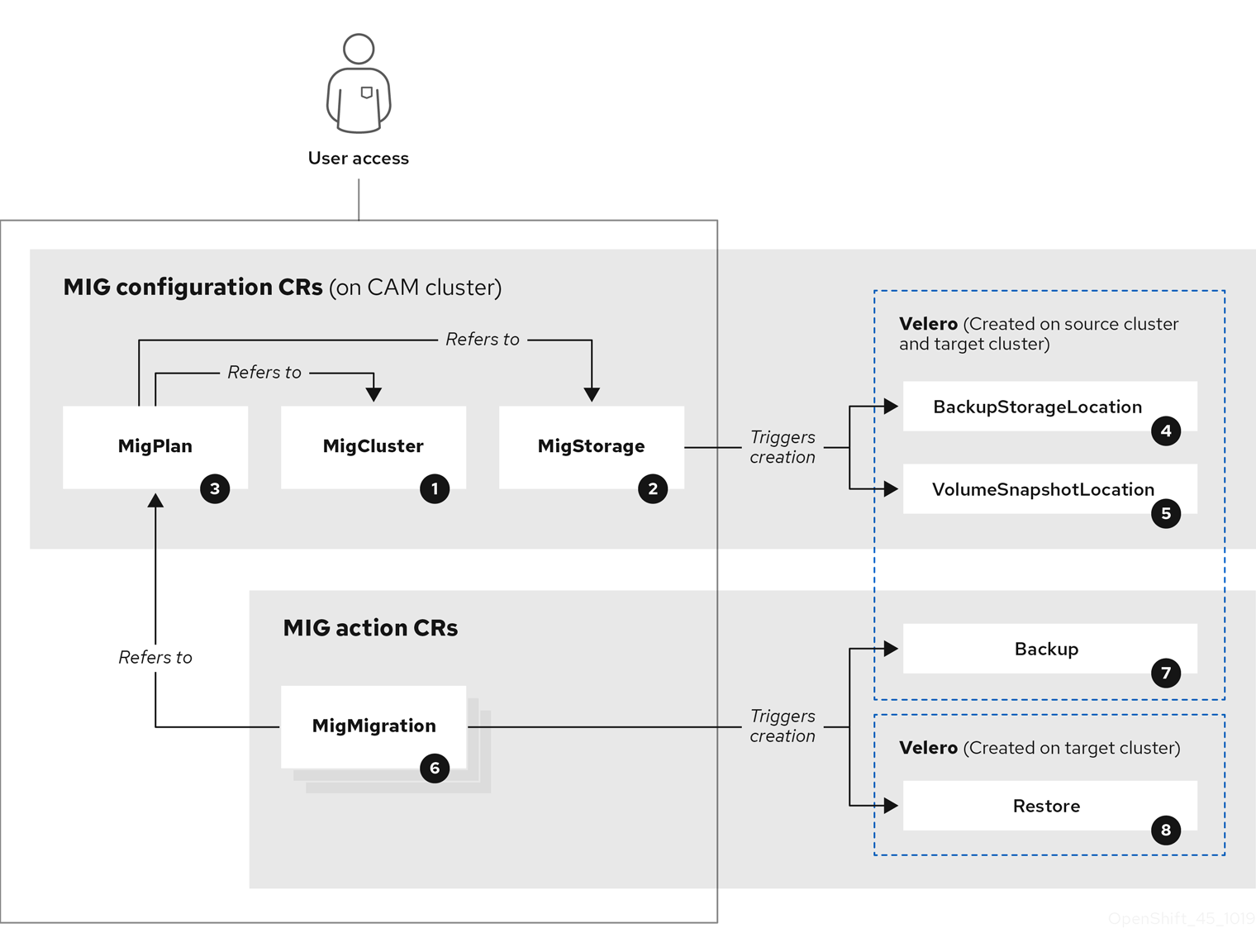
**Troubleshooting Openshift Migration**

CAM tool creates following Custom Resources (CRs) for migration :



20 [MigCluster](https://github.com/fusor/mig-controller/blob/master/pkg/apis/migration/v1alpha1/migcluster_types.go) (configuration, CAM cluster): Cluster definition

20 [MigStorage](https://github.com/fusor/mig-controller/blob/master/pkg/apis/migration/v1alpha1/migstorage_types.go) (configuration, CAM cluster): Storage definition

20 [MigPlan](https://github.com/fusor/mig-controller/blob/master/pkg/apis/migration/v1alpha1/migplan_types.go) (configuration, CAM cluster): Migration plan

The MigPlan CR describes the source and target clusters, repository, and namespace(s) being migrated. It is associated with 0, 1, or many MigMigration CRs.

|  |  |
| --- | --- |
|  | Deleting a MigPlan CR deletes the associated MigMigration CRs. |

20 [BackupStorageLocation](https://github.com/heptio/velero/blob/master/pkg/apis/velero/v1/backup_storage_location.go) (configuration, CAM cluster): Location of Velero backup objects

20 [VolumeSnapshotLocation](https://github.com/heptio/velero/blob/master/pkg/apis/velero/v1/volume_snapshot_location.go) (configuration, CAM cluster): Location of Velero volume snapshots

20 [MigMigration](https://github.com/fusor/mig-controller/blob/master/pkg/apis/migration/v1alpha1/migmigration_types.go) (action, CAM cluster): Migration, created during migration

A MigMigration CR is created every time you stage or migrate data. Each MigMigration CR is associated with a MigPlan CR.

