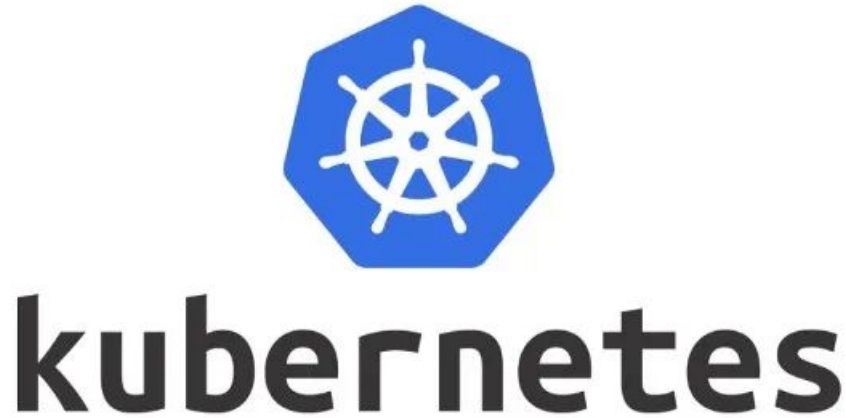


### Kubernetes 系列（三十九）Kubernetes 中的 Service 端口详解

---

对于那些刚接触 Kubernetes 的人来说，移植可能有点令人困惑。因此，我决定通过一个基本示例对端口进行简要概述，以理解端口在 Kubernetes 中的工作方式。



云原生拓展

Kubernetes 为 Service 提供了几种端口配置:

- **Port**: 公开服务的端口。其他 Pod 可以通过这个端口与它通信。
- **TargetPort**: 部署的容器的实际端口。服务向该端口发送请求，pod容器必须侦听相同的端口。
- **NodePort**: 对集群外公开服务。因此可以通过这个端口从外部访问应用程序。默认情况下，它在部署期间自动分配。

这里我有一个简单YAML文件，包括一个服务和一个带Nginx容器的pod。输出将是一个简单的Nginx欢迎页面。

该 Pod 的名称为 `nginx-welcome-test`。对于使用Nginx的Docker 镜像，我从Docker Hub中选择了 [praqma/network-multitool](#)，因为他们在容器中有Nginx和net-tools，所以我们将能够运行' curl '命令，而不需要任何额外的安装。

值得注意的是，我的 targetPort 值是80。这是Nginx将部署的实际端口。所以我的服务有一个 **targetPort:80**， pod容器将有 **containerPort:80**。

Service的 **端口8030** 是我们能够从集群中的其他pod访问Nginx页面的端口。NodePort将在部署后自动分配。

```

apiVersion: v1
kind: Service
metadata:
  name: test-service-nginx-welcome
spec:
  type: NodePort
  selector:
    app: nginx-welcome-test
  ports:
    - protocol: TCP
      port: 8030
      targetPort: 80
      nodePort: 30036
---
apiVersion: v1
kind: Pod
metadata:
  name: nginx-welcome-test
  labels:
    app: nginx-welcome-test
spec:
  containers:
    - name: nginx-container
      image: praqma/network-multitool
      ports:
        - containerPort: 80

```

## Deployment

运行下面的命令去部署

```
kubectl apply -f nginx-welcome-test-config.yaml
```

## 检查一切都在成功运行并且检查参数

```

kubectl describe service test-service-nginx-welcome
kubectl describe pod nginx-welcome-test

```

## 从其他 Pod 检查连接

正如前面提到的，Service 的端口是8030。所以其他的 Pod 可以通过端口8030到达这个Nginx页面。注意 `test-service-ngi`

`nx-welcome` 是YAML文件中服务的名称。

```
kubectl run -i --tty alpine-testpod --image=ubuntu --restart=Never -- sh
pip install requests
python3
r = requests.get('http://test-service-nginx-welcome:8030')
```

当检查响应 `r.text` 你应该看到Welcome to nginx!页面。

你可以通过 `kubectl delete pod alpine-testpod` 删除这个测试pod

## 在 Nginx 容器内部检查连接

`nginx-container` 是YAML文件中Nginx容器的名称。

容器命令内部的日志记录如下所示

```
kubectl exec --stdin --tty nginx-welcome-test -c nginx-container -- sh
```

您可以通过命令 `netstat -tulpn | grep LISTEN` 列出所有打开的端口

您将看到其中一个将具有开放的端口80，该端口在YAML文件中指定为 `targetPort`

调用 `curl http://127.0.0.1:80` 应该输出我们的Nginx欢迎页面。

## 集群外部访问

检查 Pod 信息

```
kubectl describe pod nginx-welcome-test
```

您应该看到带有Node信息的行，如下所示

```
Node: master**/X.X.X.X**
```

这就是你可以看到 pod 运行所在的节点，同时该信息显示实例的名称和内部IP。您可以很容易地找到外部访问的IP。

通过运行如下命令，获取外部访问的端口：

```
kubectl describe service test-service-nginx-welcome
```

你将会看到如下的 NodePort 信息的显示：

```
NodePort: <unset> **30036**/TCP
```

您可以从外部通过组合`external_ip:node_port`浏览这个页面，您将看到相同的Nginx欢迎页面。

我希望我的文章对你有帮助。请随意分享你的想法、建议、想法或问任何问题。感谢您的阅读。

欢迎关注我的公众号“云原生拓展”，原创技术文章第一时间推送。