# kubbernets安装前准备

## **概述**

本次安装采用 Ubuntu Server X64 18.04 LTS 版本安装 kubernetes 集群环境，集群节点为 1 主 2 从模式，此次对虚拟机会有些基本要求，如下：

* OS：Ubuntu Server X64 18.04 LTS（16.04 版本步骤相同，再之前则不同）
* CPU：最低要求，1 CPU 2 核
* 内存：最低要求，2 GB
* 磁盘：最低要求，20 GB

## **统一环境配置**

****注意：**** 以下步骤请在制作 VMware 镜像时一并完成，避免逐台安装的痛苦

### **关闭交换空间**

1. swapoff -a

### **避免开机启动交换空间**

1. # 注释 swap 开头的行
2. vi /etc/fstab

### **关闭防火墙**

1. ufw disable

### **配置 DNS**

1. # 取消 DNS 行注释，并增加 DNS 配置如：114.114.114.114，修改后重启下计算机
2. vi /etc/systemd/resolved.conf

### **安装 Docker**

1. # 更新软件源
2. sudo apt-get update
3. # 安装所需依赖
4. sudo apt-get -y install apt-transport-https ca-certificates curl software-properties-common
5. # 安装 GPG 证书
6. curl -fsSL http://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo apt-key add -
7. # 新增软件源信息
8. sudo add-apt-repository "deb [arch=amd64] http://mirrors.aliyun.com/docker-ce/linux/ubuntu $(lsb\_release -cs) stable"
9. # 再次更新软件源
10. sudo apt-get -y update
11. # 安装 Docker CE 版
12. sudo apt-get -y install docker-ce
13. # 开启 Docker Service
14. systemctl enable docker.service

### **配置 Docker 加速器**

****注意：**** 国内镜像加速器可能会很卡，请替换成你自己阿里云镜像加速器，地址如：https://yourself.mirror.aliyuncs.com，在阿里云控制台的 ****容器镜像服务 -> 镜像加速器**** 菜单中可以找到

在 /etc/docker/daemon.json 中写入如下内容（以下配置修改 cgroup 驱动为 systemd，满足 K8S 建议）

1. {
2. "exec-opts": ["native.cgroupdriver=systemd"],
3. "log-driver": "json-file",
4. "log-opts": {
5. "max-size": "100m"
6. },
7. "registry-mirrors": [
8. "https://k7da99jp.mirror.aliyuncs.com/",
9. "https://dockerhub.azk8s.cn",
10. "https://registry.docker-cn.com"
11. ],
12. "storage-driver": "overlay2"
13. }

### **安装 Kubernetes 必备工具**

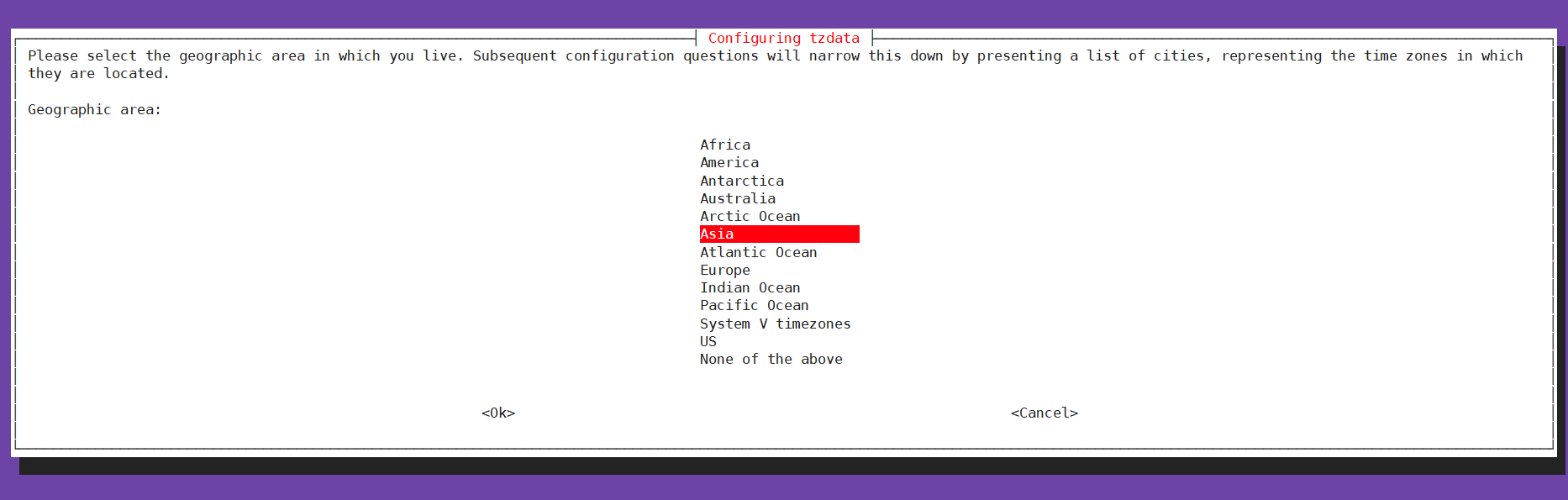
安装三个 Kubernetes 必备工具，分别为 ****kubeadm****，****kubelet****，****kubectl****

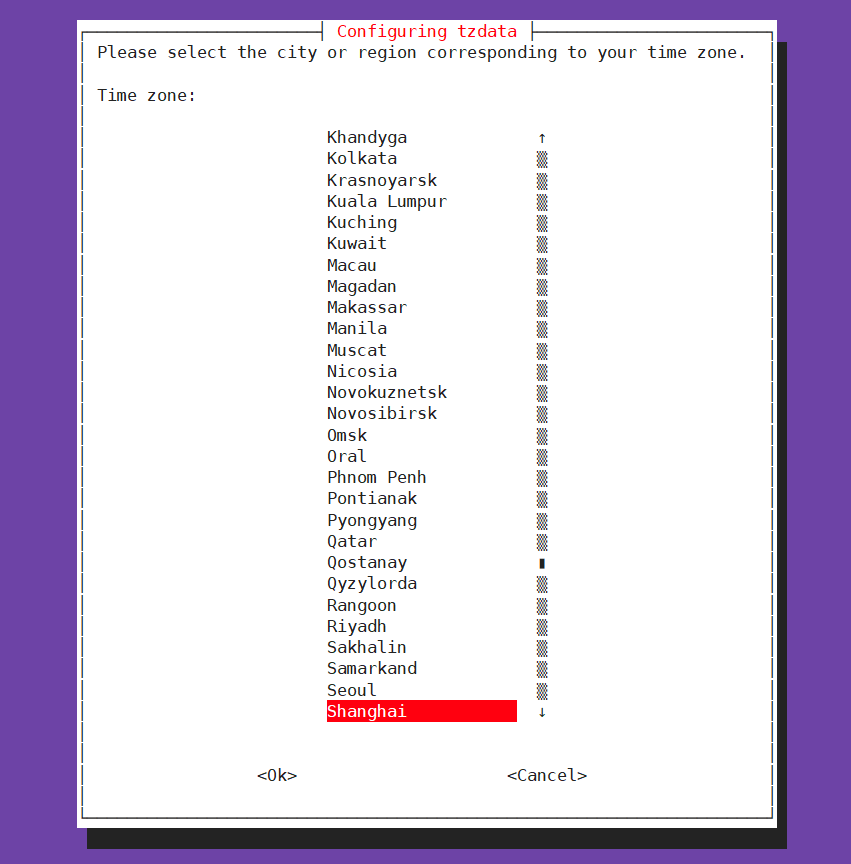
1. # 安装系统工具
2. apt-get update && apt-get install -y apt-transport-https
3. # 安装 GPG 证书
4. curl https://mirrors.aliyun.com/kubernetes/apt/doc/apt-key.gpg | apt-key add -
5. # 写入软件源；注意：我们用系统代号为 bionic，但目前阿里云不支持，所以沿用 16.04 的 xenial
6. cat << EOF >/etc/apt/sources.list.d/kubernetes.list
7. deb https://mirrors.aliyun.com/kubernetes/apt/ kubernetes-xenial main
8. EOF
9. # 安装
10. apt-get update && apt-get install -y kubelet kubeadm kubectl

### **同步时间**

* ****设置时区****

1. dpkg-reconfigure tzdata

* 选择 ****Asia（亚洲）****
* 选择 ****Shanghai（上海）****



* ****时间同步****

1. # 安装 ntpdate
2. apt-get install ntpdate
3. # 设置系统时间与网络时间同步（cn.pool.ntp.org 位于中国的公共 NTP 服务器）
4. ntpdate cn.pool.ntp.org
5. # 将系统时间写入硬件时间
6. hwclock --systohc

* ****确认时间****

1. date
2. # 输出如下（自行对照与系统时间是否一致）
3. Sun Jun 2 22:02:35 CST 2019

### **修改 cloud.cfg**

主要作用是防止重启后主机名还原

1. vi /etc/cloud/cloud.cfg
2. # 该配置默认为 false，修改为 true 即可
3. preserve\_hostname: true

## **单独节点配置**

****注意：**** 为 Master 和 Node 节点单独配置对应的 ****IP**** 和 ****主机名****

### **配置 IP**

编辑 vi /etc/netplan/50-cloud-init.yaml 配置文件，修改内容如下

1. network:
2. ethernets:
3. ens33:
4. addresses: [192.168.141.110/24]
5. gateway4: 192.168.141.2
6. nameservers:
7. addresses: [192.168.141.2]
8. version: 2

使用 netplan apply 命令让配置生效

### **配置主机名**

1. # 修改主机名
2. hostnamectl set-hostname kubernetes-master
3. # 配置 hosts
4. cat >> /etc/hosts << EOF
5. 192.168.141.110 kubernetes-master
6. EOF

## **安装过程出现的问题：**

李哥，问个问题。《微服务架构2.0》中“kubernetes-安装前的准备”这一节，最后的“配置IP”那里，最开始vmware安装ubuntu时候，虚拟机的网络选“桥接”还是”nat“？我选的是”桥接“，默认装好后是DHCP自动分配IP的。改成分配静态IP的时候，"network.ethernets.ens33.gateway4"这一项配置要怎么确定？

kelchoi7个月前 (09-20)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e5%89%8d%e7%9a%84%e5%87%86%e5%a4%87/javascript:;)

1. #2

IMG_256

李哥这块ip和主机名配完之后重启查不到ip了

scorpio996个月前 (10-16)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e5%89%8d%e7%9a%84%e5%87%86%e5%a4%87/javascript:;)

IMG_257

我这也是，改了ip后就找不到ip了。。。。

lovelifest6个月前 (10-22)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e5%89%8d%e7%9a%84%e5%87%86%e5%a4%87/javascript:;)

IMG_258

网卡名要对应上，我也才发现

lovelifest6个月前 (10-22)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e5%89%8d%e7%9a%84%e5%87%86%e5%a4%87/javascript:;)

1. #3

IMG_259

这块改完ip输入netplan apply后就连接失败了

scorpio996个月前 (10-17)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e5%89%8d%e7%9a%84%e5%87%86%e5%a4%87/javascript:;)

IMG_260

我也是我也是，不知道咋解决，磨了好久了

ycj1236个月前 (10-28)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e5%89%8d%e7%9a%84%e5%87%86%e5%a4%87/javascript:;)

IMG_261

50-cloud-init.yaml配置文件中ip的网段要改成你自己虚拟机的，改完配置后直接配置主机，然后reboot重启就好了，不用netplan apply命令

scorpio996个月前 (10-28)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e5%89%8d%e7%9a%84%e5%87%86%e5%a4%87/javascript:;)

1. #4

IMG_262

三、修改为固定ip 编辑 vi /etc/netplan/50-cloud-init.yaml 配置文件，修改内容如下 network: ethernets: ens33: dhcp4: true addresses: [192.168.141.130/24] gateway4: 192.168.141.2 nameservers: addresses: [192.168.141.2] version: 2 使用 netplan apply 命令让配置生效 注：Linux Xshell连接Linux服务器时报错Socket error Event: 32 Error: 10053 问题描述 在用Xshell连接Linux服务器时，出现错误提示"Socket error Event: 32 Error: 10053. Connection closing...Socket close." 分析处理 输入命令"sudo sshd -t"进行sshd服务的检测，如果有输出内容，则很可能是该内容中的相关文件无权限，修改其权限：sudo chmod 600 xxx; 如果无任何输出内容，则很可能是IP冲突，修改Linux的IP。 最后 输入命令"sudo sshd -t"，无任何提示后，再Xshell重新连接。

Alibaba6个月前 (10-29)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e5%89%8d%e7%9a%84%e5%87%86%e5%a4%87/javascript:;)

1. #5

IMG_263

李哥， 可以用阿里云的3台学生服务器搭建k8s集群吗

linzi4个月前 (12-13)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e5%89%8d%e7%9a%84%e5%87%86%e5%a4%87/javascript:;)

IMG_264

怎么可能能启动呢 2核4g内存呢 配置不够报各种各样的问题

123\_boot3个月前 (01-08)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e5%89%8d%e7%9a%84%e5%87%86%e5%a4%87/javascript:;)

1. #6

IMG_265

总结一下： 1. gateway4 网关地址是VM那个NAT网络连接方式的网卡地址，不是你主机的那个VM网卡地址 2. 把DNS改为网关，使得虚拟机可以正常解析域名 3. 配置好docker后，拉取busybox镜像进行测试

lishubin4个月前 (12-16)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e5%89%8d%e7%9a%84%e5%87%86%e5%a4%87/javascript:;)

1. #7

IMG_266

k8s今天用的时候，执行“kubectl get pod”报The connection to the server 192.168.181.110:6443 was refused - did you specify the right host or port?，主节点挂掉了吗，试了好多办法都不行，怎么解决？

mogo3个月前 (01-18)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e5%89%8d%e7%9a%84%e5%87%86%e5%a4%87/javascript:;)

# kubernets安装集群

## **概述**

kubeadm 是 kubernetes 的集群安装工具，能够快速安装 kubernetes 集群，安装 kubernetes 主要是安装它的各个镜像，而 kubeadm 已经为我们集成好了运行 kubernetes 所需的基本镜像。但由于国内的网络原因，在搭建环境时，无法拉取到这些镜像。此时我们只需要修改为阿里云提供的镜像服务即可解决该问题。

## **创建并修改配置**

1. # 导出配置文件
2. kubeadm config print init-defaults --kubeconfig ClusterConfiguration > kubeadm.yml
3. apiVersion: kubeadm.k8s.io/v1beta2
4. bootstrapTokens:
5. - groups:
6. - system:bootstrappers:kubeadm:default-node-token
7. token: abcdef.0123456789abcdef
8. ttl: 24h0m0s
9. usages:
10. - signing
11. - authentication
12. kind: InitConfiguration
13. localAPIEndpoint:
14. # 修改为主节点 IP
15. advertiseAddress: 192.168.141.110
16. bindPort: 6443
17. nodeRegistration:
18. criSocket: /var/run/dockershim.sock
19. name: kubernetes-master
20. taints:
21. - effect: NoSchedule
22. key: node-role.kubernetes.io/master
23. ---
24. apiServer:
25. timeoutForControlPlane: 4m0s
26. apiVersion: kubeadm.k8s.io/v1beta2
27. certificatesDir: /etc/kubernetes/pki
28. clusterName: kubernetes
29. controllerManager: {}
30. dns:
31. type: CoreDNS
32. etcd:
33. local:
34. dataDir: /var/lib/etcd
35. # 国内不能访问 Google，修改为阿里云
36. imageRepository: registry.aliyuncs.com/google\_containers
37. kind: ClusterConfiguration
38. # 修改版本号
39. kubernetesVersion: v1.15.0
40. networking:
41. dnsDomain: cluster.local
42. # 配置 POD 所在网段为我们虚拟机不重叠的网段（这里用的是 Flannel 默认网段）
43. podSubnet: "10.244.0.0/16"
44. serviceSubnet: 10.96.0.0/12
45. scheduler: {}

