**k8s部署之Ingress**

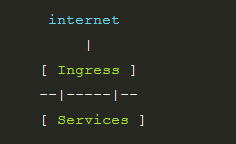
参考：

https://www.kubernetes.org.cn/ingress

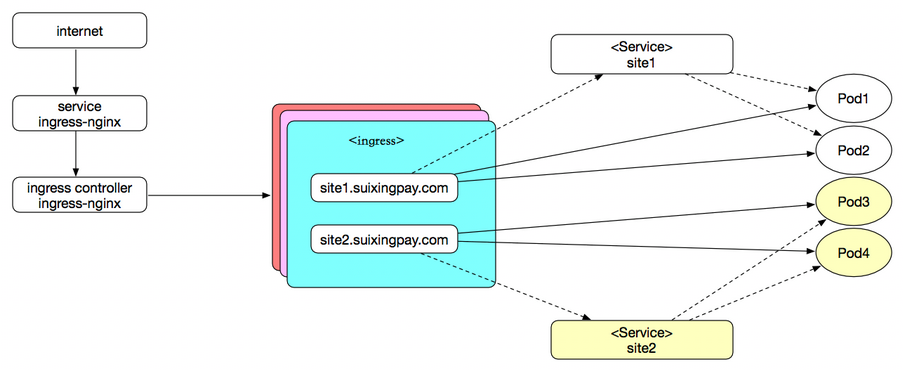
https://blog.csdn.net/qq\_27639619/article/details/88819751

**1、Ingress简介**

通常情况下，service和pod的IP仅可在集群内部访问。集群外部的请求需要通过负载均衡转发到service在Node上暴露的NodePort上，然后再由kube-proxy将其转发给相关的Pod。而Ingress就是为授权集群的请求提供路由规则的集合，如下图所示



Ingress可以给service提供集群外部访问的URL、负载均衡、SSL终止、HTTP路由等。为了配置这些Ingress规则，集群管理员需要部署一个Ingress controller，它监听Ingress和service的变化，并根据规则配置负载均衡并提供访问入口。



**2、Ingress资源定义**

rules 定义成虚拟主机调度还是url调度

backend  后端服务

**3、单服务Ingress**

apiVersion: extensions/v1beta1

kind: Ingress

metadata:

  name: test-ingress

spec:

  backend:

    serviceName: test-service

    servicePort: 80

客户端到Ingress Controller的访问请求都将被转发到后端的唯一Service(test-service)上，这种情况下Ingress 无须定义任何rule.



**4、同域名不同URL多服务Ingress**

apiVersion: extensions/v1beta1

kind: Ingress

metadata:

  name: test-ingress

spec:

  rules:

  - host: api.test.com

    http:

      paths:

      - path: /v1

        backend:

          serviceName: test-service1

          servicePort: 80

      - path: /v2

        backend:

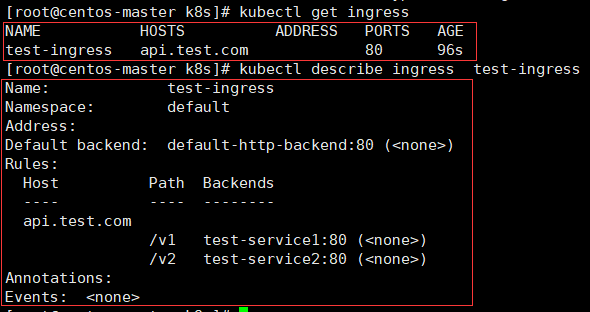
          serviceName: test-service2

          servicePort: 80

对于同一域名api.test.com的请求，不同的的路径/v1、/v2将会被转发到对应用后端服务上

api.test.com/v1 -> test-service1:80

api.test.com/v2 -> test-service2:80



**5、不同域名不同URL多服务Ingress**

apiVersion: extensions/v1beta1

kind: Ingress

metadata:

  name: test-ingress

spec:

  rules:

  - host: v1.test.com

    http:

      paths:

      - backend:

          serviceName: test-service1

          servicePort: 80

  - host: v2.test.com

    http:

      paths:

      - backend:

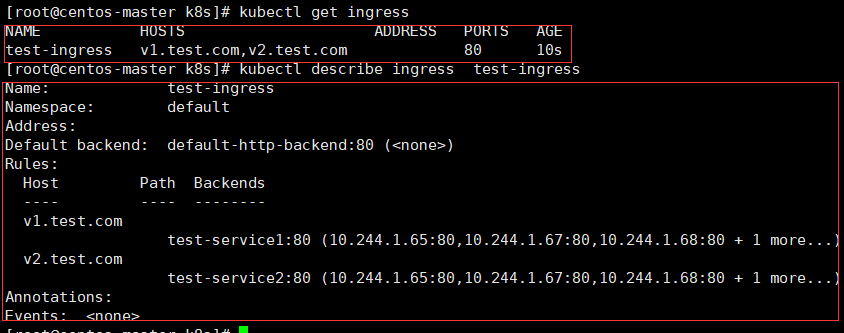
          serviceName: test-service2

          servicePort: 80

对于不同域名api.test.com的请求，将会被转发到对应用后端服务上

V1.test.com -> test-service1

V2.test.com -> test-service2



**部署ingress-nginx**

参考文档：https://kubernetes.github.io/ingress-nginx/deploy/

ingress-nginx地址：https://github.com/kubernetes/ingress-nginx

**1、所有部署都需要以下强制命令**

在这个mandatory.yaml文件中，会有以下操作：

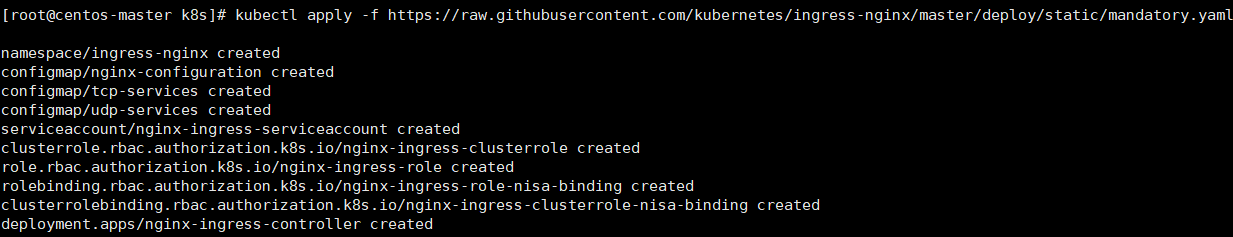
(1)-会创建独立的命名空间ingress-nginx。

(2)-创建ConfigMap来保存配置数据或环境变量。

(3)-创建Ingress用到的ServiceAccount、ClusterRole、Role、RoleBinding、ClusterRoleBinding负责Ingress的RBAC授权的控制。

(4)创建用于部署nginx-ingress的deployment。

kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/master/deploy/static/mandatory.yaml



无法科学上网，可通过如下地址下载yaml文件

<https://github.com/kubernetes/ingress-nginx/tree/master/deploy/static>

文件中使用的是默认的镜像可能无法正常拉取，因此替换为docker hub的源：

<https://hub.docker.com/r/bitnami/nginx-ingress-controller>

quay.io/kubernetes-ingress-controller/nginx-ingress-controller:0.26.1

替换

bitnami/nginx-ingress-controller:0.26.1

**2、创建nginx-ingress服务**

kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/master/deploy/static/provider/cloud-generic.yaml



**3、查看nginx-ingress服务**

kubectl get pods --all-namespaces -l app.kubernetes.io/name=ingress-nginx --watch

