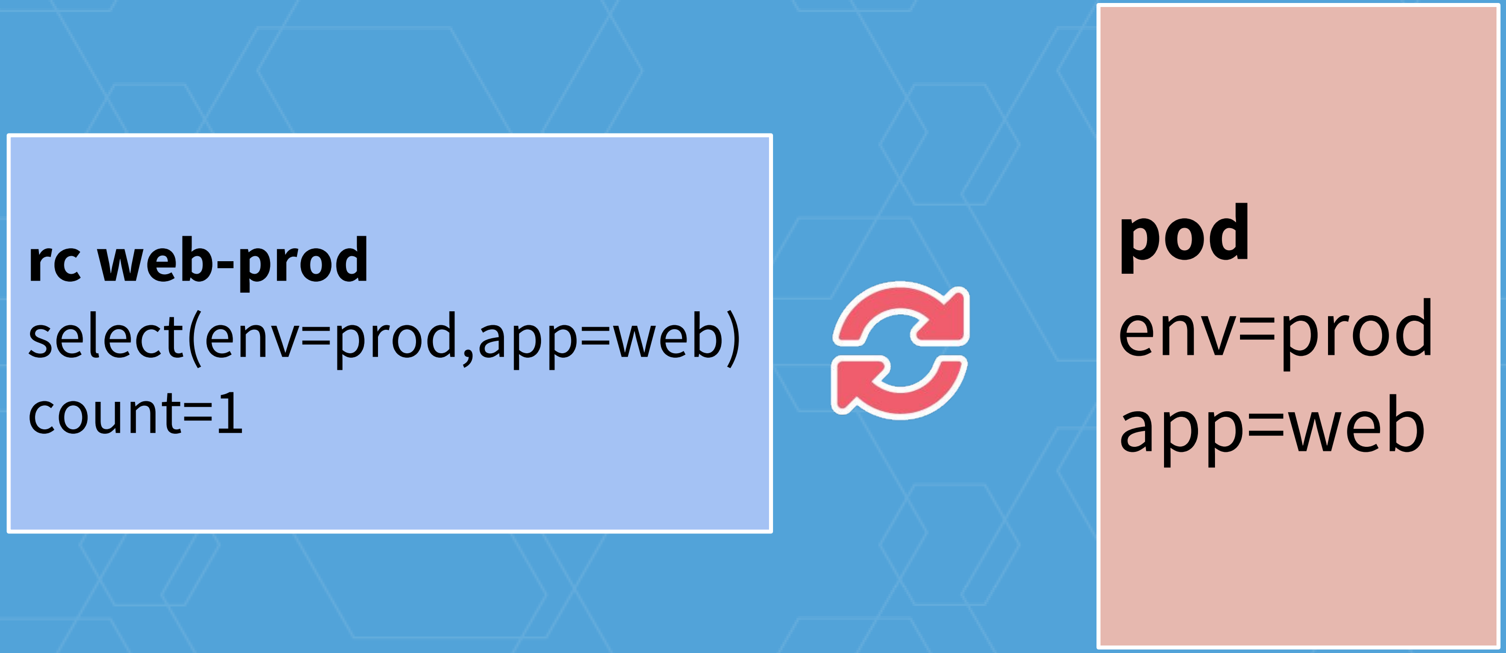
```
$ kubectl scale rc example
--replicas=2

$ kubectl get pods
POD                IP
example-97wt8  10.2.29.4
example-f839d  10.2.29.8
```

