CSI Driver for Dell EMC Unity

Version 1.1.0.1

Release Notes

REV 01
May 2020

These release notes contain supplemental information about the CSI Driver for Dell EMC Unity. Topics include:

DELL EMC

Revision history

The following table presents the revision history of this document.

Table 1 Revision history

Revision	Date	Description
01	May 2020	Updated the following sections for 1.1.0.1 release: • New features and changes • Resolved issues

Product description

This section describes the CSI Driver for Dell EMC Unity.

The CSI Driver for Dell EMC Unity is a plug-in that is installed into Kubernetes to provide persistent storage using Dell EMC Unity storage systems.

The CSI Driver adheres to the Container Storage Interface (CSI) specification v1.1. It is compatible with Kubernetes version 1.14. The Red Hat Enterprise Linux 7.6 or CentOS 7.6 host operating systems are supported.

New features and changes

The CSI Driver for Dell EMC Unity supports the following features:

There are no new features introduced as part of 1.1.0.1 release.

Resolved issues

List of issues that were fixed in this release.

The following issue is resolved as part of 1.1.0.1 release:

- Data Loss is happening when user bounces unity node pod.

Note: To avoid data loss, all users must upgrade to version 1.1.0.1 irrespective of current version (1.0 or 1.1) in use.

The procedure to upgrade is as follows:

- Environment type 1 - When user uses either Helm 2 or Helm 3:
  1. Clone csi-unity v1.1.0.1 by executing the following command:

     ```
     git clone -b v1.1.0.1 https://github.com/dell/csi-unity.git
     ```

  2. Execute upgrade.unity script to upgrade the driver to v1.1.0.1
- Environment type 2 - When user wants to upgrade from Helm 2 to Helm 3:

1. Clone csi-unity v1.1.0.1 by executing the following command:

```
git clone -b v1.1.0.1 https://github.com/dell/csi-unity.git
```

2. User must uninstall the driver using csi-unity v1.1.0.1 (Execute uninstall.unity).
3. Update helm version to Helm 3.
4. Install the driver using csi-unity v1.1.0.1 (Execute install.unity).

Known problems and limitations

This section describes the known problems and limitations associated with this release of the product.

Known problems

Issue	Resolution or workaround, if known
526: Driver is crashing while doing `install.unity` with wrong `unityUsername`, wrong `unityPassword`, or both.	Uninstall the CSI driver installation and re-install using the proper username and password.
603: Volume and snapshot name cannot have more than 63 characters.	All volume names and snapshot names (including prefix) must be less than 63 characters.
677: After successful deletion of a pod, the Node log would contain the following error, even though unmount and pod deletion happens successfully. `msg="NodeUnstageVolume error unmount stageTgt: unmount failed: exit status 32`	This is a bug in logging and can be ignored.
670: While creating a new pod, if the multipath driver on Kubernetes nodes is not able to issue the `dm-uuid-mpath` device for a given PVC, the driver picks the `wwn-0x` standard device to perform the mount operation. This may lead to data unavailability.	Restart the multipath driver, and ensure all devices are claimed with `dm-uuid-mpath`.
632: Pod stuck at container creating state for over 1 minute. Pod description shows `Unable to find device after multiple discovery attempts: rpc error: code = NotFound desc = Check for disk path /dev/disk/by-id/ wwn-0xxxxxxxxxxxxxxxxxxxxx not found readlink /dev/disk/by-id/ wwn-0xxxxxxxxxxxxxxxxxxxxx: no such file or directory`.	1. Check if the zoning is done correctly for the FC channel (if an FC-based volume is used), or that at least 1 iSCSI target is up on the array (if an iSCSI-based volume is used). 2. If the problem persists, execute `/usr/bin/ rescan-scsi-bus.sh -a -r`. The script cleans up any stale devices present on the node. Install the `sg3_utils` package if the script is not present on the node.
659: `unity` namespace got deleted.	1. Re-create the `unity` namespace. 2. Uninstall the controller and node pods, and then re-install the pods. You should be able to continue the creation and deletion of Kubernetes objects as usual.

Issue	Resolution or workaround, if known
566: When creating a PVC from a snapshot, the PVC is automatically given the attributes of the default storage class.	Create a new custom storage class. The new custom storage class should have the following properties with the same value as in the default storage class: • storage pool • thin provision • data reduction • size You must provide this storage class when creating a PVC from an existing snapshot.
326: When the same machine has two hosts with different hostnames but with the same IQN, the CSI Driver for Unity (v1.1) gives the following error: `REP 0125: unable to find host`.	Ensure that each host has a unique initiator IQN, and that the correct IQN was added to the corresponding host object on the array.

532: Not able to find storage class with `xfs` filetype.	Create a custom storage class for this type of requirement. An example follows of a custom storage class (in `yaml` format) that can be created in order to mount pvcs as an xfs filesystem.
	`apiVersion: storage.k8s.io/v1`
	`kind: StorageClass`
	`metadata:`
	`  name: sc-xfs-iscsi`
	`parameters:`
	`  FsType: xfs`
	`  isDataReductionEnabled: "false"`
	`  storagepool: pool_1`
	`  thinProvisioned: "true"`
	`  tieringPolicy: "3"`
	`  protocol: "iSCSI"`
	`provisioner: csi-unity.dellemc.com`
	`reclaimPolicy: Delete.`

After installing CSI Unity, the username/password of Unisphere changed.	1. Uninstall the current CSI Driver.
	2. Delete the existing secret `unity-creds` by issuing the following command:
	`kubectl delete secret unity-creds`
	3. Get the username and password encrypted with the following commands:
	`echo -n 'new username' | base64`
	`echo -n 'new password' | base64`
	4. Create a new secret (yaml file) in the below said format and specify user name and password:
	`apiVersion: v1`
	`data:`
	`  password: <encrypted new password string>`
	`  username: <encrypted new user string>`
	`kind: Secret`
	`metadata:`
	`  name: unity-creds`

Issue	Resolution or workaround, if known

	<pre> namespace: unity
	type: Opaque</pre>
	5. Create a new secret by issuing the command:
	`kubectl create -f <secret-file-name>`
	6. Ensure whether secret has been created by issuing the command:
	`kubectl get secrets -n unity`
	7. Re-install the CSI Driver.
	Note: Based on the SSL setup (for example, if your `unityInsecure` key has value of '`false`'), then you may have re-create the `unity-certs` secret. Refer to the "Certificate Validation for Unisphere REST API Calls" section of the *CSI Driver for Dell EMC Unity Product Guide*.

Software media, organization, and files

This section provides information about where you can find the software files for this release of the CSI Driver for Dell EMC Unity.

The software package is available for download from the CSI Driver for Dell EMC Unity GitHub page.

Additional resources

This section provides information about the CSI Driver for Dell EMC Unity, how to get support, and provide feedback.

Documentation

This section lists the related documentation for CSI Driver for Dell EMC Unity.

The CSI Driver for Dell EMC Unity is available on the CSI Driver for Dell EMC Unity GitHub page. The documentation includes the following:

- CSI Driver for Dell EMC Unity Release Notes (this document)
- CSI Driver for Dell EMC Unity Product Guide

Troubleshooting and getting help

Use the resources in this section to get help and support.

Product information

For documentation, release notes, software updates, and other information about Dell EMC products, go to Dell EMC Online Support.

Technical support

For any CSI driver configuration, setup issues, or questions, use the Dell EMC container forum. For any issues with Dell EMC Storage, contact Dell EMC Online Support.