

— // 实践课堂 / 第3季 / 深圳站 —

■ QingCloud Workshop Season 3 / Shenzhen ■

■ 第1期 / 2017.4.22 ■



青云QingCloud 容器及 Kubernetes 实践

王渊命 | 青云QingCloud 容器平台负责人

当我们谈论容器时我们在谈论什么？

容器的两个视角

- ▶ 资源隔离
- ▶ 应用封装

容器生态圈现状

- ▶ 容器技术一家独大百花齐放
 - ▶ Docker, Rocket, Mesos Universal container, LXC, Hyper Container
- ▶ 调度系统三足鼎立
 - ▶ Kubernetes, Mesos, Swarm

2016年容器技术思考：Docker, Kubernetes, Mesos 将走向何方？

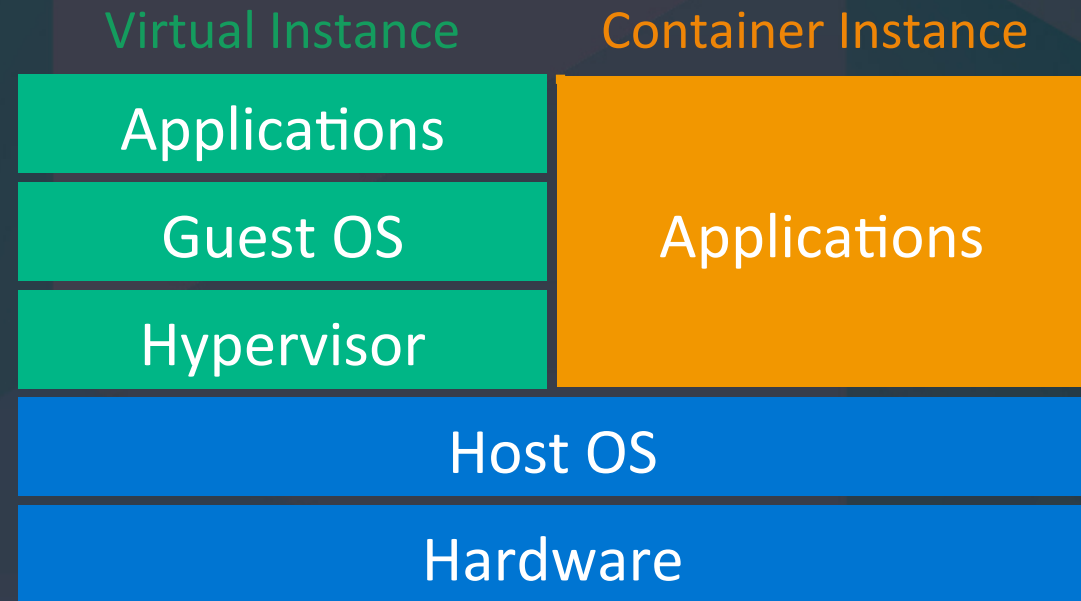
<http://jolestar.com/container-ecosystem/>

