**一 helm部署**

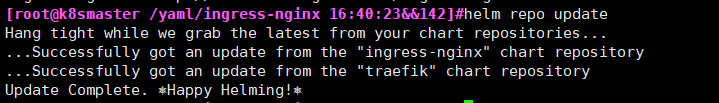
**1 官方文档**

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

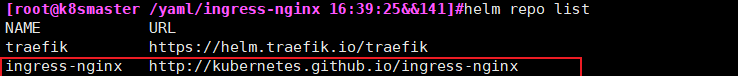
<https://github.com/kubernetes/ingress-nginx/blob/main/docs/deploy/index.md>

**2 添加仓库**

|  |
| --- |
| **helm repo add ingress-nginx http://kubernetes.github.io/ingress-nginx**  **helm repo update** |



|  |
| --- |
| **helm repo list** |

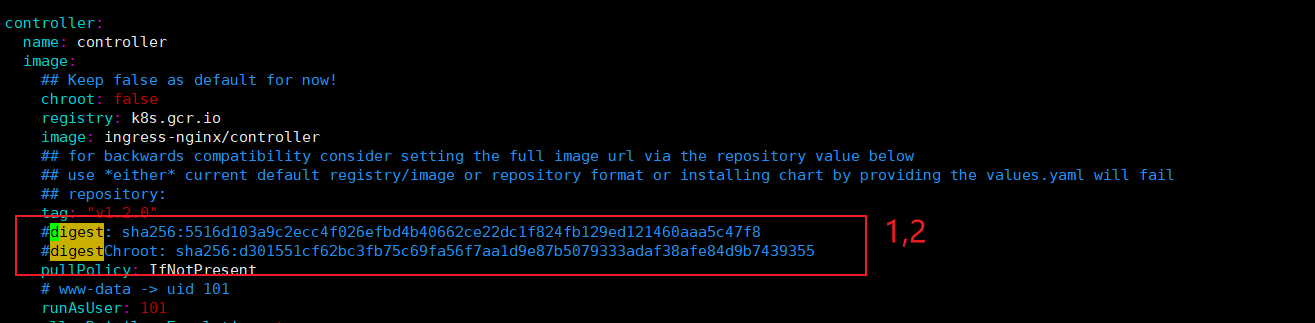


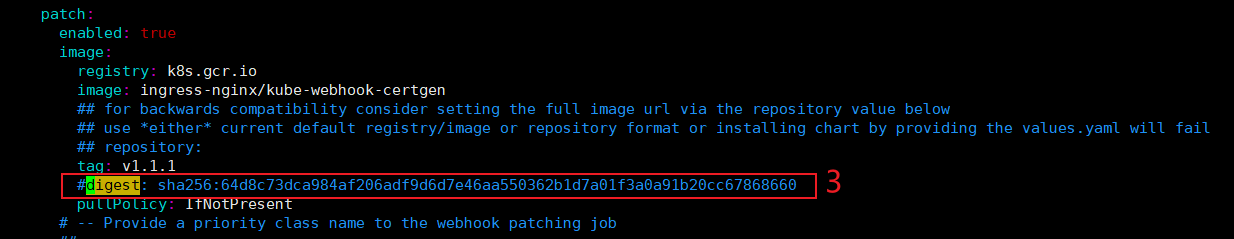
**3 拉取chart并解压修改配置**

|  |
| --- |
| **helm pull ingress-nginx/ingress-nginx**  **tar xf ingress-nginx-4.1.3.tgz**  **cd ingress-nginx** |

