kubernetes部署前端

建立日期：2020年8月21日

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 变更原因 | 变更内容简述 | 编制/修订者 | 适用环境 | 发布日期 |
| V1.0 | 建立 |  | 陈文华 | 开发、测试 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## 部署Ingress Controller

1. 在node1节点部署Ingress Controller，给Node1节点打个标签

kubectl label node node1 type="ingress"

1. 在master服务器platform-kubernetes目录下创建ingress-nginx.yml资源文件

vim ingress-nginx.yml

内容如下

**apiVersion:** v1  
**kind:** Namespace  
**metadata:  
 name:** ingress-nginx  
 **labels:  
 app.kubernetes.io/name:** ingress-nginx  
 **app.kubernetes.io/part-of:** ingress-nginx  
  
---  
  
**kind:** ConfigMap  
**apiVersion:** v1  
**metadata:  
 name:** nginx-configuration  
 **namespace:** ingress-nginx  
 **labels:  
 app.kubernetes.io/name:** ingress-nginx  
 **app.kubernetes.io/part-of:** ingress-nginx  
  
---  
**kind:** ConfigMap  
**apiVersion:** v1  
**metadata:  
 name:** tcp-services  
 **namespace:** ingress-nginx  
 **labels:  
 app.kubernetes.io/name:** ingress-nginx  
 **app.kubernetes.io/part-of:** ingress-nginx  
  
---  
**kind:** ConfigMap  
**apiVersion:** v1  
**metadata:  
 name:** udp-services  
 **namespace:** ingress-nginx  
 **labels:  
 app.kubernetes.io/name:** ingress-nginx  
 **app.kubernetes.io/part-of:** ingress-nginx  
  
---  
**apiVersion:** v1  
**kind:** ServiceAccount  
**metadata:  
 name:** nginx-ingress-serviceaccount  
 **namespace:** ingress-nginx  
 **labels:  
 app.kubernetes.io/name:** ingress-nginx  
 **app.kubernetes.io/part-of:** ingress-nginx  
  
---  
**apiVersion:** rbac.authorization.k8s.io/v1beta1  
**kind:** ClusterRole  
**metadata:  
 name:** nginx-ingress-clusterrole  
 **labels:  
 app.kubernetes.io/name:** ingress-nginx  
 **app.kubernetes.io/part-of:** ingress-nginx  
**rules:** - **apiGroups:** - **""  
 resources:** - configmaps  
 - endpoints  
 - nodes  
 - pods  
 - secrets  
 **verbs:** - list  
 - watch  
 - **apiGroups:** - **""  
 resources:** - nodes  
 **verbs:** - get  
 - **apiGroups:** - **""  
 resources:** - services  
 **verbs:** - get  
 - list  
 - watch  
 - **apiGroups:** - **""  
 resources:** - events  
 **verbs:** - create  
 - patch  
 - **apiGroups:** - **"extensions"** - **"networking.k8s.io"  
 resources:** - ingresses  
 **verbs:** - get  
 - list  
 - watch  
 - **apiGroups:** - **"extensions"** - **"networking.k8s.io"  
 resources:** - ingresses/status  
 **verbs:** - update  
  
---  
**apiVersion:** rbac.authorization.k8s.io/v1beta1  
**kind:** Role  
**metadata:  
 name:** nginx-ingress-role  
 **namespace:** ingress-nginx  
 **labels:  
 app.kubernetes.io/name:** ingress-nginx  
 **app.kubernetes.io/part-of:** ingress-nginx  
**rules:** - **apiGroups:** - **""  
 resources:** - configmaps  
 - pods  
 - secrets  
 - namespaces  
 **verbs:** - get  
 - **apiGroups:** - **""  
 resources:** - configmaps  
 **resourceNames:** - **"ingress-controller-leader-nginx"  
 verbs:** - get  
 - update  
 - **apiGroups:** - **""  
 resources:** - configmaps  
 **verbs:** - create  
 - **apiGroups:** - **""  
 resources:** - endpoints  
 **verbs:** - get  
  
---  
**apiVersion:** rbac.authorization.k8s.io/v1beta1  
**kind:** RoleBinding  
**metadata:  
 name:** nginx-ingress-role-nisa-binding  
 **namespace:** ingress-nginx  
 **labels:  
 app.kubernetes.io/name:** ingress-nginx  
 **app.kubernetes.io/part-of:** ingress-nginx  
**roleRef:  
 apiGroup:** rbac.authorization.k8s.io  
 **kind:** Role  
 **name:** nginx-ingress-role  
**subjects:** - **kind:** ServiceAccount  
 **name:** nginx-ingress-serviceaccount  
 **namespace:** ingress-nginx  
  
---  
**apiVersion:** rbac.authorization.k8s.io/v1beta1  
**kind:** ClusterRoleBinding  
**metadata:  
 name:** nginx-ingress-clusterrole-nisa-binding  
 **labels:  
 app.kubernetes.io/name:** ingress-nginx  
 **app.kubernetes.io/part-of:** ingress-nginx  
**roleRef:  
 apiGroup:** rbac.authorization.k8s.io  
 **kind:** ClusterRole  
 **name:** nginx-ingress-clusterrole  
**subjects:** - **kind:** ServiceAccount  
 **name:** nginx-ingress-serviceaccount  
 **namespace:** ingress-nginx  
  
