CoreOS

Containers and distributed applications

Xiang Li
Head of distributed system @ CoreOS

Distributed system at CoreOS

We solve hardest distributed system problems

CoreOS

Mission

Secure the Internet

CoreOS

STRATEGY

Make servers easy to upgrade

CoreOS Linux

Self update

Separate OS and the Apps

- enforce container oriented deployment

CoreOS Linux

1000+ releases

CoreOS

STRATEGY

Simplify application management

CoreOS

etcd

Configuration management

Distributed coordination

CoreOS

fleet

Distributed init system

CoreOS

Kubernetes

The “Replacement” for fleet

Manage containerized applications

CoreOS

STRATEGY

On any infrastructure

CoreOS Linux

Public cloud

- AWS, Azure, GCE, Packet...

Private cloud

- OpenStack

Bare mental

- iPXE

CoreOS

flannel

Enabling IP per container on any infrastructure

CoreOS

Accelerate the industry adoption of containers

CoreOS

Container

Run applications anywhere

Container

Package

Container

Push

Pull

Container

Run

Docker

```
$ docker build

$ docker push

$ docker run
```

Container Engine

Manage the lifecycle of a *single* application

Why CoreOS uses container

Make OS secure

Why CoreOS uses container

Deploy distributed applications

CoreOS

Why Google uses container

Run distributed applications efficiently

CoreOS

Container management

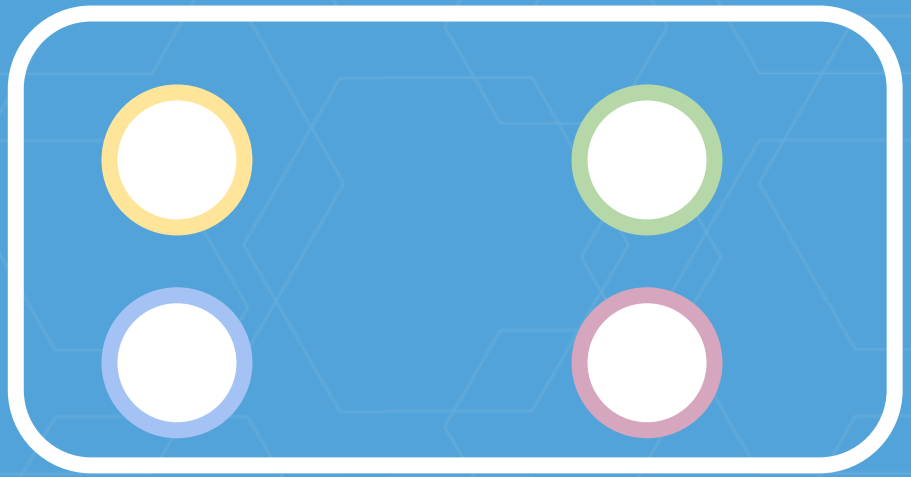
Managing the lifecycle of *one* application instance is not enough

CoreOS

Running containers on one node

CoreOS

Running containers on one node

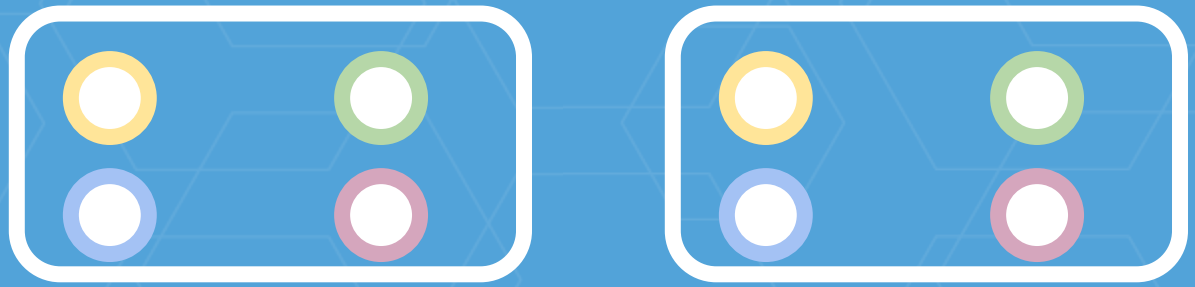


CoreOS

Running containers on two nodes

CoreOS

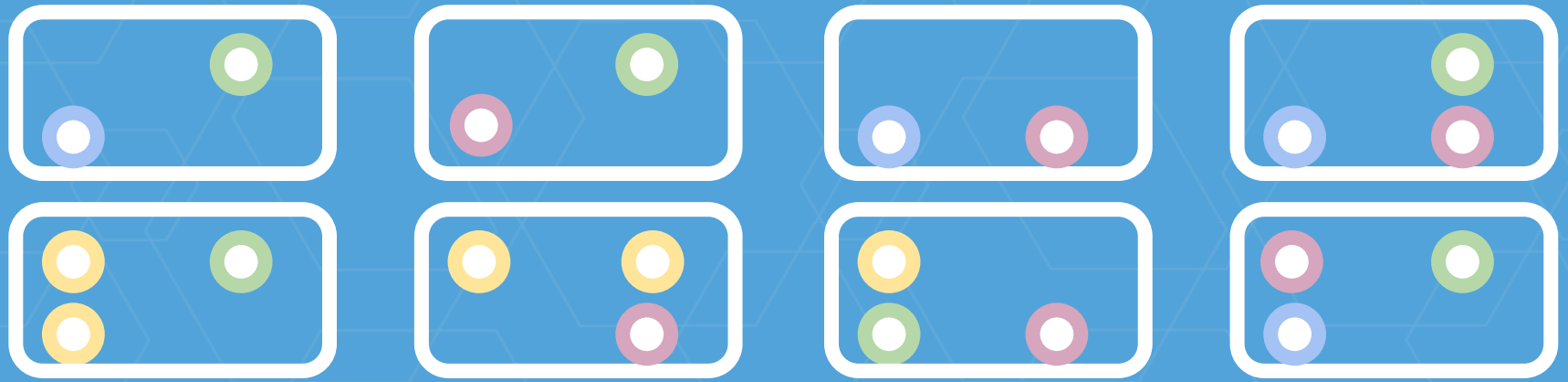
Running containers on two nodes



CoreOS

Running containers on a cluster

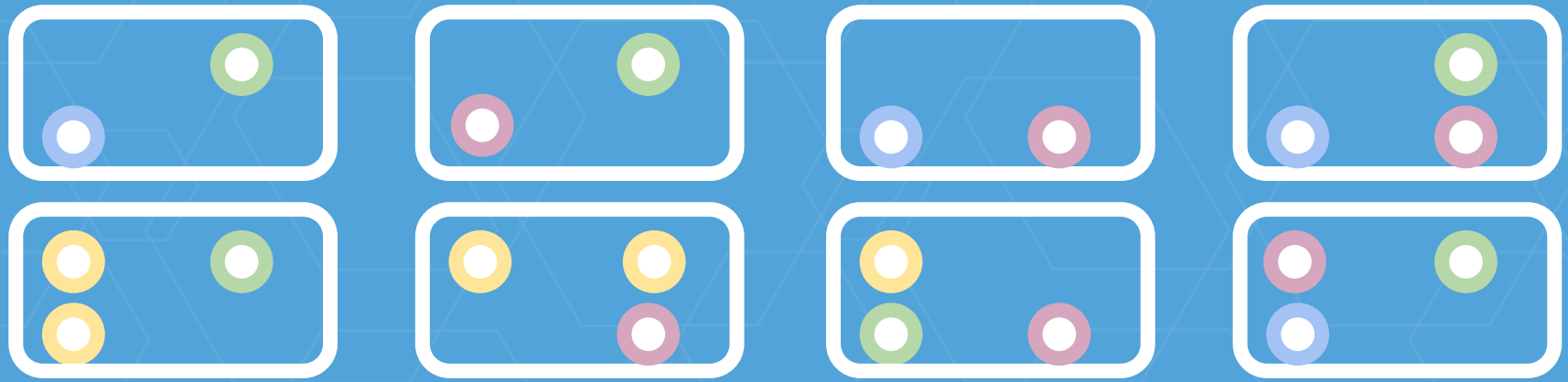
Where to put my containers?



CoreOS

Running containers on a cluster

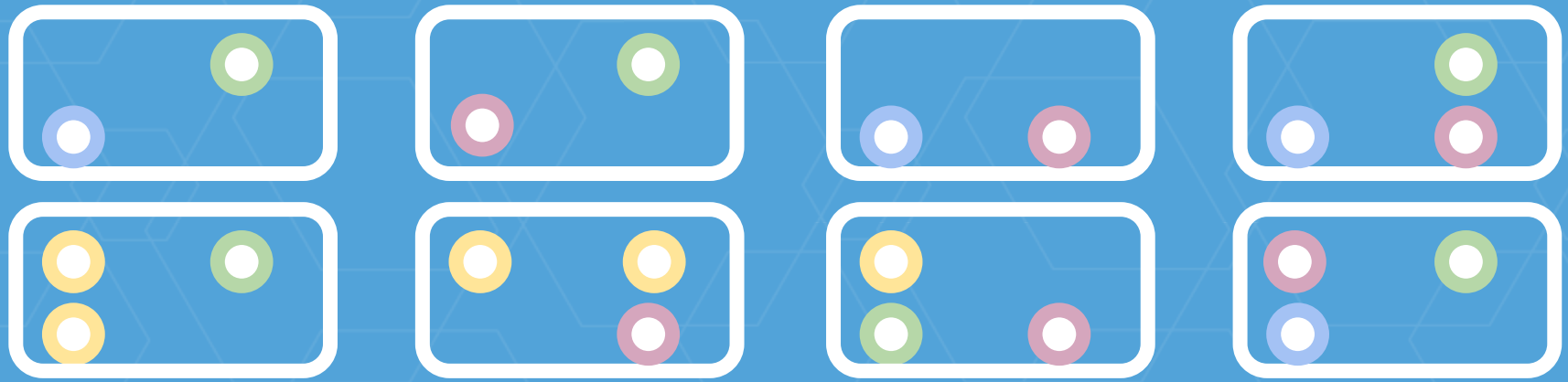
How to find my containers?



