# KubeSphere 部署

CreatedBy: 彭玲 CreatedOn: 2021/8/27



飞尚 fs-devops 文档中心资料 (SVN)：http://10.8.30.22/fs-devops/doc/services/kubesphere部署.pdf

# k8s 默认存储类创建

## 1. nfs 中创建挂载路径

详见 NFS 搭建。

## 2. 创建默认存储类 (storageclass.yaml)

StorageClass 的`yaml`配置如下：

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: nfs-storage
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
provisioner: fuseim.pri/ifs
reclaimPolicy: Retain
```

创建`storageclass.yaml`文件，执行`kubectl apply -f storageclass.yaml`创建 **默认** `存储类`：

```
# <Error> K8s API 服务器连接异常
root@node38:/home/anxin/pengling/k8s/kubesphere# kubectl apply -f
storageclass.yaml
The connection to the server 10.8.30.38:6443 was refused - did you specify the
right host or port?
# <Solution> STEP 1. 关闭系统交换区
anxin@node38:~$ sudo swapoff -a
```

```
# <Solution> STEP 2. 重启 kubelet
anxin@node38:~$ systemctl restart kubelet
# 重新执行 kubectl apply -f storageclass.yaml
anxin@node38:~/pengling/k8s/kubesphere$ kubectl apply -f storageclass.yaml
serviceaccount/nfs-client-provisioner created
clusterrole.rbac.authorization.k8s.io/nfs-client-provisioner-runner created
clusterrolebinding.rbac.authorization.k8s.io/run-nfs-provisioner created
deployment.apps/nfs-client-provisioner created
storageclass.storage.k8s.io/nfs-storage created
```

# K8s 内最小化安装 KubeSphere

## Prerequisites

- 如需在 Kubernetes 上安装 KubeSphere v3.1.1，您的 Kubernetes 版本必须为：1.17.x、1.18.x、1.19.x 或 1.20.x。
- 确保您的机器满足最低硬件要求：CPU > 1 核，内存 > 2 GB。
- 在安装之前，需要配置 Kubernetes 集群中的**默认**存储类型。

## 部署 KubeSphere

确保您的机器满足安装的前提条件之后，可以按照以下步骤安装 KubeSphere。

1、执行以下命令开始安装：

```
kubectl apply -f https://github.com/kubesphere/ks-
installer/releases/download/v3.1.1/kubesphere-installer.yaml

kubectl apply -f https://github.com/kubesphere/ks-
installer/releases/download/v3.1.1/cluster-configuration.yaml
```

2、检查安装日志：

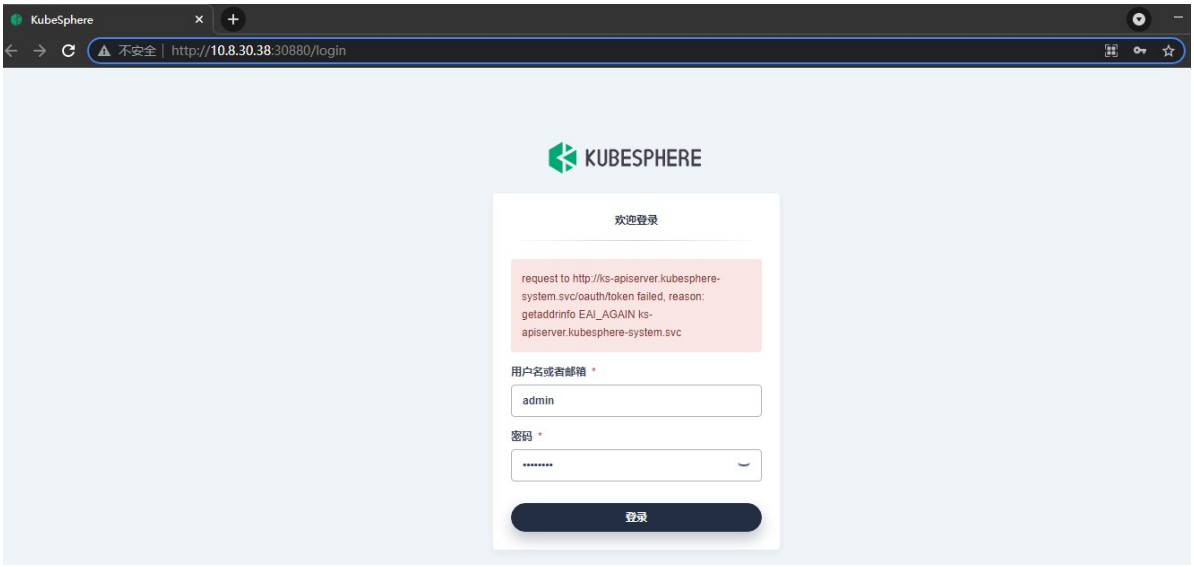
```
kubectl logs -n kubesphere-system $(kubectl get pod -n kubesphere-system -l
app=ks-install -o jsonpath='{.items[0].metadata.name}') -f
```

3、使用 `kubectl get pod --all-namespaces` 查看所有 Pod 是否在 KubeSphere 的相关命名空间中正常运行。如果是，请通过以下命令检查控制台的端口（默认为 `30880`）：

