While you're waiting...

http://bit.ly/srecon-cake0

Kubernetes From A Cake Mix

—

Liz Frost

Who am I?

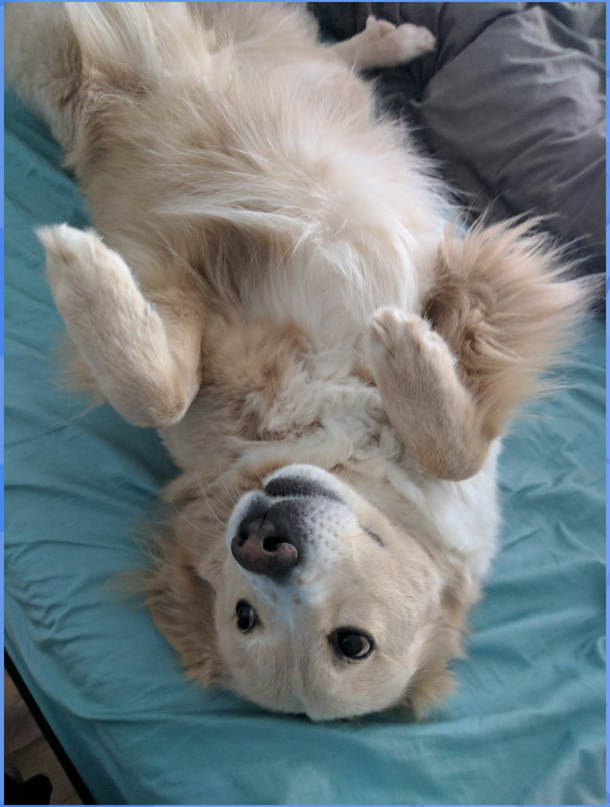
Liz Frost

Twitter: @stillinbeta

slack.k8s.io: @liz

Who am I?

Who am I?

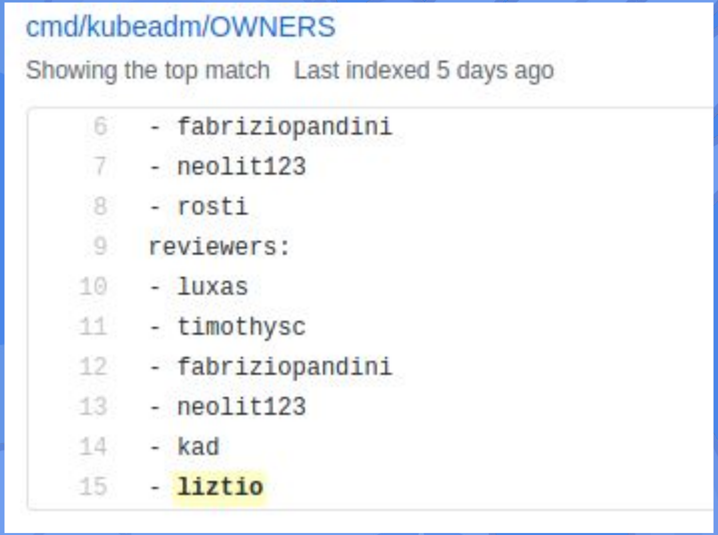


Who am I?

It’s pronounced
“kube-cuddle”

Why should you listen to me?

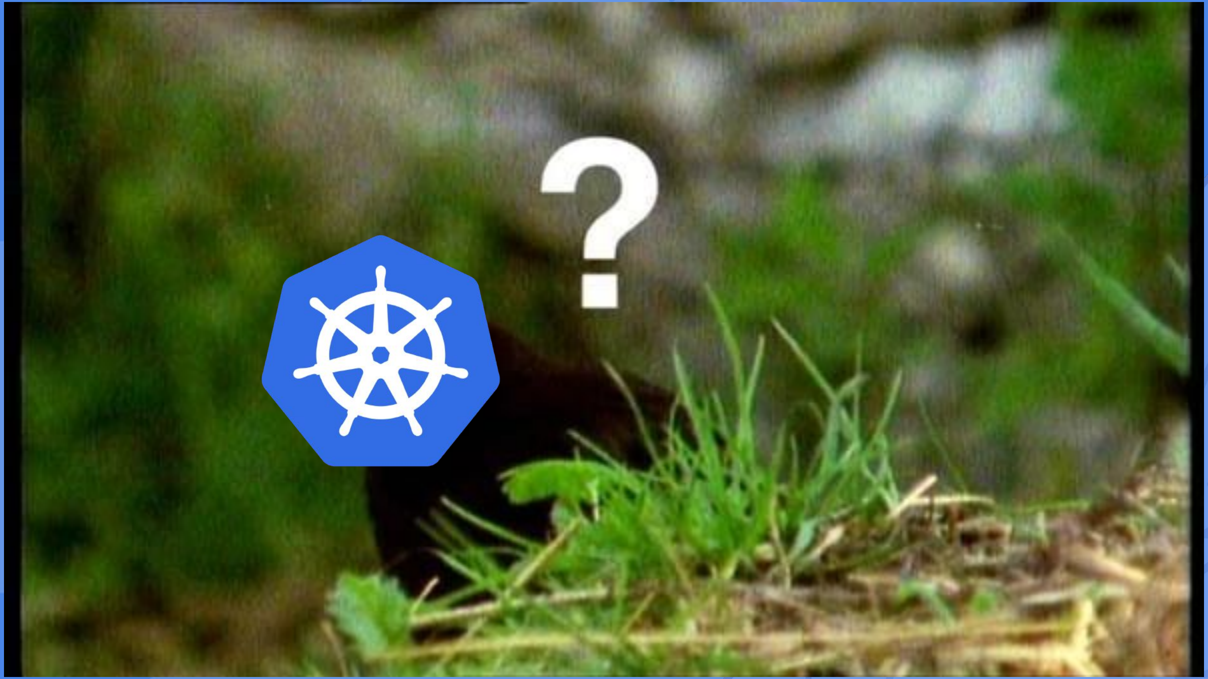
- sig-cluster-lifecycle member
- kubernetes contributor
- kubeadm developer



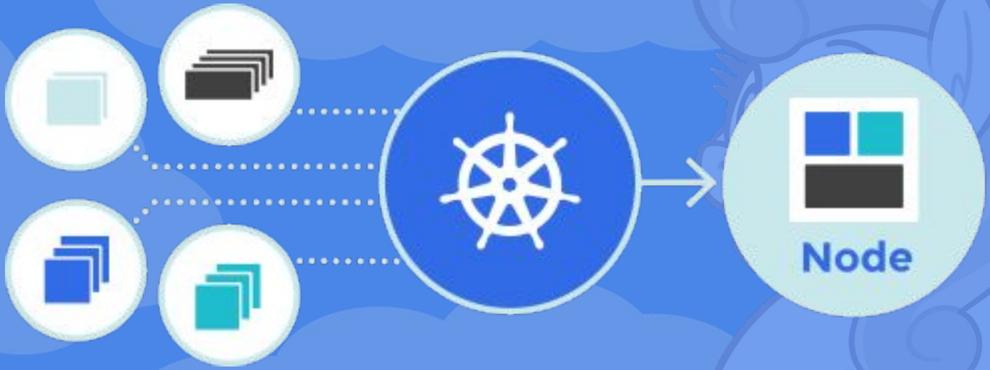
cmd/kubeadm/OWNERS

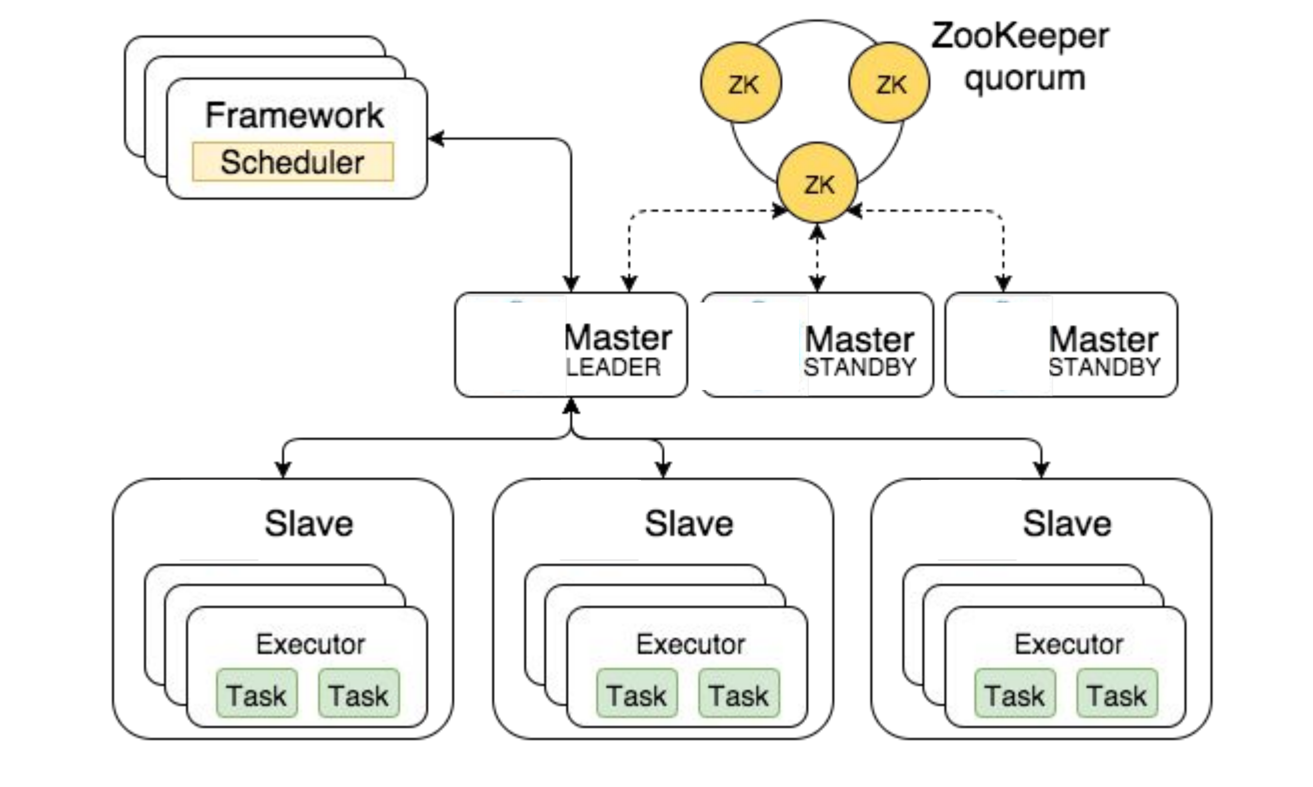
Showing the top match Last indexed 5 days ago

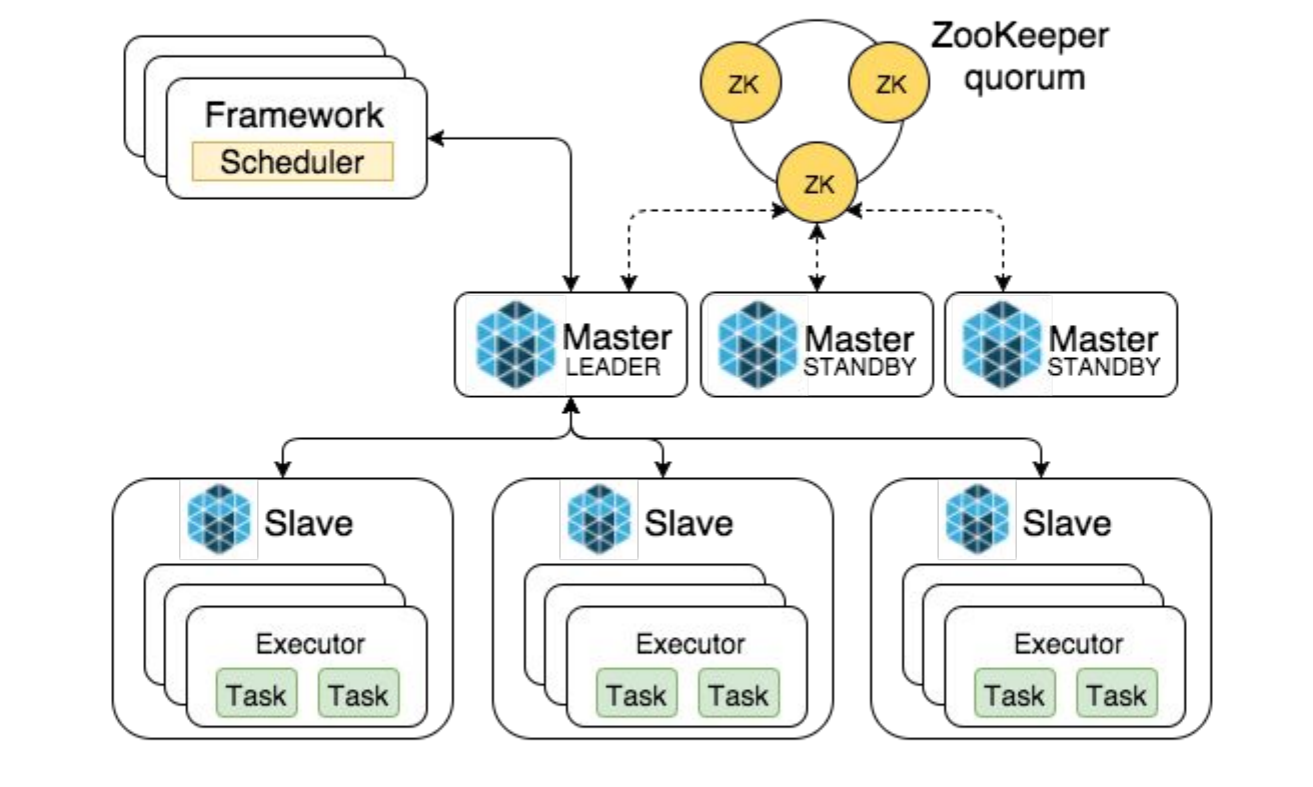
```
  - fabriziopandini
  - neolit123
  - rosti
reviewers:
  - luxas
  - timothysc
  - fabriziopandini
  - neolit123
  - kad
  - liztio
```

