

破茧-DevOps-K8S集群搭建

🕒 分类

待处理

🏷️ 属性

未填写

Kubernetes（标红的内容需要学习，标蓝内容需要重点了解）

Kubernetes (K8S) 是什么

它是一个为 **容器化** 应用提供集群部署和管理的开源工具，由 Google 开发。
Kubernetes 这个名字源于希腊语，意为“舵手”或“飞行员”。k8s 这个缩写是因为 k 和 s 之间有八个字符的关系。Google 在 2014 年开源了 Kubernetes 项目

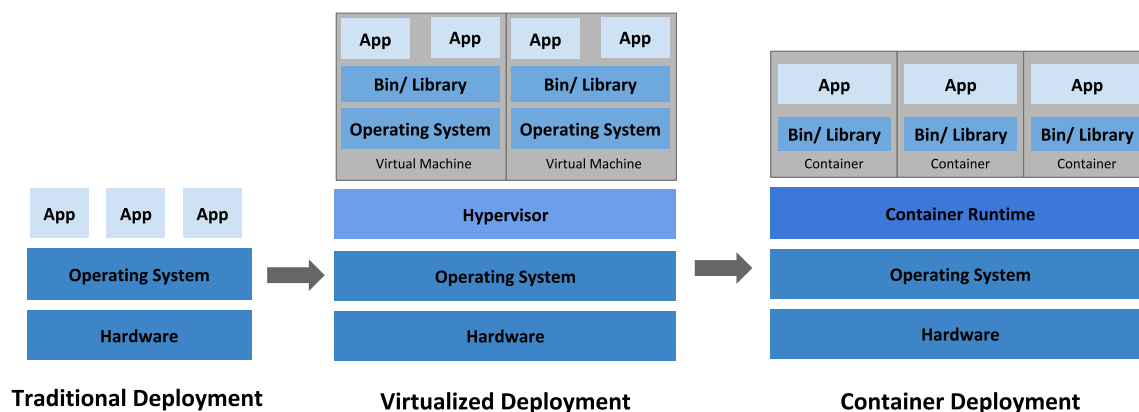
主要特性：

- 高可用，不宕机，自动灾难恢复
- 灰度更新，不影响业务正常运转
- 一键回滚到历史版本
- 方便的伸缩扩展（应用伸缩，机器加减）、提供负载均衡
- 有一个完善的生态

