1.1 Service 服务发现

1.1.1 使用 Service

步骤 1 创建 Service 实验使用的目录

```
[root@k8s-master labelfile]# mkdir /labfile/servicefile
[root@k8s-master labelfile]# cd /labfile/servicefile/
```

步骤 2 创建后端 httpd Deployment

```
[root@k8s-master servicefile]# vim httpd-deploy.yaml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: httpd
spec:
  replicas: 3
  selector:
    matchLabels:
      app: httpd
  template:
    metadata:
      labels:
        app: httpd
    spec:
      containers:
      - name: httpd
        image: httpd
        ports:
        - containerPort: 80
```

步骤 3 部署 deployment

```
[root@k8s-master servicefile]# kubectl apply -f httpd-deploy.yaml
```

```
deployment.apps/httpd created
```

查看

```
[root@k8s-master servicefile]# kubectl get deployment
```

```
NAME    READY   UP-TO-DATE   AVAILABLE   AGE
httpd   3/3     3            3           3m29s
```

步骤 4 创建 httpd-service.yaml 文件

```
[root@k8s-master servicefile]# vim httpd-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: httpd-svc
spec:
  selector:
    app: httpd
  ports:
  - protocol: TCP
    port: 8080
    targetPort: 80
```

步骤 5 创建 service

```
[root@k8s-master servicefile]# kubectl apply -f httpd-service.yaml
service/httpd-svc created
```

步骤 6 查看 service 信息，注意！千万不可删除名称为 kubernetes 的服务！该服务为系统服务！

```
[root@k8s-master servicefile]# kubectl get service
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
httpd-svc    ClusterIP   10.96.116.135   <none>        8080/TCP   53s
kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP    10d
```

步骤 7 查看 service 中的 endpoints

```
[root@k8s-master servicefile]# kubectl get endpoints
NAME         ENDPOINTS                                            AGE
httpd-svc    10.244.0.14:80,10.244.1.57:80,10.244.2.79:80   3m48s
kubernetes   192.168.137.11:6443                                  10d
```

步骤 8 测试服务可用性，使用 curl <服务 IP：端口号>查看回显，显示“It works！”表示成功。服务 IP 和端口号为步骤 6 中获取的值。

```
[root@k8s-master servicefile]# curl 10.96.116.135:8080
<html><body><h1>It works!</h1></body></html>
```

步骤 9 删除刚刚创建的服务，注意！千万不要删除 kubernetes 服务！

```
[root@k8s-master servicefile]# kubectl delete service httpd-svc
service "httpd-svc" deleted
```

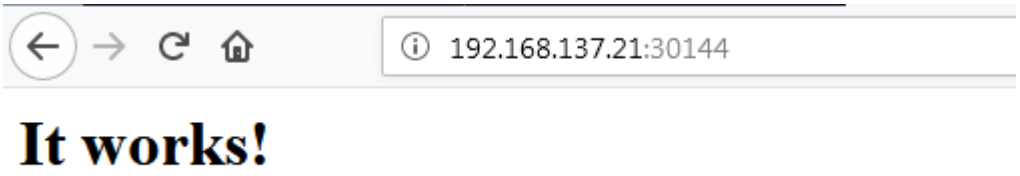
步骤 10　创建 httpd-expose.yaml

```
[root@k8s-master servicefile]# vim httpd-expose.yaml
apiVersion: v1
kind: Service
metadata:
  name: httpd-svc
spec:
  type: NodePort
  selector:
    app: httpd
  ports:
  - protocol: TCP
    port: 8080
    targetPort: 80
    nodePort: 30144
```

步骤 11　创建 httpd 服务。

```
[root@k8s-master servicefile]# kubectl apply -f httpd-expose.yaml
service/httpd-svc created
```

步骤 12　使用跳板机浏览器登录<节点 IP 地址：端口>，显示“It works！”表示成功



1.1.2 使用 DNS

步骤 1　创建 client.yaml，用于创建一个客户端 Pod，测试 DNS 功能。

```
[root@k8s-master servicefile]# vim client.yaml
apiVersion: v1
kind: Pod
metadata:
  name: clientpod
spec:
  containers:
    - name: clientpod
```

```
    image: busybox:1.28.3
    args:
    - /bin/sh
    - -c
    - sleep 30000
```

步骤 2　创建并进入 Pod 命令行界面

```
[root@k8s-master servicefile]# kubectl apply -f client.yaml
pod/clientpod created

 [root@k8s-master servicefile]# kubectl exec -it clientpod /bin/sh
/ #
```

步骤 3　使用 nslookup 命令查看服务域名

```
/ # nslookup 10.105.48.161
Server:    10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name:      10.105.48.161
Address 1: 10.105.48.161 httpd-svc.default.svc.cluster.local
```

步骤 4　使用 wget 命令通过域名访问服务

```
/ # wget httpd-svc:8080
Connecting to httpd-svc:8080 (10.105.48.161:8080)
index.html          100% |****************************************|    45
0:00:00 ETA
```

验证成功后删除 index 文件并退出容器

```
/ # rm index.html
/ # exit
```

1.1.3 使用 Headless Service

步骤 1　删除 httpd service，注意！不可删除 kubernetes 服务！

步骤 2　创建 headless 服务的 yaml 文件

```
[root@k8s-master servicefile]# vim headless-svc.yaml
```

```
apiVersion: v1
kind: Service
metadata:
  name: headless-svc
```

```
spec:
  selector:
    app: httpd
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  clusterIP: None
```

步骤 3　创建 headless 服务，可以看到 headless 服务没有 IP

```
[root@k8s-master servicefile]# kubectl apply -f headless-svc.yaml
service/headless-svc created
[root@k8s-master servicefile]# kubectl get service
NAME           TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
headless-svc   ClusterIP   None         <none>        80/TCP    25s
```

步骤 4　进入 client Pod，查看 DNS 解析

```
[root@k8s-master servicefile]# kubectl exec -it clientpod /bin/sh
/ # nslookup headless-svc
Server:    10.96.0.10
Address 1: 10.96.0.10 kube-dns.kube-system.svc.cluster.local

Name:      headless-svc
Address 1: 10.244.0.30 10-244-0-30.headless-svc.default.svc.cluster.local
Address 2: 10.244.2.130 10-244-2-130.headless-svc.default.svc.cluster.local
Address 3: 10.244.1.103 10-244-1-103.headless-svc.default.svc.cluster.local
```

步骤 5　访问服务，可以看到本次访问的服务由上一步中 address3 的 pod 提供。

```
/ # wget headless-svc
Connecting to headless-svc (10.244.1.103:80)
index.html           100% |****************************************|    45
0:00:00 ETA
```

```
/ # cat index.html
<html><body><h1>It works!</h1></body></html>
```

exit 退出