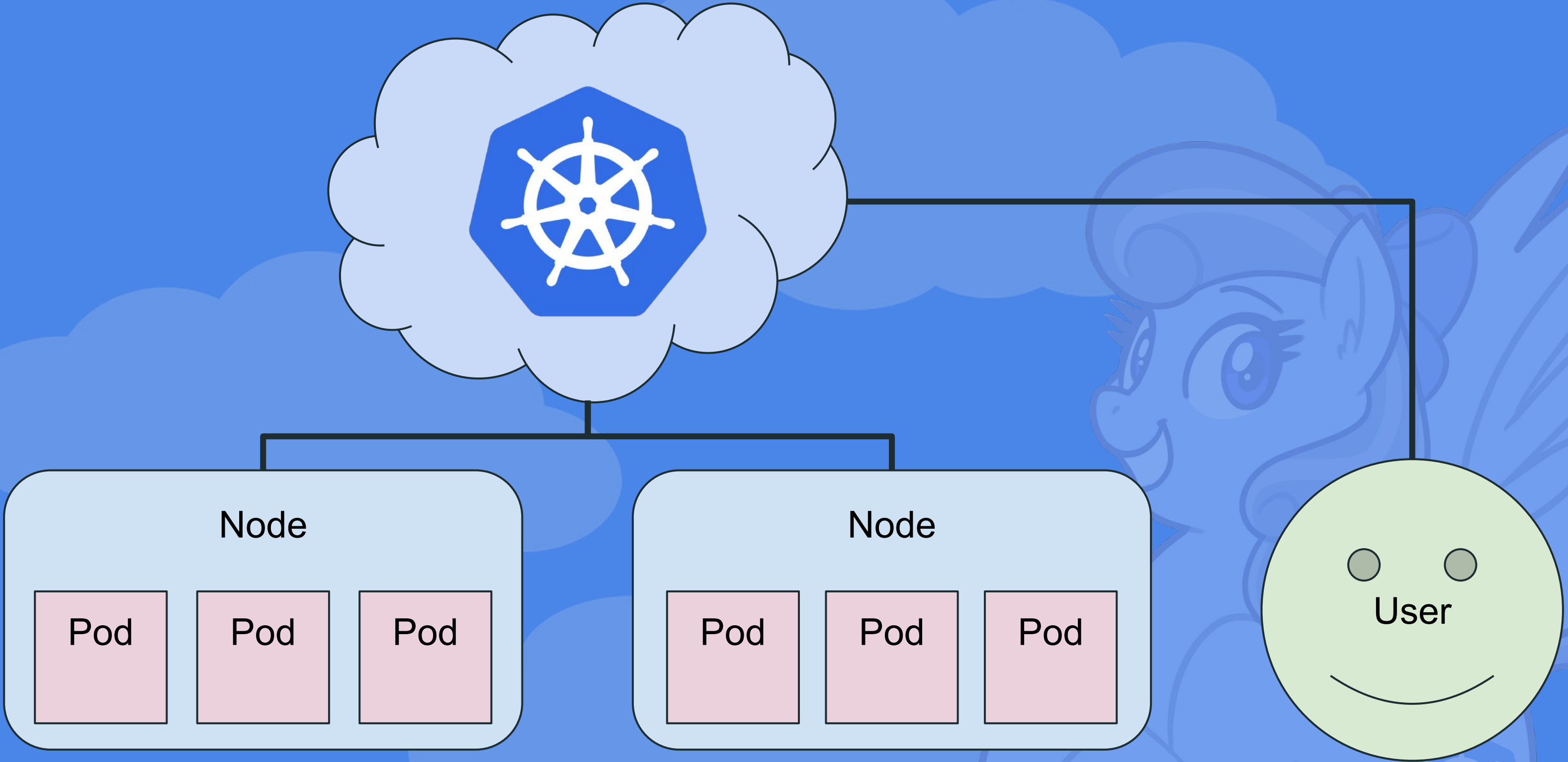


What *is* Kubernetes?


Node


Framework
Scheduler
ZK
ZK
ZK
ZooKeeper quorum
Master
LEADER
Master
STANDBY
Master
STANDBY
Slave
Executor
Task
Task
Slave
Executor
Task
Task
Slave
Executor
Task
Task


Framework
Scheduler
ZooKeeper quorum
ZK
ZK
ZK
Master LEADER
Master STANDBY
Master STANDBY
Slave
Executor
Task
Task
Slave
Executor
Task
Task
Slave
Executor
Task
Task