## **查看所需镜像**

1. kubeadm config images list --config kubeadm.yml
2. # 输出如下
3. registry.aliyuncs.com/google\_containers/kube-apiserver:v1.15.0
4. registry.aliyuncs.com/google\_containers/kube-controller-manager:v1.15.0
5. registry.aliyuncs.com/google\_containers/kube-scheduler:v1.15.0
6. registry.aliyuncs.com/google\_containers/kube-proxy:v1.15.0
7. registry.aliyuncs.com/google\_containers/pause:3.1
8. registry.aliyuncs.com/google\_containers/etcd:3.3.10
9. registry.aliyuncs.com/google\_containers/coredns:1.3.1

## **拉取所需镜像**

1. kubeadm config images pull --config kubeadm.yml
2. # 输出如下
3. [config/images] Pulled registry.aliyuncs.com/google\_containers/kube-apiserver:v1.15.0
4. [config/images] Pulled registry.aliyuncs.com/google\_containers/kube-controller-manager:v1.15.0
5. [config/images] Pulled registry.aliyuncs.com/google\_containers/kube-scheduler:v1.15.0
6. [config/images] Pulled registry.aliyuncs.com/google\_containers/kube-proxy:v1.15.0
7. [config/images] Pulled registry.aliyuncs.com/google\_containers/pause:3.1
8. [config/images] Pulled registry.aliyuncs.com/google\_containers/etcd:3.3.10
9. [config/images] Pulled registry.aliyuncs.com/google\_containers/coredns:1.3.1

## **安装主节点**

执行以下命令初始化主节点，该命令指定了初始化时需要使用的配置文件，其中添加 --upload-certs 参数可以在后续执行加入节点时自动分发证书文件。追加的 tee kubeadm-init.log 用以输出日志。

****注意：**** 如果安装 kubernetes 版本和下载的镜像版本不统一则会出现 timed out waiting for the condition 错误。中途失败或是想修改配置可以使用 kubeadm reset 命令重置配置，再做初始化操作即可。

1. kubeadm init --config=kubeadm.yml --upload-certs | tee kubeadm-init.log
2. # 输出如下
3. Flag --experimental-upload-certs has been deprecated, use --upload-certs instead
4. [init] Using Kubernetes version: v1.15.0
5. [preflight] Running pre-flight checks
6. [WARNING IsDockerSystemdCheck]: detected "cgroupfs" as the Docker cgroup driver. The recommended driver is "systemd". Please follow the guide at https://kubernetes.io/docs/setup/cri/
7. [preflight] Pulling images required for setting up a Kubernetes cluster
8. [preflight] This might take a minute or two, depending on the speed of your internet connection
9. [preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
10. [kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
11. [kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
12. [kubelet-start] Activating the kubelet service
13. [certs] Using certificateDir folder "/etc/kubernetes/pki"
14. [certs] Generating "ca" certificate and key
15. [certs] Generating "apiserver-kubelet-client" certificate and key
16. [certs] Generating "apiserver" certificate and key
17. [certs] apiserver serving cert is signed for DNS names [kubernetes-master kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs [10.96.0.1 192.168.141.110]
18. [certs] Generating "front-proxy-ca" certificate and key
19. [certs] Generating "front-proxy-client" certificate and key
20. [certs] Generating "etcd/ca" certificate and key
21. [certs] Generating "etcd/server" certificate and key
22. [certs] etcd/server serving cert is signed for DNS names [kubernetes-master localhost] and IPs [192.168.141.110 127.0.0.1 ::1]
23. [certs] Generating "apiserver-etcd-client" certificate and key
24. [certs] Generating "etcd/peer" certificate and key
25. [certs] etcd/peer serving cert is signed for DNS names [kubernetes-master localhost] and IPs [192.168.141.110 127.0.0.1 ::1]
26. [certs] Generating "etcd/healthcheck-client" certificate and key
27. [certs] Generating "sa" key and public key
28. [kubeconfig] Using kubeconfig folder "/etc/kubernetes"
29. [kubeconfig] Writing "admin.conf" kubeconfig file
30. [kubeconfig] Writing "kubelet.conf" kubeconfig file
31. [kubeconfig] Writing "controller-manager.conf" kubeconfig file
32. [kubeconfig] Writing "scheduler.conf" kubeconfig file
33. [control-plane] Using manifest folder "/etc/kubernetes/manifests"
34. [control-plane] Creating static Pod manifest for "kube-apiserver"
35. [control-plane] Creating static Pod manifest for "kube-controller-manager"
36. [control-plane] Creating static Pod manifest for "kube-scheduler"
37. [etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
38. [wait-control-plane] Waiting for the kubelet to boot up the control plane as static Pods from directory "/etc/kubernetes/manifests". This can take up to 4m0s
39. [apiclient] All control plane components are healthy after 23.006217 seconds
40. [upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
41. [kubelet] Creating a ConfigMap "kubelet-config-1.15" in namespace kube-system with the configuration for the kubelets in the cluster
42. [upload-certs] Storing the certificates in Secret "kubeadm-certs" in the "kube-system" Namespace
43. [upload-certs] Using certificate key:
44. 4a351218cf34c15b04b08b7d916c27a625418145d8d3b281ec125fd2b2ac7ab5
45. [mark-control-plane] Marking the node kubernetes-master as control-plane by adding the label "node-role.kubernetes.io/master=''"
46. [mark-control-plane] Marking the node kubernetes-master as control-plane by adding the taints [node-role.kubernetes.io/master:NoSchedule]
47. [bootstrap-token] Using token: abcdef.0123456789abcdef
48. [bootstrap-token] Configuring bootstrap tokens, cluster-info ConfigMap, RBAC Roles
49. [bootstrap-token] configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credentials
50. [bootstrap-token] configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
51. [bootstrap-token] configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
52. [bootstrap-token] Creating the "cluster-info" ConfigMap in the "kube-public" namespace
53. [addons] Applied essential addon: CoreDNS
54. [addons] Applied essential addon: kube-proxy
55. Your Kubernetes control-plane has initialized successfully!
56. To start using your cluster, you need to run the following as a regular user:
57. mkdir -p $HOME/.kube
58. sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
59. sudo chown $(id -u):$(id -g) $HOME/.kube/config
60. You should now deploy a pod network to the cluster.
61. Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
62. https://kubernetes.io/docs/concepts/cluster-administration/addons/
63. Then you can join any number of worker nodes by running the following on each as root:
64. kubeadm join 192.168.141.110:6443 --token abcdef.0123456789abcdef \
65. --discovery-token-ca-cert-hash sha256:f0759e0d352c1a5de4444782b4a676460b2ea7a2876fa0accab572b8629b72c8

### **配置 kubectl**

1. mkdir -p $HOME/.kube
2. cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
3. # 非 ROOT 用户执行
4. chown $(id -u):$(id -g) $HOME/.kube/config

### **验证是否成功**

1. kubectl get node
2. # 输出如下
3. NAME STATUS ROLES AGE VERSION
4. kubernetes-master NotReady master 4m38s v1.15.0

## **安装从节点**

将 Node 节点加入到集群中很简单，只需要在 Node 服务器上安装 ****kubeadm****，****kubectl****，****kubelet**** 三个工具，然后使用 kubeadm join 命令加入即可

1. kubeadm join 192.168.141.110:6443 --token abcdef.0123456789abcdef \
2. --discovery-token-ca-cert-hash sha256:f0759e0d352c1a5de4444782b4a676460b2ea7a2876fa0accab572b8629b72c8
3. # 输出如下
4. [preflight] Running pre-flight checks
5. [WARNING IsDockerSystemdCheck]: detected "cgroupfs" as the Docker cgroup driver. The recommended driver is "systemd". Please follow the guide at https://kubernetes.io/docs/setup/cri/
6. [preflight] Reading configuration from the cluster...
7. [preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -oyaml'
8. [kubelet-start] Downloading configuration for the kubelet from the "kubelet-config-1.15" ConfigMap in the kube-system namespace
9. [kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
10. [kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
11. [kubelet-start] Activating the kubelet service
12. [kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...
13. This node has joined the cluster:
14. \* Certificate signing request was sent to apiserver and a response was received.
15. \* The Kubelet was informed of the new secure connection details.
16. Run 'kubectl get nodes' on the control-plane to see this node join the cluster.

### **验证是否成功**

回到 Master 主节点查看是否安装成功

****注意：**** 如果 Node 节点加入 Master 时配置有问题可以在 Node 节点上使用 kubeadm reset 重置配置再使用 kubeadm join 命令重新加入即可。希望在 master 节点删除 node ，可以使用 kubectl delete nodes <NAME> 删除。

1. kubectl get node
2. # 输出如下
3. NAME STATUS ROLES AGE VERSION
4. kubernetes-master NotReady master 20m v1.15.0
5. kubernetes-node-01 NotReady <none> 16s v1.15.0
6. kubernetes-node-02 NotReady <none> 6s v1.15.0

### **查看 Pod 状态**

coredns 尚未运行，此时我们还需要安装网络插件

1. kubectl get pod -n kube-system -o wide
2. # 输出如下
3. NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
4. coredns-bccdc95cf-9s4bm 0/1 Pending 0 24m <none> <none> <none> <none>
5. coredns-bccdc95cf-s8ggd 0/1 Pending 0 24m <none> <none> <none> <none>
6. etcd-kubernetes-master 1/1 Running 0 24m 192.168.141.110 kubernetes-master <none> <none>
7. kube-apiserver-kubernetes-master 1/1 Running 0 24m 192.168.141.110 kubernetes-master <none> <none>
8. kube-controller-manager-kubernetes-master 1/1 Running 0 23m 192.168.141.110 kubernetes-master <none> <none>
9. kube-proxy-8s87d 1/1 Running 0 4m56s 192.168.141.120 kubernetes-node-01 <none> <none>
10. kube-proxy-cbnlb 1/1 Running 0 4m46s 192.168.141.121 kubernetes-node-02 <none> <none>
11. kube-proxy-vwhxj 1/1 Running 0 24m 192.168.141.110 kubernetes-master <none> <none>
12. kube-scheduler-kubernetes-master 1/1 Running 0 24m 192.168.141.110 kubernetes-master <none> <none>

## **附：扩展阅读**

### **kubeadm init 的执行过程**

* ****init：**** 指定版本进行初始化操作
* ****preflight：**** 初始化前的检查和下载所需要的 Docker 镜像文件
* ****kubelet-start：**** 生成 kubelet 的配置文件 var/lib/kubelet/config.yaml，没有这个文件 kubelet 无法启动，所以初始化之前的 kubelet 实际上启动不会成功
* ****certificates：**** 生成 Kubernetes 使用的证书，存放在 /etc/kubernetes/pki 目录中
* ****kubeconfig：**** 生成 KubeConfig 文件，存放在 /etc/kubernetes 目录中，组件之间通信需要使用对应文件
* ****control-plane：**** 使用 /etc/kubernetes/manifest 目录下的 YAML 文件，安装 Master 组件
* ****etcd：**** 使用 /etc/kubernetes/manifest/etcd.yaml 安装 Etcd 服务
* ****wait-control-plane：**** 等待 control-plan 部署的 Master 组件启动
* ****apiclient：**** 检查 Master 组件服务状态。
* ****uploadconfig：**** 更新配置
* ****kubelet：**** 使用 configMap 配置 kubelet
* ****patchnode：**** 更新 CNI 信息到 Node 上，通过注释的方式记录
* ****mark-control-plane：**** 为当前节点打标签，打了角色 Master，和不可调度标签，这样默认就不会使用 Master 节点来运行 Pod
* ****bootstrap-token：**** 生成 token 记录下来，后边使用 kubeadm join 往集群中添加节点时会用到
* ****addons：**** 安装附加组件 CoreDNS 和 kube-proxy

## **安装过程中出现的问题：**

为啥到这步连不上网络啊？？？？？？？？？？？

changchang1238个月前 (08-29)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

1. #2

IMG_256

因为网络原因ip的网段变了，master和slaver的ip都变了，改了没生效，需要怎么操作呢？

ljyinfo7个月前 (09-14)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

1. #3

IMG_257

我没出任何问题

Alibaba5个月前 (11-09)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

IMG_258

兄弟，请问你安装的是1.16的还是1.15的

mvc5个月前 (11-29)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

IMG_259

安装的1.16

Alibaba4个月前 (12-03)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

1. #4

IMG_260

我安装的1.17，咋不可以呀

linzi4个月前 (12-16)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

1. #5

IMG_261

V1.17.0安装成功。 可以避免的坑： 配置方面： 2核CPU；swapoff -a 是临时关闭交换空间 ，重启虚机后失效；永久关闭swap分区命令为sed -ri 's/.\*swap.\*/#&/' /etc/fstab 集群方面: 删除node节点 有三步 1.drain node 2.node机器执行kubeadm reset 向master注销 3.delete node（这时在没有执行delete之前，可以看到对应的drain节点处于NotReady,SchedulingDisabled）

lishubin4个月前 (12-17)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

IMG_262

请问下安装主节点这边超时 如何处理哦,我的也是v1.17.0版本哦

feilong3个月前 (01-13)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

1. #6

IMG_263

报错了 W1221 21:10:29.377758 20839 validation.go:28] Cannot validate kubelet config - no validator is available

hanhua4个月前 (12-21)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

IMG_264

兄弟 打扰了，在安装的Kubernetes1.17.0时，我也是报了你出现的 W1221 21:10:29.377758 20839 validation.go:28] Cannot validate kubelet config - no validator is available ，请问这个问题你解决了吗

xxkj\_20193个月前 (01-20)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

1. #7

IMG_265

请问在不同的主机的虚拟机上如何搭建集群

chards\_93个月前 (01-19)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

1. #8

IMG_266

root@kubernetes-base:~# kubeadm config print init-defaults --kubeconfig ClusterConfiguration > kubeadm.yml W0121 15:43:06.146010 6184 validation.go:28] Cannot validate kubelet config - no validator is available W0121 15:43:06.146091 6184 validation.go:28] Cannot validate kube-proxy config - no validator is available 遇到这个问题的朋友留下步

xxkj\_20193个月前 (01-21)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

IMG_267

遇到了

laoshiren12072个月前 (02-10)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

1. #9

IMG_268

W1221 21:10:29.377758 20839 validation.go:28] Cannot validate kubelet config - no validator is available,错误的是因为kubeadm版本高了,李哥给的会自动拉去最新版,使用apt-get remove kubernetes-cni卸载再重新安装apt-get update && apt-get install -y kubelet=1.15.2-00 kubeadm=1.15.2-00 kubectl=1.15.2-00 --allow-downgrades --allow-unauthenticated

shanguoyu2个月前 (02-02)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

IMG_269

兄弟，厉害了

having2个月前 (02-11)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

IMG_270

兄弟可以的，佩服了

mrwang2个月前 (02-20)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

IMG_271

顶你兄弟

crazy\_cpu2个月前 (02-23)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

1. #10

IMG_272

我电脑四核的可以玩吗

天狼0011个月前 (02-29)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

1. #11

IMG_273

kubeadm config images pull --config kubeadm.yml 这条命令执行完你们有输出吗

liulong6664周前 (03-18)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

IMG_274

没有。。。正常吗

shixue2周前 (03-30)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/javascript:;)

