

Kubernetes 完全教程

Kubernetes 安装和运维

王渊命 @jolestar

Agenda

1. Kubernetes 的安装
2. Kubernetes 的组件和配置介绍
3. Kubernetes 的高可用
4. Kubernetes 的使用以及运维

Kubernetes 的安装 -- 条条大路通罗马

1. 云服务商托管 GCE(Google), AWS, Azure, Bluemix, QingCloud 等。
2. 自定义安装 kops, ansible, salt, juju
3. kubeadm

准备基础的 VM 镜像

1. Ubuntu 16.04.3
2. 安装 docker 以及基础工具包

```
apt-get update
apt-get install -y ebttables socat apt-transport-https bash-completion ntp wget

apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 --recv-keys 58118E89F3A9128
apt-add-repository 'deb https://apt.dockerproject.org/repo ubuntu-xenial main'
apt-get update

apt-cache policy docker-engine
apt-get install -y docker-engine

DEBIAN_FRONTEND=noninteractive apt-get -y -o Dpkg::Options::="--force-confdef" -o Dpkg:::
```

安装 kubelet, kubeadm

当前 kubelet 版本 1.7.6

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -  
cat <<EOF >/etc/apt/sources.list.d/kubernetes.list  
deb http://apt.kubernetes.io/ kubernetes-xenial main  
EOF  
apt-get update  
apt-get install -y kubelet kubeadm  
# 安装 bash 自动提示  
kubeadm completion bash >/etc/profile.d/kubeadm.sh  
kubectl completion bash >/etc/profile.d/kubectl.sh  
source /etc/profile
```

初始化 master

```
kubeadm init --pod-network-cidr=10.244.0.0/16
```

复制并保存 init token

查看 pod 状态

```
kubectl get pods --all-namespaces
```

查看 node 状态

```
kubectl get nodes
```

初始化网络

```
kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/k
```

查看 pod 状态

```
kubectl get pods --all-namespaces
```

查看 node 状态

```
kubectl get nodes
```

新增节点

```
kubeadm join --token $init_token $apiserver-advertise-address:6443
```

查看 node 状态

```
kubectl get nodes
```

部署 helloworld

```
kubectl apply -f https://raw.githubusercontent.com/jolestar/kubernetes-complete-course/m
```

查看 pod 状态

```
kubectl get pods
```

测试 pod 之间以及 pod 和 apiserver 之间的网络

```
kubectl exec $podname -- nping $pod2ip  
kubectl exec $podname -- curl -k https://kubernetes
```

解决 flannel 网络问题