Node

Pod

Pod

Pod

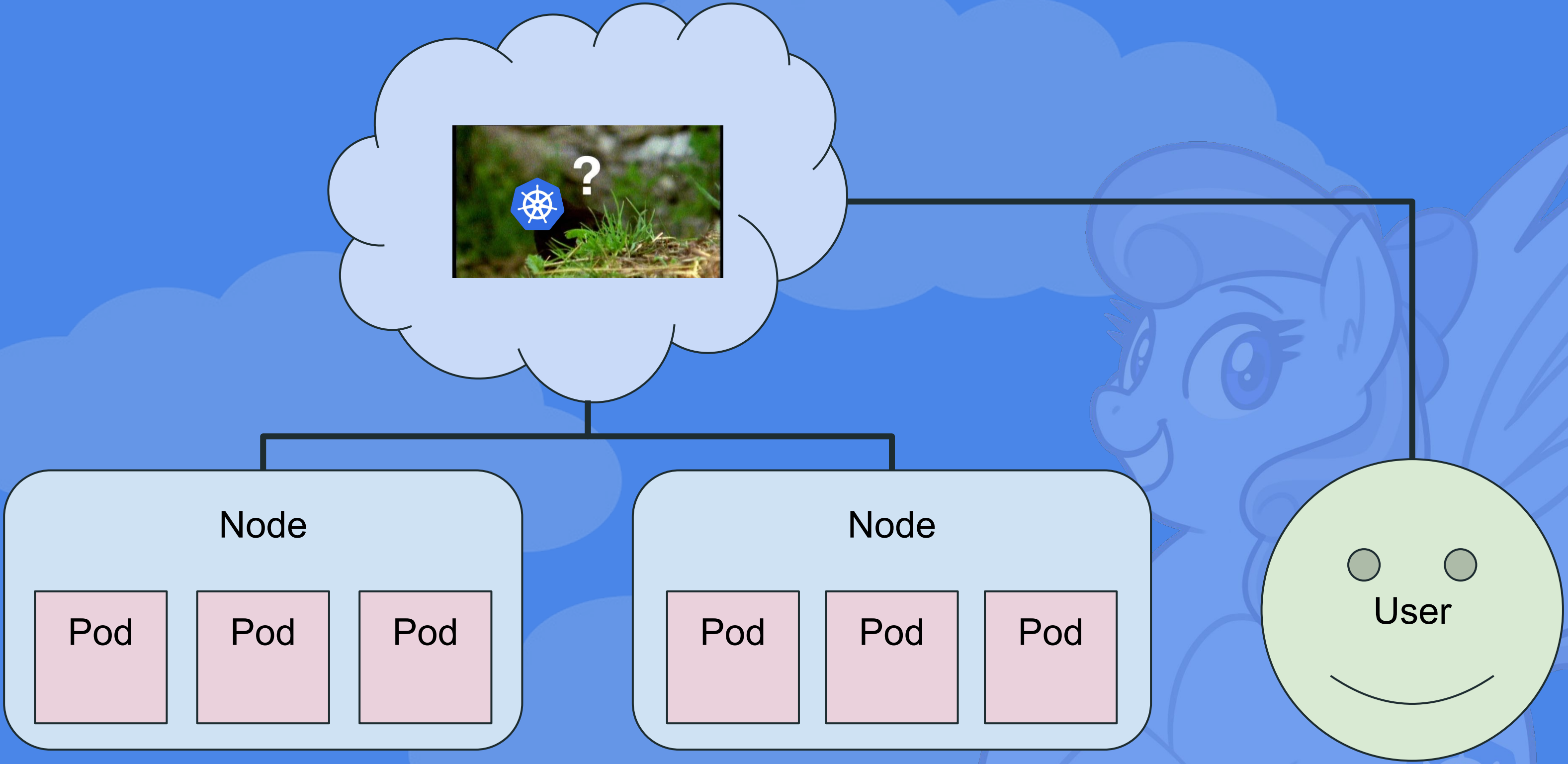
Node

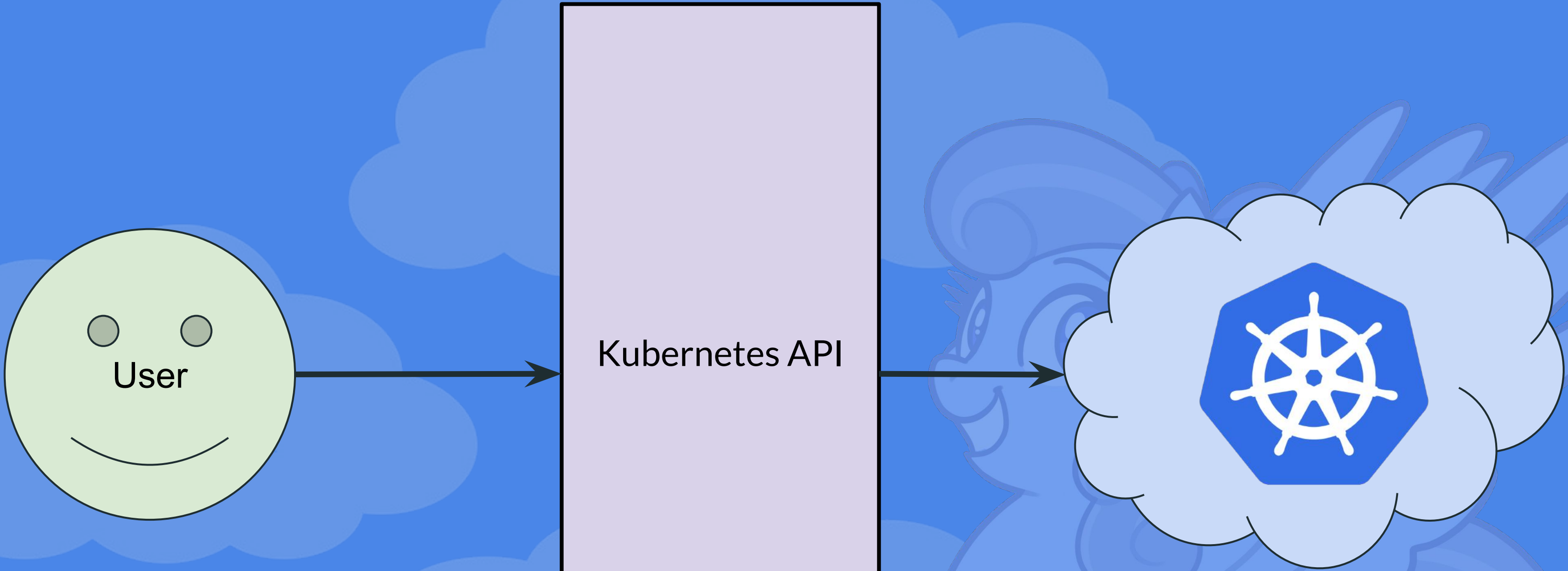
Pod

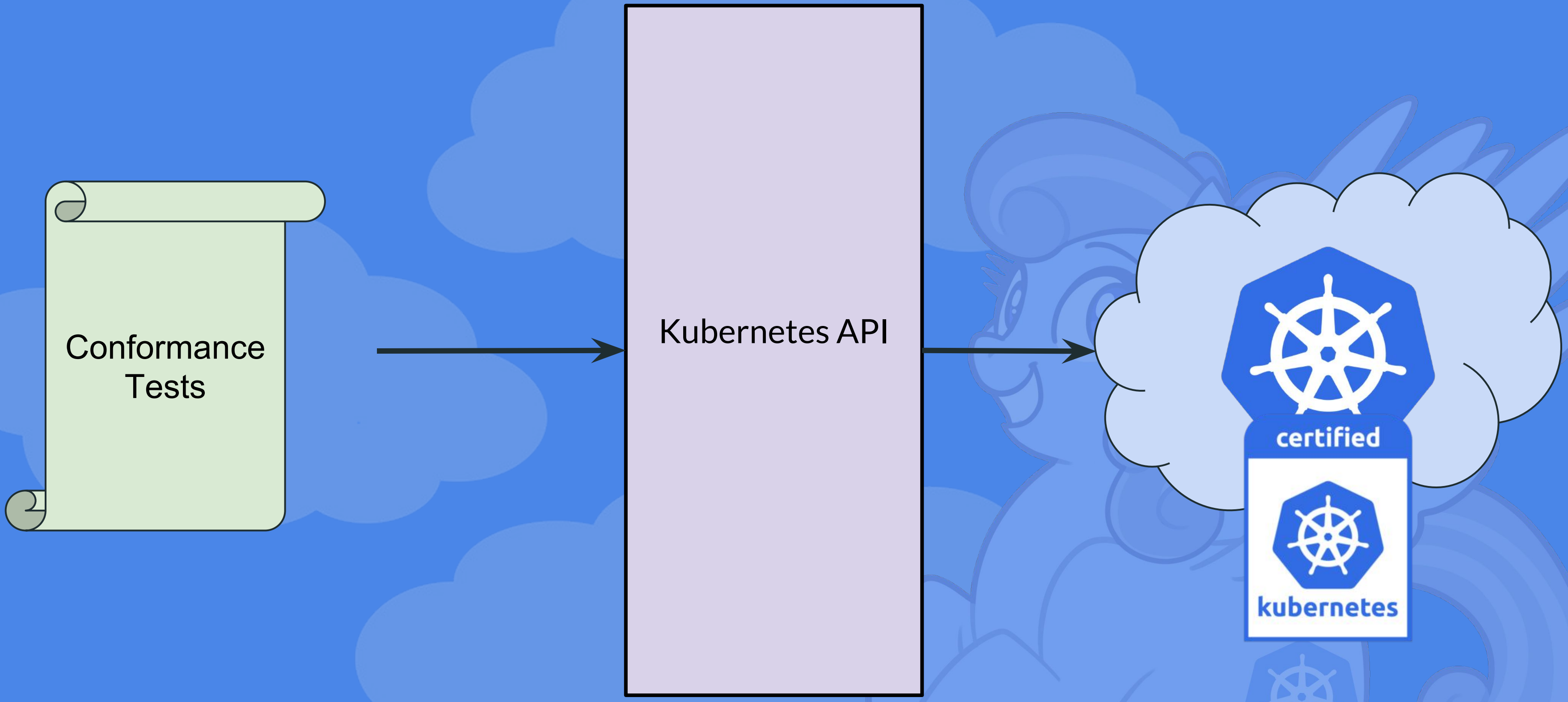
Pod

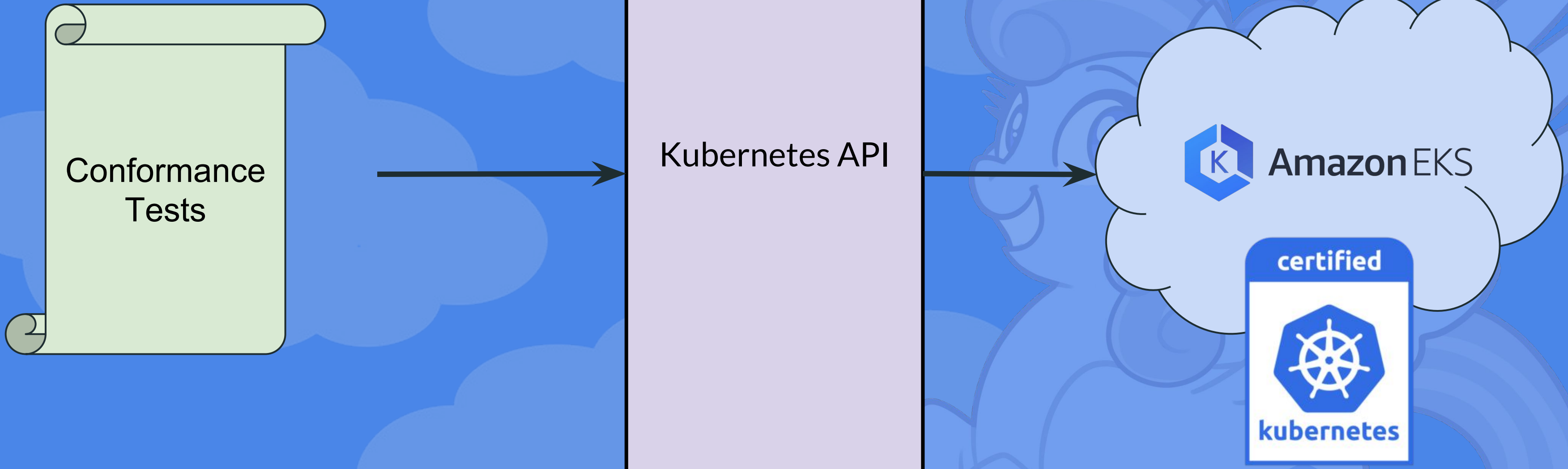
Pod

User


?
Node
Pod
Pod
Pod
Node
Pod
Pod
Pod
User

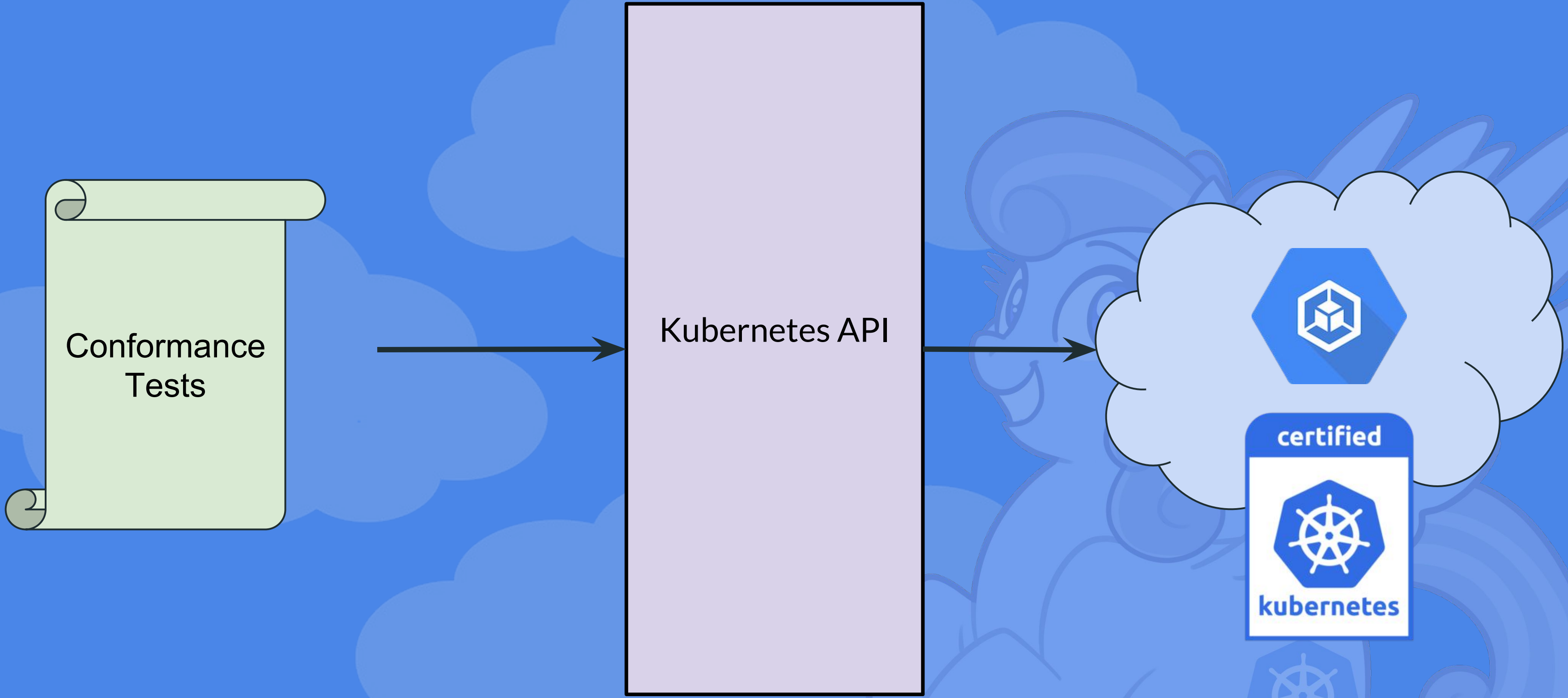

User
Kubernetes API

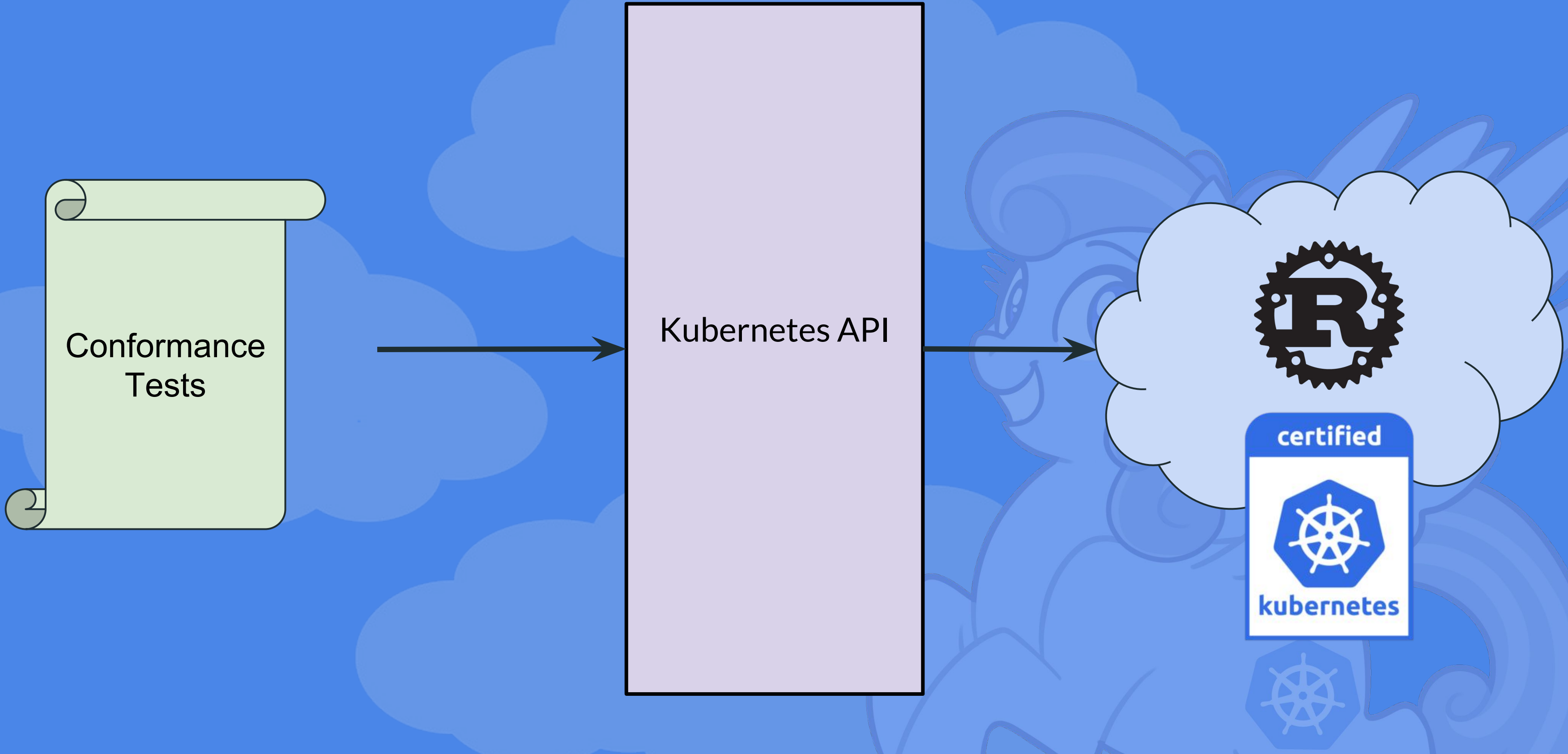

Conformance Tests
Kubernetes API
certified
kubernetes



Conformance Tests

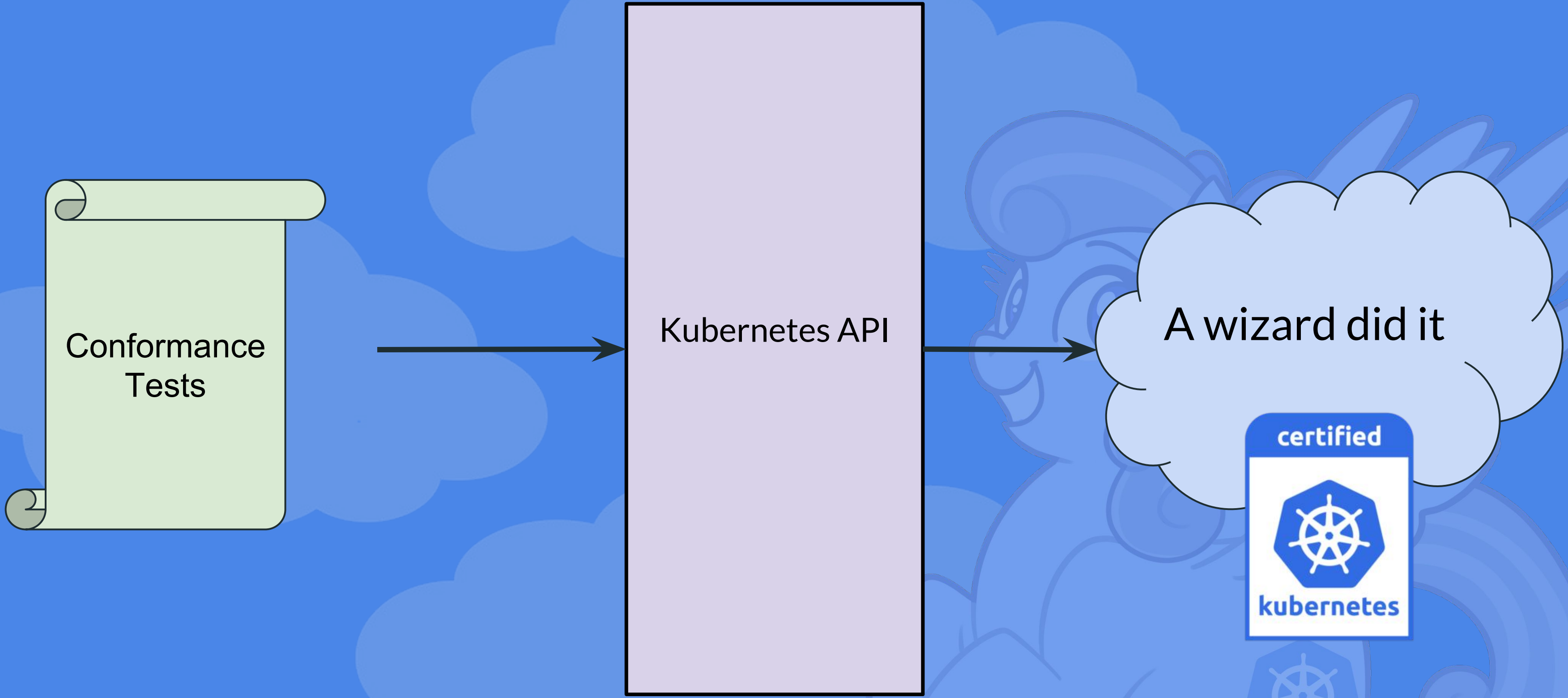
Kubernetes API


Conformance Tests
Kubernetes API
certified
kubernetes



Conformance Tests

Kubernetes API



Conformance Tests
Kubernetes API
A wizard did it
certified
kubernetes

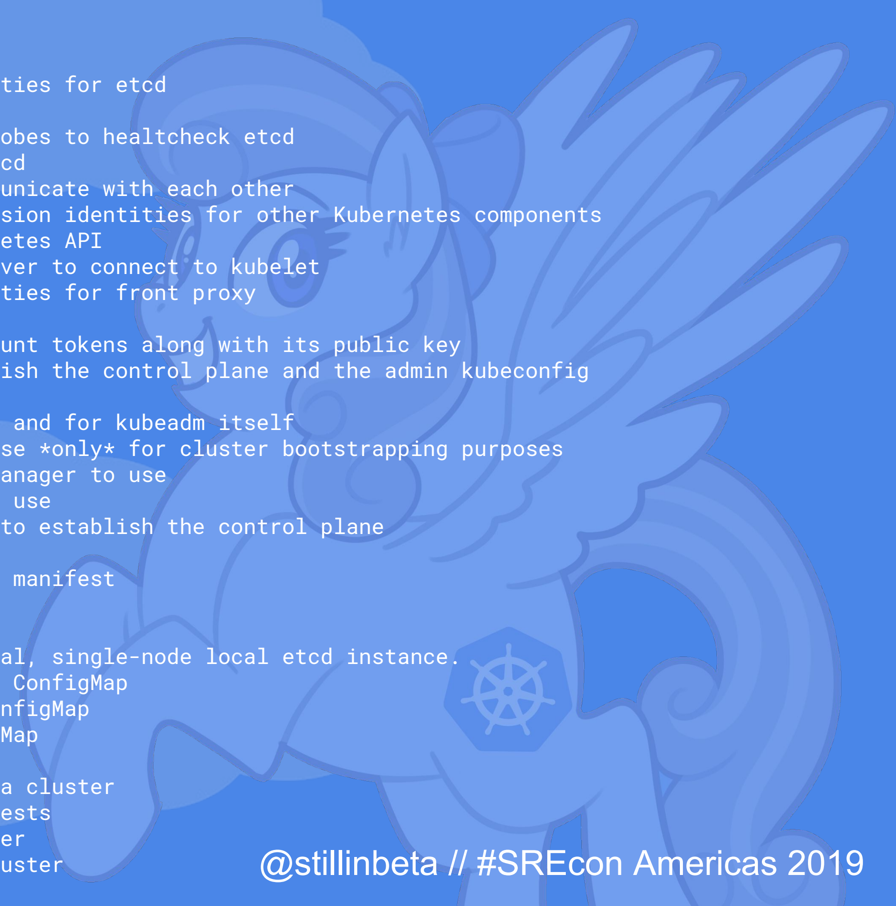
We got Opinions

Kubeadm™ brand Kubernetes

```
kubeadm init
```

```
$ kubeadm init --help
Run this command in order to set up the Kubernetes master.

The "init" command executes the following phases:
```
preflight   Run master pre-flight checks
kubelet-start   Writes kubelet settings and (re)starts the kubelet
certs   Certificate generation
  /etcd-ca   Generates the self-signed CA to provision identities for etcd
  /etcd-server   Generates the certificate for serving etcd
  /etcd-healthcheck-client   Generates the client certificate for liveness probes to healtcheck etcd
  /apiserver-etcd-client   Generates the client apiserver uses to access etcd
  /etcd-peer   Generates the credentials for etcd nodes to communicate with each other
  /ca   Generates the self-signed Kubernetes CA to provision identities for other Kubernetes components
  /apiserver   Generates the certificate for serving the Kubernetes API
  /apiserver-kubelet-client   Generates the Client certificate for the API server to connect to kubelet