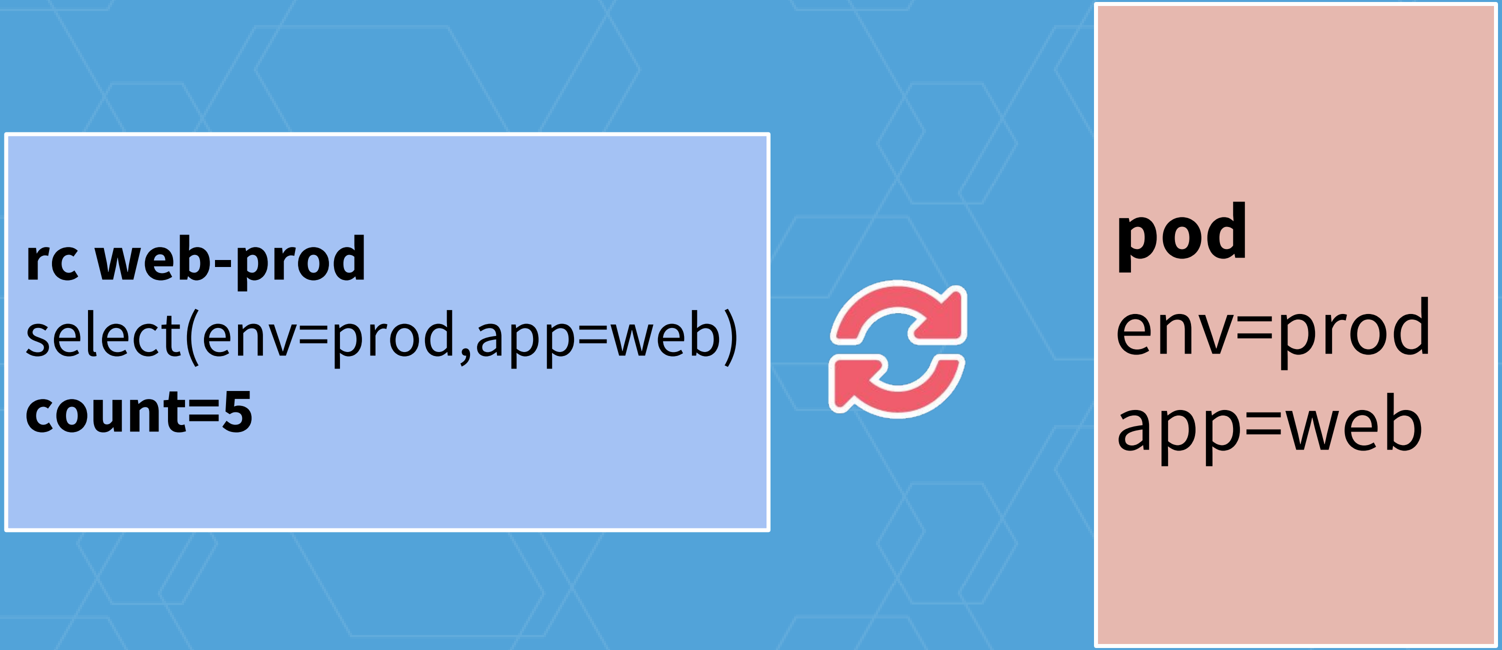
```
$ kubectl scale rc example
--replicas=2

$ kubectl get pods
POD                IP
example-97wt8  10.2.29.4
example-f839d  10.2.29.8
```

