**踏上kubernetes的第一步:**

1. **kuberntes集群环境部署**
   1. 背景介绍

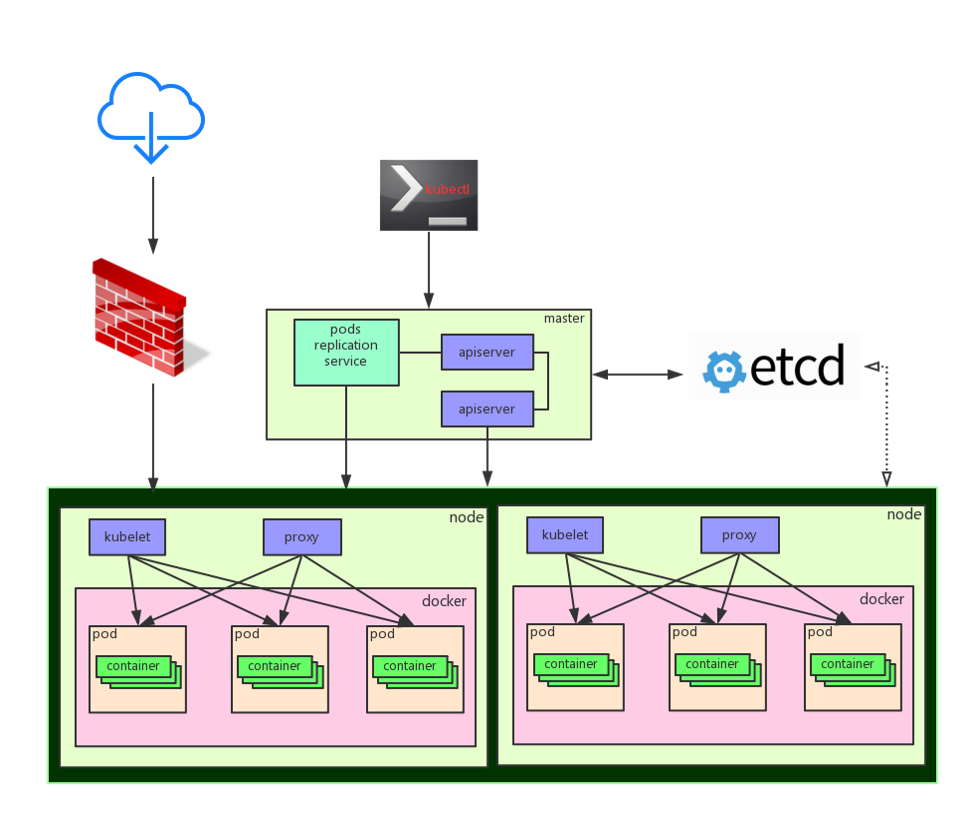
云计算飞速发展  
　　- IaaS  
　　- PaaS  
　　- SaaS  
Docker技术突飞猛进  
　　- 一次构建，到处运行  
　　- 容器的快速轻量  
　　- 完整的生态环境