CoreOS

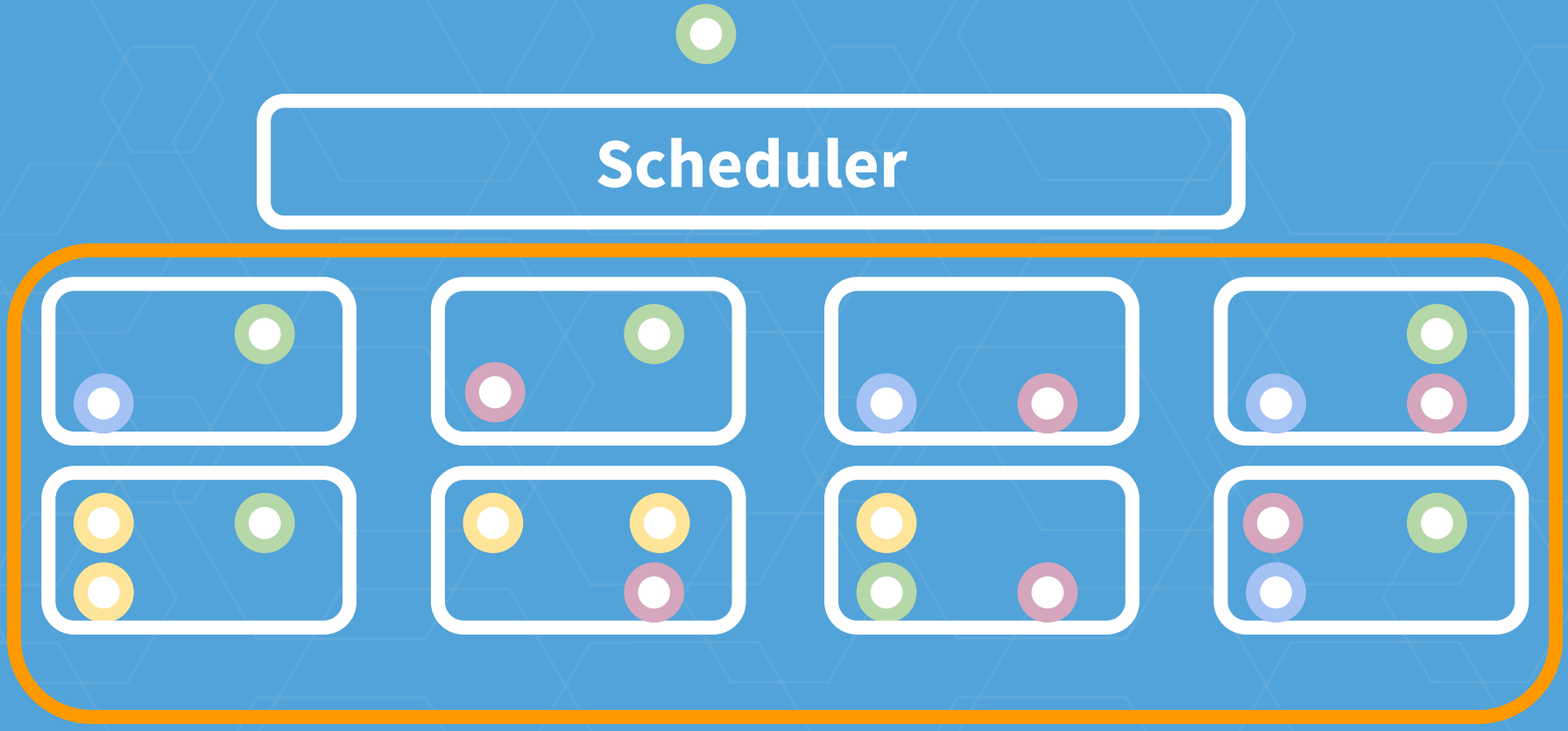
Running containers on a cluster

How to connect to my containers?