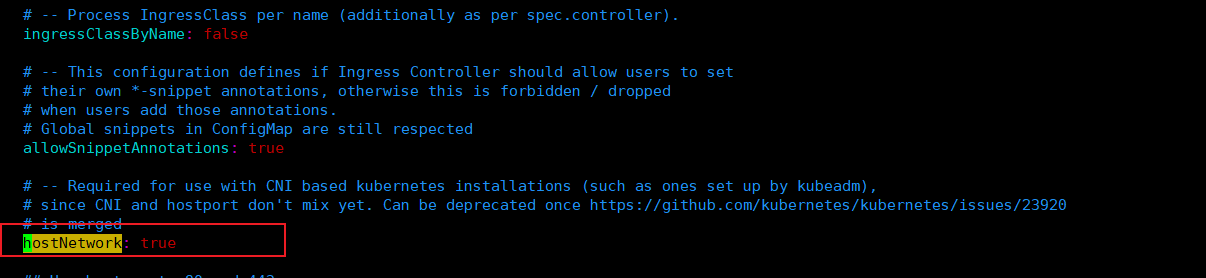
**编辑values.yaml**

**3.1 去掉hash(3处)**

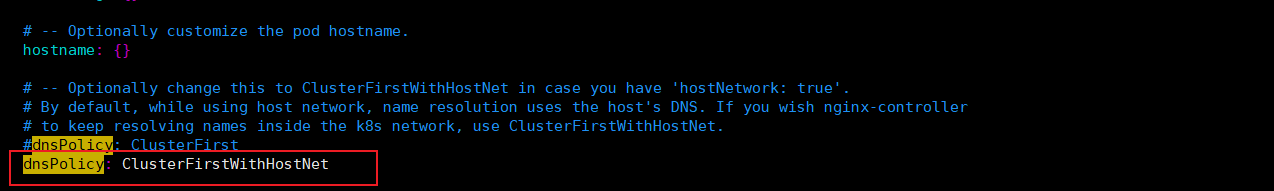




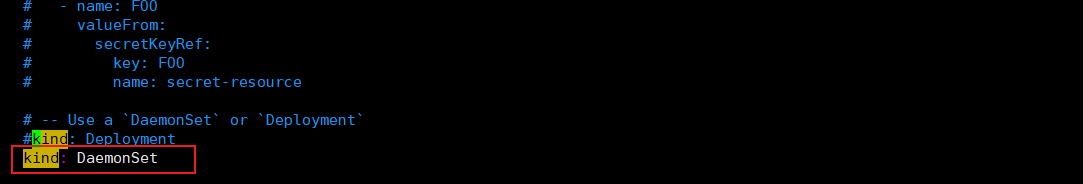
**3.2 hostNetwork: false 改为 hostNetwork: true**



**3.3 dnsPolicy: ClusterFirst 改为 dnsPolicy: ClusterFirstWithHostNet**



**3.4 kind: Deployment 改为 kind: DaemonSet**



**3.5 type: LoadBalance 改为 type: ClusterIP**

LoadBalancer #通过弹性IP向外部暴露服务

ClusterlP #向集群内部暴露服务



**3.6 添加ingress: "true"**

**部署到指定节点**

Node labels for controller pod assignment



**4 设置标签**

|  |
| --- |
| **kubectl label nodes k8snode2 ingress=true**  k8snode2 为节点标签名称 |

|  |
| --- |
| **kubectl get nodes --show-labels** |

新增一个节点部署ingress pod

|  |
| --- |
| **kubectl label nodes k8snode1 ingress=true** #设置标签    此时k8snode1 k8snode2 都被标记  设置标签后会自行在k8snode1 启一个pod服务 |

|  |
| --- |
| **kubectl label nodes k8snode1 ingress-** #取消标签    此时k8snode1会自行销毁pod |

**5 部署 ingress-nginx**

[root@k8smaster /helm/ingress-nginx]#**helm install ingress-nginx -n ingress-nginx . --create-namespace** #部署

**helm upgrade ingress-nginx -n ingress-nginx .**  #升级更新

**helm uninstall ingress-nginx -n ingress-nginx** #卸载

|  |
| --- |
| Release "ingress-nginx" has been upgraded. Happy Helming!  NAME: ingress-nginx  LAST DEPLOYED: Wed Jun 8 16:37:35 2022  NAMESPACE: ingress-nginx  STATUS: deployed  REVISION: 2  TEST SUITE: None  NOTES:  The ingress-nginx controller has been installed.  Get the application URL by running these commands:  export POD\_NAME=$(kubectl --namespace ingress-nginx get pods -o jsonpath="{.items[0].metadata.name}" -l "app=ingress-nginx,component=controller,release=ingress-nginx")  kubectl --namespace ingress-nginx port-forward $POD\_NAME 8080:80  echo "Visit http://127.0.0.1:8080 to access your application."  An example Ingress that makes use of the controller:  apiVersion: networking.k8s.io/v1  kind: Ingress  metadata:  name: example  namespace: foo  spec:  ingressClassName: nginx  rules:  - host: www.example.com  http:  paths:  - pathType: Prefix  backend:  service:  name: exampleService  port:  number: 80  path: /  # This section is only required if TLS is to be enabled for the Ingress  tls:  - hosts:  - www.example.com  secretName: example-tls  If TLS is enabled for the Ingress, a Secret containing the certificate and key must also be provided:  apiVersion: v1  kind: Secret  metadata:  name: example-tls  namespace: foo  data:  tls.crt: <base64 encoded cert>  tls.key: <base64 encoded key>  type: kubernetes.io/tls |