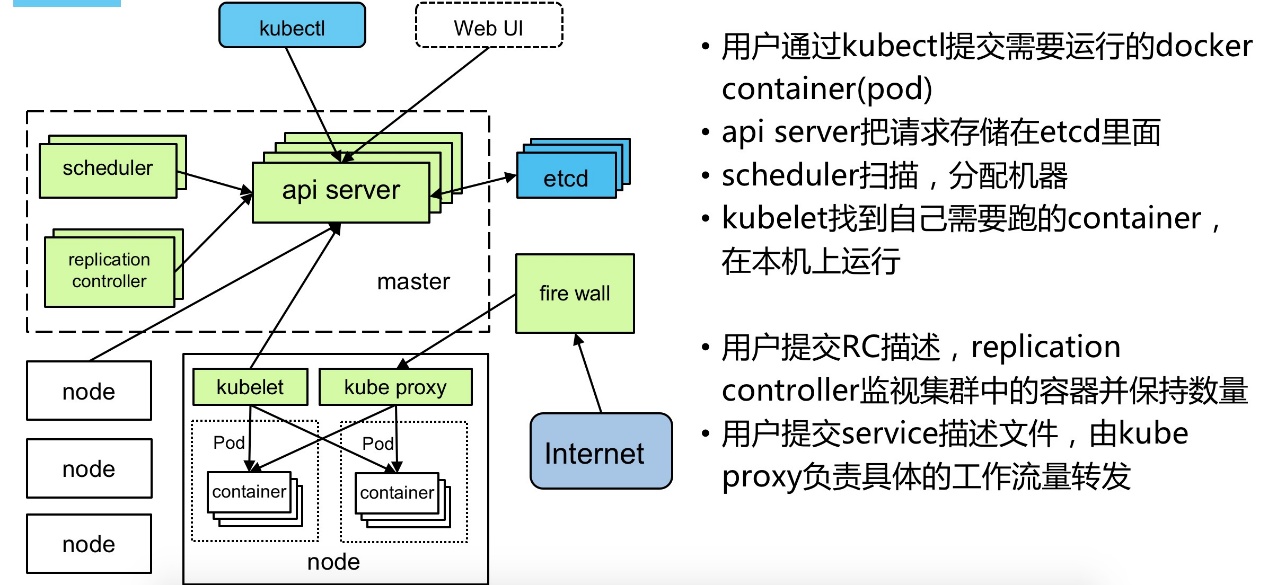
* 1. 什么是kubernetes

Kubernetes是Google开源的容器集群管理系统，实现基于Docker构建容器，利用Kubernetes能很方面管理多台Docker主机中的容器。

Kubernetes优势:  
　　- 容器编排  
　　- 轻量级  
　　- 开源  
　　- 弹性伸缩  
　　- 负载均衡

* 1. kubernetes 架构





* 1. kubernetes角色组成
     1. Pod

Pod是kubernetes的最小操作单元，一个Pod可以由一个或多个容器组成；

同一个Pod只能运行在同一个主机上，共享相同的volumes、network、namespace；

* + 1. ReplicationController（RC）

RC用来管理Pod，一个RC可以由一个或多个Pod组成，在RC被创建后，系统会根据定义好的副本数来创建Pod数量。在运行过程中，如果Pod数量小于定义的，就会重启停止的或重新分配Pod，反之则杀死多余的。当然，也可以动态伸缩运行的Pods规模或熟悉。

RC通过label关联对应的Pods，在滚动升级中，RC采用一个一个替换要更新的整个Pods中的Pod。

* + 1. Service

Service定义了一个Pod逻辑集合的抽象资源，Pod集合中的容器提供相同的功能。集合根据定义的Label和selector完成，当创建一个Service后，会分配一个Cluster IP，这个IP与定义的端口提供这个集合一个统一的访问接口，并且实现负载均衡。

* + 1. Label

Label是用于区分Pod、Service、RC的key/value键值对；

Pod、Service、RC可以有多个label，但是每个label的key只能对应一个；

主要是将Service的请求通过lable转发给后端提供服务的Pod集合；

* 1. kubernetes组件组成
     1. kubectl

客户端命令行工具，将接受的命令格式化后发送给kube-apiserver，作为整个系统的操作入口。kubelet，负责Pod的创建、销毁等，实现基本的集群管理的基本功能；每一个kubelet都会向Master注册自己，以便汇报自身的工作情况，包括Pod的cpu、内存等情况

* + 1. kube-apiserver

APIServer，是k8s资源操作的入口，这些操作包括对pod、service等对象的增删改查操作。在APIServer操作完成后，才会提交给etcd。

* + 1. kube-controller-manager

Controller Manager，是所有资源对象的自动化控制中心，包括replication controller、development controller、node controller等。