# kubernets配置网络

## **概述**

容器网络是容器选择连接到其他容器、主机和外部网络的机制。容器的 runtime 提供了各种网络模式，每种模式都会产生不同的体验。例如，Docker 默认情况下可以为容器配置以下网络：

* ****none：**** 将容器添加到一个容器专门的网络堆栈中，没有对外连接。
* ****host：**** 将容器添加到主机的网络堆栈中，没有隔离。
* ****default bridge：**** 默认网络模式。每个容器可以通过 IP 地址相互连接。
* ****自定义网桥：**** 用户定义的网桥，具有更多的灵活性、隔离性和其他便利功能。

## **什么是 CNI**

CNI(Container Network Interface) 是一个标准的，通用的接口。在容器平台，Docker，Kubernetes，Mesos 容器网络解决方案 flannel，calico，weave。只要提供一个标准的接口，就能为同样满足该协议的所有容器平台提供网络功能，而 CNI 正是这样的一个标准接口协议。

## **Kubernetes 中的 CNI 插件**

CNI 的初衷是创建一个框架，用于在配置或销毁容器时动态配置适当的网络配置和资源。插件负责为接口配置和管理 IP 地址，并且通常提供与 IP 管理、每个容器的 IP 分配、以及多主机连接相关的功能。容器运行时会调用网络插件，从而在容器启动时分配 IP 地址并配置网络，并在删除容器时再次调用它以清理这些资源。

运行时或协调器决定了容器应该加入哪个网络以及它需要调用哪个插件。然后，插件会将接口添加到容器网络命名空间中，作为一个 veth 对的一侧。接着，它会在主机上进行更改，包括将 veth 的其他部分连接到网桥。再之后，它会通过调用单独的 IPAM（IP地址管理）插件来分配 IP 地址并设置路由。

在 Kubernetes 中，kubelet 可以在适当的时间调用它找到的插件，为通过 kubelet 启动的 pod进行自动的网络配置。

Kubernetes 中可选的 CNI 插件如下：

* Flannel
* Calico
* Canal
* Weave

## **什么是 Calico**

Calico 为容器和虚拟机提供了安全的网络连接解决方案，并经过了大规模生产验证（在公有云和跨数千个集群节点中），可与 Kubernetes，OpenShift，Docker，Mesos，DC / OS 和 OpenStack 集成。

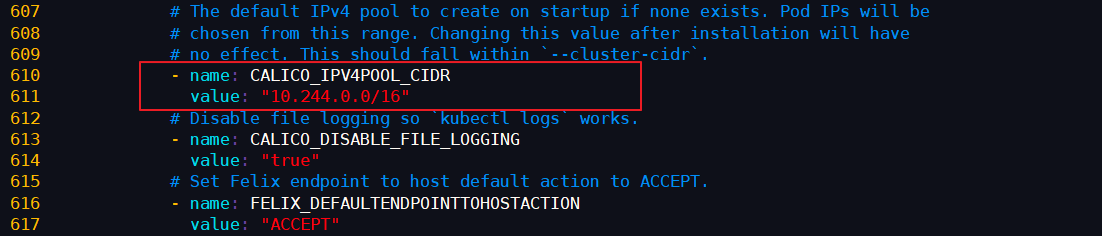
Calico 还提供网络安全规则的动态实施。使用 Calico 的简单策略语言，您可以实现对容器，虚拟机工作负载和裸机主机端点之间通信的细粒度控制。

## **下载 Calico 配置文件并修改**

1. wget https://docs.projectcalico.org/v3.8/manifests/calico.yaml
2. vi calico.yaml

修改第 611 行，将 192.168.0.0/16 修改为 10.244.0.0/16，可以通过如下命令快速查找

* 显示行号：:set number
* 查找字符：/要查找的字符，输入小写 n 下一个匹配项，输入大写 N 上一个匹配项



## **安装网络插件 Calico**

****注意：**** 截止到文章发表日期 2019 年 07 月 20 日，Calico 官方版本为 3.8

参考官方文档安装：[https://docs.projectcalico.org/v3.8/getting-started/kubernetes/](http://www.qfdmy.com/wp-content/themes/quanbaike/go.php?url=aHR0cHM6Ly9kb2NzLnByb2plY3RjYWxpY28ub3JnL3YzLjgvZ2V0dGluZy1zdGFydGVkL2t1YmVybmV0ZXMv" \t "http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e9%85%8d%e7%bd%ae%e7%bd%91%e7%bb%9c/_blank)

1. kubectl apply -f calico.yaml
2. # 输出如下
3. configmap/calico-config created
4. customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
5. customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
6. customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
7. customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
8. customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
9. customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
10. customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
11. customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
12. customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
13. customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
14. customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
15. customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
16. customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
17. customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
18. clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
19. clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
20. clusterrole.rbac.authorization.k8s.io/calico-node created
21. clusterrolebinding.rbac.authorization.k8s.io/calico-node created
22. daemonset.extensions/calico-node created
23. serviceaccount/calico-node created
24. deployment.extensions/calico-kube-controllers created
25. serviceaccount/calico-kube-controllers created

## **验证安装是否成功**

* 查看 Calico 网络插件处于 ****Running**** 状态即表示安装成功

1. watch kubectl get pods --all-namespaces
2. # 输出如下
3. NAMESPACE NAME READY STATUS RESTARTS AGE
4. kube-system calico-kube-controllers-658558ddf8-9zzjg 1/1 Running 0 90s
5. kube-system calico-node-9cr5f 1/1 Running 0 91s
6. kube-system calico-node-n99mz 1/1 Running 0 91s
7. kube-system calico-node-nl67v 1/1 Running 0 91s
8. kube-system coredns-bccdc95cf-9s4bm 1/1 Running 0 56m
9. kube-system coredns-bccdc95cf-s8ggd 1/1 Running 0 56m
10. kube-system etcd-kubernetes-master 1/1 Running 0 55m
11. kube-system kube-apiserver-kubernetes-master 1/1 Running 0 55m
12. kube-system kube-controller-manager-kubernetes-master 1/1 Running 0 55m
13. kube-system kube-proxy-8s87d 1/1 Running 0 36m
14. kube-system kube-proxy-cbnlb 1/1 Running 0 36m
15. kube-system kube-proxy-vwhxj 1/1 Running 0 56m
16. kube-system kube-scheduler-kubernetes-master 1/1 Running 0 55m

* 查看节点状态处于 ****Ready**** 即表示安装成功

1. kubectl get node
2. # 输出如下
3. NAME STATUS ROLES AGE VERSION
4. kubernetes-master Ready master 57m v1.15.0
5. kubernetes-node-01 Ready <none> 37m v1.15.0
6. kubernetes-node-02 Ready <none> 36m v1.15.0

窗体底端

**上一页**[Kubernetes 安装集群](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%ae%89%e8%a3%85%e9%9b%86%e7%be%a4/)

**下一个**[Kubernetes 第一个容器](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e7%ac%ac%e4%b8%80%e4%b8%aa%e5%ae%b9%e5%99%a8/)

### 评论 12

提交评论

窗体底端

1. #1

IMG_258

李哥有没有打算搞高可用集群的视频，比如master多节点，比如两个台同一个局域网的计算机怎么加入node节点扩容这些

changhe8个月前 (09-01)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e9%85%8d%e7%bd%ae%e7%bd%91%e7%bb%9c/javascript:;)

IMG_259

之前有做过 K8S 高可用集群的，不过碍于大家机器配置都不行，我 2.0 没说这个了，看下我视频里原先的 Kubernetes 视频合集

[李卫民](http://www.qfdmy.com/wp-content/themes/quanbaike/go.php?url=aHR0cHM6Ly93d3cuZnVudGwuY29t" \t "http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e9%85%8d%e7%bd%ae%e7%bd%91%e7%bb%9c/_blank)6个月前 (10-15)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e9%85%8d%e7%bd%ae%e7%bd%91%e7%bb%9c/javascript:;)

IMG_260

能讲讲就好，机器配置可以加的

skydianwu4天前[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e9%85%8d%e7%bd%ae%e7%bd%91%e7%bb%9c/javascript:;)

1. #2

IMG_261

李哥。啥时候我们也上传头像

shiguiwu6个月前 (10-20)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e9%85%8d%e7%bd%ae%e7%bd%91%e7%bb%9c/javascript:;)

IMG_262

这个问题不错

god\_lqs5个月前 (11-08)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e9%85%8d%e7%bd%ae%e7%bd%91%e7%bb%9c/javascript:;)

IMG_263

用全球公认头像就行了

mvc5个月前 (11-14)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e9%85%8d%e7%bd%ae%e7%bd%91%e7%bb%9c/javascript:;)

IMG_264

不行

shiguiwu5个月前 (11-14)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e9%85%8d%e7%bd%ae%e7%bd%91%e7%bb%9c/javascript:;)

1. #3

IMG_265

李哥，我从节点执行kubectl apply -f https://docs.projectcalico.org/v3.10/manifests/calico.yaml出现这个错误unable to recognize "https://docs.projectcalico.org/v3.10/manifests/calico.yaml": Get http://localhost:8080/api?timeout=32s: dial tcp 127.0.0.1:8080: connect: connection refused

chuan5个月前 (11-14)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e9%85%8d%e7%bd%ae%e7%bd%91%e7%bb%9c/javascript:;)

IMG_266

可以把配置文件 下到本地执行apply

lishubin4个月前 (12-17)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e9%85%8d%e7%bd%ae%e7%bd%91%e7%bb%9c/javascript:;)

IMG_267

朋友，你说的这个配置文件在哪儿下载，可以发个地址不？

john3个月前 (01-09)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e9%85%8d%e7%bd%ae%e7%bd%91%e7%bb%9c/javascript:;)

IMG_268

就是命令里的地址https://docs.projectcalico.org/v3.10/manifests/calico.yaml

\_0o0\_1个月前 (02-29)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e9%85%8d%e7%bd%ae%e7%bd%91%e7%bb%9c/javascript:;)

1. #4

IMG_269

李哥 kubectl get node status 都是ready 状态 但是 Calico 两个子节点的插件还是0的状态 两个子节点dns插件也是这样

leonheart1个月前 (03-01)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e9%85%8d%e7%bd%ae%e7%bd%91%e7%bb%9c/javascript:;)

# kubernets第一个容器

## **检查组件运行状态**

1. kubectl get cs
2. # 输出如下
3. NAME STATUS MESSAGE ERROR
4. # 调度服务，主要作用是将 POD 调度到 Node
5. scheduler Healthy ok
6. # 自动化修复服务，主要作用是 Node 宕机后自动修复 Node 回到正常的工作状态
7. controller-manager Healthy ok
8. # 服务注册与发现
9. etcd-0 Healthy {"health":"true"}

## **检查 Master 状态**

1. kubectl cluster-info
2. # 输出如下
3. Kubernetes master is running at https://192.168.141.110:6443
4. KubeDNS is running at https://192.168.141.110:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
5. To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

## **检查 Nodes 状态**

1. kubectl get nodes
2. # 输出如下
3. NAME STATUS ROLES AGE VERSION
4. kubernetes-master Ready master 72m v1.15.0
5. kubernetes-node-01 Ready <none> 52m v1.15.0
6. kubernetes-node-02 Ready <none> 51m v1.15.0

## **运行第一个容器实例**

1. # 使用 kubectl 命令创建两个监听 80 端口的 Nginx Pod（Kubernetes 运行容器的最小单元）
2. kubectl run nginx --image=nginx --replicas=2 --port=80
3. # 输出如下
4. kubectl run --generator=deployment/apps.v1 is DEPRECATED and will be removed in a future version. Use kubectl run --generator=run-pod/v1 or kubectl create instead.
5. deployment.apps/nginx created

## **查看全部 Pods 的状态**

1. kubectl get pods
2. # 输出如下
3. NAME READY STATUS RESTARTS AGE
4. nginx-7c45b84548-mv8n8 1/1 Running 0 36s
5. nginx-7c45b84548-vp2x6 1/1 Running 0 36s

## **查看已部署的服务**

1. kubectl get deployment
2. # 输出如下
3. NAME READY UP-TO-DATE AVAILABLE AGE
4. nginx 2/2 2 2 70s

## **发布服务**

1. # 使用负载均衡模式发布服务，让用户可以访问
2. kubectl expose deployment nginx --port=80 --type=LoadBalancer
3. # 输出如下
4. service/nginx exposed

## **查看已发布的服务**

1. kubectl get services
2. # 输出如下
3. NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
4. kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 81m
5. nginx LoadBalancer 10.102.37.240 <pending> 80:31406/TCP 115s

## **查看服务详情**

1. kubectl describe service nginx
2. # 输出如下
3. Name: nginx
4. Namespace: default
5. Labels: run=nginx
6. Annotations: <none>
7. Selector: run=nginx
8. Type: LoadBalancer
9. IP: 10.102.37.240
10. Port: <unset> 80/TCP
11. TargetPort: 80/TCP
12. # 发布到外网的服务端口
13. NodePort: <unset> 31406/TCP
14. Endpoints: 192.168.140.66:80,192.168.141.194:80
15. Session Affinity: None
16. External Traffic Policy: Cluster
17. Events: <none>

## **验证是否成功**

通过浏览器访问 Node 服务器，此时 Kubernetes 会以负载均衡的方式访问部署的 Nginx 服务，能够正常看到 Nginx 的欢迎页即表示成功

1. http://192.168.141.120:31406/

## **停止服务**

* 删除已部署的服务

1. kubectl delete deployment nginx
2. # 输出如下
3. deployment.extensions "nginx" deleted

* 删除已发布的服务

1. kubectl delete service nginx
2. # 输出如下
3. service "nginx" deleted

窗体顶端

完成

窗体底端

**上一页**[Kubernetes 配置网络](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e9%85%8d%e7%bd%ae%e7%bd%91%e7%bb%9c/)

**下一个**[Kubernetes 概念总结](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e6%a6%82%e5%bf%b5%e6%80%bb%e7%bb%93/)

### 评论 7

窗体顶端

IMG_256

李久成

提交评论

窗体底端

1. #1

IMG_257

怎么回事 scheduler controller-manager etcd-0

xuhui1996765个月前 (11-07)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e7%ac%ac%e4%b8%80%e4%b8%aa%e5%ae%b9%e5%99%a8/javascript:;)

1. #2

IMG_258

检查组件运行状态的时候显示,但是能正常使用，怎么回事？

Alibaba5个月前 (11-09)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e7%ac%ac%e4%b8%80%e4%b8%aa%e5%ae%b9%e5%99%a8/javascript:;)