**6 实例**

**whoami.yaml**

|  |
| --- |
| ---  apiVersion: v1  kind: Pod  metadata:  name: whoami-ing  labels:  app: whoami-ing  spec:  containers:  - name: whoami-ing  image: traefik/whoami:latest  ports:  - containerPort: 80  ---  apiVersion: v1  kind: Service  metadata:  name: whoami-ing  spec:  ports:  - port: 80  protocol: TCP  targetPort: 80  selector:  app: whoami-ing  type: ClusterIP |

**whoami-ingressroute.yaml**

|  |
| --- |
| apiVersion: networking.k8s.io/v1  kind: Ingress  metadata:  name: whoami-test-ingress-route  spec:  ingressClassName: nginx  rules:  - host: www.abc.com  http:  paths:  - pathType: Prefix  path: /  backend:  service:  name: whoami-ing  port:  number: 80 |

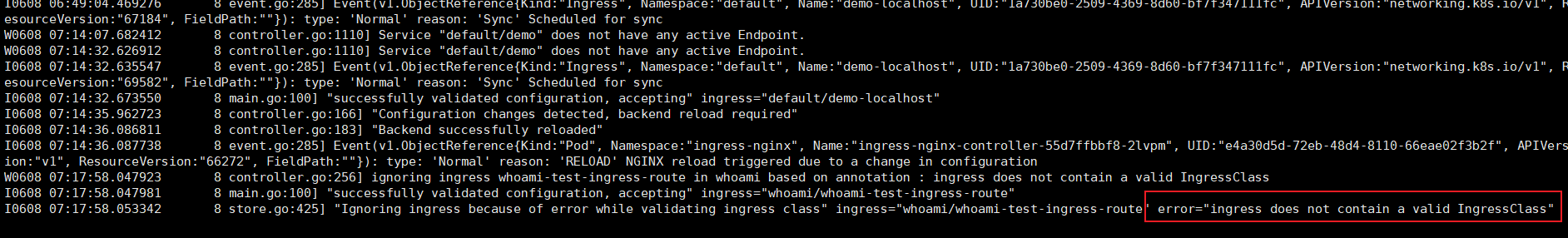
**ingressClassName: nginx 要加这条 否则404**

**并报以下错**

error="ingress does not contain a valid IngressClass"

**Ingress pod日志查看**

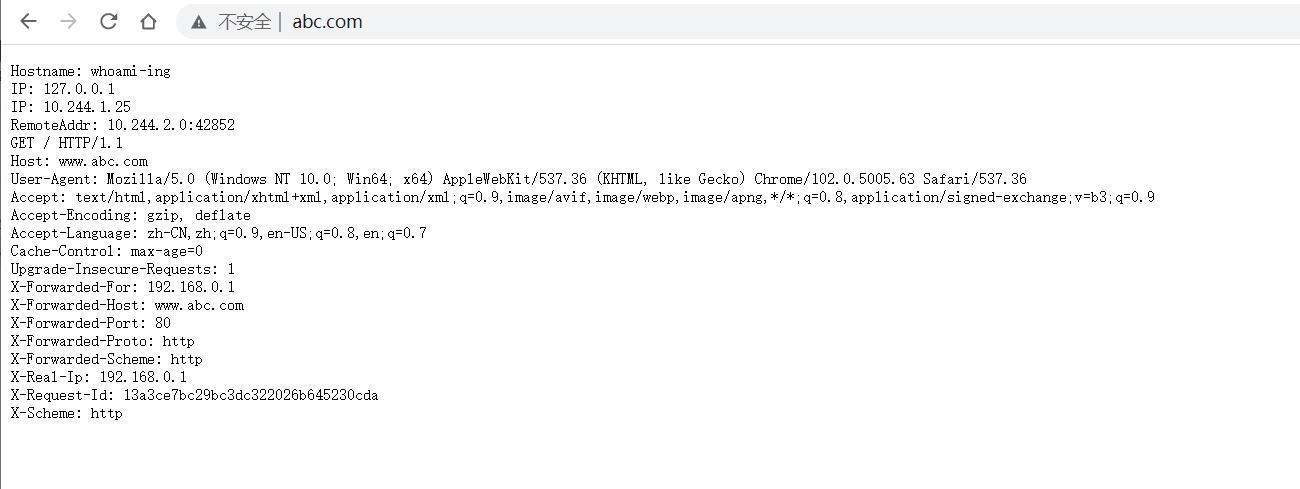
k logs -f ingress-nginx-controller-55d7ffbbf8-2lvpm -n ingress-nginx



