

Deploy a k8s cluster in arm64

Liang(Leon) Yan



Kubernetes

What is Kubernetes?

- A highly collaborative open source project originally conceived by Google
- Google has 10+ years experience w/ containerized apps

many ideas coming from geard

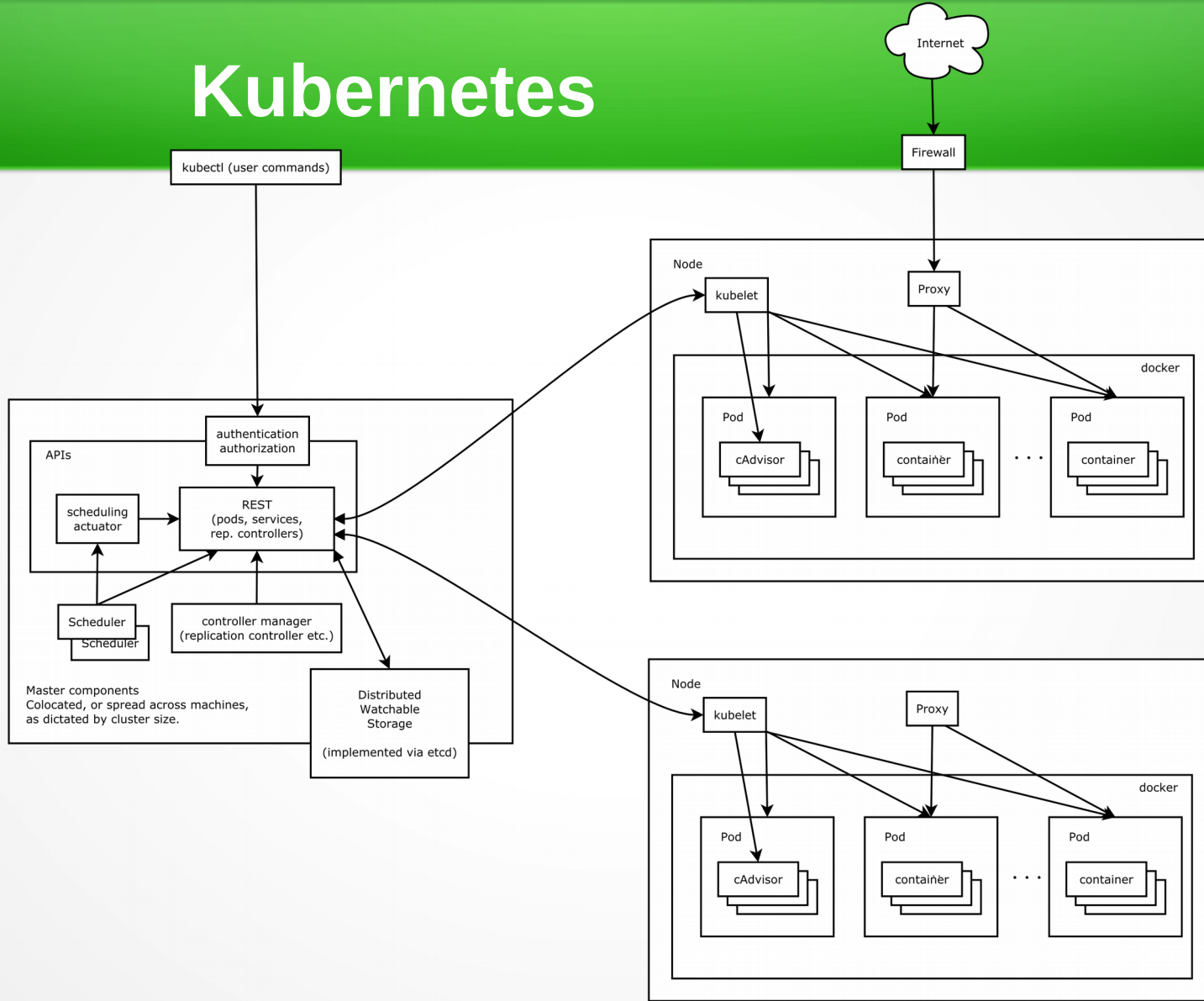
- Sometimes called:
- kube
- k8s (that's 'k' + 8 letters + 's')
- Start, stop, update, and manage a cluster of machines running containers in a consistent and maintainable way.

