OpenShift Container Platform 4.3

Migration

Migrating from OpenShift Container Platform 3 to 4
OpenShift Container Platform 4.3 Migration

Migrating from OpenShift Container Platform 3 to 4
Abstract

This document provides instructions for migrating your OpenShift Container Platform cluster from version 3 to version 4.
# Table of Contents

CHAPTER 1. MIGRATING OPENSSHIFT CONTAINER PLATFORM 3 TO 4 ........................................ 5
  1.1. ABOUT MIGRATING OPENSHEET CONTAINER PLATFORM 3 TO 4 ......................... 5
  1.2. PLANNING YOUR MIGRATION .............................................................................. 5
  1.2.1. Comparing OpenShift Container Platform 3 and OpenShift Container Platform 4 .................................................. 6
  1.2.1.1. Architecture differences .............................................................................. 6
  1.2.1.2. Installation and update differences ................................................................. 6
  1.2.2. Migration considerations .................................................................................. 7
    1.2.2.1. Storage considerations .............................................................................. 7
      Local volume persistent storage .......................................................................... 7
      FlexVolume persistent storage .......................................................................... 7
      Container Storage Interface (CSI) persistent storage ........................................... 7
      Red Hat OpenShift Container Storage .................................................................. 7
      Unsupported persistent storage options ............................................................... 8
    1.2.2.2. Networking considerations ........................................................................ 8
      Network isolation mode ....................................................................................... 8
      Encrypting traffic between hosts ........................................................................ 8
    1.2.2.3. Logging considerations ............................................................................... 8
      Deploying cluster logging .................................................................................. 8
      Aggregated logging data .................................................................................... 8
    1.2.2.4. Security considerations ............................................................................. 9
      Unauthenticated access to discovery endpoints .................................................... 9
      Identity providers ............................................................................................... 9
    1.2.2.5. Monitoring considerations ........................................................................ 9
      Alert for monitoring infrastructure availability .................................................. 9
  1.3. MIGRATING APPLICATION WORKLOADS FROM OPENSHEET CONTAINER PLATFORM 3.7 TO 4.3 .................................................. 9
    1.3.1. Migration prerequisites .................................................................................. 10
    1.3.2. Understanding the Cluster Application Migration tool ................................... 10
  1.4. CONFIGURING A REPLICATION REPOSITORY ...................................................... 12
    1.4.1. Understanding the data copy methods for migration ...................................... 12
      1.4.1.1. File system copy method ......................................................................... 13
      1.4.1.2. Snapshot copy method ........................................................................... 13
    1.4.2. Configuring a Multi-Cloud Object Gateway storage bucket as a replication repository .................................................. 13
      1.4.2.1. Installing the OpenShift Container Storage Operator .................................. 13
      1.4.2.2. Creating the Multi-Cloud Object Gateway storage bucket ....................... 14
    1.4.3. Configuring an AWS S3 storage bucket as a replication repository ............... 16
    1.4.4. Configuring a Google Cloud Provider storage bucket as a replication repository .................................................. 19
    1.4.5. Configuring a Microsoft Azure Blob storage container as a replication repository .................................................. 20
  1.5. DEPLOYING THE CLUSTER APPLICATION MIGRATION (CAM) TOOL ............. 22
    1.5.1. Installing the Cluster Application Migration Operator on an OpenShift Container Platform 3 source cluster ............... 22
    1.5.2. Installing the Cluster Application Migration Operator on an OpenShift Container Platform 4.3 target cluster ............... 23
  1.6. MIGRATING APPLICATIONS WITH THE CAM WEB CONSOLE ..................... 24
    1.6.1. Launching the CAM web console .................................................................. 24
    1.6.2. Adding a cluster to the CAM web console .................................................... 25
    1.6.3. Adding a replication repository to the CAM web console ................................ 26
1.6.4. Changing migration plan limits for large migrations
1.6.5. Creating a migration plan in the CAM web console
1.6.6. Running a migration plan in the CAM web console

1.7. MIGRATING CONTROL PLANE SETTINGS WITH THE CONTROL PLANE MIGRATION ASSISTANT (CPMA)

1.7.1. Understanding the Control Plane Migration Assistant
1.7.2. Installing the Control Plane Migration Assistant
1.7.3. Using the Control Plane Migration Assistant

1.8. TROUBLESHOOTING

1.8.1. Viewing migration Custom Resources (CRs)
1.8.2. Downloading migration logs
1.8.3. Restic timeout error
1.8.4. Manually rolling back a migration
1.8.5. Known issues

CHAPTER 2. MIGRATING OPENSHIFT CONTAINER PLATFORM 4.1 TO 4.3

2.1. MIGRATING APPLICATION WORKLOADS FROM OPENSHIFT CONTAINER PLATFORM 4.1 TO 4.3

2.2. CONFIGURING A REPLICATION REPOSITORY

2.2.1. Understanding the data copy methods for migration
   2.2.1.1. File system copy method
   2.2.1.2. Snapshot copy method

2.2.2. Configuring a Multi-Cloud Object Gateway storage bucket as a replication repository
   2.2.2.1. Installing the OpenShift Container Storage Operator
   2.2.2.2. Creating the Multi-Cloud Object Gateway storage bucket

2.2.3. Configuring an AWS S3 storage bucket as a replication repository
2.2.4. Configuring a Google Cloud Provider storage bucket as a replication repository
2.2.5. Configuring a Microsoft Azure Blob storage container as a replication repository

2.3. DEPLOYING THE CLUSTER APPLICATION MIGRATION (CAM) TOOL

2.3.1. Installing the Cluster Application Migration Operator on an OpenShift Container Platform 4.1 source cluster
2.3.2. Installing the Cluster Application Migration Operator on an OpenShift Container Platform 4.3 target cluster

2.4. MIGRATING APPLICATIONS WITH THE CAM WEB CONSOLE

2.4.1. Launching the CAM web console
2.4.2. Adding a cluster to the CAM web console
2.4.3. Adding a replication repository to the CAM web console
2.4.4. Changing migration plan limits for large migrations
2.4.5. Creating a migration plan in the CAM web console
2.4.6. Running a migration plan in the CAM web console

2.5. TROUBLESHOOTING

2.5.1. Viewing migration Custom Resources (CRs)
2.5.2. Downloading migration logs
2.5.3. Restic timeout error
2.5.4. Manually rolling back a migration
2.5.5. Known issues

CHAPTER 3. MIGRATING OPENSHIFT CONTAINER PLATFORM 4.2 TO 4.3

3.1. MIGRATING APPLICATION WORKLOADS FROM OPENSHIFT CONTAINER PLATFORM 4.2 TO 4.3

3.2. CONFIGURING A REPLICATION REPOSITORY

3.2.1. Understanding the data copy methods for migration
3.2.1.1. File system copy method
3.2.1.2. Snapshot copy method
3.2.2. Configuring a Multi-Cloud Object Gateway storage bucket as a replication repository
   3.2.2.1. Installing the OpenShift Container Storage Operator
   3.2.2.2. Creating the Multi-Cloud Object Gateway storage bucket
3.2.3. Configuring an AWS S3 storage bucket as a replication repository
3.2.4. Configuring a Google Cloud Provider storage bucket as a replication repository
3.2.5. Configuring a Microsoft Azure Blob storage container as a replication repository
3.3. DEPLOYING THE CLUSTER APPLICATION MIGRATION (CAM) TOOL
   3.3.1. Installing the Cluster Application Migration Operator on an OpenShift Container Platform 4.2 source cluster
   3.3.2. Installing the Cluster Application Migration Operator on an OpenShift Container Platform 4.3 target cluster
3.4. MIGRATING APPLICATIONS WITH THE CAM WEB CONSOLE
   3.4.1. Launching the CAM web console
   3.4.2. Adding a cluster to the CAM web console
   3.4.3. Adding a replication repository to the CAM web console
   3.4.4. Changing migration plan limits for large migrations
   3.4.5. Creating a migration plan in the CAM web console
   3.4.6. Running a migration plan in the CAM web console
3.5. TROUBLESHOOTING
   3.5.1. Viewing migration Custom Resources (CRs)
   3.5.2. Downloading migration logs
   3.5.3. Restic timeout error
   3.5.4. Manually rolling back a migration
   3.5.5. Known issues
CHAPTER 1. MIGRATING OPENSHIFT CONTAINER PLATFORM 3 TO 4

1.1. ABOUT MIGRATING OPENSHIFT CONTAINER PLATFORM 3 TO 4

OpenShift Container Platform 4 includes new technologies and functionality that results in a cluster that is self-managing, flexible, and automated. The way that OpenShift Container Platform 4 clusters are deployed and managed drastically differs from OpenShift Container Platform 3.

To successfully transition from OpenShift Container Platform 3 to OpenShift Container Platform 4, it is important that you review the following information:

Planning your transition
Learn about the differences between OpenShift Container Platform versions 3 and 4. Prior to transitioning, be sure that you have reviewed and prepared for storage, networking, logging, security, and monitoring considerations.

Performing your migration
Learn about and use the tools to perform your migration:

- Cluster Application Migration (CAM) tool to migrate your application workloads
- Control Plane Migration Assistant (CPMA) to migrate your control plane

1.2. PLANNING YOUR MIGRATION

Before performing your migration to OpenShift Container Platform 4.3, it is important to take the time to properly plan for the transition. OpenShift Container Platform 4 introduces architectural changes and enhancements, so the procedures that you used to manage your OpenShift Container Platform 3 cluster might not apply for OpenShift Container Platform 4.

NOTE
This planning document assumes that you are transitioning from OpenShift Container Platform 3.11 to OpenShift Container Platform 4.3.

This document provides high-level information on the most important differences between OpenShift Container Platform 3 and OpenShift Container Platform 4 and the most noteworthy migration considerations. For detailed information on configuring your OpenShift Container Platform 4 cluster, review the appropriate sections of the OpenShift Container Platform documentation. For detailed information on new features and other notable technical changes, review the OpenShift Container Platform 4.3 release notes.

It is not possible to upgrade your existing OpenShift Container Platform 3 cluster to OpenShift Container Platform 4. You must start with a new OpenShift Container Platform 4 installation. Tools are available to assist in migrating your control plane settings and application workloads.

1.2.1. Comparing OpenShift Container Platform 3 and OpenShift Container Platform 4
With OpenShift Container Platform 3, administrators individually deployed Red Hat Enterprise Linux (RHEL) hosts, and then installed OpenShift Container Platform on top of these hosts to form a cluster. Administrators were responsible for properly configuring these hosts and performing updates.

OpenShift Container Platform 4 represents a significant change in the way that OpenShift Container Platform clusters are deployed and managed. OpenShift Container Platform 4 includes new technologies and functionality, such as Operators, MachineSets, and Red Hat Enterprise Linux CoreOS (RHCOS), which are core to the operation of the cluster. This technology shift enables clusters to self-manage some functions previously performed by administrators. This also ensures platform stability and consistency, and simplifies installation and scaling.

For more information, see OpenShift Container Platform architecture.

1.2.1.1. Architecture differences

Immutable infrastructure
OpenShift Container Platform 4 uses Red Hat Enterprise Linux CoreOS (RHCOS), which is designed to run containerized applications, and provides efficient installation, Operator-based management, and simplified upgrades. RHCOS is an immutable container host, rather than a customizable operating system like RHEL. RHCOS enables OpenShift Container Platform 4 to manage and automate the deployment of the underlying container host. RHCOS is a part of OpenShift Container Platform, which means that everything runs inside a container and is deployed using OpenShift Container Platform.

In OpenShift Container Platform 4, control plane nodes must run RHCOS, ensuring that full-stack automation is maintained for the control plane. This makes rolling out updates and upgrades a much easier process than in OpenShift Container Platform 3.

For more information, see Red Hat Enterprise Linux CoreOS.

Operators
Operators are a method of packaging, deploying, and managing a Kubernetes application. Operators ease the operational complexity of running another piece of software. They watch over your environment and use the current state to make decisions in real time. Advanced Operators are designed to upgrade and react to failures automatically.

For more information, see Understanding Operators.

1.2.1.2. Installation and update differences

Installation process
To install OpenShift Container Platform 3.11, you prepared your Red Hat Enterprise Linux (RHEL) hosts, set all of the configuration values your cluster needed, and then ran an Ansible playbook to install and set up your cluster.

In OpenShift Container Platform 4.3, you use the OpenShift installation program to create a minimum set of resources required for a cluster. Once the cluster is running, you use Operators to further configure your cluster and to install new services. After first boot, Red Hat Enterprise Linux CoreOS (RHCOS) systems are managed by the Machine Config Operator (MCO) that runs in the OpenShift Container Platform cluster.

For more information, see Installation process.

If you want to add RHEL worker machines to your OpenShift Container Platform 4.3 cluster, you use an Ansible playbook to join the RHEL worker machines after the cluster is running. For more information, see Adding RHEL compute machines to an OpenShift Container Platform cluster.
Infrastructure options
In OpenShift Container Platform 3.11, you installed your cluster on infrastructure that you prepared and maintained. In addition to providing your own infrastructure, OpenShift Container Platform 4 offers an option to deploy a cluster on infrastructure that the OpenShift Container Platform installation program provisions and the cluster maintains.

For more information, see OpenShift Container Platform installation overview.

Upgrading your cluster
In OpenShift Container Platform 3.11, you upgraded your cluster by running Ansible playbooks. In OpenShift Container Platform 4.3, the cluster manages its own updates, including updates to Red Hat Enterprise Linux CoreOS (RHCOS) on cluster nodes. You can easily upgrade your cluster by using the web console or by using the `oc adm upgrade` command from the OpenShift CLI and the Operators will automatically upgrade themselves. If your OpenShift Container Platform 4.3 cluster has Red Hat Enterprise Linux worker machines, then you will still need to run an Ansible playbook to upgrade those worker machines.

For more information, see Upgrading clusters.

1.2.2. Migration considerations

Review the changes and other considerations that might affect your transition from OpenShift Container Platform 3.11 to OpenShift Container Platform 4.

1.2.2.1. Storage considerations

Review the following storage changes to consider when transitioning from OpenShift Container Platform 3.11 to OpenShift Container Platform 4.3.

Local volume persistent storage
Local storage is only supported by using the Local Storage Operator in OpenShift Container Platform 4.3. It is not supported to use the local provisioner method from OpenShift Container Platform 3.11.

For more information, see Persistent storage using local volumes.

FlexVolume persistent storage
The FlexVolume plug-in location changed from OpenShift Container Platform 3.11. The new location in OpenShift Container Platform 4.3 is `/etc/kubernetes/kubelet-plugins/volume/exec`. Attachable FlexVolume plug-ins are no longer supported.

For more information, see Persistent storage using FlexVolume.

Container Storage Interface (CSI) persistent storage
Persistent storage using the Container Storage Interface (CSI) was Technology Preview in OpenShift Container Platform 3.11. CSI version 1.1.0 is fully supported in OpenShift Container Platform 4.3, but does not ship with any CSI drivers. You must install your own driver.

For more information, see Persistent storage using the Container Storage Interface (CSI).

Red Hat OpenShift Container Storage
Red Hat OpenShift Container Storage 3, which is available for use with OpenShift Container Platform 3.11, uses Red Hat Gluster Storage as the backing storage.

Red Hat OpenShift Container Storage 4, which is available for use with OpenShift Container Platform 4, uses Red Hat Ceph Storage as the backing storage.
Unsupported persistent storage options
Support for the following persistent storage options from OpenShift Container Platform 3.11 has changed in OpenShift Container Platform 4.3:

- GlusterFS is no longer supported.
- CephFS as a standalone product is no longer supported.
- Ceph RBD as a standalone product is no longer supported.

If you used one of these in OpenShift Container Platform 3.11, you must choose a different persistent storage option for full support in OpenShift Container Platform 4.3.

1.2.2.2. Networking considerations

Review the following networking changes to consider when transitioning from OpenShift Container Platform 3.11 to OpenShift Container Platform 4.3.

Network isolation mode
The default network isolation mode for OpenShift Container Platform 3.11 was `ovs-subnet`, though users frequently switched to use `ovn-multitenant`. The default network isolation mode for OpenShift Container Platform 4.3 is now `NetworkPolicy`.

