



课程推荐，提供跪舔式售后服务：

K8s 高薪全栈架构师课程：<https://edu.51cto.com/sd/518e5>

K8s CKA 认证课程：<https://edu.51cto.com/sd/fbbc8>

K8s CKS 认证课程：<https://edu.51cto.com/sd/affb5>

K8s 全栈架构师+CKA 套餐：<https://edu.51cto.com/topic/4973.html>

超级套购：

K8s 全栈架构师+CKA+CKS：<https://edu.51cto.com/topic/5174.html>

关注宽哥得永生：

<https://edu.51cto.com/lecturer/11062970.html?type=2>

本实验需要 K8s 版本 1.17 以上

## 1. 下载安装文件

```
[root@k8s-master01 ~]# mkdir csi-hostpath
[root@k8s-master01 ~]# cd csi-hostpath/
[root@k8s-master01 csi-hostpath]#
[root@k8s-master01 csi-hostpath]# ls
[root@k8s-master01 csi-hostpath]# git clone
https://gitee.com/dukuan/k8s-ha-install.git
Cloning into 'k8s-ha-install'...
remote: Enumerating objects: 588, done.
remote: Counting objects: 100% (588/588), done.
remote: Compressing objects: 100% (327/327), done.
remote: Total 588 (delta 221), reused 588 (delta 221), pack-reused 0
Receiving objects: 100% (588/588), 19.58 MiB | 10.00 MiB/s, done.
Resolving deltas: 100% (221/221), done.
[root@k8s-master01 csi-hostpath]# cd k8s-ha-install/
[root@k8s-master01 k8s-ha-install]# git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/manual-installation
remotes/origin/manual-installation-v1.16.x
remotes/origin/manual-installation-v1.17.x
remotes/origin/manual-installation-v1.18.x
remotes/origin/manual-installation-v1.19.x
remotes/origin/manual-installation-v1.20.x
remotes/origin/master
[root@k8s-master01 k8s-ha-install]# git checkout manual-installation-
v1.20.x-csi-hostpath
Branch manual-installation-v1.20.x set up to track remote branch manual-
installation-v1.20.x from origin.
Switched to a new branch 'manual-installation-v1.20.x'
[root@k8s-master01 k8s-ha-install]# ls
bootstrap calico CoreDNS dashboard kube-proxy metrics-server-0.4.x
metrics-server-0.4.x-kubeadm pki snapshotter
```

```
[root@k8s-master01 k8s-ha-install]# kubectl create -f snapshotter/
```

## 2. 查看 VolumeSnapshot CRDs 和 Snapshot Controller 是否已经安装

```
$ kubectl get volumesnapshotclasses.snapshot.storage.k8s.io
$ kubectl get volumesnapshots.snapshot.storage.k8s.io
$ kubectl get volumesnapshotcontents.snapshot.storage.k8s.io
```

如果返回值不是 `error: the server doesn't have a resource type "volumesnapshotclasses"` 表示安装成功

```
kubectl get pods --all-namespaces -
o=jsonpath='{range .items[*]}{"\n"}{range .spec.containers[*]}{.image}{",
"}{end}{end}' | grep snapshot-controller
```

有返回值说明 Snapshot Controller 已经安装

## 3. 安装 csi-hostpath

```
[root@k8s-master01 k8s-ha-install]# ls
bootstrap calico CoreDNS csi-hostpath dashboard kube-proxy metrics-
server-0.4.x metrics-server-0.4.x-kubeadm pki snapshotter
[root@k8s-master01 k8s-ha-install]# cd csi-hostpath/
[root@k8s-master01 csi-hostpath]#
[root@k8s-master01 csi-hostpath]# ls
csi-hostpath-attacher.yaml csi-hostpath-provisioner.yaml csi-
hostpath-snapshotter.yaml external-provisioner-rbac.yaml
csi-hostpath-driverinfo.yaml csi-hostpath-resizer.yaml csi-
hostpath-testing.yaml external-resizer-rbac.yaml
csi-hostpath-plugin.yaml csi-hostpath-snapshotclass.yaml external-
attacher-rbac.yaml rbac-csi-snapshotter.yaml
[root@k8s-master01 csi-hostpath]# kubectl apply -f .
statefulset.apps/csi-hostpath-attacher created
...
rolebinding.rbac.authorization.k8s.io/external-snapshotter-
leaderelection created
```

查看状态

```
[root@k8s-master01 csi-hostpath]# kubectl get po
NAME                                READY   STATUS    RESTARTS   AGE
csi-hostpath-attacher-0             1/1     Running   0           3m26s
csi-hostpath-provisioner-0          1/1     Running   0           3m26s
csi-hostpath-resizer-0              1/1     Running   0           3m26s
csi-hostpath-snapshotter-0          1/1     Running   0           3m26s
csi-hostpath-socat-0                1/1     Running   0           3m26s
csi-hostpathplugin-0                5/5     Running   0           3m26s
```