本地hosts绑定设置标签的节点（k8snode2）IP

192.168.91.102 www.abc.com

访问 <http://www.abc.com/>



**二 yaml文件部署**

**1 官方文档**

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

<https://github.com/kubernetes/ingress-nginx/blob/main/docs/deploy/index.md>

**2 下载yaml**

**wget <https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.2.0/deploy/static/provider/cloud/deploy.yaml>**

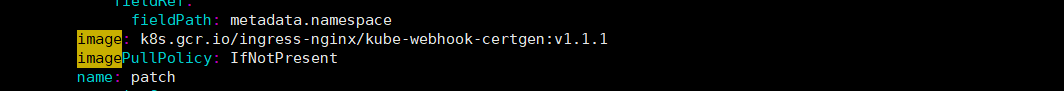
**3 修改deploy.yaml**

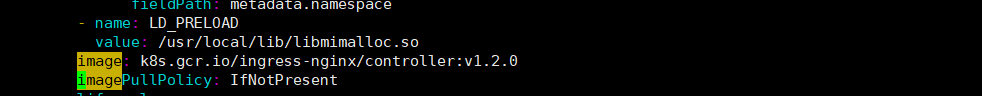
**3.1 修改image**

@sha256:d8196e3bc1e72547c5dec66d6556c0ff92a23f6d0919b206be170bc90d5f9185 去掉尾部

**总共有三处需要去掉**

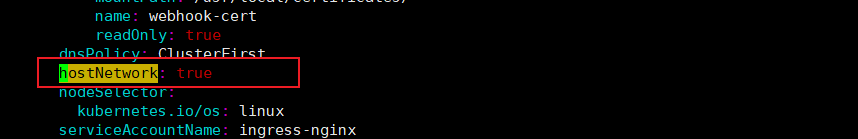
**搜索256**



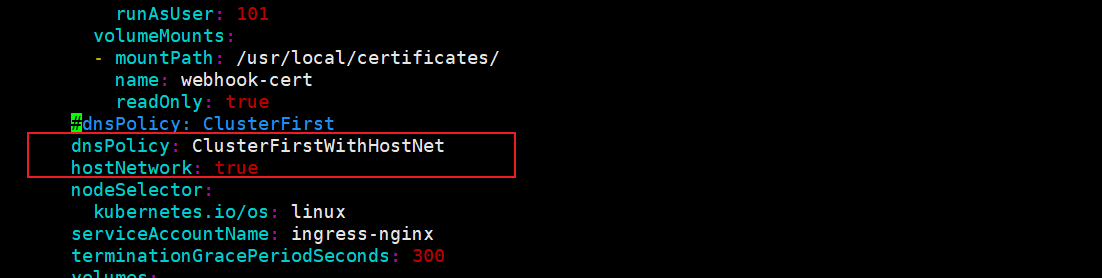


**3.2 hostNetwork: true**

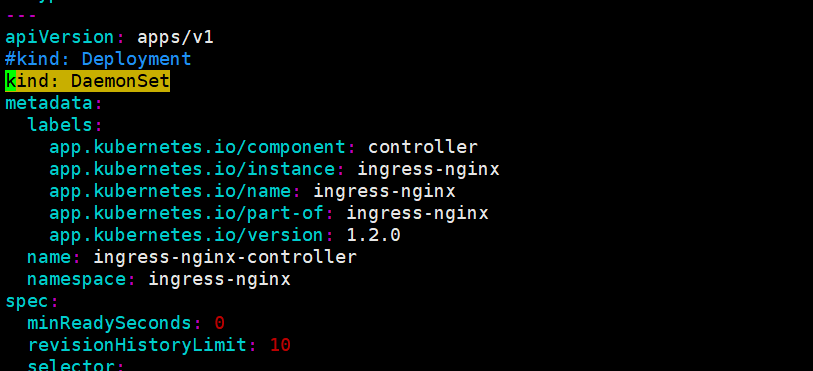
**在dnsPolicy: ClusterFirst 下面添加hostNetwork: true**