Kubernetes

- K8s-master:
api-server controller-manager scheduler ETCD
- K8s-minion
kubelet kube-proxy cAdvisor
- Cli
kubectl/ui
- Others
Flanneld or other network and storage components
- Concepts
Pod Replication Controller Service Label Namespace

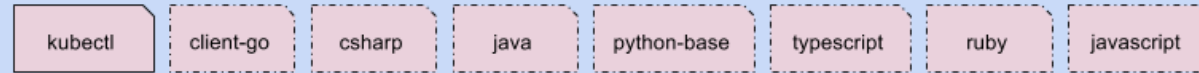


Kubernetes

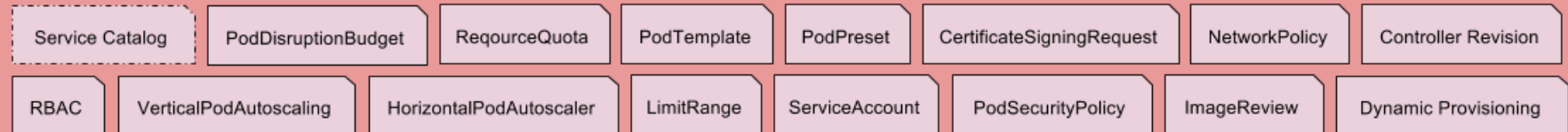


Kubernetes Core Components

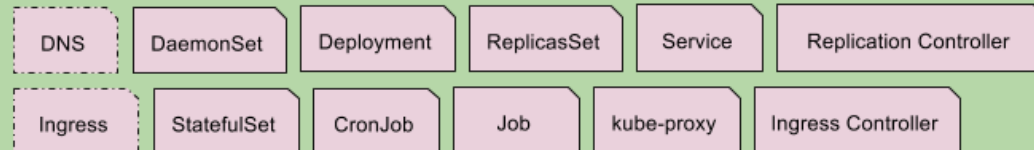
Interface Layer: Client Libraries and Tools



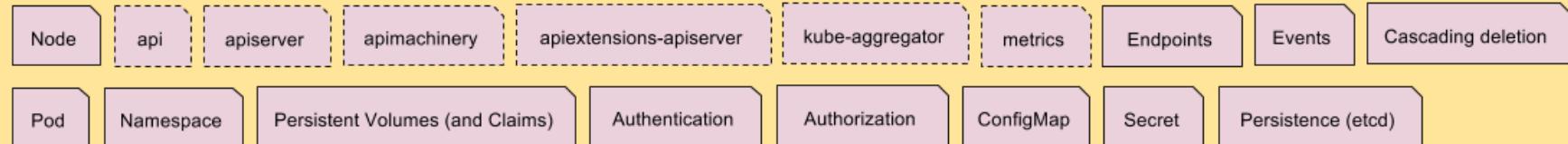
Governance Layer: Automation and Policy Enforcement (APIs optional and pluggable)



Application Layer: Deployment and Routing (APIs required and pluggable)



Nucleus: API and Execution (APIs required and not pluggable)