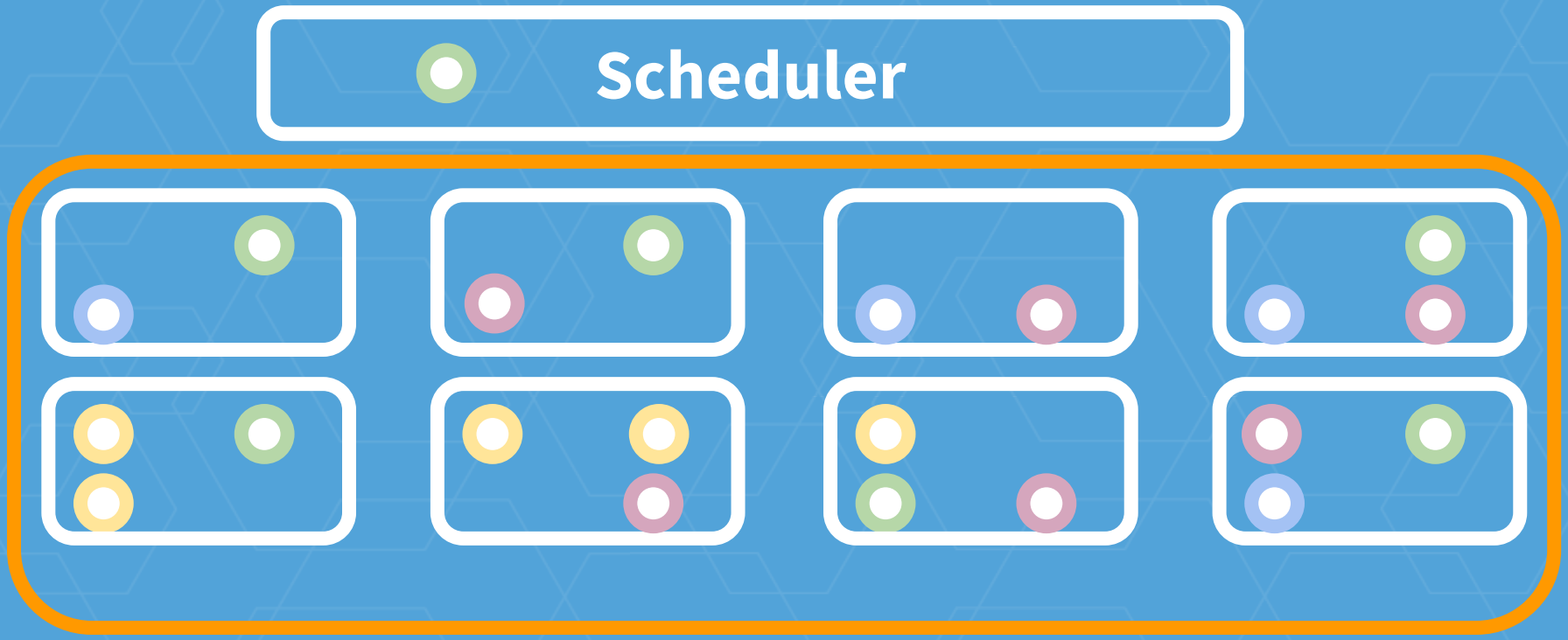
CoreOS

Running containers on a cluster



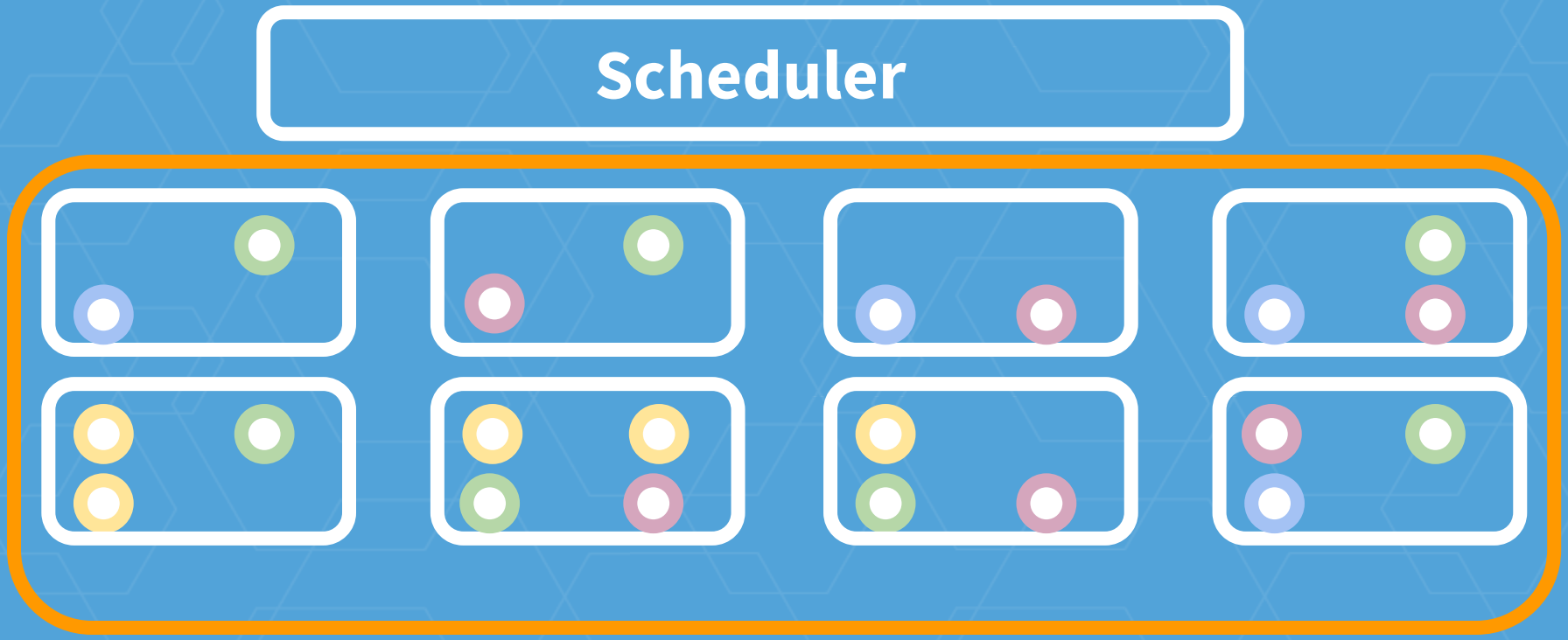
CoreOS

Running containers on a cluster



CoreOS

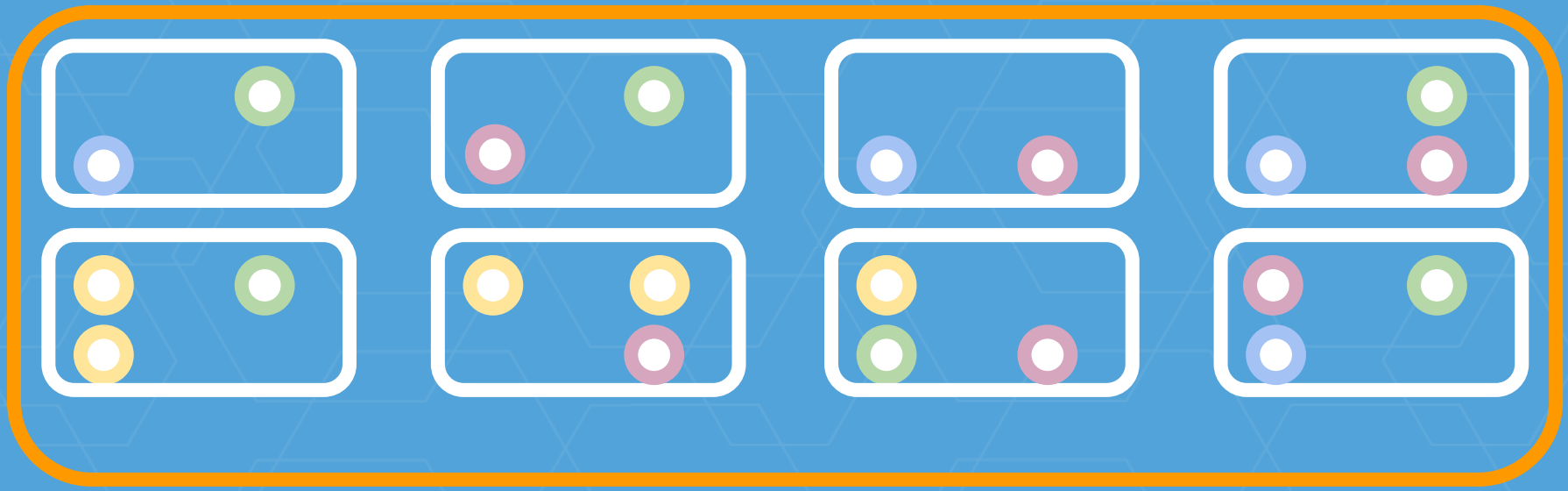
Running containers on a cluster



CoreOS

Running containers on a cluster

color=yellow



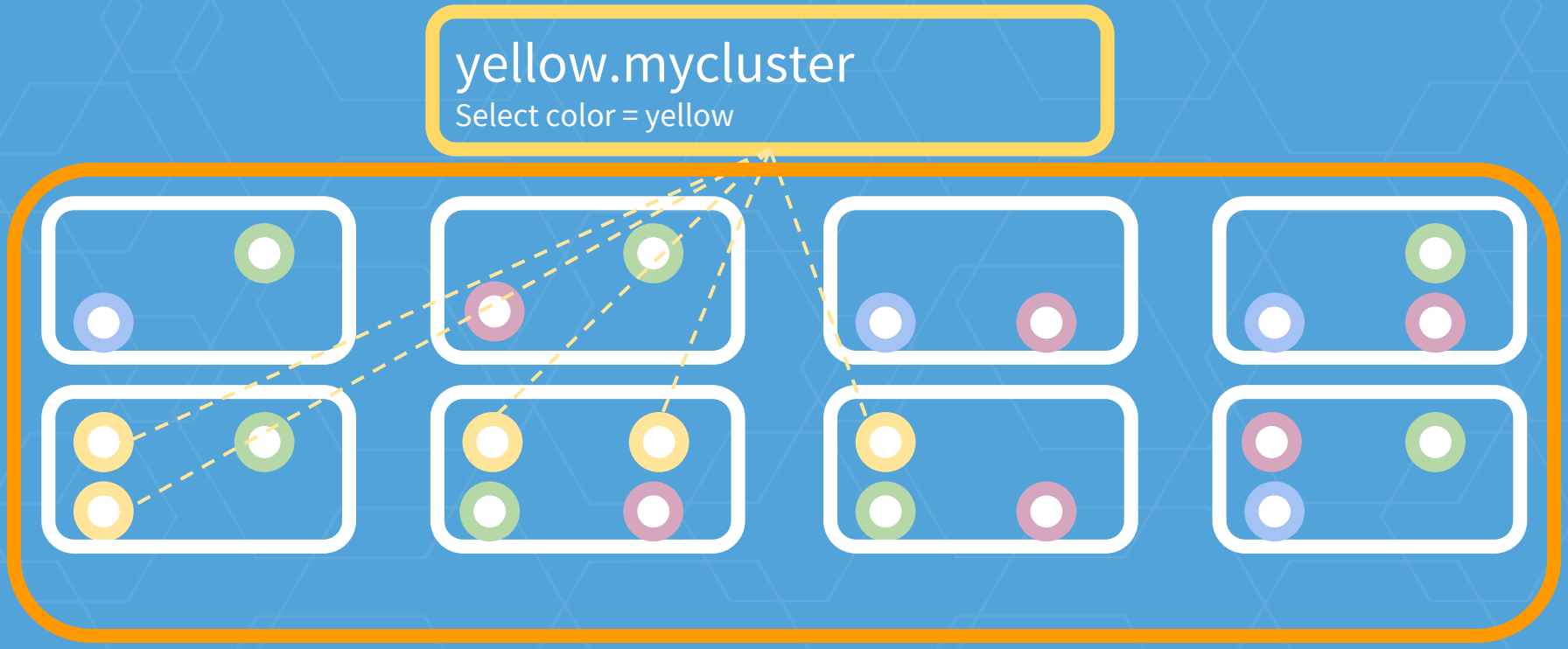
Running containers on a cluster

color=yellow

Select color = yellow

CoreOS

Running containers on a cluster



CoreOS

Running containers on a cluster

- Application deployment
  - Replication controller
  - Rolling upgrades
  - Auto scaling
- Configuration and secret
- Resource management
  - Storage
  - Network

CoreOS

GIFEE

Google infrastructure for everyone else - A flexible, secure, reliable infrastructure for running distributed applications.

- CoreOS

Application developers

I hate dealing with operations!

CoreOS

Application developers

Run my application on a “cluster manager” now!

CoreOS

System builders

Trust us!

Operators

Here is Kubernetes!

Operators

Here is ~~Kubernetes!~~

Docker Swarm

DC/OS

But …

Can your applications run smoothly on the cluster manager?