Container@QingCloud

- ▶ 资源视角
- ▶ 应用视角

Container@QingCloud — 资源视图

- ▶ QingCloud IaaS 调度系统
支持 Container Instance

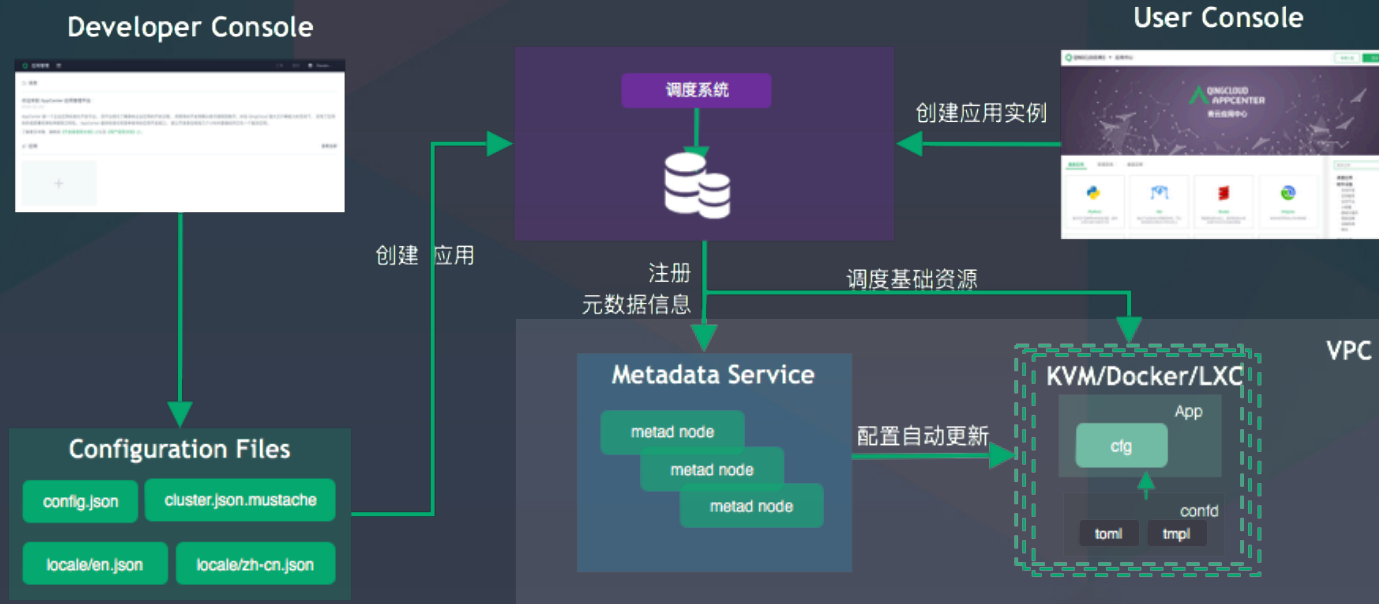


Container@QingCloud — 应用视图

▶ AppCenter 支持

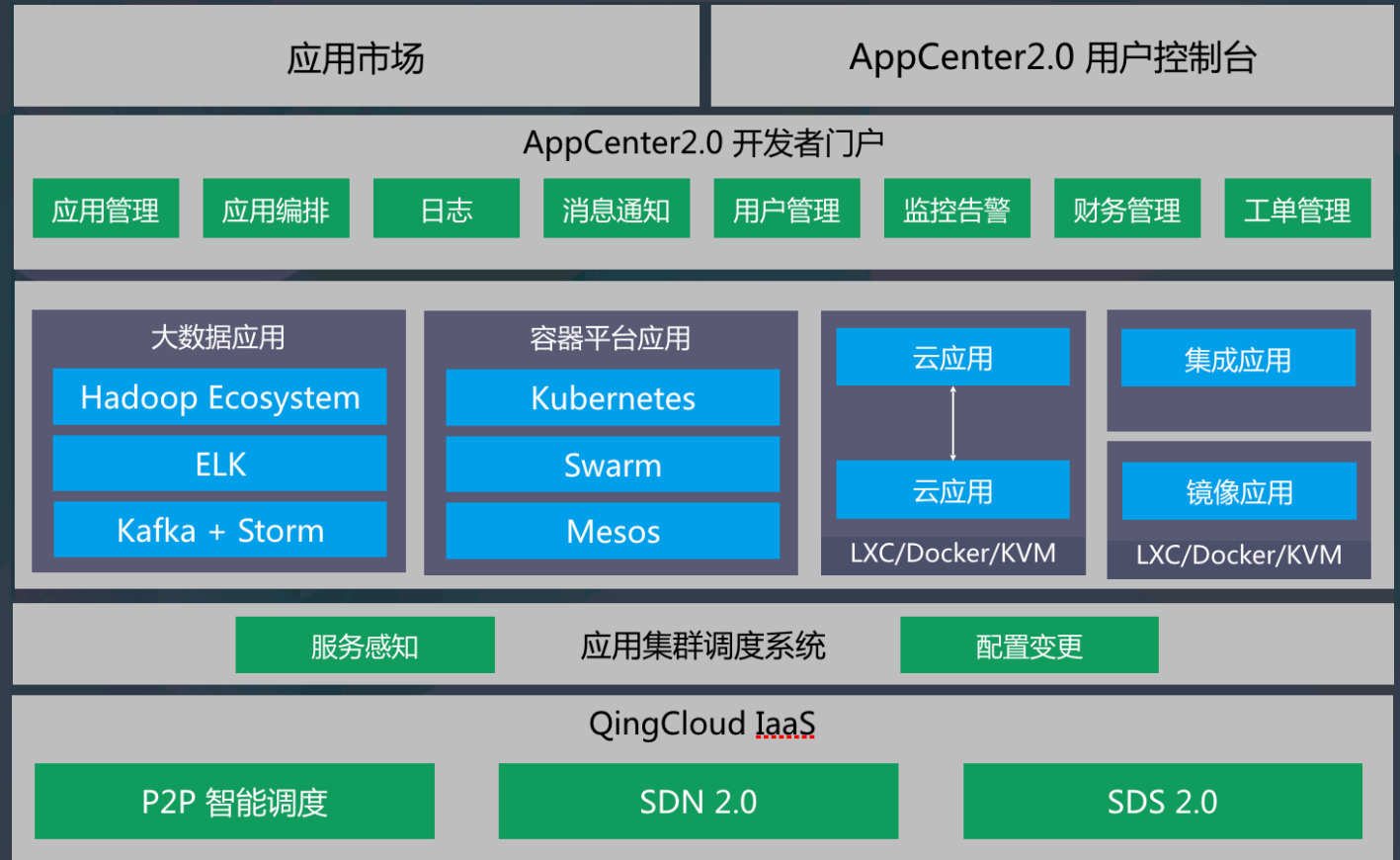
Docker 镜像

```
"container": {
  "type": "docker",
  "image": "zookeeper"
}
```



