原文地址：https://blog.csdn.net/lxwh0323/category\_11505875.html

**Service yarm文件清单**

kind: Service # 资源类型

apiVersion: v1 # 资源版本

metadata: # 元数据

name: service # 资源名称

namespace: dev # 命名空间

spec: # 描述

selector: # 标签选择器，用于确定当前service代理哪些pod

app: nginx

type: # 类型，值NodePort允许外部访问，值 ClusterIP 只能在集群内访问

clusterIP: # 虚拟服务的ip地址

sessionAffinity: # session亲和性，支持ClientIP、None两个选项

ports: # 端口信息

- protocol: TCP

port: 3017 # service端口

targetPort: 5003 # pod端口

nodePort: 31122 # 主机端口

**创建deployment**

在开始之前，我们先创建一个deployment用于后面的示例

我们新建一个deployment.yaml文件，用于创建deployment

apiVersion: apps/v1

kind: Deployment

metadata:

name: pc-deployment

namespace: dev

spec:

# 副本数量

replicas: 3

selector:

matchLabels:

app: nginx-pod

template:

# pods 的标签

metadata:

labels:

app: nginx-pod

# 使用的镜像

spec:

containers:

- name: nginx

image: nginx:1.17.1

ports:

- containerPort: 80

执行如下命令创建deployment

root@k8s-master:~# kubectl create -f deployment.yaml

deployment.apps/pc-deployment created

使用如下命令查看创建的pods

root@k8s-master:~# kubectl get pods -n dev -o wide

NAME READY STATUS RESTARTS AGE IP NODE pc-deployment-6756f95949-gnt28 1/1 Running 0 6m55s 10.244.2.47 k8s-node2 pc-deployment-6756f95949-k9gvd 1/1 Running 0 6m55s 10.244.2.46 k8s-node2 pc-deployment-6756f95949-pzfhm 1/1 Running 0 6m55s 10.244.1.43 k8s-node1

**创建Service**

我们创建一个service-nodeport.yaml文件

apiVersion: v1

kind: Service

metadata:

name: service-nodeport

namespace: dev

spec:

selector:

app: nginx-pod

type: NodePort # service类型

ports:

- port: 80

# 指定绑定的主机的端口, 如果不指定，会默认分配

nodePort: 30002

targetPort: 80

执行如下命令创建service

root@k8s-master:~# kubectl create -f service-nodeport.yaml

service/service-nodeport created

我们查询创建的service

root@k8s-master:~# kubectl get services -n dev

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

service-nodeport NodePort 10.1.122.141 <none> 80:30002/TCP 3m33s

我们访问如下url

其中192.168.102.130是我master的ip地址

而返回的1、2、3，是我为了方便测试修改了我们生成的3个pod的index.html

root@k8s-master:~# curl 192.168.102.130:30002

1

root@k8s-master:~# curl 192.168.102.130:30002

2

root@k8s-master:~# curl 192.168.102.130:30002

2

root@k8s-master:~# curl 192.168.102.130:30002

2

root@k8s-master:~# curl 192.168.102.130:30002

3

可以看到kubernetes随机将我们的请求转发至pod

**ingress-nginx**

通过service的type: NodePort可以在外部访问pod，如果我们有多个service，其映射到不同的url前缀（如/svc1影视到service1，/svc2映射到service2）,这时候应该怎么实现呢，这就要使用到下面介绍的ingress

1. 安装ingress-nginx

# kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.1.2/deploy/static/provider/cloud/deploy.yaml

注意版本，如下可以查看版本地址和对应的kubernetes版本：

[https://github.com/kubernetes/ingress-nginx](https://github.com/kubernetes/ingress-nginx（可以查看适用的kubernetes版本）)

详细的安装请查阅：<https://kubernetes.github.io/ingress-nginx/deploy/#quick-start>

1. 查看是否安装成功

root@k8s-master:~# kubectl get service ingress-nginx-controller --namespace=ingress-nginx

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

ingress-nginx-controller LoadBalancer 10.1.122.150 <pending> 80:31045/TCP,443:32045/TCP 173m

可以看到ingress暴露了一个服务，其对应的主机的端口是31045和32045

如果想更改其对应的主机端口，执行如下命令

# kubectl edit service ingress-nginx-controller --namespace=ingress-nginx

1. 创建新的Deployment和Service

我们创建tomcat-nginx.yaml文件，其内容如下

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

namespace: dev

spec:

replicas: 3

selector:

matchLabels:

app: nginx-pod

template:

metadata:

labels:

app: nginx-pod

spec:

containers:

- name: nginx

image: nginx:1.17.1

ports:

- containerPort: 80

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: tomcat-deployment

namespace: dev

spec:

replicas: 3

selector:

matchLabels:

app: tomcat-pod

template:

metadata:

labels:

app: tomcat-pod

spec:

containers:

- name: tomcat

image: tomcat:8.5-jre10-slim

ports:

- containerPort: 8080

---

apiVersion: v1

kind: Service

metadata:

name: nginx-service

namespace: dev

spec:

selector:

app: nginx-pod

clusterIP: None

type: ClusterIP

ports:

- port: 80

targetPort: 80

---

apiVersion: v1

kind: Service

metadata:

name: tomcat-service

namespace: dev

spec:

selector:

app: tomcat-pod

clusterIP: None

type: ClusterIP

ports:

- port: 8080

targetPort: 8080

该文件创建了2个Deployment（nginx-deployment和tomcat-deployment）并暴露了2个Service（nginx-service和tomcat-service）

执行如下命令创建这些资源

# kubectl create -f tomcat-nginx.yaml

查看资源是否创建成功

root@k8s-master:~# kubectl get services -n dev

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

nginx-service ClusterIP None <none> 80/TCP 66s

tomcat-service ClusterIP None <none> 8080/TCP 66s

1. Ingress代理http

Ingress的yaml文件参考地址：

<https://kubernetes.io/docs/concepts/services-networking/ingress/>

我们创建ingress-http.yaml文件

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: ingress-http

namespace: dev

annotations:

nginx.ingress.kubernetes.io/rewrite-target: /$2

spec:

ingressClassName: nginx

rules:

- http:

paths:

# 将 /nginx 前缀的url代理到 nginx-service Service的80端口

- path: /nginx(/|$)(.\*)

pathType: Prefix

backend:

service:

name: nginx-service

port:

number: 80

# 将 /tomcat 前缀的url代理到 tomcat-service Service的80端口

- path: /tomcat(/|$)(.\*)

pathType: Prefix

backend:

service:

name: tomcat-service

port:

number: 8080

注意：ingressClassName: nginx，在k8s官网给的示例是没有这一行，导致我访问一直404

关于我上面使用的path的路径格式请参考：

<https://kubernetes.github.io/ingress-nginx/examples/rewrite/>

执行如下命令创建ingress资源

# kubectl create -f ingress-http.yaml

查看ingress是否创建成功

root@k8s-master:~# kubectl get ingress ingress-http -n dev

NAME CLASS HOSTS ADDRESS PORTS AGE

ingress-http ingress-http \* 80 4m3s

1. 测试

我们访问 192.168.102.130（我master的地址）就可以看到返回tomcat的页面

# curl http://192.168.102.130:31045/tomcat

1. 关于ingress应用https

https://kubernetes.io/zh/docs/concepts/services-networking/ingress/#tls