Orchestrating Orchestrators: Lessons learned and challenges faced while running Kubernetes at scale

Sidhartha Mani
@utter_babbage
github.com/wlan0

RANCHER

My journey so far...

Senior Software Engineer at Rancher Labs

Open Source Contributions

docker

Docker Contributions

1. Syslog

RANCHER

Open Source Contributions

docker

Docker Contributions

1. Syslog `--log-driver=syslog`

RANCHER

Open Source Contributions

docker

Docker Contributions

1. Syslog `--log-driver=syslog`
2. Logging flags

RANCHER

Open Source Contributions

docker

Docker Contributions

1. Syslog `--log-driver=syslog`
2. Logging flags `--log-opt=max-size=1m`

RANCHER

Open Source Contributions

docker

Docker Contributions

1. Syslog `--log-driver=syslog`
2. Logging flags `--log-opt=max-size=1m`
3. Log rotate

RANCHER

Open Source Contributions

Kubernetes Contributions

1. Cloud provider enhancements

kubernetes

RANCHER

Open Source Contributions

Kubernetes Contributions

1. Cloud provider enhancements
2. Affects APIServer, Kubelet and Controller-Manager

kubernetes

RANCHER

Open Source Contributions

Kubernetes Contributions

1. Cloud provider enhancements
2. Affects APIServer, Kubelet and Controller-Manager
3. New binary in Kubernetes (Cloud-Controller-Manager)

kubernetes

RANCHER

Open Source Contributions

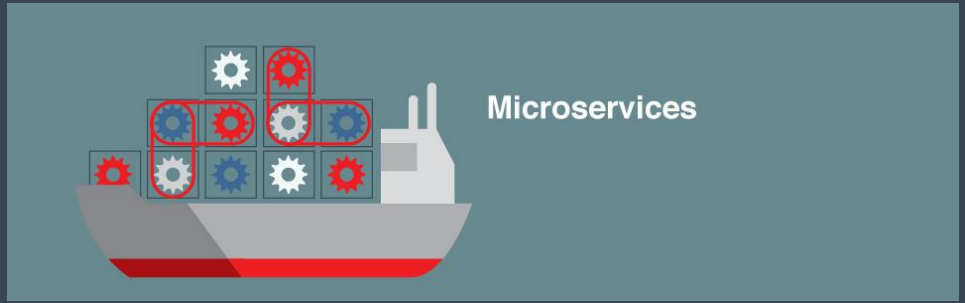
kubernetes

Kubernetes Contributions

1. Cloud provider enhancements
2. Affects APIServer, Kubelet and Controller-Manager
3. New binary in Kubernetes (Cloud-Controller-Manager)
4. More complexity in setup and operations

Kubernetes

Kubernetes is a set of microservices that work together to act as a framework for running distributed platforms



RANCHER

Things we will cover

1. Setup of Kubernetes
    a. Setup
    b. Upgrades
2. Kubernetes Networking
    a. Choosing a provider
    b. Networking
3. Secrets and Config Management