1. #3

IMG_259

检查组件运行状态的时候都显示《unknow》,但是能正常使用，怎么回事？

Alibaba5个月前 (11-09)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e7%ac%ac%e4%b8%80%e4%b8%aa%e5%ae%b9%e5%99%a8/javascript:;)

IMG_260

我也一样，这个文章https://segmentfault.com/a/1190000020912684说的很清楚

chuan5个月前 (11-14)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e7%ac%ac%e4%b8%80%e4%b8%aa%e5%ae%b9%e5%99%a8/javascript:;)

1. #4

IMG_261

能演示个不是80端口pod吗？80这个端口太特殊了，我自己改个随便个端口访问不通，但是照着例子是只可以的

hailou2个月前 (02-16)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e7%ac%ac%e4%b8%80%e4%b8%aa%e5%ae%b9%e5%99%a8/javascript:;)

1. #5

IMG_262

1.18版本删除了--replicas 没找到解决办法，最后还是用的1.16.0版本

ty\_qfdmy2周前 (04-02)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e7%ac%ac%e4%b8%80%e4%b8%aa%e5%ae%b9%e5%99%a8/javascript:;)

1. #6

IMG_263

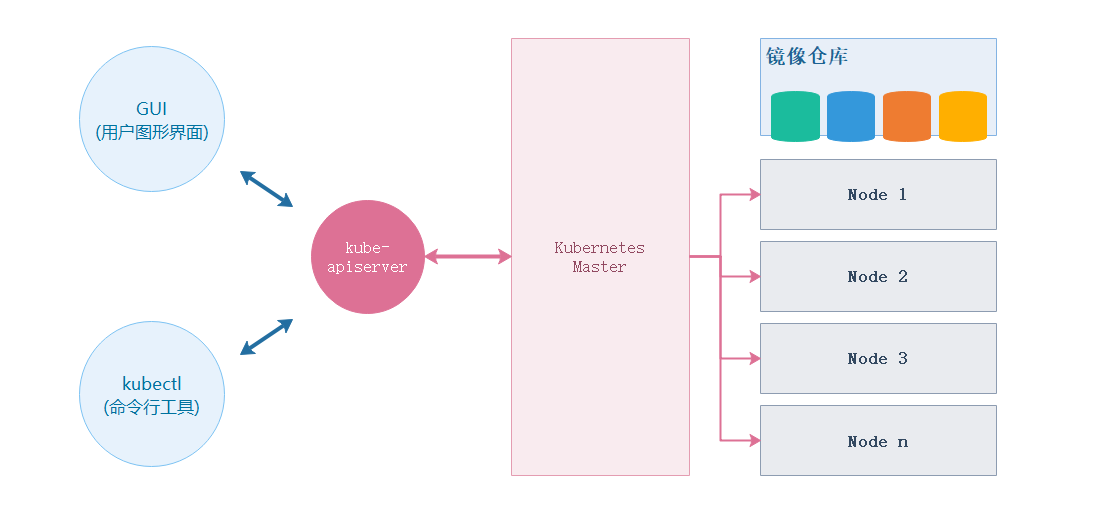
1.18.0没有--replicas，需要写yaml配置文件，亲测可以启动成功

古娜拉黑暗之神5天前[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e7%ac%ac%e4%b8%80%e4%b8%aa%e5%ae%b9%e5%99%a8/javascript:;)

# kubernets概念总结

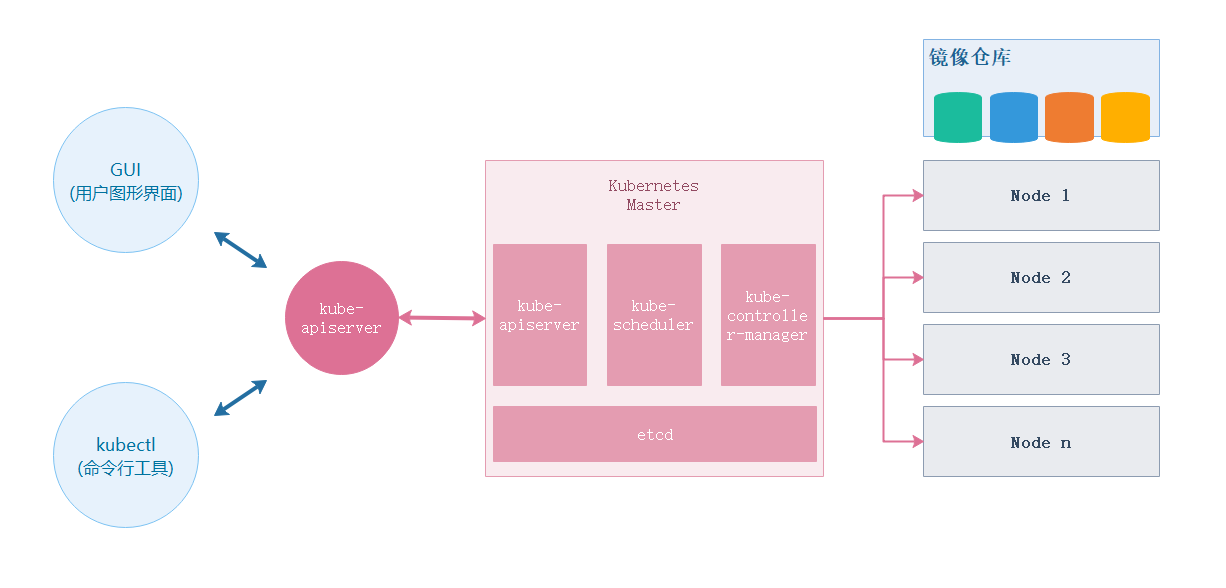
## **什么是 Kubernetes**

Kubernetes 是一个开源的 Docker 容器编排系统，它可以调度计算集群的节点，动态管理上面的作业，保证它们按用户期望的状态运行。通过使用「labels」和「pods」的概念，Kubernetes 将应用按逻辑单元进行分组，方便管理和服务发现。



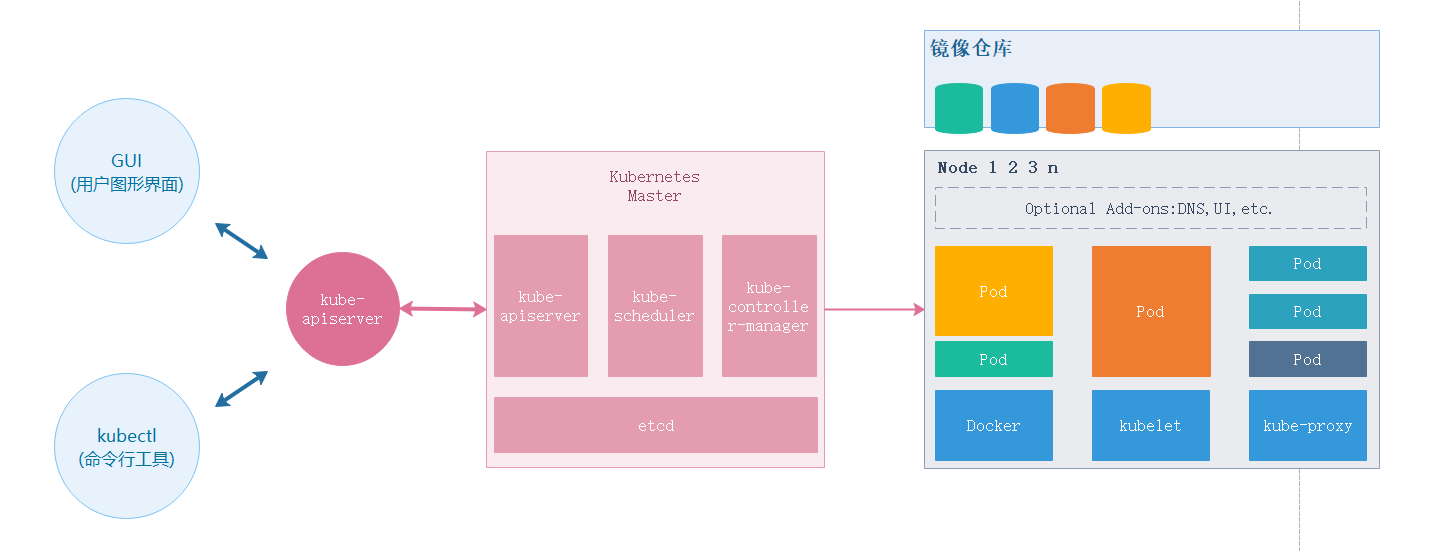
* ****pods：**** 是一组紧密关联的容器集合，它们共享 IPC(进程间通信)、Network(网络) 和 UTS namespace(UTS 命名空间是 Linux 命名空间的一个子系统，主要作用是完成对容器 Hostname 和 Domain 的隔离，同时保存内核名称、版本、以及底层体系结构类型等信息)，是 Kubernetes 调度的基本单位。
* ****labels：**** 键值对(key/value)标签，可以被关联到如 Pod 这样的对象上，主要作用是给用户一个直观的感受，比如这个 Pod 是用来放置数据库的
* ****GUI：**** 用户图形界面，可以是 Web 用户界面，比如使用 kubernetes-dashboard 组件，用户可以通过 Dashboard 在 Kubernetes 集群中部署容器化的应用，可以查看集群中应用的运行情况，同时也能够基于 Dashboard 创建或修改部署、任务、服务等 Kubernetes 的资源。通过部署向导，用户能够对部署进行扩缩容，进行滚动更新、重启 Pod 和部署新应用。当然，通过 Dashboard 也能够查看 Kubernetes 资源的状态
* ****kubectl：**** 用于管理 Kubernetes 集群的命令行工具
* ****kube-apiserver：**** 提供了资源操作的唯一入口，并提供认证、授权、访问控制、API 注册和发现等机制
* ****Kubernetes Master：**** Kubernetes 集群主节点，主要由 kube-apiserver、kube-scheduler、kube-controller-manager、etcd 四个模块组成
* ****Kubernetes Node：**** Kubernetes 集群子节点，主要由 kubelet、kube-proxy、runtime 三个模块组成
* ****Image Registry：**** 镜像仓库，比如：Ducker HUB 或 Docker 私服

## **Kubernetes Master**



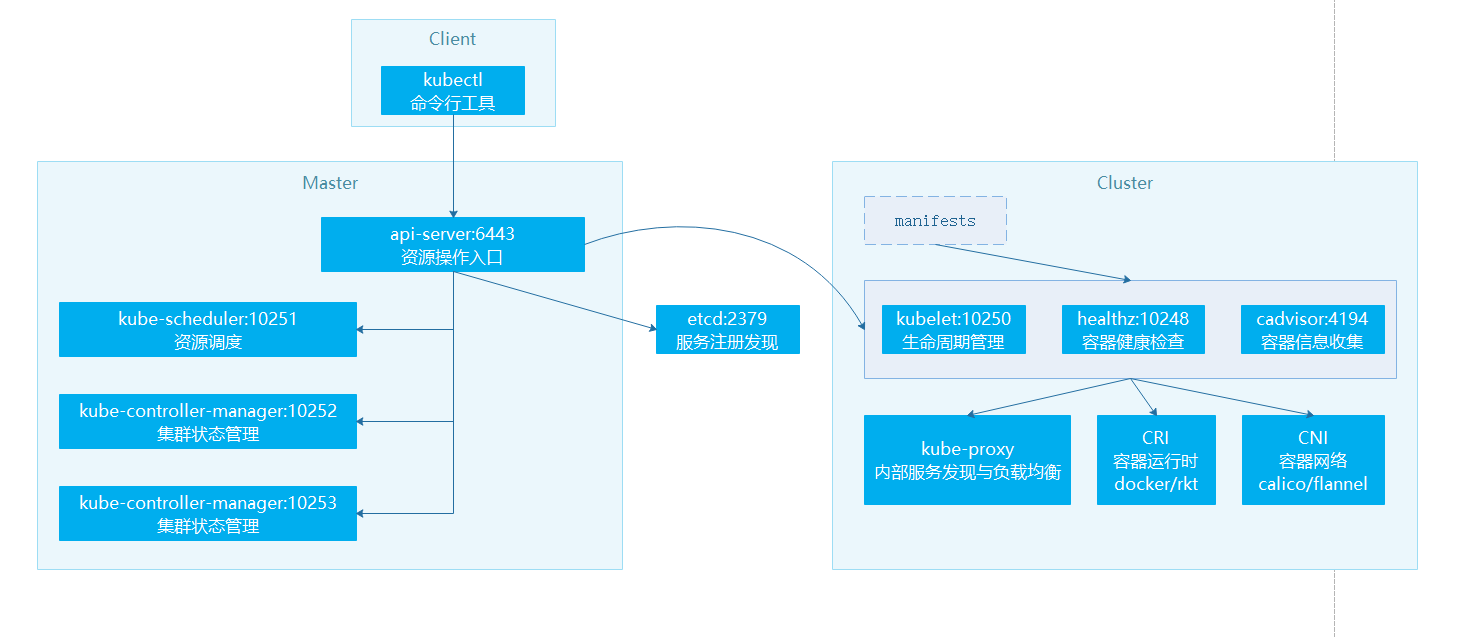
* ****kube-apiserver：**** 提供了资源操作的唯一入口，并提供认证、授权、访问控制、API 注册和发现等机制
* ****kube-scheduler：**** 负责资源的调度，按照预定的调度策略将 Pod 调度到相应的机器上
* ****kube-controller-manager：**** 负责维护集群的状态，比如故障检测、自动扩展、滚动更新等
* ****etcd：**** CoreOS 基于 Raft 开发的分布式 key-value 存储，可用于服务发现、共享配置以及一致性保障（如数据库选主、分布式锁等）

