

# 一、基本操作

- 查看节点

```
[root@zhaowa-edu-01 ~]# kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
zhaowa-edu-01	Ready	control-plane,master	21h	v1.20.2

## 【解释】

node是用于承载运行中的容器的。

有一个node，这个node的角色是master。

k8s也有集群的概念，即：cluster，包含master和node，该节点即是master节点，又是node节点。

- 创建nginx的deployment

```
[root@zhaowa-edu-01 ~]# kubectl create deployment my-nginx --image nginx:latest
deployment.apps/my-nginx created
```

## 【解释】

创建deployment其实有两种方式

1> 基于命令行配置运行

- kubectl create
- kubectl run

2> 基于yaml配置文件的

- kubectl apply -f xxx.yml

- 查看所有pod信息以及ip和port

```
[root@zhaowa-edu-01 ~]# kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE
	NOMINATED	NODE	READINESS	GATES		
my-nginx-b7d7bc74d-jgfkx	1/1	Running	0	7m35s	172.18.0.3	
zhaowa-edu-01	<none>		<none>			

- 查看所有service

```
[root@zhaowa-edu-01 ~]# kubectl get service
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	21h

## 【解释】

- pod是不稳定的。服务可以保证稳定。
- 双十一之前，订单系统需要20个服务实例，双十一促销阶段，扩容至100个服务实例。双十一之后，对服务缩容至30个服务实例。
- 我们是一个稳定的请求方式，统一一个ip的。这个就是service的作用。

- 查询所有命名空间

```
[root@zhaowa-edu-01 ~]# kubectl get namespace
```

NAME	STATUS	AGE
default	Active	21h
kube-node-lease	Active	21h
kube-public	Active	21h
kube-system	Active	21h

【解释】

默认的就是default。

其它kube-\*的都是k8s系统自己的命名空间。

- 将副本数从1个修改为3个

```
[root@zhaowa-edu-01 ~]# kubectl scale deployments/my-nginx --replicas=3
deployment.apps/my-nginx scaled
[root@zhaowa-edu-01 ~]# kubectl get deploy
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
my-nginx	1/3	3	1	19m

```
[root@zhaowa-edu-01 ~]# kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
my-nginx-b7d7bc74d-jgfkx	1/1	Running	0	19m
my-nginx-b7d7bc74d-r9gsg	0/1	ContainerCreating	0	17s
my-nginx-b7d7bc74d-vr958	0/1	ContainerCreating	0	17s

【解释】

- 什么是副本个数？  
pod的个数
- 如果我们不指定副本个数的话，那么默认就是一个pod

- 将副本数从3个修改为2个

```
[root@zhaowa-edu-01 ~]# kubectl scale deployments/my-nginx --replicas=2
deployment.apps/my-nginx scaled
[root@zhaowa-edu-01 ~]# kubectl get deploy
NAME          READY    UP-TO-DATE    AVAILABLE    AGE
my-nginx      2/2      2             2            22m
[root@zhaowa-edu-01 ~]# kubectl get pod
NAME                                READY    STATUS    RESTARTS    AGE
my-nginx-b7d7bc74d-jgfkx           1/1      Running   0           22m
my-nginx-b7d7bc74d-r9gsg           1/1      Running   0           2m39s
```

## 二、重要概念

- Cluster

资源的集合。我们的k8s利用这些资源运行各种基于容器的应用。

- Master

Cluster的大脑。司令部。主要的任务就是用来调度的。决定我们的应用应该放到哪里去执行。  
为了高可用，也可以运行多个master。

- Node

来负责运行容器应用。  
Node是由Master去管理的，负责监控和容器状态的汇报。

- Pod

是k8s的最小的工作单元。包含1orN个容器。

Pod的使用方式：

- 一个pod运行一个容器  
最常用的就是这种情况。
- 一个pod运行多个容器  
一定是非常紧密相关的一组容器，并且需要资源的共享。一起启动、一起停止。

- Controller

k8s通过它来管理Pod

针对不同的业务场景，k8s提供了多种Controller，其中包含：Deployment、ReplicaSet、DaemonSet、StatefulSet、Job

- Deployment

就是我们最常用的Controller。它可以管理Pod的多个副本。（即：--replicas=3），并且可以确保Pod按照期望的状态去运行。

- ReplicaSet

它也是管理Pod的多个副本。

我们使用deployment的时候，会自动的创建ReplicaSet，最终是有ReplicaSet去创建的pod，而我们并不是去直接的使用它。

- DaemonSet

用于每个Node最多只运行一个Pod副本的创建。

- StatefulSet

保证副本按照固定的顺序启动、更新、删除。

- Service

为Pod提供了负载均衡、固定的IP和Port

pod是不稳定的，ip会变化的。所以我们需要一个固定的ip或port。

区别：

- Controller ——> 负责k8s运行容器的。
- Service ——> 负责k8s访问容器的。

- Namespace

资源隔离！

## 三、k8s架构

---

### 3.1> master

- API-Server

属于前端交互接口。提供基于Http/https RESTful API。

接收对应的指令。

- Scheduler

负责决定将pod放到那个Node上去运行的。

- Controller Manager

非常关键的组件。管理Cluster中的各种资源。

- etcd

负责保存k8s的配置信息和各种资源的状态信息。

如果数据发生了变化，etcd会快速通知相关的组件。

## 3.2> Node

- kubelet

创建和运行容器。

- kube-proxy

负责我们请求的转发。

如果对于多个副本，它会实现负载均衡。

## 四、label