Kubernetes

Etcd

API Server

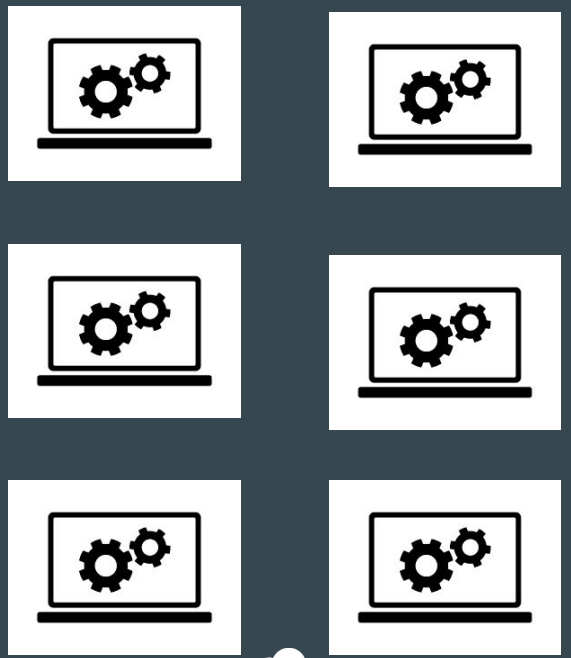
Controller Manager

Scheduler

Proxy

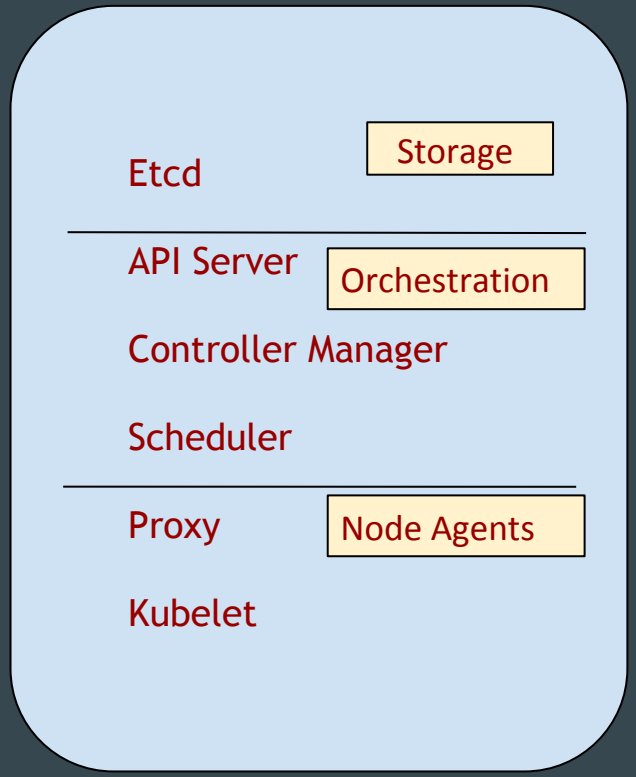
Kubelet

Cluster Nodes

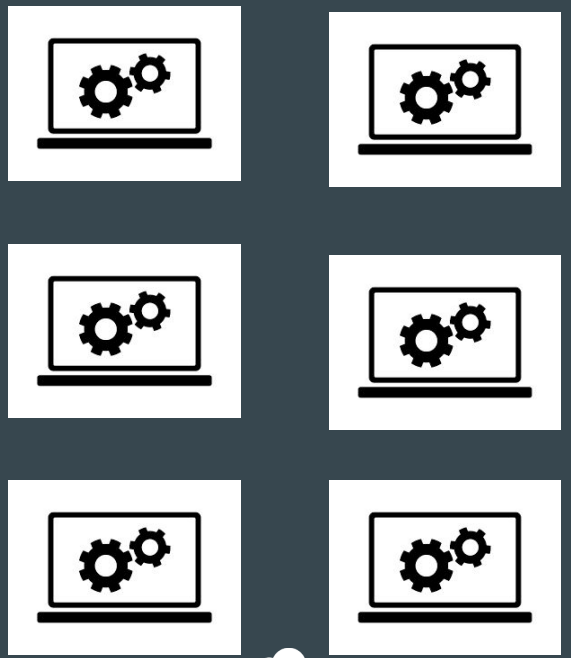


RANCHER

Kubernetes

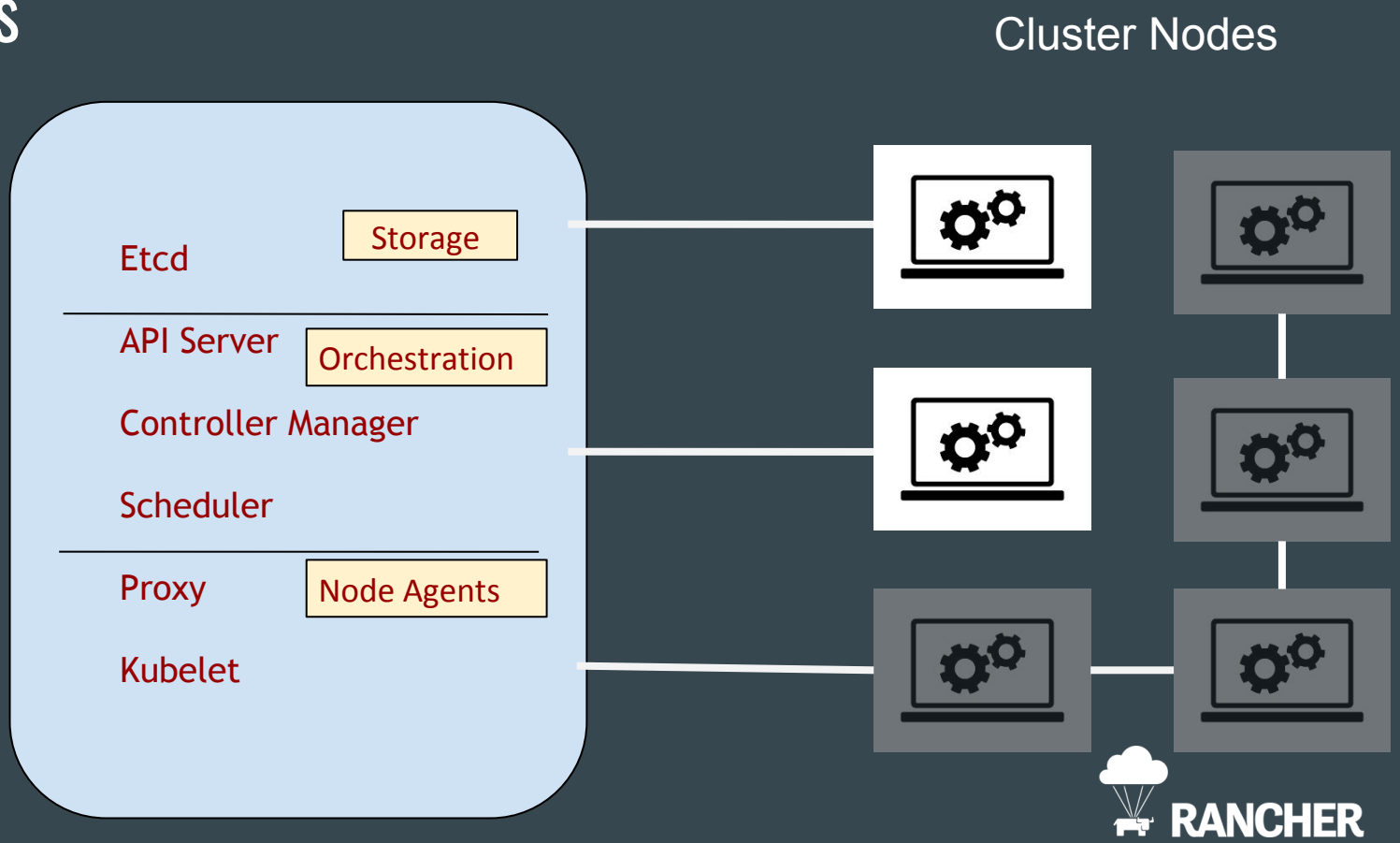


Cluster Nodes

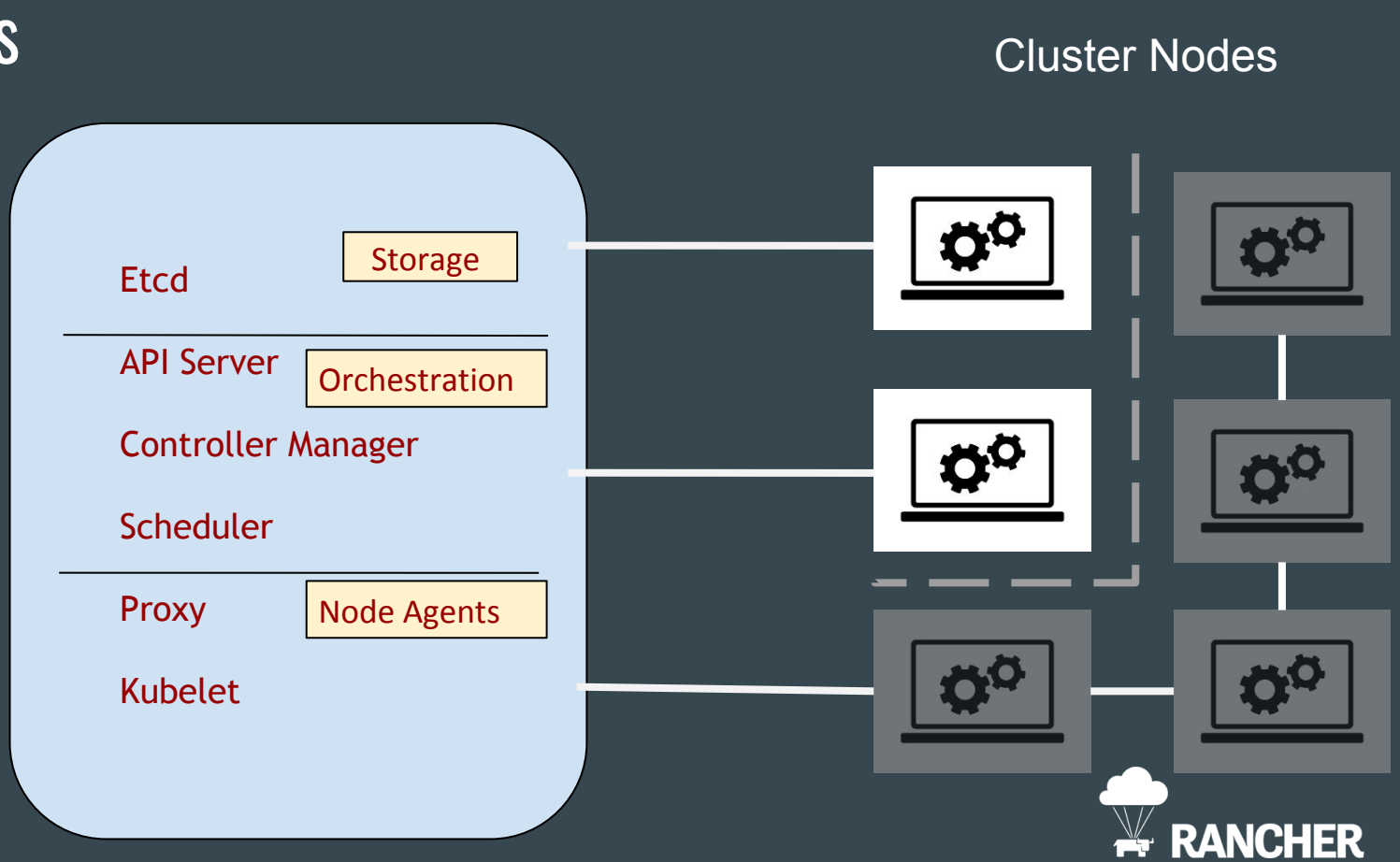


RANCHER

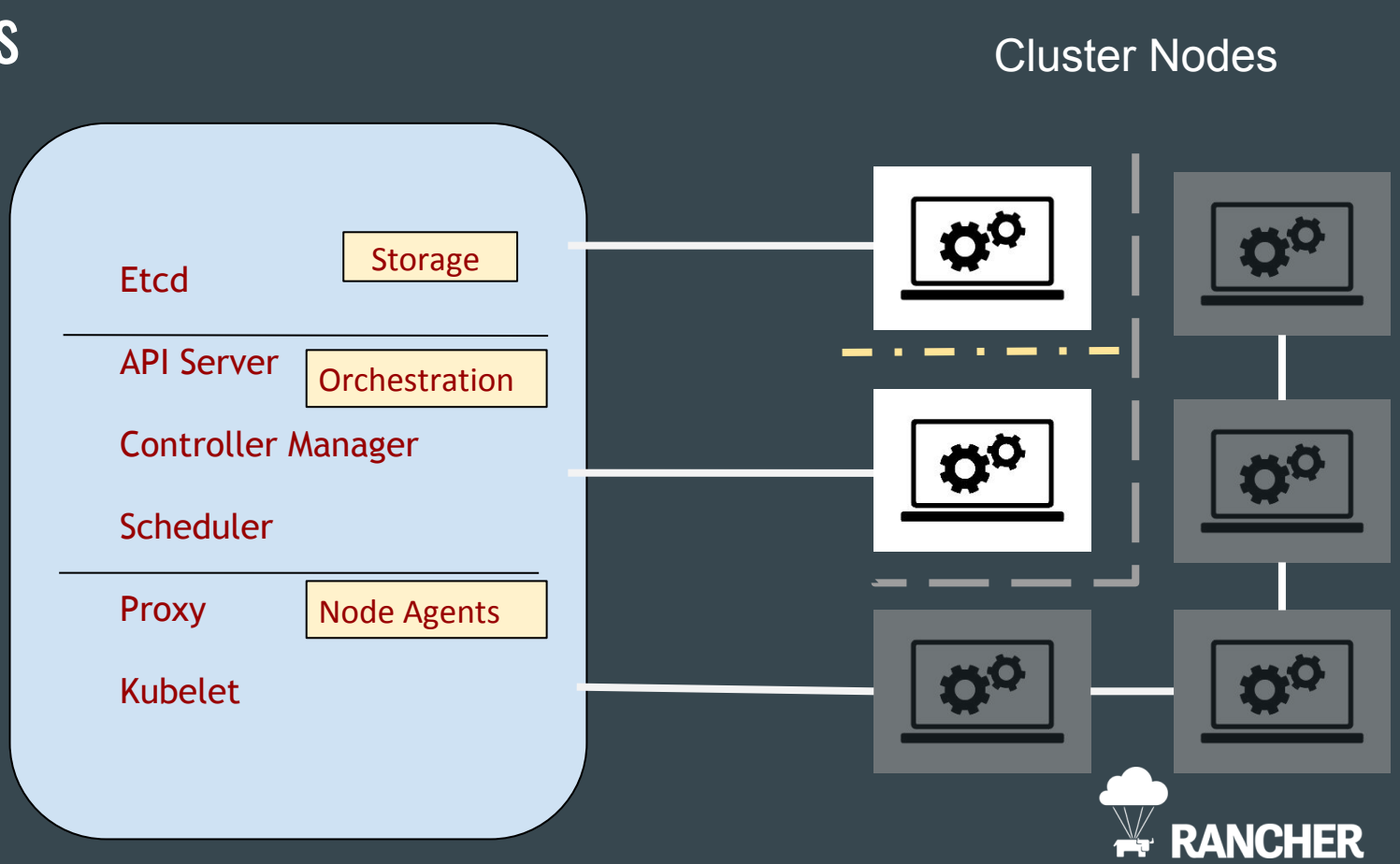
Kubernetes


Cluster Nodes
Etcd
Storage
API Server
Orchestration
Controller Manager