* + 1. kube-scheduler

负责节点资源管理，接受来自kube-apiserver创建Pods任务，并分配到某个节点。

* + 1. etcd

负责节点间的服务发现和配置共享。

* + 1. kube-proxy

运行在每个计算节点上，负责Pod网络代理。定时从etcd获取到service信息来做相应的策略。

* + 1. kubelet

运行在每个计算节点上，作为agent，接受分配该节点的Pods任务及管理容器，周期性获取容器状态，反馈给kube-apiserver。

* + 1. DNS

一个可选的DNS服务，用于为每个Service对象创建DNS记录，这样所有的Pod就可以通过DNS访问服务了。

1. **架构部署**
   1. **基本信息**

#服务器分配

10.44.25.132 kubernetes etcd docker flannel ---master

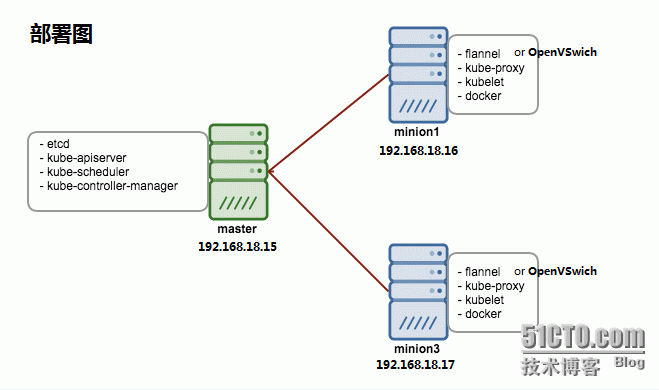
10.45.40.247 kubernetes  docker flannel ---node1

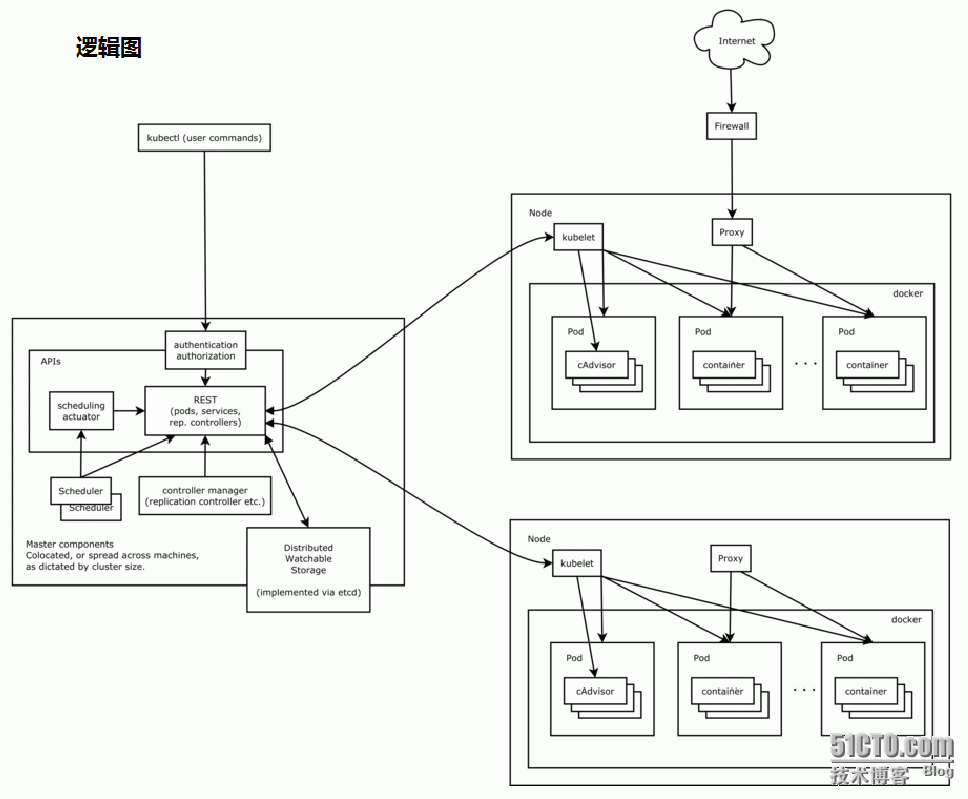
10.44.155.223 kubernetes  docker flannel ---node2

#版本选择

Centos7 yum源默认

* 1. **部署图**





* 1. **组件安装**

Master节点：

systemctl stop firewalld && sudo systemctl disable firewalld

yum install -y kubernetes etcd docker flannel     ---master同时作为node

Node节点：

systemctl stop firewalld && sudo systemctl disable firewalld

yum install -y kubernetes  docker flannel

* 1. **master配置**
     1. **etcd配置**

[root@iZ258kmltygZ k8s]# egrep -v '^#|^$' /etc/etcd/etcd.conf

ETCD\_NAME=default

ETCD\_DATA\_DIR="/var/lib/etcd/default.etcd"

ETCD\_LISTEN\_CLIENT\_URLS="http://0.0.0.0:2379"