If your OpenShift Container Platform 3.11 cluster used the `ovs-subnet` or `ovs-multitenant` mode, it is recommended to switch to the `NetworkPolicy` mode for your OpenShift Container Platform 4.3 cluster. `NetworkPolicy` is supported upstream, is more flexible, and also provides the functionality that `ovs-multitenant` does. If you want to maintain the `ovs-multitenant` behavior while using `NetworkPolicy` in OpenShift Container Platform 4.3, follow the steps to configure multitenant isolation using `NetworkPolicy`.

For more information, see About network policy.

Encrypting traffic between hosts
In OpenShift Container Platform 3.11, you could use IPsec to encrypt traffic between hosts. OpenShift Container Platform 4.3 does not support IPsec. It is recommended to use Red Hat OpenShift Service Mesh to enable mutual TLS between services.

For more information, see Understanding Red Hat OpenShift Service Mesh.

1.2.2.3. Logging considerations

Review the following logging changes to consider when transitioning from OpenShift Container Platform 3.11 to OpenShift Container Platform 4.3.

Deploying cluster logging
OpenShift Container Platform 4 provides a simple deployment mechanism for cluster logging, by using a Cluster Logging custom resource. Once deployed, the cluster logging experience is the same as it was in OpenShift Container Platform 3.11.

For more information, see About deploying and configuring cluster logging.

Aggregated logging data
You cannot transition your aggregate logging data from OpenShift Container Platform 3.11 into your new OpenShift Container Platform 4 cluster.

For more information, see About cluster logging.

1.2.2.4. Security considerations

Review the following security changes to consider when transitioning from OpenShift Container Platform 3.11 to OpenShift Container Platform 4.3.

**Unauthenticated access to discovery endpoints**

In OpenShift Container Platform 3.11, an unauthenticated user could access the discovery endpoints (for example, /api/* and /apis/*). For security reasons, unauthenticated access to the discovery endpoints is no longer allowed in OpenShift Container Platform 4.3. If you do need to allow unauthenticated access, you can configure the RBAC settings as necessary; however, be sure to consider the security implications as this can expose internal cluster components to the external network.

**Identity providers**

Configuration for identity providers has changed for OpenShift Container Platform 4, including the following notable changes:

- The request header identity provider in OpenShift Container Platform 4.3 requires mutual TLS, where in OpenShift Container Platform 3.11 it did not.

- The configuration of the OpenID Connect identity provider was simplified in OpenShift Container Platform 4.3. It now obtains data, which previously had to be specified in OpenShift Container Platform 3.11, from the provider’s .well-known/openid-configuration endpoint.

For more information, see Understanding identity provider configuration.

1.2.2.5. Monitoring considerations

Review the following monitoring changes to consider when transitioning from OpenShift Container Platform 3.11 to OpenShift Container Platform 4.3.

**Alert for monitoring infrastructure availability**

The default alert that triggers to ensure the availability of the monitoring structure was called DeadMansSwitch in OpenShift Container Platform 3.11. This was renamed to Watchdog in OpenShift Container Platform 4. If you had PagerDuty integration set up with this alert in OpenShift Container Platform 3.11, you must set up the PagerDuty integration for the Watchdog alert in OpenShift Container Platform 4.

For more information, see Applying custom Alertmanager configuration.

1.3. MIGRATING APPLICATION WORKLOADS FROM OPENSHIFT CONTAINER PLATFORM 3.7 TO 4.3

You can migrate application workloads from OpenShift Container Platform 3.7 (and later) to OpenShift Container Platform 4.3 with the Cluster Application Migration (CAM) tool. The CAM tool enables you to control the migration and to minimize application downtime.

The CAM tool’s web console and API, based on Kubernetes Custom Resources, enable you to migrate stateful application workloads at the granularity of a namespace.
Optionally, you can use the Control Plane Migration Assistant (CPMA) to assist you in migrating control plane settings.

**IMPORTANT**

Before you begin your migration, be sure to review the information on planning your migration.

1.3.1. Migration prerequisites

- The source cluster must be OpenShift Container Platform 3.7, 3.9, 3.10, or 3.11.
- You must have podman installed.
- You must have cluster-admin privileges on all clusters.
- You must configure a replication repository that is accessible to the source and target clusters.
- If your application uses images from the openshift namespace, the required versions of the images must be present on the target cluster.

If the required images are not present, you must update the imagestreamtags references to use an available version that is compatible with your application. If the imagestreamtags cannot be updated, you can manually upload equivalent images to the application namespaces and update the applications to reference them.

The following imagestreamtags have been removed from OpenShift Container Platform 4.3:

- dotnet:1.0, dotnet:1.1, dotnet:2.0
- dotnet-runtime:2.0
- mariadb:10.1
- mongodb:2.4, mongodb:2.6
- mysql:5.5, mysql:5.6
- nginx:1.8
- nodejs:0.10, nodejs:4, nodejs:6
- perl:5.16, perl:5.20
- php:5.5, php:5.6
- python:3.3, python:3.4
- ruby:2.0, ruby:2.2

1.3.2. Understanding the Cluster Application Migration tool

The Cluster Application Migration (CAM) tool enables you to migrate Kubernetes resources, persistent volume data, and internal container images from an OpenShift Container Platform source cluster to an OpenShift Container Platform 4.3 target cluster, using the CAM web console or the Kubernetes API.
Migrating an application with the CAM web console involves the following steps:

1. Install the Cluster Application Migration Operator on all clusters.

   **NOTE**

   The Cluster Application Migration Operator installs the CAM tool (CAM web console and Migration controller) on the target cluster by default. You can configure the Migration controller to install the CAM tool on another cluster. The cluster on which the CAM tool is installed is referred to as the CAM tool cluster.

2. Configure the replication repository, an intermediate object storage that the CAM tool uses to migrate data.

3. Add the source cluster to the CAM web console.

4. Add the replication repository to the CAM web console.

5. Create a migration plan, with one of the following data migration options:

   - **Copy**: The CAM tool copies the data from the source cluster to the replication repository, and from the replication repository to the target cluster.

   ![Diagram of Copy](image)

   - **Move**: The CAM tool unmounts a remote volume (for example, NFS) from the source cluster, creates a PV resource on the target cluster pointing to the remote volume, and then mounts the remote volume on the target cluster. Applications running on the target cluster use the same remote volume that the source cluster was using. The remote volume must be accessible to the source and target clusters.

   ![Diagram of Move](image)

   **NOTE**

   Although the replication repository does not appear in this diagram, it is required for the actual migration.
6. Run the migration plan, with one of the following options:

- **Stage** (optional) copies data to the target cluster without stopping the application. Staging can be run multiple times so that most of the data is copied to the target before migration. This minimizes the actual migration time and application downtime.

- **Migrate** stops the application on the source cluster and recreates its resources on the target cluster. Optionally, you can migrate the workload without stopping the application.

1.4. CONFIGURING A REPLICATION REPOSITORY

You must configure an object storage to use as a replication repository. The Cluster Application Migration tool copies data from the source cluster to the replication repository, and then from the replication repository to the target cluster, using either the file system or the snapshot data copy method.

The following storage providers are supported:

- Generic S3 object storage, for example, Minio or Ceph S3
- Multi-Cloud Object Gateway (MCG)
- Amazon Web Services (AWS) S3
- Google Cloud Provider (GCP)
- Microsoft Azure

1.4.1. Understanding the data copy methods for migration
The CAM tool supports the file system and snapshot data copy methods for migrating data from the source cluster to the target cluster. You can select a method that is suited for your environment and is supported by your storage provider.

### 1.4.1.1. File system copy method

The CAM tool copies data files from the source cluster to the replication repository, and from there to the target cluster.

**Table 1.1. File system copy method summary**

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Clusters can have different storage classes</td>
<td>● Slower than the snapshot copy method</td>
</tr>
<tr>
<td>● Supported for all S3 storage providers</td>
<td></td>
</tr>
</tbody>
</table>

### 1.4.1.2. Snapshot copy method

The CAM tool copies a snapshot of the source cluster’s data to a cloud provider’s object storage, configured as a replication repository. The data is restored on the target cluster.

AWS, Google Cloud Provider, and Microsoft Azure support the snapshot copy method.

**Table 1.2. Snapshot copy method summary**

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Faster than the file system copy method</td>
<td>● Cloud provider must support snapshots.</td>
</tr>
<tr>
<td></td>
<td>● Clusters must be on the same cloud provider.</td>
</tr>
<tr>
<td></td>
<td>● Clusters must be in the same location or region.</td>
</tr>
<tr>
<td></td>
<td>● Clusters must have the same storage class.</td>
</tr>
<tr>
<td></td>
<td>● Storage class must be compatible with snapshots.</td>
</tr>
</tbody>
</table>

### 1.4.2. Configuring a Multi-Cloud Object Gateway storage bucket as a replication repository

You can install the OpenShift Container Storage Operator and configure a Multi-Cloud Object Gateway (MCG) storage bucket as a replication repository on the CAM tool cluster.

#### 1.4.2.1. Installing the OpenShift Container Storage Operator

You can install the OpenShift Container Storage Operator from OperatorHub on the CAM tool cluster.
Procedure

1. In the OpenShift Container Platform web console, click Administration → Namespaces.

2. Click Create Namespace.

3. Enter openshift-storage in the Name field and click Create.

4. Click Operators → OperatorHub.

5. Use Filter by keyword (in this case, OCS) to find the OpenShift Container Storage Operator.

6. Select the OpenShift Container Storage Operator and click Install.

7. On the Create Operator Subscription page, select the openshift-storage namespace.

8. Specify your update channel and approval strategy.

9. Click Subscribe.
   On the Installed Operators page, the OpenShift Container Storage Operator appears in the openshift-storage project with the status Succeeded.

1.4.2.2. Creating the Multi-Cloud Object Gateway storage bucket

You can create the Multi-Cloud Object Gateway (MCG) storage bucket’s Custom Resources (CRs) on the CAM tool cluster.

Procedure

1. Log in to the CAM tool cluster.

2. Create the NooBaa CR configuration file, noobaa.yml, with the following content:

   ```yaml
   apiVersion: noobaa.io/v1alpha1
   kind: NooBaa
   metadata:
     name: noobaa
   namespace: openshift-storage
   spec:
     dbResources:
       requests:
         cpu: 0.5
         memory: 1Gi
     coreResources:
       requests:
         cpu: 0.5
         memory: 1Gi
   
   1 2 For a very small cluster, you can change the cpu value to 0.1.
   
3. Create the NooBaa object:

   ```bash
   $ oc create -f noobaa.yml
   
4. Create the BackingStore CR configuration file, bs.yml, with the following content:
1. Specify the number of volumes in the PV pool.
2. Specify the size of the volumes.
3. Specify the storage class.

5. Create the **BackingStore** object:

```
$ oc create -f bs.yml
```

6. Create the **BucketClass** CR configuration file, `bc.yml`, with the following content:

```
apiVersion: noobaa.io/v1alpha1
kind: BucketClass
metadata:
  labels:
    app: noobaa
    name: mcg-pv-pool-bc
    namespace: openshift-storage
spec:
  placementPolicy:
    tiers:
      - backingStores:
        - mcg-pv-pool-bs
        placement: Spread
```

7. Create the **BucketClass** object:

```
$ oc create -f bc.yml
```

8. Create the **ObjectBucketClaim** CR configuration file, `obc.yml`, with the following content:

```
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
```
1.4.3. Configuring an AWS S3 storage bucket as a replication repository

You can configure an AWS S3 storage bucket as a replication repository.

Prerequisites

- The AWS S3 storage bucket must be accessible to the source and target clusters.
- You must have the AWS CLI installed.
- If you are using the snapshot copy method:
  - You must have access to EC2 Elastic Block Storage (EBS).
  - The source and target clusters must be in the same region.
The source and target clusters must have the same storage class.

The storage class must be compatible with snapshots.

Procedure

1. Create an AWS S3 bucket:
   ```bash
   $ aws s3api create-bucket \
       --bucket <bucket_name> \
       --region <bucket_region>
   ```
   1. Specify your S3 bucket name.
   2. Specify your S3 bucket region, for example, `us-east-1`.

2. Create the IAM user `velero`:
   ```bash
   $ aws iam create-user --user-name velero
   ```

3. Create an EC2 EBS snapshot policy:
   ```bash
   $ cat > velero-ec2-snapshot-policy.json <<EOF
   {
   "Version": "2012-10-17",
   "Statement": [
   {
   "Effect": "Allow",
   "Action": [
   "ec2:DescribeVolumes",
   "ec2:DescribeSnapshots",
   "ec2:CreateTags",
   "ec2:CreateVolume",
   "ec2:CreateSnapshot",
   "ec2:DeleteSnapshot"
   ],
   "Resource": "*"
   }
   ]
   "Resource": "*"
   }
   EOF
   ```

4. Create an AWS S3 access policy for one or for all S3 buckets:
   ```bash
   $ cat > velero-s3-policy.json <<EOF
   {
   "Version": "2012-10-17",
   "Statement": [
   {
   "Effect": "Allow",
   "Action": [
   "s3:GetObject",
   "s3:DeleteObject",
   "s3:PutObject",
   
   ```
To grant access to a single S3 bucket, specify the bucket name. To grant access to all AWS S3 buckets, specify * instead of a bucket name:

```
"Resource": [
  "arn:aws:s3:::*"
]
```

5. Attach the EC2 EBS policy to *velero*:

```
$ aws iam put-user-policy \
   --user-name velero \
   --policy-name velero-ebs \
   --policy-document file://velero-ec2-snapshot-policy.json
```

6. Attach the AWS S3 policy to *velero*:

```
$ aws iam put-user-policy \
   --user-name velero \
   --policy-name velero-s3 \
   --policy-document file://velero-s3-policy.json
```

7. Create an access key for *velero*:

```
$ aws iam create-access-key --user-name velero
{
  "AccessKey": {
    "UserName": "velero",
    "Status": "Active",
    "CreateDate": "2017-07-31T22:24:41.576Z",
    "SecretAccessKey": "<AWS_SECRET_ACCESS_KEY>"
  }
}
```
"AccessKeyId": <AWS_ACCESS_KEY_ID>  
}

1. Record the AWS_SECRET_ACCESS_KEY and the AWS_ACCESS_KEY_ID for adding the AWS repository to the CAM web console.

1.4.4. Configuring a Google Cloud Provider storage bucket as a replication repository

You can configure a Google Cloud Provider (GCP) storage bucket as a replication repository.

Prerequisites

- The GCP storage bucket must be accessible to the source and target clusters.
- You must have gsutil installed.
- If you are using the snapshot copy method:
  - The source and target clusters must be in the same region.
  - The source and target clusters must have the same storage class.
  - The storage class must be compatible with snapshots.

Procedure

1. Run gsutil init to log in:

   $ gsutil init
   Welcome! This command will take you through the configuration of gcloud.

   Your current configuration has been set to: [default]

   To continue, you must login. Would you like to login (Y/n)?

2. Set the BUCKET variable:

   $ BUCKET=<bucket_name>  

   Specify your bucket name.

3. Create a storage bucket:

   $ gsutil mb gs://$BUCKET/

4. Set the PROJECT_ID variable to your active project:

   $ PROJECT_ID=$(gcloud config get-value project)

5. Create a velero service account:
6. Set the `SERVICE_ACCOUNT_EMAIL` variable to the service account’s email address:

```sh
$ SERVICE_ACCOUNT_EMAIL=$(gcloud iam service-accounts list
  --filter="displayName:Velero Storage"
  --format 'value(email)')
```

7. Grant permissions to the service account:

```sh
$ ROLE_PERMISSIONS=(
  compute.disks.get
  compute.disks.create
  compute.disks.createSnapshot
  compute.snapshots.get
  compute.snapshots.create
  compute.snapshots.useReadOnly
  compute.snapshots.delete
  compute.zones.get
)

gcloud iam roles create velero.server
  --project $PROJECT_ID
  --title "Velero Server"
  --permissions "$(IFS=","; echo "${ROLE_PERMISSIONS[*]}")"

gcloud projects add-iam-policy-binding $PROJECT_ID
  --member serviceAccount:$SERVICE_ACCOUNT_EMAIL
  --role projects/$PROJECT_ID/roles/velero.server

gsutil iam ch serviceAccount:$SERVICE_ACCOUNT_EMAIL:objectAdmin gs://${BUCKET}
```

8. Save the service account’s keys to the `credentials-velero` file in the current directory:

```sh
$ gcloud iam service-accounts keys create credentials-velero
  --iam-account $SERVICE_ACCOUNT_EMAIL
```

### 1.4.5. Configuring a Microsoft Azure Blob storage container as a replication repository

You can configure a Microsoft Azure Blob storage container as a replication repository.

