

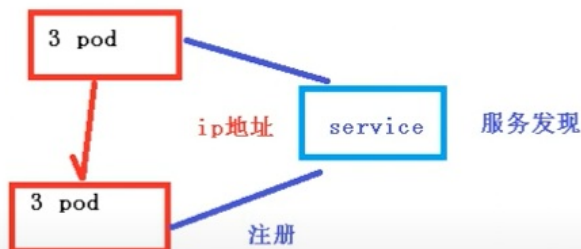
Service 基本概念

服務發現

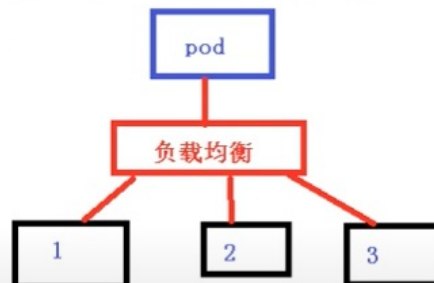
Service

✦ 1、service存在意义

(1) 防止Pod失联 (服务发现)

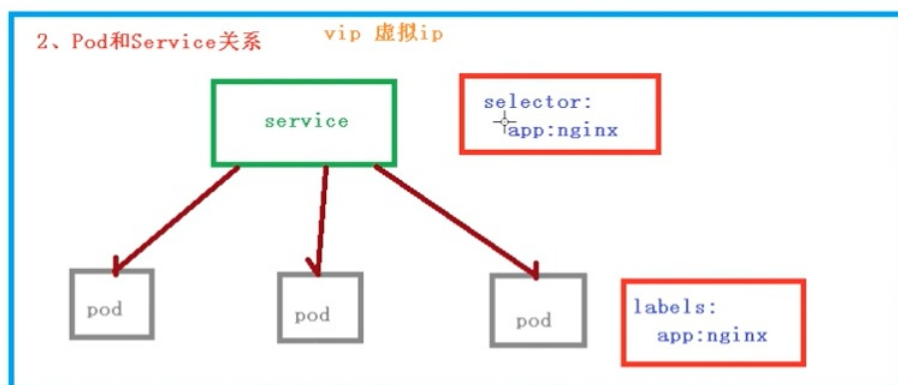


2、定义一组Pod访问策略 (负载均衡)



2、Pod和Service关系

vip 虚拟ip



根据label和selector
标签建立关联的

通过service实现Pod的负载均衡

以下是圖片內容的文字辨識及繁體中文翻譯：

1. 無狀態和有狀態

(1) 無狀態：

- 認為 **Pod** 都是一樣的
- 沒有順序要求
- 不用考慮在哪個 **Node** 運行
- 可以隨意進行伸縮和擴展

(2) 有狀態：

- 上述因素都需要考慮到
- 讓每個 **Pod** 獨立，保持 **Pod** 啟動順序和唯一性
- 唯一的網路標識符，持久存儲
- 有順序，例如 **MySQL** 主從

2. 部署有狀態應用

(1) 無頭 Service

- **ClusterIP: None**

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  ports:
    - port: 80
      name: web
  clusterIP: None
  selector:
    app: nginx
```

(2) StatefulSet 部署有狀態應用

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: nginx-statefulset
  namespace: default
```

指令：

```
[root@k8smaster ~]# vi sts.yaml
[root@k8smaster ~]# kubectl apply -f sts.yaml
service/nginx created
statefulset.apps/nginx-statefulset created
```

說明：

- **無頭 Service**：ClusterIP: None 用於有狀態應用，允許 Pod 之間直接通訊。
- **StatefulSet**：適用於有狀態應用，確保每個 Pod 都有唯一標識，並保證啟動順序。

以下是圖片內容的文字辨識及繁體中文翻譯：

查看 Pod

- 有三個 Pod，每個都有唯一名稱：

```
[root@k8smaster ~]# kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
nginx-statefulset-0  1/1     Running   0           74s
nginx-statefulset-1  1/1     Running   0           40s
nginx-statefulset-2  1/1     Running   0           21s
```

Deployment 和 StatefulSet 的區別

- StatefulSet 有身份的（唯一標識的）。
- 根據主機名 + 按照一定規則生成域名。

查看創建的無頭 Service

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	6d17h
nginx	ClusterIP	None	<none>	80/TCP	2m12s

web	NodePort	10.96.18.7	<none>	80:31819/TCP	50m
-----	----------	------------	--------	--------------	-----

- 每個 Pod 有唯一主機名
- 唯一域名

格式：

主機名稱.service名稱.命名空間.svc.cluster.local

示例：

nginx-statefulset-0.nginx.default.svc.cluster.local

說明：

- StatefulSet 通過主機名和域名，確保每個 Pod 都有唯一標識並支持有狀態應用。
- 無頭 Service 的 ClusterIP 為 `None`，允許 Pod 之間直接通信。

以下是圖片內容的文字辨識及繁體中文翻譯：

3. 部署守護進程 DaemonSet

- 在每個 Node 上運行一個 Pod，新加入的 Node 也同樣運行在一個 Pod 裡面。

示例：

在每個 Node 節點安裝數據採集工具。

指令操作：

1. 部署 DaemonSet：

```
[root@k8smaster ~]# kubectl apply -f ds.yaml
daemonset.apps/ds-test created
```

2. 查看 Pod：

```
[root@k8smaster ~]# kubectl get pods
NAME          READY   STATUS             RESTARTS   AGE
ds-test-cbk6v 0/1     ContainerCreating  0          0s
ds-test-cx6fk 1/1     Running            0          30s
```

3. 進入 Pod：

```
[root@k8smaster ~]# kubectl exec -it ds-test-cbk6v bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in future versions.
```

4. 查看文件：

```
root@ds-test-cx6fk:/# ls /tmp/log
anaconda boot.log-20200902 containers firewalld
```

說明：

- **DaemonSet** 用於確保每個 Node 上都執行一個 Pod，例如安裝系統監控、日誌收集、數據採集等工具。
- **特點：** 新加入的 Node 會自動部署相應的 Pod，適合全局性的守護進程需求。

🕒 修訂版本 #5

★ 由 treeman 建立於 14 🕒@🕒🕒 2025 10:28:31

✎ 由 treeman 更新於 17 🕒@🕒🕒 2025 18:18:45