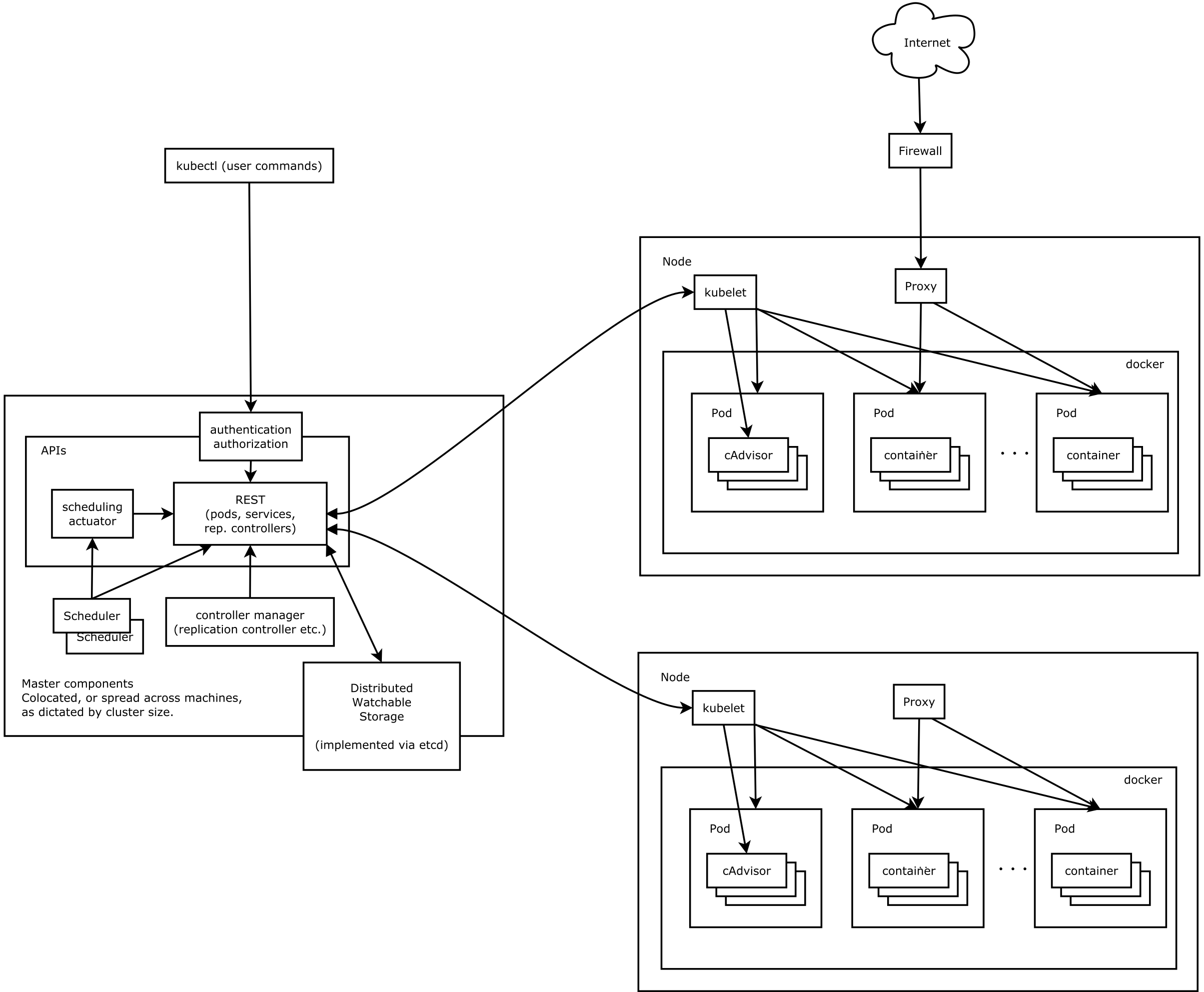
## **Kubernetes Node**



* ****runtime：**** 负责镜像管理以及 Pod 和容器的真正运行（CRI，Container Runtime Interface），默认的容器运行时为 Docker，还支持 RKT 容器
* ****kubelet：**** 负责维持容器的生命周期，同时也负责 Volume（CVI，Container Volume Interface）和网络（CNI，Container Network Interface）的管理
* ****kube-proxy：**** 负责为 Service 提供 cluster 内部的服务发现和负载均衡

## **Kubernetes 架构**





窗体顶端

完成

窗体底端

# kubernetes通过资源配置运行容器

## **概述**

我们知道通过 run 命令启动容器非常麻烦，Docker 提供了 Compose 为我们解决了这个问题。那 Kubernetes 是如何解决这个问题的呢？其实很简单，使用 kubectl create 命令就可以做到和 Compose 一样的效果了，该命令可以通过配置文件快速创建一个集群资源对象。

## **创建 YAML 配置文件**

### **部署 Deployment**

创建一个名为 nginx-deployment.yml 的配置文件

1. # API 版本号
2. apiVersion: apps/v1
3. # 类型，如：Pod/ReplicationController/Deployment/Service/Ingress
4. kind: Deployment
5. # 元数据
6. metadata:
7. # Kind 的名称
8. name: nginx-app
9. spec:
10. selector:
11. matchLabels:
12. app: nginx
13. # 部署的实例数量
14. replicas: 2
15. template:
16. metadata:
17. labels:
18. # 容器标签的名字，发布 Service 时，selector 需要和这里对应
19. app: nginx
20. spec:
21. # 配置容器，数组类型，说明可以配置多个容器
22. containers:
23. # 容器名称
24. - name: nginx
25. # 容器镜像
26. image: nginx:1.17
27. # 只有镜像不存在时，才会进行镜像拉取
28. imagePullPolicy: IfNotPresent
29. # 暴露端口
30. ports:
31. # Pod 端口
32. - containerPort: 80
33. # 部署
34. kubectl create -f nginx-deployment.yml
35. # 删除
36. kubectl delete -f nginx-deployment.yml

### **发布 Service**

创建一个名为 nginx-service.yml 的配置文件

1. API 版本号
2. apiVersion: v1
3. # 类型，如：Pod/ReplicationController/Deployment/Service/Ingress
4. kind: Service
5. # 元数据
6. metadata:
7. # Kind 的名称
8. name: nginx-http
9. spec:
10. # 暴露端口
11. ports:
12. ## Service 暴露的端口
13. - port: 80
14. ## Pod 上的端口，这里是将 Service 暴露的端口转发到 Pod 端口上
15. targetPort: 80
16. # 类型
17. type: LoadBalancer
18. # 标签选择器
19. selector:
20. # 需要和上面部署的 Deployment 标签名对应
21. app: nginx
22. # 部署
23. kubectl create -f nginx-service.yml
24. # 删除
25. kubectl delete -f nginx-service.yml

### **镜像拉取策略**

支持三种 ImagePullPolicy

* ****Always：**** 不管镜像是否存在都会进行一次拉取
* ****Never：**** 不管镜像是否存在都不会进行拉取
* ****IfNotPresent：**** 只有镜像不存在时，才会进行镜像拉取

注意

* 默认为 IfNotPresent，但 :latest 标签的镜像默认为 Always
* 拉取镜像时 Docker 会进行校验，如果镜像中的 MD5 码没有变，则不会拉取镜像数据
* 生产环境中应该尽量避免使用 :latest 标签，而开发环境中可以借助 :latest 标签自动拉取最新的镜像

## **验证是否生效**

### **查看 Pod 列表**

1. kubectl get pods
2. # 输出如下
3. NAME READY STATUS RESTARTS AGE
4. nginx-app-64bb598779-2pplx 1/1 Running 0 25m
5. nginx-app-64bb598779-824lc 1/1 Running 0 25m

### **查看 Deployment 列表**

1. kubectl get deployment
2. # 输出如下
3. NAME READY UP-TO-DATE AVAILABLE AGE
4. nginx-app 2/2 2 2 25m

### **查看 Service 列表**

1. kubectl get service
2. # 输出如下
3. NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
4. kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 20h
5. nginx-http LoadBalancer 10.98.49.142 <pending> 80:31631/TCP 14m

### **查看 Service 详情**

1. kubectl describe service nginx-app
2. # 输出如下
3. Name: nginx-http
4. Namespace: default
5. Labels: <none>
6. Annotations: <none>
7. Selector: name=nginx
8. Type: LoadBalancer
9. IP: 10.98.49.142
10. Port: <unset> 80/TCP
11. TargetPort: 80/TCP
12. NodePort: <unset> 31631/TCP
13. Endpoints: 10.244.141.205:80,10.244.2.4:80
14. Session Affinity: None
15. External Traffic Policy: Cluster
16. Events: <none>

### **通过浏览器访问**

通过浏览器访问 [http://192.168.141.120:31631/](http://www.qfdmy.com/wp-content/themes/quanbaike/go.php?url=aHR0cDovLzE5Mi4xNjguMTQxLjEyMDozMTYzMS8=" \t "http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e9%80%9a%e8%bf%87%e8%b5%84%e6%ba%90%e9%85%8d%e7%bd%ae%e8%bf%90%e8%a1%8c%e5%ae%b9%e5%99%a8/_blank) ，出现 Nginx 欢迎页即表示成功

## **集成环境部署**

也可以不区分配置文件，一次性部署 Deployment 和 Service，创建一个名为 nginx.yml 的配置文件，配置内容如下：

1. apiVersion: apps/v1
2. kind: Deployment
3. metadata:
4. name: nginx-app
5. spec:
6. selector:
7. matchLabels:
8. app: nginx
9. replicas: 2
10. template:
11. metadata:
12. labels:
13. app: nginx
14. spec:
15. containers:
16. - name: nginx
17. image: nginx
18. ports:
19. - containerPort: 80
20. ---
21. apiVersion: v1
22. kind: Service
23. metadata:
24. name: nginx-http
25. spec:
26. ports:
27. - port: 80
28. targetPort: 80
29. # 可以指定 NodePort 端口，默认范围是：30000-32767
30. # nodePort: 30080
31. type: LoadBalancer
32. selector:
33. app: nginx
34. # 部署
35. kubectl create -f nginx.yml
36. # 删除
37. kubectl delete -f nginx.yml

## **附：扩展阅读**

### **修改默认的端口范围**

Kubernetes 服务的 NodePort 默认端口范围是 30000-32767，在某些场合下，这个限制不太适用，我们可以自定义它的端口范围，操作步骤如下：

编辑 vi /etc/kubernetes/manifests/kube-apiserver.yaml 配置文件，增加配置 --service-node-port-range=2-65535

1. apiVersion: v1
2. kind: Pod
3. metadata:
4. creationTimestamp: null
5. labels:
6. component: kube-apiserver
7. tier: control-plane
8. name: kube-apiserver
9. namespace: kube-system
10. spec:
11. containers:
12. - command:
13. - kube-apiserver
14. # 在这里增加配置即可
15. - --service-node-port-range=2-65535
16. - --advertise-address=192.168.141.150
17. - --allow-privileged=true
18. - --authorization-mode=Node,RBAC
19. - --client-ca-file=/etc/kubernetes/pki/ca.crt
20. - --enable-admission-plugins=NodeRestriction
21. - --enable-bootstrap-token-auth=true
22. - --etcd-cafile=/etc/kubernetes/pki/etcd/ca.crt
23. - --etcd-certfile=/etc/kubernetes/pki/apiserver-etcd-client.crt
24. // 以下配置省略...

使用 docker ps 命令找到 kube-apiserver 容器，再使用 docker restart <ApiServer 容器 ID> 即可生效。

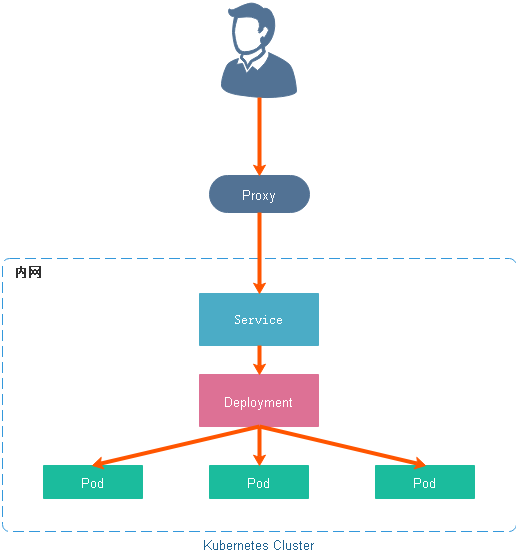
# kubernetes Ingress 简介

## **术语**

* ****节点：**** Kubernetes 集群中的服务器
* ****集群：**** Kubernetes 管理的一组服务器集合
* ****边界路由器：**** 为局域网和 Internet 路由数据包的路由器，执行防火墙保护局域网络
* ****集群网络：**** 遵循 Kubernetes 网络模型实现集群内的通信的具体实现，比如 Flannel 和 Calico
* ****服务：**** Kubernetes 的服务 (Service) 是使用标签选择器标识的一组 Pod Service (Deployment)。 ****除非另有说明，否则服务的虚拟 IP 仅可在集群内部访问****

## **内部访问方式 ClusterIP**

ClusterIP 服务是 Kubernetes 的默认服务。它给你一个集群内的服务，集群内的其它应用都可以访问该服务。集群外部无法访问它。在某些场景下我们可以使用 Kubernetes 的 Proxy 模式来访问服务，比如调试服务时。



## **三种外部访问方式**

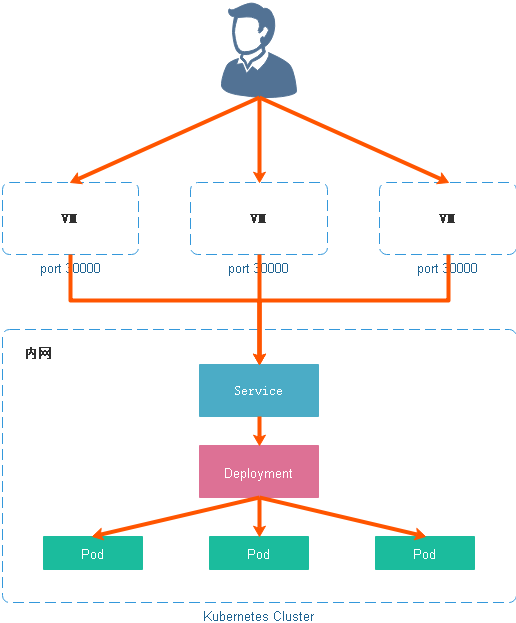
### **NodePort**

NodePort 服务是引导外部流量到你的服务的最原始方式。NodePort，正如这个名字所示，****在所有节点（虚拟机）上开放一个特定端口****，任何发送到该端口的流量都被转发到对应服务。

NodePort 服务特征如下：

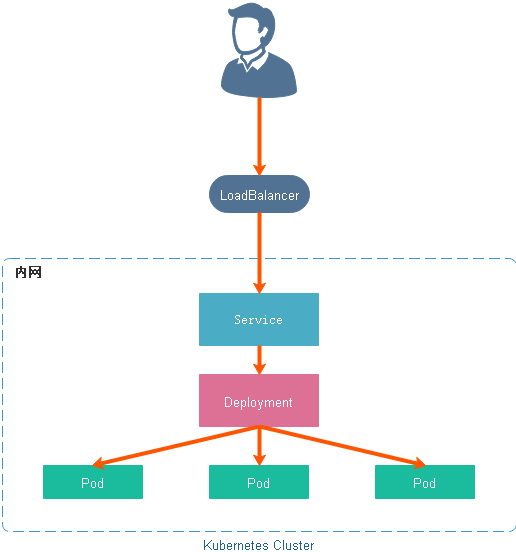
* 每个端口只能是一种服务
* 端口范围只能是 30000-32767（可调）
* 不在 YAML 配置文件中指定则会分配一个默认端口

****建议：**** 不要在生产环境中使用这种方式暴露服务，大多数时候我们应该让 Kubernetes 来选择端口



### **LoadBalancer**

LoadBalancer 服务是暴露服务到 Internet 的标准方式。所有通往你指定的端口的流量都会被转发到对应的服务。它没有过滤条件，没有路由等。这意味着你几乎可以发送任何种类的流量到该服务，像 HTTP，TCP，UDP，WebSocket，gRPC 或其它任意种类。