**Prerequisites**

- You must have an [Azure storage account](#).
- You must have the [Azure CLI](#) installed.
- The Azure Blob storage container must be accessible to the source and target clusters.
- If you are using the snapshot copy method:
  - The source and target clusters must be in the same region.
- The source and target clusters must have the same storage class.
- The storage class must be compatible with snapshots.

**Procedure**

1. Set the `AZURE_RESOURCE_GROUP` variable:
   ```bash
   $ AZURE_RESOURCE_GROUP=Velero_Backups
   ```

2. Create an Azure resource group:
   ```bash
   $ az group create -n $AZURE_RESOURCE_GROUP --location <CentralUS>  
   1 Specify your location.
   ```

3. Set the `AZURE_STORAGE_ACCOUNT_ID` variable:
   ```bash
   $ AZURE_STORAGE_ACCOUNT_ID=velerobackups
   ```

4. Create an Azure storage account:
   ```bash
   $ az storage account create 
   --name $AZURE_STORAGE_ACCOUNT_ID 
   --resource-group $AZURE_RESOURCE_GROUP 
   --sku Standard_GRS 
   --encryption-services blob 
   --https-only true 
   --kind BlobStorage 
   --access-tier Hot
   ```

5. Set the `BLOB_CONTAINER` variable:
   ```bash
   $ BLOB_CONTAINER=velero
   ```

6. Create an Azure Blob storage container:
   ```bash
   $ az storage container create 
   -n $BLOB_CONTAINER 
   --public-access off 
   --account-name $AZURE_STORAGE_ACCOUNT_ID
   ```

7. Create a service principal and credentials for `velero`:
   ```bash
   $ AZURE_SUBSCRIPTION_ID=`az account list --query '[?isDefault].id' -o tsv`
   $ AZURE_TENANT_ID=`az account list --query '[?isDefault].tenantId' -o tsv`
   $ AZURE_CLIENT_SECRET=`az ad sp create-for-rbac --name "velero" --role "Contributor" --query 'password' -o tsv`
   $ AZURE_CLIENT_ID=`az ad sp list --display-name "velero" --query '[0].appId' -o tsv`
   ```

8. Save the service principal’s credentials in the `credentials-velero` file:
1.5. DEPLOYING THE CLUSTER APPLICATION MIGRATION (CAM) TOOL

Deploying the Cluster Application Migration (CAM) tool requires installing the Cluster Application Migration Operator on the OpenShift Container Platform 3 source and OpenShift Container Platform 4.3 target clusters.

1.5.1. Installing the Cluster Application Migration Operator on an OpenShift Container Platform 3 source cluster

You can install the Cluster Application Migration Operator manually on an OpenShift Container Platform 3.7 (or later) source cluster, which does not support OLM.

Prerequisites

- You must have podman installed.
- Your OpenShift Container Platform 3 cluster must be configured to pull images from registry.redhat.io.
  To pull images, you must create an imagestreamsecret and copy it to each node in your cluster.

Procedure

1. Log in to registry.redhat.io with your Red Hat Customer Portal credentials:

   ```bash
   $ sudo podman login registry.redhat.io
   ```

   **NOTE**
   If your system is configured for rootless Podman containers, sudo is not required for this procedure.

2. Download the operator.yml file:

   ```bash
   $ sudo podman cp $(sudo podman create registry.redhat.io/rhcam-1-1/openshift-migration-rhel7-operator:v1.1):/operator.yml ./
   ```

3. Download the controller-3.yml file:

   ```bash
   $ sudo podman cp $(sudo podman create registry.redhat.io/rhcam-1-1/openshift-migration-rhel7-operator:v1.1):/controller-3.yml ./
   ```
4. Log in to your OpenShift Container Platform 3 cluster.

5. Verify that the cluster can authenticate with registry.redhat.io:
   
   ```
   $ oc run test --image registry.redhat.io/ubi8 --command sleep infinity
   ```

6. Create the Cluster Application Migration Operator CR object:
   
   ```
   $ oc create -f operator.yml
   ```
   
   namespace/openshift-migration created
   rolebinding.rbac.authorization.k8s.io/system:deployers created
   serviceaccount/migration-operator created
   customresourcedefinition.apiextensions.k8s.io/migrationcontrollers.migration.openshift.io created
   role.rbac.authorization.k8s.io/migration-operator created
   rolebinding.rbac.authorization.k8s.io/migration-operator created
   clusterrolebinding.rbac.authorization.k8s.io/migration-operator created
   deployment.apps/migration-operator created
   
   Error from server (AlreadyExists): error when creating "/operator.yml":
   rolebindings.rbac.authorization.k8s.io "system:image-builders" already exists
   Error from server (AlreadyExists): error when creating "/operator.yml":
   rolebindings.rbac.authorization.k8s.io "system:image-pullers" already exists
   
   You can ignore the Error from server (AlreadyExists) messages. They are caused by the Cluster Application Migration Operator creating resources for earlier versions of OpenShift Container Platform 3 that are provided in later releases.

7. Create the Migration controller CR object:
   
   ```
   $ oc create -f controller-3.yml
   ```

8. Verify that the Velero and Restic Pods are running:
   
   ```
   $ oc get pods -n openshift-migration
   ```

1.5.2. Installing the Cluster Application Migration Operator on an OpenShift Container Platform 4.3 target cluster

You can install the Cluster Application Migration Operator on an OpenShift Container Platform 4.3 target cluster with OLM.

The Cluster Application Migration Operator installs the CAM tool on the target cluster by default. If you want to install the CAM tool on a different cluster, you must update the Migration controller configuration so that the Cluster Application Migration Operator does not install the CAM tool on this cluster.

Procedure

1. In the OpenShift Container Platform web console, click Administration → Namespaces.

2. Click Create Namespace.

3. Enter openshift-migration in the Name field and click Create.
4. Click Operators → OperatorHub.

5. Use the Filter by keyword field (in this case, Migration) to find the Cluster Application Migration Operator.

6. Select the Cluster Application Migration Operator and click Install.

7. On the Create Operator Subscription page, select the openshift-migration namespace, and specify an approval strategy.

8. Click Subscribe.
   On the Installed Operators page, the Cluster Application Migration Operator appears in the openshift-migration project with the status InstallSucceeded.

9. Under Provided APIs, click View 12 more....

10. Click Create New → MigrationController.

11. If you do not want to install the CAM tool on the target cluster, update the migration_controller and migration_ui parameters in the spec stanza:

```yaml
spec:
  [...]  
migration_controller: false
migration_ui: false
  [...]  
```

12. Click Create.

13. Click Workloads → Pods to verify that the Restic and Velero Pods are running.

14. Click Workloads → Pods to verify that the Controller Manager, Migration UI, Restic, and Velero Pods are running.

1.6. MIGRATING APPLICATIONS WITH THE CAM WEB CONSOLE

1.6.1. Launching the CAM web console

You can launch the CAM web console that is installed on the target cluster.

Procedure

1. Log in to the target cluster.

2. Obtain the CAM web console URL:

   ```shell
   $ oc get -n openshift-migration route/migration -o go-template='(?i)//{{ .spec.host }}(:|\z){{
   println }}' | sed 's,,\.,\.,g'
   ```

3. Launch a browser and navigate to the CAM web console.
NOTE
If you try to access the CAM web console immediately after installing the Cluster Application Migration Operator, the console may not load because the Operator is still configuring the cluster. Wait a few minutes and retry.

4. If you are using self-signed CA certificates, you will be prompted to accept the CA certificate of the source cluster’s API server. The web page will guide you through the process of accepting the remaining certificates.

5. Log in with your OpenShift Container Platform username and password.

1.6.2. Adding a cluster to the CAM web console

You can add a source cluster to the CAM web console.

Prerequisites for Azure

If you are using Azure snapshots to copy data: * You must provide the Azure resource group name when you add the source cluster. * The source and target clusters must be in the same Azure resource group and in the same location.

Procedure

1. Log in to the source cluster.

2. Obtain the service account token:

   ```bash
   $ oc sa get-token mig -n openshift-migration
   eyJhbGciOiJSUzI1NiIsImtpZCI6IiJ9.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50iwi3ViZXJuZXRlc3pbybpbmciXCI6NTE1MzUyMCIsImNlbnUiOiJ1c2VyIn0.
   ```

3. Log in to the CAM web console.

4. In the Clusters section, click Add cluster.

5. Fill in the following fields:

   - **Cluster name**: May contain lower-case letters (a-z) and numbers (0-9). Must not contain spaces or international characters.
   - **Url**: URL of the cluster’s API server, for example, `https://<master1.example.com>:8443`.
   - **Service account token**: String that you obtained from the source cluster.
- **Azure cluster**: Optional. Select it if you are using Azure snapshots to copy your data.
- **Azure resource group**: This field appears if **Azure cluster** is checked.

6. Click **Add cluster**.
   The cluster appears in the **Clusters** section.

### 1.6.3. Adding a replication repository to the CAM web console

You can add an object storage bucket as a replication repository to the CAM web console.

**Prerequisites**
- You must configure an object storage bucket for migrating the data.

**Procedure**

1. Log in to the CAM web console.
2. In the **Replication repositories** section, click **Add repository**.
3. Select a **Storage provider type** and fill in the following fields:
   - **AWS** for AWS S3, MCG, and generic S3 providers:
     - **Replication repository name**: Specify the replication repository name in the CAM web console.
     - **S3 bucket name**: Specify the name of the S3 bucket you created.
     - **S3 bucket region**: Specify the S3 bucket region. **Required** for AWS S3. **Optional** for other S3 providers.
     - **S3 endpoint**: Specify the URL of the S3 service, not the bucket, for example, `https://<s3-storage.apps.cluster.com>`. **Required** for a generic S3 provider. You must use the `https://` prefix.
     - **S3 provider access key**: Specify the `<AWS_SECRET_ACCESS_KEY>` for AWS or the S3 provider access key for MCG.
     - **S3 provider secret access key**: Specify the `<AWS_ACCESS_KEY_ID>` for AWS or the S3 provider secret access key for MCG.
     - **Require SSL verification**: Clear this check box if you are using a generic S3 provider.
   - **GCP**:
     - **Replication repository name**: Specify the replication repository name in the CAM web console.
     - **GCP bucket name**: Specify the name of the GCP bucket.
     - **GCP credential JSON blob**: Specify the string in the `credentials-velero` file.
   - **Azure**:
- **Replication repository name**: Specify the replication repository name in the CAM web console.

- **Azure resource group**: Specify the resource group of the Azure Blob storage.

- **Azure storage account name**: Specify the Azure Blob storage account name.

- **Azure credentials - INI file contents**: Specify the string in the `credentials-velero` file.

4. Click **Add repository** and wait for connection validation.

5. Click **Close**.
   The new repository appears in the **Replication repositories** section.

### 1.6.4. Changing migration plan limits for large migrations

You can change the migration plan limits for large migrations.

**IMPORTANT**

Changes should first be tested in your environment to avoid a failed migration.

A single migration plan has the following default limits:

- **10 namespaces**
  If this limit is exceeded, the CAM web console displays a **Namespace limit exceeded** error and you cannot create a migration plan.

- **100 Pods**
  If the Pod limit is exceeded, the CAM web console displays a warning message similar to the following example: **Plan has been validated with warning condition(s). See warning message. Pod limit: 100 exceeded, found: 104.**

- **100 persistent volumes**
  If the persistent volume limit is exceeded, the CAM web console displays a similar warning message.

**Procedure**

1. Edit the Migration controller CR:

   ```bash
   $ oc get migrationcontroller -n openshift-migration
   NAME AGE
   migration-controller 5d19h
   $ oc edit migrationcontroller -n openshift-migration
   ```

2. Update the following parameters:

   ```yaml
   [...]  
   migration_controller: true
   # This configuration is loaded into mig-controller, and should be set on the
   # cluster where 'migration_controller: true'
   mig_pv_limit: 100
   ```
1.6.5. Creating a migration plan in the CAM web console

You can create a migration plan in the CAM web console.

Prerequisites

- The CAM web console must contain the following:
  - Source cluster
  - Target cluster, which is added automatically during the CAM tool installation
  - Replication repository
- If you want to copy your data by using snapshots, the source and target clusters must be running on the same cloud provider (AWS, GCP, or Azure) and in the same region.

Procedure

1. Log in to the CAM web console.
2. In the Plans section, click Add plan.
3. Enter the Plan name and click Next. The Plan name can contain up to 253 lower-case alphanumeric characters (a-z, 0-9). It must not contain spaces or underscores (_).
4. Select a Source cluster.
5. Select a Target cluster.
6. Select a Replication repository.
7. Select the projects to be migrated and click Next.
8. Select Copy or Move for the PVs:
   - Copy copies the data in a source cluster’s PV to the replication repository and then restores it on a newly created PV, with similar characteristics, in the target cluster.
   - Move unmounts a remote volume (for example, NFS) from the source cluster, creates a PV resource on the target cluster pointing to the remote volume, and then mounts the remote volume on the target cluster. Applications running on the target cluster use the same remote volume that the source cluster was using. The remote volume must be accessible to the source and target clusters.
9. Click Next.
10. Select a Copy method for the PVs:
    - Snapshot backs up and restores the disk using the cloud provider’s snapshot functionality. It is significantly faster than Filesystem.
NOTE
The storage and clusters must be in the same region and the storage class must be compatible.

- **Filesystem** copies the data files from the source disk to a newly created target disk.

11. Select a **Storage class** for the PVs.
   If you selected the **Filesystem** copy method, you can change the storage class during migration, for example, from Red Hat Gluster Storage or NFS storage to Red Hat Ceph Storage.

12. Click **Finish**.

13. Click **Close**.
   The migration plan appears in the **Plans** section.

1.6.6. Running a migration plan in the CAM web console
You can stage or migrate applications and data with the migration plan you created in the CAM web console.

**Prerequisites**
The CAM web console must contain the following:

- Source cluster
- Target cluster, which is added automatically during the CAM tool installation
- Replication repository
- Valid migration plan

**Procedure**

1. Log in to the CAM web console on the target cluster.

2. Select a migration plan.

3. Click **Stage** to copy data from the source cluster to the target cluster without stopping the application.
   You can run **Stage** multiple times to reduce the actual migration time.

4. When you are ready to migrate the application workload, click **Migrate**.
   **Migrate** stops the application workload on the source cluster and recreates its resources on the target cluster.

5. Optionally, in the **Migrate** window, you can select **Do not stop applications on the source cluster during migration**.

6. Click **Migrate**.

7. When the migration is complete, verify that the application migrated successfully in the OpenShift Container Platform web console:
   a. Click **Home → Projects**.
b. Click the migrated project to view its status.

c. In the Routes section, click Location to verify that the application is functioning, if applicable.

d. Click Workloads → Pods to verify that the Pods are running in the migrated namespace.

e. Click Storage → Persistent volumes to verify that the migrated persistent volume is correctly provisioned.

1.7. MIGRATING CONTROL PLANE SETTINGS WITH THE CONTROL PLANE MIGRATION ASSISTANT (CPMA)

1.7.1. Understanding the Control Plane Migration Assistant

The Control Plane Migration Assistant (CPMA) is a CLI-based tool that assists you in migrating the control plane from OpenShift Container Platform 3.7 (or later) to OpenShift Container Platform 4.3. The CPMA processes the OpenShift Container Platform 3 configuration files and generates Custom Resource (CR) manifest files, which are consumed by OpenShift Container Platform 4.3 Operators.