**3.3 dnsPolicy: ClusterFirst 改为 dnsPolicy: ClusterFirstWithHostNet**

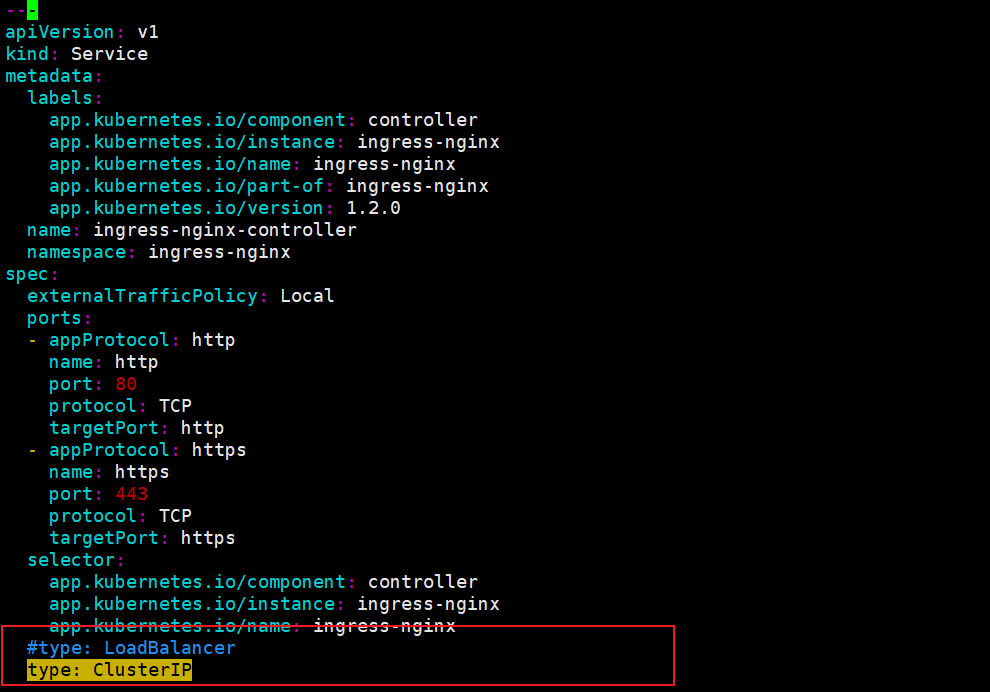


**3.4 kind: Deployment 改为 kind: DaemonSet**



**3.5修改Service （ingress-nginx-controller）**

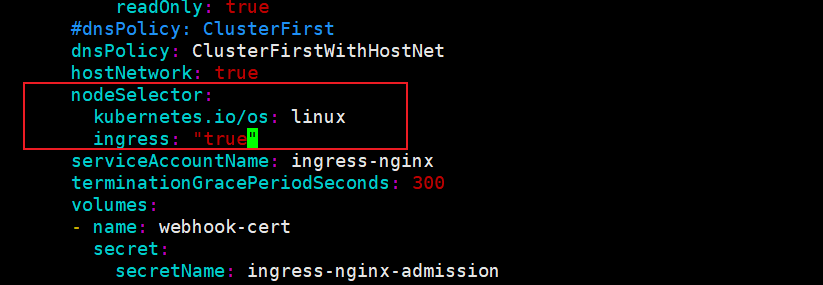
**kind: Service**



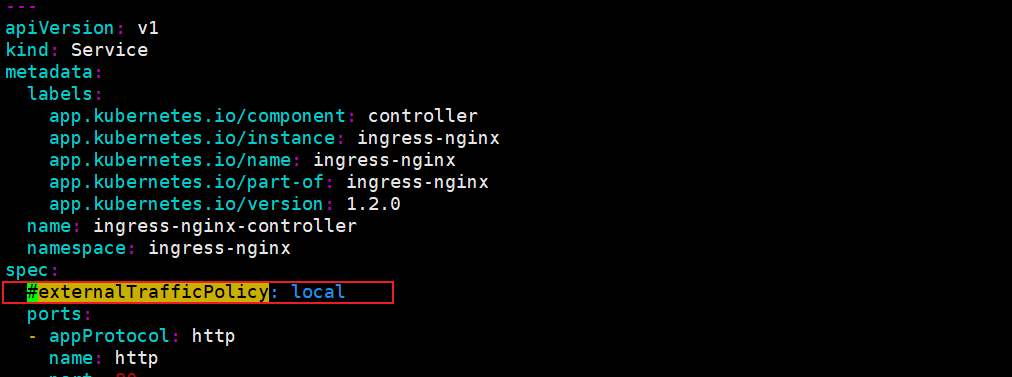
**type: LoadBalancer --》type: ClusterIP**

**3.6 添加ingress: "true"**

**DaemonSet -》nodeSelector**



**3.7 注释externalTrafficPolicy: local**



否则会报

The Service "ingress-nginx-controller" is invalid: spec.externalTrafficPolicy: Invalid value: "local": ExternalTrafficPolicy must be empty, Cluster or Local

**4 部署**

**4.0 work节点导入镜像**

**# docker pull k8s.gcr.io/ingress-nginx/controller:v1.2.0**

**# docker pull