学习课程前提

熟悉 Docker 的基本使用，如果你还不了解 Docker，先看视频 [Docker 快速上手](#)
熟悉 Linux 操作系统

不同的应用部署方案



传统部署方式：

应用直接在物理机上部署，机器资源分配不好控制，出现Bug时，可能机器的大部分资源被某个应用占用，导致其他应用无法正常运行，无法做到应用隔离。

虚拟机部署

在单个物理机上运行多个虚拟机，每个虚拟机都是完整独立的系统，性能损耗大。

容器部署

所有容器共享主机的系统，轻量级的虚拟机，性能损耗小，资源隔离，CPU和内存可按需分配

什么时候需要 Kubernetes

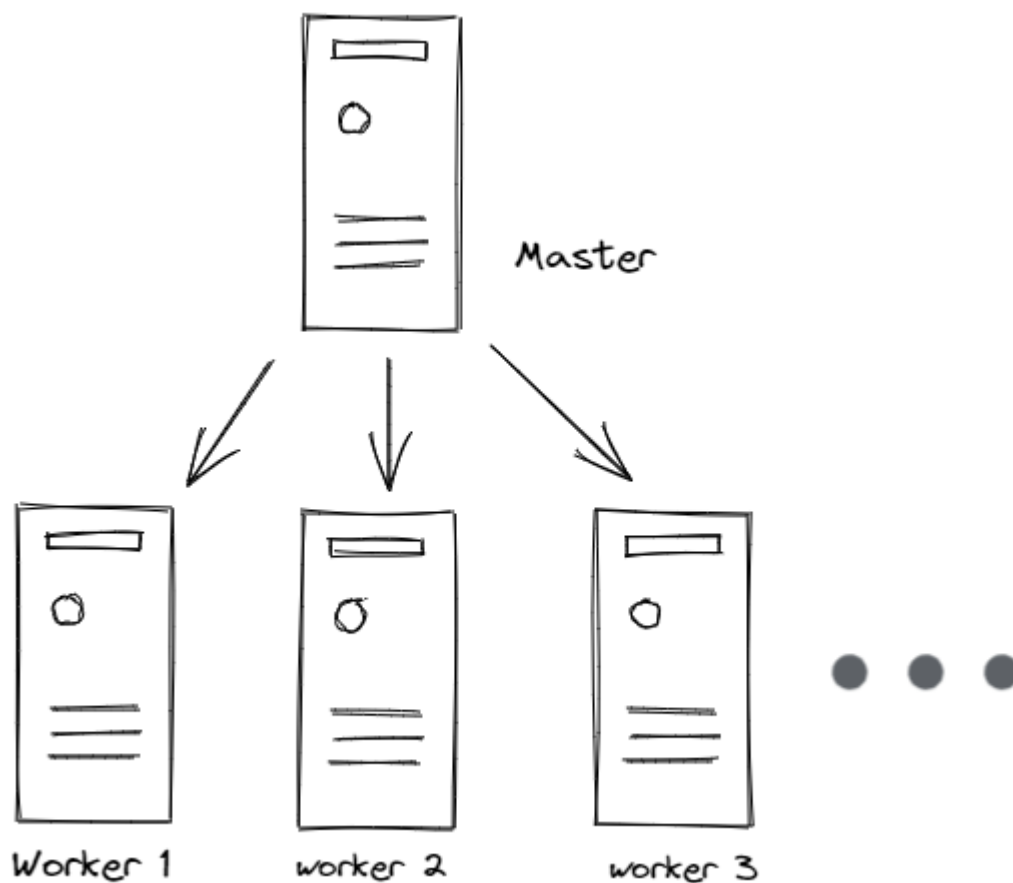
当你的应用只是跑在一台机器，直接一个 docker + docker-compose 就够了，方便轻松；

当你的应用需要跑在 3，4 台机器上，你依旧可以每台机器单独配置运行环境 + 负载均衡器；

当你应用访问数不断增加，机器逐渐增加到十几台、上百台、上千台时，每次加机器、软件更新、版本回滚，都会变得非常麻烦。

Kubernetes 可以为你提供集中式的管理集群机器和应用，加机器、版本升级、版本回滚，那都是一个命令就搞定的事，不停机的灰度更新，确保高可用、高性能、高扩展。