  /front-proxy-ca   Generates the self-signed CA to provision identities for front proxy
  /front-proxy-client   Generates the client for the front proxy
  /sa   Generates a private key for signing service account tokens along with its public key
kubeconfig   Generates all kubeconfig files necessary to establish the control plane and the admin kubeconfig
file
  /admin   Generates a kubeconfig file for the admin to use and for kubeadm itself
  /kubelet   Generates a kubeconfig file for the kubelet to use *only* for cluster bootstrapping purposes
  /controller-manager   Generates a kubeconfig file for the controller manager to use
  /scheduler   Generates a kubeconfig file for the scheduler to use
control-plane   Generates all static Pod manifest files necessary to establish the control plane
  /apiserver   Generates the kube-apiserver static Pod manifest
  /controller-manager   Generates the kube-controller-manager static Pod manifest
  /scheduler   Generates the kube-scheduler static Pod manifest
etcd   Generates static Pod manifest file for local etcd.
  /local   Generates the static Pod manifest file for a local, single-node local etcd instance.
upload-config   Uploads the kubeadm and kubelet configuration to a ConfigMap
  /kubeadm   Uploads the kubeadm ClusterConfiguration to a ConfigMap
  /kubelet   Uploads the kubelet component config to a ConfigMap
mark-control-plane   Mark a node as a control-plane
bootstrap-token   Generates bootstrap tokens used to join a node to a cluster
addon   Installs required addons for passing Conformance tests
  /coredns   Installs the CoreDNS addon to a Kubernetes cluster
  /kube-proxy   Installs the kube-proxy addon to a Kubernetes cluster
```