Because OpenShift Container Platform 3 and 4 have significant configuration differences, not all parameters are processed. The CPMA can generate a report that describes whether features are supported fully, partially, or not at all.

Configuration files

CPMA uses the Kubernetes and OpenShift Container Platform APIs to access the following configuration files on an OpenShift Container Platform 3 cluster:

- Master configuration file (default: /etc/origin/master/master-config.yaml)
- CRI-O configuration file (default: /etc/crio/crio.conf)
- etcd configuration file (default: /etc/etcd/etcd.conf)
- Image registries file (default: /etc/containers/registries.conf)
- Dependent configuration files:
  - Password files (for example, HTPasswd)
  - ConfigMaps
  - Secrets

CR Manifests

CPMA generates CR manifests for the following configurations:

- API server CA certificate: 100_CPMA-cluster-config-APISecret.yaml

**NOTE**

If you are using an unsigned API server CA certificate, you must add the certificate manually to the target cluster.
1.7.2. Installing the Control Plane Migration Assistant

You can download the Control Plane Migration Assistant (CPMA) binary file from the Red Hat Customer Portal and install it on Linux, MacOSX, or Windows operating systems.

Procedure

1. In the Red Hat Customer Portal, navigate to Downloads → Red Hat OpenShift Container Platform.
3. Select CPMA 1.0 for RHEL 7 from the Version list. This binary works on RHEL 7 and RHEL 8.
4. Click Download Now to download cpma for Linux or MacOSX or cpma.exe for Windows.
5. Save the file in a directory defined as $PATH for Linux or MacOSX or %PATH% for Windows.
6. For Linux, make the file executable:

   $ sudo chmod +x cpma

1.7.3. Using the Control Plane Migration Assistant

The Control Plane Migration Assistant (CPMA) generates CR manifests, which are consumed by OpenShift Container Platform 4.3 Operators, and a report that indicates which OpenShift Container Platform 3 features are supported fully, partially, or not at all.

The CPMA can run in remote mode, retrieving the configuration files from the source cluster using SSH, or in local mode, using local copies of the source cluster’s configuration files.

Prerequisites

- The source cluster must be OpenShift Container Platform 3.7 or later.
- The source cluster must be updated to the latest synchronous release.
An environment health check must be run on the source cluster to confirm that there are no diagnostic errors or warnings.

The CPMA binary must be executable.

You must have `cluster-admin` privileges for the source cluster.

**Procedure**

1. Log in to the OpenShift Container Platform 3 cluster:

   ```shell
   $ oc login https://<master1.example.com>
   ```

   OpenShift Container Platform 3 master node. You must be logged in to the cluster to receive a token for the Kubernetes and OpenShift Container Platform APIs.

2. Run the CPMA. Each prompt requires you to provide input, as in the following example:

   ```shell
   $ cpma --manifests=false
   ? Do you wish to save configuration for future use? true
   ? What will be the source for OCP3 config files? Remote host
   ? Path to crio config file /etc/crio/crio.conf
   ? Path to etcd config file /etc/etcd/etcd.conf
   ? Path to master config file /etc/origin/master/master-config.yaml
   ? Path to node config file /etc/origin/node/node-config.yaml
   ? Path to registries config file /etc/containers/registries.conf
   ? Do wish to find source cluster using KUBECONFIG or prompt it? KUBECONFIG
   ? Select cluster obtained from KUBECONFIG contexts master1-example-com:443
   ? Select master node master1.example.com
   ? SSH login root
   ? SSH Port 22
   ? Path to private SSH key /home/user/.ssh/openshift_key
   ? Path to application data, skip to use current directory.
   INFO[29 Aug 19 00:07 UTC] Starting manifest and report generation
   INFO[29 Aug 19 00:07 UTC] Transform: Starting for - API
   INFO[29 Aug 19 00:07 UTC] APITransform::Extract
   INFO[29 Aug 19 00:07 UTC] APITransform::Transform:Reports
   INFO[29 Aug 19 00:07 UTC] Transform: Starting for - Cluster
   INFO[29 Aug 19 00:08 UTC] ClusterTransform::Transform:Reports
   INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportQuotas
   INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportPVs
   INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportNamespaces
   INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportNodes
   INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportRBAC
   INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportStorageClasses
   INFO[29 Aug 19 00:08 UTC] Transform: Starting for - Crio
   INFO[29 Aug 19 00:08 UTC] CrioTransform::Extract
   WARN[29 Aug 19 00:08 UTC] Skipping Crio: No configuration file available
   INFO[29 Aug 19 00:08 UTC] Transform: Starting for - Docker
   INFO[29 Aug 19 00:08 UTC] DockerTransform::Extract
   INFO[29 Aug 19 00:08 UTC] DockerTransform::Transform:Reports
   INFO[29 Aug 19 00:08 UTC] Transform: Starting for - ETCD
   INFO[29 Aug 19 00:08 UTC] ETCDTransform::Extract
   INFO[29 Aug 19 00:08 UTC] ETCDTransform::Transform:Reports
   ```
INFO[29 Aug 19 00:08 UTC] Transform:Starting for - OAuth
INFO[29 Aug 19 00:08 UTC] OAuthTransform::Extract
INFO[29 Aug 19 00:08 UTC] OAuthTransform::Transform::Reports
INFO[29 Aug 19 00:08 UTC] Transform::Starting for - SDN
INFO[29 Aug 19 00:08 UTC] SDNTransform::Extract
INFO[29 Aug 19 00:08 UTC] SDNTransform::Transform::Reports
INFO[29 Aug 19 00:08 UTC] Transform::Starting for - Image
INFO[29 Aug 19 00:08 UTC] ImageTransform::Extract
INFO[29 Aug 19 00:08 UTC] ImageTransform::Transform::Reports
INFO[29 Aug 19 00:08 UTC] Transform::Starting for - Project
INFO[29 Aug 19 00:08 UTC] ProjectTransform::Extract
INFO[29 Aug 19 00:08 UTC] ProjectTransform::Transform::Reports
INFO[29 Aug 19 00:08 UTC] Flushing reports to disk
INFO[29 Aug 19 00:08 UTC] Report:Added: report.json
INFO[29 Aug 19 00:08 UTC] Report:Added: report.html
INFO[29 Aug 19 00:08 UTC] Successfully finished transformations

--manifests=false: Without generating CR manifests

Remote host: Remote mode

SSH login: The SSH user must have sudo permissions on the OpenShift Container Platform 3 cluster in order to access the configuration files.

The CPMA creates the following files and directory in the current directory if you did not specify an output directory:

- cpma.yaml file: Configuration options that you provided when you ran the CPMA
- master1.example.com/: Configuration files from the master node
- report.json: JSON-encoded report
- report.html: HTML-encoded report

3. Open the report.html file in a browser to view the CPMA report.

4. If you generate CR manifests, apply the CR manifests to the OpenShift Container Platform 4.3 cluster, as in the following example:

   $ oc apply -f 100_CPMA-cluster-config-secret-htpasswd-secret.yaml

1.8. TROUBLESHOOTING

You can view the migration Custom Resources (CRs) and download logs to troubleshoot a failed migration.

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

**NOTE**

Manual rollback is not required if the application was not stopped during migration, because the original application is still running on the source cluster.
1.8.1. Viewing migration Custom Resources (CRs)

The Cluster Application Migration (CAM) tool creates the following CRs for migration:

- **MigCluster** (configuration, CAM cluster): Cluster definition
- **MigStorage** (configuration, CAM cluster): Storage definition
- **MigPlan** (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.

**NOTE**

Deleting a MigPlan CR deletes the associated MigMigration CRs.

- **BackupStorageLocation** (configuration, CAM cluster): Location of Velero backup objects
- **VolumeSnapshotLocation** (configuration, CAM cluster): Location of Velero volume snapshots
- **MigMigration** (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.

7 Backup (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

8 Restore (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

Procedure

1. Obtain the CR name:

   ```
   $ oc get <cr> -n openshift-migration
   NAME       AGE
   88435fe0-c9f8-11e9-85e6-5d593ce65e10 6m42s
   ```

   Specify the migration CR you want to view.

2. View the CR:

   ```
   $ oc describe <cr> 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
   ```

   The output is similar to the following examples.

MigMigration example

   ```
   $ oc describe migmigration 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
   Name: 88435fe0-c9f8-11e9-85e6-5d593ce65e10
   Namespace: openshift-migration
   Labels: <none>
   Annotations: touch: 3b48b543-b53e-4e44-9d34-33563f0f8147
   API Version: migration.openshift.io/v1alpha1
   Kind: MigMigration
   Metadata:
   Creation Timestamp: 2019-08-29T01:01:29Z
   Generation: 20
   Resource Version: 88179
   Self Link: /apis/migration.openshift.io/v1alpha1/namespaces/openshift-migration/migmigrations/88435fe0-c9f8-11e9-85e6-5d593ce65e10
   UID: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
   Spec:
   Mig Plan Ref:
   Name: socks-shop-mig-plan
   Namespace: openshift-migration
   ```
Quiesce Pods:  true
Stage:         false
Status:       

Conditions:
  Category:              Advisory
  Durable:               true
  Last Transition Time:  2019-08-29T01:03:40Z
  Message:               The migration has completed successfully.
  Reason:                Completed
  Status:                True
  Type:                  Succeeded
  Phase:                   Completed
  Start Timestamp:         2019-08-29T01:01:29Z
  Events:                    <none>

Velero backup CR #2 example (PV data)

apiVersion: velero.io/v1
kind: Backup
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.105.179:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-44dd3bd5-c9f8-11e9-95ad-0205fe66cbb6
  creationTimestamp: "2019-08-29T01:03:15Z"
  generateName: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-
  generation: 1
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    migration-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    velero.io/storage-location: myrepo-vpzo9
  name: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
  namespace: openshift-migration
  resourceVersion: "87313"
  selfLink: /apis/velero.io/v1/namespaces/openshift-migration/backups/88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
  uid: c80dbbc0-c9f8-11e9-95ad-0205fe66cbb6
spec:
  excludedNamespaces: []
  excludedResources: []
  hooks:
    resources: []
  includeClusterResources: null
  includedNamespaces: 
    - sock-shop
  includedResources: 
    - persistentvolumes
    - persistentvolumeclaims
    - namespaces
    - imagestreams
    - imagestreamtags
    - secrets
    - configmaps

OpenShift Container Platform 4.3 Migration
36
CHAPTER 1. MIGRATING OPENSOURCE CONTAINER PLATFORM 3 TO 4

- pods
  labelSelector:
    matchLabels:
      migration-included-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
  storageLocation: myrepo-vpzq9
  ttl: 720h0m0s
  volumeSnapshotLocations:
  - myrepo-wv6fx
  status:
    completionTimestamp: "2019-08-29T01:02:36Z"
    errors: 0
    expiration: "2019-09-28T01:02:35Z"
    phase: Completed
    startTimestamp: "2019-08-29T01:02:35Z"
    validationErrors: null
    version: 1
    volumeSnapshotsAttempted: 0
    volumeSnapshotsCompleted: 0
    warnings: 0

Velero restore CR #2 example (Kubernetes resources)

apiVersion: velero.io/v1
kind: Restore
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.90.187:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-36f54ca7-c925-11e9-825a-06fa9fb68c88
  creationTimestamp: "2019-08-28T00:09:49Z"
  generateName: e13a1b60-c927-11e9-9555-d129df7f3b96-
generation: 3
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: e18252c9-c927-11e9-825a-06fa9fb68c88
    migration-final-restore: e18252c9-c927-11e9-825a-06fa9fb68c88
    name: e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
  namespace: openshift-migration
  resourceVersion: "82329"
  selfLink: /apis/velero.io/v1/namespaces/openshift-migration/restores/e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
  uid: 26983ec0-c928-11e9-825a-06fa9fb68c88
spec:
  backupName: e13a1b60-c927-11e9-9555-d129df7f3b96-sz24f
  excludedNamespaces: null
  excludedResources: null
  - nodes
  - events
  - events.events.k8s.io
  - backups.velero.io
  - restores.velero.io
  - resticrepositories.velero.io
  includedNamespaces: null
  includedResources: null
1.8.2. 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 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:
   a. 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.

   b. Click Download Selected to download the selected log.

Optionally, you can 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
```

1.8.3. Restic timeout error

If a migration fails because Restic times out, the following error appears in the Velero log:

```
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. You can increase this for large migrations, keeping in mind that a higher value may delay the return of error messages.

**Procedure**

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:

   ```yaml
   spec:
     restic_timeout: 1h
   ```

   Valid units are **h** (hours), **m** (minutes), and **s** (seconds), for example, **3h30m15s**.
5. Click **Save**.

**1.8.4. Manually rolling back a migration**

If your application was stopped during a failed migration, you must roll it back manually in order to prevent data corruption in the PV.

This procedure is not required if the application was not stopped during migration, because the original application is still running on the source cluster.

**Procedure**

1. On the target cluster, switch to the migrated project:

   ```bash
   $ oc project <project>
   ```
2. Get the deployed resources:

   ```bash
   $ oc get all
   ```
3. Delete the deployed resources to ensure that the application is not running on the target cluster and accessing data on the PVC:

   ```bash
   $ oc delete <resource_type>
   ```
4. To stop a DaemonSet without deleting it, update the **nodeSelector** in the YAML file:

   ```yaml
   apiVersion: extensions/v1beta1
   kind: DaemonSet
   metadata:
   ```
1. Specify a `nodeSelector` value that does not exist on any node.

5. 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 the source cluster. You can remove these PVs during rollback.

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0001
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
```

Specify **Recycle** or **Delete**.

6. On the 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>
```

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

### 1.8.5. Known issues

This release has the following known issues:

- During migration, the Cluster Application Migration (CAM) tool preserves the following namespace annotations:
- `openshift.io/sa.scc.mcs`

- `openshift.io/sa.scc.supplemental-groups`

- `openshift.io/sa.scc.uid-range`
  These annotations preserve the UID range, ensuring that the containers retain their file system permissions on the target cluster. There is a risk that the migrated UIDs could duplicate UIDs within an existing or future namespace on the target cluster. ([BZ#1748440](#))

- If an AWS bucket is added to the CAM web console and then deleted, its status remains `True` because the MigStorage CR is not updated. ([BZ#1738564](#))

- Migration fails if the Migration controller is running on a cluster other than the target cluster. The `EnsureCloudSecretPropagated` phase is skipped with a logged warning. ([BZ#1757571](#))

- Most cluster-scoped resources are not yet handled by the CAM tool. If your applications require cluster-scoped resources, you may have to create them manually on the target cluster.

- Incorrect source cluster storage class is displayed when creating the migration plan. ([BZ#1777869](#))

- If a cluster in the CAM web console becomes inaccessible, it blocks attempts to close open migration plans. ([BZ#1758269](#))

- If a migration fails, the migration plan does not retain custom PV settings for quiesced pods. You must manually roll back the migration, delete the migration plan, and create a new migration plan with your PV settings. ([BZ#1784899](#))
CHAPTER 2. MIGRATING OPENSSHIFT CONTAINER PLATFORM 4.1 TO 4.3

2.1. MIGRATING APPLICATION WORKLOADS FROM OPENSHEET CONTAINER PLATFORM 4.1 TO 4.3

You can migrate application workloads from OpenShift Container Platform 4.1 to 4.3 with the Cluster Application Migration (CAM) tool. The CAM tool enables you to control the migration and to minimize application downtime.

The CAM tool’s web console and API, based on Kubernetes Custom Resources, enable you to migrate stateful and stateless application workloads at the granularity of a namespace.

2.1.1. Migration prerequisites

- You must have `cluster-admin` privileges on all clusters.
- You must configure a replication repository that is accessible to the source and target clusters.
- If your application uses images from the `openshift` namespace, the required versions of the images must be present on the target cluster. If the required images are not present, you must update the `imagestreamtags` references to use an available version that is compatible with your application. If the `imagestreamtags` cannot be updated, you can manually upload equivalent images to the application namespaces and update the applications to reference them.