Container@QingCloud — 应用视图

▶ AppCenter 支持各
种容器编排系统



Kubernetes@QingCloud



网络



存储

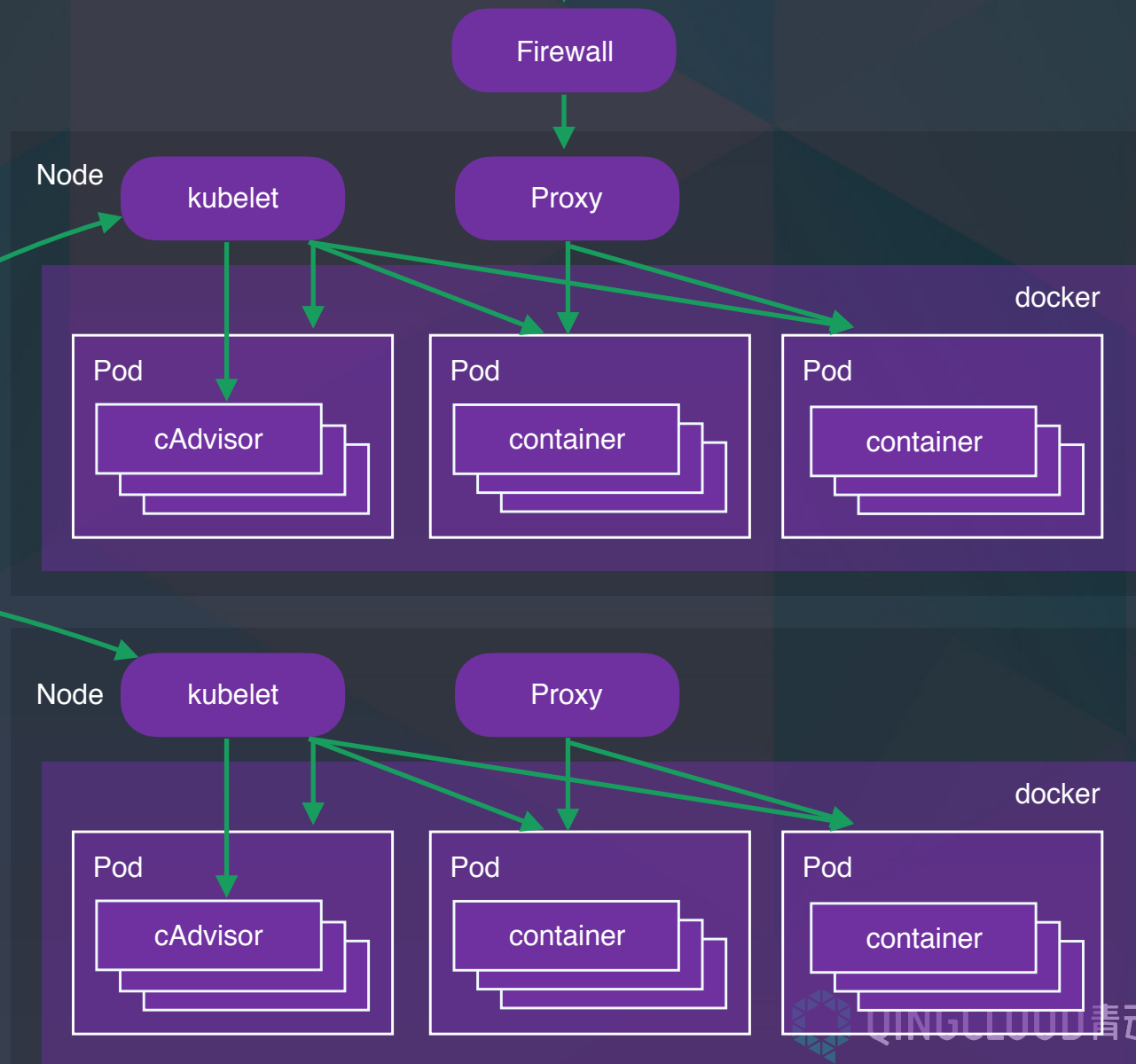
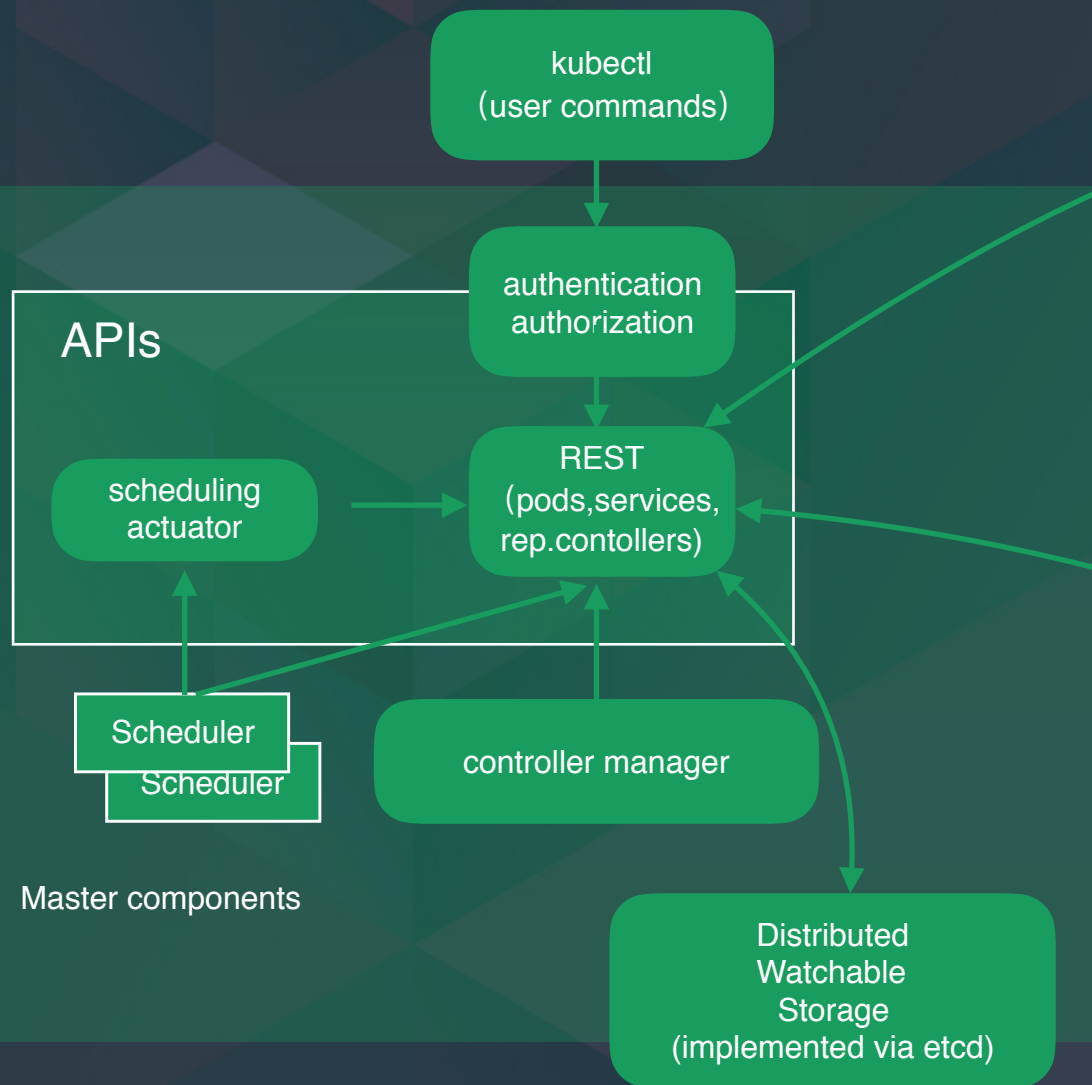