```
iptables -P FORWARD ACCEPT
```

1. <https://github.com/coreos/flannel/issues/799>
2. https://docs.docker.com/engine/userguide/networking/default_network/container-communication/#container-communication-between-hosts

For security reasons, Docker configures the iptables rules to prevent containers from forwarding traffic from outside the host machine, on Linux hosts. Docker sets the default policy of the FORWARD chain to DROP

Note: In Docker 1.12 and earlier, the default FORWARD chain policy was ACCEPT. When you upgrade to Docker 1.13 or higher, this default is automatically changed for you

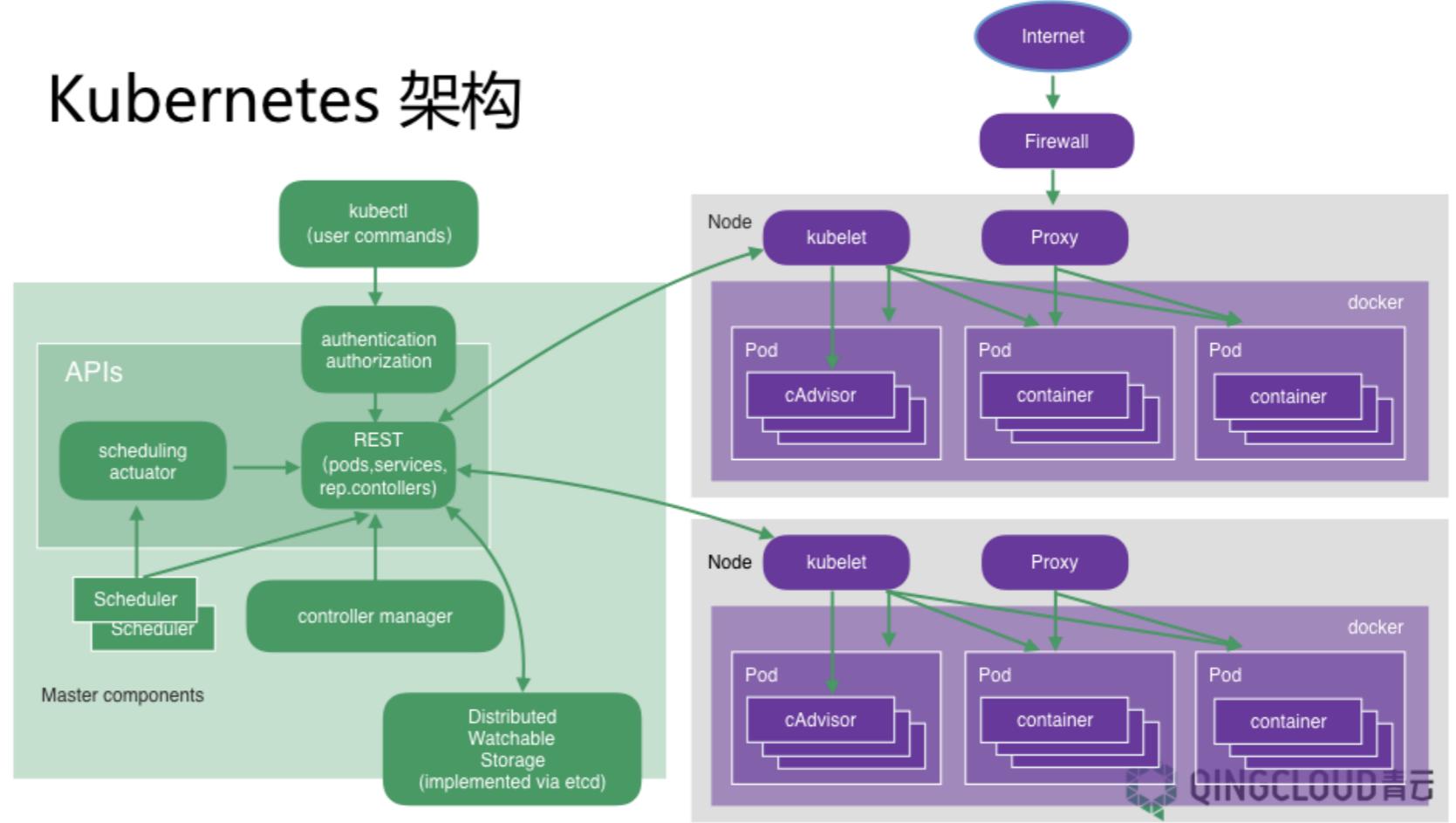
重新测试 pod 之间以及 pod 和 apiserver 之间的网络

关于 Kubernetes 网络以及网络故障的排查，将在后面的 Kubernetes 网络课程里介绍

Kubernetes 的组件

1. kubelet
2. kube-controller-manager
3. kube-scheduler
4. kube-apiserver
5. kube-proxy
6. kube-dns
7. etcd
8. flannel

Kubernetes 架构



Kubelet

1. 启动方式 系统进程
2. 配置文件
 - /lib/systemd/system/kubelet.service
 - /etc/systemd/system/kubelet.service.d/10-kubeadm.conf
3. 主要参数
 - kubeconfig bootstrap-kubeconfig
 - pod-manifest-path
 - allow-privileged(host-network-sources,host-pid-sources,host-ipc-sources)
(file,http,api)
 - network-plugin
 - authorization-mod (Webhook、AlwaysAllow)
 - cluster-dns=10.96.0.10 --cluster-domain=cluster.local
 - feature-gates

Kube-controller-manager 和 kube-scheduler

1. 启动方式 StaticPod 或 系统进程
2. 配置文件
 - /etc/kubernetes/manifests/kube-controller-manager.yaml kube-scheduler.yaml
3. 主要参数 (kube-controller-manager)
 - kubeconfig
 - allocate-node-cidrs
 - cluster-cidr=10.244.0.0/16
 - service-cluster-ip-range=10.96.0.0/12
 - leader-elect
 - feature-gates

Kube-apiserver

1. 启动方式 StaticPod 或 系统进程
2. 配置文件
 - /etc/kubernetes/manifests/kube-apiserver.yaml
3. 主要参数
 - kubeconfig
 - insecure-port insecure-bind-address
 - allow-privileged
 - kubelet-preferred-address-types=InternalIP,Hostname
 - authorization-mode=Node,RBAC
 - etcd-servers
 - experimental-bootstrap-token-auth=true
 - service-cluster-ip-range=10.96.0.0/12
 - feature-gates

Kube-proxy

1. 启动方式 系统进程 或 DaemonSet

2. 配置文件

- kubernetes ds yaml

3. 主要参数

- kubeconfig
- masquerade-all
- feature-gates

feature-gates