Cloud native application

Distributed
Stateless

Distributed and stateless

Scale

Distributed and stateless

Fault tolerance

CoreOS

Distributed and stateless

Easy to operate

CoreOS

Stateless

No dependency on local resources

Stateless

Treat storage as services

CoreOS

Stateless

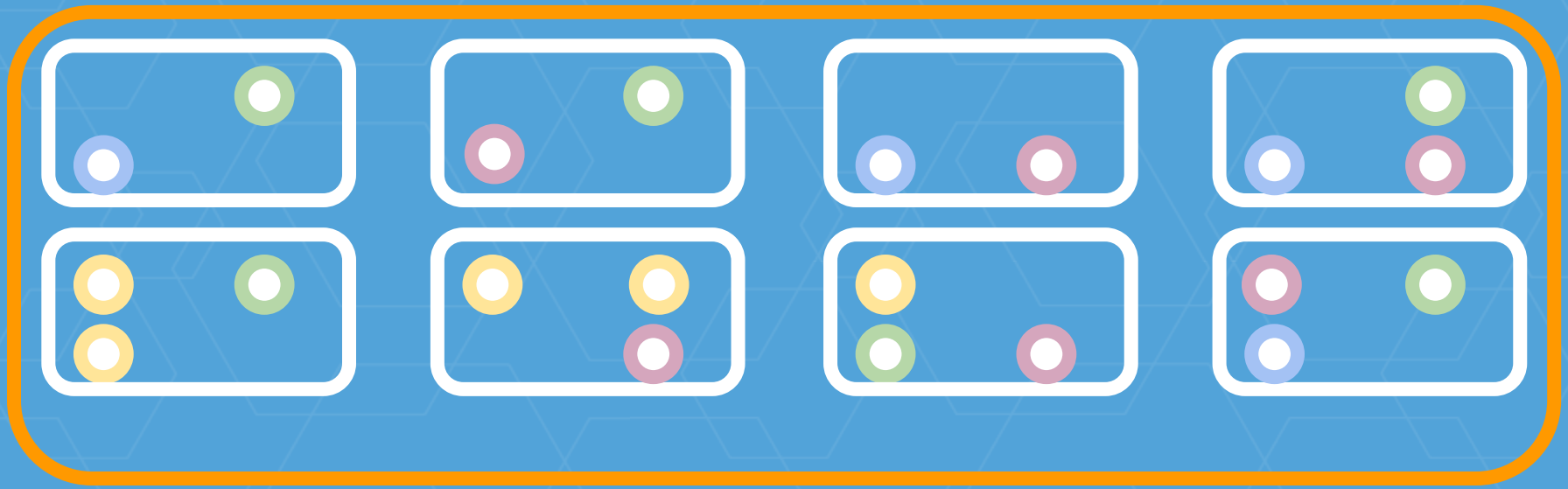
Store configuration in the environment

CoreOS

Managed stateless application

Spec

- Container: example.com/myApp?version=1
- Replica: 3
- Restart: Always

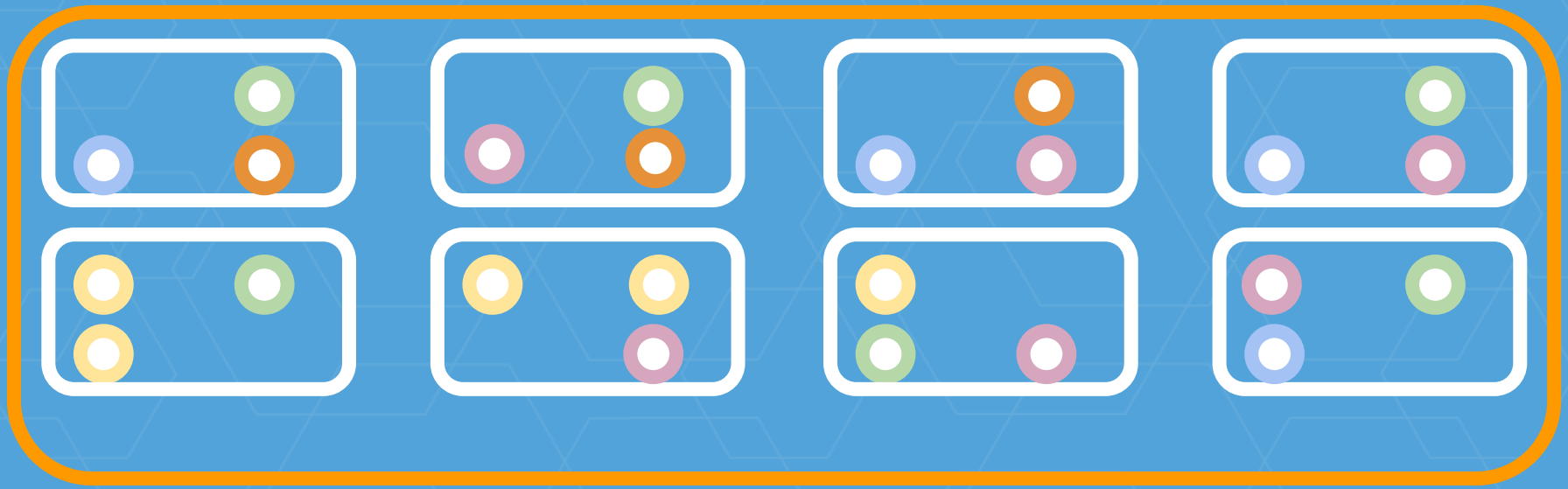


CoreOS

Distributed and stateless

Spec

- Container: example.com/myApp?version=1
- Replica: 3
- Restart: Always

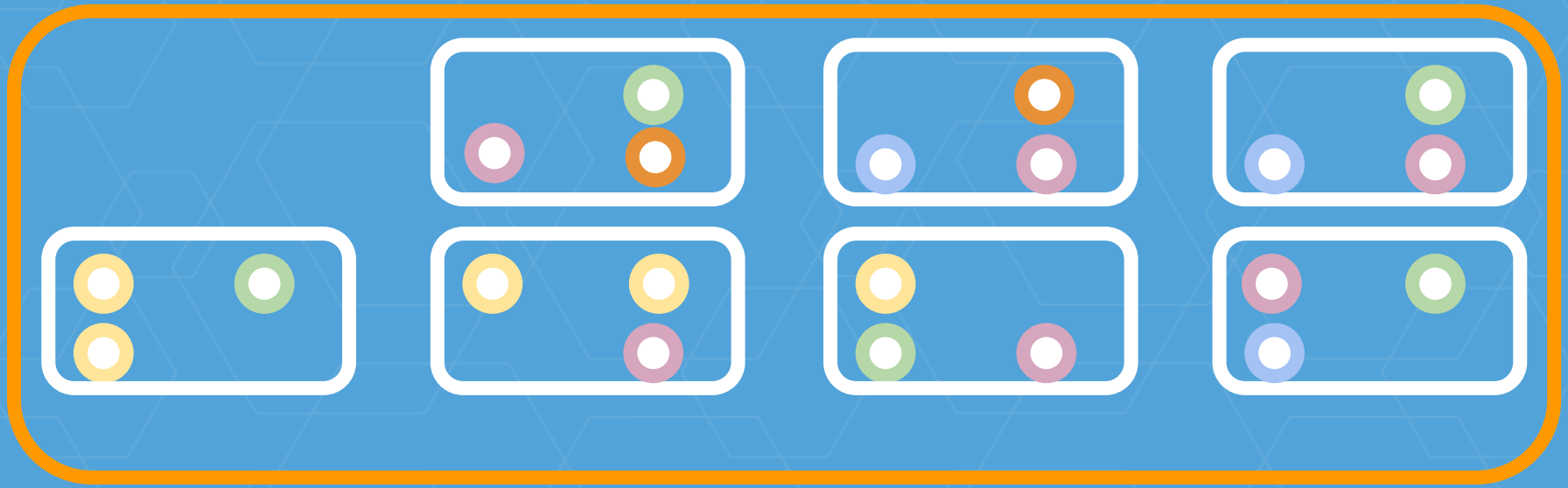


CoreOS

Distributed and stateless

Spec

- Container: example.com/myApp?version=1
- Replica: 3
- Restart: Always

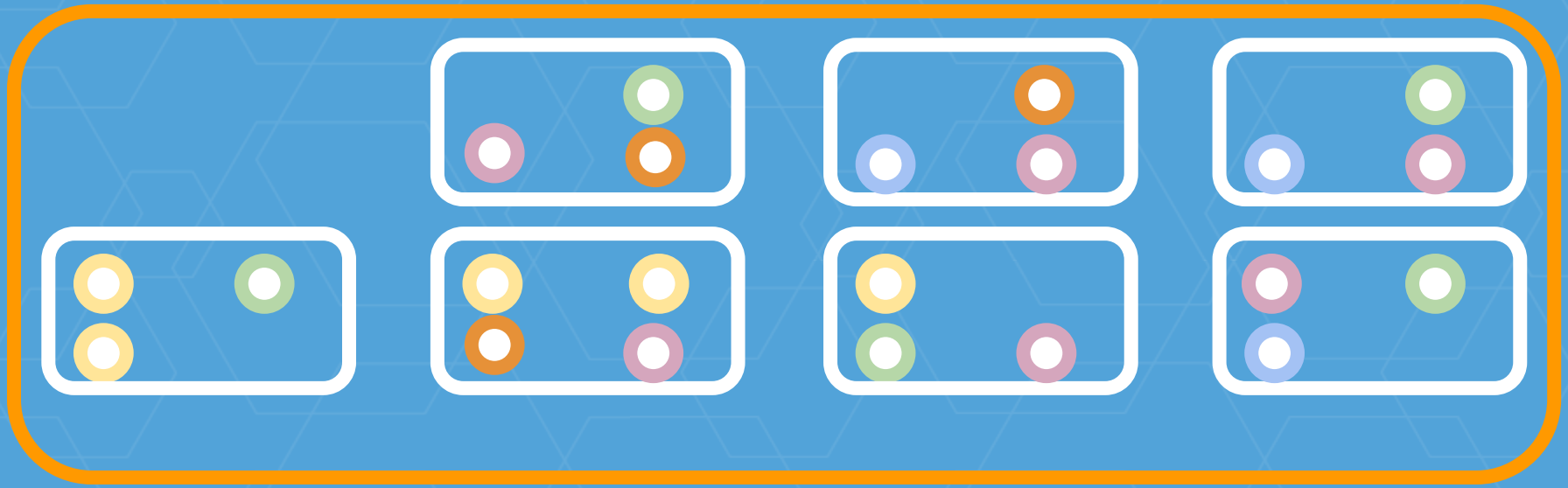


CoreOS

Distributed and stateless