负载均衡



弹性

Kubernetes 概览



Kubernetes 抽象概念

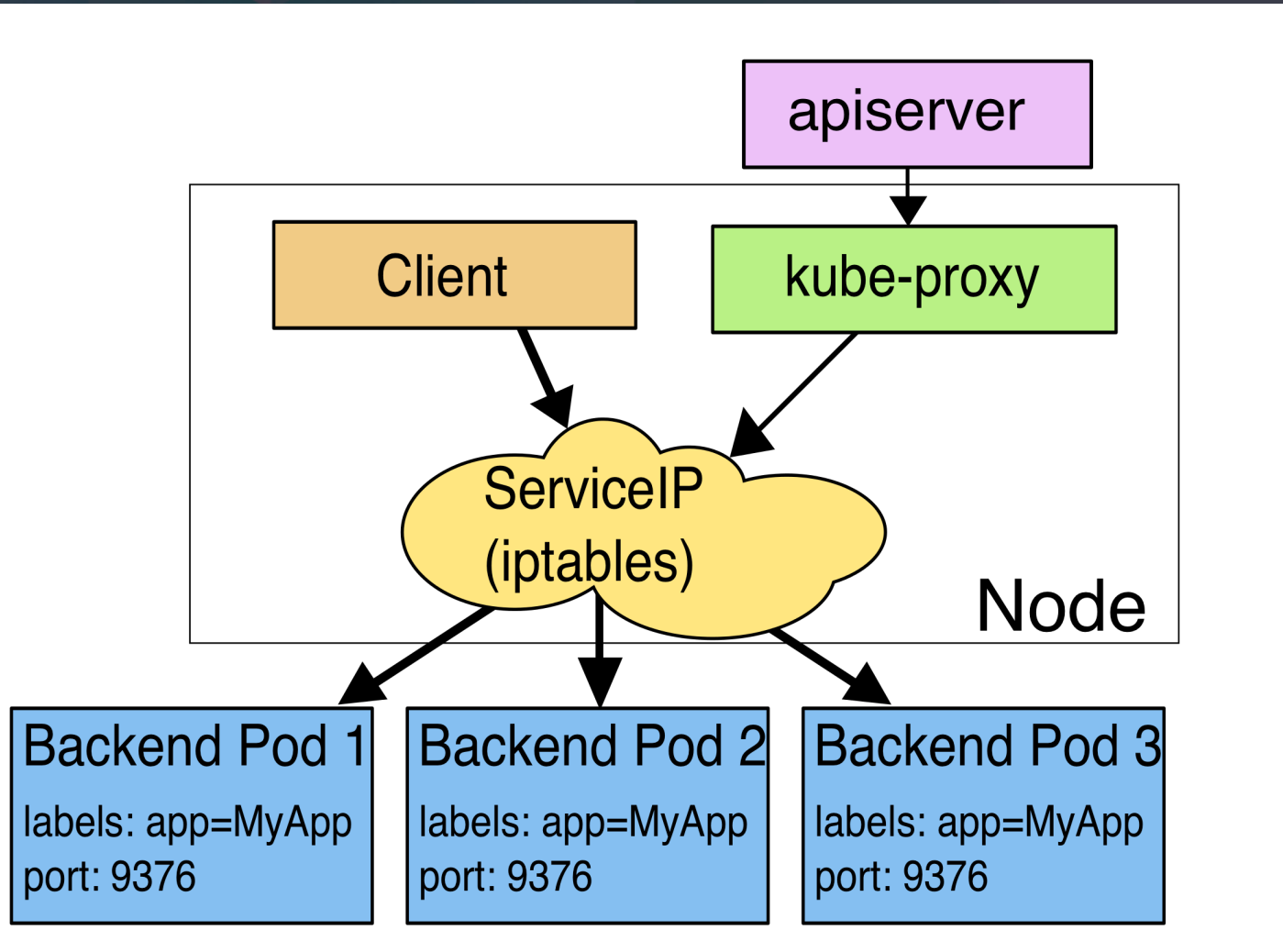
Service	StatefulSet(PetSet)
Job	PersistentVolume
ReplicaSet(ReplicationController)	PersistentVolumeClaim
Deployment	Namespace
DaemonSet	Label/Selector

Kubernetes 网络

- ▶ 容器之间可以直接互通，不需要 NAT
- ▶ 节点可以喝容器直接互通，不需要 NAT
- ▶ 容器看到自己的 IP 应该和其他容器看到的一样

- CNI (Container Network Interface)
 - Flannel
 - Calico
 - Contiv
- ClusterIP
 - Virtual IP
 - Iptables

