## Ingress概念

通俗来讲，ingress和之前提到的Service、Deployment，也是一个k8s的资源类型，ingress用于实现用域名的方式访问k8s内部应用。

## Ingress安装

首先安装helm管理工具：<https://helm.sh/docs/intro/install/>

使用helm安装ingress：[https://kubernetes.github.io/ingress-nginx/deploy/#using-helm](https://kubernetes.github.io/ingress-nginx/deploy/" \l "using-helm)

1. 添加ingress的helm仓库（课程讲解的版本已经上传至百度网盘）

helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx

1. 下载ingress的helm包至本地

helm pull ingress-nginx/ingress-nginx

1. 更改对应的配置

tar xf ingress-nginx-3.6.0.tgz

cd ingress-nginx

vim values.yaml

1. 需要修改的位置
   1. Controller和admissionWebhook的镜像地址，需要将公网镜像同步至公司内网镜像仓库（和课程不一致的版本，需要自行同步gcr镜像的，可以百度查一下使用阿里云同步gcr的镜像，也可以参考这个连接https://blog.csdn.net/weixin\_39961559/article/details/80739352，或者参考这个连接： https://blog.csdn.net/sinat\_35543900/article/details/103290782）
   2. hostNetwork设置为true
   3. dnsPolicy设置为 ClusterFirstWithHostNet
   4. NodeSelector添加ingress: "true"部署至指定节点
   5. 类型更改为kind: DaemonSet
2. 部署ingress

给需要部署ingress的节点上打标签

kubectl label node k8s-master03 ingress=true

kubectl create ns ingress-nginx

helm install ingress-nginx -n ingress-nginx .

1. 将ingress controller部署至Node节点（ingress controller不能部署在master节点，需要安装视频中的步骤将ingress controller部署至Node节点，生产环境最少三个ingress controller，并且最好是独立的节点）

kubectl label node k8s-node01 ingress=true

kubectl label node k8s-master03 ingress-

## Ingress入门使用

创建一个ingress：

# cat ingress.yaml

apiVersion: networking.k8s.io/v1beta1 # networking.k8s.io/v1 / extensions/v1beta1

kind: Ingress

metadata:

annotations:

kubernetes.io/ingress.class: "nginx"

name: example

spec:

rules: # 一个Ingress可以配置多个rules

- host: foo.bar.com # 域名配置，可以不写，匹配\*， \*.bar.com

http:

paths: # 相当于nginx的location配合，同一个host可以配置多个path / /abc

- backend:

serviceName: nginx-svc

servicePort: 80

path: /

创建一个多域名ingress

cat ingress-mulDomain.yaml

apiVersion: networking.k8s.io/v1beta1 # networking.k8s.io/v1 / extensions/v1beta1

kind: Ingress

metadata:

annotations:

kubernetes.io/ingress.class: "nginx"

name: example

spec:

rules: # 一个Ingress可以配置多个rules

- host: foo.bar.com # 域名配置，可以不写，匹配\*， \*.bar.com

http:

paths: # 相当于nginx的location配合，同一个host可以配置多个path / /abc

- backend:

serviceName: nginx-svc

servicePort: 80

path: /

- host: foo2.bar.com # 域名配置，可以不写，匹配\*， \*.bar.com

http:

paths: # 相当于nginx的location配合，同一个host可以配置多个path / /abc

- backend:

serviceName: nginx-svc-external

servicePort: 80

path: /