Kubernetes 集群架构



master

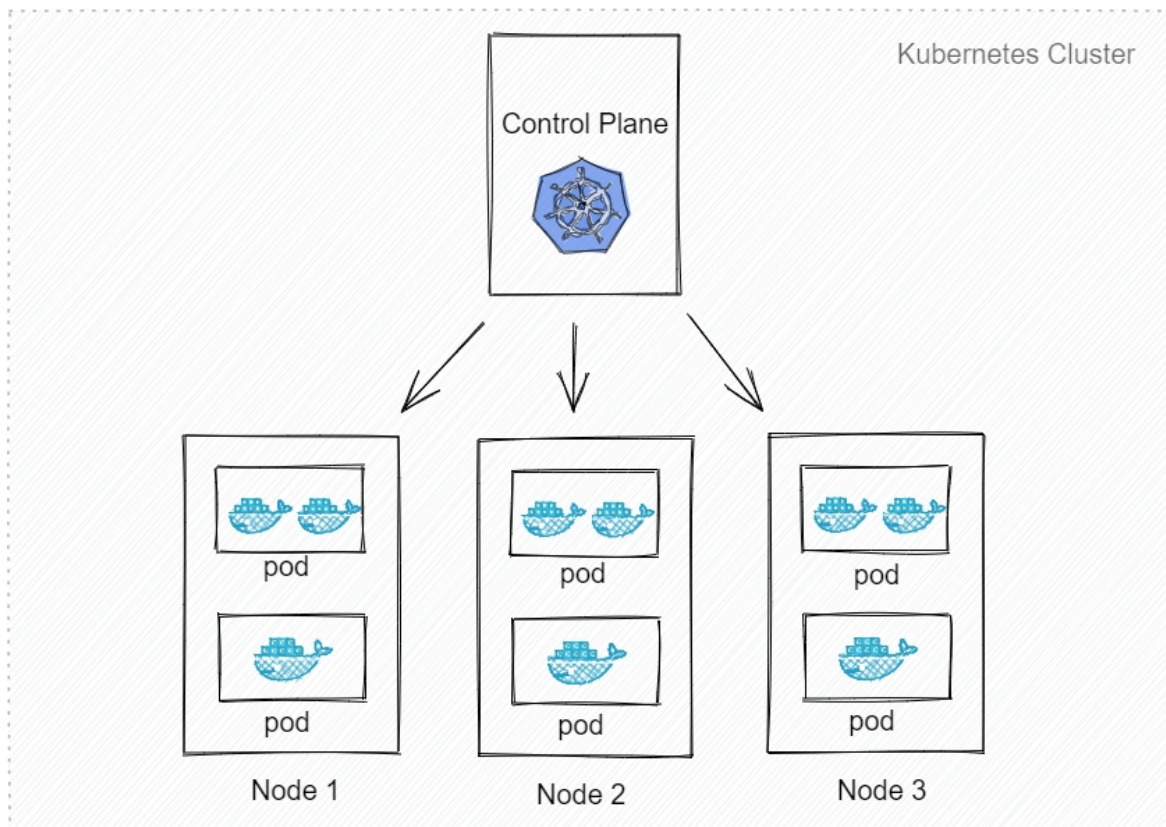
主节点，控制平台，不需要很高性能，不跑任务，通常一个就行了，也可以开多个主节点来提高集群可用度。

worker

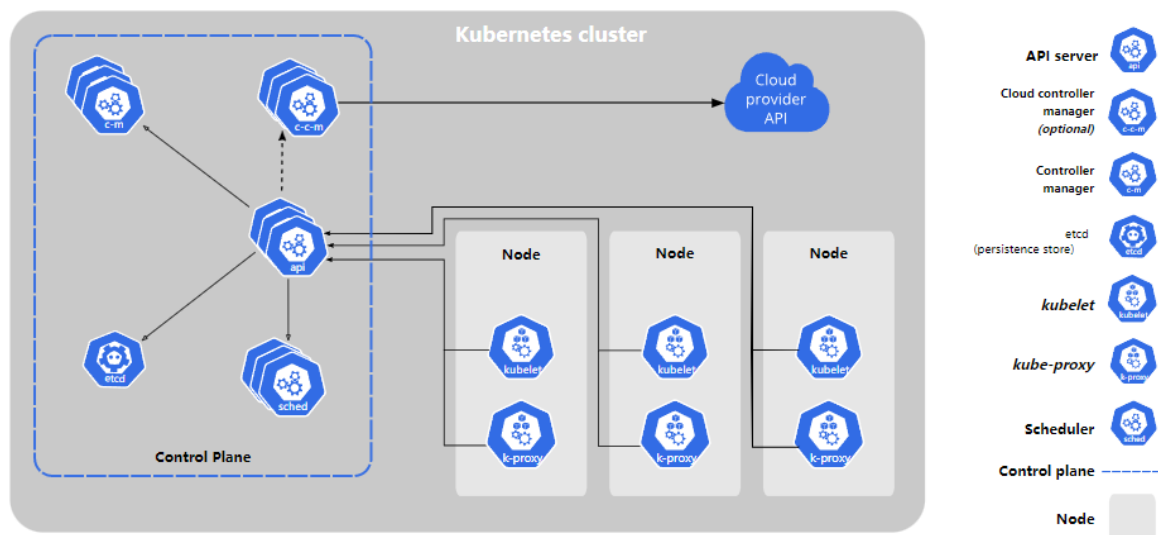
工作节点，可以是虚拟机或物理计算机，任务都在这里跑，机器性能需要好点；通常都有很多个，可以不断加机器扩大集群；每个工作节点由主节点管理