```

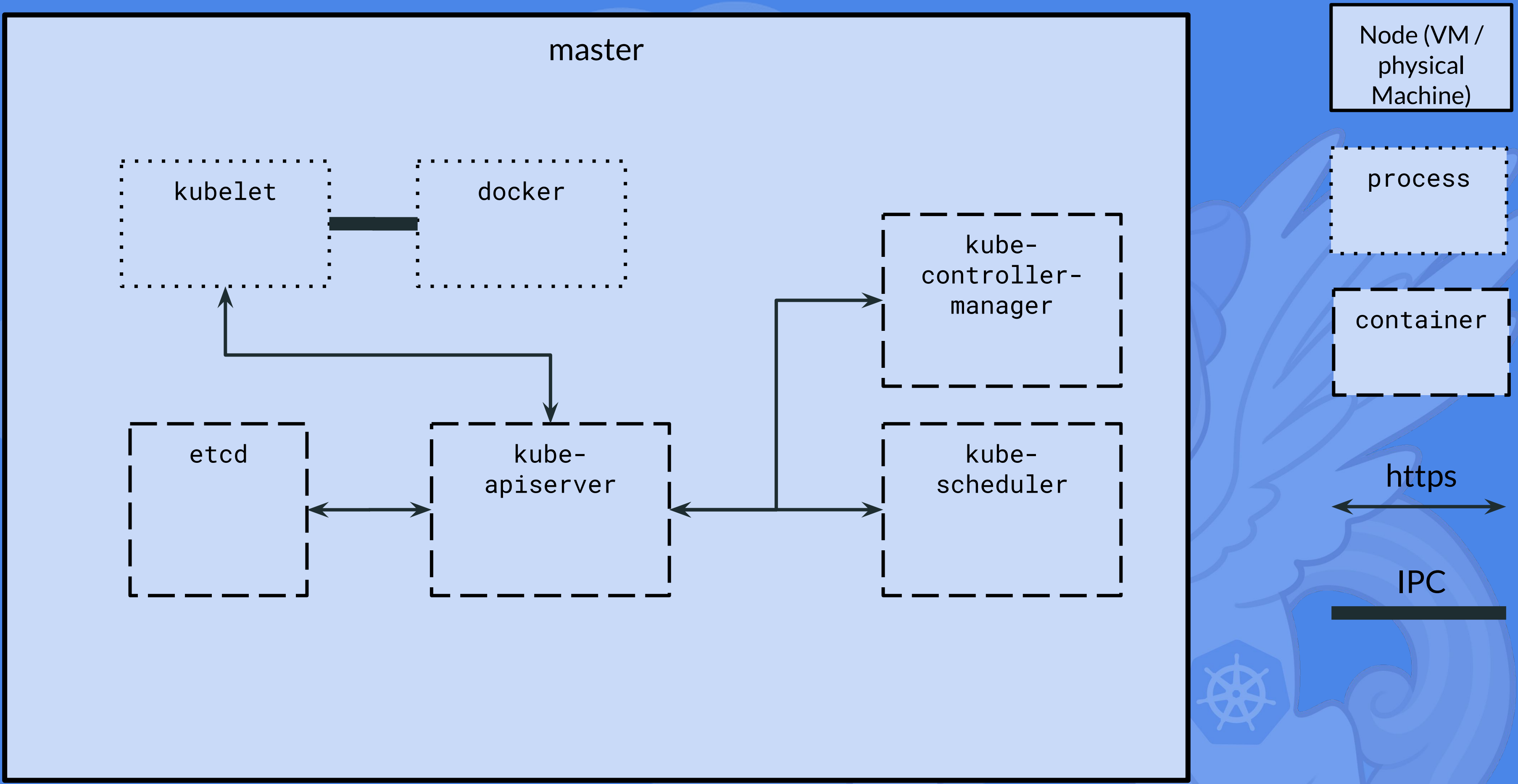
~~~
$ kubeadm init --help
Run this command in order to set up the Kubernetes master.

The "init" command executes the following phases:
```
preflight                  Run master pre-flight checks
kubelet-start              Writes kubelet settings and (re)starts the kubelet
certs                      Certificate generation
kubeconfig                 Generates all kubeconfig files necessary
control-plane              Generates all static Pod manifest files necessary to establish the
  /apiserver                 Generates the kube-apiserver static Pod manifest
  /controller-manager        Generates the kube-controller-manager static Pod manifest
  /scheduler                 Generates the kube-scheduler static Pod manifest
etcd                       Generates static Pod manifest file for local etcd.
  /local                     Generates the static Pod manifest file for a local etcd instance.
upload-config              Uploads the kubeadm and kubelet configuration to a ConfigMap
  /kubeadm                   Uploads the kubeadm ClusterConfiguration to a ConfigMap
  /kubelet                   Uploads the kubelet component config to a ConfigMap
mark-control-plane         Mark a node as a control-plane
bootstrap-token            Generates bootstrap tokens used to join a node to a cluster
addon                      Installs required addons for passing Conformance tests