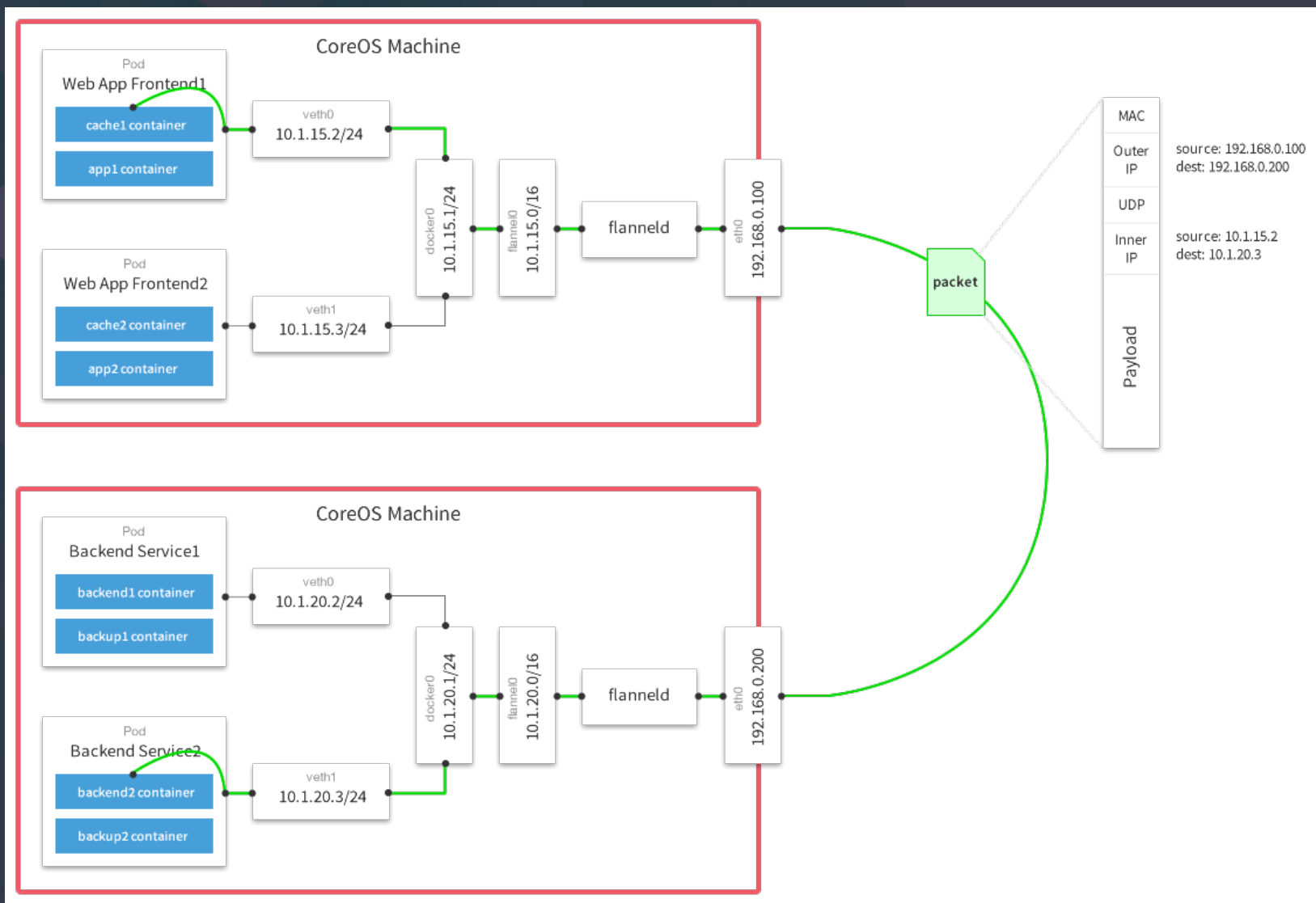
Kubernetes 网络之 ClusterIP



```

{
  "kind": "Service",
  "apiVersion": "v1",
  "metadata": {
    "name": "my-service"
  },
  "spec": {
    "selector": {
      "app": "MyApp"
    },
    "ports": [
      {
        "protocol": "TCP",
        "port": 80,
        "targetPort": 9376
      }
    ],
    "clusterIP": "10.0.171.239",
    "type": "LoadBalancer/NodePort"
  }
}
  