k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.1.1**

**# docker save -o ingress-nginx-controller.v1.2.0.tar k8s.gcr.io/ingress-nginx/controller:v1.2.0**

**# docker save -o ingress-nginx-kube-webhook-certgen.v1.1.1.tar k8s.gcr.io/ingress-nginx/kube-webhook-certgen:v1.1.1**

#**docker load -i ingress-nginx-controller.v1.2.0.tar**

#**docker load -i ingress-nginx-kube-webhook-certgen.v1.1.1.tar**

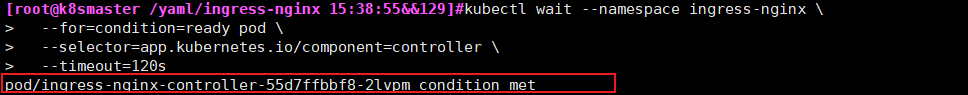
**4.1 apply deploy.yaml**

**kubectl apply -f deploy.yaml**

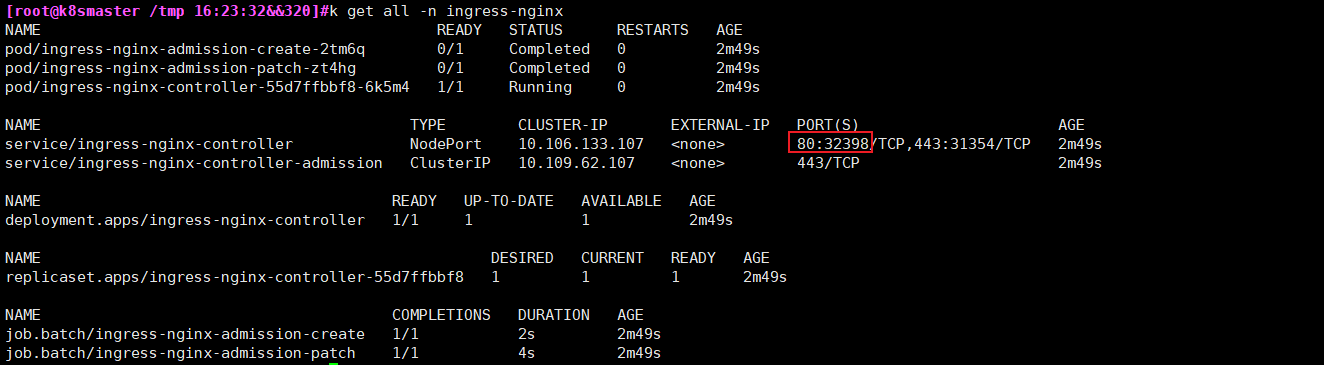
**4.2 检查**

**kubectl get pods --namespace=ingress-nginx**

|  |
| --- |
| **kubectl wait --namespace ingress-nginx \**  **--for=condition=ready pod \**  **--selector=app.kubernetes.io/component=controller \**  **--timeout=120s** |