重要概念 Pod

豆荚，K8S 调度、管理的最小单位，一个 Pod 可以包含一个或多个容器，每个 Pod 有自己的虚拟IP。一个工作节点可以有多个 pod，主节点会考量负载自动调度 pod 到哪个节点运行。



Kubernetes 组件



如果你想要了解更多 K8S 组成细节，主节点、工作节点分别有哪些程序，各有什么作用，可以查看 [官网详细介绍](#)

安装 Kubernetes 集群

工具介绍

- **集群命令行工具(kubectl)**

Kubernetes 提供 kubectl 是使用 Kubernetes API 与 Kubernetes 集群的[控制面](#)进行通信的命令行工具，[官网详细介绍](#)

安装方式介绍

- **minikube**

只是一个 K8S 集群模拟器，只有一个节点的集群，只为测试用，master 和 worker 都在一起

- **直接用云平台 Kubernetes**

可视化搭建，只需简单几步就可以创建好一个集群。

优点：安装简单，生态齐全，负载均衡器、存储等都给你配套好，简单操作就搞定

- **裸机安装 (Bare Metal)**

至少需要两台机器（主节点、工作节点个一台），需要自己安装 Kubernetes 组件，配置会稍微麻烦点。

可以到各云厂商按时租用服务器，费用低，用完就销毁。

缺点：配置麻烦，缺少生态支持，例如负载均衡器、云存储。

minikube

安装非常简单，支持各种平台，[安装方法](#)

需要提前安装好 Docker

▼

```
1 # 启动集群
2 minikube start
3 # 查看节点。kubectl 是一个用来跟 K8S 集群进行交互的命令行工具
4 kubectl get node
5 # 停止集群
6 minikube stop
7 # 清空集群
8 minikube delete --all
9 # 安装集群可视化 Web UI 控制台
10 minikube dashboard
```

云平台搭建

- [腾讯云 TKE](#)（控制台搜索容器）
- 登录阿里云控制台 - 产品搜索 Kubernetes

裸机搭建

主节点需要组件

- docker（也可以是其他容器运行时）
- kubectl 集群命令行交互工具
- kubeadm 集群初始化工具

工作节点需要组件 [文档](#)

- docker（也可以是其他容器运行时）
- kubelet 管理 Pod 和容器，确保他们健康稳定运行。
- kube-proxy 网络代理，负责网络相关的工作

开始安装

你也可以试下 [这个项目](#)，用脚本快速搭建 K8S 裸机集群
当然，为了更好的理解，你应该先手动搭建一次

▼

```
1 # 每个节点分别设置对应主机名
2 hostnamectl set-hostname master
3 hostnamectl set-hostname node1
4 hostnamectl set-hostname node2
```

▼

```
1 # 所有节点都修改 hosts
2 vim /etc/hosts
3 172.16.32.2 node1
4 172.16.32.6 node2
5 172.16.0.4 master
```

▼

```
1 # 所有节点关闭 SELinux
```

```
2 setenforce 0
3 sed -i --follow-symlinks 's/SELINUX=enforcing/SELINUX=disabled/g'
   /etc/sysconfig/selinux
```

所有节点确保防火墙关闭

```
systemctl stop firewalld
```

```
systemctl disable firewalld
```

添加安装源（所有节点）

```
1 # 添加 k8s 安装源
2 cat <<EOF > kubernetes.repo
3 [kubernetes]
4 name=Kubernetes
5 baseurl=https://mirrors.aliyun.com/kubernetes/yum/repos/kubernetes-
  el7-x86_64
6 enabled=1
7 gpgcheck=0
8 repo_gpgcheck=0
9 gpgkey=https://mirrors.aliyun.com/kubernetes/yum/doc/yum-key.gpg
  https://mirrors.aliyun.com/kubernetes/yum/doc/rpm-package-key.gpg
10 EOF
11 mv kubernetes.repo /etc/yum.repos.d/
12
13 # 添加 Docker 安装源
14 yum-config-manager --add-repo http://mirrors.aliyun.com/docker-
  ce/linux/centos/docker-ce.repo
```

安装所需组件（所有节点）

```
yum install -y kubelet-1.22.4 kubect-1.22.4 kubeadm-1.22.4 docker-ce
```