```

Kubernetes 网络之 flannel



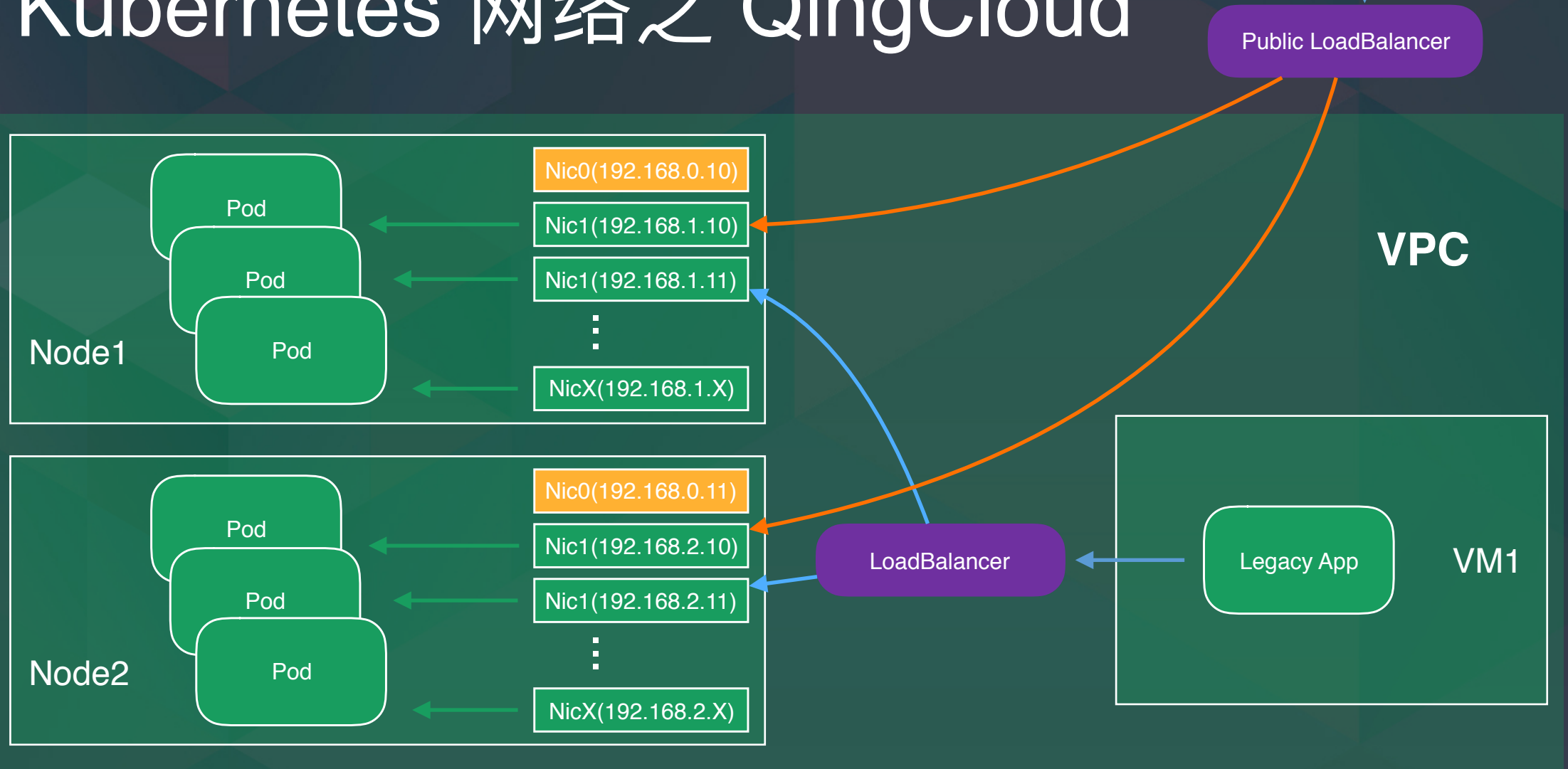
Discovery

- etcd
- kube-subnet-mgr

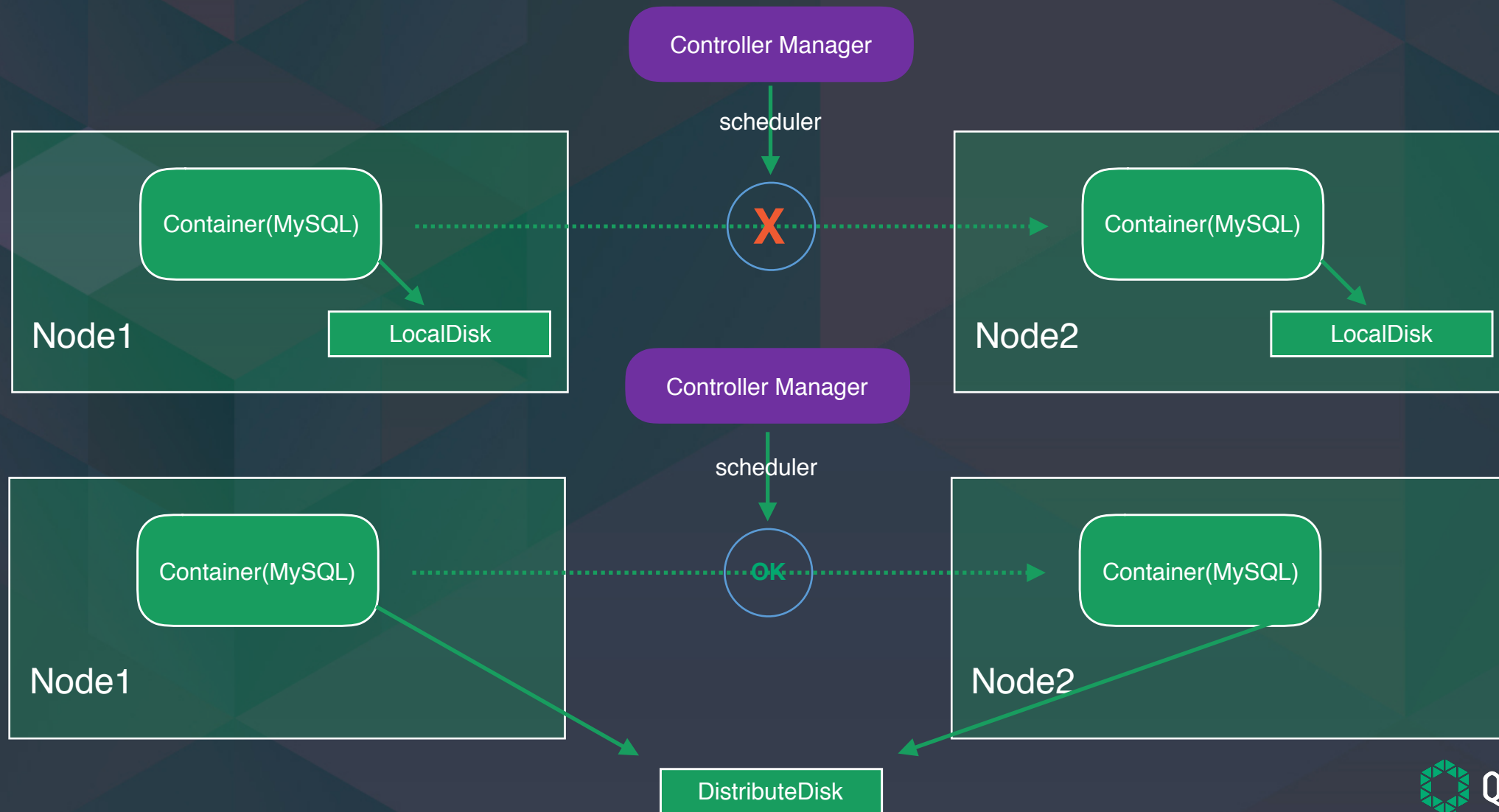
Backend

- vxlan
- aws-vpc

Kubernetes 网络之 QingCloud



Kubernetes 存储



Kubernetes 存储

▶ DistributeDisk

- nfs
- ceph
- glusterfs
- PersistentVolume plugin
 - gcePersistentDisk
 - awsElasticBlockStore
 - qingCloudStore

Kubernetes 存储之 QingCloudStore