ETCD\_ADVERTISE\_CLIENT\_URLS=<http://localhost:2379>

执行命令：etcdctl set /atomic.io/network/config '{"Network": "10.0.0.0/16","SubnetLen": 24,"SubnetMin": "10.0.1.0","SubnetMax": "10.0.200.0","Backend": {"Type":"vxlan"}}' --- 路径是否是atomic.io查看flanneld文件

* + 1. **apiserver配置**

[root@iZ258kmltygZ k8s]# egrep -v '^#|^$' /etc/kubernetes/apiserver

KUBE\_API\_ADDRESS="--insecure-bind-address=0.0.0.0"

KUBE\_API\_PORT="--port=8080"

KUBELET\_PORT="--kubelet-port=10250"

KUBE\_ETCD\_SERVERS="--etcd-servers=http://10.44.25.132:2379"

KUBE\_SERVICE\_ADDRESSES="--service-cluster-ip-range=10.254.0.0/16"

KUBE\_ADMISSION\_CONTROL="--admission-control=NamespaceLifecycle,NamespaceExists,LimitRanger,ResourceQuota"

KUBE\_API\_ARGS=""

* + 1. **kubelet配置**

[root@iZ258kmltygZ k8s]# egrep -v '^#|^$' /etc/kubernetes/kubelet

KUBELET\_ADDRESS="--address=0.0.0.0"

KUBELET\_PORT="--port=10250"

KUBELET\_HOSTNAME="--hostname-override=10.44.25.132"

KUBELET\_API\_SERVER="--api-servers=http://10.44.25.132:8080"

KUBELET\_POD\_INFRA\_CONTAINER="--pod-infra-container-image=registry.access.redhat.com/rhel7/pod-infrastructure:latest"

KUBELET\_ARGS=""

* + 1. **config配置**

[root@iZ258kmltygZ k8s]# egrep -v '^#|^$' /etc/kubernetes/config

KUBE\_LOGTOSTDERR="--logtostderr=true"

KUBE\_LOG\_LEVEL="--v=0"

KUBE\_ALLOW\_PRIV="--allow-privileged=false"

KUBE\_MASTER="--master=http://10.44.25.132:8080"

* + 1. **flannel配置**

[root@iZ258kmltygZ k8s]# egrep -v '^#|^$' /etc/sysconfig/flanneld

FLANNEL\_ETCD\_ENDPOINTS="http://10.44.25.132:2379"

FLANNEL\_ETCD\_PREFIX="/atomic.io/network"

FLANNEL\_OPTIONS="-ip-masq=true"

* + 1. **master启动**

for SERVICES in etcd kube-apiserver kube-controller-manager kube-scheduler   kube-proxy  kubelet docker flanneld   ; do systemctl restart   $SERVICES; systemctl enable $SERVICES; systemctl status $SERVICES; done;

* 1. **node1/2配置**
     1. **kubelet配置**

[rd@iZ2557l9sefZ ~]$ egrep -v '^#|^$' /etc/kubernetes/kubelet

KUBELET\_ADDRESS="--address=0.0.0.0"

KUBELET\_PORT="--port=10250"

KUBELET\_HOSTNAME="--hostname-override=10.45.40.247"

KUBELET\_API\_SERVER="--api-servers=http://10.44.25.132:8080"

KUBELET\_POD\_INFRA\_CONTAINER="--pod-infra-container-image=registry.access.redhat.com/rhel7/pod-infrastructure:latest"

KUBELET\_ARGS="--cluster-dns=10.10.0.10 --cluster-domain=cluster.local"

* + 1. **config配置**

[rd@iZ2557l9sefZ ~]$ egrep -v '^#|^$' /etc/kubernetes/config

KUBE\_LOGTOSTDERR="--logtostderr=true"

KUBE\_LOG\_LEVEL="--v=0"

KUBE\_ALLOW\_PRIV="--allow-privileged=false"

KUBE\_MASTER="--master=http://10.44.25.132:8080"

* + 1. **flannel配置**

[rd@iZ2557l9sefZ ~]$ egrep -v '^#|^$' /etc/sysconfig/flanneld

FLANNEL\_ETCD\_ENDPOINTS="http://10.44.25.132:2379"

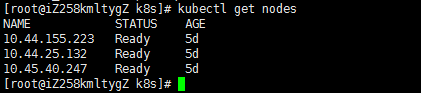
FLANNEL\_ETCD\_PREFIX="/atomic.io/network"

FLANNEL\_OPTIONS="-ip-masq=true"

