Ingress：

Ingress-nginx：kubernetes官方维护的ingress

Nginx-ingress：nginx官方维护的ingress

差别：<https://github.com/nginxinc/kubernetes-ingress/blob/master/docs/nginx-ingress-controllers.md>

Ingress-nginx的官方文档：[https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/annotations/#rewrite](https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/annotations/" \l "rewrite)

Nginx-ingress的官方文档：<https://docs.nginx.com/nginx-ingress-controller/configuration/ingress-resources/advanced-configuration-with-annotations/>

Ingress-nginx Github：<https://github.com/kubernetes/ingress-nginx>

Nginx-ingress Github：<https://github.com/nginxinc/kubernetes-ingress/blob/master/docs/nginx-ingress-controllers.md>

Traefik、HAProxy、Istio

DaemonSet，找几台专门的服务器进行配置ingress，QoS

hostNetwork: true。

创建一个ingress实例

vim ingress-demo.yaml

apiVersion: extensions/v1beta1

kind: Ingress

metadata:

name: ingress-test

namespace: ratel-test1

spec:

rules:

- host: ingress.test.com

http:

paths:

- backend:

serviceName: ingress-test

servicePort: 80

path: /

kubectl create -f ingress-demo.yaml

Redirect：

apiVersion: v1

items:

- apiVersion: extensions/v1beta1

kind: Ingress

metadata:

annotations:

nginx.ingress.kubernetes.io/permanent-redirect: https://www.baidu.com

name: ingress-test

namespace: ratel-test1

spec:

rules:

- host: ingress.test.com

http:

paths:

- backend:

serviceName: ingress-test

servicePort: 80

path: /

kind: List

metadata:

resourceVersion: ""

selfLink: ""

Rewrite：

apiVersion: extensions/v1beta1

kind: Ingress

metadata:

annotations:

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

generation: 4

name: ingress-test

namespace: ratel-test1

spec:

rules:

- host: rewrite.test.com

http:

paths:

- backend:

serviceName: ingress-test

servicePort: 80

path: /something(/|$)(.\*)

<https://kubernetes.github.io/ingress-nginx/user-guide/tls/>

1.a.com 2.a.com x.a.com

\*.a.com

禁用https强制跳转

nginx.ingress.kubernetes.io/ssl-redirect: "false"

apiVersion: extensions/v1beta1

kind: Ingress

metadata:

annotations:

nginx.ingress.kubernetes.io/ssl-redirect: "false"

generation: 1

name: test-tls

namespace: ratel-test1

spec:

rules:

- host: test-tls.test.com

http:

paths:

- backend:

serviceName: ingress-test

servicePort: 80

path: /

tls:

- hosts:

- test-tls.test.com

secretName: ca-cert

设置默认证书：--default-ssl-certificate=default/foo-tls

更改的ingress-controller的启动参数

Dashboard自定义证书

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

k8s-app: kubernetes-dashboard

name: kubernetes-dashboard

namespace: kubernetes-dashboard

spec:

progressDeadlineSeconds: 600

replicas: 1

revisionHistoryLimit: 10

selector:

matchLabels:

k8s-app: kubernetes-dashboard

strategy:

rollingUpdate:

maxSurge: 25%

maxUnavailable: 25%

type: RollingUpdate

template:

metadata:

labels:

k8s-app: kubernetes-dashboard

spec:

affinity:

nodeAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

nodeSelectorTerms:

- matchExpressions:

- key: normal

operator: In

values:

- "true"

containers:

- args:

- --auto-generate-certificates=false

- --tls-key-file=server.key

- --tls-cert-file=server.pem

- --token-ttl=21600

- --authentication-mode=basic,token

- --namespace=kubernetes-dashboard

image: kubernetesui/dashboard:v2.0.0-rc5

imagePullPolicy: Always

lifecycle: {}

livenessProbe:

failureThreshold: 3

httpGet:

path: /

port: 8443

scheme: HTTPS

initialDelaySeconds: 30

periodSeconds: 10

successThreshold: 1

timeoutSeconds: 30

name: kubernetes-dashboard

ports:

- containerPort: 8443

protocol: TCP

resources: {}

securityContext:

privileged: false

procMount: Default

runAsNonRoot: false

terminationMessagePath: /dev/termination-log

terminationMessagePolicy: File

volumeMounts:

- mountPath: /certs

name: kubernetes-dashboard-new

- mountPath: /tmp

name: tmp-volume

dnsPolicy: ClusterFirst

restartPolicy: Always

schedulerName: default-scheduler

securityContext: {}

serviceAccount: kubernetes-dashboard

serviceAccountName: kubernetes-dashboard

terminationGracePeriodSeconds: 30

tolerations:

- effect: NoSchedule

key: node-role.kubernetes.io/master

operator: Exists

volumes:

- name: kubernetes-dashboard-new

secret:

defaultMode: 420

secretName: kubernetes-dashboard-new

- emptyDir: {}

name: tmp-volume

黑白名单：

Annotations：只对指定的ingress生效

ConfigMap：全局生效

黑名单可以使用ConfigMap去配置，白名单建议使用Annotations去配置。

<https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/annotations/#whitelist-source-range>

[root@k8s-master01 install-some-apps]# kubectl get cm -n ingress-nginx nginx-configuration -oyaml

apiVersion: v1

data:

block-cidrs: 192.168.1.19

kind: ConfigMap

metadata:

labels:

app.kubernetes.io/name: ingress-nginx

app.kubernetes.io/part-of: ingress-nginx

name: nginx-configuration

namespace: ingress-nginx

<https://github.com/kubernetes/ingress-nginx/blob/master/docs/user-guide/nginx-configuration/annotations.md#canary>

官方错误页面：

<https://github.com/kubernetes/ingress-nginx/blob/nginx-0.20.0/docs/examples/customization/custom-errors/custom-default-backend.yaml>

ingress nginx 监控：<https://kubernetes.github.io/ingress-nginx/user-guide/monitoring/>

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

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