- ▶ PersistentVolume Plugin
- ▶ PersistentVolume
- ▶ StorageClass
- ▶ PersistentVolumeClaim

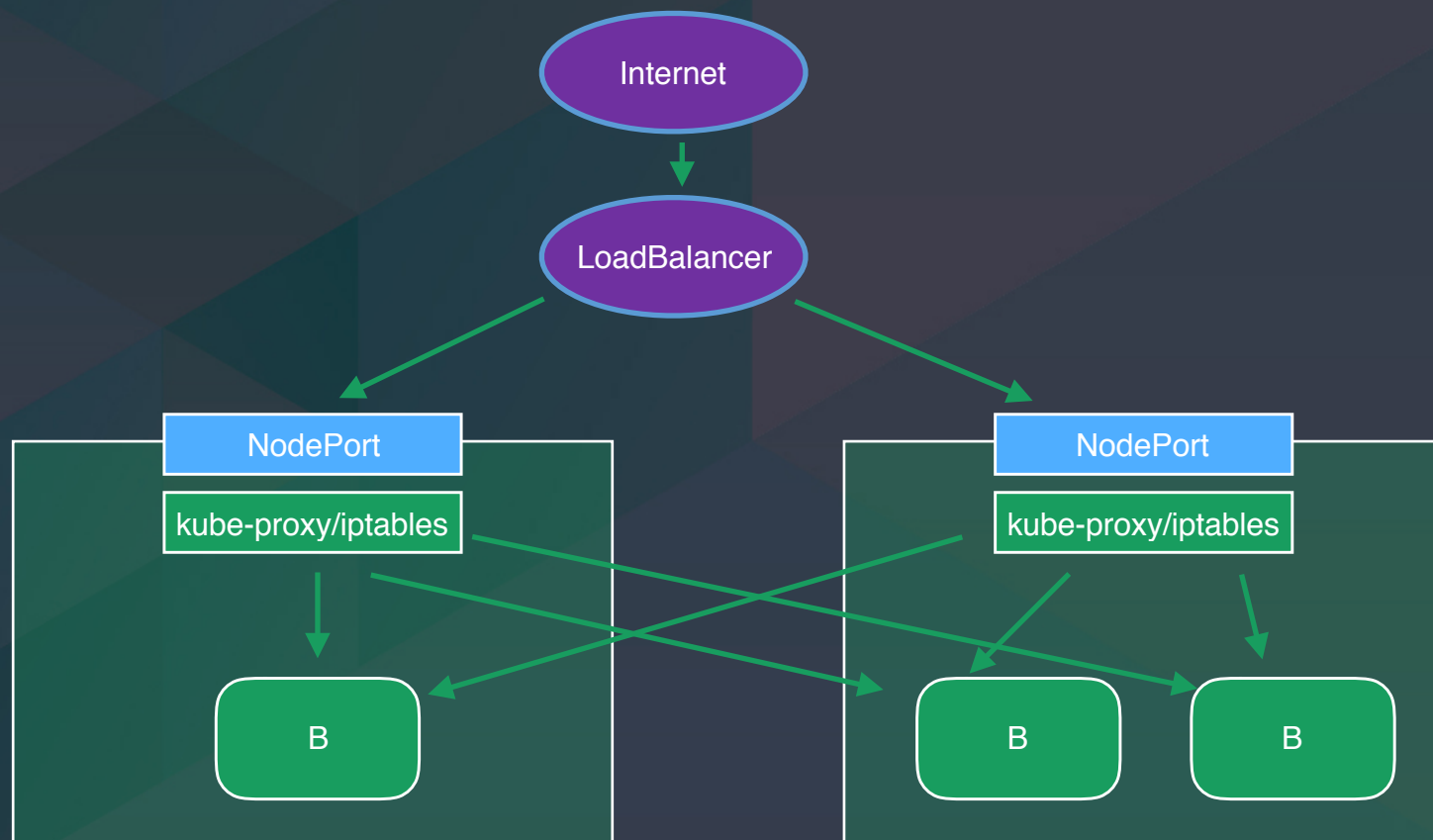
Kubernetes 存储之 QingCloudStore

```
kind: PersistentVolume
apiVersion: v1
metadata:
  name: qingcloud-pv
  labels:
    type: qingcloud
spec:
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteOnce
  qingCloudStore:
    volumeID: vol-caoxtg3
    fsType: ext4
```

