



Troubleshooting

- [Troubleshooting, on page 1](#)
- [ImagePullBackOff Status Errors when Deploying HXCSI Pods, on page 1](#)
- [Volume Delete Fails, on page 2](#)
- [Application Pod in ContainerCreating State During Node Deletion, on page 2](#)
- [Deleted Pod Scheduled Back on the Same Node, on page 2](#)

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.

ImagePullBackOff Status Errors when Deploying HXCSI Pods

- **Symptom 1:** Running the command “`kubectl get pods [-n <namespace>]`” shows that the HXCSI 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: ErrImagePull” and “Back-off pulling image...”

Solution:

- **Solution 1:** Ensure the HXCSI container image name provided to the `hxcsi-setup` script is correct
- **Solution 2:** Ensure the HXCSI container image exists, either directly within docker on each Kubernetes worker node or on the local container image registry.
- **Solution 3:** Ensure that 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`
- **Solution 4:** Ensure that the following images are loaded on the local container image registry on each Kubernetes node:
`csi-attacher-3.0.2-cisco1.tar`, `csi-node-driver-registrar-2.0.1-cisco1.tar`,
`csi-resizer-1.0.1-cisco1.tar`, `csi-provisioner-2.0.4-cisco1.tar`

Volume Delete Fails

Volume delete fails due to stale volume attachments present even after NodeUnpublish succeeds and the volume is unmounted. This happens when the delete volumeattachment kubernetes api is lost during etcd leader election. Even after nodeUnpublish completes and the volume is successfully unmounted from the node, delete volume fails.

The log in the external-provisioner appears as:

```
volume deletion failed: persistentvolume <pv-name> is still attached to node <node-name>.
```

The log in the external-attacher appears as:

```
<Volume-attachment> is already attached.
```

Solution:

Delete the stale volumeattachments using the following command:

```
kubectl delete volumeattachments <VA-name>
```

After few seconds when the provisioner retries, pv is deleted.

Alternatively, you can delete manually using the following command:

```
kubectl delete pv <pv-name>
```

Application Pod in ContainerCreating State During Node Deletion

Application pods that get stuck in a ContainerCreating state or multi attach error state and are not able to mount the volumes. This may happen during the remove or delete a k8s worker node from the cluster and pods migrate to a new worker node.

The recommended way to delete a K8s worker node is by using the following commands:

```
kubectl drain <node-name>  
kubectl delete node <node-name>
```

For more information, see [How to gracefully remove a node from Kubernetes?](#).

Deleted Pod Scheduled Back on the Same Node

A pod in **Running** state may get recreated after deleting using the `kubectl delete pod` command and then get stuck in **Terminating** state on deleting its namespace.

Instead of using the `kubectl delete pod` command on any running pod, the following best practices way is recommended:

1. Note the node name on which pod to be deleted is running on.

```
kubectl get pods -o wide --all-namespaces
```

2. Cordon off the node on which the pod is running.

```
kubectl cordon <node-name>
```

3. Delete the pod.

```
kubectl delete pod <pod-name>
```

4. Check that the deleted pod is scheduled on a different node using the following command:

```
kubectl get pods -o wide --all-namespaces
```

5. Uncordon the cordoned node.

```
kubectl uncordon <node-name>
```