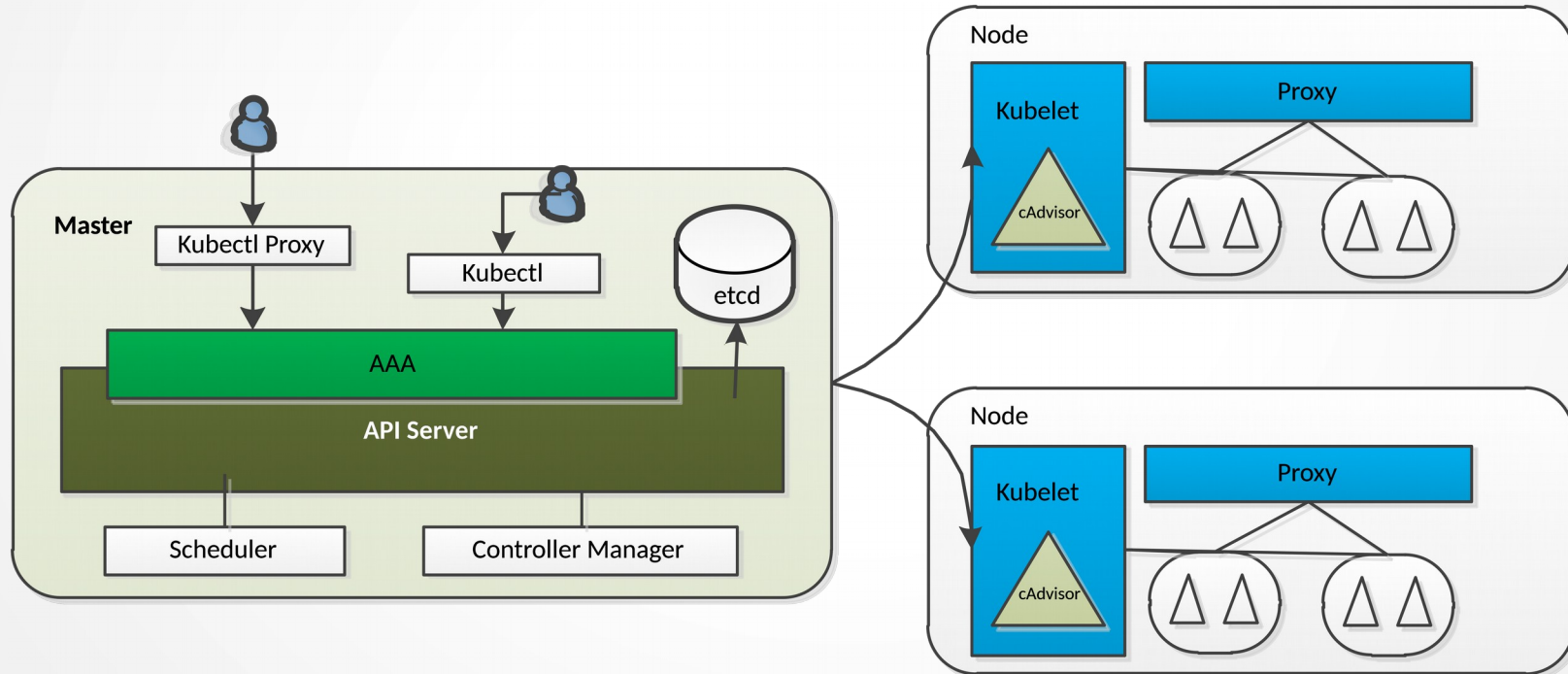
Notes:

1. Dashed border items to be broken out of kubernetes/kubernetes and become their own repos.

2. Dashed dot bordered are already in their own repo.

Kubernetes

API server



Kubernetes

Controller Manager

Replication
Controller

Node Controller

ResourceQuota
Controller

Namespace
Controller

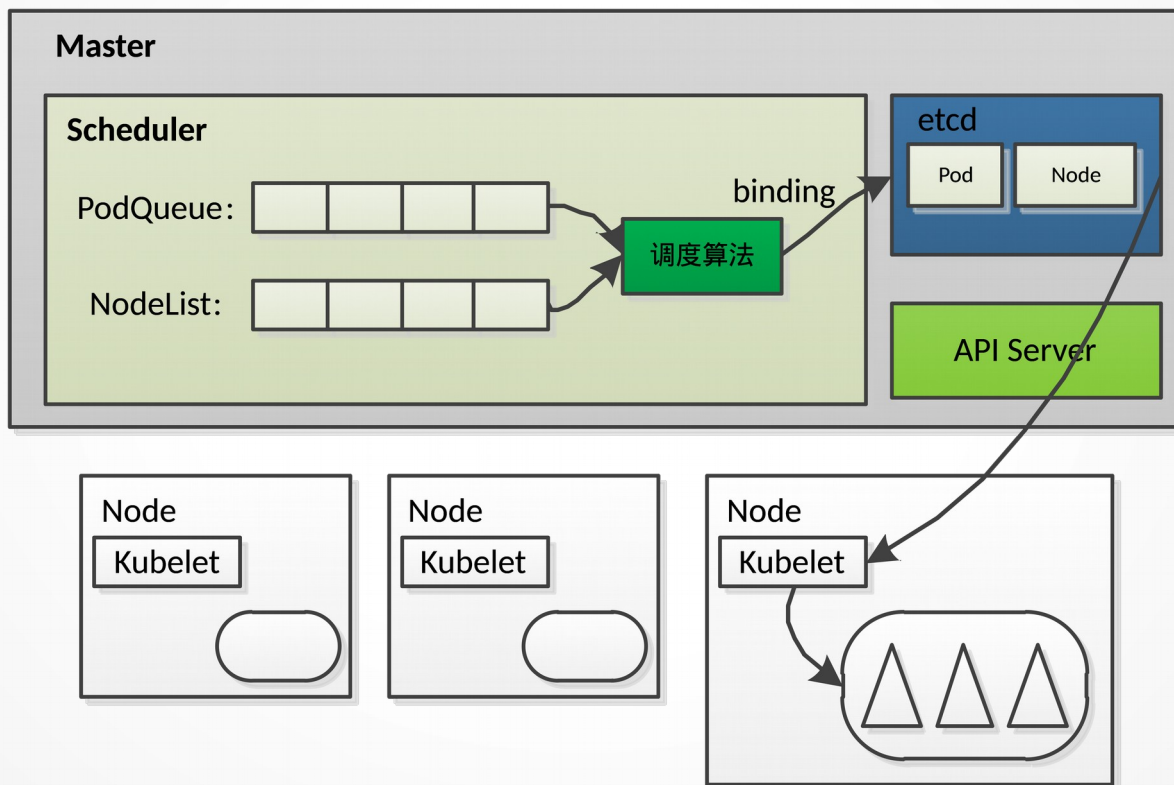
ServiceAccount
Controller

Token
Controller

Service
Controller

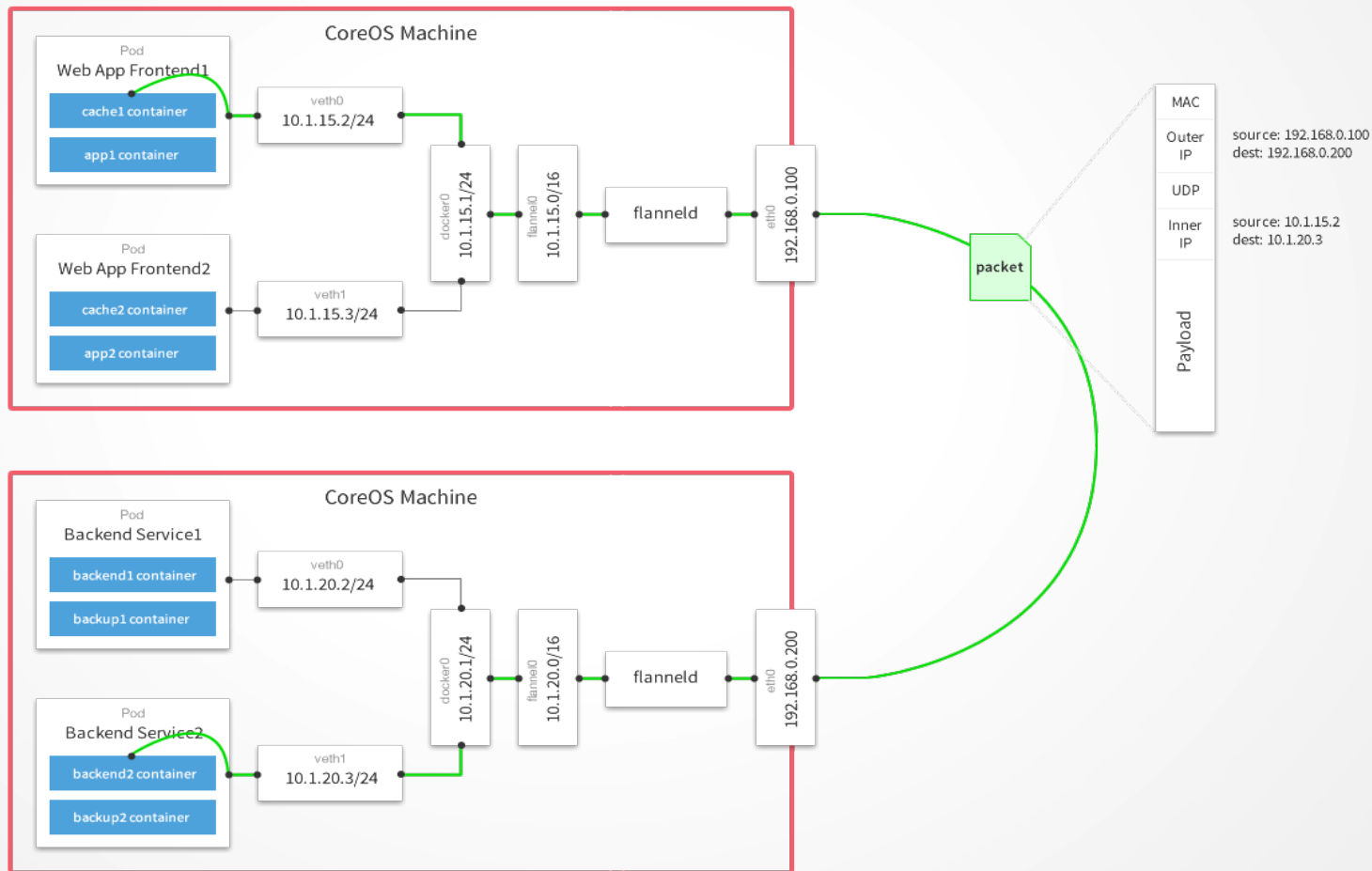
Endpoint
Controller

Kubernetes



Kubernetes

Overlay Network Setup



Kubernetes

Hosted Services: Google Compute Engine

- Support for a wide variety of Infrastructure (Azure, Rackspace, vSphere, AWS)
- Support for several OS' (SUSE, RHEL, Debian, Ubuntu, Atomic, CoreOS)
- Local but automated (Vagrant/Ansible) * Magic *
- Local but manual * What I use *

<https://github.com/liayan/deployk8sonarm64>

Kubernetes

Use cfssl create Certificate Authority:

```
cfssl print-defaults config > config.json
```

```
cfssl print-defaults csr > csr.json
```

```
cfssl gencert -initca ca-csr.json | cfssljson -bare ca
```

```
$ ls ca*
```

```
ca-config.json  ca.csr  ca-csr.json  ca-key.pem  ca.pem
```

Kubernetes

Etcd clusters

```
ExecStart=/root/local/bin/etcd \\  
  --name=${NODE_NAME} \\  
  --cert-file=/etc/etcd/ssl/etcd.pem \\  
  --key-file=/etc/etcd/ssl/etcd-key.pem \\  
  --peer-cert-file=/etc/etcd/ssl/etcd.pem \\  
  --peer-key-file=/etc/etcd/ssl/etcd-key.pem \\  
  --trusted-ca-file=/etc/kubernetes/ssl/ca.pem \\  
  --peer-trusted-ca-file=/etc/kubernetes/ssl/ca.pem \\  
  --initial-advertise-peer-urls=https://${NODE_IP}:2380 \\  
  --listen-peer-urls=https://${NODE_IP}:2380 \\  
  --listen-client-urls=https://${NODE_IP}:2379,http://127.0.0.1:2379 \\  