```
Accelerators=true|false (ALPHA - default=false)
AdvancedAuditing=true|false (ALPHA - default=false)
AffinityInAnnotations=true|false (ALPHA - default=false)
AllAlpha=true|false (ALPHA - default=false)
AllowExtTrafficLocalEndpoints=true|false (default=true)
AppArmor=true|false (BETA - default=true)
DynamicKubeletConfig=true|false (ALPHA - default=false)
DynamicVolumeProvisioning=true|false (ALPHA - default=true)
ExperimentalCriticalPodAnnotation=true|false (ALPHA - default=false)
ExperimentalHostUserNamespaceDefaulting=true|false (BETA - default=false)
LocalStorageCapacityIsolation=true|false (ALPHA - default=false)
PersistentLocalVolumes=true|false (ALPHA - default=false)
RotateKubeletClientCertificate=true|false (ALPHA - default=false)
RotateKubeletServerCertificate=true|false (ALPHA - default=false)
StreamingProxyRedirects=true|false (BETA - default=true)
TaintBasedEvictions=true|false (ALPHA - default=false)
```

Kube-dns

1. 启动方式 Deployment
2. 配置文件
 - kubernetes deployment yaml
3. 主要参数(参看配置文件)

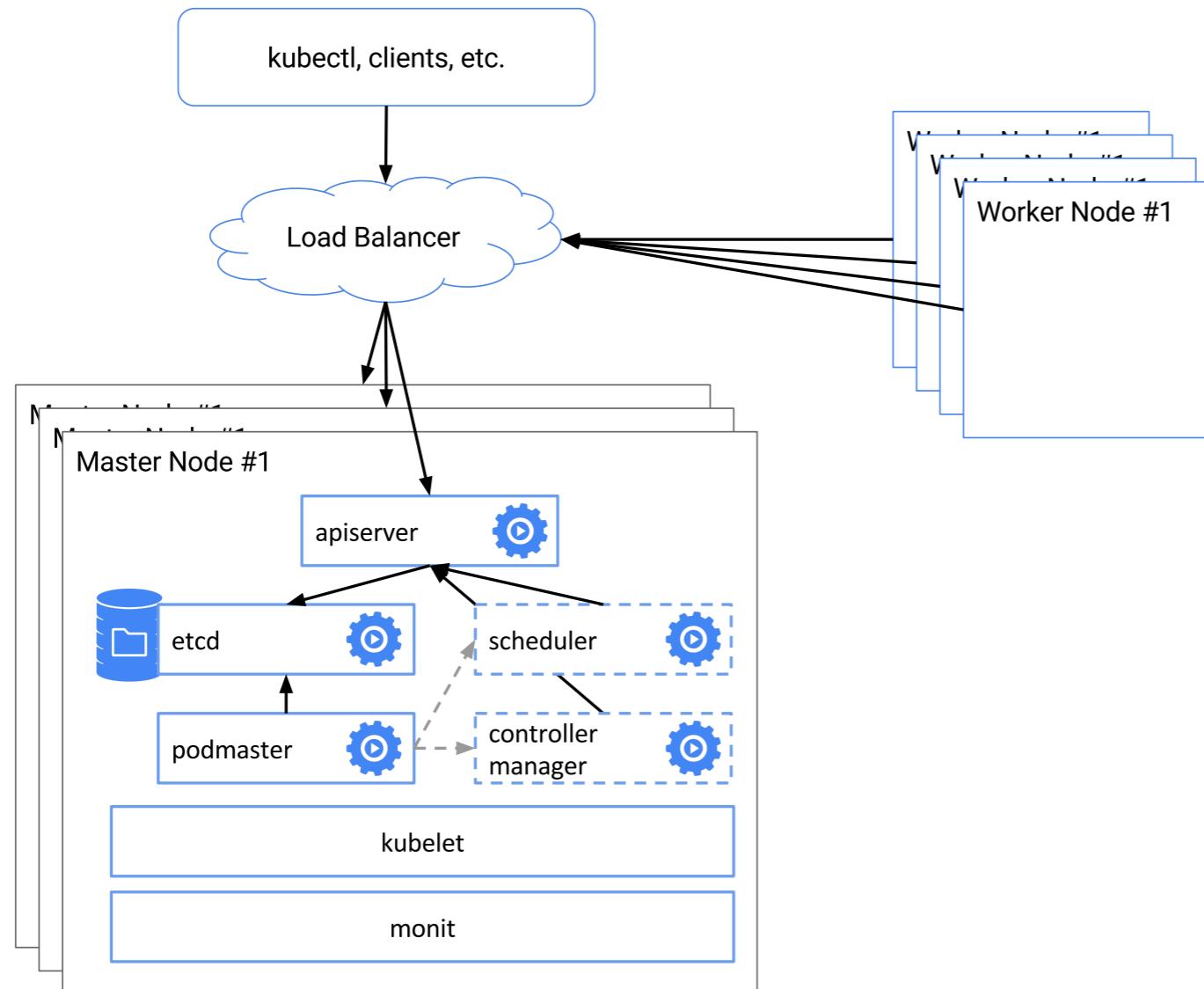
Etcd

1. 启动方式 StaticPod 或 外部集群
2. 配置文件
 - /etc/kubernetes/manifests/etcd.yaml
3. 主要参数(参看配置文件)

Kube-addon-manager

1. 启动方式 StaticPod
2. 作用 确保系统组件一直存在
3. 配置文件
 - <https://github.com/kubernetes/kubernetes/blob/master/cluster/saltbase/salt/kube-addons/kube-addon-manager.yaml>
