

Installing the KumoScale CSI Driver

This page describes the installation of the KumoScale software and Kubernetes orchestration CSI plug-in (driver). The following steps are detailed at the links provided.

1. [Prepare for installation](#)
2. [Update the CSI driver installation file.](#)
3. [Install the KumoScale CSI driver.](#)
4. For Kubernetes 1.20 and higher, [install the snapshot plugins](#).
For Kubernetes 1.16 through 1.19, [Prepare the environment for features in beta](#)

Step 1. Prepare for Installation

Before you start the installation, prepare your environment:

1. Confirm your initiators and deployment meet the requirements listed in [Environment Requirements](#)
2. The KumoScale CSI driver must be loaded from a container repository and is provided by KIOXIA as a binary container. If you are using KumoScale for Managed Mode, these steps may already have been completed as part of [Download KumoScale Software and Prepare for Installation](#). Below are instructions for those who need to prepare the container. From the **Kubernetes_CSI** directory:

```
gunzip ks-csi-plugin-v3.22-<version>.tar.gz
```

to extract the file **ks-csi-plugin-v3.22-<version>.tar**

Log into your repository, load, tag, and push the image. Using Docker as an example:

```
docker login <yourlocalregistry>
username: <>
password: <>
docker load -i ks-csi-plugin-v3.22-<version>.tar
docker image tag registry.local:5000/ks-csi-plugin-v3.22-<version> <yourlocalregistry>/ks-install-operator-v3.22-<version>
docker push <yourlocalregistry>/ks-csi-plugin-v3.22-<version>
```

3. Copy the appropriate Kubernetes manifest file below to **ks-csi.yaml**. The manifest file you need to use is based on your version of Kubernetes
- Kubernetes orchestration release 1.20 or higher, **ks-csi-<KumoScale version>.yaml**. For example, ks-csi-v3.22-2098.yaml.

• Kubernetes orchestration release 1.16 through 1.19, **ks-csi1.16-19-<KumoScale version>.yaml** . For example, ks-csi.1.16-19-v3.22-2098.yaml.

• Kubernetes orchestration release 1.15, **ks-csi15.yaml**.

4. If kubelet is not running with the default root directory (the default root directory is /var/lib/kubelet), you will need to replace every occurrence of /var/lib/kubelet in the installation yaml with the root directory which is in use

5. Confirm your have access to the Provisioner Secret file for:

• **Managed Mode**, the KumoScale Provisioner service secret file **provisioner-secret.yaml** was created as part of [installation](#).

• **Appliance Mode**, you will need to create the secret. To do this, use the **provisioner-secret.yaml** included in the **Kubernetes_CSI** directory. Update this file with the KumoScale Provisioner URL and token or OIDC specification information. You may optionally also provide the *tenantID* of an existing Provisioner service tenant. Below is a table showing the parameters and possible values for each.
- | Argument | Description |
|-------------------------|---|
| url | Provisioner full URL. Applies to all authentication methods. |
| token | Provisioner valid authorization token; applies to Local and LDAP authentication mode only. |
| authServerTokenUrl | URL for the authorization server used for generating tokens; applies to OpenIDC authentication mode only. |
| provisionerClientID | Client ID of a client, which has a service account role of ADMIN at the provisioner resource; applies to OpenIDC authentication mode only. |
| provisionerClientSecret | The client secret; applies to OpenIDC authentication mode only. |
| provisionerClientScope | The client scope; applies to OpenIDC authentication mode only and required for Active Directory Federation Services (ADFS) and Azure. |
| ksClientID | Client ID of a client, which has a service account role of ADMIN at the KumoScale resource; not for a private cluster. Applies to OpenIDC authentication mode only. |

ksClientSecret	The client secret; not for a private cluster. Applies to OpenIDC authentication mode only.
ksClientScope	The client scope; required for Active Directory Federation Services (ADFS) and Azure. Applies to OpenIDC authentication mode only.

