Building Kubernetes cloud: real world deployment examples, challenges and approaches

Alena Prokharchyk, Rancher Labs
Making a right choice is not easy
The illustrated children guide to Kubernetes

https://www.youtube.com/watch?v=4ht22ReBjno
Kubernetes cluster components

How they fit together

- etcd
  - stores all the state!
  - runs on every node
- kubectl
- API server
  - controller manager
  - scheduler
- Kubelet
- kube-dns
- dashboard
- kube-proxy

Image courtesy of Julia Evans https://stripe.com/blog/operating-kubernetes
How do I architect my Kubernetes cluster? Or clusters?

- What is my infrastructure
- Do I need one big cluster, or many small ones
- Challenges specific to a particular installation
Today we are going to build Kubernetes clusters for ...
a Big Corporate Company
and a Franchise Chain
Questions to answer

- Do I provision k8s on baremetal hosts or vms
- If I chose vms, do I host them on premises or in the cloud
- If in the cloud, do I use hosted k8s service, or provision k8s myself
- How to structure k8s cluster - one big cluster or multiple smaller ones
Bare metal or vm

Bare metal

- Pros: performance, customization
- Cons: time consuming to provision and maintain

Edge: usually bare metal - a host or a device
VM: on premises or in the cloud

Cloud

- **Pros:** autoscaling, pay-as-you-go
- **Cons:** less control over infrastructure, vendor lock in, security
Cloud managed Kubernetes

- **Pros:** HA is managed by a cloud vendor
- **Cons:** less control over Kubernetes configuration
How to structure your k8s cluster

- One big cluster cross all the hosts
- Many smaller clusters collocated by a location or device
The Big Corporate Company

- One cluster might work if the company is not Geo distributed
- Otherwise deploy one cluster per region
The Franchise Chain

One cluster per
- Location
- Device

Image courtesy of Daniel sanchehttps://goo.gl/AbMUJS
In any case you need a cluster management platform

- Rancher
- OpenShift
- Pivotal
Rancher 2.0

- Built on top of Kubernetes, with *etcd* as a database
- Written in *Go*
- Rancher API extends Kubernetes API
- Every Rancher resource is a Kubernetes *CRD* (Custom resource definition)
- Functional components run as Kubernetes *controllers*
Rancher Architecture

- Secrets propagation
- Public endpoints population
- NetworkPolicy configuration

RANCHER SERVER

Management controllers

User cluster controllers

User cluster

User cluster

User cluster

- Cluster provisioner
- Cluster stats
- Cluster healthcheck
- Authentication
- Catalog manager

etcd

etcd

etcd
The Big Corporate Company
Things to take care of

- Multi tenancy (RBAC)
- Network isolation between tenants via Network Policies
- Resource quotas
RBAC in Rancher

- Extends Kubernetes RBAC

- Introduces a concept of a **Project** to allow namespaces grouping

- Self service access - once user is added to the project, it automatically inherits all the permissions
Network Isolation via Network policies
Network Isolation via Network policies
Resource quotas

Rancher Admin creates Project Resource Quota. Resource quota propagates to namespaces.
If you have multiple clusters, you might want to provide a global access across them.
Public access cross clusters using Global DNS

- Rancher dynamically collects public endpoints from user clusters
- Programs them to an external DNS
- External DNS provider is pluggable
The Franchise Chain
Things to take care of

- Cluster should be able to run on a really small device
- ARM support is desirable
- A manageable way to push an application to multiple clusters and do its upgrade with 1 click
Kubernetes distro is not small

Running it on 1GB RAM node can be problematic
k3s: The lightweight Kubernetes distribution built for the Edge

- Production grade and certified
- One Kubernetes binary with zero host dependencies
- Using `containerd` instead of Docker as a runtime
- SQLite as an optional datastore
- 40 Megabytes in size
- Opensource
**Removes**

- Legacy and non-default features
- Alpha features
- In-tree cloud providers
- In-tree storage drivers
- Docker (optional)

**Adds**

- Simplified installation
- SQLite3 support in addition to etcd
- TLS management
- Automatic Manifest and Helm Chart management
  - containerd, CoreDNS, Flannel
Edge use case also calls for

- Ability to push the same app to many clusters at once
- With a way to override the config on per cluster level
- Do an upgrade with 1 click
Multi cluster app
Multi cluster app

- A single global Interface to deploy same application to multiple Kubernetes clusters

- User can choose clusters/projects to target app scheduling and customize the configuration per target cluster

- The app can be deployed in Kubernetes clusters no matter how the clusters are provisioned, with no cloud provider lock-in

- Using Helm Support, deploy/upgrade functionality helps to manage the multi cluster app and maintain versioning
Common and crucial to have for all types of clusters

- Monitoring
- Alerting
- Logging
Built in Prometheus metrics

On multiple levels:

- Cluster
- Node
- Project
- Pod
Demo time
Links

- Rancher - Kubernetes management system - https://github.com/rancher/rancher
- k3s - lightweight Kubernetes - https://github.com/rancher/k3s
Thank you!

Alena Prokharchyk,
Principal Software Engineer @RancherLabs

@lemonjet

alken1108