The following `imagestreamtags` have been removed from OpenShift Container Platform 4.3:

- dotnet:1.0, dotnet:1.1, dotnet:2.0
- dotnet-runtime:2.0
- mariadb:10.1
- mongodb:2.4, mongodb:2.6
- mysql:5.5, mysql:5.6
- nginx:1.8
- nodejs:0.10, nodejs:4, nodejs:6
- perl:5.16, perl:5.20
- php:5.5, php:5.6
- python:3.3, python:3.4
- ruby:2.0, ruby:2.2

2.1.2. Understanding the Cluster Application Migration tool
The Cluster Application Migration (CAM) tool enables you to migrate Kubernetes resources, persistent volume data, and internal container images from an OpenShift Container Platform source cluster to an OpenShift Container Platform 4.3 target cluster, using the CAM web console or the Kubernetes API.

Migrating an application with the CAM web console involves the following steps:

1. Install the Cluster Application Migration Operator on all clusters.

   **NOTE**
   
The Cluster Application Migration Operator installs the CAM tool (CAM web console and Migration controller) on the target cluster by default. You can configure the Migration controller to install the CAM tool on another cluster. The cluster on which the CAM tool is installed is referred to as the CAM tool cluster.

2. Configure the replication repository, an intermediate object storage that the CAM tool uses to migrate data.

3. Add the source cluster to the CAM web console.

4. Add the replication repository to the CAM web console.

5. Create a migration plan, with one of the following data migration options:

   - **Copy**: The CAM tool copies the data from the source cluster to the replication repository, and from the replication repository to the target cluster.

   ![Diagram](image)

   - **Move**: The CAM tool unmounts a remote volume (for example, NFS) from the source cluster, creates a PV resource on the target cluster pointing to the remote volume, and then mounts the remote volume on the target cluster. Applications running on the target cluster use the same remote volume that the source cluster was using. The remote volume must be accessible to the source and target clusters.

   **NOTE**
   
   Although the replication repository does not appear in this diagram, it is required for the actual migration.
6. Run the migration plan, with one of the following options:

- **Stage** (optional) copies data to the target cluster without stopping the application. Staging can be run multiple times so that most of the data is copied to the target before migration. This minimizes the actual migration time and application downtime.

- **Migrate** stops the application on the source cluster and recreates its resources on the target cluster. Optionally, you can migrate the workload without stopping the application.

2.2. CONFIGURING A REPLICATION REPOSITORY

You must configure an object storage to use as a replication repository. The Cluster Application Migration tool copies data from the source cluster to the replication repository, and then from the replication repository to the target cluster, using either the **file system** or the **snapshot** data copy method.

The following storage providers are supported:

- Generic S3 object storage, for example, Minio or Ceph S3
- Multi-Cloud Object Gateway (MCG)
2.2.1. Understanding the data copy methods for migration

The CAM tool supports the file system and snapshot data copy methods for migrating data from the source cluster to the target cluster. You can select a method that is suited for your environment and is supported by your storage provider.

2.2.1.1. File system copy method

The CAM tool copies data files from the source cluster to the replication repository, and from there to the target cluster.

Table 2.1. File system copy method summary

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Clusters can have different storage classes</td>
<td>● Slower than the snapshot copy method</td>
</tr>
<tr>
<td>● Supported for all S3 storage providers</td>
<td></td>
</tr>
</tbody>
</table>

2.2.1.2. Snapshot copy method

The CAM tool copies a snapshot of the source cluster’s data to a cloud provider’s object storage, configured as a replication repository. The data is restored on the target cluster.

AWS, Google Cloud Provider, and Microsoft Azure support the snapshot copy method.

Table 2.2. Snapshot copy method summary

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Faster than the file system copy method</td>
<td>● Cloud provider must support snapshots.</td>
</tr>
<tr>
<td></td>
<td>● Clusters must be on the same cloud provider.</td>
</tr>
<tr>
<td></td>
<td>● Clusters must be in the same location or region.</td>
</tr>
<tr>
<td></td>
<td>● Clusters must have the same storage class.</td>
</tr>
<tr>
<td></td>
<td>● Storage class must be compatible with snapshots.</td>
</tr>
</tbody>
</table>

2.2.2. Configuring a Multi-Cloud Object Gateway storage bucket as a replication repository
You can install the OpenShift Container Storage Operator and configure a Multi-Cloud Object Gateway (MCG) storage bucket as a replication repository on the CAM tool cluster.

### 2.2.2.1. Installing the OpenShift Container Storage Operator

You can install the OpenShift Container Storage Operator from OperatorHub on the CAM tool cluster.

**Procedure**

1. In the OpenShift Container Platform web console, click **Administration → Namespaces**.
2. Click **Create Namespace**.
3. Enter `openshift-storage` in the **Name** field and click **Create**.
4. Click **Operators → OperatorHub**.
5. Use **Filter by keyword** (in this case, OCS) to find the **OpenShift Container Storage Operator**.
6. Select the **OpenShift Container Storage Operator** and click **Install**.
7. On the **Create Operator Subscription** page, select the `openshift-storage` namespace.
8. Specify your update channel and approval strategy.
9. Click **Subscribe**.
   
   On the **Installed Operators** page, the **OpenShift Container Storage Operator** appears in the `openshift-storage` project with the status **Succeeded**.

### 2.2.2.2. Creating the Multi-Cloud Object Gateway storage bucket

You can create the Multi-Cloud Object Gateway (MCG) storage bucket’s Custom Resources (CRs) on the CAM tool cluster.

**Procedure**

1. Log in to the CAM tool cluster.
2. Create the **NooBaa** CR configuration file, `noobaa.yml`, with the following content:

```yaml
apiVersion: noobaa.io/v1alpha1
kind: NooBaa
metadata:
  name: noobaa
  namespace: openshift-storage
spec:
  dbResources:
    requests:
      cpu: 0.5
      memory: 1Gi
  coreResources:
    requests:
      cpu: 0.5
      memory: 1Gi
```
For a very small cluster, you can change the cpu value to 0.1.

3. Create the **NooBaa** object:

   ```bash
   $ oc create -f noobaa.yml
   ``

4. Create the **BackingStore** CR configuration file, `bs.yml`, with the following content:

   ```yaml
   apiVersion: noobaa.io/v1alpha1
   kind: BackingStore
   metadata:
     finalizers:
     - noobaa.io/finalizer
     labels:
       app: noobaa
       name: mcg-pv-pool-bs
       namespace: openshift-storage
   spec:
     pvPool:
       numVolumes: 3
       resources:
         requests:
           storage: 50Gi
       storageClass: gp2
       type: pv-pool
   ``

   1. Specify the number of volumes in the PV pool.
   2. Specify the size of the volumes.
   3. Specify the storage class.

5. Create the **BackingStore** object:

   ```bash
   $ oc create -f bs.yml
   ``

6. Create the **BucketClass** CR configuration file, `bc.yml`, with the following content:

   ```yaml
   apiVersion: noobaa.io/v1alpha1
   kind: BucketClass
   metadata:
     labels:
       app: noobaa
       name: mcg-pv-pool-bc
       namespace: openshift-storage
   spec:
     placementPolicy:
       tiers:
         - backingStores:
           - mcg-pv-pool-bs
             placement: Spread
   ``

7. Create the **BucketClass** object:
8. Create the ObjectBucketClaim CR configuration file, `obc.yml`, with the following content:

```yaml
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: migstorage
  namespace: openshift-storage
spec:
  bucketName: migstorage
  storageClassName: openshift-storage.noobaa.io
  additionalConfig:
    bucketclass: mcg-pv-pool-bc
```

1. Record the bucket name for adding the replication repository to the CAM web console.

9. Create the ObjectBucketClaim object:

```bash
$ oc create -f obc.yml
```

10. Watch the resource creation process to verify that the ObjectBucketClaim status is Bound:

```bash
$ watch -n 30 'oc get -n openshift-storage objectbucketclaim migstorage -o yaml'
```

This process can take five to ten minutes.

11. Obtain and record the following values, which are required when you add the replication repository to the CAM web console:

- S3 endpoint:
  ```bash
  $ oc get route -n openshift-storage s3
  ```

- S3 provider access key:
  ```bash
  $ oc get secret -n openshift-storage migstorage -o go-template='{{ .data.AWS_ACCESS_KEY_ID }}' | base64 -d
  ```

- S3 provider secret access key:
  ```bash
  $ oc get secret -n openshift-storage migstorage -o go-template='{{ .data.AWS_SECRET_ACCESS_KEY }}' | base64 -d
  ```

2.2.3. Configuring an AWS S3 storage bucket as a replication repository

You can configure an AWS S3 storage bucket as a replication repository.

**Prerequisites**

- The AWS S3 storage bucket must be accessible to the source and target clusters.
You must have the AWS CLI installed.

If you are using the snapshot copy method:

- You must have access to EC2 Elastic Block Storage (EBS).
- The source and target clusters must be in the same region.
- The source and target clusters must have the same storage class.
- The storage class must be compatible with snapshots.

Procedure

1. Create an AWS S3 bucket:

```bash
$ aws s3api create-bucket \
  --bucket <bucket_name>  
  --region <bucket_region>
```

1. Specify your S3 bucket name.
2. Specify your S3 bucket region, for example, us-east-1.

2. Create the IAM user velero:

```bash
$ aws iam create-user --user-name velero
```

3. Create an EC2 EBS snapshot policy:

```bash
$ cat > velero-ec2-snapshot-policy.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVolumes",
        "ec2:DescribeSnapshots",
        "ec2:CreateTags",
        "ec2:CreateVolume",
        "ec2:CreateSnapshot",
        "ec2:DeleteSnapshot"
      ],
      "Resource": "*"
    }
  ]
}
EOF
```

4. Create an AWS S3 access policy for one or for all S3 buckets:

```bash
$ cat > velero-s3-policy.json <<EOF
{
```

```bash
EOF
```
To grant access to a single S3 bucket, specify the bucket name. To grant access to all AWS S3 buckets, specify * instead of a bucket name:

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:DeleteObject",
      "s3:PutObject",
      "s3:AbortMultipartUpload",
      "s3:ListMultipartUploadParts"
    ],
    "Resource": [
      "arn:aws:s3::<bucket_name>/**"  
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:ListBucket",
      "s3:GetBucketLocation",
      "s3:ListBucketMultipartUploads"
    ],
    "Resource": [
      "arn:aws:s3::<bucket_name>"
    ]
  }
]
EOF
```

1 2 To grant access to a single S3 bucket, specify the bucket name. To grant access to all AWS S3 buckets, specify * instead of a bucket name:

```
"Resource": [
  "arn:aws:s3:::*"
```

5. Attach the EC2 EBS policy to velero:

```
$ aws iam put-user-policy \
--user-name velero \
--policy-name velero-ebs \
--policy-document file://velero-ec2-snapshot-policy.json
```

6. Attach the AWS S3 policy to velero:

```
$ aws iam put-user-policy \
--user-name velero \
--policy-name velero-s3 \
--policy-document file://velero-s3-policy.json
```

7. Create an access key for velero:

```
$ aws iam create-access-key --user-name velero
```
Record the `AWS_SECRET_ACCESS_KEY` and the `AWS_ACCESS_KEY_ID` for adding the AWS repository to the CAM web console.

2.2.4. Configuring a Google Cloud Provider storage bucket as a replication repository

You can configure a Google Cloud Provider (GCP) storage bucket as a replication repository.

Prerequisites

- The GCP storage bucket must be accessible to the source and target clusters.
- You must have `gsutil` installed.
- If you are using the snapshot copy method:
  - The source and target clusters must be in the same region.
  - The source and target clusters must have the same storage class.
  - The storage class must be compatible with snapshots.

Procedure

1. Run `gsutil init` to log in:

   ```
   $ gsutil init
   Welcome! This command will take you through the configuration of gcloud.
   Your current configuration has been set to: [default]
   To continue, you must login. Would you like to login (Y/n)?
   ```

2. Set the `BUCKET` variable:

   ```
   $ BUCKET=<bucket_name>  
   Specify your bucket name.
   ```

3. Create a storage bucket:

   ```
   $ gsutil mb gs://$BUCKET/
   ```
4. Set the **PROJECT_ID** variable to your active project:

   ```
   $ PROJECT_ID=$(gcloud config get-value project)
   ```

5. Create a **velero** service account:

   ```
   $ gcloud iam service-accounts create velero \\n   --display-name "Velero Storage"
   ```

6. Set the **SERVICE_ACCOUNT_EMAIL** variable to the service account’s email address:

   ```
   $ SERVICE_ACCOUNT_EMAIL=$(gcloud iam service-accounts list \\n   --filter="displayName:Velero Storage" \\n   --format 'value(email)')
   ```

7. Grant permissions to the service account:

   ```
   $ ROLE_PERMISSIONS=(
   compute.disks.get 
   compute.disks.create 
   compute.disks.createSnapshot 
   compute.snapshots.get 
   compute.snapshots.create 
   compute.snapshots.useReadOnly 
   compute.snapshots.delete 
   compute.zones.get 
   )

   gcloud iam roles create velero.server \\n   --project $PROJECT_ID \\n   --title "Velero Server" \\n   --permissions "$(IFS=","; echo "${ROLE_PERMISSIONS[*]}")"

   gcloud projects add-iam-policy-binding $PROJECT_ID \\n   --member serviceAccount:$SERVICE_ACCOUNT_EMAIL \\n   --role projects/$PROJECT_ID/roles/velero.server \\n
   gsutil iam ch serviceAccount:$SERVICE_ACCOUNT_EMAIL:objectAdmin gs://$[BUCKET]
   ```

8. Save the service account’s keys to the **credentials-velero** file in the current directory:

   ```
   $ gcloud iam service-accounts keys create credentials-velero \\n   --iam-account $SERVICE_ACCOUNT_EMAIL
   ```

**2.2.5. Configuring a Microsoft Azure Blob storage container as a replication repository**

You can configure a Microsoft Azure Blob storage container as a replication repository.

**Prerequisites**

- You must have an Azure storage account.
You must have the Azure CLI installed.

The Azure Blob storage container must be accessible to the source and target clusters.

If you are using the snapshot copy method:
- The source and target clusters must be in the same region.
- The source and target clusters must have the same storage class.
- The storage class must be compatible with snapshots.

Procedure

1. Set the AZURE_RESOURCE_GROUP variable:
   
   ```
   $ AZURE_RESOURCE_GROUP=Velero_Backups
   ```

2. Create an Azure resource group:
   
   ```
   $ az group create -n $AZURE_RESOURCE_GROUP --location <CentralUS>  
   ```
   
   Specify your location.

3. Set the AZURE_STORAGE_ACCOUNT_ID variable:
   
   ```
   $ AZURE_STORAGE_ACCOUNT_ID=velerobackups
   ```

4. Create an Azure storage account:
   
   ```
   $ az storage account create 
   --name $AZURE_STORAGE_ACCOUNT_ID 
   --resource-group $AZURE_RESOURCE_GROUP 
   --sku Standard_GRS 
   --encryption-services blob 
   --https-only true 
   --kind BlobStorage 
   --access-tier Hot
   ```

5. Set the BLOB_CONTAINER variable:
   
   ```
   $ BLOB_CONTAINER=velero
   ```

6. Create an Azure Blob storage container:
   
   ```
   $ az storage container create 
   -n $BLOB_CONTAINER 
   --public-access off 
   --account-name $AZURE_STORAGE_ACCOUNT_ID
   ```

7. Create a service principal and credentials for velero:
   
   ```
   $ AZURE_SUBSCRIPTION_ID=`az account list --query '[?isDefault].id' -o tsv`
   ```
$ AZURE_TENANT_ID=`az account list --query '[?isDefault].tenantId' -o tsv`
$ AZURE_CLIENT_SECRET=`az ad sp create-for-rbac --name "velero" --role "Contributor" --
query 'password' -o tsv`
$ AZURE_CLIENT_ID=`az ad sp list --display-name "velero" --query '[0].appId' -o tsv`

8. Save the service principal’s credentials in the **credentials-velero** file:

