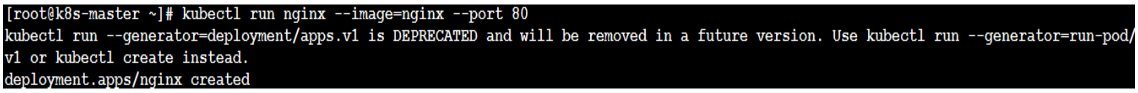
1.1 Kubernetes 网络

1.1.1 Node 与 Pod 间通信

步骤 1 创建一个 pod

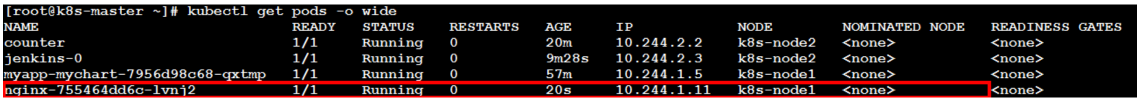
```
[root@k8s-master ~]# kubectl run nginx --image=nginx --port 80
```



```
[root@k8s-master ~]# kubectl run nginx --image=nginx --port 80
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.
deployment.apps/nginx created
```

步骤 2 查看该 pod 的 ip 地址及所处节点

```
[root@k8s-master ~]# kubectl get pods -o wide
```

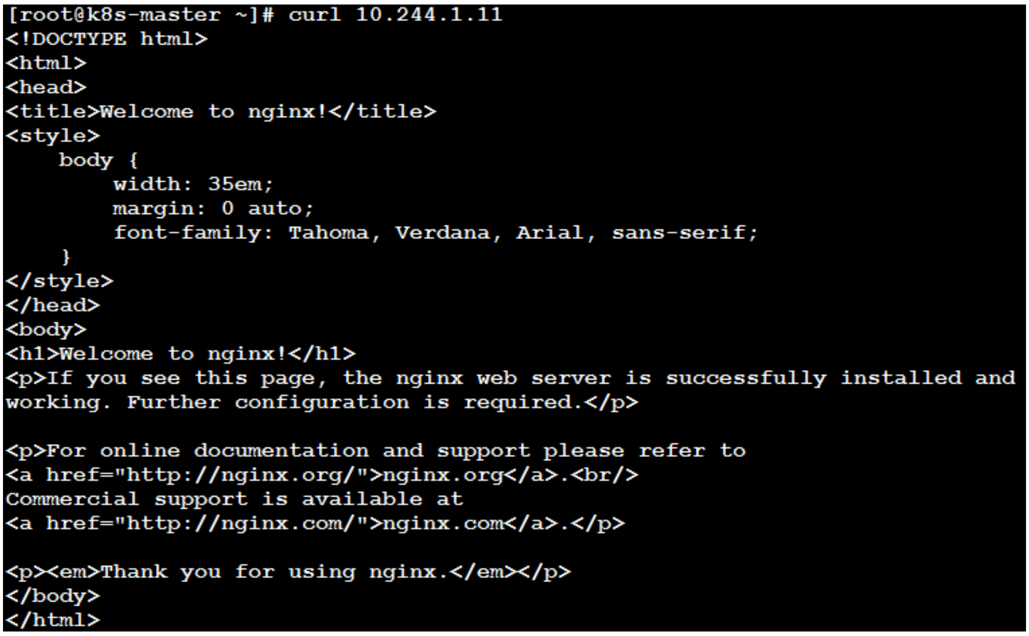


```
[root@k8s-master ~]# kubectl get pods -o wide
NAME                               READY   STATUS    RESTARTS   AGE     IP            NODE        NOMINATED NODE   READINESS GATES
counter                            1/1     Running   0          20m     10.244.2.2    k8s-node2   <none>           <none>
jenkins-0                          1/1     Running   0          9m28s   10.244.2.3    k8s-node2   <none>           <none>
myapp-mychart-7956d98c68-qxtmp     1/1     Running   0          57m     10.244.1.5    k8s-node1   <none>           <none>
nginx-755464dd6c-lvnj2             1/1     Running   0          20s     10.244.1.11   k8s-node1   <none>           <none>
```

步骤 3 从 k8s-master 节点访问该 pod

```
[root@k8s-master ~]# curl 10.244.1.11
```

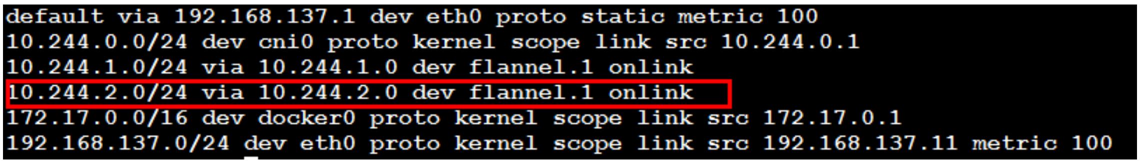
# 实际 IP 以实际输出为准



```
[root@k8s-master ~]# curl 10.244.1.11
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

步骤 4 查看主机上的路由可知，主机通过 Flannel 网络与 pod 通信