---  
  
**apiVersion:** apps/v1  
**kind:** Deployment  
**metadata:  
 name:** nginx-ingress-controller  
 **namespace:** ingress-nginx  
 **labels:  
 app.kubernetes.io/name:** ingress-nginx  
 **app.kubernetes.io/part-of:** ingress-nginx  
**spec:  
 replicas:** 1  
 **selector:  
 matchLabels:  
 app.kubernetes.io/name:** ingress-nginx  
 **app.kubernetes.io/part-of:** ingress-nginx  
 **template:  
 metadata:  
 labels:  
 app.kubernetes.io/name:** ingress-nginx  
 **app.kubernetes.io/part-of:** ingress-nginx  
 **annotations:  
 prometheus.io/port: "10254"  
 prometheus.io/scrape: "true"  
 spec:  
 terminationGracePeriodSeconds:** 300  
 **serviceAccountName:** nginx-ingress-serviceaccount  
 **nodeSelector:  
 kubernetes.io/os:** linux  
 **type:** ingress  
 **hostNetwork:** true  
 **containers:** - **name:** nginx-ingress-controller  
 **image:** quay.io/kubernetes-ingress-controller/nginx-ingress-controller:0.26.1  
 **args:** - /nginx-ingress-controller  
 - --configmap=$(POD\_NAMESPACE)/nginx-configuration  
 - --tcp-services-configmap=$(POD\_NAMESPACE)/tcp-services  
 - --udp-services-configmap=$(POD\_NAMESPACE)/udp-services  
 - --publish-service=$(POD\_NAMESPACE)/ingress-nginx  
 - --annotations-prefix=nginx.ingress.kubernetes.io  
 **securityContext:  
 allowPrivilegeEscalation:** true  
 **capabilities:  
 drop:** - ALL  
 **add:** - NET\_BIND\_SERVICE  
 **runAsUser:** 33  
 **env:** - **name:** POD\_NAME  
 **valueFrom:  
 fieldRef:  
 fieldPath:** metadata.name  
 - **name:** POD\_NAMESPACE  
 **valueFrom:  
 fieldRef:  
 fieldPath:** metadata.namespace  
 **ports:** - **name:** http  
 **containerPort:** 80  
 **protocol:** TCP  
 - **name:** https  
 **containerPort:** 443  
 **protocol:** TCP  
 **livenessProbe:  
 failureThreshold:** 3  
 **httpGet:  
 path:** /healthz  
 **port:** 10254  
 **scheme:** HTTP  
 **initialDelaySeconds:** 10  
 **periodSeconds:** 10  
 **successThreshold:** 1  
 **timeoutSeconds:** 10  
 **readinessProbe:  
 failureThreshold:** 3  
 **httpGet:  
 path:** /healthz  
 **port:** 10254  
 **scheme:** HTTP  
 **periodSeconds:** 10  
 **successThreshold:** 1  
 **timeoutSeconds:** 10  
 **lifecycle:  
 preStop:  
 exec:  
 command:** - /wait-shutdown  
---

以上配置地址

<https://raw.githubusercontent.com/kubernetes/ingress-nginx/master/deploy/static/mandatory.yaml>

对该资源配置文件添加了如下内容

serviceAccountName: nginx-ingress-serviceaccount

nodeSelector:

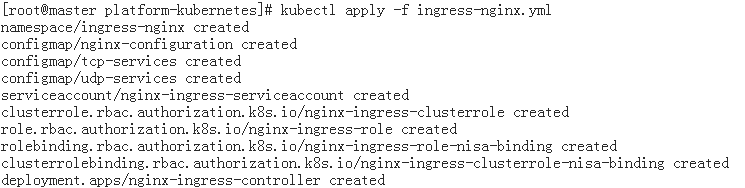
kubernetes.io/os: linux

type: ingress # 根据节点标签选择器，部署到node1上

hostNetwork: true # 集群外部使用node1的IP + 端口访问服务

1. 创建资源

kubectl apply -f ingress-nginx.yml



1. 查看是否创建成功

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



如果状态为镜像拉取失败，可以删除资源后，在node1节点先进行镜像拉取再创建资源

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

1. 浏览器访问<http://192.168.84.48> 此时可以看到页面为404效果
2. 创建Ingress资源配置文件

vim ingress.yml

内容如下：

**apiVersion:** networking.k8s.io/v1beta1  
**kind:** Ingress  
**metadata:  
 name:** platform-ingress  
**spec:  
 rules:** - **host:** 192.168.84.48  
 **http:  
 paths:** - **path:** /  
 **backend:  
 serviceName:** platform-gateway-service  
 **servicePort:** 8301

7、创建Ingress资源

kubectl apply -f ingress.yml

1. 打开C:\Windows\System32\drivers\etc的hosts文件添加本地域名解析

192.168.84.48 cloud.api.com

浏览器访问<http://cloud.api.com/auth/captcha?key=666> 测试验证码生成

## 部署前端工程

1、修改.env.production文件,，VUE\_APP\_BASE\_API地址改为Ingress的host

# just a flag

ENV = 'production'

# base api

VUE\_APP\_BASE\_API = 'http://cloud.api.com/'

2、前端工程打包

npm run build:prod

3、构建前端镜像，推送镜像至Harbor

4、在node1或者node2节点/home/platform-cloud文件夹下创建docker-compose

mkdir platform-cloud

vim platform-cloud-manage.yml

内容如下：

**version: '3'  
  
services:  
 platform-cloud-manage:  
 image:** docker.oumasoft.com/platform/platform-cloud-manage:2.0-RELEASE  
 **container\_name:** platform-cloud-manage  
 **volumes:** - **"/platform/log:/log"  
 environment:** - **"TZ=Asia/Shanghai"  
 ports:** - 9527:80  
 **restart:** on-failure

5、启动容器

docker-compose -f platform-cloud-manage.yml up –d

6、浏览器访问<http://192.168.84.48:9527>

## 部署Spring Boot Admin

1、