  /coredns                   Installs the CoreDNS addon to a Kubernetes cluster
  /kube-proxy                Installs the kube-proxy addon to a Kubernetes cluster
~~~

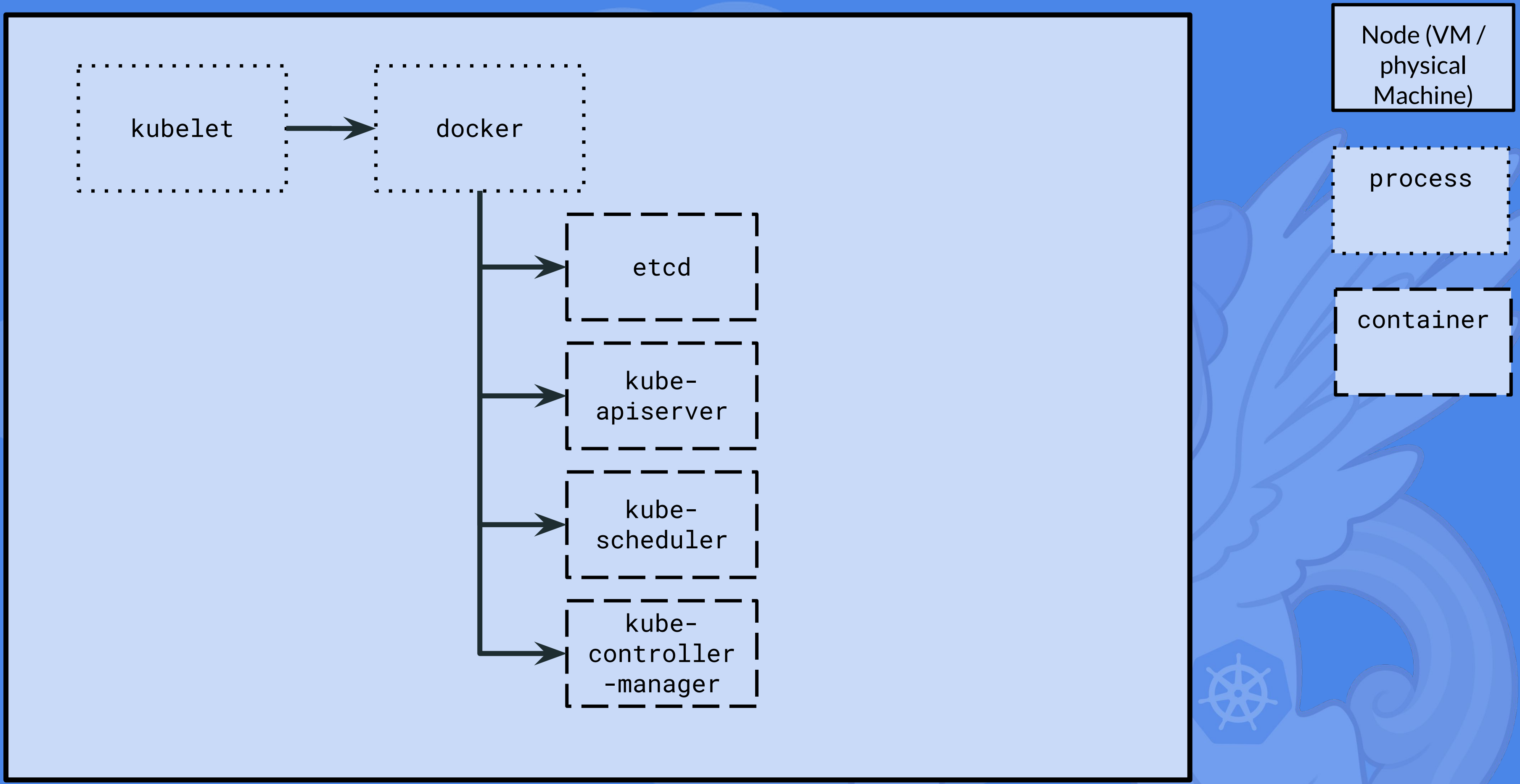
```
preflight
kubelet-start
certs
kubeconfig
control-plane
  /apiserver
  /controller-manager
  /scheduler
etcd
  /local
upload-config
  /kubeadm
  /kubelet
mark-control-plane
bootstrap-token
addon
  /coredns
  /kube-proxy
```

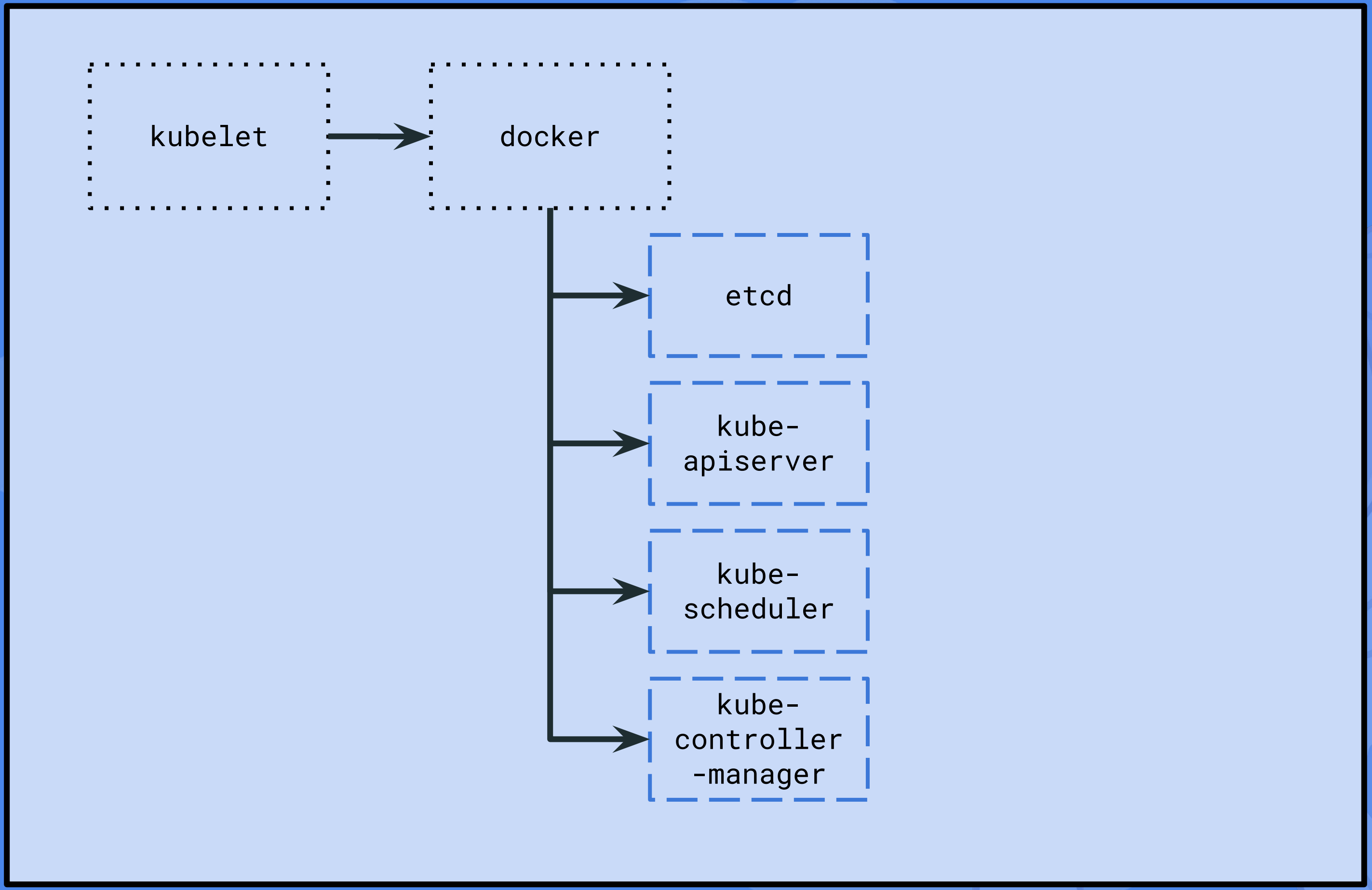
What are we making?

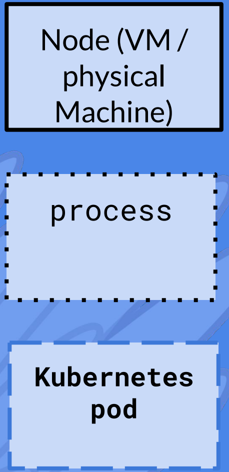
- kubelet
- etcd
- kube-apiserver
- kube-contoller-manager
- kube-scheduler

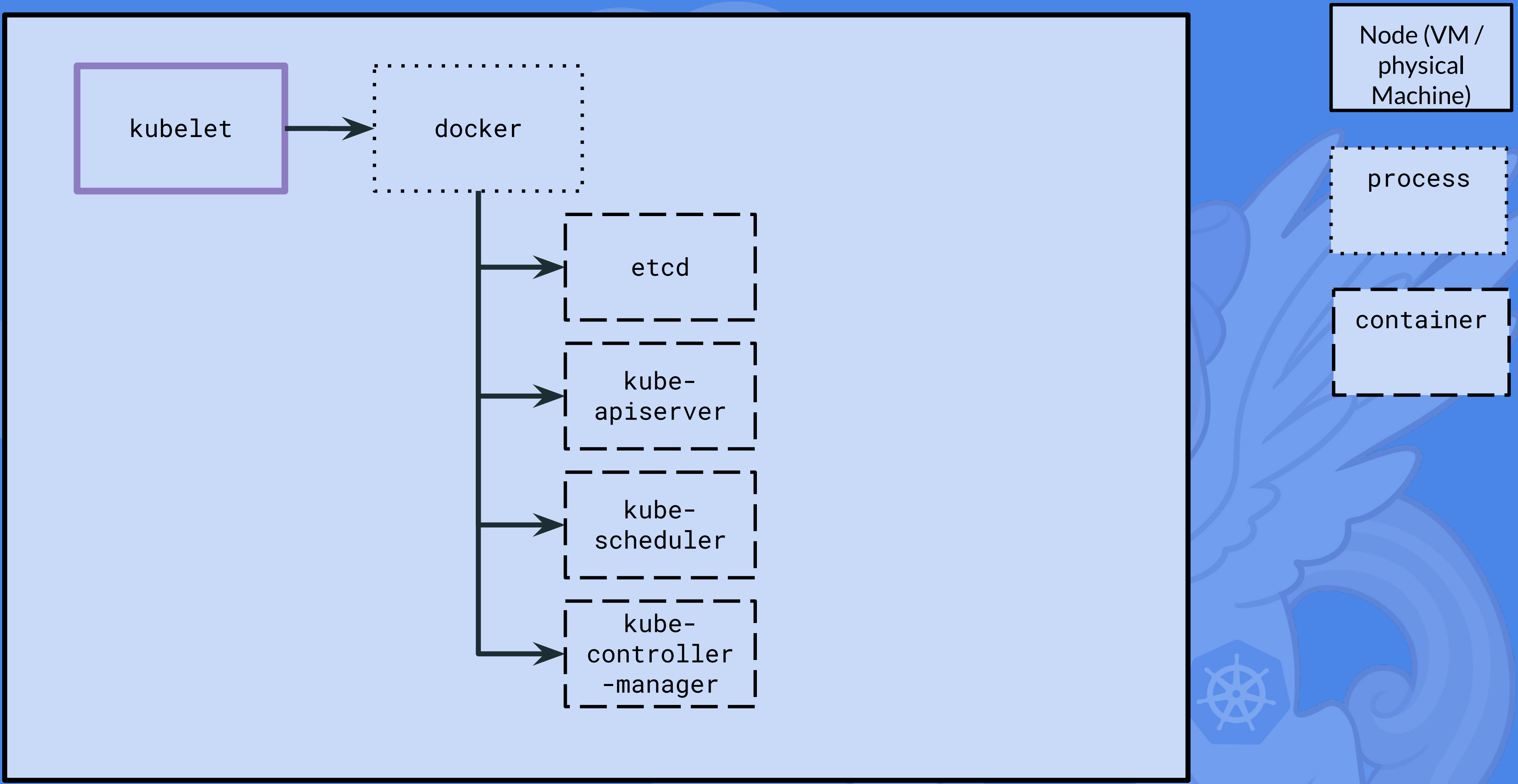


@stillinbeta // #SREcon Americas 2019


kubelet
docker
etcd
kube-apiserver
kube-scheduler
kube-controller-manager
Node (VM / physical Machine)
process
container


kubelet
docker
etcd
kube-apiserver
kube-scheduler
kube-controller-manager


Node (VM / physical Machine)
process
Kubernetes pod

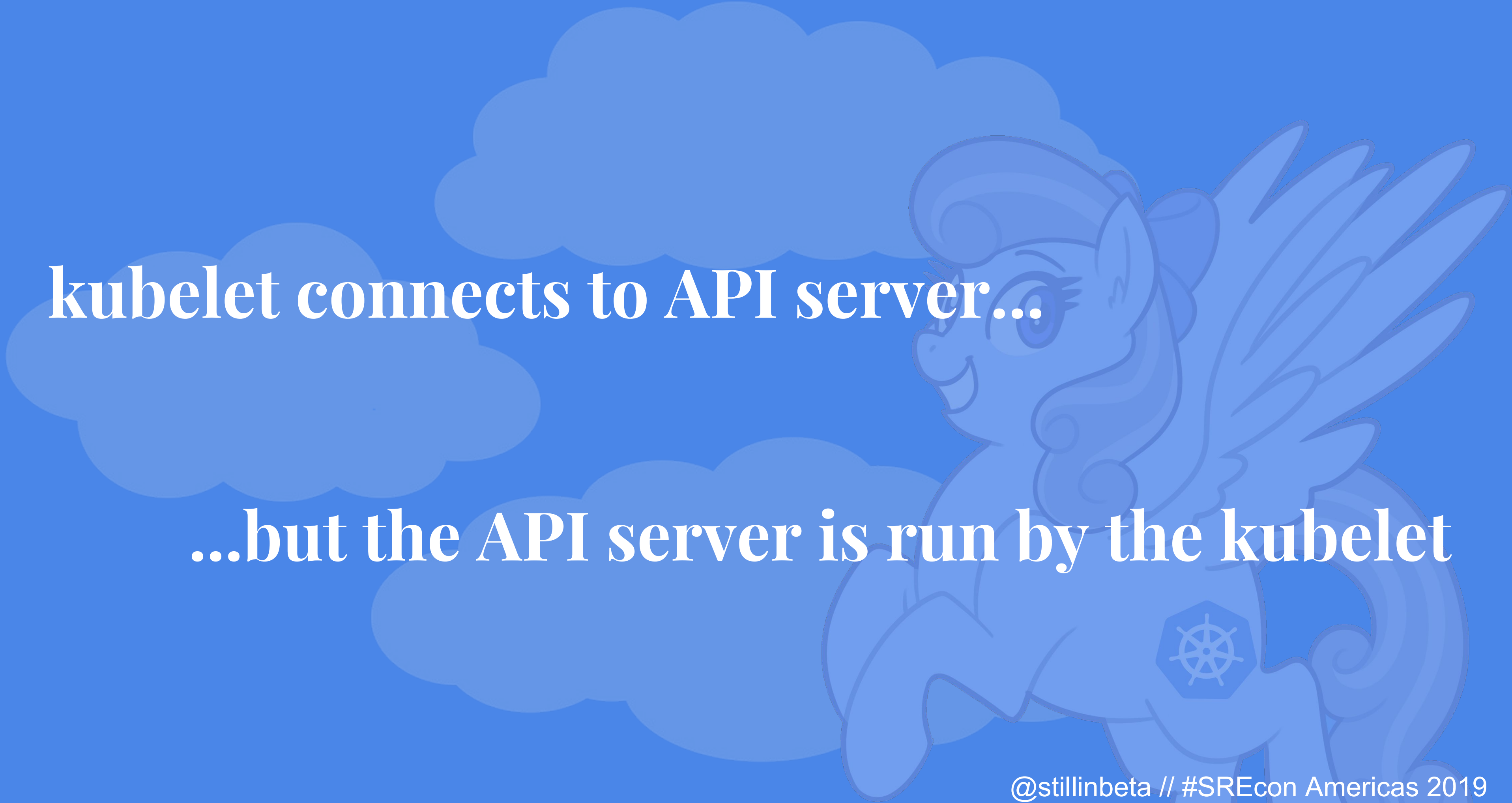


kubelet
docker
etcd
kube-apiserver
kube-scheduler
kube-controller-manager
Node (VM / physical Machine)
process
container

kubelet

What will we need locally?

- kubelet
- kubeadm
- docker

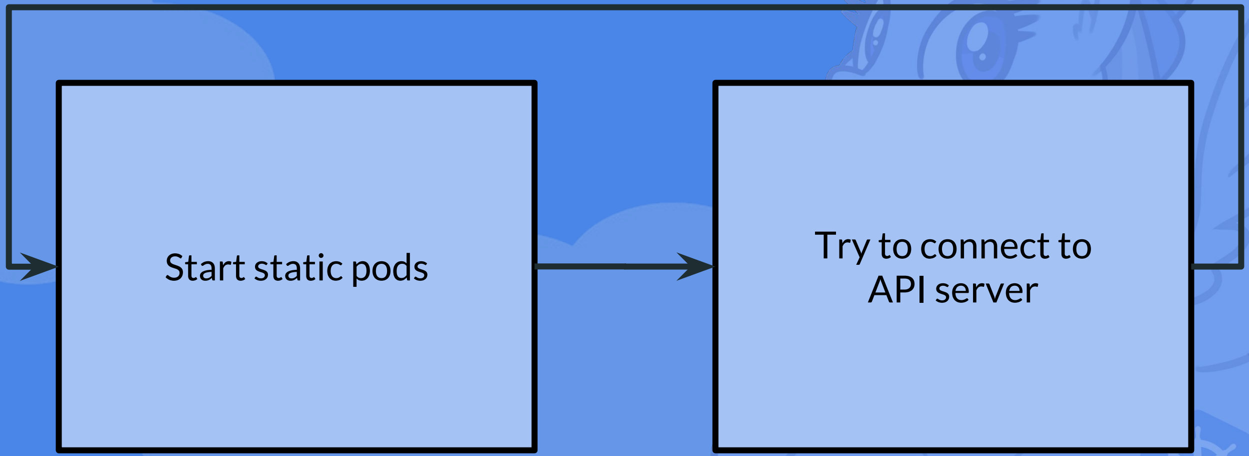


kubelet connects to API server...

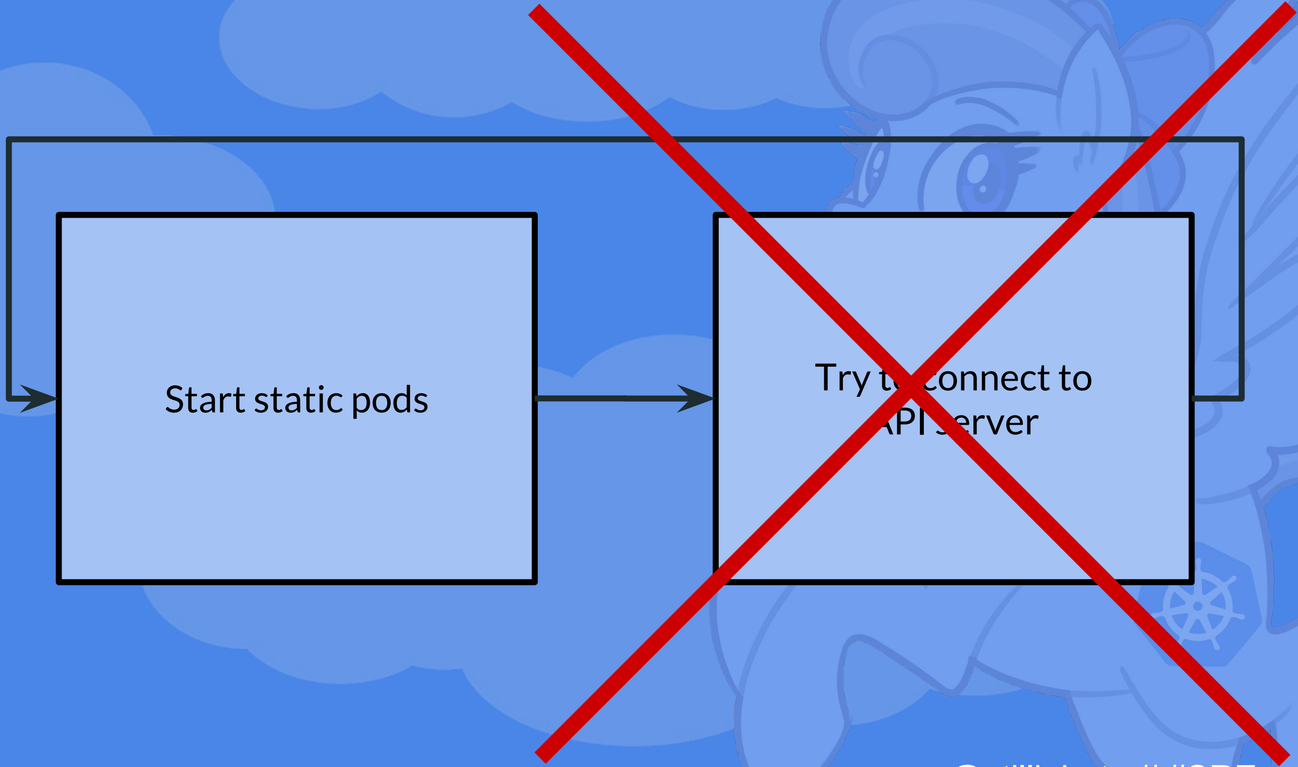
...but the API server is run by the kubelet

MANIFEST

Kubelet boot



Kubelet boot



Start static pods

Try to connect to API server

Kubelet boot