```
kubectl get svc/ks-console -n kubesphere-system
```

4、确保在安全组中打开了端口 `30880`，并通过 NodePort `(IP:30880)` 使用默认帐户和密码 `(admin/P@88w0rd)` 访问 Web 控制台。

5、登录控制台后，可以在服务组件中检查不同组件的状态。如果要使用相关服务，可能需要等待某些组件启动并运行。




访问 http://10.8.30.38:30880/login 并使用默认帐户和密码 (admin/P@88w0rd) 登录，问题：

```
request to http://ks-apiserver.kubesphere-system.svc/oauth/token failed, reason:
getaddrinfo EAI_AGAIN ks-apiserver.kubesphere-system.svc
```

说明：基本上是 node38 服务器集群的问题。

# 验证安装

## 查看 pod

```
anxin@node38:~$ kubectl get po -n kubesphere-system
NAME                              READY   STATUS    RESTARTS   AGE
ks-installer-7bd6b699df-l6lf4   1/1     Running   0          15d
```

## 查看 svc

```
anxin@node38:~$ kubectl get svc/ks-console -n kubesphere-system
Error from server (NotFound): services "ks-console" not found
```

问题解决：重启通过 ks-installer 安装的 kubesphere-system 服务 ( ks-apiserver 、 ks-console 、 ks-controller-manager )。

```
# 查看 kubesphere-system 相关 pod
anxin@node38:~/pengling/k8s/kubesphere$ kubectl get po -n kubesphere-system
NAME                                    READY   STATUS    RESTARTS   AGE
ks-apiserver-6778dcc677-gfcvk           1/1     Running   0          2d
ks-console-74cf8b9487-xzmss             1/1     Running   0          2d
ks-controller-manager-788c89f994-9lmb6  1/1     Running   0          2d
ks-installer-7bd6b699df-fhbpr           1/1     Running   0          2d1h
anxin@node38:~/pengling/k8s/kubesphere$
# 查看 kubesphere-system 服务
anxin@node38:~/pengling/k8s/kubesphere$ kubectl get svc -n kubesphere-system
NAME                    TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)
AGE
ks-apiserver            ClusterIP   10.1.28.118   <none>        80/TCP
2d1h
```

```
ks-console              NodePort    10.1.95.112    <none>        80:30880/TCP
2d1h
ks-controller-manager   ClusterIP   10.1.0.86      <none>        443/TCP
2d1h
anxin@node38:~/pengling/k8s/kubesphere$
# 查看 ks-console 服务
anxin@node38:~/pengling/k8s/kubesphere$ kubectl get svc/ks-console -n
kubesphere-system
NAME         TYPE       CLUSTER-IP    EXTERNAL-IP   PORT(S)        AGE
ks-console   NodePort   10.1.95.112   <none>        80:30880/TCP   2d1h
```

## 附：storageclass.yaml

```yaml
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: nfs-client-provisioner
---
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: nfs-client-provisioner-runner
  namespace: default
rules:
  - apiGroups: [""]
    resources: ["nodes"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["persistentvolumes"]
    verbs: ["get", "list", "watch", "create", "delete"]
  - apiGroups: [""]
    resources: ["persistentvolumeclaims"]
    verbs: ["get", "list", "watch", "update"]
  - apiGroups: ["storage.k8s.io"]
    resources: ["storageclasses"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["events"]
    verbs: ["watch", "create", "update", "patch"]
  - apiGroups: [""]
    resources: ["services", "endpoints"]
    verbs: ["get","create","list", "watch","update"]
  - apiGroups: ["extensions"]
    resources: ["podsecuritypolicies"]
    resourceNames: ["nfs-provisioner"]
    verbs: ["use"]
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: run-nfs-provisioner
subjects:
  - kind: ServiceAccount
    name: nfs-provisioner
```

```yaml
    namespace: default
roleRef:
  kind: ClusterRole
  name: nfs-provisioner-runner
  apiGroup: rbac.authorization.k8s.io
---
kind: Deployment
apiVersion: apps/v1
metadata:
  name: nfs-client-provisioner
spec:
  selector:
    matchLabels:
      app: nfs-client-provisioner
  replicas: 1
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: nfs-client-provisioner
    spec:
      serviceAccount: nfs-provisioner
      containers:
        - name: nfs-client-provisioner
          image: quay.io/external_storage/nfs-client-provisioner:latest
          imagePullPolicy: IfNotPresent
          volumeMounts:
            - name: nfs-client
              mountPath: /persistentvolumes
          env:
            - name: PROVISIONER_NAME
              value: fuseim.pri/ifs
            - name: NFS_SERVER
              value: 10.8.30.199 #此处修改为nfs服务器ip
            - name: NFS_PATH
              value: /nfsdata #这里为nfs共享目录
      volumes:
        - name: nfs-client
          nfs:
            server: 10.8.30.199 #此处修改为nfs服务器ip
            path: /nfsdata #这里为nfs共享目录
---
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: nfs-storage
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
provisioner: fuseim.pri/ifs
reclaimPolicy: Retain
```