Kubernetes & zke

zcloud

Kubernetes

- helmsman in Greek, k8s in short
- Open source by Google in 2014 summer, and hand over to CNCF

Orchestration

- container scheduling
- Scaling
- Health checking
- Upgrading
- Service discovery
- High level abstraction

Design principle

- Declare over imperative
- No hidden api
- Event driven
- Work load portability

primitives

- pod
- deployment
- service
- Ingress

pod

- A set of containers
  - Atomic unit for scheduling
  - Share one ip address and port space
  - Communicate through localhost & IPC

deployment

- One or more pods with replication
- self-healing support
- Application upgrade support

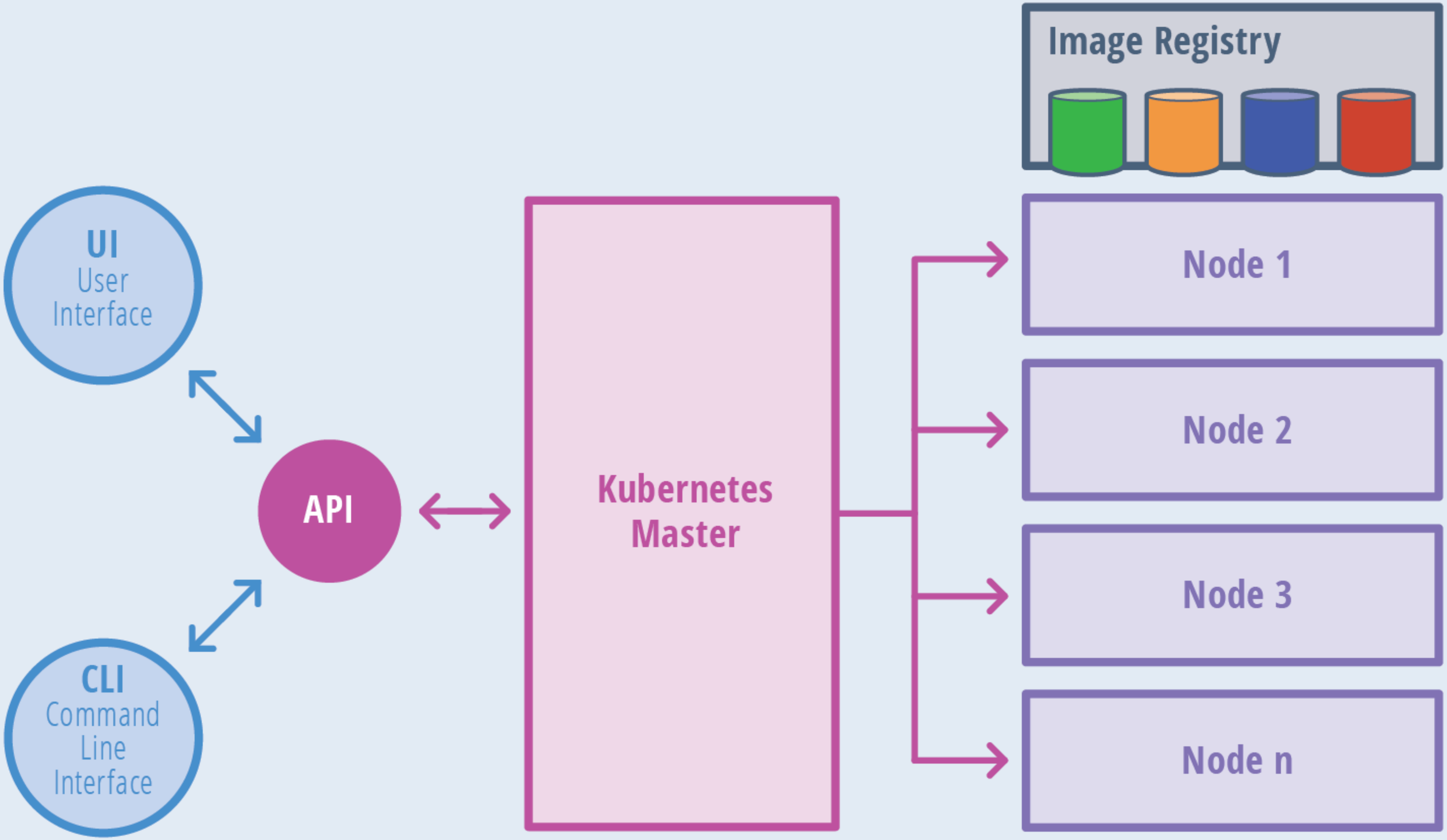
Service

- Expose a set of pod, normally a deployment with one virtual ip address
- Support load balance
- Use DNS as service + ENV variable as discovery mechanism
- Service types
    - Clusterip : expose as a inner cluster virtual endpoint
    - nodeport: nodeip + specific port
    - Loadblancer: cloud provider's solution
- Service network functionality is implemented by kube-proxy.