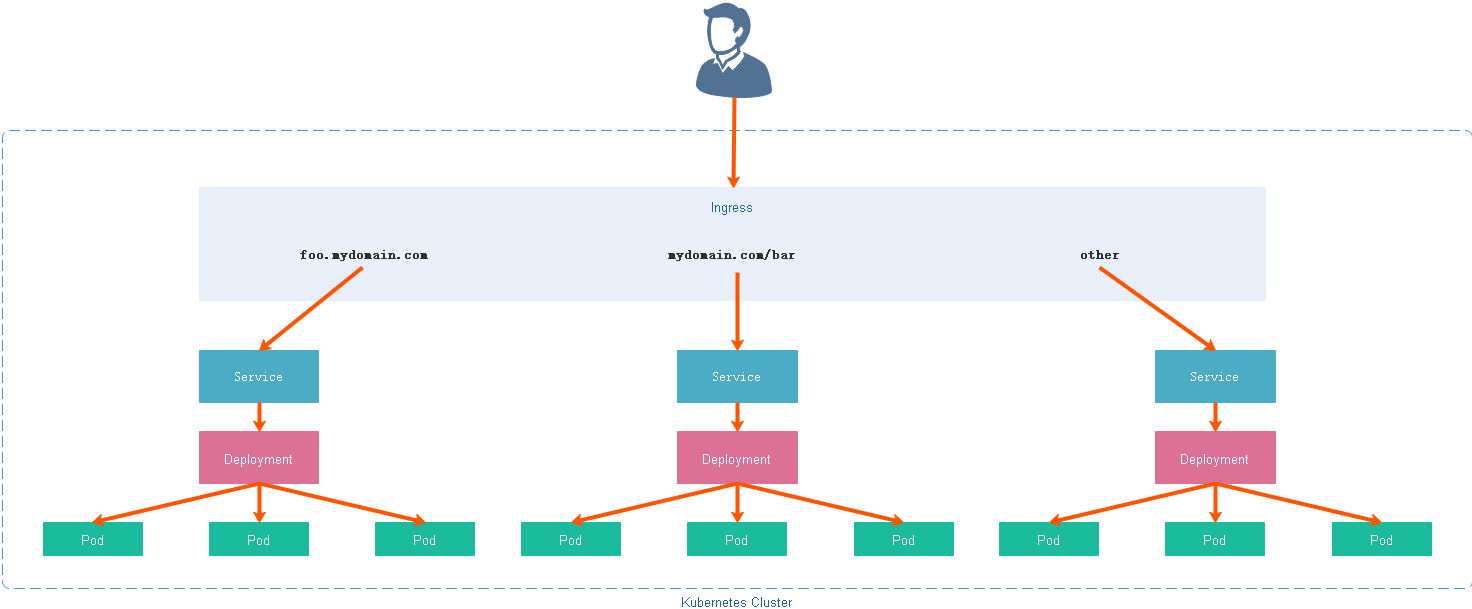


### **Ingress**

Ingress 事实上不是一种服务类型。相反，它处于多个服务的前端，扮演着 “智能路由” 或者集群入口的角色。你可以用 Ingress 来做许多不同的事情，各种不同类型的 Ingress 控制器也有不同的能力。它允许你基于路径或者子域名来路由流量到后端服务。

Ingress 可能是暴露服务的最强大方式，但同时也是最复杂的。Ingress 控制器有各种类型，包括 Google Cloud Load Balancer， Nginx，Contour，Istio，等等。它还有各种插件，比如 cert-manager (它可以为你的服务自动提供 SSL 证书)/

如果你想要使用同一个 IP 暴露多个服务，这些服务都是使用相同的七层协议（典型如 HTTP），你还可以获取各种开箱即用的特性（比如 SSL、认证、路由等等）



## **什么是 Ingress**

通常情况下，Service 和 Pod 的 IP 仅可在集群内部访问。集群外部的请求需要通过负载均衡转发到 Service 在 Node 上暴露的 NodePort 上，然后再由 kube-proxy 通过边缘路由器 (edge router) 将其转发给相关的 Pod 或者丢弃。而 Ingress 就是为进入集群的请求提供路由规则的集合

Ingress 可以给 Service 提供集群外部访问的 URL、负载均衡、SSL 终止、HTTP 路由等。为了配置这些 Ingress 规则，集群管理员需要部署一个 Ingress Controller，它监听 Ingress 和 Service 的变化，并根据规则配置负载均衡并提供访问入口。

# Nginx虚拟主机

## **什么是 Nginx**

Nginx 是一款高性能的 HTTP 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器。由俄罗斯的程序设计师 Igor Sysoev 所开发，官方测试 Nginx 能够支支撑 5 万并发链接，并且 CPU、内存等资源消耗却非常低，运行非常稳定。

## **Nginx 的应用场景**

* HTTP 服务器：Nginx 是一个 HTTP 服务可以独立提供 HTTP 服务。可以做网页静态服务器。
* 虚拟主机：可以实现在一台服务器虚拟出多个网站。例如个人网站使用的虚拟主机。
* 反向代理，负载均衡：当网站的访问量达到一定程度后，单台服务器不能满足用户的请求时，需要用多台服务器集群可以使用 Nginx 做反向代理。并且多台服务器可以平均分担负载，不会因为某台服务器负载高宕机而某台服务器闲置的情况。

## **Nginx 虚拟主机**

### **什么是虚拟主机**

虚拟主机是一种特殊的软硬件技术，它可以将网络上的每一台计算机分成多个虚拟主机，每个虚拟主机可以独立对外提供 www 服务，这样就可以实现一台主机对外提供多个 web 服务，每个虚拟主机之间是独立的，互不影响的。

通过 Nginx 可以实现虚拟主机的配置，Nginx 支持三种类型的虚拟主机配置

* 基于 IP 的虚拟主机
* 基于域名的虚拟主机
* 基于端口的虚拟主机

### **Nginx 配置文件的结构**

1. # ...
2. events {
3. # ...
4. }
5. http {
6. # ...
7. server{
8. # ...
9. }
10. # ...
11. server{
12. # ...
13. }
14. }

****注意：**** 每个 server 就是一个虚拟主机

## **基于 Docker 部署 Nginx**

1. version: '3.1'
2. services:
3. nginx:
4. restart: always
5. image: nginx
6. container\_name: nginx
7. ports:
8. - 80:80
9. volumes:
10. - ./conf/nginx.conf:/etc/nginx/nginx.conf
11. - ./html:/usr/share/nginx/html

### **基于端口的虚拟主机配置**

#### **需求**

* Nginx 对外提供 80 和 8080 两个端口监听服务
* 请求 80 端口则请求 html80 目录下的 html
* 请求 8080 端口则请求 html8080 目录下的 html

#### **操作流程**

* 创建目录及文件，在 /usr/local/docker/nginx/html 目录下创建 html80 和 html8080 两个目录，并分别创建两个 index.html 文件
* 配置虚拟主机，创建并修改 /usr/local/docker/nginx/conf 目录下的 nginx.conf

1. # 启动进程,通常设置成和 CPU 的数量相等
2. worker\_processes 1;
3. events {
4. # epoll 是多路复用 IO(I/O Multiplexing) 中的一种方式
5. # 但是仅用于 linux2.6 以上内核,可以大大提高 nginx 的性能
6. use epoll;
7. # 单个后台 worker process 进程的最大并发链接数
8. worker\_connections 1024;
9. }
10. http {
11. # 设定 mime 类型,类型由 mime.type 文件定义
12. include mime.types;
13. default\_type application/octet-stream;
14. # sendfile 指令指定 nginx 是否调用 sendfile 函数（zero copy 方式）来输出文件，对于普通应用，
15. # 必须设为 on，如果用来进行下载等应用磁盘 IO 重负载应用，可设置为 off，以平衡磁盘与网络 I/O 处理速度，降低系统的 uptime.
16. sendfile on;
17. # 连接超时时间
18. keepalive\_timeout 65;
19. # 设定请求缓冲
20. client\_header\_buffer\_size 2k;
21. # 配置虚拟主机 192.168.141.121
22. server {
23. # 监听的 IP 和端口，配置 192.168.141.121:80
24. listen 80;
25. # 虚拟主机名称这里配置 IP 地址
26. server\_name 192.168.141.121;
27. # 所有的请求都以 / 开始，所有的请求都可以匹配此 location
28. location / {
29. # 使用 root 指令指定虚拟主机目录即网页存放目录
30. # 比如访问 http://ip/index.html 将找到 /usr/local/docker/nginx/html/html80/index.html
31. # 比如访问 http://ip/item/index.html 将找到 /usr/local/docker/nginx/html/html80/item/index.html
32. root /usr/share/nginx/html/html80;
33. # 指定欢迎页面，按从左到右顺序查找
34. index index.html index.htm;
35. }
36. }
37. # 配置虚拟主机 192.168.141.121
38. server {
39. listen 8080;
40. server\_name 192.168.141.121;
41. location / {
42. root /usr/share/nginx/html/html8080;
43. index index.html index.htm;
44. }
45. }
46. }

### **基于域名的虚拟主机配置**

#### **需求**

* 两个域名指向同一台 Nginx 服务器，用户访问不同的域名显示不同的网页内容
* 两个域名是 ingress.funtl.com 和 service.funtl.com
* Nginx 服务器使用虚拟机 192.168.141.121

#### **操作流程**

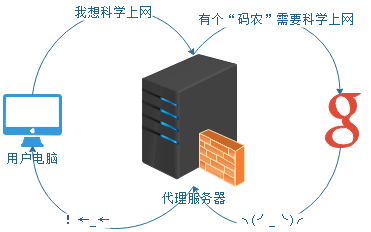
* 配置 Hosts，通过 host 文件指定 service1.funtl.com 和 service2.funtl.com 对应 192.168.141.121 虚拟机，修改 window 的 hosts 文件：（C:\Windows\System32\drivers\etc）
* 创建目录及文件，在 /usr/local/docker/nginx/html 目录下创建 service1 和 service1 两个目录，并分别创建两个 index.html 文件
* 配置虚拟主机，修改 /usr/local/docker/nginx/conf 目录下的 nginx.conf

1. user nginx;
2. worker\_processes 1;
3. events {
4. use epoll;
5. worker\_connections 1024;
6. }
7. http {
8. include mime.types;
9. default\_type application/octet-stream;
10. sendfile on;
11. keepalive\_timeout 65;
12. server {
13. listen 80;
14. server\_name service1.funtl.com;
15. location / {
16. root /usr/share/nginx/html/service1;
17. index index.html index.htm;
18. }
19. }
20. server {
21. listen 80;
22. server\_name service2.funtl.com;
23. location / {
24. root /usr/share/nginx/html/service2;
25. index index.html index.htm;
26. }
27. }
28. }

# Nginx反向代理

## **什么是代理服务器**

代理服务器，客户机在发送请求时，不会直接发送给目的主机，而是先发送给代理服务器，代理服务接受客户机请求之后，再向主机发出，并接收目的主机返回的数据，存放在代理服务器的硬盘中，再发送给客户机。

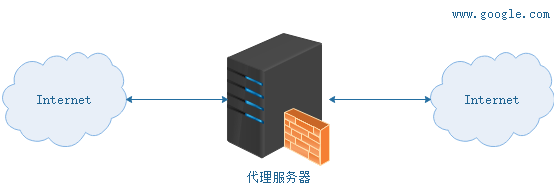


## **为什么要使用代理服务器**

* ****提高访问速度：**** 由于目标主机返回的数据会存放在代理服务器的硬盘中，因此下一次客户再访问相同的站点数据时，会直接从代理服务器的硬盘中读取，起到了缓存的作用，尤其对于热门站点能明显提高请求速度。
* ****防火墙作用：**** 由于所有的客户机请求都必须通过代理服务器访问远程站点，因此可在代理服务器上设限，过滤某些不安全信息。
* ****通过代理服务器访问不能访问的目标站点：**** 互联网上有许多开放的代理服务器，客户机在访问受限时，可通过不受限的代理服务器访问目标站点，通俗说，我们使用的翻墙浏览器就是利用了代理服务器，虽然不能出国，但也可直接访问外网。

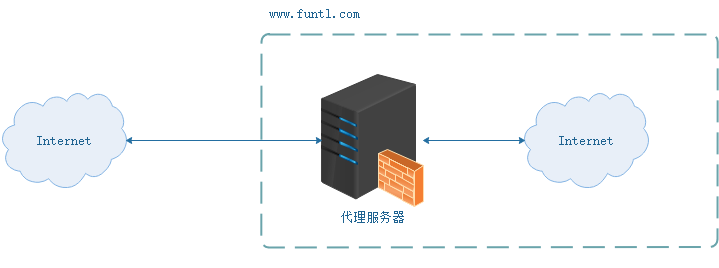
## **什么是正向代理**

正向代理，架设在客户机与目标主机之间，只用于代理内部网络对 Internet 的连接请求，客户机必须指定代理服务器,并将本来要直接发送到 Web 服务器上的 HTTP 请求发送到代理服务器中。



## **什么是反向代理**

反向代理服务器架设在服务器端，通过缓冲经常被请求的页面来缓解服务器的工作量，将客户机请求转发给内部网络上的目标服务器；并将从服务器上得到的结果返回给 Internet 上请求连接的客户端，此时代理服务器与目标主机一起对外表现为一个服务器。



## **反向代理有哪些主要应用**

现在许多大型 web 网站都用到反向代理。除了可以防止外网对内网服务器的恶性攻击、缓存以减少服务器的压力和访问安全控制之外，还可以进行负载均衡，将用户请求分配给多个服务器。

## **使用 Nginx 反向代理 Tomcat**

### **需求**

* 两个 Tomcat 服务通过 Nginx 反向代理
* Nginx 服务器：192.168.141.121:80
* Tomcat1 服务器：192.168.141.121:8081
* Tomcat2 服务器：192.168.141.121:8082

### **启动 Tomcat 容器**

启动两个 Tomcat 容器，映射端口为 8081 和 8082，docker-compose.yml 如下：

1. version: '3.1'
2. services:
3. tomcat1:
4. image: tomcat
5. container\_name: tomcat1
6. ports:
7. - 8081:8080
8. tomcat2:
9. image: tomcat
10. container\_name: tomcat2
11. ports:
12. - 8082:8080

### **配置 Nginx 反向代理**

修改 /usr/local/docker/nginx/conf 目录下的 nginx.conf 配置文件：

1. user nginx;
2. worker\_processes 1;
3. events {
4. use epoll;
5. worker\_connections 1024;
6. }
7. http {
8. include mime.types;
9. default\_type application/octet-stream;
10. sendfile on;
11. keepalive\_timeout 65;
12. # 配置一个代理即 tomcat1 服务器
13. upstream tomcatServer1 {
14. server 192.168.141.121:8081;
15. }
16. # 配置一个代理即 tomcat2 服务器
17. upstream tomcatServer2 {
18. server 192.168.141.121:8082;
19. }
20. # 配置一个虚拟主机
21. server {
22. listen 80;
23. server\_name service1.funtl.com;
24. location / {
25. # 域名 service1.funtl.com 的请求全部转发到 tomcatServer1 即 tomcat1 服务上
26. proxy\_pass http://tomcatServer1;
27. # 欢迎页面，按照从左到右的顺序查找页面
28. index index.jsp index.html index.htm;
29. }
30. }
31. server {
32. listen 80;
33. server\_name service2.funtl.com;
34. location / {
35. # 域名 service2.funtl.com 的请求全部转发到 tomcatServer2 即 tomcat2 服务上
36. proxy\_pass http://tomcatServer2;
37. index index.jsp index.html index.htm;
38. }
39. }
40. }