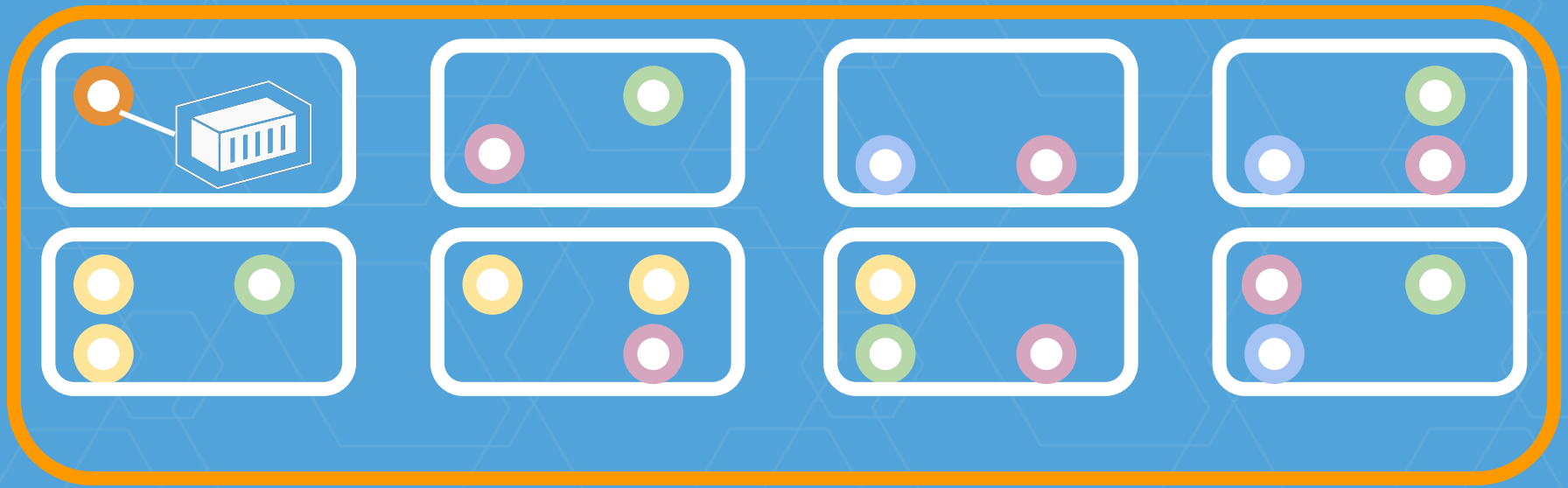
Spec

- Container: example.com/myApp?version=1
- Replica: 3
- Restart: Always



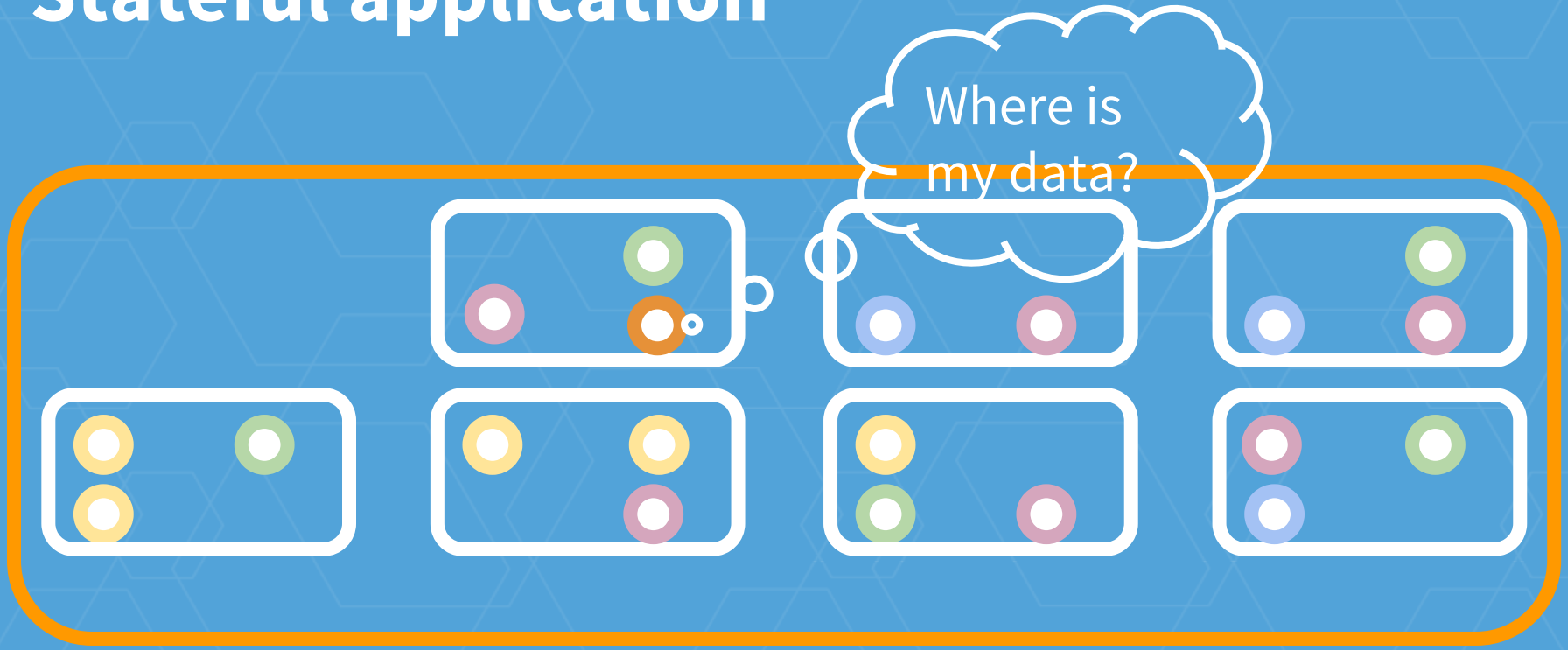
CoreOS

Stateful application



CoreOS

Stateful application



CoreOS

Stateful application

Fully managed by human

Or semi-fully managed by human

CoreOS

Container 2.0

Container 2.0 is the ability to run (and orchestrate) both stateless and stateful services on the same set of resources.

- Mesosphere

CoreOS

Example: manage etcd clusters

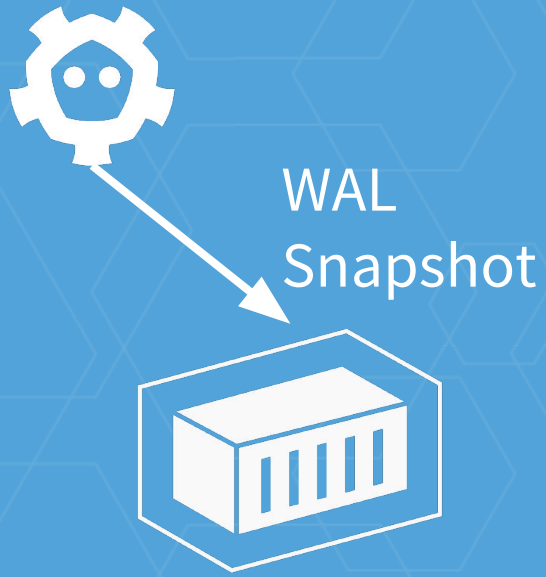
etcd

- Clustered key-value store
- Writes are persisted on disk
- Writes are replicated to all nodes

CoreOS

Example: manage etcd clusters

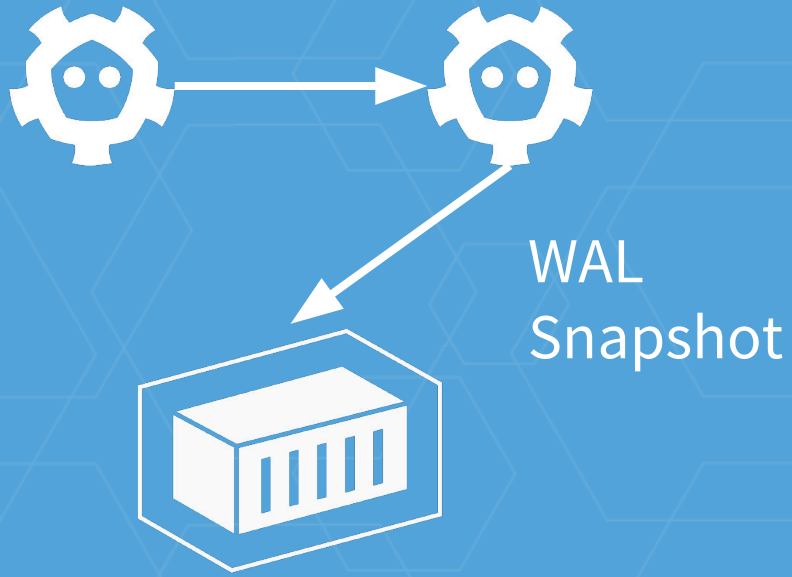
Tightly coupled with local storage



CoreOS

Example: manage etcd clusters

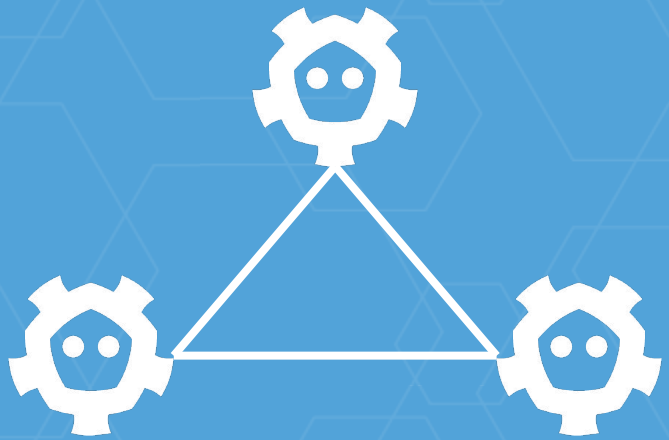
Indirectly coupled with local storage of it peers