```
[root@zhaowa-edu-01 k8sConfigFiles]# kubectl get node
NAME                STATUS    ROLES                  AGE      VERSION
zhaowa-edu-01       Ready     control-plane,master   22h      v1.20.2
[root@zhaowa-edu-01 k8sConfigFiles]# kubectl get node --show-labels
NAME                STATUS    ROLES                  AGE      VERSION    LABELS
zhaowa-edu-01       Ready     control-plane,master   22h      v1.20.2
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kube
rnetes.io/hostname=zhaowa-edu-
01,kubernetes.io/os=linux,minikube.k8s.io/commit=b017ea15ffbf8bcd6ce31e13ba16f59fd40910
79,minikube.k8s.io/name=minikube,minikube.k8s.io/updated_at=2021_09_18T13_04_08_0700,mi
nikube.k8s.io/version=v1.20.0,node-role.kubernetes.io/control-plane=,node-
role.kubernetes.io/master=
[root@zhaowa-edu-01 k8sConfigFiles]# kubectl label node zhaowa-edu-01 disktype=ssd
node/zhaowa-edu-01 labeled
[root@zhaowa-edu-01 k8sConfigFiles]# kubectl get node --show-labels
NAME                STATUS    ROLES                  AGE      VERSION    LABELS
zhaowa-edu-01       Ready     control-plane,master   22h      v1.20.2
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,disktype=ssd,kubernetes.io/ar
ch=amd64,kubernetes.io/hostname=zhaowa-edu-
01,kubernetes.io/os=linux,minikube.k8s.io/commit=b017ea15ffbf8bcd6ce31e13ba16f59fd40910
79,minikube.k8s.io/name=minikube,minikube.k8s.io/updated_at=2021_09_18T13_04_08_0700,mi
nikube.k8s.io/version=v1.20.0,node-role.kubernetes.io/control-plane=,node-
role.kubernetes.io/master=
[root@zhaowa-edu-01 k8sConfigFiles]# kubectl label node zhaowa-edu-01 disktype-
node/zhaowa-edu-01 labeled
[root@zhaowa-edu-01 k8sConfigFiles]# kubectl get node --show-labels
NAME                STATUS    ROLES                  AGE      VERSION    LABELS
zhaowa-edu-01       Ready     control-plane,master   22h      v1.20.2
beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kube
rnetes.io/hostname=zhaowa-edu-
01,kubernetes.io/os=linux,minikube.k8s.io/commit=b017ea15ffbf8bcd6ce31e13ba16f59fd40910
79,minikube.k8s.io/name=minikube,minikube.k8s.io/updated_at=2021_09_18T13_04_08_0700,mi
nikube.k8s.io/version=v1.20.0,node-role.kubernetes.io/control-plane=,node-
role.kubernetes.io/master=
```