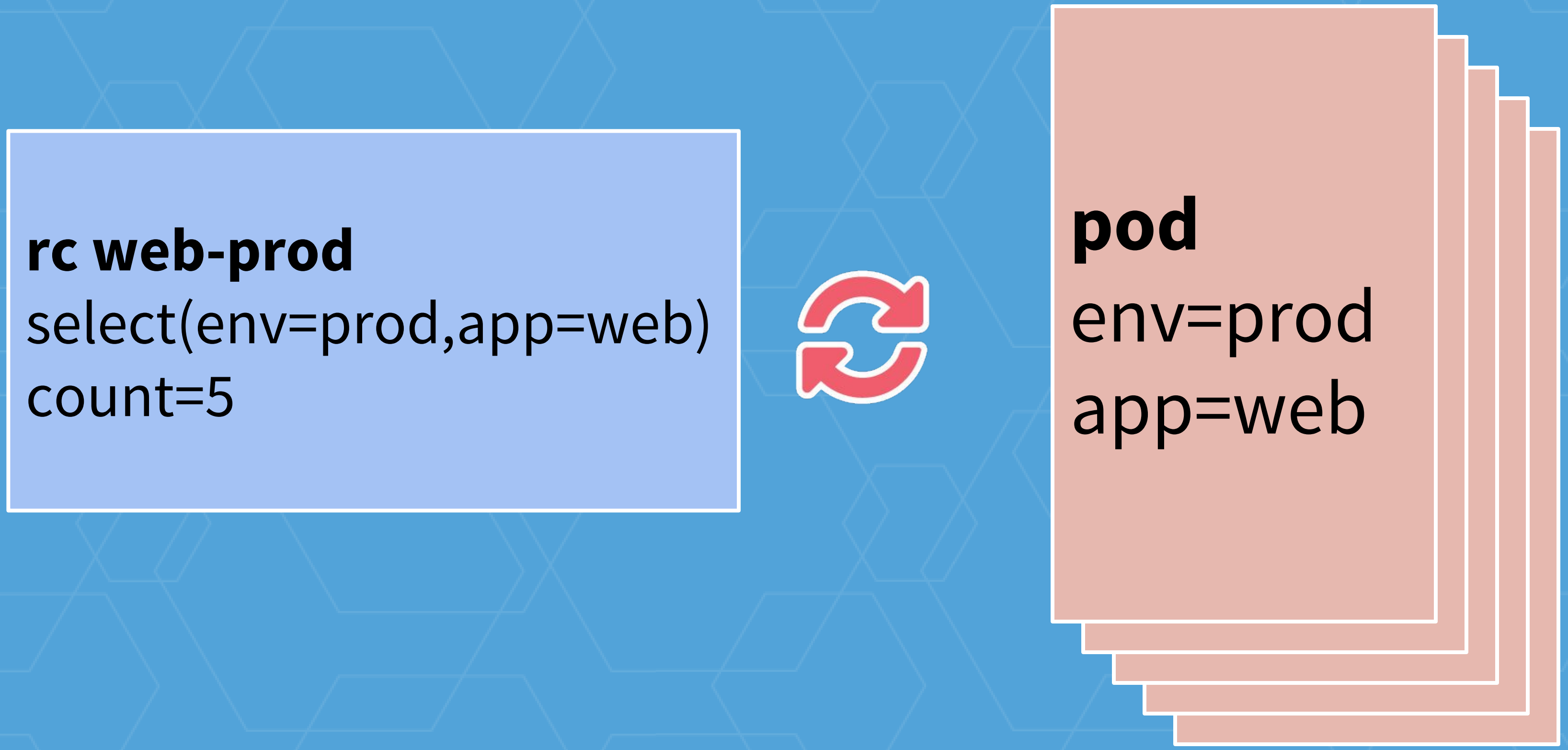
```
$ kubectl scale rc example
--replicas=2

$ kubectl get pods
POD               IP
example-97wt8  10.2.29.4
example-f839d  10.2.29.8
```


rc web-prod
select(env=prod,app=web)
count=3
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