CoreOS

Example: manage etcd clusters

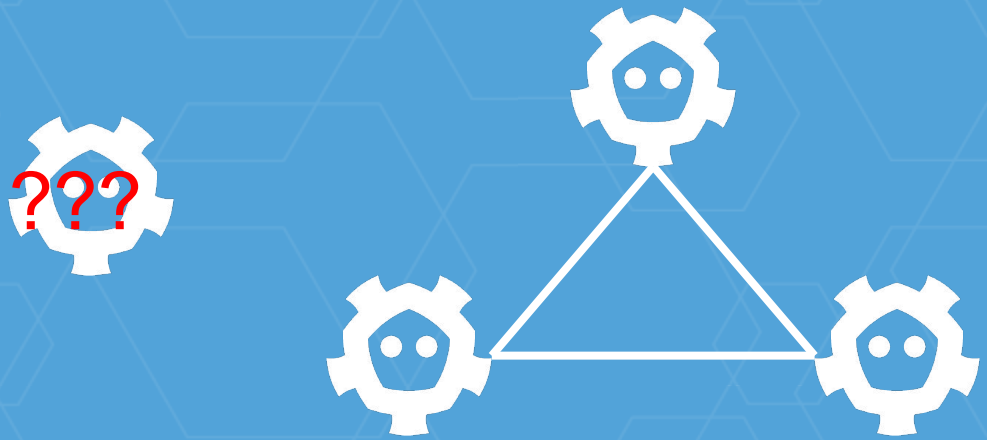
Strong membership



CoreOS

Example: manage etcd clusters

Strong membership



CoreOS

Example: manage etcd clusters

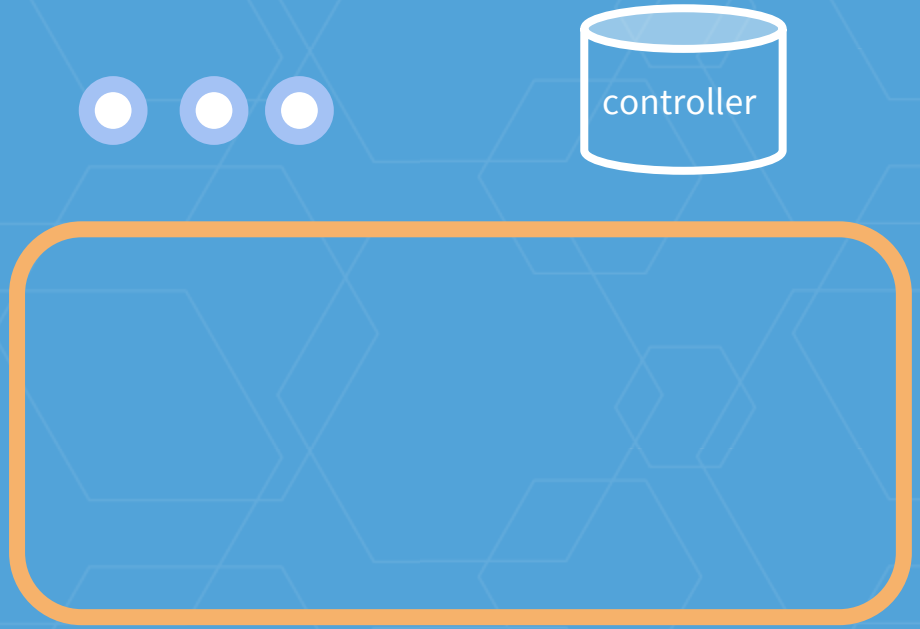
External controller

- A big for loop to simulate human operator
  - Manage data migration
  - Manage membership changes
  - Manage configuration changes

