Fully Automated Kubernetes Deployment and Management

Peng Jiang @ Rancher Labs

RANCHER

Kubernetes is awesome

- Automates your application deployment
- Manages its health, and recovers from failures
- Takes care of rolling upgrades of the app

But who takes care of doing the same for the Kubernetes cluster?

Challenges

1. Cloud provider agnostic infrastructure management
2. Kubernetes cluster deployment automation
3. Health monitoring of Kubernetes cluster
4. Detect and recover from etcd node failures
5. Automate the upgrade of Kubernetes cluster

Tools and sources

kubeadm

ANSIBLE

RANCHER

Terraform

Kubernetes is the most popular orchestration platforms among Rancher users.

And we wanted to make Kubernetes deployments and upgrades easy!

#1 Lay the foundation

- Pick the network driver
- Get internal DNS in place
- Choose storage driver solution

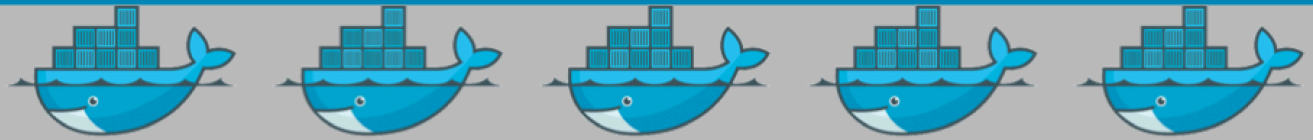
All of the above should work across clouds

Sounds diverse and complex, but..

- In Rancher we think of container as **not** just another layer on top of existing stack

- Containers redefine **ALL** layers
- **Everything** is a container

Applications (Containerized)



Application Orchestration (Containerized)

And more to come...

Infrastructure Services

- Monitoring
- Reliable Storage
- Load Balancing
- Network (SDN)
- DNS
- Database RDBMS, K/V
- And More...

RANCHER

Raw Hardware Physical/Virtual

- Compute CPU x86/ARM
- Storage SSD
- Network Fast Fabric

RANCHER

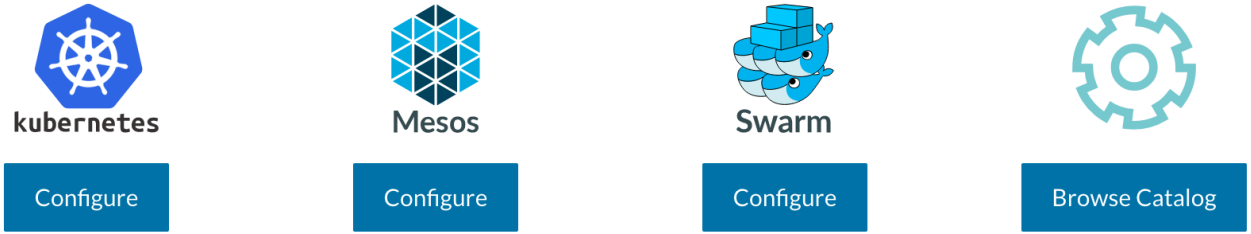
#2 Kubernetes cluster deployment

- Bootstrap an HA etcd cluster
- Deploy an HA Kubernetes Control Plane
- Bring up Kubernetes workers – kubelets

All with one click, with secure access to and between components

Add Infrastructure Stacks

Infrastructure stacks add capabilities like new Orchestration engines, storage and networking providers to Rancher.



RANCHER

Docker Compose file format

```
kubelet:
    labels:
        io.rancher.container.dns: "true"
        io.rancher.container.create_agent: "true"
        io.rancher.container.agent.role: environmentAdmin
        io.rancher.scheduler.global: "true"
        io.rancher.scheduler.affinity:host_label_ne: nopods=true
    command:
        - kubelet
        - --kubeconfig=/etc/kubernetes/ssl/kubeconfig
        - --api_servers=https://kubernetes.kubernetes.rancher.internal:6443
        - --allow-privileged=true
        - --register-node=true
        - --cloud-provider=rancher
        - --healthz-bind-address=0.0.0.0
        - --cluster-dns=169.254.169.250
        - --cluster-domain=cluster.local
        - --network-plugin=cni
        - --network-plugin-dir=/etc/cni/net.d
    image: wlan0/k8s:v1.4.0-rancher1
```

Supported deployment types

- **Standalone**

No resilience to host failure, mostly used for development and training

- **Resilient Overlapping Data and Control Planes**

Resilient to hosts failures. Potential performance issues with etcd/kubernetes components