Scheduler
Proxy
Node Agents
Kubelet
RANCHER

Kubernetes


Cluster Nodes
Etcd
Storage
API Server
Orchestration
Controller Manager
Scheduler
Proxy
Node Agents
Kubelet
RANCHER

Kubernetes


Cluster Nodes
Etcd
Storage
API Server
Orchestration
Controller Manager
Scheduler
Proxy
Node Agents
Kubelet
RANCHER

Kubernetes: Lesson 1

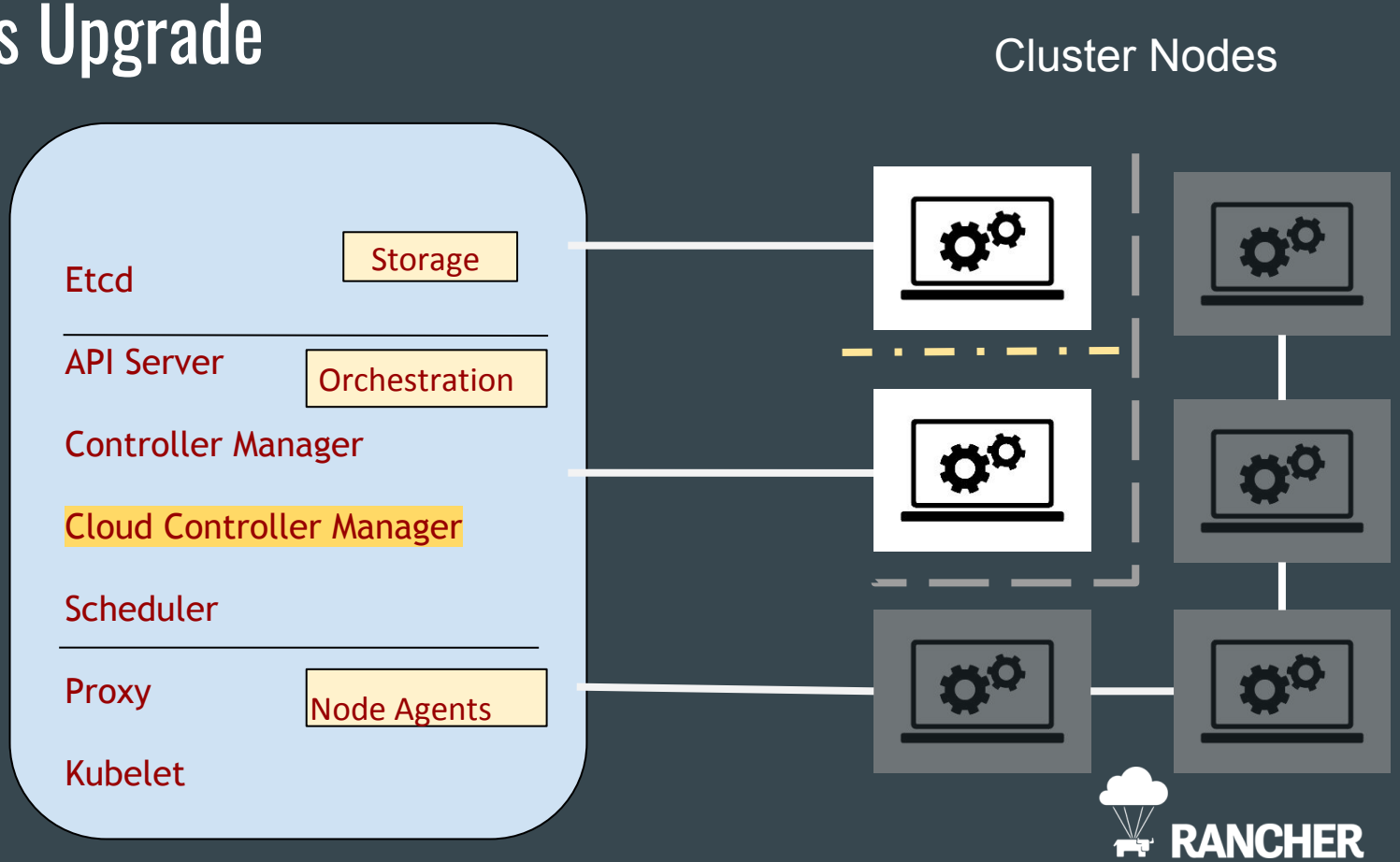
Run Kubernetes core, storage and workloads on separate machines to avoid setups with single points of failure

Things we will cover

1. Setup of Kubernetes
    a. Setup
    b. Upgrades
2. Kubernetes Networking
    a. Networking
    b. Choosing a provider
3. Secrets and Config Management

RANCHER

Kubernetes Upgrade



Kubernetes: Lesson 2

Always upgrade master before upgrading nodes, and etcd before master

Things we will cover

1. Setup of Kubernetes
    a. Setup
    b. Upgrades
2. Kubernetes Networking
    a. Choosing a provider
    b. Networking
3. Secrets and Config Management