```bash
$ cat << EOF  > ./credentials-velero
AZURE_SUBSCRIPTION_ID=${AZURE_SUBSCRIPTION_ID}
AZURE_TENANT_ID=${AZURE_TENANT_ID}
AZURE_CLIENT_ID=${AZURE_CLIENT_ID}
AZURE_CLIENT_SECRET=${AZURE_CLIENT_SECRET}
AZURE_RESOURCE_GROUP=${AZURE_RESOURCE_GROUP}
AZURE_CLOUD_NAME=AzurePublicCloud
EOF
```

### 2.3. DEPLOYING THE CLUSTER APPLICATION MIGRATION (CAM) TOOL

Deploying the Cluster Application Migration (CAM) tool requires installing the Cluster Application Migration Operator on the OpenShift Container Platform 4.1 source and OpenShift Container Platform 4.3 target clusters.

#### 2.3.1. Installing the Cluster Application Migration Operator on an OpenShift Container Platform 4.1 source cluster

You can install the Cluster Application Migration Operator on an OpenShift Container Platform 4.1 source cluster with OLM.

**Procedure**

1. In the OpenShift Container Platform web console, click Administration → Namespaces.

2. Click **Create Namespace**.

3. Enter **openshift-migration** in the Name field and click **Create**.

4. Click Catalog → OperatorHub.

5. Use the Filter by keyword field (in this case, Migration) to find the Cluster Application Migration Operator.

6. Select the **Cluster Application Migration Operator** and click Install.

7. On the Create Operator Subscription page, select the openshift-migration namespace, and specify an approval strategy.

8. Click **Subscribe**.

   On the Installed Operators page, the Cluster Application Migration Operator appears in the openshift-migration project with the status **InstallSucceeded**.

9. Under Provided APIs, click **View 12 more...**

10. Click **Create New → MigrationController**.
11. Update the `migration_controller` and `migration_ui` parameters and add the `deprecated_cors_configuration` parameter to the `spec` stanza:

```yaml
spec:
  [...]  
migration_controller: false
migration_ui: false
  [...]  
deprecated_cors_configuration: true
```

12. Click Create.

13. Click Workloads → Pods to verify that the Restic and Velero Pods are running.

### 2.3.2. Installing the Cluster Application Migration Operator on an OpenShift Container Platform 4.3 target cluster

You can install the Cluster Application Migration Operator on an OpenShift Container Platform 4.3 target cluster with OLM.

The Cluster Application Migration Operator installs the CAM tool on the target cluster by default. If you want to install the CAM tool on a different cluster, you must update the Migration controller configuration so that the Cluster Application Migration Operator does not install the CAM tool on this cluster.

**Procedure**

1. In the OpenShift Container Platform web console, click Administration → Namespaces.

2. Click Create Namespace.

3. Enter `openshift-migration` in the Name field and click Create.

4. Click Operators → OperatorHub.

5. Use the Filter by keyword field (in this case, Migration) to find the Cluster Application Migration Operator.

6. Select the Cluster Application Migration Operator and click Install.

7. On the Create Operator Subscription page, select the `openshift-migration` namespace, and specify an approval strategy.

8. Click Subscribe.

   On the Installed Operators page, the Cluster Application Migration Operator appears in the `openshift-migration` project with the status InstallSucceeded.

9. Under Provided APIs, click View 12 more…

10. Click Create New → MigrationController.

11. If you do not want to install the CAM tool on the target cluster, update the `migration_controller` and `migration_ui` parameters in the `spec` stanza:

```yaml
spec:
  [...]  
```
12. Click **Create**.

13. Click **Workloads → Pods** to verify that the Controller Manager, Migration UI, Restic, and Velero Pods are running.

### 2.4. MIGRATING APPLICATIONS WITH THE CAM WEB CONSOLE

#### 2.4.1. Launching the CAM web console

You can launch the CAM web console that is installed on the target cluster.

**Procedure**

1. Log in to the target cluster.

2. Obtain the CAM web console URL:

```bash
$ oc get -n openshift-migration route/migration -o go-template='(?i)//{{ .spec.host }}(:|\z){"printIn"}'' | sed 's,.,\.,g'
```

3. Launch a browser and navigate to the CAM web console.

**NOTE**
If you try to access the CAM web console immediately after installing the Cluster Application Migration Operator, the console may not load because the Operator is still configuring the cluster. Wait a few minutes and retry.

4. If you are using self-signed CA certificates, you will be prompted to accept the CA certificate of the source cluster’s API server. The web page will guide you through the process of accepting the remaining certificates.

5. Log in with your OpenShift Container Platform **username** and **password**.

#### 2.4.2. Adding a cluster to the CAM web console

You can add a source cluster to the CAM web console.

**Prerequisites for Azure**

If you are using Azure snapshots to copy data: * You must provide the Azure resource group name when you add the source cluster. * The source and target clusters must be in the same Azure resource group and in the same location.

**Procedure**

1. Log in to the source cluster.

2. Obtain the service account token:

```bash

```
3. Log in to the CAM web console.

4. In the Clusters section, click Add cluster.

5. Fill in the following fields:
   - **Cluster name**: May contain lower-case letters (a-z) and numbers (0-9). Must not contain spaces or international characters.
   - **Url**: URL of the cluster’s API server, for example, `https://<master1.example.com>:8443`.
   - **Service account token**: String that you obtained from the source cluster.
   - **Azure cluster**: Optional. Select it if you are using Azure snapshots to copy your data.
   - **Azure resource group**: This field appears if Azure cluster is checked.

6. Click Add cluster.
The cluster appears in the Clusters section.

### 2.4.3. Adding a replication repository to the CAM web console

You can add an object storage bucket as a replication repository to the CAM web console.

#### Prerequisites

- You must configure an object storage bucket for migrating the data.

#### Procedure

1. Log in to the CAM web console.

2. In the Replication repositories section, click Add repository.

3. Select a Storage provider type and fill in the following fields:
   - **AWS** for AWS S3, MCG, and generic S3 providers:
     - **Replication repository name**: Specify the replication repository name in the CAM web console.
○ **S3 bucket name:** Specify the name of the S3 bucket you created.

○ **S3 bucket region:** Specify the S3 bucket region. **Required** for AWS S3. **Optional** for other S3 providers.

○ **S3 endpoint:** Specify the URL of the S3 service, not the bucket, for example, `https://<s3-storage.apps.cluster.com>`. **Required** for a generic S3 provider. You must use the `https://` prefix.

○ **S3 provider access key:** Specify the `<AWS_SECRET_ACCESS_KEY>` for AWS or the S3 provider access key for MCG.

○ **S3 provider secret access key:** Specify the `<AWS_ACCESS_KEY_ID>` for AWS or the S3 provider secret access key for MCG.

○ **Require SSL verification:** Clear this check box if you are using a generic S3 provider.

● **GCP:**

○ **Replication repository name:** Specify the replication repository name in the CAM web console.

○ **GCP bucket name:** Specify the name of the GCP bucket.

○ **GCP credential JSON blob:** Specify the string in the `credentials-velero` file.

● **Azure:**

○ **Replication repository name:** Specify the replication repository name in the CAM web console.

○ **Azure resource group:** Specify the resource group of the Azure Blob storage.

○ **Azure storage account name:** Specify the Azure Blob storage account name.

○ **Azure credentials - INI file contents:** Specify the string in the `credentials-velero` file.

4. Click **Add repository** and wait for connection validation.

5. Click **Close**.
   The new repository appears in the **Replication repositories** section.

### 2.4.4. Changing migration plan limits for large migrations

You can change the migration plan limits for large migrations.

**IMPORTANT**

Changes should first be tested in your environment to avoid a failed migration.

A single migration plan has the following default limits:

● **10 namespaces**
  If this limit is exceeded, the CAM web console displays a **Namespace limit exceeded** error and you cannot create a migration plan.
- 100 Pods
  If the Pod limit is exceeded, the CAM web console displays a warning message similar to the following example: **Plan has been validated with warning condition(s). See warning message. Pod limit: 100 exceeded, found: 104.**

- 100 persistent volumes
  If the persistent volume limit is exceeded, the CAM web console displays a similar warning message.

**Procedure**

1. Edit the Migration controller CR:

   ```bash
   $ oc get migrationcontroller -n openshift-migration
   NAME AGE
   migration-controller 5d19h
   $ oc edit migrationcontroller -n openshift-migration
   ```

2. Update the following parameters:

   ```yaml
   migration_controller: true

   # This configuration is loaded into mig-controller, and should be set on the # cluster where 'migration_controller: true'
   mig_pv_limit: 100
   mig_pod_limit: 100
   mig_namespace_limit: 10
   ```

**2.4.5. Creating a migration plan in the CAM web console**

You can create a migration plan in the CAM web console.

**Prerequisites**

- The CAM web console must contain the following:
  - Source cluster
  - Target cluster, which is added automatically during the CAM tool installation
  - Replication repository

- If you want to copy your data by using snapshots, the source and target clusters must be running on the same cloud provider (AWS, GCP, or Azure) and in the same region.

**Procedure**

1. Log in to the CAM web console.

2. In the **Plans** section, click **Add plan.**

3. Enter the **Plan name** and click **Next.**
The Plan name can contain up to 253 lower-case alphanumeric characters (a-z, 0-9). It must not contain spaces or underscores (_).

4. Select a Source cluster.

5. Select a Target cluster.

6. Select a Replication repository.

7. Select the projects to be migrated and click Next.

8. Select Copy or Move for the PVs:
   - Copy copies the data in a source cluster’s PV to the replication repository and then restores it on a newly created PV, with similar characteristics, in the target cluster.
   - Move unmounts a remote volume (for example, NFS) from the source cluster, creates a PV resource on the target cluster pointing to the remote volume, and then mounts the remote volume on the target cluster. Applications running on the target cluster use the same remote volume that the source cluster was using. The remote volume must be accessible to the source and target clusters.

9. Click Next.

10. Select a Copy method for the PVs:
    - Snapshot backs up and restores the disk using the cloud provider’s snapshot functionality. It is significantly faster than Filesystem.

    NOTE
    The storage and clusters must be in the same region and the storage class must be compatible.

    - Filesystem copies the data files from the source disk to a newly created target disk.

11. Select a Storage class for the PVs.
    If you selected the Filesystem copy method, you can change the storage class during migration, for example, from Red Hat Gluster Storage or NFS storage to Red Hat Ceph Storage.

12. Click Finish.

13. Click Close.
The migration plan appears in the Plans section.

2.4.6. Running a migration plan in the CAM web console

You can stage or migrate applications and data with the migration plan you created in the CAM web console.

Prerequisites
The CAM web console must contain the following:

- Source cluster
- Target cluster, which is added automatically during the CAM tool installation
2.5. TROUBLESHOOTING

You can view the migration Custom Resources (CRs) and download logs to troubleshoot a failed migration.

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

**NOTE**

Manual rollback is not required if the application was not stopped during migration, because the original application is still running on the source cluster.

### 2.5.1. Viewing migration Custom Resources (CRs)

The Cluster Application Migration (CAM) tool creates the following CRs for migration:

- Replication repository
- Valid migration plan

**Procedure**

1. Log in to the CAM web console on the target cluster.

2. Select a migration plan.

3. Click **Stage** to copy data from the source cluster to the target cluster without stopping the application.
   
   You can run **Stage** multiple times to reduce the actual migration time.

4. When you are ready to migrate the application workload, click **Migrate**.
   
   **Migrate** stops the application workload on the source cluster and recreates its resources on the target cluster.

5. Optionally, in the **Migrate** window, you can select **Do not stop applications on the source cluster during migration**.

6. Click **Migrate**.

7. When the migration is complete, verify that the application migrated successfully in the OpenShift Container Platform web console:
   
   a. Click **Home** → **Projects**.
   
   b. Click the migrated project to view its status.
   
   c. In the **Routes** section, click **Location** to verify that the application is functioning, if applicable.
   
   d. Click **Workloads** → **Pods** to verify that the Pods are running in the migrated namespace.
   
   e. Click **Storage** → **Persistent volumes** to verify that the migrated persistent volume is correctly provisioned.
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.

**NOTE**
Deleting a MigPlan CR deletes the associated MigMigration CRs.

1. **MigCluster** (configuration, CAM cluster): Cluster definition
2. **MigStorage** (configuration, CAM cluster): Storage definition
3. **MigPlan** (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.

4. **BackupStorageLocation** (configuration, CAM cluster): Location of Velero backup objects
5. **VolumeSnapshotLocation** (configuration, CAM cluster): Location of Velero volume snapshots
6. **MigMigration** (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.
Backup (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

Restore (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

Procedure

1. Obtain the CR name:

   ```bash
   $ oc get <cr> -n openshift-migration
   NAME       AGE
   88435fe0-c9f8-11e9-85e6-5d593ce65e10 6m42s
   ```

   Specify the migration CR you want to view.

2. View the CR:

   ```bash
   $ oc describe <cr> 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
   ```

   The output is similar to the following examples.

MigMigration example

```bash
$ oc describe migmigration 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
Name: 88435fe0-c9f8-11e9-85e6-5d593ce65e10
Namespace: openshift-migration
Labels: <none>
Annotations: touch: 3b48b543-b53e-4e44-9d34-33563f0f8147
API Version: migration.openshift.io/v1alpha1
Kind: MigMigration
Metadata:
Creation Timestamp: 2019-08-29T01:01:29Z
Generation: 20
Resource Version: 88179
Self Link: /apis/migration.openshift.io/v1alpha1/namespaces/openshift-migration/migmigrations/88435fe0-c9f8-11e9-85e6-5d593ce65e10
UID: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
Spec:
Mig Plan Ref:
  Name: socks-shop-mig-plan
  Namespace: openshift-migration
Quiesce Pods: true
Stage: false
Status:
```
Velero backup CR #2 example (PV data)

```yaml
apiVersion: velero.io/v1
category:            
kind:                Backup
metadata:            
annotations:         
  openshift.io/migrate-copy-phase: final   
  openshift.io/migrate-quiesce-pods: "true"  
  openshift.io/migration-registry: 172.30.105.179:5000   
  openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-44dd3bd5-c9f8-11e9-95ad-0205fe66cbb6   
creationTimestamp:  "2019-08-29T01:03:15Z"   
generateName:  88435fe0-c9f8-11e9-85e6-5d593ce65e10-   
generation:  1   
labels:              
  app.kubernetes.io/part-of: migration   
migmigration:  8886de4c-c9f8-11e9-95ad-0205fe66cbb6   
migration-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6   
velero.io/storage-location: myrepo-vpzq9   
name:  88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7   
namespace:  openshift-migration   
resourceVersion: "87313"   
selfLink: /apis/velero.io/v1/namespaces/openshift-migration/backups/88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7   
uid:  c80dbbc0-c9f8-11e9-95ad-0205fe66cbb6   
spec:              
excludedNamespaces: []   
excludedResources: []   
hooks:              
  resources: []   
includeClusterResources: null   
includedNamespaces:   
  - sock-shop   
includedResources:   
  - persistentvolumes   
  - persistentvolumeclaims   
  - namespaces   
  - imagestreams   
  - imagestreamtags   
  - secrets   
  - configmaps   
  - pods   
labelSelector:   
  matchLabels:   
```
migration-included-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
storageLocation: myrepo-vpzq9
ttl: 720h0m0s
volumeSnapshotLocations:
  - myrepo-wv6fx
status:
  completionTimestamp: "2019-08-29T01:02:36Z"
  errors: 0
  expiration: "2019-09-28T01:02:35Z"
  phase: Completed
  startTimestamp: "2019-08-29T01:02:35Z"
  validationErrors: null
  version: 1
  volumeSnapshotsAttempted: 0
  volumeSnapshotsCompleted: 0
  warnings: 0

Velero restore CR #2 example (Kubernetes resources)

apiVersion: velero.io/v1
kind: Restore
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.90.187:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-36f54ca7-c925-11e9-825a-06fa9fb68c88
  creationTimestamp: "2019-08-28T00:09:49Z"
  generateName: e13a1b60-c927-11e9-9555-d129df7f3b96-
generation: 3
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: e18252c9-c927-11e9-825a-06fa9fb68c88
    migration-final-restore: e18252c9-c927-11e9-825a-06fa9fb68c88
    name: e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
    namespace: openshift-migration
resourceVersion: "82329"
  selfLink: /apis/velero.io/v1/namespaces/openshift-migration/restores/e13a1b60-c927-11e9-9555-
d129df7f3b96-gb8nx
  uid: 26983ec0-c928-11e9-825a-06fa9fb68c88
spec:
  backupName: e13a1b60-c927-11e9-9555-d129df7f3b96-sz24f
  excludedNamespaces: null
  excludedResources:
    - nodes
    - events
    - events.events.k8s.io
    - backups.velero.io
    - restores.velero.io
    - resticrepositories.velero.io
  includedNamespaces: null
  includedResources: null
  namespaceMapping: null
  restorePVs: true
  status:
2.5.2. 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 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:
   a. 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.
   b. Click Download Selected to download the selected log.

Optionally, you can 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
```