****注意：**** 新版 Nginx 的 MARKDOWN\_HASHbc3b0556316b0ba241ae6bb86b76e8a2MARKDOWNHASH 配置中的名称不可以有下划线("")，否则会报 400 错误

# Nginx 负载均衡

## **什么是负载均衡**

负载均衡建立在现有网络结构之上，它提供了一种廉价有效透明的方法扩展网络设备和服务器的带宽、增加吞吐量、加强网络数据处理能力、提高网络的灵活性和可用性。

负载均衡，英文名称为 Load Balance，其意思就是分摊到多个操作单元上进行执行，例如 Web 服务器、FTP 服务器、企业关键应用服务器和其它关键任务服务器等，从而共同完成工作任务。

## **Nginx 实现负载均衡**

* Nginx 作为负载均衡服务器，用户请求先到达 Nginx，再由 Nginx 根据负载配置将请求转发至 Tomcat 服务器
* Nginx 负载均衡服务器：192.168.141.121:80
* Tomcat1 服务器：192.168.141.121:8081
* Tomcat2 服务器：192.168.141.121:8082
* 修改 /usr/local/docker/nginx/conf 目录下的 nginx.conf 配置文件：

1. user nginx;
2. worker\_processes 1;
3. events {
4. use epoll;
5. worker\_connections 1024;
6. }
7. http {
8. include mime.types;
9. default\_type application/octet-stream;
10. sendfile on;
11. keepalive\_timeout 65;
12. upstream myapp1 {
13. server 192.168.141.121:8081 weight=10;
14. server 192.168.141.121:8082 weight=10;
15. }
16. server {
17. listen 80;
18. server\_name nginx.funtl.com;
19. location / {
20. proxy\_pass http://myapp1;
21. index index.jsp index.html index.htm;
22. }
23. }
24. }

## **相关配置说明**

1. # 定义负载均衡设备的 Ip及设备状态
2. upstream myServer {
3. server 127.0.0.1:9090 down;
4. server 127.0.0.1:8080 weight=2;
5. server 127.0.0.1:6060;
6. server 127.0.0.1:7070 backup;
7. }

在需要使用负载的 Server 节点下添加

1. proxy\_pass http://myServer;

* upstream：每个设备的状态:
* down：表示当前的 server 暂时不参与负载
* weight：默认为 1 weight 越大，负载的权重就越大。
* max\_fails：允许请求失败的次数默认为 1 当超过最大次数时，返回 proxy\_next\_upstream 模块定义的错误
* fail\_timeout:max\_fails 次失败后，暂停的时间。
* backup：其它所有的非 backup 机器 down 或者忙的时候，请求 backup 机器。所以这台机器压力会最轻

# Nginx Ingress Controller

## **概述**

本次实践的主要目的就是将入口统一，不再通过 LoadBalancer 等方式将端口暴露出来，而是使用 Ingress 提供的反向代理负载均衡功能作为我们的唯一入口。通过以下步骤操作仔细体会。

****注意：**** 下面包含资源配置的步骤都是自行创建 YAML 配置文件通过 kubectl create -f <YAML> 或 kubectl apply -f <YAML> 部署，kubectl delete -f <YAML> 删除

## **安装 Ingress**

Ingress Controller 有许多种，我们选择最熟悉的 Nginx 来处理请求，其它可以参考 [官方文档](http://www.qfdmy.com/wp-content/themes/quanbaike/go.php?url=aHR0cHM6Ly9rdWJlcm5ldGVzLmlvL2RvY3MvY29uY2VwdHMvc2VydmljZXMtbmV0d29ya2luZy9pbmdyZXNzLWNvbnRyb2xsZXJzLw==" \t "http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/_blank)

* 下载 Nginx Ingress Controller 配置文件

wget https://raw.githubusercontent.com/kubernetes/ingress-nginx/master/deploy/static/mandatory.yaml

修改配置文件，找到配置如下位置 (搜索 serviceAccountName) 在下面增加一句 hostNetwork: true

1. apiVersion: apps/v1
2. kind: Deployment
3. metadata:
4. name: nginx-ingress-controller
5. namespace: ingress-nginx
6. labels:
7. app.kubernetes.io/name: ingress-nginx
8. app.kubernetes.io/part-of: ingress-nginx
9. spec:
10. # 可以部署多个实例
11. replicas: 1
12. selector:
13. matchLabels:
14. app.kubernetes.io/name: ingress-nginx
15. app.kubernetes.io/part-of: ingress-nginx
16. template:
17. metadata:
18. labels:
19. app.kubernetes.io/name: ingress-nginx
20. app.kubernetes.io/part-of: ingress-nginx
21. annotations:
22. prometheus.io/port: "10254"
23. prometheus.io/scrape: "true"
24. spec:
25. serviceAccountName: nginx-ingress-serviceaccount
26. # 增加 hostNetwork: true，意思是开启主机网络模式，暴露 Nginx 服务端口 80
27. hostNetwork: true
28. containers:
29. - name: nginx-ingress-controller
30. # 使用 Azure 中国镜像
31. image: quay.azk8s.cn/kubernetes-ingress-controller/nginx-ingress-controller:0.24.1
32. args:
33. - /nginx-ingress-controller
34. - --configmap=$(POD\_NAMESPACE)/nginx-configuration
35. - --tcp-services-configmap=$(POD\_NAMESPACE)/tcp-services
36. - --udp-services-configmap=$(POD\_NAMESPACE)/udp-services
37. - --publish-service=$(POD\_NAMESPACE)/ingress-nginx
38. // 以下代码省略...

* 通过命令 kubectl apply -f mandatory.yaml 部署
* 通过命令 kubectl get pods -n ingress-nginx -o wide 查看

## **部署 Ingress**

* 创建一个名为 ingress.yml 的资源配置文件

1. apiVersion: networking.k8s.io/v1beta1
2. kind: Ingress
3. metadata:
4. name: nginx-web
5. annotations:
6. # 指定 Ingress Controller 的类型
7. kubernetes.io/ingress.class: "nginx"
8. # 指定我们的 rules 的 path 可以使用正则表达式
9. nginx.ingress.kubernetes.io/use-regex: "true"
10. # 连接超时时间，默认为 5s
11. nginx.ingress.kubernetes.io/proxy-connect-timeout: "600"
12. # 后端服务器回转数据超时时间，默认为 60s
13. nginx.ingress.kubernetes.io/proxy-send-timeout: "600"
14. # 后端服务器响应超时时间，默认为 60s
15. nginx.ingress.kubernetes.io/proxy-read-timeout: "600"
16. # 客户端上传文件，最大大小，默认为 20m
17. nginx.ingress.kubernetes.io/proxy-body-size: "10m"
18. # URL 重写
19. nginx.ingress.kubernetes.io/rewrite-target: /
20. spec:
21. # 路由规则
22. rules:
23. # 主机名，只能是域名，修改为你自己的
24. - host: k8s.funtl.com
25. http:
26. paths:
27. - path:
28. backend:
29. # 后台部署的 Service Name
30. serviceName: tomcat-http
31. # 后台部署的 Service Port
32. servicePort: 8080

* 通过命令 kubectl apply -f ingress.yml 部署
* 通过命令 kubectl get ingress 查看

## **部署 Tomcat**

部署 Tomcat 但仅允许在内网访问，我们要通过 Ingress 提供的反向代理功能路由到 Tomcat 之上，创建一个名为 tomcat.yml 资源配置文件

1. apiVersion: extensions/v1beta1
2. kind: Deployment
3. metadata:
4. name: tomcat-app
5. spec:
6. replicas: 2
7. template:
8. metadata:
9. labels:
10. name: tomcat
11. spec:
12. containers:
13. - name: tomcat
14. image: tomcat:8.5.43
15. imagePullPolicy: IfNotPresent
16. ports:
17. - containerPort: 8080
18. ---
19. apiVersion: v1
20. kind: Service
21. metadata:
22. name: tomcat-http
23. spec:
24. ports:
25. - port: 8080
26. targetPort: 8080
27. # ClusterIP, NodePort, LoadBalancer
28. type: ClusterIP
29. selector:
30. name: tomcat

通过命令 kubectl apply -f tomcat.yml 部署

## **验证是否成功**

### **查看 Tomcat**

1. kubectl get deployment
2. # 输出如下
3. NAME READY UP-TO-DATE AVAILABLE AGE
4. tomcat-app 2/2 2 2 88m
5. kubectl get service
6. # 输出如下
7. NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
8. kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 2d5h
9. tomcat-http ClusterIP 10.97.222.179 <none> 8080/TCP 89m

### **查看 Ingress**

1. kubectl get pods -n ingress-nginx -o wide
2. # 输出如下，注意下面的 IP 地址，就是我们实际访问地址
3. NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
4. nginx-ingress-controller-76f9fddcf8-vzkm5 1/1 Running 0 61m 192.168.141.121 kubernetes-node-02 <none> <none>
5. kubectl get ingress
6. # 输出如下
7. NAME HOSTS ADDRESS PORTS AGE
8. nginx-web k8s.funtl.com 80 61m

### **测试访问**

成功代理到 Tomcat 即表示成功

1. # 不设置 Hosts 的方式请求地址，下面的 IP 和 HOST 均在上面有配置
2. curl -v http://192.168.141.121 -H 'host: k8s.funtl.com'

[Kubernetes 准备数据卷](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%87%86%e5%a4%87%e6%95%b0%e6%8d%ae%e5%8d%b7/)

### 评论 48

IMG_257

root@kubernetes-master:/usr/local/kubernetes/ingress# kubectl get pods -n ingress-nginx -o wide NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES nginx-ingress-controller-b75989d44-5hmbx 0/1 ErrImagePull 0 37m 192.168.1.211 kubernetes-node-01 不知道为啥出现了ErrImagePull，重新试一次一直卡在ContainerCreating

happyxianyu8个月前 (09-01)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_258

解决了更改了配置中的镜像 image: quay.azk8s.cn/kubernetes-ingress-controller/nginx-ingress-controller:0.25.1

happyxianyu8个月前 (09-01)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_259

换成什么了？我也是这个问题

changhe7个月前 (09-05)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_260

这个 registry.aliyuncs.com/google\_containers/nginx-ingress-controller:0.26.1

ycj1236个月前 (10-29)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

1. #2

IMG_261

quay.io不能拉取，替换为 quay.mirrors.ustc.edu.cn

gexc6个月前 (10-11)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

1. #3

IMG_262

Tomcat创建出现这个问题 error: unable to recognize "tomcat.yml": no matches for kind "Deployment" in version "extensions/v1beta1" 不知道咋回事，改成apps/v1 又出其他问题

hs20196个月前 (10-16)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_263

需要修改name 为 app

lishubin4个月前 (12-23)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

1. #4

IMG_264

error: unable to recognize "tomcat.yml": no matches for kind "Deployment" in version "extensions/v1beta1" v.1.16.0 版本的yaml 配置跟上面的又区别，查看博客https://www.funtl.com/zh/service-mesh-kubernetes/ 通过资源配置运行容器.html#创建-yaml-配置文件 ，更改即可

hs20196个月前 (10-16)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_265

感谢喔！按照你说的，成功解决！

beyond\_5206个月前 (10-17)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_266

谢谢大佬

35238839594个月前 (12-14)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

1. #5

IMG_267

按照步骤配置成功，访问域名 k8s.xx.com，提示下面的信息 -------------- 503 Service Temporarily Unavailable ------------------------ 老大遇到过吗？

kuailewangzi12126个月前 (10-18)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_268

我的版本是v1.16.0

kuailewangzi12126个月前 (10-18)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_269

tomcat 配置如下 -------------------- apiVersion: apps/v1 kind: Deployment metadata: name: tomcat-app spec: # 增加了选择器配置 selector: matchLabels: app: tomcat replicas: 2 template: metadata: labels: app: tomcat spec: containers: - name: tomcat image: tomcat:8.5.43 imagePullPolicy: IfNotPresent ports: - containerPort: 8080 --- apiVersion: v1 kind: Service metadata: name: tomcat-http spec: ports: - port: 8080 targetPort: 8080 # ClusterIP, NodePort, LoadBalancer type: ClusterIP selector: name: tomcat

kuailewangzi12126个月前 (10-18)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_270

我跟你出现的问题一样

cookiezzjie\_5206个月前 (10-26)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_271

问了李老师 解决了 tomcat.yml里面selector下面的app: tomcat必须一样 不能一个是name: tomcat一个是app: tomcat

cookiezzjie\_5206个月前 (10-27)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_272

一样问题.怎么解决的

god\_lqs5个月前 (11-12)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_273

将你之前的tomcat.yml部署和ingress.yml部署全部都删除 kubectl delete -f tomcat.yml和 kubectl delete -f ingress.yml，然后将tomcat.yml中发布部署的apiVersion的span下面增加selector: matchLabels: app: nginx，template的name改为app。最重要的是要将发布服务也就是apiVersion: v1 中的spec下面的selector的name改为app。然后在启动一遍 kubectl apply -f tomcat.yml和kubectl apply -f ingress.yml

35238839594个月前 (12-14)

1. #6

IMG_274

`kubectl get pods -n ingress-nginx -o wide` 获取的 `nginx-ingress-controller`，始终处于`ContainerCreating`状态，有木有小伙伴碰到，镜像地址改成azk8s或者utsc的都不行

ericshen5个月前 (11-05)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_275

是不是之前没有拉取过tomcat镜像，没拉过的话 拉取镜像应该就得一会功夫了

ruanxingke5个月前 (11-08)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_276

现在好了，替换成了阿里云的就行了

ericshen5个月前 (11-08)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_277

换的镜像发一下看看

liulong6664周前 (03-19)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_278

前缀替换为registry.cn-shenzhen.aliyuncs.com

ericshen4周前 (03-20)

IMG_279

看错了。。。我以为是tomcat。。。

ruanxingke5个月前 (11-08)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

1. #7

IMG_280

如果使用 wget https://raw.githubusercontent.com/kubernetes/ingress-nginx/master/deploy/static/mandatory.yaml 出现failed: Connection refused.又或者是拒绝连接 可以尝试wget -e robots=off https://raw.githubusercontent.com/kubernetes/ingress-nginx/master/deploy/static/mandatory.yaml 又或者是wget --no-check-certificate https://raw.githubusercontent.com/kubernetes/ingress-nginx/master/deploy/static/mandatory.yaml 就ok了，如果还不行，那就谷歌吧

35238839594个月前 (12-14)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_281

你说的可以

feilong3个月前 (01-14)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_282

可以的

chards\_92个月前 (02-16)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

1. #8

IMG_283

上面的那些镜像都不能用了，这个可以用：quay.mirrors.ustc.edu.cn/kubernetes-ingress-controller/nginx-ingress-controller:master

markenick3个月前 (01-21)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_284

用你这个解决 了，感谢。

abo19982个月前 (02-28)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

1. #9

IMG_285

如果访问k8s.funtl.com出现503，tomcat.yml可以用这个-----分割线----- apiVersion: apps/v1 kind: Deployment metadata: name: tomcat-app spec: # 增加了选择器配置 selector: matchLabels: app: tomcat replicas: 2 template: metadata: labels: app: tomcat spec: containers: - name: tomcat image: tomcat:8.5.43 imagePullPolicy: IfNotPresent ports: - containerPort: 8080 --- apiVersion: v1 kind: Service metadata: name: tomcat-http spec: ports: - port: 8080 targetPort: 8080 # ClusterIP, NodePort, LoadBalancer type: ClusterIP selector: app: tomcat

markenick3个月前 (01-22)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_286

你想说的是什么, 你贴的就添加了选择器 --- 本身就自带的啊

feilong2个月前 (02-04)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_287

如果k8s不是1.15的，部署Tomcat的tomcat.yml后访问k8s.funtl.com出现503，可以用我贴的那个配置

markenick2个月前 (02-04)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_288

我是你那个配置呀,但是还是503

feilong2个月前 (02-04)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_289

template: metadata: labels: name: tomcat还有最下面的 LoadBalancer type: ClusterIP selector: name: tomcat，把name换成app了吗?如果换了，你得考虑其他问题了

markenick2个月前 (02-04)