Kubernetes Network Providers

1. Flannel
2. Project Calico
3. Weave Net
4. Contiv
5. GCE
6. OpenVSwitch
7. Open Virtual Networking
8. Romana

Kubernetes Network Providers

1. Flannel
2. Project Calico
3. Weave Net
4. Contiv
5. GCE
6. OpenVSwitch
7. Open Virtual Networking
8. Romana

Kubernetes Network Providers

	Project Calico	Weave	Flannel
Application Isolation	Profile schema	CIDR schema	CIDR schema
Networking Model	Pure L3	VxLan or UDP	VxLan, UDP, host-gw...
Name Service	No	Yes	No
Distributed Storage	Yes	No	Yes
Encryption	No	Yes	Yes
Protocol	TCP, UDP, ICMP, ICMP6	ALL	ALL
Partially connected	No	Yes	No
Performance	Near Native	Near native in VxLan	Near native in VxLan

Kubernetes Network Providers

	Project Calico	Weave	Flannel
Application Isolation	Profile schema	CIDR schema	CIDR schema
Networking Model	Pure L3	VxLan or UDP	VxLan, UDP, host-gw...
Name Service	No	Yes	No
Distributed Storage	Yes	No	Yes
Encryption	No	Yes	Yes
Protocol	TCP, UDP, ICMP, ICMP6	ALL	ALL
Partially connected	No	Yes	No
Performance	Near Native	Near native in VxLan	Near native in VxLan

RANCHER

Kubernetes Network Providers

	Project Calico	Weave	Flannel
Application Isolation	Profile schema	CIDR schema	CIDR schema
Networking Model	Pure L3	VxLan or UDP	VxLan, UDP, host-gw...
Name Service	No	Yes	No
Distributed Storage	Yes	No	Yes
Encryption	No	Yes	Yes
Protocol	TCP, UDP, ICMP, ICMP6	ALL	ALL
Partially connected	No	Yes	No
Performance	Near Native	Near native in VxLan	Near native in VxLan

RANCHER

Kubernetes Network Providers

	Project Calico	Weave	Flannel
Application Isolation	Profile schema	CIDR schema	CIDR schema
Networking Model	Pure L3	VxLan or UDP	VxLan, UDP, host-gw...
Name Service	No	Yes	No
Distributed Storage	Yes	No	Yes
Encryption	No	Yes	Yes
Protocol	TCP, UDP, ICMP, ICMP6	ALL	ALL
Partially connected	No	Yes	No
Performance	Near Native	Near native in VxLan	Near native in VxLan

Kubernetes Network Providers

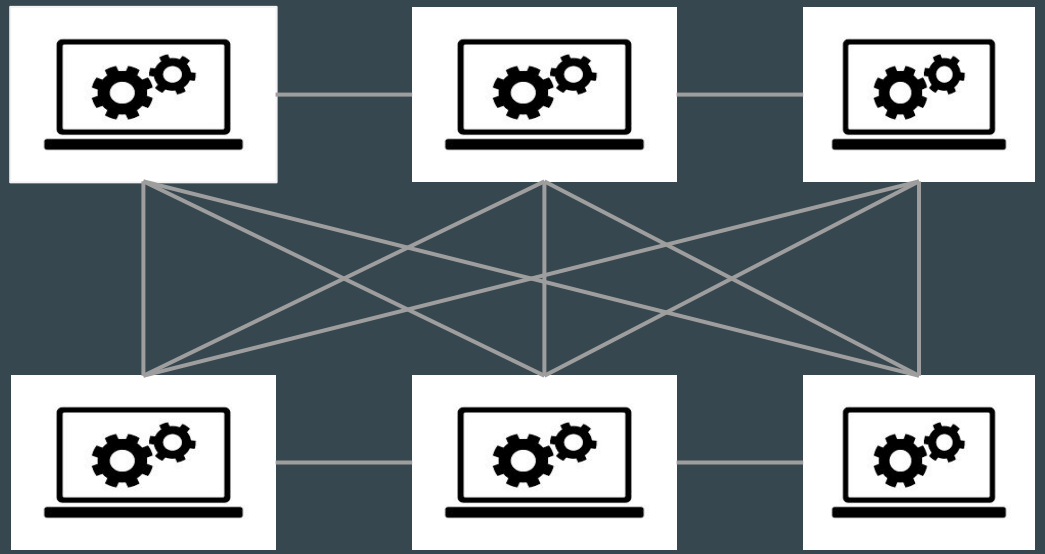
	Project Calico	Weave	Flannel
Application Isolation	Profile schema	CIDR schema	CIDR schema
Networking Model	Pure L3	VxLan or UDP	VxLan, UDP, host-gw...
Name Service	No	Yes	No
Distributed Storage	Yes	No	Yes
Encryption	No	Yes	Yes
Protocol	TCP, UDP, ICMP, ICMP6	ALL	ALL
Partially connected	No	Yes	No
Performance	Near Native	Near native in VxLan	Near native in VxLan

Things we will cover

1. Setup of Kubernetes
    a. Setup
    b. Upgrades
2. Kubernetes Networking
    a. Choosing a provider
    b. Networking
3. Secrets and Config Management

Kubernetes Networking Model

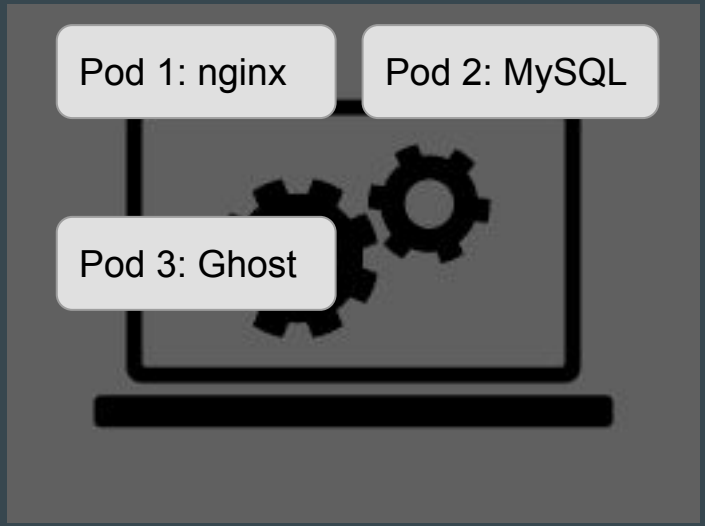
1. Designed for portability from VMs
2. Every POD gets it own IP