Ingress

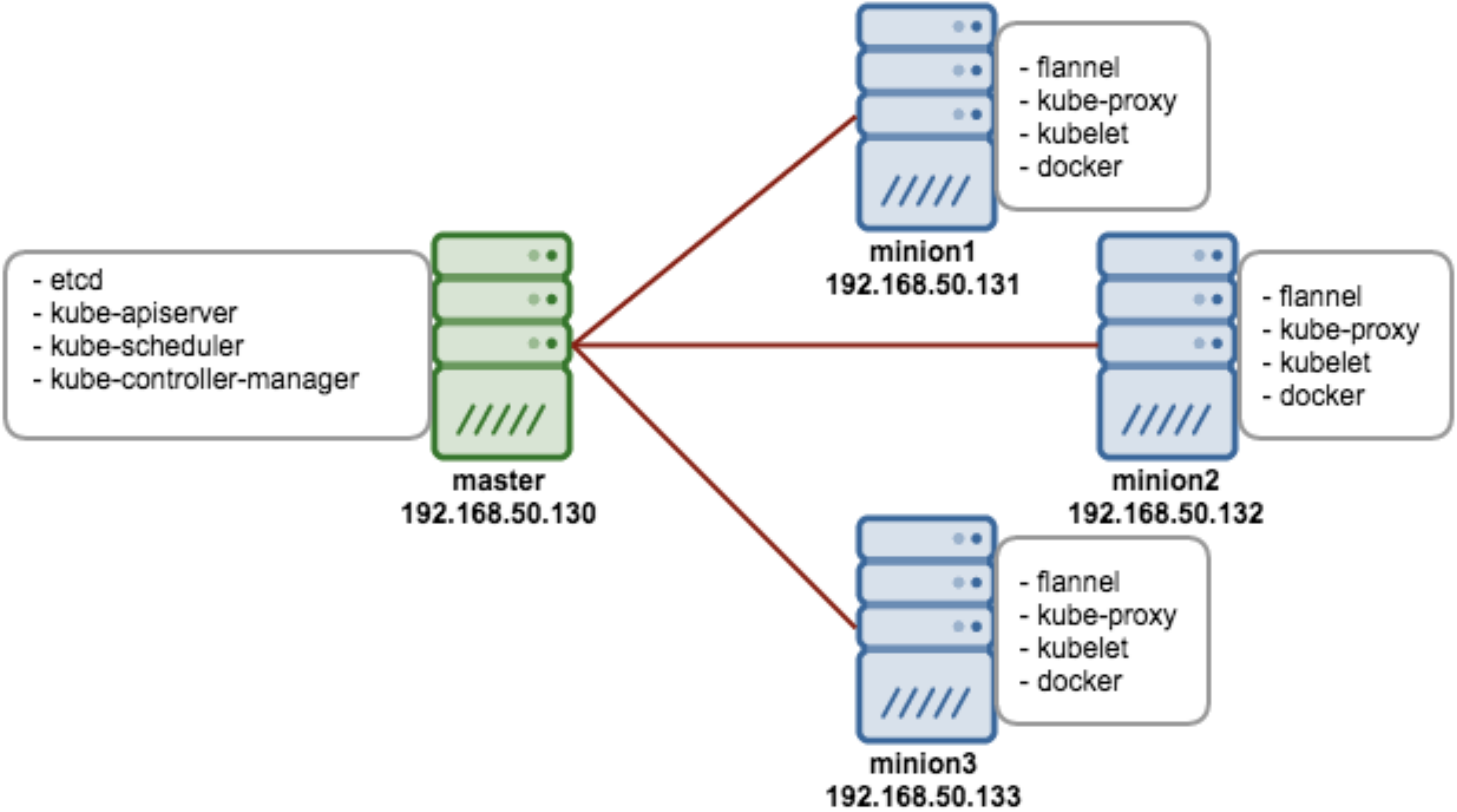
- Expose cluster service to internet(north-south traffic)
- Rule based, use url as key to dispatch to different inner service
- Third party controller to implement the functionality

Kubernetes Architecture



Source: Janakiram MSV

THENEWSTACK



- etcd
- kube-apiserver
- kube-scheduler
- kube-controller-manager

master
192.168.50.130

minion1
192.168.50.131

- flannel
- kube-proxy
- kubelet
- docker

minion2
192.168.50.132

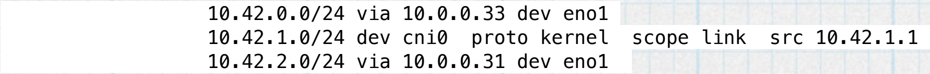
- flannel
- kube-proxy
- kubelet
- docker

minion3
192.168.50.133

- flannel
- kube-proxy
- kubelet
- docker

Network

- Three address spaec
    - Node address space
    - Pod address space
    - Service address space
- Pod to pod, pod to node(east to west) without nat
    - Pod address is allocated by CNI
    - L2 + l3 solution: linux bridge + node route table
    - Flannel add route info into each node



```
10.42.0.0/24 via 10.0.0.33 dev eno1
10.42.1.0/24 dev cni0  proto kernel  scope link  src 10.42.1.1
10.42.2.0/24 via 10.0.0.31 dev eno1
```

- Service network is implemented by kube-proxy

zke

- based on RKE (from rancher)
- Focus on local cluster

Requirements

- Linux distribution
- Docker
    - 17.03.x, 17.06.x, 17.09.x, 18.06.x
- Ssh connection without password
- User in docker group (has sudo privileged?)

Nodes

- Etcd cluster
- Control plane
  - API server
  - Kube controller
  - Kube scheduler
  - kubelet/kubeproxy
- Worker plane
  - Kubelet
  - Kubeproxy

Steps

- Create certificate and keys
- Copy keys to all nodes
- Run etcd plane
- Run control plan
- Run worker plan
- Deploy network plugin
- Deploy other addons
    - DNS
    - Metric
    - Ingress

- Demo