  --advertise-client-urls=https://${NODE_IP}:2379 \\  
  --initial-cluster-token=etcd-cluster-0 \\  
  --initial-cluster=${ETCD_NODES} \\  
  --initial-cluster-state=new \\  
  --data-dir=/var/lib/etcd
```

cfssl gencert

```
-ca=/etc/kubernetes/ssl/ca.pem \  
-ca-key=/etc/kubernetes/ssl/ca-key.pem \  
-config=/etc/kubernetes/ssl/ca-  
  config.json \  
-profile=kubernetes etcd-csr.json |  
  cfssljson -bare etcd
```

Kubernetes

Kubectrl: for all nodes that running kubectrl

1. CA
2. kubeconfig
./kube/config

Flannel:

- 1.CA
- 2.ExecStart=/root/local/bin/flanneld \\
-etcd-cafile=/etc/kubernetes/ssl/ca.pem \\
-etcd-certfile=/etc/flanneld/ssl/flanneld.pem \\
-etcd-keyfile=/etc/flanneld/ssl/flanneld-key.pem \\
-etcd-endpoints=\${ETCD_ENDPOINTS} \\
-etcd-prefix=\${FLANNEL_ETCD_PREFIX}

Will create: flannel.1:

Kubernetes

Kube master:

Kube-apiserver

kube- controller-manager

Kube-scheduler

Kube minion:

Docker

Kubelet

Kube-proxy

```
$ kubectl get componentstatuses
```

NAME	STATUS	MESSAGE
ERROR		

controller-manager	Healthy	ok
--------------------	---------	----

scheduler	Healthy	ok
-----------	---------	----

etcd-0	Healthy	{"health": "true"}
--------	---------	--------------------

etcd-1	Healthy	{"health": "true"}
--------	---------	--------------------

etcd-2	Healthy	{"health": "true"}
--------	---------	--------------------

```
kubectl get csr
```

```
kubectl certificate approve csr-2b308
```

```
$ kubectl get nodes
```

NAME	STATUS	AGE	VERSION
192.168.2.119	Ready	4h	v1.7.2
192.168.2.218	Ready	4h	v1.7.2

Kubernetes

Add-on:

Kube-dns

Kube-dashboard

Kube-heapster

Kube-influxdb

Kube-grafana

=====

Kube-ELK

- Thanks for your time here!!!
- Any questions ?