 - <https://github.com/kubernetes/kubernetes/tree/master/cluster/addons/addon-manager>

Kubernetes 高可用



Kubernetes 高可用

1. Etcd -- Cluster
2. Apiserver -- LoadBalancer
3. kube-controller-manager kube-scheduler -- Master elected
 - <https://github.com/kubernetes/contrib/tree/master/election>
 - <http://jolestar.com/kubernetes-and-microservice/>

Kubernetes selfhosting

[kubeadm selfhosting.go 源码](#)

1. Load the Static Pod specification from disk (from /etc/kubernetes/manifests)
2. Extract the PodSpec from that Static Pod specification
3. Mutate the PodSpec to be compatible with self-hosting (add the right labels, taints, etc. so it can schedule correctly)
4. Build a new DaemonSet object for the self-hosted component in question. Use the above mentioned PodSpec
5. Create the DaemonSet resource. Wait until the Pods are running.
6. Remove the Static Pod manifest file. The kubelet will stop the original Static Pod-hosted component that was running.
7. The self-hosted containers should now step up and take over.
8. In order to avoid race conditions, we're still making sure the API /healthz endpoint is healthy
9. Do that for the kube-apiserver, kube-controller-manager and kube-scheduler in a loop

Kubernetes selfhosting

1. 更新 kubeadm