启动 kubelet、docker，并设置开机启动（所有节点）

```
1 systemctl enable kubelet
2 systemctl start kubelet
3 systemctl enable docker
4 systemctl start docker
```

修改 docker 配置（所有节点）

```
1 # kubernetes 官方推荐 docker 等使用 systemd 作为 cgroupdriver, 否则
  kubelet 启动不了
2 cat <<EOF > daemon.json
3 {
4   "exec-opts": ["native.cgroupdriver=systemd"],
5   "registry-mirrors": ["https://ud6340vz.mirror.aliyuncs.com"]
6 }
7 EOF
8 mv daemon.json /etc/docker/
9
10 # 重启生效
11 systemctl daemon-reload
12 systemctl restart docker
```

用 kubeadm 初始化集群（仅在主节点跑），

```
1 # 初始化集群控制台 Control plane
2 # 失败了可以用 kubeadm reset 重置
3 kubeadm init --image-
  repository=registry.aliyuncs.com/google_containers
4
5 # 记得把 kubeadm join xxx 保存起来
6 # 忘记了重新获取: kubeadm token create --print-join-command
7
8 # 复制授权文件, 以便 kubectl 可以有权限访问集群
9 # 如果你其他节点需要访问集群, 需要从主节点复制这个文件过去其他节点
10 mkdir -p $HOME/.kube
11 cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
12 chown $(id -u):$(id -g) $HOME/.kube/config
13
14 # 在其他机器上创建 ~/.kube/config 文件也能通过 kubectl 访问到集群
```

有兴趣了解 kubeadm init 具体做了什么的, 可以 [查看文档](#)

把工作节点加入集群（只在工作节点跑）


```
1 kubeadm join 172.16.32.10:6443 --token xxx --discovery-token-ca-  
cert-hash xxx
```

安装网络插件，否则 node 是 NotReady 状态（主节点跑）

```
1 # 很有可能国内网络访问不到这个资源，你可以网上找找国内的源安装 flannel  
2 kubectl apply -f  
  https://raw.githubusercontent.com/coreos/flannel/master/Documentation/  
  flannel.yml  
3  
4 # 如果上面的插件安装失败，可以选用 Weave，下面的命令二选一就可以了。  
5 kubectl apply -f  
  https://github.com/weaveworks/weave/releases/download/v2.8.1/weave-  
  daemonset-k8s.yaml  
6 kubectl apply -f http://static.corecore.cn/weave.v2.8.1.yaml  
7  
8 # 更多其他网路插件查看下面介绍，自行网上找 yaml 安装  
9 https://blog.csdn.net/ChaITSimpleLove/article/details/117809007
```

请注意，flannel 插件很多学员都安装不上（因为网络问题），这样会导致后面名字访问服务用不了，

大家可以改用 weave 或者使用香港服务器或者寻找国内的源安装

```
[root@master mongo]# kubectl get nodes  
NAME      STATUS    ROLES    AGE     VERSION  
master    Ready     control-plane,master  3h46m   v1.22.4  
node1     Ready     <none>    3h40m   v1.22.4  
node2     Ready     <none>    3h40m   v1.22.4  
[root@master mongo]#
```

学习内容参考链接：

- 集群命令行工具(kubectl)

<https://kubernetes.io/zh-cn/docs/reference/kubectl/>

Kubernetes in Action - 2.2.3

- Minikube搭建

<https://kubernetes.io/zh-cn/docs/tutorials/kubernetes-basics/create-cluster/cluster-intro/>

<https://minikube.sigs.k8s.io/docs/start/>

Kubernetes in Action - 2.2.1

- Kubeadm搭建

<https://kubernetes.io/zh-cn/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>

<https://cloud.tencent.com/developer/article/2160663>

<https://cloud.tencent.com/developer/article/2151843>

达成目标：

- 了解Kubernetes是什么、架构模式、重要组件有哪些
- 掌握命令行工具(kubectl)-常用命令及作用
- 使用Minikube搭建集群，并在集群中使用kubectl常用命令
- 了解如何使用二进制搭建集群
- 完成Kubernetes in Action第二章节内容阅读