20 [Backup](https://github.com/heptio/velero/blob/master/pkg/apis/velero/v1/backup.go) (action, source cluster): When you run a migration plan, the MigMigration CR creates two Velero backup CRs on each source cluster:

* Backup CR #1 for Kubernetes objects
* Backup CR #2 for PV data

20 [Restore](https://github.com/heptio/velero/blob/master/pkg/apis/velero/v1/restore.go) (action, target cluster): When you run a migration plan, the MigMigration CR creates two Velero restore CRs on the target cluster:

* Restore CR #1 (using Backup CR #2) for PV data
* Restore CR #2 (using Backup CR #1) for Kubernetes objects

Obtain the CR name:  
 $ oc get <cr> -n openshift-migration

NAME AGE

1. 88435fe0-c9f8-11e9-85e6-5d593ce65e10 6m42s

View the CR:  
 $ oc describe <cr> 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration

#### **Downloading migration logs**

You can download the Velero, Restic, and Migration controller logs in the CAM web console to troubleshoot a failed migration.

Procedure

1. Log in to the CAM console.
2. Click **Plans** to view the list of migration plans.
3. Click the **Options** menu kebab of a specific migration plan and select **Logs**.
4. Click **Download Logs** to download the logs of the Migration controller, Velero, and Restic for all clusters.
5. To download a specific log:
   1. Specify the log options:
      * **Cluster**: Select the source, target, or CAM host cluster.
      * **Log source**: Select **Velero**, **Restic**, or **Controller**.
      * **Pod source**: Select the Pod name, for example, **controller-manager-78c469849c-v6wcf**The selected log is displayed.  
        You can clear the log selection settings by changing your selection.
   2. Click **Download Selected** to download the selected log.

Optionally, access the logs by using the CLI, as in the following example:

$ oc get pods -n openshift-migration | grep controller

controller-manager-78c469849c-v6wcf 1/1 Running 0 4h49m

$ oc logs controller-manager-78c469849c-v6wcf -f -n openshift-migration

#### **Restic timeout error**

The following error appears in the Velero log, if migration fails due to restic timeout:

level=error msg="Error backing up item" backup=velero/monitoring error="timed out waiting for all PodVolumeBackups to complete" error.file="/go/src/github.com/heptio/velero/pkg/restic/backupper.go:165" error.function="github.com/heptio/velero/pkg/restic.(\*backupper).BackupPodVolumes" group=v1

The default value of **restic\_timeout** is one hour.

*Steps to increase the timeout value:*

1. In the OpenShift Container Platform web console, navigate to **Operators** → **Installed Operators**.
2. Click **Cluster Application Migration Operator**.
3. In the **MigrationController** tab, click **migration-controller**.
4. In the **YAML** tab, update the following parameter value:

**Spec:**

**restic\_timeout: 1h**

|  |  |
| --- | --- |
|  | Valid units are **h** (hours), **m** (minutes), and **s** (seconds), for example, **3h30m15s**. |

1. Click **Save**.

#### **Manually rolling back a migration**

If application was stopped during a failed migration, roll it back manually in order to prevent data corruption in the Persistent Volume.

1. On target cluster, switch to migrated project:  
    $ oc project <project>
2. Get the deployed resources:  
    $ oc get all
3. Delete deployed resources to ensure that application is not running on target cluster and accessing data on PVC:  
    $ oc delete <resource\_type>
4. To stop a DaemonSet without deleting it, update the **nodeSelector** in the YAML file:

**apiVersion: extensions/v1beta1**

**kind: DaemonSet**

**metadata:**

**name: hello-daemonset**

**spec:**

**selector:**

**matchLabels:**

**name: hello-daemonset**

**template:**

**metadata:**

**labels:**

**name: hello-daemonset**

**spec:**

**nodeSelector:**

**role: worker**

|  |  |
| --- | --- |
|  | Specify a **nodeSelector** value that does not exist on any node. |

1. Update each PV’s reclaim policy so that unnecessary data is removed. During migration, the reclaim policy for bound PVs is **Retain**, to ensure that data is not lost when an application is removed from source cluster. You can remove these PVs during rollback.

**apiVersion: v1**

**kind: PersistentVolume**

**metadata:**

**name: pv0001**

**spec:**

**capacity:**

**storage: 5Gi**

**accessModes:**

**- ReadWriteOnce**

**persistentVolumeReclaimPolicy: Retain**

**…**

**Status:**

|  |  |
| --- | --- |
|  | Specify **Recycle** or **Delete**. |

6. On source cluster, switch to the migrated project and get its deployed resources:  
 $ oc project <project>

$ oc get all

7. Start one or more replicas of each deployed resource:  
 $ oc scale --replicas=1 <resource\_type>/<resource\_name>

Update the **nodeSelector** of a DaemonSet to its original value, if you changed it during the procedure.

#### **Reference**

1. <https://docs.openshift.com/container-platform/4.3/migration/migrating_4_1_4/troubleshooting.html>