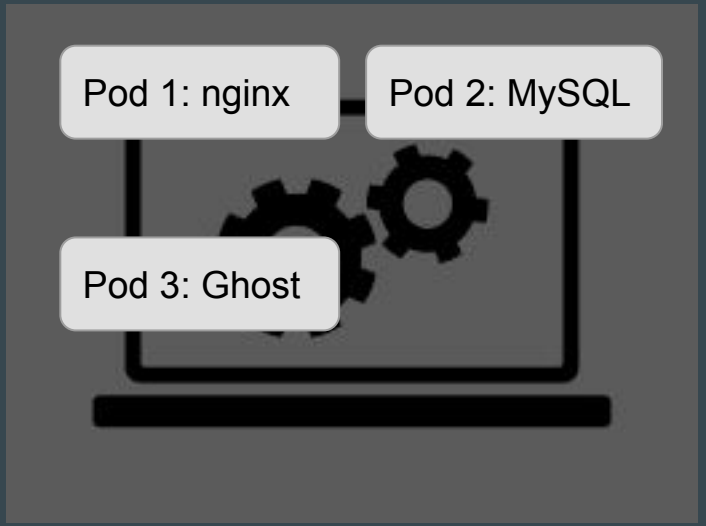


RANCHER

Kubernetes Networking: Services

Logical set of pods





RANCHER

Kubernetes Networking Features

1. Selects Pods using labels
2. Best to create service before creating replication controllers
3. Create 1 replica first to check that it works and then scale it up

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
  labels:
    k8s-app: nginx-service
spec:
  ports:
  - port: 80
    targetPort: 80
    protocol: TCP
    name: http
  selector:
    k8s-app: nginx-service
```

RANCHER

Kubernetes Networking

1. Microservices architecture
2. Built-in service discovery

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
  labels:
    k8s-app: nginx-service
spec:
  ports:
  - port: 80
    targetPort: 80
    protocol: TCP
    name: http
  selector:
    k8s-app: nginx-service
```

Kubernetes Networking: Services

RANCHER

Kubernetes Networking

1. Microservices architecture
2. Built-in service discovery
3. Built-in load balancing

Kubernetes Networking: Services

RANCHER

Kubernetes: Lesson 3

Move ALL your legacy apps to microservice architecture in order to fully leverage Kubernetes

Kubernetes: Lesson 4

Do not use hostPort or nodePort services, unless absolutely necessary.

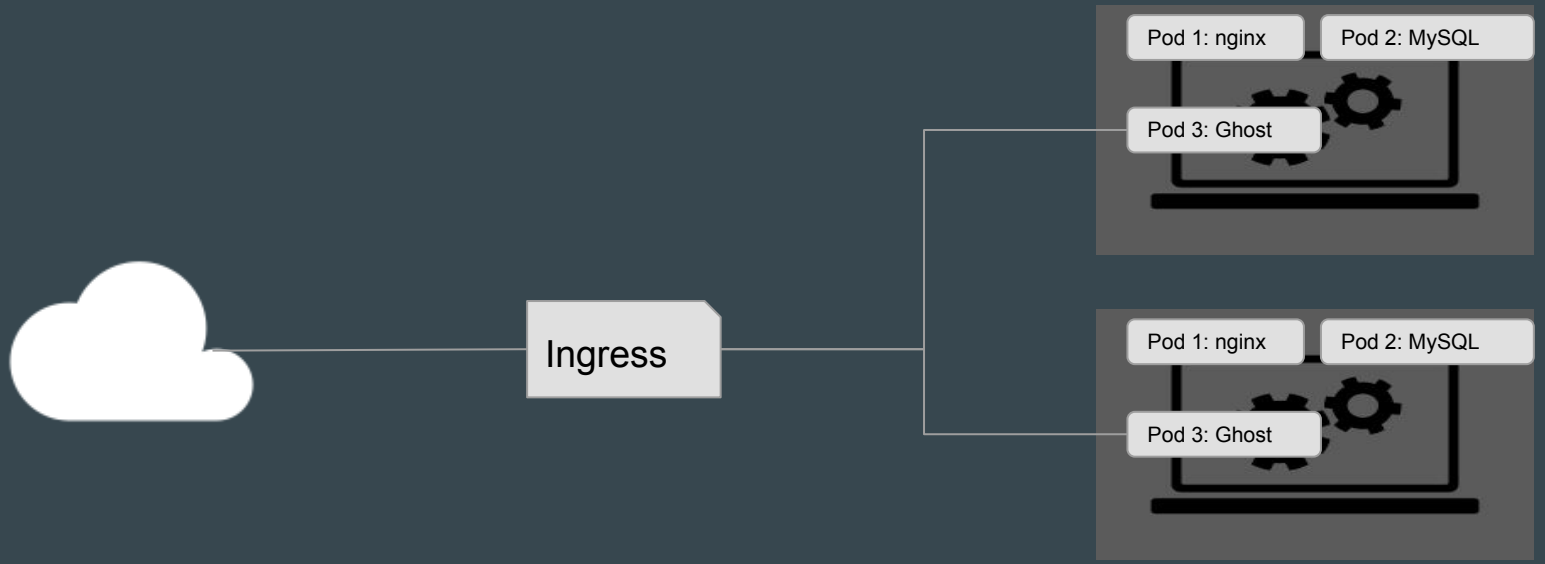
Kubernetes Networking

1. Microservices architecture
2. Built-in service discovery
3. Built-in load balancing
4. Ingress Support
    a. TLS terminations/passthrough
    b. SNI
    c. Wildcard based routing