```
wget https://k8s-qingcloud.pek3a.qingstor.com/k8s%2Fv1.7.4%2Fbin%2Fkubeadm -O /usr/b
```

2. selfhosting

```
kubeadm alpha phase selfhosting  
kubectl get pods -n kube-system  
kubectl get ds -n kube-system  
kubeadm alpha phase mark-master $node  
kubectl get pods -n kube-system  
kubectl get ds -n kube-system
```

Kops 和 Minikube

1. Kops
2. Minikube <https://github.com/kubernetes/minikube>

Kubernetes 使用以及运维

1. kubectl
2. kubectl proxy
3. node 相关操作

kubectl

Basic Commands (Beginner):

create	Create a resource by filename or stdin
run	Run a particular image on the cluster
set	Set specific features on objects (image/resource/selector/subject)
get	Display one or many resources
edit	Edit a resource on the server
delete	Delete resources by filenames, stdin, resources and names, or by resour

Deploy Commands:

rollout	Manage the rollout of a resource
rollingupdate	Perform a rolling update of the given ReplicationController
scale	Set a new size for a Deployment, ReplicaSet, Replication Controller, or
resize	Set a new size for a Deployment, ReplicaSet, Replication Controller, or
autoscale	Auto-scale a Deployment, ReplicaSet, or ReplicationController

Cluster Management Commands:

certificate	Modify certificate resources.
clusterinfo	Display cluster info
top	Display Resource (CPU/Memory/Storage) usage.
cordon	Mark node as unschedulable
uncordon	Mark node as schedulable
drain	Drain node in preparation for maintenance
taint	Update the taints on one or more nodes

kubectl

Troubleshooting and Debugging Commands:

describe	Show details of a specific resource or group of resources
logs	Print the logs for a container in a pod
attach	Attach to a running container
exec	Execute a command in a container
port-forward	Forward one or more local ports to a pod
proxy	Run a proxy to the Kubernetes API server
cp	Copy files and directories to and from containers.
auth	Inspect authorization

Advanced Commands:

apply	Apply a configuration to a resource by filename or stdin
patch	Update field(s) of a resource using strategic merge patch
replace	Replace a resource by filename or stdin
update	Replace a resource by filename or stdin
convert	Convert config files between different API versions

Settings Commands:

label	Update the labels on a resource
annotate	Update the annotations on a resource

Kubectl proxy

安装 dashboard

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy
```

通过 kubectl proxy 在本地查看

```
kubectl proxy
```

维护或者删除节点

cordon	Mark node as unschedulable
uncordon	Mark node as schedulable
drain	Drain node in preparation for maintenance
taint	Update the taints on one or more nodes

总结

Kubernetes 安装本身并不复杂，但是

1. 镜像 (gcr.io/quay.io)
2. 网络
3. 安全
4. 扩展插件管理
5. 配置变更
6. 集群的伸缩
7. HA
8. 升级

作业

1. 手动通过 kubeadm 搭建一个 Kubernetes 集群，然后安装 kube-addon-manager，通过 kube-addon-manager 管理系统组件，比如 dashboard, kube-proxy, kubedns, heapster 等。
2. 在本地通过 minikube 搭建一个 Kubernetes 开发集群。
3. 通过云服务商部署一个托管 Kubernetes 集群，（推荐通过 QingCloud 青云的 appcenter 进行部署。<https://appcenter.qingcloud.com/apps/app-u0llx5j8>）研究 Kubernetes 的 CloudProvider 机制，以及网络和存储方案。

关于我

个人博客: <http://jolestar.com>

课程 Github: <https://github.com/jolestar/kubernetes-complete-course>



午夜咖啡

工具 • 架构 • 成长 • 思考

公众号: jolestar-blog

个人博客: <http://jolestar.com>

☞ 微信扫描关注 午夜咖啡

关注我们



QingCloud-IaaS



青云QingCloud

www.qingcloud.com