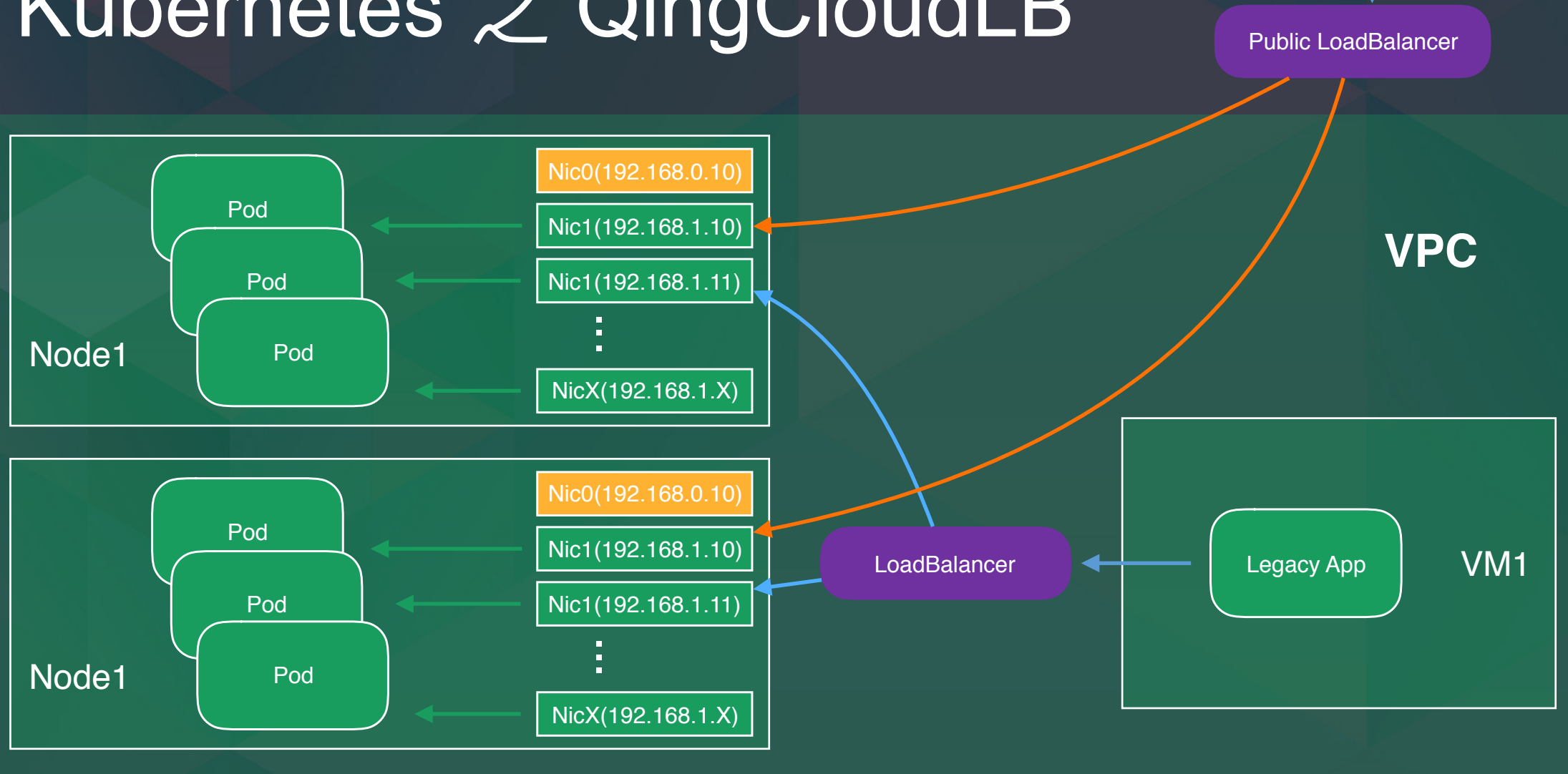
```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: qingcloud-pvc
  annotations:
    volume.beta.kubernetes.io/storage-class: qingcloud-storageclass
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 3Gi
```

```
kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
  name: qingcloud-storageclass
provisioner: kubernetes.io/qingcloud-volume
parameters:
  type: "3"
```

Kubernetes 负载均衡器



Kubernetes 之 QingCloudLB



Kubernetes 自动伸缩

► Deployment

```
kubectl autoscale deployment php-apache --cpu-percent=50 --min=1 --max=10
```

► 集群的自动伸缩 (TriggeredScaleUp Event)

```
gcloud container clusters create mytestcluster \  
--zone=us-central1-b --enable-autoscaling --min-nodes=3 --max-nodes=10 --num-nodes=5
```

Kubernetes@QingCloud

- ▶ 网络 (SDN Passthrough)
- ▶ 存储 (PersistentVolumePlugin)
- ▶ 负载均衡 (LB + SDN)
- ▶ 自动伸缩 (Event + IaaS API)

计划

- ▶ 5 月份公开试用
- ▶ AppCenter 支持 Kubernetes 应用规范
- ▶ Kubernetes 之上的服务