#k get all -n ingress-nginx



**5 实例**

**whoami.yaml**

|  |
| --- |
| ---  apiVersion: v1  kind: Pod  metadata:  name: whoami-ing  labels:  app: whoami-ing  spec:  containers:  - name: whoami-ing  image: traefik/whoami:latest  ports:  - containerPort: 80  ---  apiVersion: v1  kind: Service  metadata:  name: whoami-ing  spec:  ports:  - port: 80  protocol: TCP  targetPort: 80  selector:  app: whoami-ing  type: ClusterIP |

**whoami-ingressroute.yaml**

|  |
| --- |
| apiVersion: networking.k8s.io/v1  kind: Ingress  metadata:  name: whoami-test-ingress-route  spec:  ingressClassName: nginx  rules:  - host: www.abc.com  http:  paths:  - pathType: Prefix  path: /  backend:  service:  name: whoami-ing  port:  number: 80 |

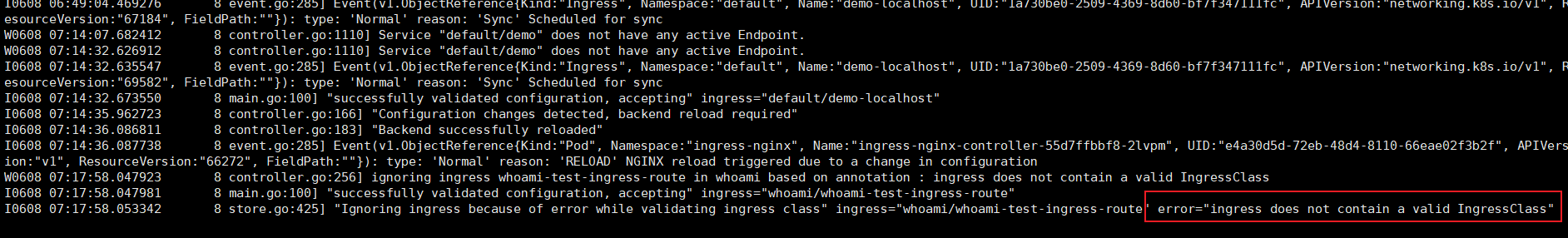
**ingressClassName: nginx 要加这条 否则404**

**并报以下错**

error="ingress does not contain a valid IngressClass"