cluster operations

where is it running

services
dns, LBs, k8s labels

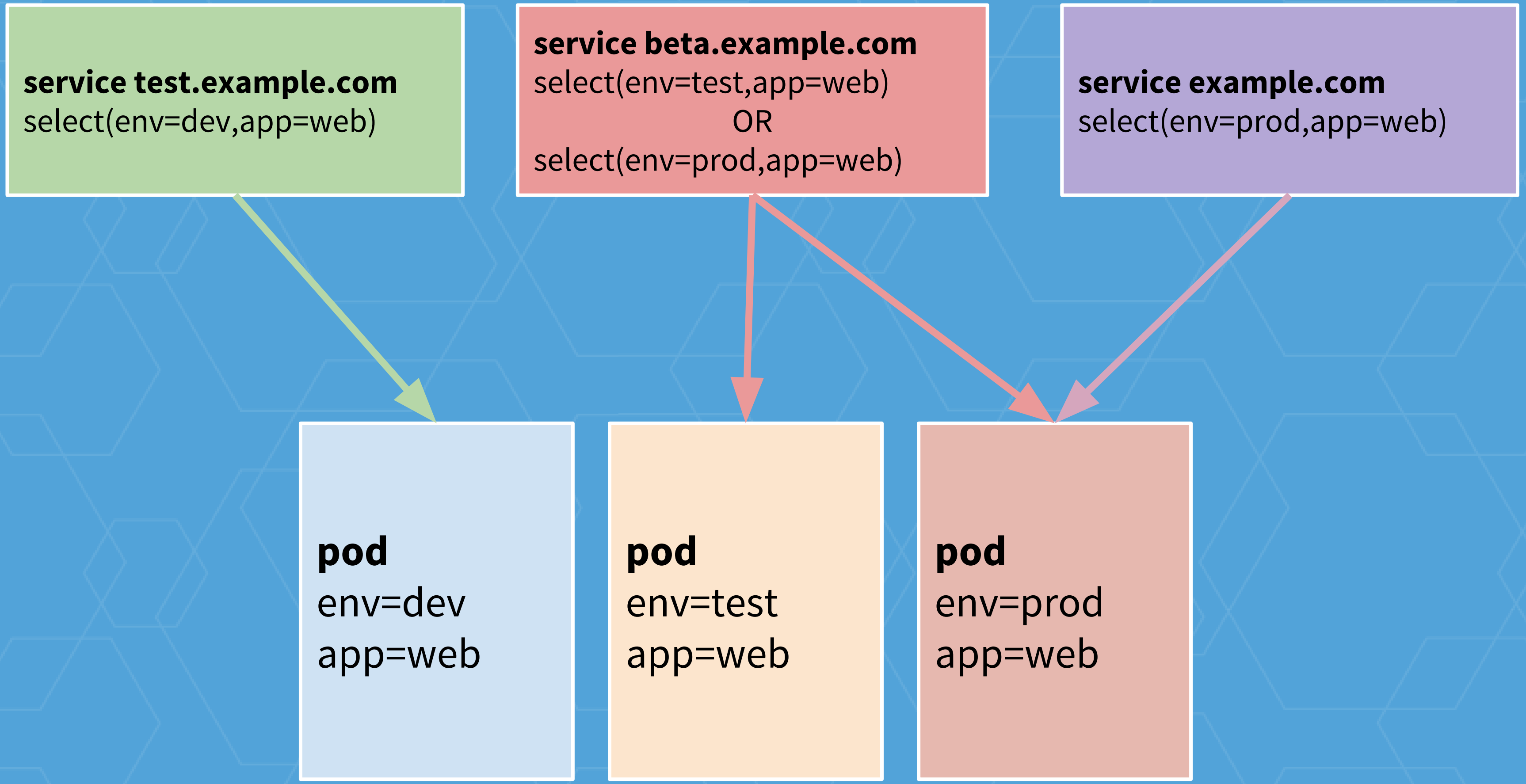
k8s labels

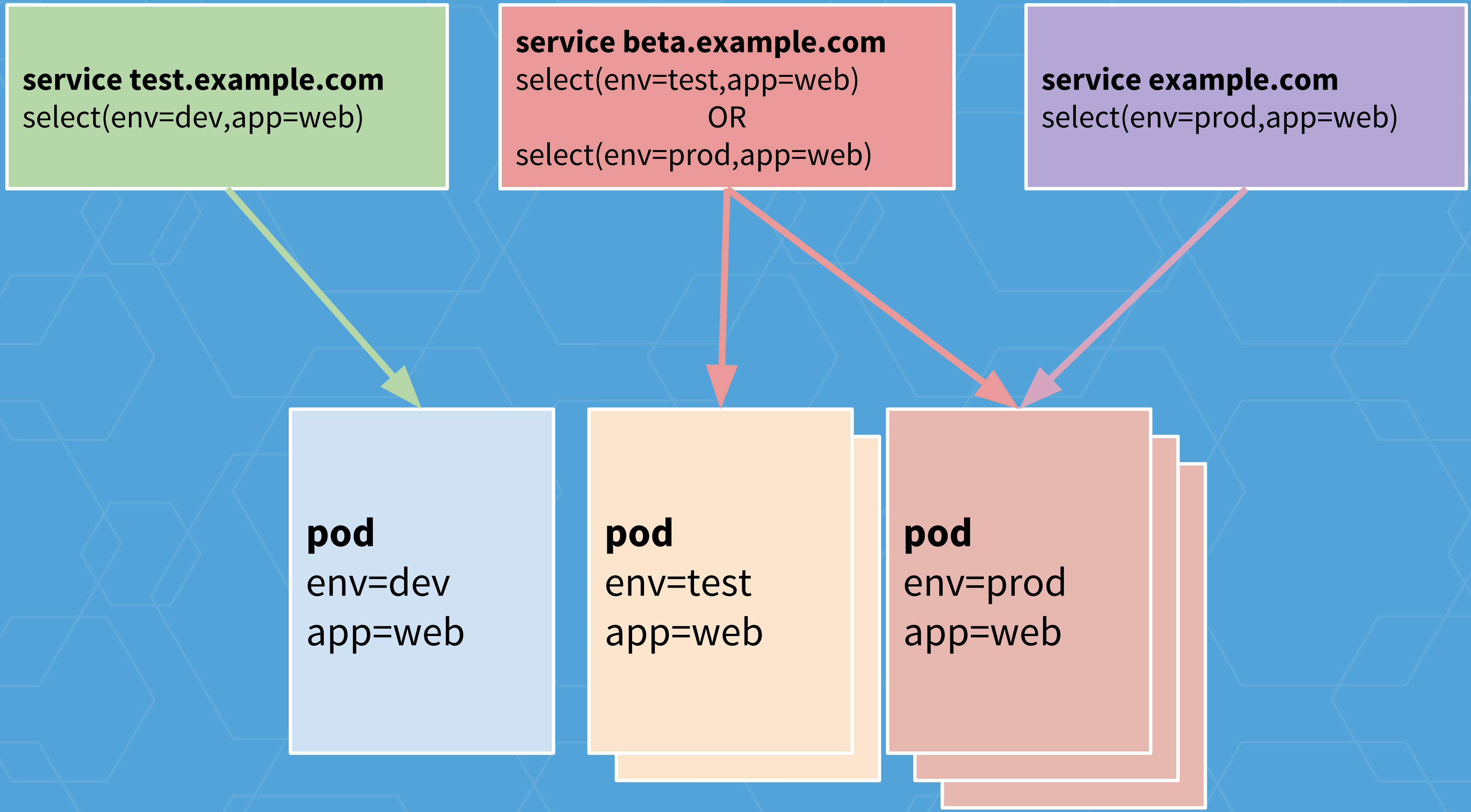
flexible service discovery

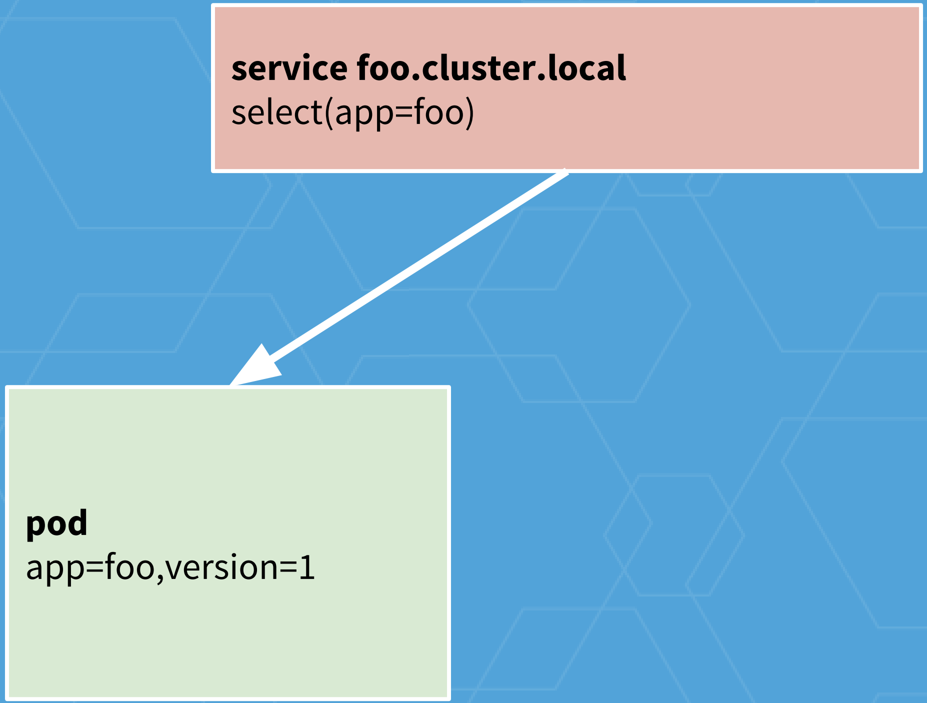