```
[root@k8s-master ~]# ip route
```



```
default via 192.168.137.1 dev eth0 proto static metric 100
10.244.0.0/24 dev cni0 proto kernel scope link src 10.244.0.1
10.244.1.0/24 via 10.244.1.0 dev flannel.1 onlink
10.244.2.0/24 via 10.244.2.0 dev flannel.1 onlink
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1
192.168.137.0/24 dev eth0 proto kernel scope link src 192.168.137.11 metric 100
```

flannel.1 是 flannel 插件生成的逻辑接口

1.1.2 Pod 和 Pod 间通信

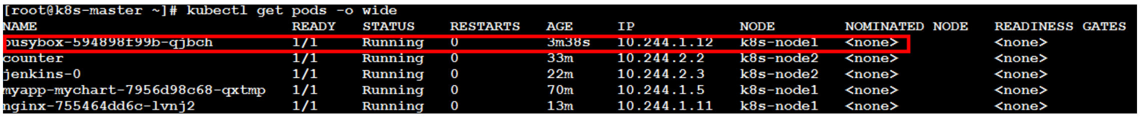
步骤 1 紧接上一个实验，创建一个 client pod，并进入该 pod 的 tty

```
[root@k8s-master ~]# kubectl run -i -t busybox --image=busybox
```

#该命令执行后将自动进入容器环境

#另起一个终端，查看 pods 运行所在节点

```
[root@k8s-master ~]# kubectl get pods -o wide
```

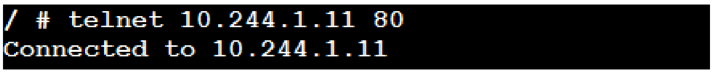


```
[root@k8s-master ~]# kubectl get pods -o wide
NAME                            READY   STATUS    RESTARTS   AGE     IP            NODE        NOMINATED NODE   READINESS GATES
busybox-594898f99b-qjbch        1/1     Running   0          3m38s   10.244.1.12   k8s-node1   <none>           <none>
counter                         1/1     Running   0          33m     10.244.2.2    k8s-node2   <none>           <none>
jenkins-0                       1/1     Running   0          22m     10.244.2.3    k8s-node2   <none>           <none>
myapp-mychart-7956d98c68-qxtmp  1/1     Running   0          70m     10.244.1.5    k8s-node1   <none>           <none>
nginx-755464dd6c-lvnj2          1/1     Running   0          13m     10.244.1.11   k8s-node1   <none>           <none>
```

步骤 2 连通性测试，从该 busybox pod 访问 nginx pod 服务。

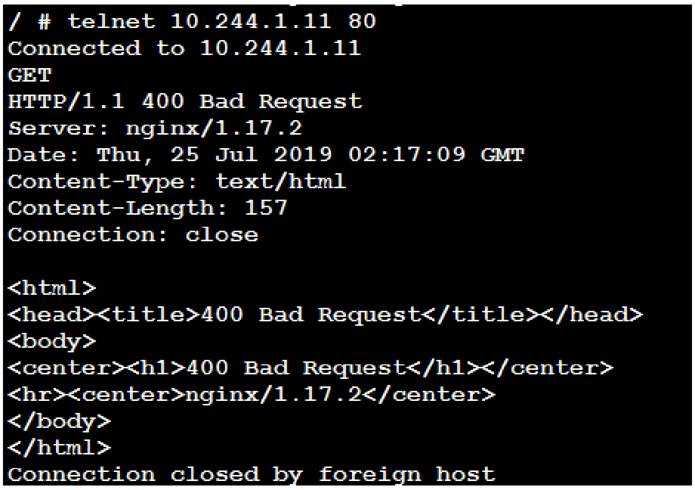
容器中无 curl，用 telnet 命令模拟访问

```
/ # telnet 10.244.1.41 80
```



```
/ # telnet 10.244.1.11 80
Connected to 10.244.1.11
```

GET #使用 GET 方法获取



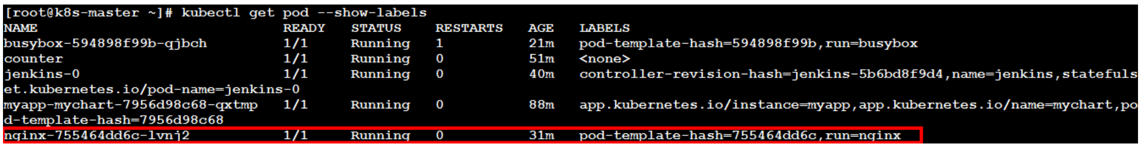
```
/ # telnet 10.244.1.11 80
Connected to 10.244.1.11
GET
HTTP/1.1 400 Bad Request
Server: nginx/1.17.2
Date: Thu, 25 Jul 2019 02:17:09 GMT
Content-Type: text/html
Content-Length: 157
Connection: close