2.5.3. Restic timeout error

If a migration fails because Restic times out, the following error appears in the Velero log:

```
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. You can increase this for large migrations, keeping in mind that a higher value may delay the return of error messages.

Procedure

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:

   ```yaml
   spec:
   restic_timeout: 1h
   ```

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

5. Click **Save**.

### 2.5.4. Manually rolling back a migration

If your application was stopped during a failed migration, you must roll it back manually in order to prevent data corruption in the PV.

This procedure is not required if the application was not stopped during migration, because the original application is still running on the source cluster.

Procedure

1. On the target cluster, switch to the migrated project:

   ```bash
   $ oc project <project>
   ```

2. Get the deployed resources:

   ```bash
   $ oc get all
   ```

3. Delete the deployed resources to ensure that the application is not running on the target cluster and accessing data on the PVC:

   ```bash
   $ oc delete <resource_type>
   ```

4. To stop a DaemonSet without deleting it, update the **nodeName** in the YAML file:

   ```yaml
   apiVersion: extensions/v1beta1
   kind: DaemonSet
   metadata:
   name: hello-daemonset
   spec:
   selector:
   matchLabels:
   ```
Specify a `nodeSelector` value that does not exist on any node.

5. 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 the 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 the source cluster, switch to the migrated project and get its deployed resources:

   ```bash
   $ oc project <project>
   $ oc get all
   ```

7. Start one or more replicas of each deployed resource:

   ```bash
   $ oc scale --replicas=1 <resource_type>/<resource_name>
   ```

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

### 2.5.5. Known issues

This release has the following known issues:

- During migration, the Cluster Application Migration (CAM) tool preserves the following namespace annotations:
  - `openshift.io/sa.scc.mcs`
  - `openshift.io/sa.scc.supplemental-groups`
- **openshift.io/sa.scc.uid-range**
  These annotations preserve the UID range, ensuring that the containers retain their file system permissions on the target cluster. There is a risk that the migrated UIDs could duplicate UIDs within an existing or future namespace on the target cluster. (BZ#1748440)

- If an AWS bucket is added to the CAM web console and then deleted, its status remains **True** because the MigStorage CR is not updated. (BZ#1738564)

- Migration fails if the Migration controller is running on a cluster other than the target cluster. The **EnsureCloudSecretPropagated** phase is skipped with a logged warning. (BZ#1757571)

- Most cluster-scoped resources are not yet handled by the CAM tool. If your applications require cluster-scoped resources, you may have to create them manually on the target cluster.

- Incorrect source cluster storage class is displayed when creating the migration plan. (BZ#1777869)

- If a cluster in the CAM web console becomes inaccessible, it blocks attempts to close open migration plans. (BZ#1758269)

- If a migration fails, the migration plan does not retain custom PV settings for quiesced pods. You must manually roll back the migration, delete the migration plan, and create a new migration plan with your PV settings. (BZ#1784899)
CHAPTER 3. MIGRATING OPENSSHIFT CONTAINER PLATFORM 4.2 TO 4.3

3.1. MIGRATING APPLICATION WORKLOADS FROM OPENSSHIFT CONTAINER PLATFORM 4.2 TO 4.3

You can migrate application workloads from OpenShift Container Platform 4.2 to 4.3 with the Cluster Application Migration (CAM) tool. The CAM tool enables you to control the migration and to minimize application downtime.

The CAM tool’s web console and API, based on Kubernetes Custom Resources, enable you to migrate stateful and stateless application workloads at the granularity of a namespace.

3.1.1. Migration prerequisites

- You must have `cluster-admin` privileges on all clusters.
- You must configure a replication repository that is accessible to the source and target clusters.

3.1.2. Understanding the Cluster Application Migration tool

The Cluster Application Migration (CAM) tool enables you to migrate Kubernetes resources, persistent volume data, and internal container images from an OpenShift Container Platform source cluster to an OpenShift Container Platform 4.3 target cluster, using the CAM web console or the Kubernetes API.

Migrating an application with the CAM web console involves the following steps:

1. Install the Cluster Application Migration Operator on all clusters.

   **NOTE**

   The Cluster Application Migration Operator installs the CAM tool (CAM web console and Migration controller) on the target cluster by default. You can configure the Migration controller to install the CAM tool on another cluster. The cluster on which the CAM tool is installed is referred to as the CAM tool cluster.

2. Configure the replication repository, an intermediate object storage that the CAM tool uses to migrate data.

3. Add the source cluster to the CAM web console.

4. Add the replication repository to the CAM web console.

5. Create a migration plan, with one of the following data migration options:
   - **Copy**: The CAM tool copies the data from the source cluster to the replication repository, and from the replication repository to the target cluster.
• **Move**: The CAM tool unmounts a remote volume (for example, NFS) from the source cluster, creates a PV resource on the target cluster pointing to the remote volume, and then mounts the remote volume on the target cluster. Applications running on the target cluster use the same remote volume that the source cluster was using. The remote volume must be accessible to the source and target clusters.

**NOTE**
Although the replication repository does not appear in this diagram, it is required for the actual migration.

6. Run the migration plan, with one of the following options:

• **Stage** (optional) copies data to the target cluster without stopping the application. Staging can be run multiple times so that most of the data is copied to the target before migration. This minimizes the actual migration time and application downtime.

• **Migrate** stops the application on the source cluster and recreates its resources on the target cluster. Optionally, you can migrate the workload without stopping the application.
3.2. CONFIGURING A REPLICATION REPOSITORY

You must configure an object storage to use as a replication repository. The Cluster Application Migration tool copies data from the source cluster to the replication repository, and then from the replication repository to the target cluster, using either the file system or the snapshot data copy method.

The following storage providers are supported:

- Generic S3 object storage, for example, Minio or Ceph S3
- Multi-Cloud Object Gateway (MCG)
- Amazon Web Services (AWS) S3
- Google Cloud Provider (GCP)
- Microsoft Azure

3.2.1. Understanding the data copy methods for migration

The CAM tool supports the file system and snapshot data copy methods for migrating data from the source cluster to the target cluster. You can select a method that is suited for your environment and is supported by your storage provider.

3.2.1.1. File system copy method

The CAM tool copies data files from the source cluster to the replication repository, and from there to the target cluster.
Table 3.1. File system copy method summary

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Clusters can have different storage classes</td>
<td>• Slower than the snapshot copy method</td>
</tr>
<tr>
<td>• Supported for all S3 storage providers</td>
<td></td>
</tr>
</tbody>
</table>

3.2.1.2. Snapshot copy method

The CAM tool copies a snapshot of the source cluster’s data to a cloud provider’s object storage, configured as a replication repository. The data is restored on the target cluster.

AWS, Google Cloud Provider, and Microsoft Azure support the snapshot copy method.

Table 3.2. Snapshot copy method summary

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Faster than the file system copy method</td>
<td>• Cloud provider must support snapshots.</td>
</tr>
<tr>
<td></td>
<td>• Clusters must be on the same cloud provider.</td>
</tr>
<tr>
<td></td>
<td>• Clusters must be in the same location or region.</td>
</tr>
<tr>
<td></td>
<td>• Clusters must have the same storage class.</td>
</tr>
<tr>
<td></td>
<td>• Storage class must be compatible with snapshots.</td>
</tr>
</tbody>
</table>

3.2.2. Configuring a Multi-Cloud Object Gateway storage bucket as a replication repository

You can install the OpenShift Container Storage Operator and configure a Multi-Cloud Object Gateway (MCG) storage bucket as a replication repository on the CAM tool cluster.

3.2.2.1. Installing the OpenShift Container Storage Operator

You can install the OpenShift Container Storage Operator from OperatorHub on the CAM tool cluster.

Procedure

1. In the OpenShift Container Platform web console, click Administration → Namespaces.
2. Click Create Namespace.
3. Enter openshift-storage in the Name field and click Create.
4. Click Operators → OperatorHub.
5. Use Filter by keyword (in this case, OCS) to find the OpenShift Container Storage Operator.

6. Select the OpenShift Container Storage Operator and click Install.

7. On the Create Operator Subscription page, select the openshift-storage namespace.

8. Specify your update channel and approval strategy.

9. Click Subscribe.

   On the Installed Operators page, the OpenShift Container Storage Operator appears in the openshift-storage project with the status Succeeded.

3.2.2.2. Creating the Multi-Cloud Object Gateway storage bucket

You can create the Multi-Cloud Object Gateway (MCG) storage bucket’s Custom Resources (CRs) on the CAM tool cluster.

**Procedure**

1. Log in to the CAM tool cluster.

2. Create the Noobaa CR configuration file, noobaa.yml, with the following content:

   ```yaml
   apiVersion: noobaa.io/v1alpha1
   kind: NooBaa
   metadata:
     name: noobaa
     namespace: openshift-storage
   spec:
     dbResources:
       requests:     
         cpu: 0.5
         memory: 1Gi
     coreResources:
       requests:     
         cpu: 0.5
         memory: 1Gi
   ```

   1 2 For a very small cluster, you can change the cpu value to 0.1.

3. Create the Noobaa object:

   ```bash
   $ oc create -f noobaa.yml
   ```

4. Create the BackingStore CR configuration file, bs.yml, with the following content:

   ```yaml
   apiVersion: noobaa.io/v1alpha1
   kind: BackingStore
   metadata:
     finalizers:
     - noobaa.io/finalizer
   labels:
     app: noobaa
   name: mcg-pv-pool-bs
   ```
Specify the number of volumes in the PV pool.

Specify the size of the volumes.

Specify the storage class.

5. Create the **BackingStore** object:

```
$ oc create -f bs.yml
```

6. Create the **BucketClass** CR configuration file, `bc.yml`, with the following content:

```
apiVersion: noobaa.io/v1alpha1
kind: BucketClass
metadata:
  labels:
    app: noobaa
  name: mcg-pv-pool-bc
namespace: openshift-storage
spec:
  placementPolicy:
    tiers:
    - backingStores:
      - mcg-pv-pool-bs
    placement: Spread
```

7. Create the **BucketClass** object:

```
$ oc create -f bc.yml
```

8. Create the **ObjectBucketClaim** CR configuration file, `obc.yml`, with the following content:

```
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: migstorage
  namespace: openshift-storage
spec:
  bucketName: migstorage
  storageClassName: openshift-storage.noobaa.io
  additionalConfig:
    bucketclass: mcg-pv-pool-bs
```
1. Record the bucket name for adding the replication repository to the CAM web console.

9. Create the **ObjectBucketClaim** object:

   $ oc create -f obc.yml

10. Watch the resource creation process to verify that the **ObjectBucketClaim** status is **Bound**:

    $ watch -n 30 'oc get -n openshift-storage objectbucketclaim migstorage -o yaml'

    This process can take five to ten minutes.

11. Obtain and record the following values, which are required when you add the replication repository to the CAM web console:

    - **S3 endpoint:**
      
      $ oc get route -n openshift-storage s3

    - **S3 provider access key:**
      
      $ oc get secret -n openshift-storage migstorage -o go-template='{{ .data.AWS_ACCESS_KEY_ID }}' | base64 -d

    - **S3 provider secret access key:**
      
      $ oc get secret -n openshift-storage migstorage -o go-template='{{ .data.AWS_SECRET_ACCESS_KEY }}' | base64 -d

3.2.3. Configuring an AWS S3 storage bucket as a replication repository

You can configure an AWS S3 storage bucket as a replication repository.

**Prerequisites**

- The AWS S3 storage bucket must be accessible to the source and target clusters.
- You must have the **AWS CLI** installed.
- If you are using the snapshot copy method:
  - You must have access to EC2 Elastic Block Storage (EBS).
  - The source and target clusters must be in the same region.
  - The source and target clusters must have the same storage class.
  - The storage class must be compatible with snapshots.

**Procedure**

1. Create an AWS S3 bucket:
Specify your S3 bucket name.

Specify your S3 bucket region, for example, **us-east-1**.

2. Create the IAM user **velero**:

```
$ aws iam create-user --user-name velero
```

3. Create an EC2 EBS snapshot policy:

```
$ cat > velero-ec2-snapshot-policy.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeVolumes",
      "ec2:DescribeSnapshots",
      "ec2:CreateTags",
      "ec2:CreateVolume",
      "ec2:CreateSnapshot",
      "ec2:DeleteSnapshot"
    ],
    "Resource": "*"
  }
  ]
}
EOF
```

4. Create an AWS S3 access policy for one or for all S3 buckets:

```
$ cat > velero-s3-policy.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:DeleteObject",
      "s3:PutObject",
      "s3:AbortMultipartUpload",
      "s3:ListMultipartUploadParts"
    ],
    "Resource": [
      "arn:aws:s3:::<bucket_name>/**"
    ]
  }
  ]
}
EOF
```
To grant access to a single S3 bucket, specify the bucket name. To grant access to all AWS S3 buckets, specify * instead of a bucket name:

```
"Resource": [
    "arn:aws:s3:::*"
]
```

5. Attach the EC2 EBS policy to velero:

```
$ aws iam put-user-policy \
   --user-name velero \
   --policy-name velero-ebs \
   --policy-document file://velero-ec2-snapshot-policy.json
```

6. Attach the AWS S3 policy to velero:

```
$ aws iam put-user-policy \
   --user-name velero \
   --policy-name velero-s3 \
   --policy-document file://velero-s3-policy.json
```

7. Create an access key for velero:

```
$ aws iam create-access-key --user-name velero
{
   "AccessKey": {
      "UserName": "velero",
      "Status": "Active",
      "CreateDate": "2017-07-31T22:24:41.576Z",
      "SecretAccessKey": <AWS_SECRET_ACCESS_KEY>, 1
      "AccessKeyId": <AWS_ACCESS_KEY_ID> 2
   }
}
```

1.2 Record the AWS_SECRET_ACCESS_KEY and the AWS_ACCESS_KEY_ID for adding the AWS repository to the CAM web console.
3.2.4. Configuring a Google Cloud Provider storage bucket as a replication repository

You can configure a Google Cloud Provider (GCP) storage bucket as a replication repository.

**Prerequisites**

- The GCP storage bucket must be accessible to the source and target clusters.
- You must have `gsutil` installed.
- If you are using the snapshot copy method:
  - The source and target clusters must be in the same region.
  - The source and target clusters must have the same storage class.
  - The storage class must be compatible with snapshots.

**Procedure**

1. Run `gsutil init` to log in:

   ```
   $ gsutil init
   Welcome! This command will take you through the configuration of gcloud.
   Your current configuration has been set to: [default]
   To continue, you must login. Would you like to login (Y/n)?
   ```

2. Set the `BUCKET` variable:

   ```
   $ BUCKET=<bucket_name>  
   Specify your bucket name.
   ```

3. Create a storage bucket:

   ```
   $ gsutil mb gs://$BUCKET/
   ```

4. Set the `PROJECT_ID` variable to your active project:

   ```
   $ PROJECT_ID=$(gcloud config get-value project)
   ```

5. Create a `velero` service account:

   ```
   $ gcloud iam service-accounts create velero 
   --display-name "Velero Storage"
   ```

6. Set the `SERVICE_ACCOUNT_EMAIL` variable to the service account’s email address:


```bash
$ SERVICE_ACCOUNT_EMAIL=$(gcloud iam service-accounts list
    --filter="displayName:Velero Storage" \
    --format 'value(email)')

7. Grant permissions to the service account:

```bash
$ ROLE_PERMISSIONS=(
    compute.disks.get
    compute.disks.create
    compute.disks.createSnapshot
    compute.snapshots.get
    compute.snapshots.create
    compute.snapshots.useReadOnly
    compute.snapshots.delete
    compute.zones.get
)

gcloud iam roles create velero.server \
    --project $PROJECT_ID \
    --title "Velero Server" \
    --permissions "$(IFS=","; echo "${ROLE_PERMISSIONS[*]}")"

gcloud projects add-iam-policy-binding $PROJECT_ID \
    --member serviceAccount:$SERVICE_ACCOUNT_EMAIL \
    --role projects/$PROJECT_ID/roles/velero.server

gsutil iam ch serviceAccount:$SERVICE_ACCOUNT_EMAIL:objectAdmin gs://${BUCKET}
```

8. Save the service account’s keys to the credentials-velero file in the current directory:

```bash
$ gcloud iam service-accounts keys create credentials-velero \
    --iam-account $SERVICE_ACCOUNT_EMAIL
```

3.2.5. Configuring a Microsoft Azure Blob storage container as a replication repository

You can configure a Microsoft Azure Blob storage container as a replication repository.