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: test
  annotations:
    http.port: "99"
spec:
  rules:
  - host: foo.bar.com
    http:
      paths:
      - path: /foo
        backend:
          serviceName: nginx-service
          servicePort: 80
```

RANCHER

Kubernetes Networking: Ingress

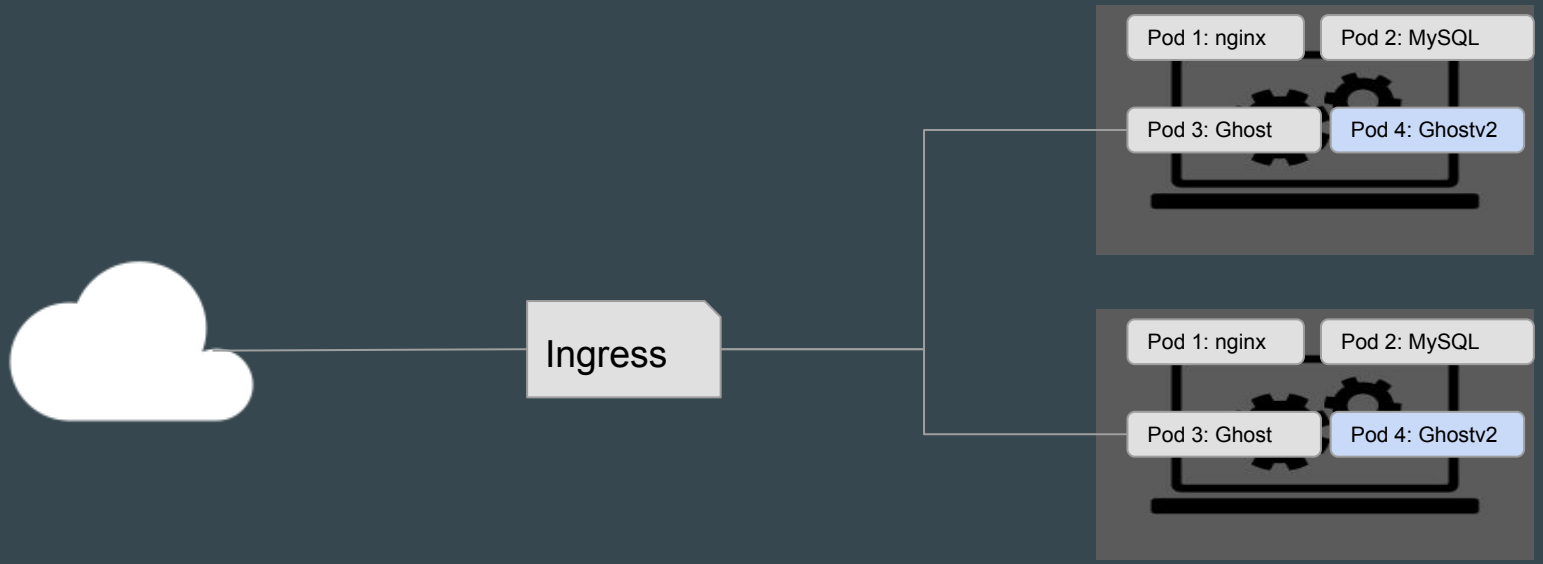


RANCHER

Kubernetes Networking

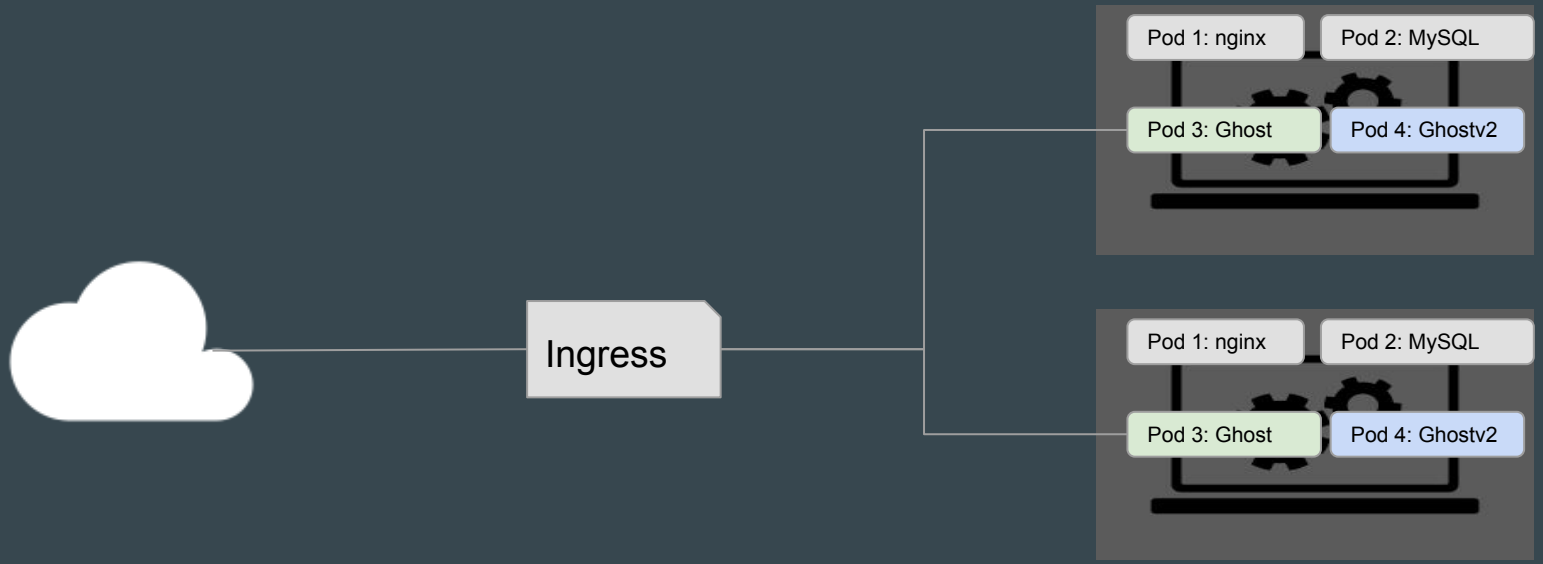
1. Microservices architecture
2. Built-in service discovery
3. Built-in load balancing
4. Ingress Support
    a. TLS terminations/passthrough
    b. SNI
    c. Wildcard based routing
5. Zero Downtime Upgrades

Kubernetes Networking



RANCHER

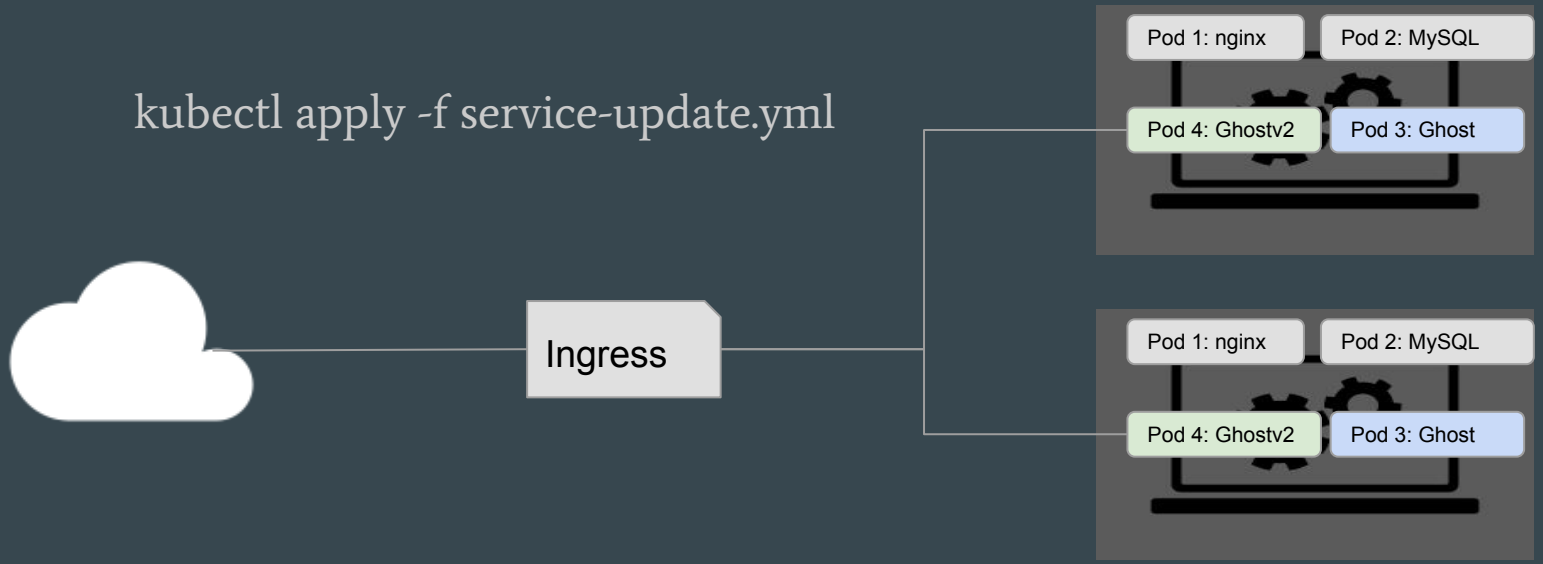
Kubernetes Networking



RANCHER

Kubernetes Networking

kubectl apply -f service-update.yml



RANCHER

Kubernetes: Lesson 5

Define and use labels that identify semantic attributes of your application or resource

Things we will cover

1. Setup of Kubernetes
    a. Setup
    b. Upgrades
2. Kubernetes Networking
    a. Choosing a provider
    b. Networking
3. Secrets and Config Management

Kubernetes Secrets

1. In-Built Secret Management
2. Encrypted over the network
3. Mounted directly into the containers

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
type: Opaque
data:
  username: YWRtaW4=
  password: MWYyZDFlMmU2N2Rm
```