CoreOS

Example: manage etcd clusters

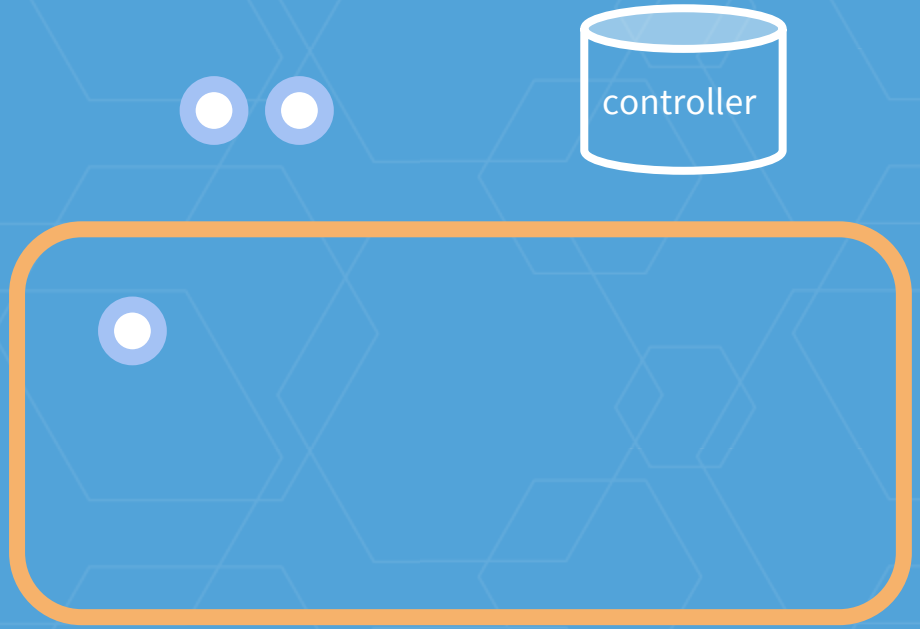
Bootstrap a 3 member cluster



CoreOS

Example: manage etcd clusters

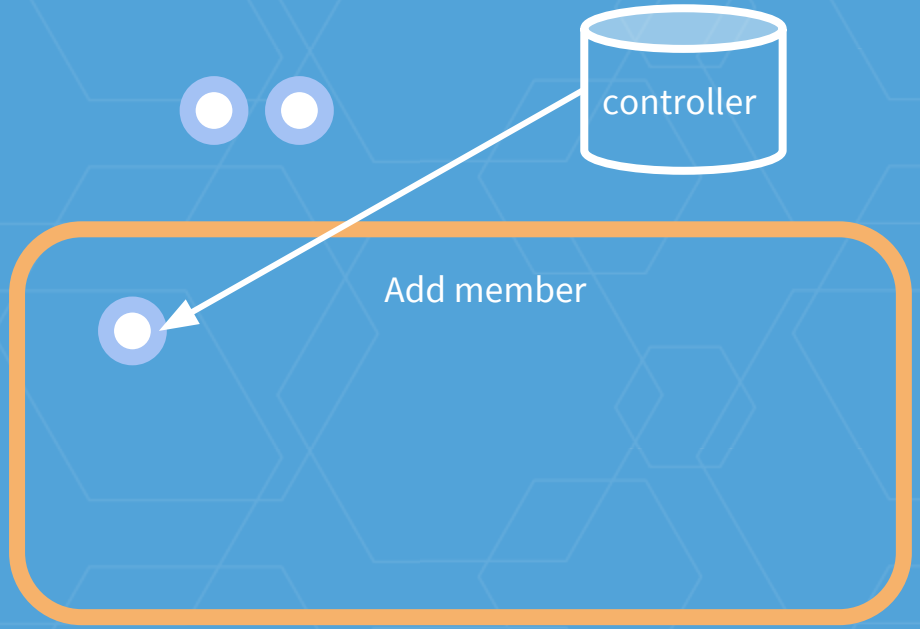
Bootstrap a 3 member cluster



CoreOS

Example: manage etcd clusters

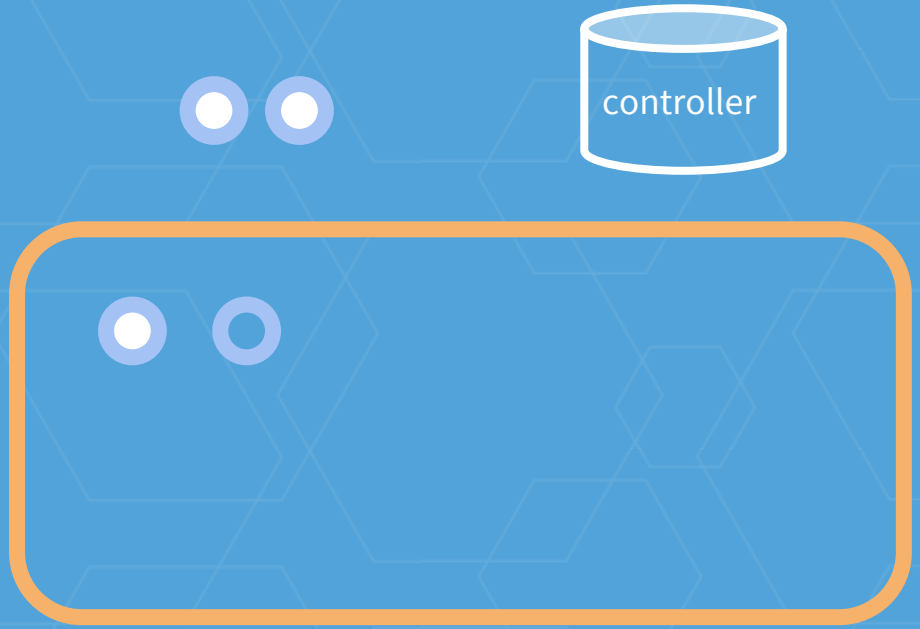
Bootstrap a 3 member cluster



CoreOS

Example: manage etcd clusters

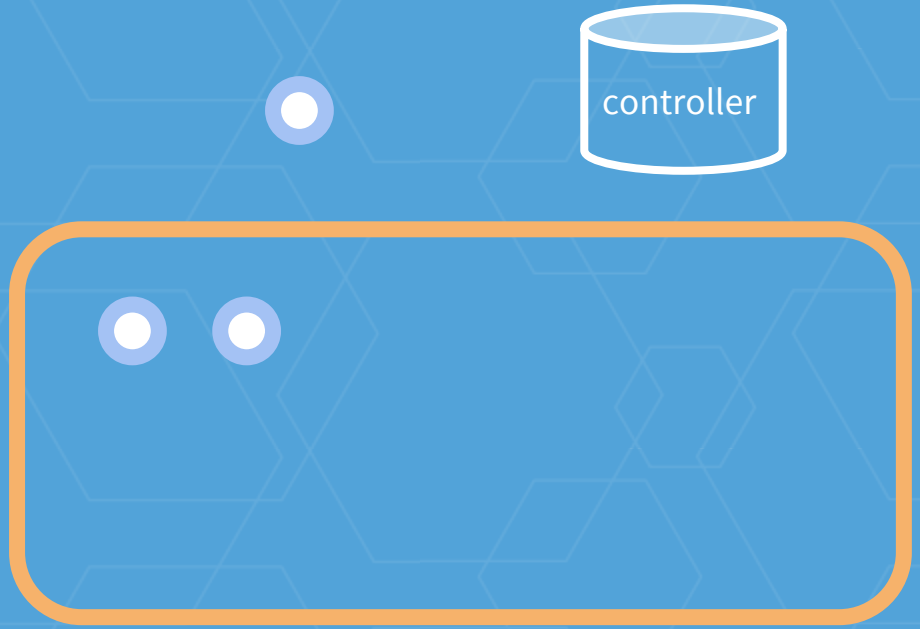
Bootstrap a 3 member cluster



CoreOS

Example: manage etcd clusters

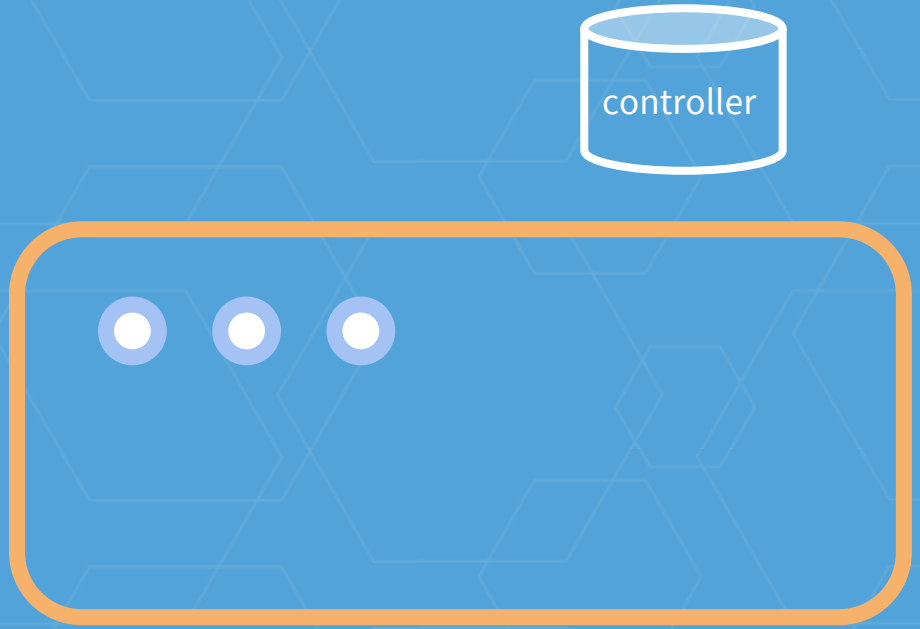
Bootstrap a 3 member cluster



CoreOS

Example: manage etcd clusters

Bootstrap a 3 member cluster

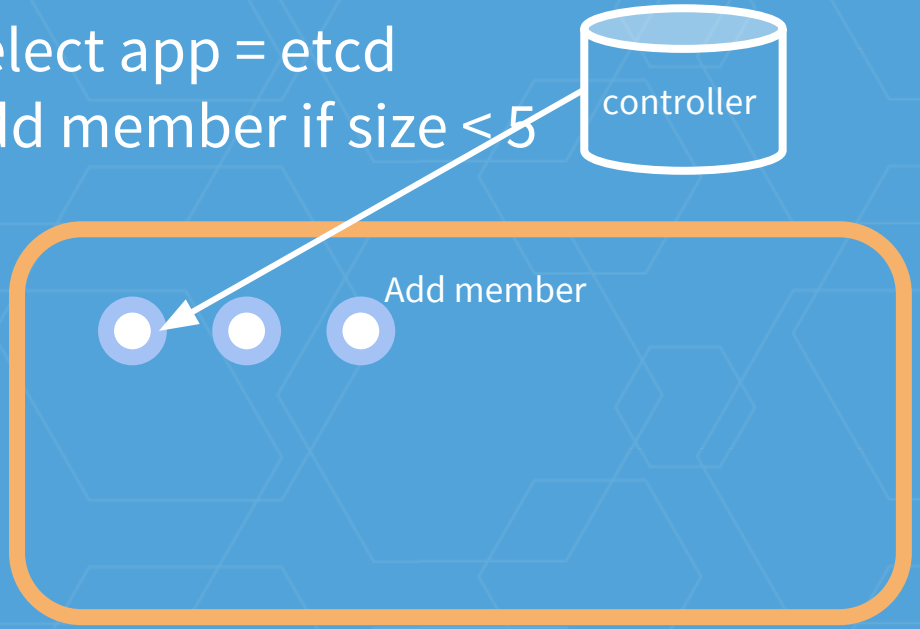


CoreOS

Example: manage etcd clusters

Resize from 3 to 5

Select app = etcd
Add member if size < 5

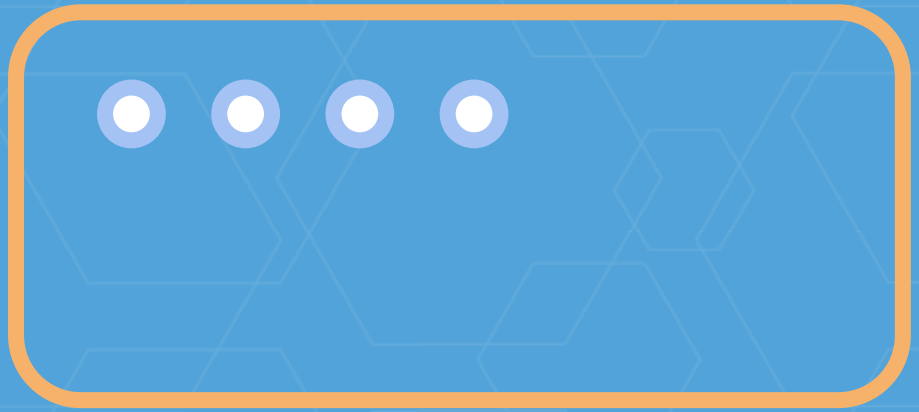


CoreOS

Example: manage etcd clusters

Resize from 3 to 5

Select app = etcd
Add member if size < 5

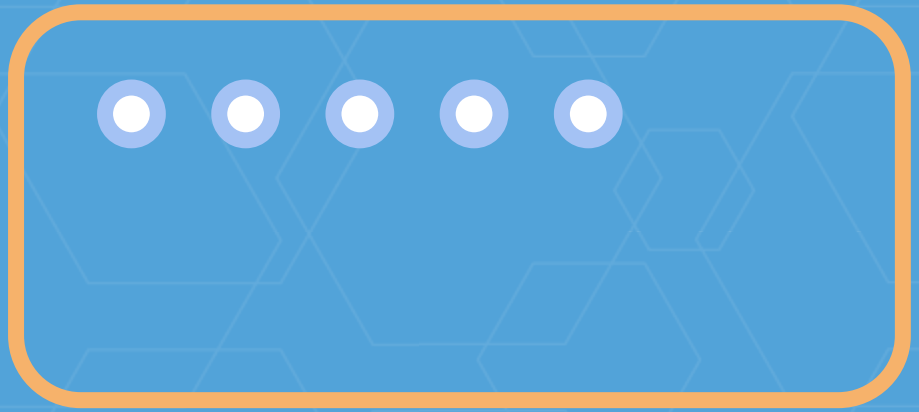


CoreOS

Example: manage etcd clusters

Resize from 3 to 5

Select app = etcd
Add member if size < 5

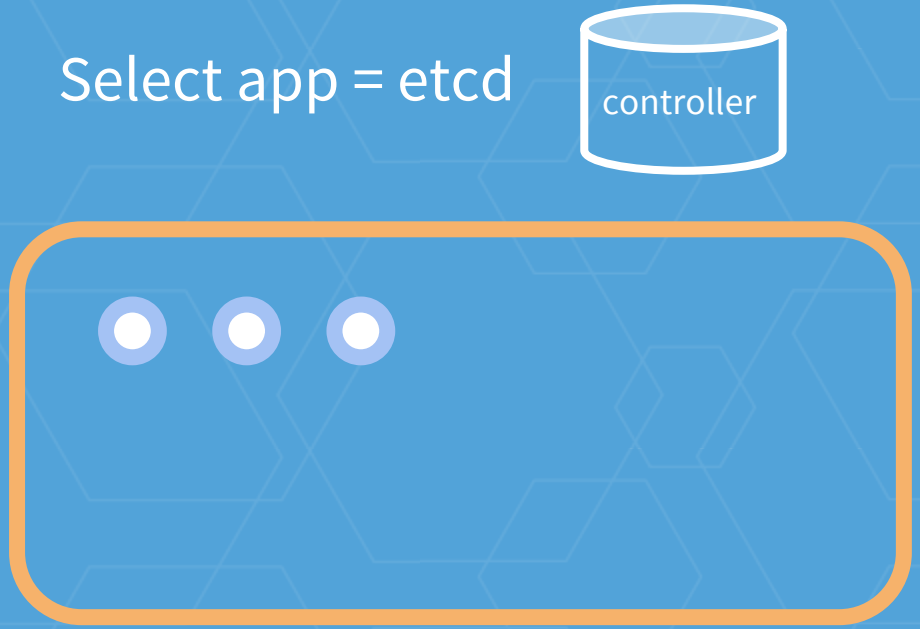


CoreOS

Example: manage etcd clusters

