



Troubleshooting

- [Troubleshooting](#), on page 1
- [HyperFlex CSI Node Plugin \(“csi-nodeplugin...”\) Pods Fail to Start](#), on page 1
- [Stateful Applications Stuck in Container Creating Stage](#), on page 2
- [ImagePullBackOff Status Errors when Deploying HX CSI Pods](#), on page 2
- [Enable Kubernetes Operation Fails on Clusters Deployed By Cisco Intersight](#), on page 3

Troubleshooting

The following section highlights common issues seen when installing and using the HyperFlex CSI integration. The information provided includes symptoms to help diagnose the issue as well as a solution to resolve the issue.

HyperFlex CSI Node Plugin (“csi-nodeplugin...”) Pods Fail to Start

- **Symptom 1:** HyperFlex CSI components have been installed and the “csi-attacher...” and “csi-provisioner...” pods are running however the “csi-nodeplugin...” pods on each node fail to start.
- **Symptom 2:** Running the command “`kubectl describe pod <csi-nodeplugin_pod_name>`” shows a message containing the following error: “MountVolume.SetUp failed for volume “iscsi-dir” : hostPath type check failed: /etc/iscsi is not a directory”

Solution:

Ensure that the “iscsi-initiator-utils” package has been installed on each of the Kubernetes worker nodes. The HyperFlex CSI integration uses the software iSCSI initiator within the guest operating system to connect to the persistent volume storage objects via iSCSI. The “iscsi-initiator-utils” package is required for this operation. Perform the following steps:

1. Remove the deployed HyperFlex CSI components using the “`kubectl delete -f ./hxcsi-deploy`” command.
2. Install the “iscsi-initiator-utils” package on each Kubernetes worker node. Depending on the guest operation system, the command will vary. As an example, on Ubuntu the command would be “`apt-get install iscsi-initiator-utils`”.

3. Re-apply the HyperFlex CSI deployment YAML files to the Kubernetes cluster using the “`kubectl create -f ./hxcsi-deploy`” command

Stateful Applications Stuck in Container Creating Stage

- **Symptom 1:** After deploying a statful Kubernetes workload using HX CSI, the pods as part of that workload remain in the ContainerCreating stage indefinitely.
- **Symptom 2:** Running the command “`kubectl describe pod <pod_name>`” shows a message container the following error: “`rpc error: code = Unkown desc = unable to find matching device for volume id`”
- **Symptom 3:** Your Kubernetes nodes (VMs) are running RHEL7 or CentOS7 or later guest operating system.

Solution:

In versions of RHEL7 and CentOS7 (or later), changes to SELINUX cause the “`iscsi_tcp`” kernel module to be loaded when called, rather than at boot. This causes issues when using the HyperFlex CSI integration. Ensure the “`iscsi_tcp`” kernel module is loaded at boot.

1. On each Kubernetes worker node, run the following command “`echo iscsi_tcp >> /etc/modules-load.d/iscsi.conf`”.

ImagePullBackOff Status Errors when Deploying HX CSI Pods

- **Symptom 1:** Running the command “`kubectl get pods [-n <namespace>]`” shows that the HX CSI pods are showing a status of “ImagePullBackOff”.
- **Symptom 2:** Running the command “`kubectl describe pod <csi-pod_name>`” shows a message containing the following error: “`Error: ErrImaePull`” and “`Back-off pulling image...`”

Solution:

- **Solution 1:** Ensure the HX CSI container image name provided to the `hxcsi-setup` script is correct
- **Solution 2:** Ensure the HX CSI container image exists, either directly within docker on each Kubernetes worker node or on the local container image registry depending on which deployment option was chosen.
- **Solution 3:** Ensure the “`imagePullPolicy`” lines in the following YAML files generated by the `hxcsi-setup` script are set to “`IfNotPresent`”.
 - `csi-attacher-hxcsi.yaml`
 - `csi-nodeplugin-hxcsi.yaml`
 - `csi-provisioner-hxcsi.yaml`

Enable Kubernetes Operation Fails on Clusters Deployed By Cisco Intersight

- **Symptom:** The enable Kubernetes operation hangs on the “Volume Access” stage or does not complete when run on HyperFlex clusters that were initially deployed by Cisco Intersight.

Solution:

There are two solutions for this symptom; determined by whether you have run **Enable Kubernetes** or not.

The solution for users who have already run **Enable Kubernetes**:

- Solution 1 = Required
- Solution 2 = Required

The solution for users who have not run **Enable Kubernetes**:

- Solution 1 = Optional
- Solution 2 = Required

Solution 1:

Run the following command on all ESX hosts:

```
esxcfg-vmknic -d -p k8-priv-iscsi
esxcli network vswitch standard portgroup remove -p k8-priv-iscsi -v k8-iscsi
esxcli network vswitch standard portgroup remove -p k8-priv-iscsivm-network -v k8-iscsi
esxcli network vswitch standard remove -v k8-iscsi
```

Solution 2:

This solution is required for all users. Run the following commands on all HX Controller VMs:

```
sed -i -e "s/255.255.0.0/255.255.255.0/g"
/opt/springpath/storfs-mgmt/stMgr-1.0/conf/application.conf

sed -i -e "s/255.255.0.0/255.255.255.0/g" \
-e "s/169.254.1./169.254.254./g" \
-e "s/except_vnic:$/except_vnic.device:/g"
/usr/share/springpath/storfs-misc/hx-scripts/iscsiVolumeAccessCheck.py

sed -i -e "s/255.255.0.0/255.255.255.0/g" \
-e "s/169.254.1./169.254.254./g" \
-e "s/except_vnic:$/except_vnic.device:/g"
/usr/share/springpath/storfs-misc/hx-scripts/iscsiVolumeAccessEnable.py

restart hxSvcMgr
restart stMgr
```