Kubernetes Secrets: Creation

1. From File

   kubectl create secret generic demo-secret --from-file=demo-secret.txt

2. Manually

   kubectl create -f demo-secret.yaml

Kubernetes Secrets: Usage

1. Mount Secret entirely into directory
2. Project only certain keys
3. Secrets as environment variables
4. Secret File Modes
5. Secret updates

Kubernetes Config Maps

1. In built Config Management
2. Similar to secrets but designed for non-sensitive information

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: special-config
  namespace: default
data:
  special.how: very
  special.type: charm
```

Kubernetes Config Maps: Creation

1. From File/Directory

   kubectl create configmap demo-config --from-file=demo-config.txt

2. Manually

   kubectl create -f demo-config.yaml

3. From Literal

   kubectl create config-map special-config --from-literal=special.how=very --from-literal=special.type=charm

Kubernetes Config Maps: Usage

1. Environment variables
2. Volume plugins
3. Command line arguments

Thank you

I will be at the booth on the Expo floor. (Look for Rancher)

 @utter_babbage

Email: **sid@rancher.com**

Community Users: **https://forums.rancher.com**

Slack: **https://rancher-users.slack.com**

If you want to play with containers/microservices: **https://try.rancher.com**

Flannel

Configurable virtual overlay network

Requires that every machine should be able to talk to each other

Need to manage IP Address spaces

Need to manage another daemon - flanneld

Requires extra software to run - etcd

Need to setup and run etcd

Project Calico

Kubernetes Networking: DNS

Requirement from Kubernetes:

> The IP address that a container sees itself as is the Same IP address that others see it as

Kubernetes Networking: DNS

Requirement from Kubernetes:

> The IP address that a container sees itself as is the
> Same IP address that others see it as