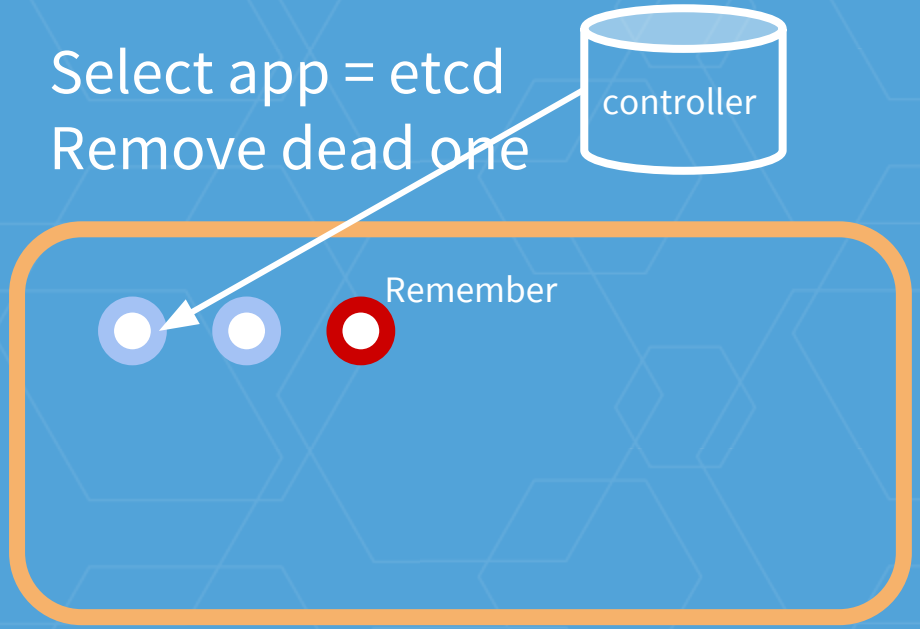
Failure recovery

Select app = etcd



Example: manage etcd clusters

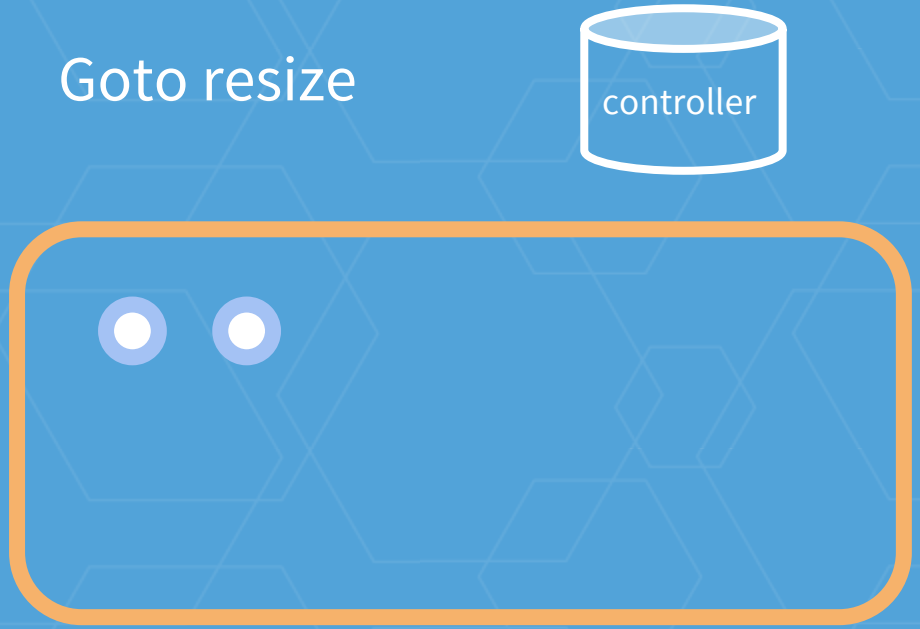
Failure recovery



Example: manage etcd clusters

Failure recovery

Goto resize

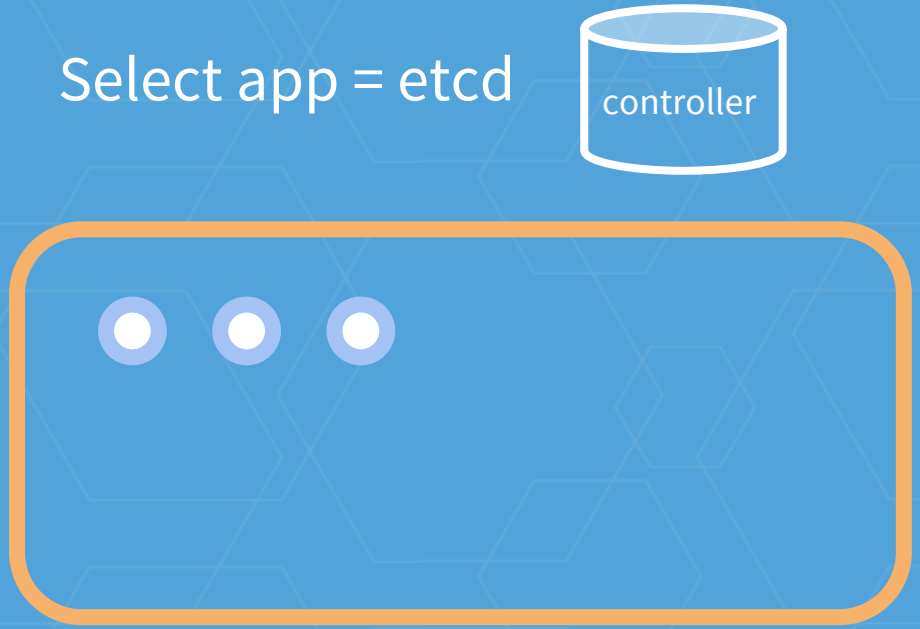


CoreOS

Example: manage etcd clusters

Failure recovery

Select app = etcd

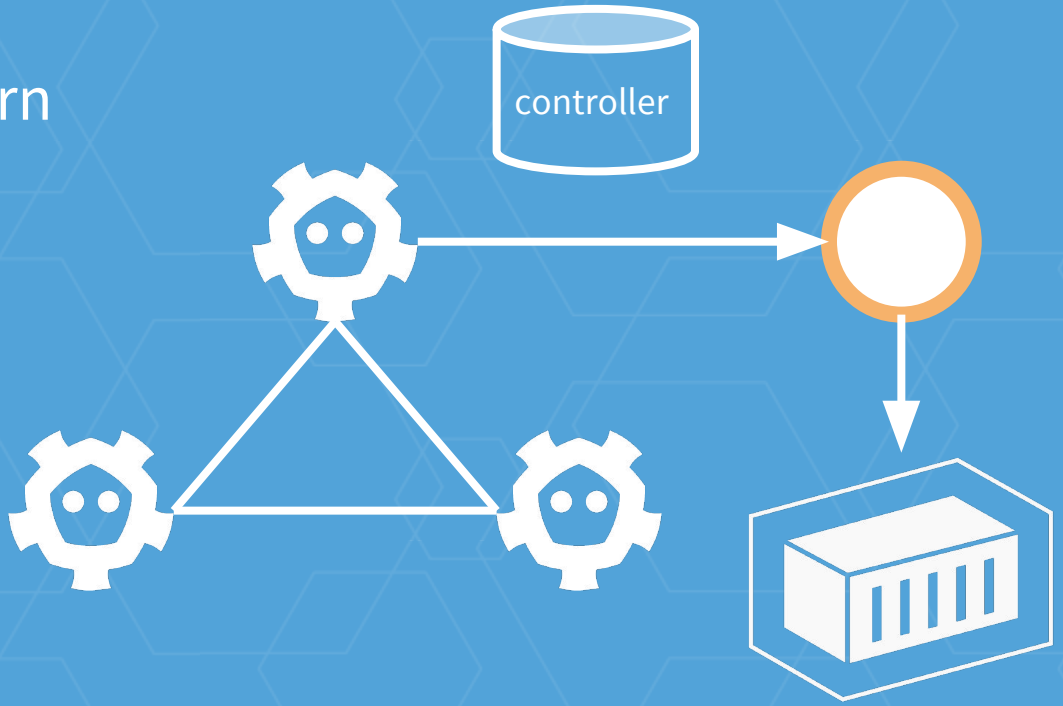


CoreOS

Example: manage etcd clusters

Backup

- Sidecar pattern



CoreOS

Stateful distributed application

More controllers to replace human

Tips for building distributed application

CoreOS

Use coordination libraries/softwares

- Leader election
- Locking
- Queue
- Barrier

Tips for building distributed application

CoreOS

Use a RPC framework

- strict contract between components
- across languages support
- client side load-balancing
- client side naming resolution
- auth

Tips for building distributed application

CoreOS

Metrics

- Request rate/duration
- Error rate
- Internal state

Tips for building distributed application

CoreOS

Logging

- Human actionable events
- Critical State changes

CoreOS

Future is bright