## 4. 创建 storageClass

```
[root@k8s-master01 csi-hostpath]# cd examples/
[root@k8s-master01 examples]# ls
csi-app-inline.yaml csi-block-pvc-restore.yaml csi-pod-block.yaml
csi-restore.yaml    csi-storageclass.yaml
csi-app.yaml         csi-block-pvc-snapshot.yaml csi-pvc-block.yaml csi-
snapshot-v1beta1.yaml
csi-block-clone.yaml csi-clone.yaml          csi-pvc.yaml      csi-
snapshot.yaml
[root@k8s-master01 examples]# kubectl create -f csi-storageclass.yaml
storageclass.storage.k8s.io/csi-hostpath-sc created
[root@k8s-master01 examples]# kubectl get sc
NAME                                PROVISIONER          RECLAIMPOLICY   VOLUMEBINDINGMODE
ALLOWVOLUMEEXPANSION  AGE
csi-hostpath-sc        hostpath.csi.k8s.io  Delete          Immediate
true                    5s
```

## 5. 创建测试 PVC

```
[root@k8s-master01 examples]# kubectl create -f csi-pvc.yaml
persistentvolumeclaim/csi-pvc created
You have new mail in /var/spool/mail/root
```

```
[root@k8s-master01 examples]# kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS
csi-pvc	Bound	pvc-587b559b-ee3a-4dfb-9444-f12c258cee9e	1Gi	

```
[root@k8s-master01 examples]# kubectl get pv
```

NAME	CAPACITY	ACCESS	MODES	RECLAIM
pvc-587b559b-ee3a-4dfb-9444-f12c258cee9e	1Gi		RWO	Delete

```
[root@k8s-master01 examples]# kubectl get pv
```

POLICY	STATUS	CLAIM	STORAGECLASS	REASON	AGE
Bound	default/csi-pvc	csi-hostpath-sc		17s	

## 6. 创建测试应用

```
[root@k8s-master01 csi-hostpath]# cd examples/
[root@k8s-master01 examples]# ls
csi-app-inline.yaml  csi-block-pvc-restore.yaml  csi-pod-block.yaml
csi-restore.yaml      csi-storageclass.yaml
csi-app.yaml          csi-block-pvc-snapshot.yaml  csi-pvc-block.yaml  csi-
snapshot-v1beta1.yaml
csi-block-clone.yaml  csi-clone.yaml              csi-pvc.yaml        csi-
snapshot.yaml
[root@k8s-master01 examples]# kubectl create -f csi-app.yaml
[root@k8s-master01 csi-hostpath]# kubectl get po
```

NAME	READY	STATUS	RESTARTS	AGE
csi-hostpath-attacher-0	1/1	Running	0	62s
csi-hostpath-provisioner-0	1/1	Running	0	13m
csi-hostpath-resizer-0	1/1	Running	0	13m
csi-hostpath-snapshotter-0	1/1	Running	0	13m
csi-hostpath-socat-0	1/1	Running	0	13m
csi-hostpathplugin-0	5/5	Running	0	13m
my-csi-app	1/1	Running	0	9m45s

扩容: <https://edu.51cto.com/lecturer/11062970.html?type=2>

```
[root@k8s-master01 examples]# kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS
csi-pvc	Bound	pvc-1f259dcc-129f-4170-a06a-48fd588b7f5e	1Gi	

```
[root@k8s-master01 examples]# kubectl edit pvc csi-pvc
```

```
annotations:
  pv.kubernetes.io/bind-completed: "yes"
  pv.kubernetes.io/bound-by-controller: "yes"
  volume.beta.kubernetes.io/storage-provisioner: hostpath.csi.k8s.io
  kubernetes.io/change-cause: "resize"
creationTimestamp: "2021-03-17T03:47:03Z"
finalizers:
- kubernetes.io/pvc-protection
name: csi-pvc
namespace: default
resourceVersion: "12944139"
uid: 1f259dcc-129f-4170-a06a-48fd588b7f5e
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
  storageClassName: csi-hostpath-sc
  volumeMode: Filesystem
```

```
[root@k8s-master01 csi-hostpath]# kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS
csi-pvc	Bound	pvc-1f259dcc-129f-4170-a06a-48fd588b7f5e	2Gi	

```
[root@k8s-master01 csi-hostpath]# kubectl get pv
```

POLICY	STATUS	CLAIM	STORAGECLASS	REASON	AGE
Bound	default/csi-pvc	csi-hostpath-sc		24m	

---

#### patch 命令

```
[root@k8s-master01 csi-hostpath]# kubectl patch pvc csi-pvc -p  
'{"spec":{"resources":{"requests":{"storage": "3Gi"}}}}' --record  
persistentvolumeclaim/csi-pvc patched
```

<https://edu.51cto.com/lecturer/11062970.html?type=2>