Start static pods

Tutorial 1!

http://bit.ly/srecon-cake1

kubeadm

kubeadm

REMOVE BEFORE FLIGHT

Preflight Checks

- Avoid common problems
- Handy guard-rails
- Ignore at your own peril!

Tutorial 2!

http://bit.ly/srecon-cake2

etcd

From the Website

- Simple: well-defined, user-facing API (gRPC)
- Secure: automatic TLS with optional client cert authentication
- Fast: benchmarked 10,000 writes/sec
- Reliable: properly distributed using Raft

What you care about

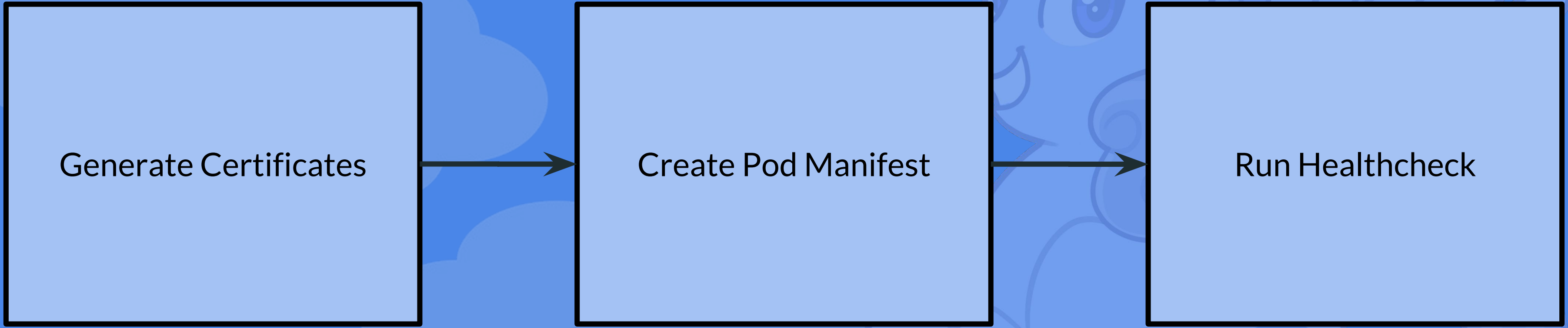
- Key/value store
- *Optionally* distributed
- Only stateful part of kubernetes
- Secured by client certificates

Where do we put it?

- External?
- HA?

- Self-Hosted
- Single Node

Setting up etcd



Generate Certificates

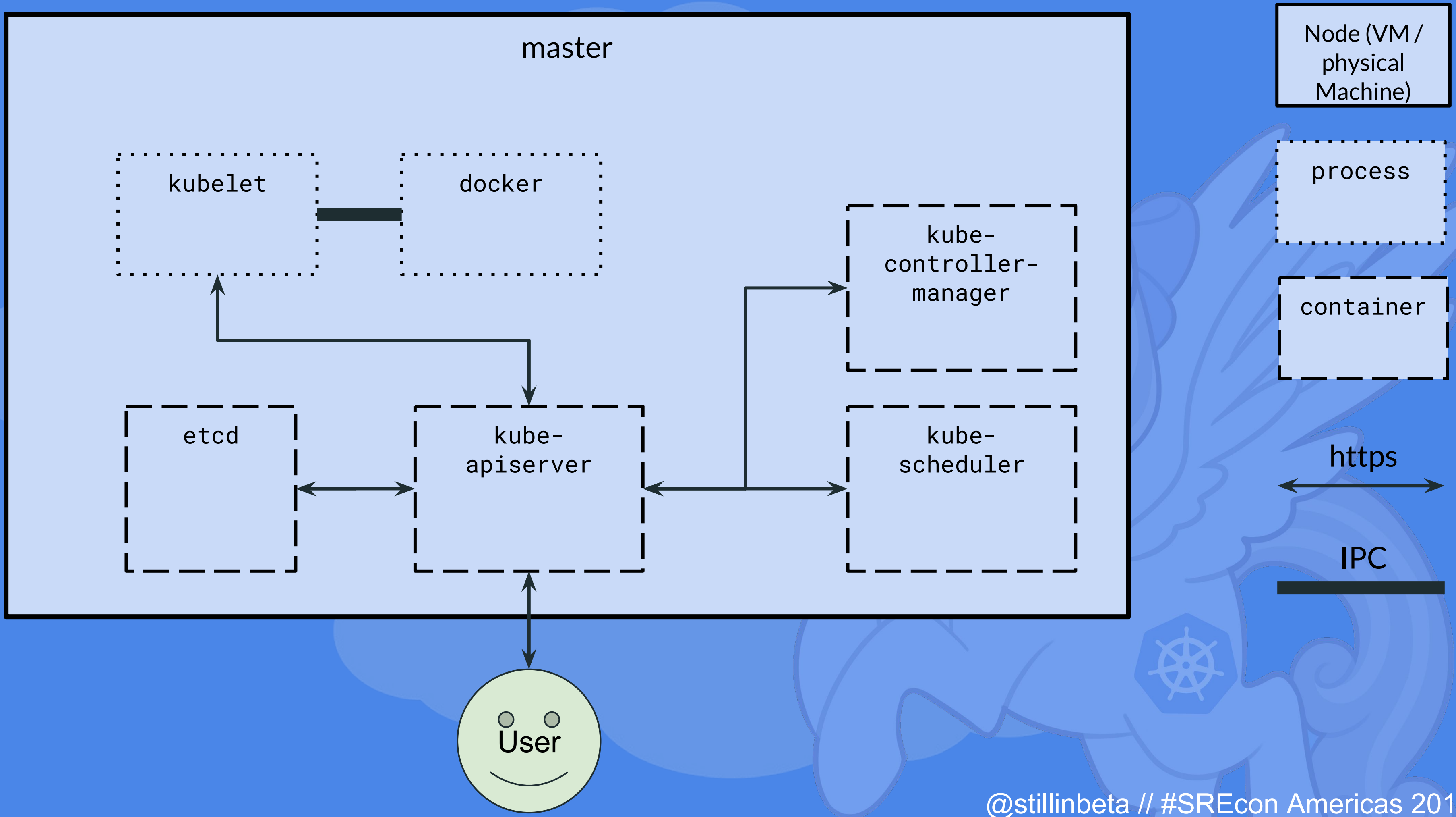
Create Pod Manifest

Run Healthcheck

Tutorial 3!

http://bit.ly/srecon-cake3

apiserver



@stillinbeta // #SREcon Americas 2019

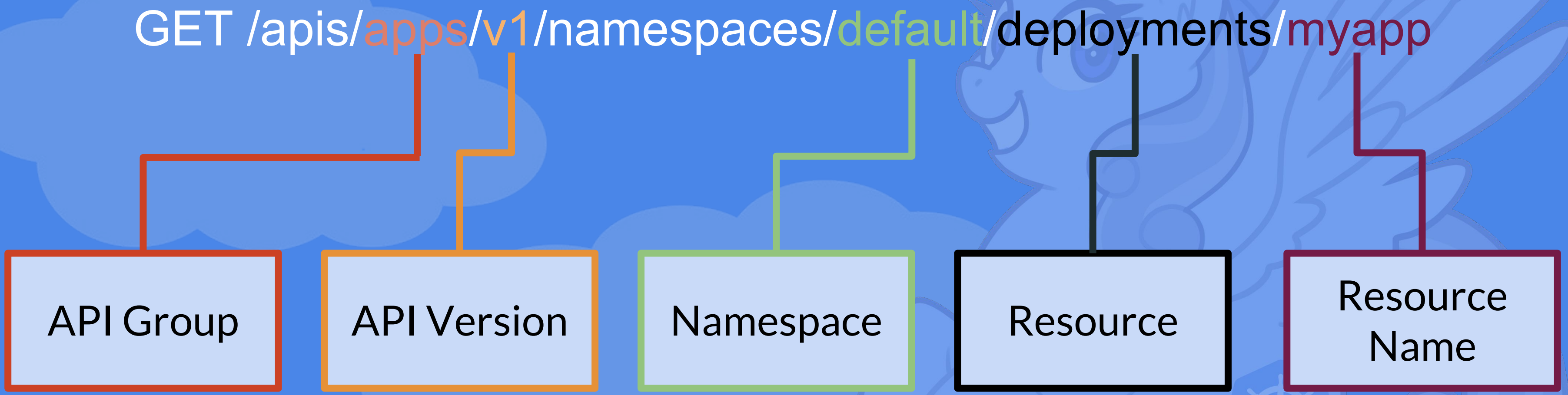
The centre of everything

kube-apiserver

kube-apiserver

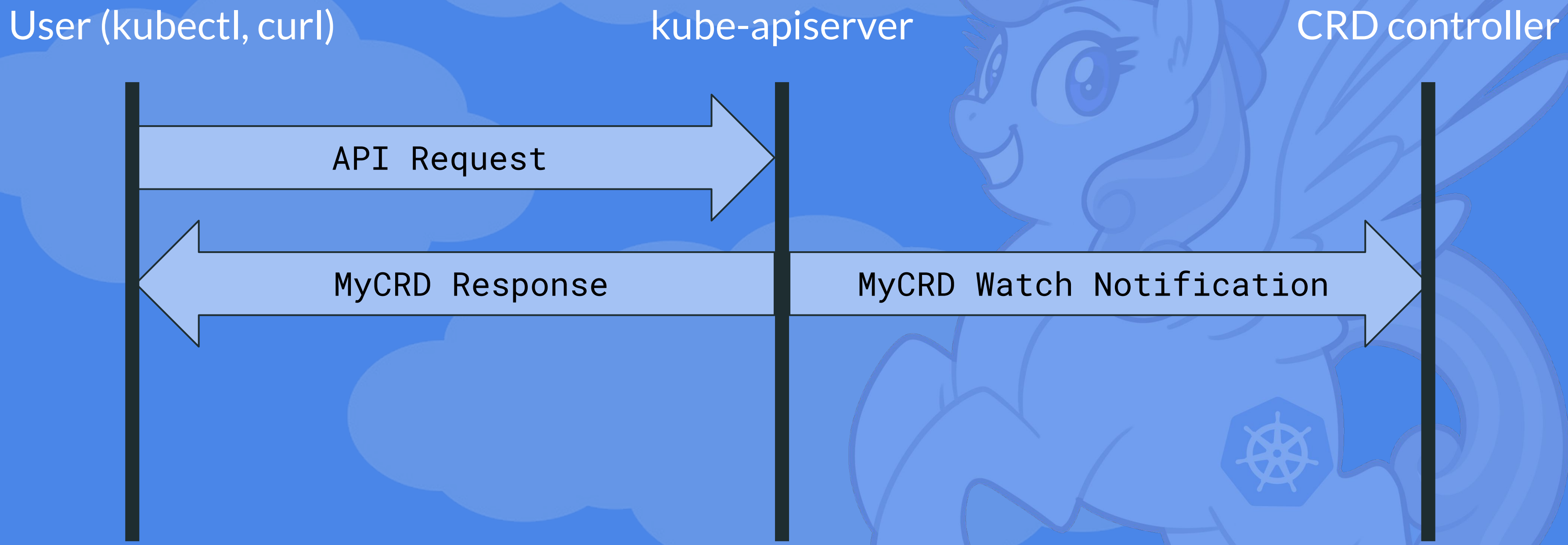
- Exclusive control of etcd
- Handles all access control
- Stores some (but not all!) APIs
- Proxies everything

API Groups?



GET /apis/apps/v1/namespaces/default/deployments/myapp

Custom Resource Definitions



User (kubectl, curl)

kube-apiserver

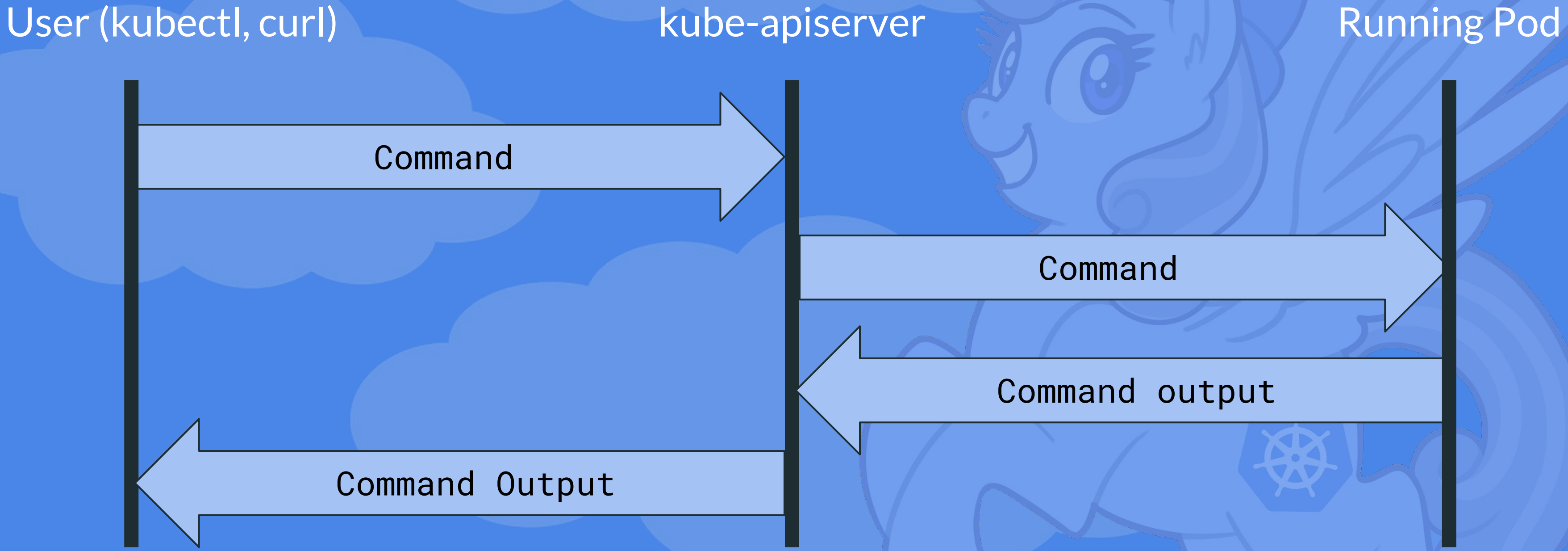
CRD controller

API Request

MyCRD Response

MyCRD Watch Notification

kubectl exec