pod
env=dev
app=web

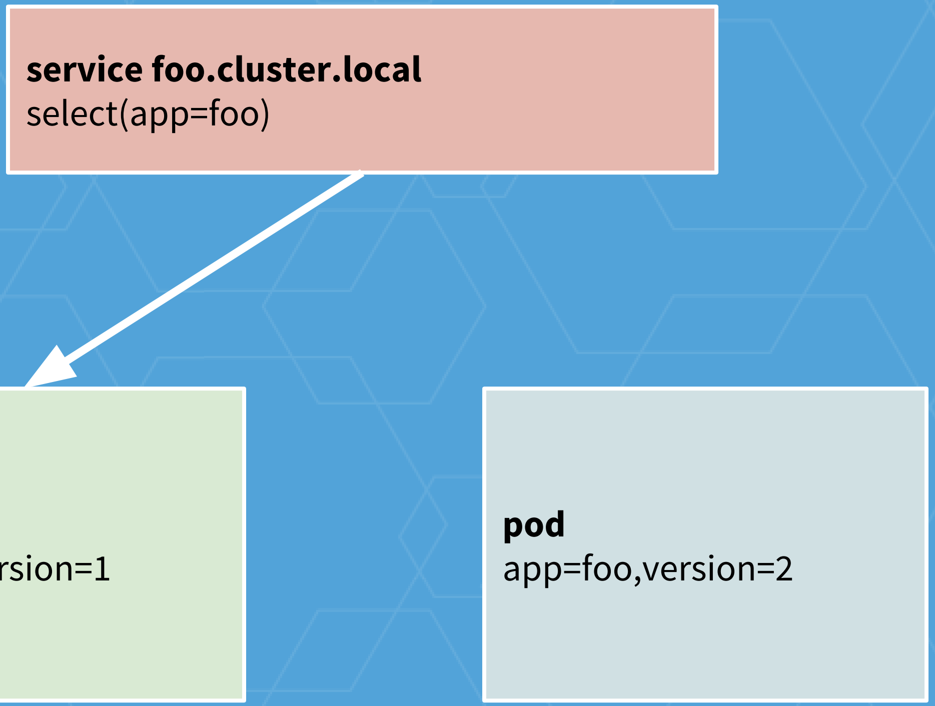
pod
env=test
app=web

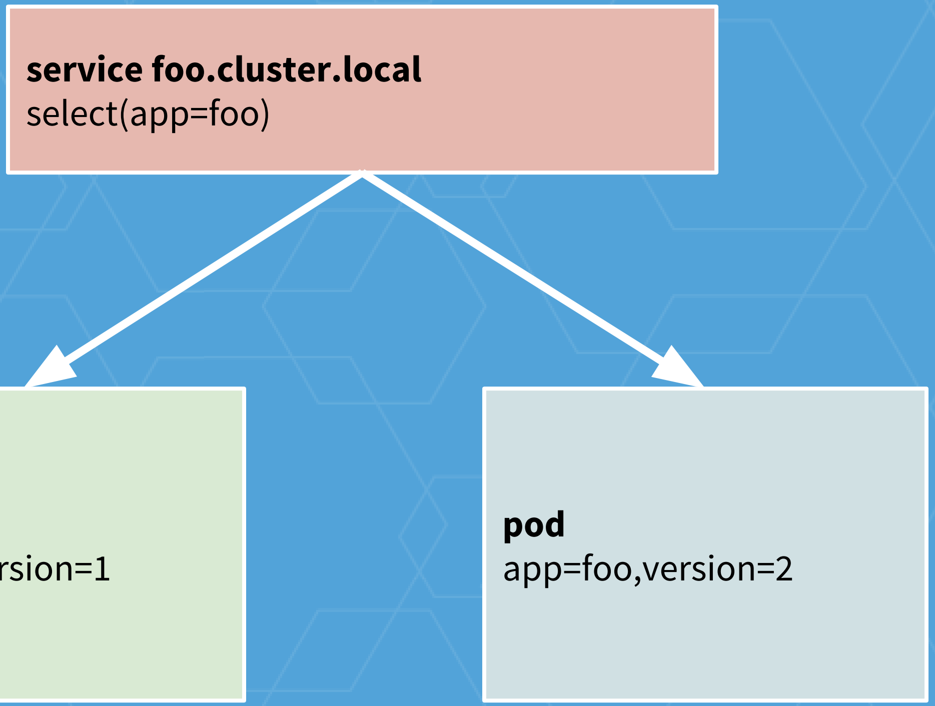
pod
env=prod
app=web

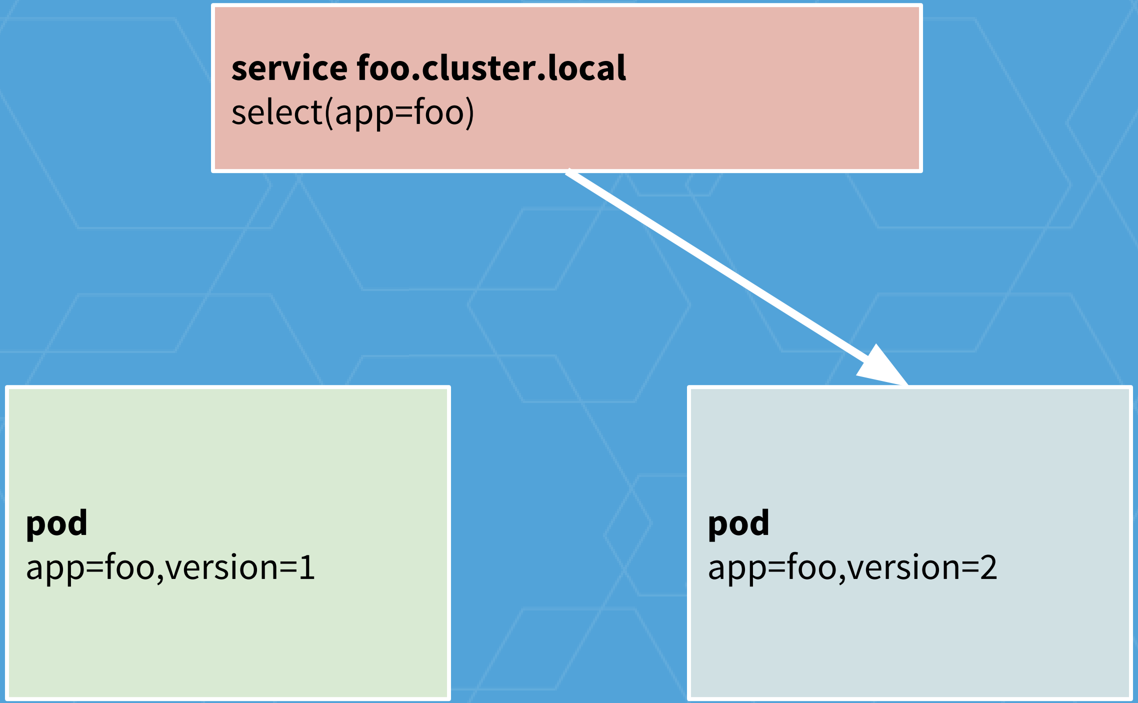

service test.example.com