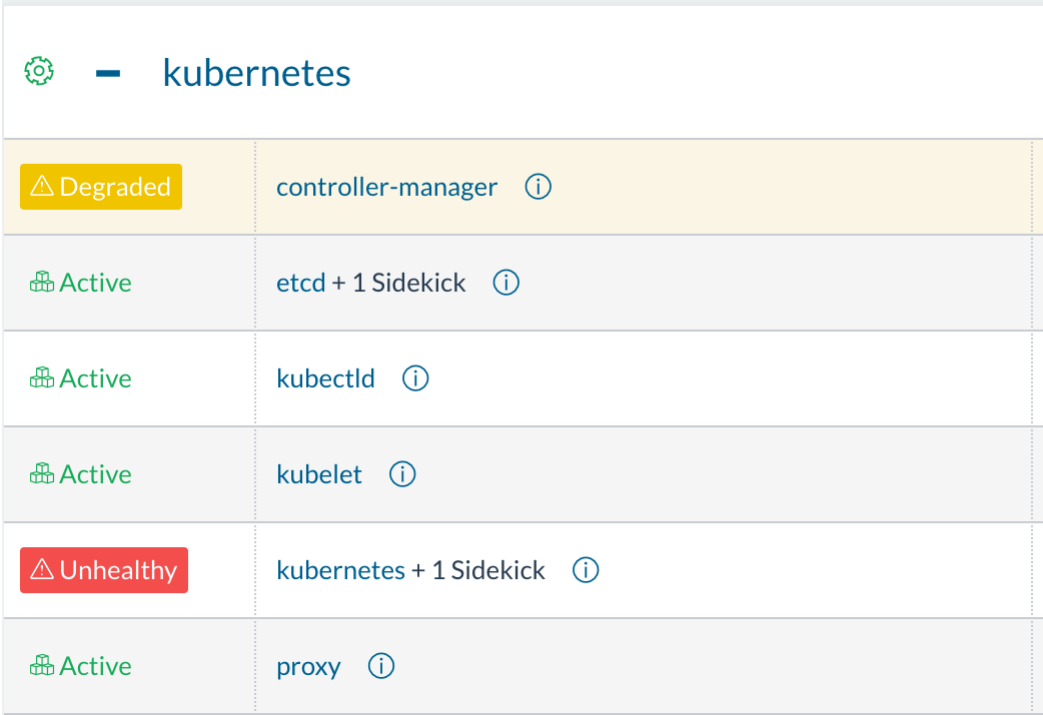
- **Resilient Separated-Planes**

High-performance, production-ready environment

Security

- Certificates are seeded to all components automatically providing secure access **within** the cluster
- Rancher Authentication plugin manages access **to** the cluster

#3 Controller plane HA



RANCHER

Health Checks in Rancher

- Health check agent runs on every host
- Up to 3 agents check on every health check enabled container
- Unhealthy container gets reported and replaced

09:12:05 PM	INFO	service.update.info	Service reconciled: Requested: 1, Created: 1, Unhealthy: 0, Bad: 0, Incomplete: 0
09:12:03 PM	INFO	service.instance.create	Creating extra service instance
09:12:03 PM	INFO	service.update.wait (2 sec)	Waiting for instances to start
09:12:02 PM	INFO	service.instance.delete	Removing unhealthy service instance

Various health failure recover strategies to suit different types of services

Regular aka “always recreate”

```
kubernetes:
  scale: 1
  retain_ip: true
  health_check:
    port: 80
    interval: 2000
    unhealthy_threshold: 3
    strategy: recreate
    response_timeout: 2000
    request_line: GET /healthz HTTP/1.0
    healthy_threshold: 2
```

Recreate on quorum, used by etcd cluster

```
etcd:
  scale: 1
  retain_ip: true
  scale_policy: &id001
    increment: 1
    max: 3
    min: 1
  health_check:
    port: 2378
    interval: 5000
    recreate_on_quorum_strategy_config:
      quorum: 2
    unhealthy_threshold: 3
    strategy: recreateOnQuorum
    response_timeout: 3000
    request_line: GET /health HTTP/1.0
    healthy_threshold: 2
```


RANCHER

#4 Reliable etcd cluster

- Having 3 or more hosts usually helps to be resilient to minority of hosts failing
- But for an enterprise, even a large one, losing quorum is not uncommon
- Periodic backup of etcd clusters helps to restore and recover after quorum loss

Etcd cluster in Rancher supports:

1. Configuring remote backups
2. Failure Recovery
3. Disaster Recovery
4. Restoring Backups

#5 Automated upgrades

- In Rancher we follow Kubernetes release process and test every major and almost every minor version
- Once tested, Rancher Infrastructure Catalog gets a new published Kubernetes template
- Current Kubernetes clusters get a notification and an option to upgrade to the latest

One click upgrade

Stack: kubernetes

Add Service | Upgrade available

Upgrade: kubernetes

kubernetes

Catalog: Infra

Category: Orchestration

Upgrade Kubernetes Stack

Rancher Kubernetes service

Template Version

v1.4.0-rancher1

Select a version of the template to upgrade to

RANCHER

RANCHER