**Ingress pod日志查看**

k logs -f ingress-nginx-controller-55d7ffbbf8-2lvpm -n ingress-nginx

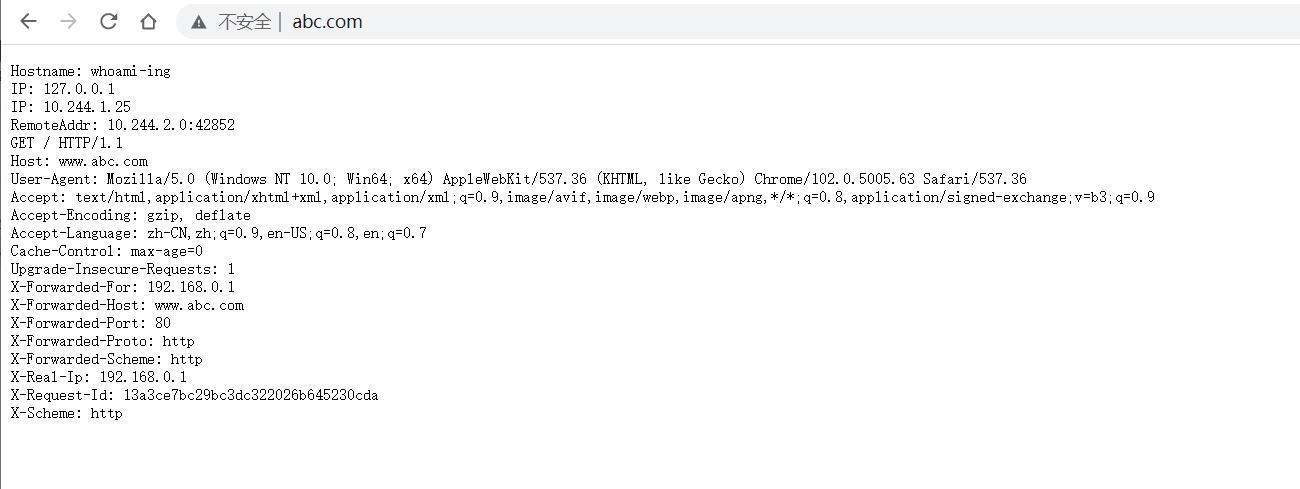


本地hosts绑定任意节点IP

192.168.91.100 www.abc.com

访问 <http://www.abc.com/>

<http://www.abc.com:32398/>

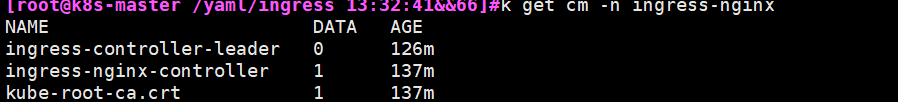




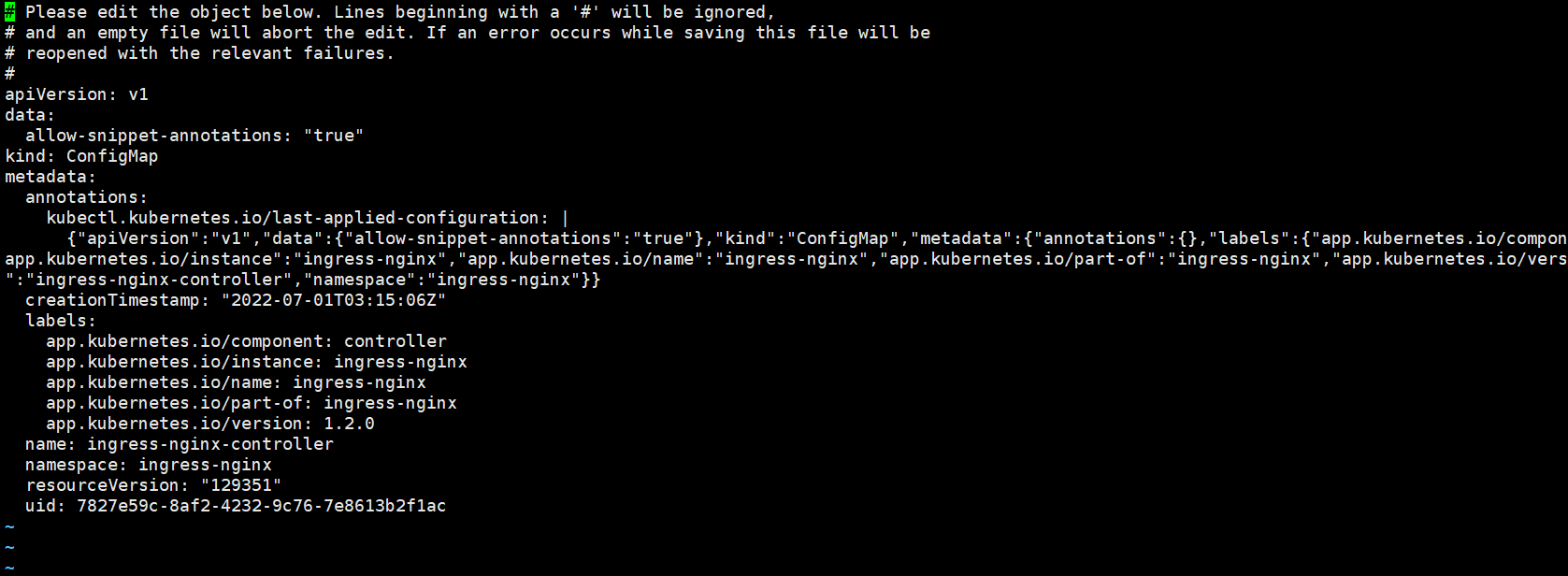
**6 全局配置**

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

k get cm -n ingress-nginx



k edit cm ingress-nginx-controller -n ingress-nginx



需要删除pod 重新生成pod

**7 路径重写**

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

**8、限速**

<https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/annotations/#rate-limiting>

**9、会话保持-Session亲和性**

<https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/annotations/#session-affinity>