Create the secret with the following command:

```
kubect1 create -f provisioner-secret.yaml
```

6. If topology labels have not been specified for your storage nodes, add them to your storage node CRD and apply the changes now. If not, you will need to redeploy the CSI driver after labeling. Set the storage node's Region, Rack, and Zone with corresponding values:

```
topology.kubernetes.io/region

topology.kubernetes.io/zone

topology.kubernetes.io/rack
```

Step 2. Update the CSI Driver Installation File

1. Create the CSI driver installation file **ks-csi.yaml**. See [Prepare for Installation](#) for the example file to copy to **ks-csi.yaml**.
2. Update **ks-csi.yaml** with the parameters of the KumoScale CSI controller plug-in, **ks-csi-plugin**. The plug-in enables storage provisioning from Kubernetes on top of KumoScale and the Provisioner. Update with the:

- IP address and port of the container repository or the location of the binary container (in this example – xx.xx.xx:8082). You will need to replace the value of \$REPO in multiple places
- KumoScale CSI driver version (in this example – 1.4.244).

```
- name: ks-csi-plugin

  image: 10.xx.xx.xx:8082/ks-csi-plugin:v1.4.244
```

- If you are using KumoScale for Managed Mode with a private local registry, you will also need to add a secret storing the authentication credentials in the **kube-system** namespace. If you don't have such a secret, create one. For example, the command below creates a secret called regcred to log into a Docker registry.

```
kubect1 -n kube-system create secret docker-registry regcred --docker-server=<registry-server> --docker-username=<uname> --docker-password=<pword> --docker-email=<email>
```

Add the secret to ks-csi.yaml in line with containers for csi-kumoscale-controller and csi-kumoscale-node:

```
<...>
  imagePullSecrets:
    - name: regcred

  volumes:
    - name: socket-dir
      emptyDir: {}
<...>
```

Step 3. Install the KumoScale CSI Driver

1. Enter the following command:

```
kubect1 create -f ks-csi.yaml
```

The driver will start running if it was successfully installed.

2. Verify that all the driver pods are up, by running:

```
kubect1 -n kube-system get pods
```

3. A node driver will be visible on each node, with the name:

```
kubect1 -n kube-system get pods | grep csi-kumoscale-node

csi-kumoscale-node-2zgxc   2/2      Running   0           77d
csi-kumoscale-node-j7gqm   2/2      Running   0           77d
```

One controller driver is visible, with the name:

```
kubect1 -n kube-system get pods | grep csi-kumoscale-controller

csi-kumoscale-controller-12345-67890   6/6      Running   0           77d
```

Step 4. Kubernetes 1.20 and higher, Install the Snapshot Plug-ins

For Kubernetes version 1.20 and higher, make sure snapshot-controller and snapshot-crd's are installed. These should be included in your Kubernetes distribution. If they are not, you can use the example yaml files included in Kubernetes_CSI to install them.

Step 4. Kubernetes 1.16-1.19, Prepare the Environment for Features in Beta

The Kubernetes orchestration ExpandVolume command is in beta for versions 1.16 through 1.19. . Kubernetes snapshots are generally available in version 1.20 and higher. The following sections discuss how to enable one or both features as required by the version of Kubernetes you are using.

The Kubernetes Environment

Update the status of the relevant feature by enabling the command through running kubelet and kube-apiserver with the additional parameters:

Expand Volume, for Kubernetes versions 1.16 and higher:

```
--feature-gates=ExpandCSIVolumes=true,ExpandInUsePersistentVolumes=true
```

Snapshots, for Kubernetes versions below 1.20 but higher than 1.17:

```
--feature-gates=VolumeSnapshotDataSource=true
```

For example, in CentOS operating system Kube nodes:

In all Kube nodes:

- Edit the following file:

```
/var/lib/kubelet/kubeadm-flags.env
```

- To configure Kublet, add the feature gates listed above to the variable:

```
KUBELET_KUBEADM_ARGS
```

In the master node:

- To configure kube-apiserver, add the feature gates listed above to the API server in the file:

```
/etc/kubernetes/manifests/kube-apiserver.yaml
```

Enable Volume Expansion in the Storage Class

To expand a volume, you will need to set allowVolumeExpansion:true in the **yaml** for your storage class.

See Next: [CSI Storage Provisioning and Volume Management](#)