<html>
<head><title>400 Bad Request</title></head>
<body>
<center><h1>400 Bad Request</h1></center>
<hr><center>nginx/1.17.2</center>
</body>
</html>
Connection closed by foreign host
```

exit 退出

1.1.3 集群外访问实验（NodePort 方式）

步骤 1 查看已经运行的 pod 的 label

```
[root@k8s-master ~]# kubectl get pod --show-labels
```



```
[root@k8s-master ~]# kubectl get pod --show-labels
NAME                          READY   STATUS    RESTARTS   AGE   LABELS
busybox-594898f99b-qjbch      1/1     Running   1          21m   pod-template-hash=594898f99b,run=busybox
counter                       1/1     Running   0          51m   <none>
jenkins-0                     1/1     Running   0          40m   controller-revision-hash=jenkins-5b6bd8f9d4,name=jenkins,statefuls
et.kubernetes.io/pod-name=jenkins-0
myapp-mychart-7956d98c68-qxtmp   1/1     Running   0          88m   app.kubernetes.io/instance=myapp,app.kubernetes.io/name=mychart,po
d-template-hash=7956d98c68
nginx-755464dd6c-lvnj2        1/1     Running   0          31m   pod-template-hash=755464dd6c,run=nginx
```

步骤 2 在 master 节点创建/labfile/log 目录，用于保存配置文件。后续本章的 yaml 文件保存在此处。

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

步骤 3 创建 type 为 nodePort 的 Service 的 yaml 文件

```
[root@k8s-master network]# vim serviceNodeport.yaml
```

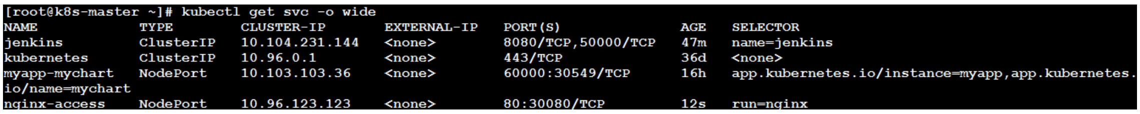
```
apiVersion: v1
kind: Service
metadata:
  name: nginx-access
  namespace: default
spec:
  selector:
    run: nginx
  clusterIP: 10.96.123.123
  type: NodePort
  ports:
  - port: 80   #service 端口
    targetPort: 80 #容器端口
    nodePort: 30080    #node 节点端口
```

步骤 4 创建该 Service

```
[root@k8s-master network]# kubectl apply -f serviceNodeport.yaml
```

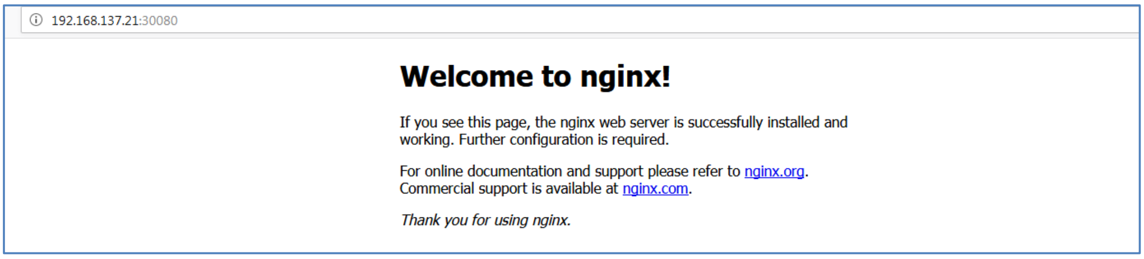
步骤 5 查看该 service 状态

```
[root@k8s-master ~]# kubectl get svc -o wide
```



```
[root@k8s-master ~]# kubectl get svc -o wide
NAME            TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)              AGE   SELECTOR
jenkins         ClusterIP   10.104.231.144   <none>        8080/TCP,50000/TCP   47m   name=jenkins
kubernetes      ClusterIP   10.96.0.1        <none>        443/TCP              36d   <none>
myapp-mychart   NodePort    10.103.103.36    <none>        60000:30549/TCP      16h   app.kubernetes.io/instance=myapp,app.kubernetes.
io/name=mychart
nginx-access    NodePort    10.96.123.123    <none>        80:30080/TCP         12s   run=nginx
```

步骤 6 集群外主机可使用 Node 节点地址访问该 Pod 服务（任一 Node 节点地址均可）。



192.168.137.21:30080

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

步骤 7 在节点上用 ss 命令查看可知，每个节点上都有一个 kube-proxy 发起的进程监听 30080 端口

```
[root@k8s-master ~]# ss -ntlp | grep 30080
LISTEN   0    128    :::30080    :::*    users:(("kube-proxy",pid=15859,fd=12))
```