IMG_290

那两个地方的name要换成app

markenick2个月前 (02-04)

IMG_291

nacos路由你配置过了吗

feilong2个月前 (02-04)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_292

最下面name改成app好了, 你那贴的很容易让人忽视. 你价格#注解

feilong2个月前 (02-04)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_293

apiVersion: apps/v1 kind: Deployment metadata: name: tomcat-app spec: # 增加了选择器配置 selector: matchLabels: app: tomcat replicas: 2 template: # 下面的name要换成app metadata: labels: app: tomcat spec: containers: - name: tomcat image: tomcat:8.5.43 imagePullPolicy: IfNotPresent ports: - containerPort: 8080 --- apiVersion: v1 kind: Service metadata: name: tomcat-http spec: ports: - port: 8080 targetPort: 8080 # 下面的name要换成app ClusterIP, NodePort, LoadBalancer type: ClusterIP selector: app: tomcat

markenick2个月前 (02-04)

IMG_294

你实战的nacos集群路由 ingress配置有更改吗,那个不行

feilong2个月前 (02-05)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_295

部署NFS的时候 要加上这个selector: matchLabels: app: nfs-client-provisioner

markenick2个月前 (02-05)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_296

ingress配置没更改

markenick2个月前 (02-05)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_297

Nginx Ingress Controller 应该也要安装吧. 我装起来.网页没反应

feilong2个月前 (02-05)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

1. #10

IMG_298

如果是k8s1.7的话部署tomcat会报错，error: unable to recognize "tomcat.yml": no matches for kind "Deployment" in version "extensions/v1beta1" 因为1.6就不支持extensions/v1beta1将apiVersion改成apps/v1后面又会报错;error: error validating "test\_pod\_svc.yaml": error validating data: ValidationError(Deployment.spec): missing required field "selector" in io.k8s.api.apps.v1.DeploymentSpec; if you choose to ignore these errors, turn validation off with --validate=false; 可以参考一下这个博客http://www.mamicode.com/info-detail-2621292.html

asura2个月前 (02-09)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

1. #11

IMG_299

第一个命令下不赖配置文件的,试试这个:wget https://github.com/kubernetes/ingress-nginx/blob/master/deploy/static/mandatory.yaml

12478387002个月前 (02-27)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_300

非常感谢~！

jm\_qq8543777421个月前 (02-29)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

IMG_301

不对呀，兄弟

jm\_qq8543777421个月前 (02-29)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

1. #12

IMG_302

李哥 部署完之后 在同一局域网的linux环境中用curl 可以获取到tomcat的页面 但是在windows里浏览器就是nginx 404的页面 能讲解下吗

leonheart1个月前 (03-03)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

1. #13

IMG_303

extensions/v1beta1改为apps/v1

132501289193周前 (03-25)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

1. #14

IMG_304

报错 unexpected error storing fake SSL Cert: could not create PEM certificate file /etc/ingress-controller/ssl/default-fake-certificate.pem: open /etc/ingress-controller/ssl/default-fake-certificate.pem: permission denied 最后解决将runAsUser: 101改为runAsUser: 33

ty\_qfdmy2周前 (04-03)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/javascript:;)

# Kubernetes 准备数据卷

## **概述**

在 Docker 中就有数据卷的概念，当容器删除时，数据也一起会被删除，想要持久化使用数据，需要把主机上的目录挂载到 Docker 中去，在 K8S 中，数据卷是通过 Pod 实现持久化的，如果 Pod 删除，数据卷也会一起删除，k8s 的数据卷是 docker 数据卷的扩展，K8S 适配各种存储系统，包括本地存储 EmptyDir，HostPath， 网络存储（NFS，GlusterFS，PV/PVC）等。

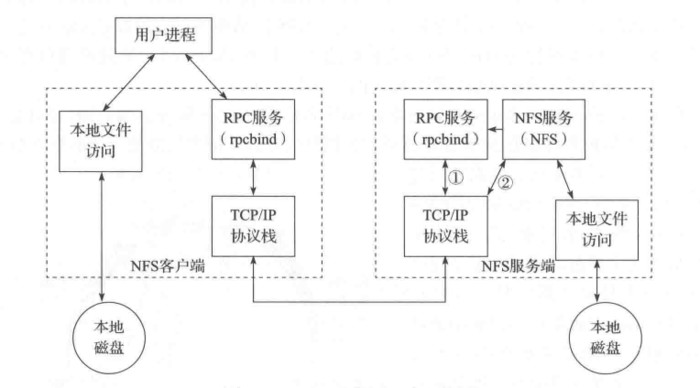
我们以部署 MySQL8 为例，采用 ****NFS + PV/PVC**** 网络存储方案实现我们的 Kubernetes 数据持久化。

## **什么是 NFS**

NFS 是 Network File System 的简写，即网络文件系统，NFS 是 FreeBSD 支持的文件系统中的一种。NFS 基于 RPC (Remote Procedure Call) 远程过程调用实现，其允许一个系统在网络上与它人共享目录和文件。通过使用 NFS，用户和程序就可以像访问本地文件一样访问远端系统上的文件。NFS 是一个非常稳定的，可移植的网络文件系统。具备可扩展和高性能等特性，达到了企业级应用质量标准。由于网络速度的增加和延迟的降低，NFS 系统一直是通过网络提供文件系统服务的有竞争力的选择 。

### **NFS 原理**

NFS 使用 RPC (Remote Procedure Call) 的机制进行实现，RPC 使得客户端可以调用服务端的函数。同时，由于有 VFS 的存在，客户端可以像使用其它普通文件系统一样使用 NFS 文件系统。经由操作系统的内核，将 NFS 文件系统的调用请求通过 TCP/IP 发送至服务端的 NFS 服务。NFS 服务器执行相关的操作，并将操作结果返回给客户端。



### **NFS 服务主要进程**

* rpc.nfsd：最主要的 NFS 进程，管理客户端是否可登录
* rpc.mountd：挂载和卸载 NFS 文件系统，包括权限管理
* rpc.lockd：非必要，管理文件锁，避免同时写出错
* rpc.statd：非必要，检查文件一致性，可修复文件

### **NFS 的关键工具**

* 主要配置文件：/etc/exports
* NFS 文件系统维护命令：/usr/bin/exportfs
* 共享资源的日志文件： /var/lib/nfs/\*tab
* 客户端查询共享资源命令： /usr/sbin/showmount
* 端口配置： /etc/sysconfig/nfs

### **NFS 服务端配置**

在 NFS 服务器端的主要配置文件为 /etc/exports 时，通过此配置文件可以设置共享文件目录。每条配置记录由 NFS 共享目录、NFS 客户端地址和参数这 3 部分组成，格式如下：

1. [NFS 共享目录] [NFS 客户端地址 1 (参数 1, 参数 2, 参数 3……)] [客户端地址 2 (参数 1, 参数 2, 参数 3……)]

* NFS 共享目录：服务器上共享出去的文件目录
* NFS 客户端地址：允许其访问的 NFS 服务器的客户端地址，可以是客户端 IP 地址，也可以是一个网段 (192.168.141.0/24)
* 访问参数：括号中逗号分隔项，主要是一些权限选项

#### **访问权限参数**

| **序号** | **选项** | **描述** |
| --- | --- | --- |
| 1 | ro | 客户端对于共享文件目录为只读权限。默认 |
| 2 | rw | 客户端对于共享文件目录具有读写权限 |

#### **用户映射参数**

| **序号** | **选项** | **描述** |
| --- | --- | --- |
| 1 | root\_squash | 使客户端使用 root 账户访冋时，服务器映射为服务器本地的匿名账号 |
| 2 | no\_root\_squash | 客户端连接服务端时如果使用的是 root，那么也拥有对服务端分享的目录的 root 权限 |
| 3 | all\_squash | 将所有客户端用户请求映射到匿名用户或用户组（nfsnobody) |
| 4 | no\_all\_squash | 与上相反。默认 |
| 5 | anonuid=xxx | 将远程访问的所有用户都映射为匿名用户，并指定该用户为本地用户(UID=xxx) |
| 6 | anongid=xxx | 将远程访问的所有用户组都映射为匿名用户组账户，并指定该匿名用户组账户为本地用户组账户(GUI=xxx) |

#### **其它配置参数**

| **序号** | **选项** | **描述** |
| --- | --- | --- |
| 1 | sync | 同步写操作，数据写入存储设备后返回成功信息。默认 |
| 2 | async | 异步写提作，数据在未完全写入存储设备前就返回成功信息，实际还在内存， |
| 3 | wdelay | 延迟写入选项，将多个写提请求合并后写入硬盘，减少 I/O 次数， NFS 非正常关闭数据可能丢失。默认 |
| 4 | no\_wdelay | 与上相反，不与 async 同时生效，如果 NFS 服务器主要收到小且不相关的请求，该选项实际会降低性能 |
| 5 | subtree | 若输出目录是一个子目录，则 NFS 服务器将检查其父目录的权限。默认 |
| 6 | no\_subtree | 即使输出目录是一个子目录， NFS 服务器也不检查其父目录的权限，这样可以提高效率 |
| 7 | secure | 限制客户端只能从小于 1024 的 TCP/IP 端口连接 NFS 服务器。默认 |
| 8 | insecure | 允许客户端从大于 1024 的 TCP/IP 端口连接服务器 |

## **安装 NFS 服务端**

由于 NFS 是一套分布式文件系统，我们再创建一台独立的虚拟机作为我们 NFS 服务端，配置如下

| **主机名** | **IP** | **系统** | **CPU/内存** | **磁盘** |
| --- | --- | --- | --- | --- |
| kubernetes-volumes | 192.168.141.130 | Ubuntu Server 18.04 | 2 核 2G | 20G |

* 创建一个目录作为共享文件目录

1. mkdir -p /usr/local/kubernetes/volumes

* 给目录增加读写权限

1. chmod a+rw /usr/local/kubernetes/volumes

* 安装 NFS 服务端

1. apt-get update
2. apt-get install -y nfs-kernel-server

* 配置 NFS 服务目录，打开文件 vi /etc/exports，在尾部新增一行，内容如下
  + **/usr/local/kubernetes/volumes**：**** 作为服务目录向客户端开放
  + \*****：****表示任何 IP 都可以访问
  + ****rw：**** 读写权限
  + ****sync：**** 同步权限
  + ****no\_subtree\_check：**** 表示如果输出目录是一个子目录，NFS 服务器不检查其父目录的权限
  + ****no\_root\_squash：**** 客户端连接服务端时如果使用的是 root，那么也拥有对服务端分享的目录的 root 权限

1. /usr/local/kubernetes/volumes \*(rw,sync,no\_subtree\_check,no\_root\_squash)

* 重启服务，使配置生效

1. /etc/init.d/nfs-kernel-server restart

## **安装 NFS 客户端**

安装客户端的目的是验证是否可以上传文件到服务端，安装命令如下

1. apt-get install -y nfs-common

* 创建 NFS 客户端挂载目录

1. mkdir -p /usr/local/kubernetes/volumes-mount

* 将 NFS 服务器的 /usr/local/kubernetes/volumes 目录挂载到 NFS 客户端的 /usr/local/kubernetes/volumes-mount 目录

1. mount 192.168.141.130:/usr/local/kubernetes/volumes /usr/local/kubernetes/volumes-mount

* 使用 df 命令查看挂载信息

1. df
2. # 输出如下
3. Filesystem 1K-blocks Used Available Use% Mounted on
4. udev 977544 0 977544 0% /dev
5. tmpfs 201732 1232 200500 1% /run
6. /dev/mapper/ubuntu--vg-ubuntu--lv 19475088 4780912 13681852 26% /
7. tmpfs 1008648 0 1008648 0% /dev/shm
8. tmpfs 5120 0 5120 0% /run/lock
9. tmpfs 1008648 0 1008648 0% /sys/fs/cgroup
10. /dev/loop0 90624 90624 0 100% /snap/core/7270
11. /dev/loop1 93184 93184 0 100% /snap/core/6350
12. /dev/sda2 999320 77944 852564 9% /boot
13. tmpfs 201728 0 201728 0% /run/user/0
14. # 有此输出表示挂载成功
15. 192.168.141.130:/usr/local/kubernetes/volumes 19475200 4780800 13681920 26% /usr/local/kubernetes/volumes-mount

## **验证 NFS 服务**

* 测试文件上传

1. ip addr > /usr/local/kubernetes/volumes-mount/test.txt

* 查看 /usr/local/kubernetes/volumes 目录下是否有 test.txt 文件，有则表示成功

## **取消 NFS 客户端挂载**

****注意：**** 不要直接在挂载目录下执行，否则会报错

1. umount /usr/local/kubernetes/volumes-mount

窗体顶端

完成

窗体底端

**上一页**[Nginx Ingress Controller](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/nginx-ingress-controller/)

**下一个**[Kubernetes 使用数据卷](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e4%bd%bf%e7%94%a8%e6%95%b0%e6%8d%ae%e5%8d%b7/)

### 评论 4

IMG_258

ubuntu 19 在安装nfs-kernel-server的时候出现依赖问题，客户端和服务端版本相同，总是在nfs-common这里出现问题，换了很多下载源安装nfs-kernel-server都出现同样的问题，有谁遇到同样的问题？能分享你的解决方案吗？

lingboshuiyue7个月前 (09-19)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%87%86%e5%a4%87%e6%95%b0%e6%8d%ae%e5%8d%b7/javascript:;)

IMG_259

在终端输入下面命令强制解锁 sudo rm /var/cache/apt/archives/lock sudo rm /var/lib/dpkg/lock

hanhua4个月前 (12-24)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%87%86%e5%a4%87%e6%95%b0%e6%8d%ae%e5%8d%b7/javascript:;)

1. #2

IMG_260

报错了Could not get lock /var/lib/dpkg/lock-frontend - open (11: Resource temporarily unavailable) Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), is another process using it?

hanhua4个月前 (12-24)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%87%86%e5%a4%87%e6%95%b0%e6%8d%ae%e5%8d%b7/javascript:;)

IMG_261

我也是

chards\_92个月前 (02-16)[回复](http://www.qfdmy.com/%E8%AF%BE%E7%A8%8B/%e5%be%ae%e6%9c%8d%e5%8a%a1%e6%9e%b6%e6%9e%84-2-0/lessons/kubernetes-%e5%87%86%e5%a4%87%e6%95%b0%e6%8d%ae%e5%8d%b7/javascript:;)