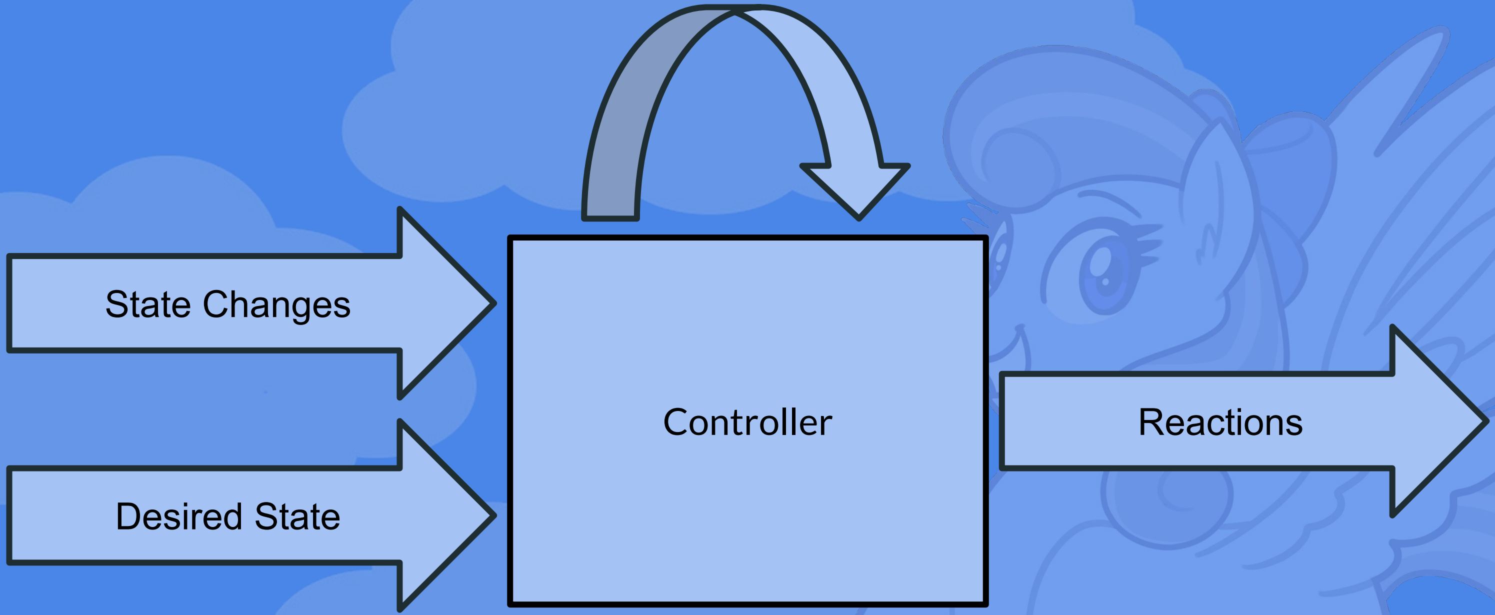
@stillinbeta // #SREcon Americas 2019

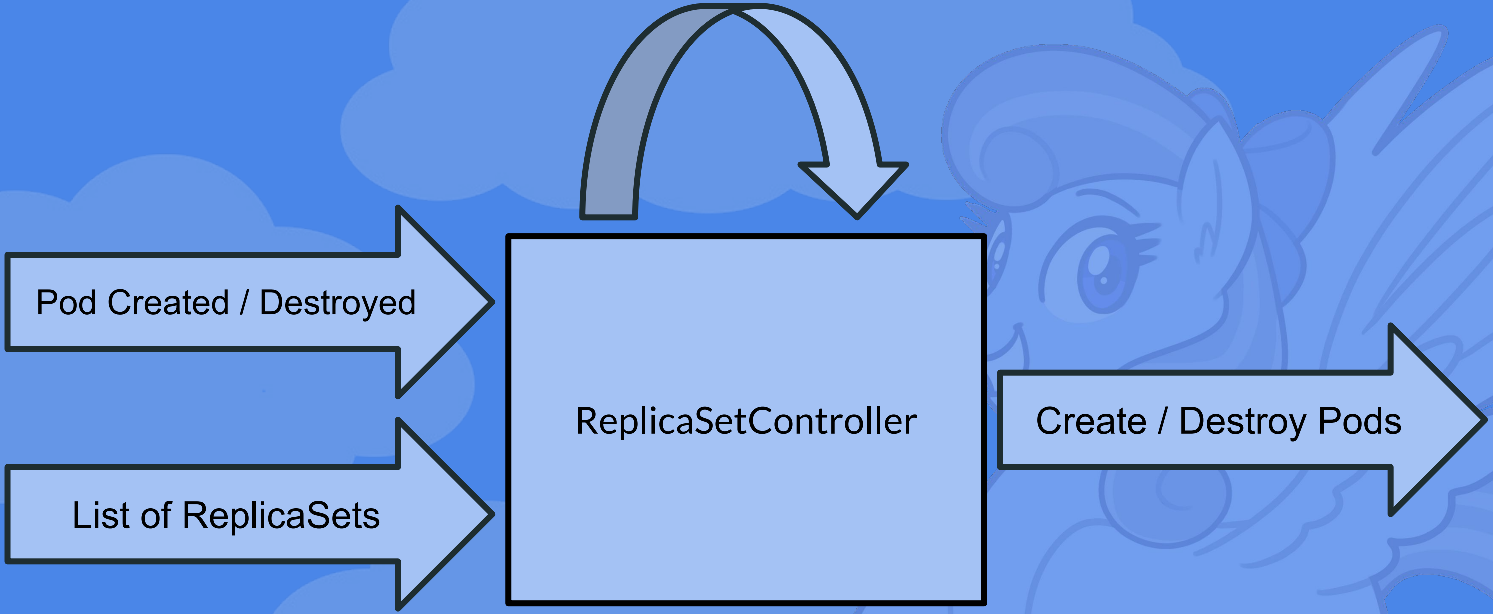
Tutorial 4!

http://bit.ly/srecon-cake4

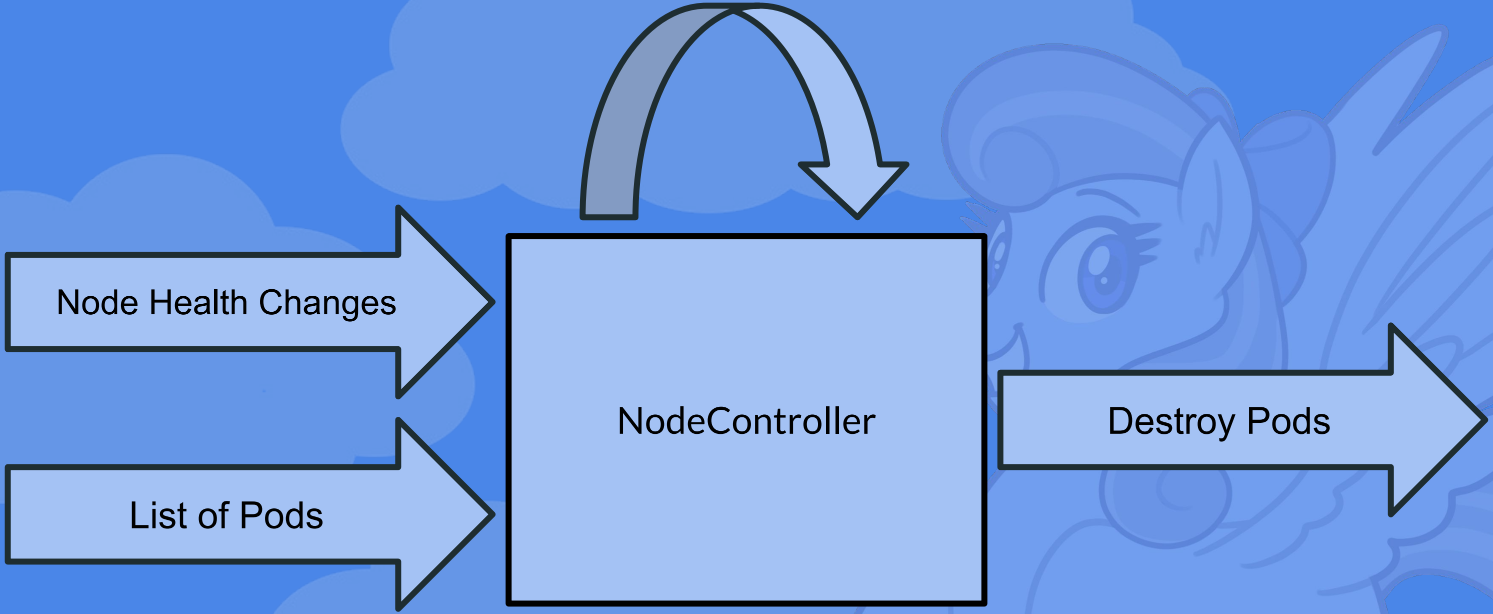
kube-controller-manager

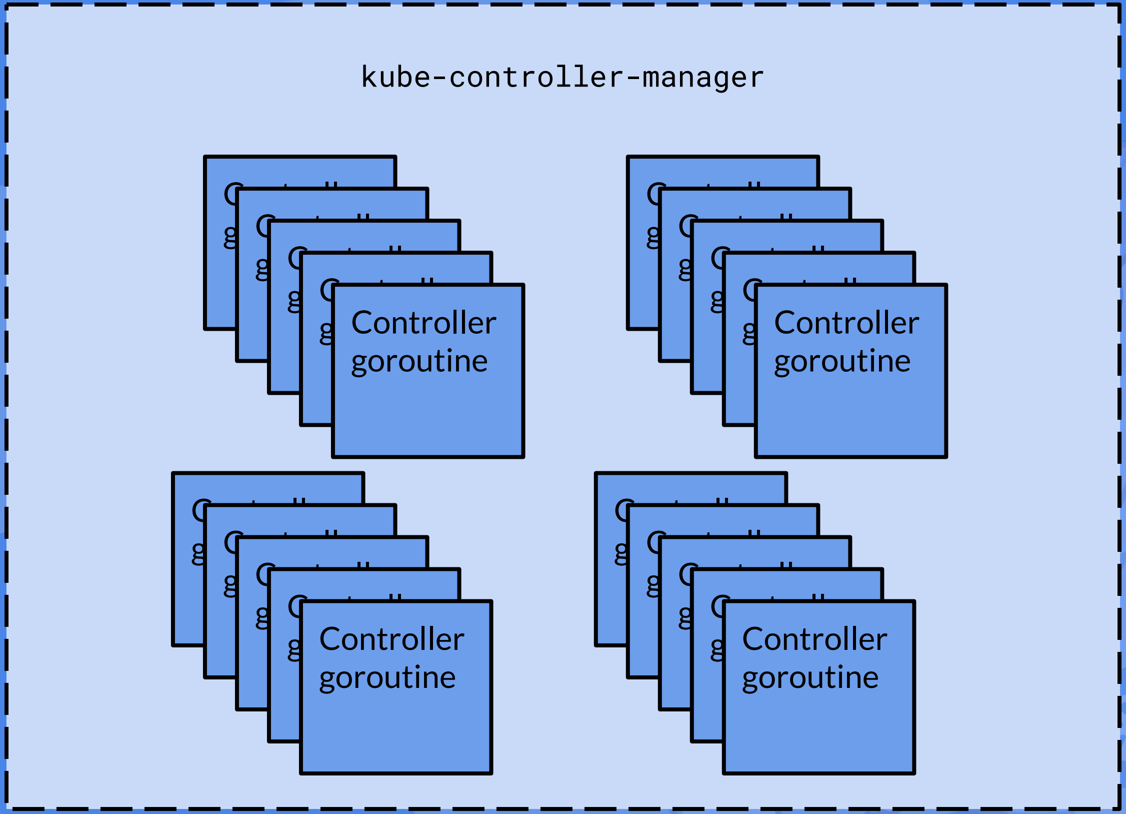
…that could mean anything


State Changes
Desired State
Controller
Reactions



Pod Created / Destroyed
List of ReplicaSets
ReplicaSetController
Create / Destroy Pods

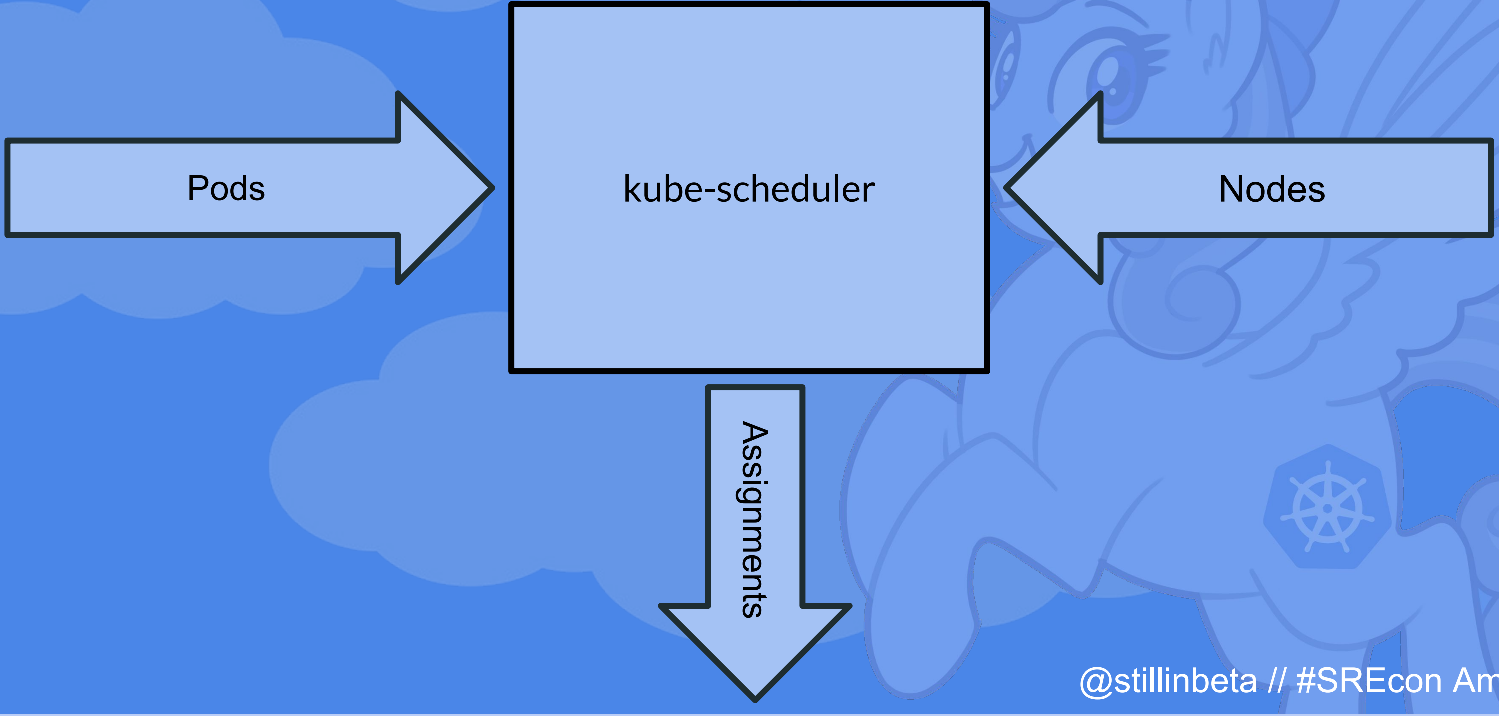

Node Health Changes
List of Pods
NodeController
Destroy Pods


kube-controller-manager
Controller goroutine
Controller goroutine
Controller goroutine
Controller goroutine

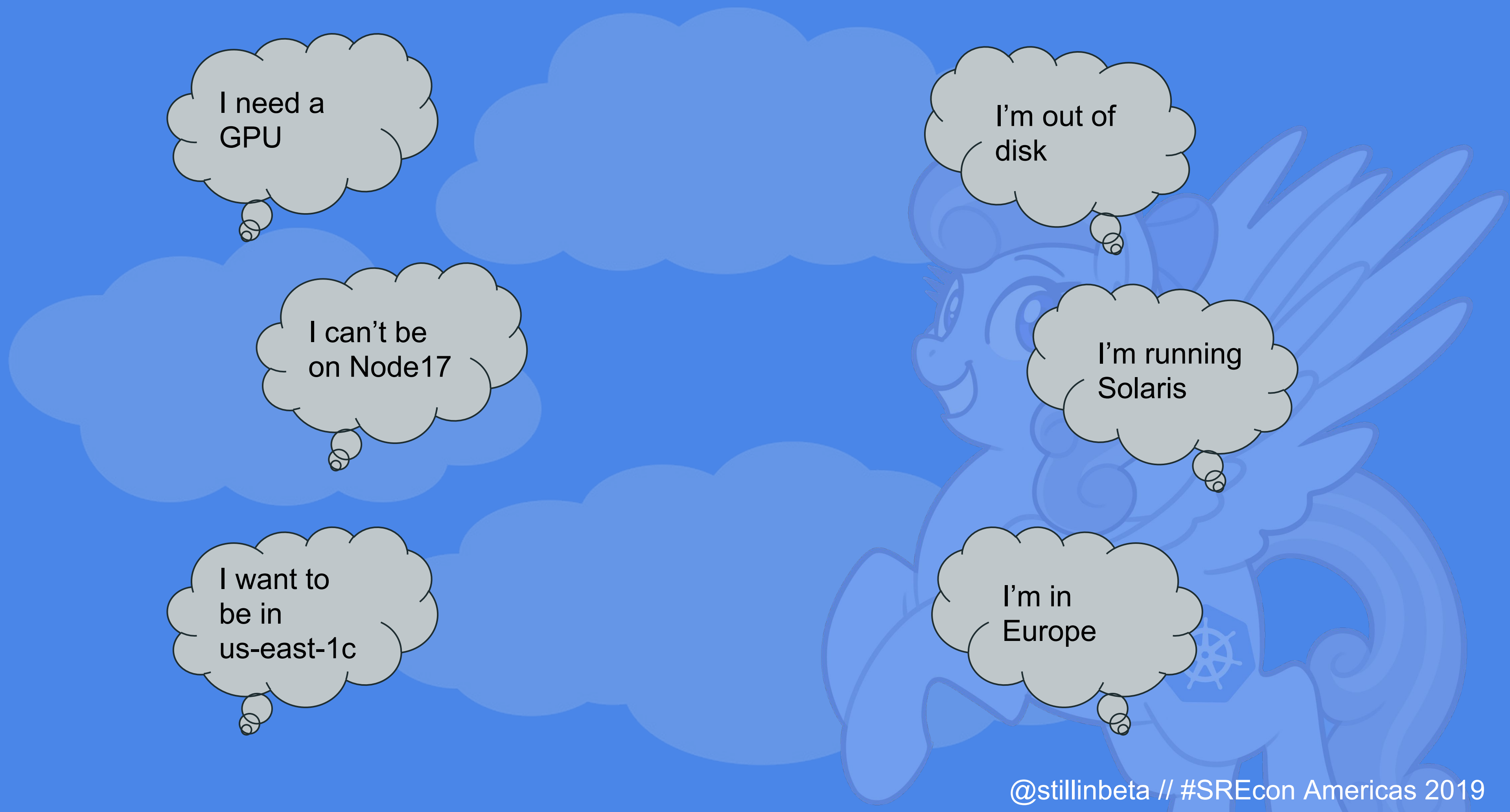
Tutorial 5!

http://bit.ly/srecon-cake5

Scheduler



Pods
kube-scheduler
Nodes
Assignments



I need a GPU

I can't be on Node17

I want to be in us-east-1c

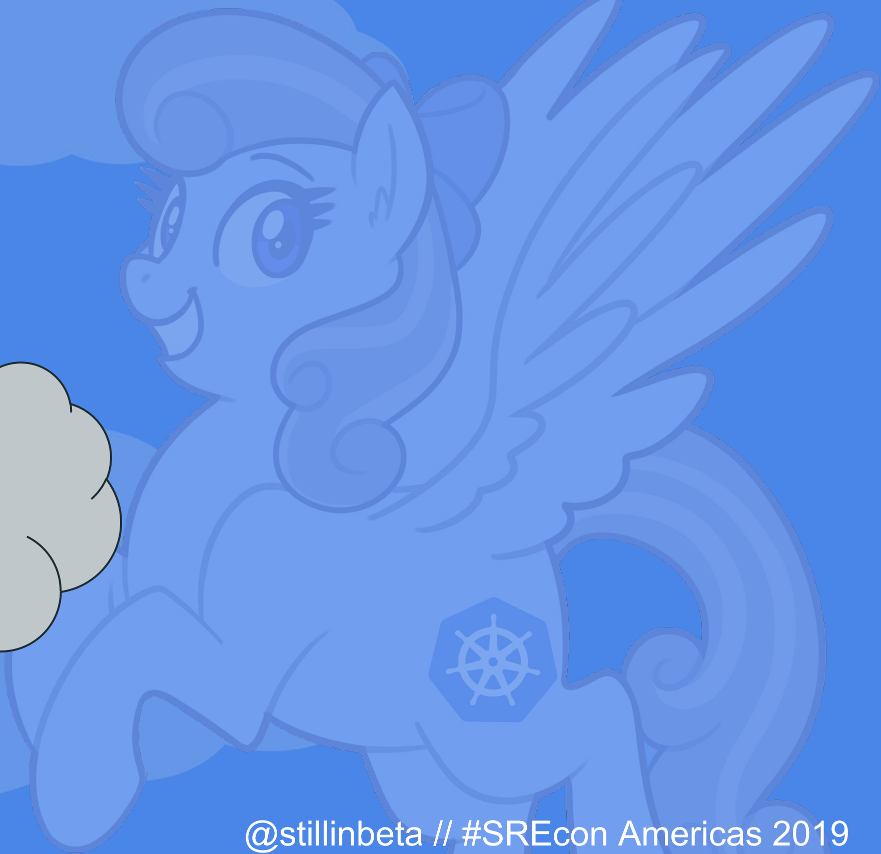
I'm out of disk

I'm running Solaris

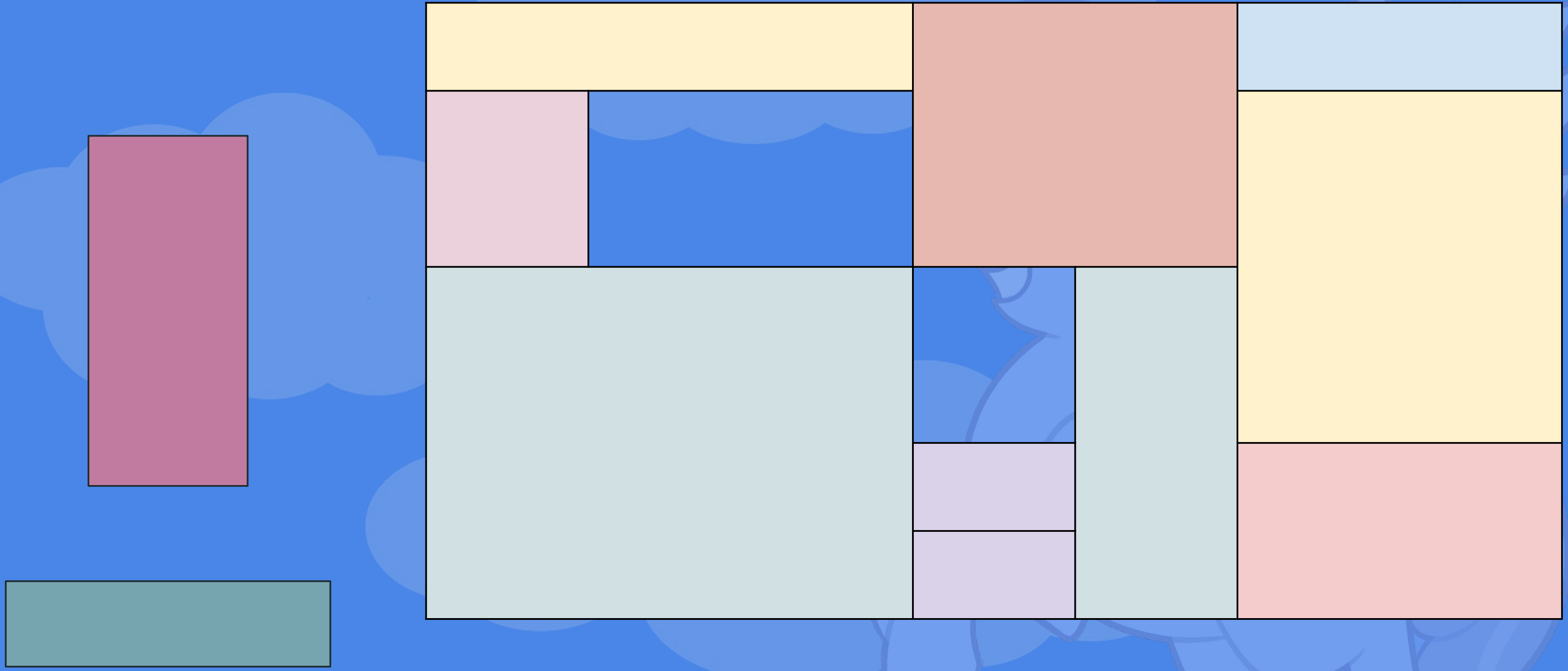
I'm in Europe

I *prefer* Europe

I prefer not to run on the same node as pod17



Bin Packing



Tutorial 6!

http://bit.ly/srecon-cake6

What would be next?

- DNS
- Container Networking
- Proxying
- Multiple Nodes

Your Cluster is *Not:*

- Secure
- Production-Ready
- Resilient
- High-Availability
- Certified Kubernetes

But it is *yours.*

Questions? Concerns?

Come find me!
I’m the one with pink hair!

Twitter: @stillinbeta

slack.k8s.io: @liz

Github: github.com/stillinbeta