select(env=dev,app=web)
service beta.example.com
select(env=test,app=web)
OR
select(env=prod,app=web)
service example.com
select(env=prod,app=web)
pod
env=dev
app=web
pod
env=test
app=web
pod
env=prod
app=web

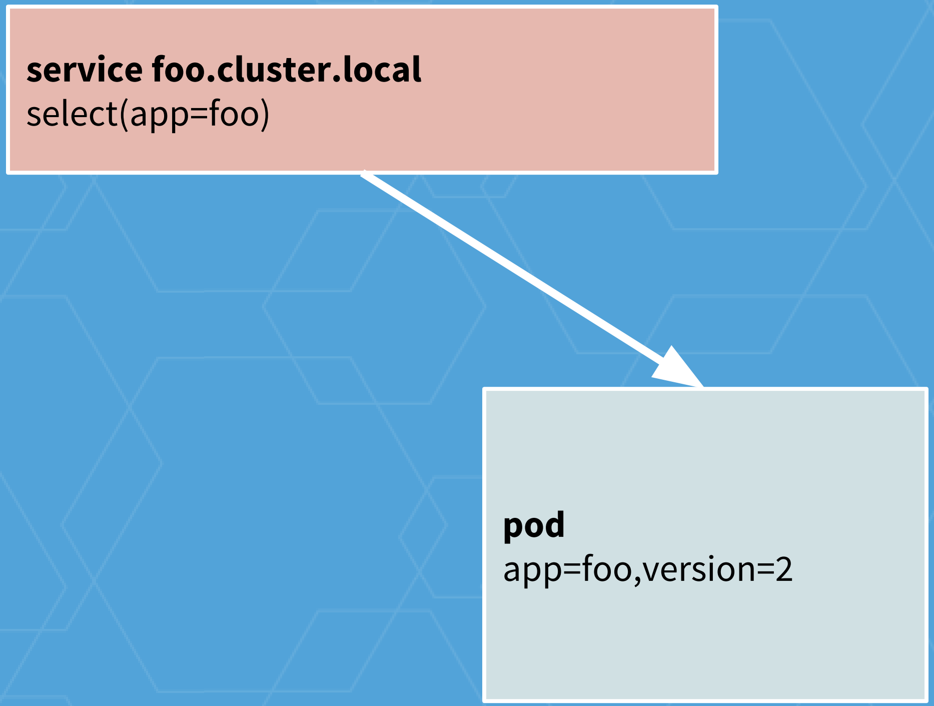

service test.example.com
select(env=dev,app=web)
service beta.example.com
select(env=test,app=web)
OR
select(env=prod,app=web)
service example.com
select(env=prod,app=web)
pod
env=dev
app=web
pod
env=test
app=web
pod
env=prod
app=web


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


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=2

work with us

coreos.com/careers

Questions?