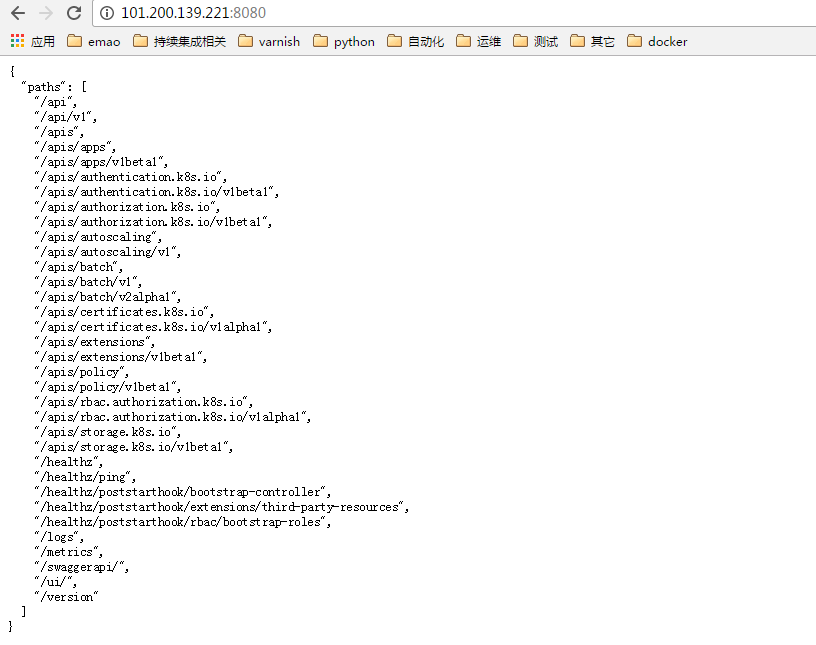
* + 1. **node启动**

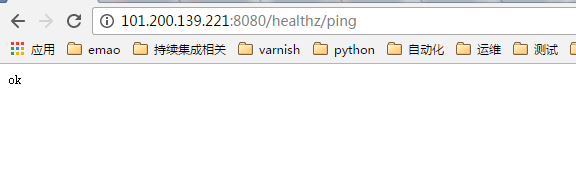
for SERVICES in kube-proxy kubelet docker flanneld; do  
        systemctl restart $SERVICES  
        systemctl enable $SERVICES  
        systemctl status $SERVICES  
    done;

* 1. **验证查看**
     1. **查看所有node是否正常**



* + 1. **查看请求url及健康状况**





1. **kube-ui 安装**

kubernetes 自带了一个kube-ui,用来展示kubernetes 集群状态,目前kube-ui 还比较简单,只展现了kube 节点资源使用率,rc,service,pod ,nodes 情况,默认情况下安装kubernetes 集群是没有安装kube-ui的,需要我们手工安装

* 1. **下载kubernetes-dashboard.yaml**

wget  <https://rawgit.com/kubernetes/dashboard/master/src/deploy/kubernetes-dashboard.yaml>

* 1. **修改kubernetes-dashboard.yaml**

spec:

containers:

- name: kubernetes-dashboard

image: gcr.io/google\_containers/kubernetes-dashboard-amd64:v1.6.0

imagePullPolicy: Always

ports:

- containerPort: 9090

protocol: TCP

args:

# Uncomment the following line to manually specify Kubernetes API server Host

# If not specified, Dashboard will attempt to auto discover the API server and connect

# to it. Uncomment only if the default does not work.

- --apiserver-host=http://10.44.25.132:8080 ---此处修改为master地址

livenessProbe:

httpGet:

path: /

port: 9090

initialDelaySeconds: 30

timeoutSeconds: 30

kubectl create -f kubernetes-dashboard.yaml ---master上执行

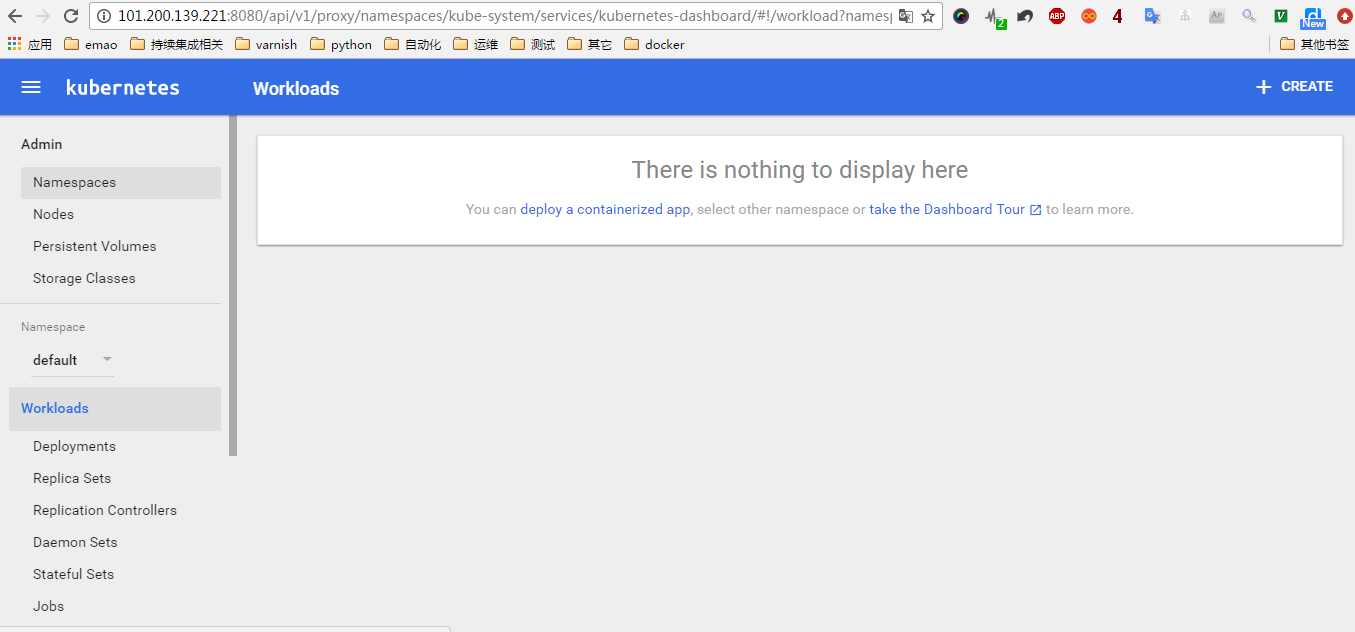
gcr.io被墙掉了需要修改hosts文件

61.91.161.217 gcr.io

61.91.161.217 www.gcr.io

61.91.161.217 packages.cloud.google.com

如果以上地址失效请前往<https://github.com/racaljk/hosts>查找



1. **总结**

前面的配置,基本绕过了网上的一些构建k8s集群不完整文档的坑,主要有如下几个坑

* 1. 问题1:

Unable to listen for secure (open /var/run/kubernetes/apiserver.crt: no such file or directory

解决:

vi /etc/systemd/system/kube-apiserver.service.d/10-varrun-build.conf

[Service]

# Run ExecStartPre with root-permissions

PermissionsStartOnly=true

ExecStartPre=-/usr/bin/mkdir /var/run/kubernetes

ExecStartPre=/usr/bin/chown -R kube:kube /var/run/kubernetes/

* 1. 问题2:

no API token found for service account default/default, retry after the token is automatically created and added to the service account

解决:

1. openssl genrsa -out /tmp/serviceaccount.key 2048

2.

vim /etc/kubernetes/apiserver:

KUBE\_API\_ARGS="--service\_account\_key\_file=/tmp/serviceaccount.key"

3.

vim /etc/kubernetes/controller-manager

KUBE\_CONTROLLER\_MANAGER\_ARGS="--service\_account\_private\_key\_file=/tmp/serviceaccount.key"

systemctl restart kube-controller-manager.service

* 1. 问题3：

The connection to the server localhost:8080 was refused - did you specify the right host or port?

解决方法

export KUBERNETES\_MASTER=http://10.44.25.132:8080 --10.44.25.132为masterip地址

启动service后不知道分配到哪个node上，需要确定每个node上网络，证书都是可用的

1. **参考**

[高可用及自动发现的Docker基础架构](http://example.net/)  
[CentOS 7实战Kubernetes部署](http://www.infoq.com/cn/articles/centos7-practical-kubernetes-deployment/)  
[kubernetes 官方doc](http://kubernetes.io/v1.0/docs/user-guide/ui.html)