**Prerequisites**

- You must have an Azure storage account.
- You must have the Azure CLI installed.
- The Azure Blob storage container must be accessible to the source and target clusters.
- If you are using the snapshot copy method:
  - The source and target clusters must be in the same region.
  - The source and target clusters must have the same storage class.
  - The storage class must be compatible with snapshots.
Procedure

1. Set the `AZURE_RESOURCE_GROUP` variable:
   
   ```bash
   $ AZURE_RESOURCE_GROUP=Velero_Backups
   ```

2. Create an Azure resource group:
   
   ```bash
   $ az group create -n $AZURE_RESOURCE_GROUP --location <CentralUS>  
   ```
   
   Specify your location.

3. Set the `AZURE_STORAGE_ACCOUNT_ID` variable:
   
   ```bash
   $ AZURE_STORAGE_ACCOUNT_ID=velerobackups
   ```

4. Create an Azure storage account:
   
   ```bash
   $ az storage account create \
   --name $AZURE_STORAGE_ACCOUNT_ID \
   --resource-group $AZURE_RESOURCE_GROUP \
   --sku Standard_GRS \
   --encryption-services blob \
   --https-only true \
   --kind BlobStorage \
   --access-tier Hot
   ```

5. Set the `BLOB_CONTAINER` variable:
   
   ```bash
   $ BLOB_CONTAINER=velero
   ```

6. Create an Azure Blob storage container:
   
   ```bash
   $ az storage container create \
   -n $BLOB_CONTAINER \
   --public-access off \
   --account-name $AZURE_STORAGE_ACCOUNT_ID
   ```

7. Create a service principal and credentials for `velero`:
   
   ```bash
   $ AZURE_SUBSCRIPTION_ID=`az account list --query '[?isDefault].id' -o tsv`
   $ AZURE_TENANT_ID=`az account list --query '[?isDefault].tenantId' -o tsv`
   $ AZURE_CLIENT_SECRET=`az ad sp create-for-rbac --name "velero" --role "Contributor" --query 'password' -o tsv`
   $ AZURE_CLIENT_ID=`az ad sp list --display-name "velero" --query '[0].appId' -o tsv`
   ```

8. Save the service principal’s credentials in the `credentials-velero` file:
   
   ```bash
   $ cat << EOF  > ./credentials-velero
   AZURE_SUBSCRIPTION_ID=${AZURE_SUBSCRIPTION_ID}
   AZURE_TENANT_ID=${AZURE_TENANT_ID}
   AZURE_CLIENT_ID=${AZURE_CLIENT_ID}
   ```
3.3. DEPLOYING THE CLUSTER APPLICATION MIGRATION (CAM) TOOL

Deploying the Cluster Application Migration (CAM) tool requires installing the Cluster Application Migration Operator on the OpenShift Container Platform 4.2 source and OpenShift Container Platform 4.3 target clusters.

3.3.1. Installing the Cluster Application Migration Operator on an OpenShift Container Platform 4.2 source cluster

You can install the Cluster Application Migration Operator on an OpenShift Container Platform 4.2 source cluster with OLM.

Procedure

1. In the OpenShift Container Platform web console, click Administration → Namespaces.

2. Click Create Namespace.

3. Enter openshift-migration in the Name field and click Create.

4. Click Operators → OperatorHub.

5. Use the Filter by keyword field (in this case, Migration) to find the Cluster Application Migration Operator.

6. Select the Cluster Application Migration Operator and click Install.

7. On the Create Operator Subscription page, select the openshift-migration namespace, and specify an approval strategy.

8. Click Subscribe.
   - On the Installed Operators page, the Cluster Application Migration Operator appears in the openshift-migration project with the status InstallSucceeded.

9. Under Provided APIs, click View 12 more....

10. Click Create New → MigrationController.

11. Update the migration_controller and migration_ui parameters in the spec stanza:

```yaml
spec:
  [...]  
migration_controller: false
migration_ui: false
[...]
```

12. Click Create.
13. Click Workloads → Pods to verify that the Restic and Velero Pods are running.

3.3.2. Installing the Cluster Application Migration Operator on an OpenShift Container Platform 4.3 target cluster

You can install the Cluster Application Migration Operator on an OpenShift Container Platform 4.3 target cluster with OLM.

The Cluster Application Migration Operator installs the CAM tool on the target cluster by default. If you want to install the CAM tool on a different cluster, you must update the Migration controller configuration so that the Cluster Application Migration Operator does not install the CAM tool on this cluster.

Procedure

1. In the OpenShift Container Platform web console, click Administration → Namespaces.

2. Click Create Namespace.

3. Enter openshift-migration in the Name field and click Create.

4. Click Operators → OperatorHub.

5. Use the Filter by keyword field (in this case, Migration) to find the Cluster Application Migration Operator.

6. Select the Cluster Application Migration Operator and click Install.

7. On the Create Operator Subscription page, select the openshift-migration namespace, and specify an approval strategy.

8. Click Subscribe.
   On the Installed Operators page, the Cluster Application Migration Operator appears in the openshift-migration project with the status InstallSucceeded.

9. Under Provided APIs, click View 12 more…

10. Click Create New → MigrationController.

11. If you do not want to install the CAM tool on the target cluster, update the migration_controller and migration_ui parameters in the spec stanza:

```yaml
spec:
  [...]  
migration_controller: false  
migration_ui: false  
[...]
```

12. Click Create.

13. Click Workloads → Pods to verify that the Controller Manager, Migration UI, Restic, and Velero Pods are running.

3.4. MIGRATING APPLICATIONS WITH THE CAM WEB CONSOLE
3.4.1. Launching the CAM web console

You can launch the CAM web console that is installed on the target cluster.

**Procedure**

1. Log in to the target cluster.

2. Obtain the CAM web console URL:

   ```
   $ oc get -n openshift-migration route/migration -o go-template=('(?i)//{{ .spec.host }}(:|\{){{ println }}') | sed 's,.,.g'
   ```

3. Launch a browser and navigate to the CAM web console.

   **NOTE**
   If you try to access the CAM web console immediately after installing the Cluster Application Migration Operator, the console may not load because the Operator is still configuring the cluster. Wait a few minutes and retry.

4. If you are using self-signed CA certificates, you will be prompted to accept the CA certificate of the source cluster’s API server. The web page will guide you through the process of accepting the remaining certificates.

5. Log in with your OpenShift Container Platform **username** and **password**.

3.4.2. Adding a cluster to the CAM web console

You can add a source cluster to the CAM web console.

**Prerequisites for Azure**

If you are using Azure snapshots to copy data: * You must provide the Azure resource group name when you add the source cluster. * The source and target clusters must be in the same Azure resource group and in the same location.

**Procedure**

1. Log in to the source cluster.

2. Obtain the service account token:

   ```
   $ oc sa get-token mig -n openshift-migration
   ```

   eyJhbGciOiJSUzI1NiIsImtpZCI6IiJ9.eyJpc3MiOiJwdWljcm5ldGVzL3Nlc3Npb25zZCI6IjIiLCJhY2NvdW50cyI6eyJpc3MiOiJwdWljcm5ldGVzL3Nlc3Npb25zZCIsImVuYWJsZSI6WyJ1c2VyIiwicm9maWxlIiwicHJvZGllbnQiLCJjaXNjZm9yZCI6IjEwMzY1MDE4MjI5ZjAwYmU0MmUyNjcxMmUxODI0In0.
   ```
3. Log in to the CAM web console.

4. In the Clusters section, click Add cluster.

5. Fill in the following fields:
   - **Cluster name**: May contain lower-case letters (a-z) and numbers (0-9). Must not contain spaces or international characters.
   - **Url**: URL of the cluster’s API server, for example, `https://<master1.example.com>:8443`.
   - **Service account token**: String that you obtained from the source cluster.
   - **Azure cluster**: Optional. Select it if you are using Azure snapshots to copy your data.
   - **Azure resource group**: This field appears if Azure cluster is checked.

6. Click Add cluster.
   The cluster appears in the Clusters section.

### 3.4.3. Adding a replication repository to the CAM web console

You can add an object storage bucket as a replication repository to the CAM web console.

**Prerequisites**
- You must configure an object storage bucket for migrating the data.

**Procedure**

1. Log in to the CAM web console.

2. In the Replication repositories section, click Add repository.

3. Select a Storage provider type and fill in the following fields:
   - **AWS** for AWS S3, MCG, and generic S3 providers:
     - **Replication repository name**: Specify the replication repository name in the CAM web console.
     - **S3 bucket name**: Specify the name of the S3 bucket you created.
     - **S3 bucket region**: Specify the S3 bucket region. Required for AWS S3. Optional for other S3 providers.
     - **S3 endpoint**: Specify the URL of the S3 service, not the bucket, for example, `https://<s3-storage.apps.cluster.com>`. Required for a generic S3 provider. You must use the `https://` prefix.
     - **S3 provider access key**: Specify the `<AWS_SECRET_ACCESS_KEY>` for AWS or the S3 provider access key for MCG.
- **S3 provider secret access key**: Specify the `<AWS_ACCESS_KEY_ID>` for AWS or the S3 provider secret access key for MCG.

- **Require SSL verification**: Clear this check box if you are using a generic S3 provider.

- **GCP**:
  - **Replication repository name**: Specify the replication repository name in the CAM web console.
  - **GCP bucket name**: Specify the name of the GCP bucket.
  - **GCP credential JSON blob**: Specify the string in the `credentials-velero` file.

- **Azure**:
  - **Replication repository name**: Specify the replication repository name in the CAM web console.
  - **Azure resource group**: Specify the resource group of the Azure Blob storage.
  - **Azure storage account name**: Specify the Azure Blob storage account name.
  - **Azure credentials - INI file contents**: Specify the string in the `credentials-velero` file.

4. Click **Add repository** and wait for connection validation.

5. Click **Close**.
   The new repository appears in the **Replication repositories** section.

### 3.4.4. Changing migration plan limits for large migrations

You can change the migration plan limits for large migrations.

**IMPORTANT**

Changes should first be tested in your environment to avoid a failed migration.

A single migration plan has the following default limits:

- **10 namespaces**
  If this limit is exceeded, the CAM web console displays a **Namespace limit exceeded** error and you cannot create a migration plan.

- **100 Pods**
  If the Pod limit is exceeded, the CAM web console displays a warning message similar to the following example: **Plan has been validated with warning condition(s). See warning message. Pod limit: 100 exceeded, found: 104.**

- **100 persistent volumes**
  If the persistent volume limit is exceeded, the CAM web console displays a similar warning message.

**Procedure**

1. Edit the Migration controller CR:
$ oc get migrationcontroller -n openshift-migration
NAME AGE
migration-controller 5d19h

$ oc edit migrationcontroller -n openshift-migration

2. Update the following parameters:

```yaml
[...]
migration_controller: true

# This configuration is loaded into mig-controller, and should be set on the
# cluster where `migration_controller: true`
mig_pv_limit: 100
mig_pod_limit: 100
mig_namespace_limit: 10
[...]
```

### 3.4.5. Creating a migration plan in the CAM web console

You can create a migration plan in the CAM web console.

**Prerequisites**

- The CAM web console must contain the following:
  - Source cluster
  - Target cluster, which is added automatically during the CAM tool installation
  - Replication repository
- If you want to copy your data by using snapshots, the source and target clusters must be running on the same cloud provider (AWS, GCP, or Azure) and in the same region.

**Procedure**

1. Log in to the CAM web console.
2. In the Plans section, click **Add plan**.
3. Enter the **Plan name** and click **Next**.
   - The Plan name can contain up to 253 lower-case alphanumeric characters (a-z, 0-9). It must not contain spaces or underscores (_).
4. Select a **Source cluster**.
5. Select a **Target cluster**.
6. Select a **Replication repository**.
7. Select the projects to be migrated and click **Next**.
8. Select **Copy** or **Move** for the PVs:
   - **Copy** copies the data in a source cluster’s PV to the replication repository and then restores
it on a newly created PV, with similar characteristics, in the target cluster.

- **Move** unmounts a remote volume (for example, NFS) from the source cluster, creates a PV resource on the target cluster pointing to the remote volume, and then mounts the remote volume on the target cluster. Applications running on the target cluster use the same remote volume that the source cluster was using. The remote volume must be accessible to the source and target clusters.

9. Click **Next**.

10. Select a **Copy method** for the PVs:

    - **Snapshot** backs up and restores the disk using the cloud provider’s snapshot functionality. It is significantly faster than **Filesystem**.

    **NOTE**

    The storage and clusters must be in the same region and the storage class must be compatible.

    - **Filesystem** copies the data files from the source disk to a newly created target disk.

11. Select a **Storage class** for the PVs.

    If you selected the **Filesystem** copy method, you can change the storage class during migration, for example, from Red Hat Gluster Storage or NFS storage to Red Hat Ceph Storage.

12. Click **Finish**.

13. Click **Close**.

    The migration plan appears in the **Plans** section.

### 3.4.6. Running a migration plan in the CAM web console

You can stage or migrate applications and data with the migration plan you created in the CAM web console.

**Prerequisites**

The CAM web console must contain the following:

- Source cluster
- Target cluster, which is added automatically during the CAM tool installation
- Replication repository
- Valid migration plan

**Procedure**

1. Log in to the CAM web console on the target cluster.

2. Select a migration plan.

3. Click **Stage** to copy data from the source cluster to the target cluster without stopping the application.
You can run **Stage** multiple times to reduce the actual migration time.

4. When you are ready to migrate the application workload, click **Migrate**. **Migrate** stops the application workload on the source cluster and recreates its resources on the target cluster.

5. Optionally, in the **Migrate** window, you can select **Do not stop applications on the source cluster during migration**.

6. Click **Migrate**.

7. When the migration is complete, verify that the application migrated successfully in the OpenShift Container Platform web console:
   a. Click **Home** → **Projects**.
   b. Click the migrated project to view its status.
   c. In the **Routes** section, click **Location** to verify that the application is functioning, if applicable.
   d. Click **Workloads** → **Pods** to verify that the Pods are running in the migrated namespace.
   e. Click **Storage** → **Persistent volumes** to verify that the migrated persistent volume is correctly provisioned.

### 3.5. TROUBLESHOOTING

You can view the migration Custom Resources (CRs) and download logs to troubleshoot a failed migration.

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

**NOTE**

Manual rollback is not required if the application was not stopped during migration, because the original application is still running on the source cluster.

#### 3.5.1. Viewing migration Custom Resources (CRs)

The Cluster Application Migration (CAM) tool creates the following CRs for migration:
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.

NOTE
Deleting a MigPlan CR deletes the associated MigMigration CRs.

1 MigCluster (configuration, CAM cluster): Cluster definition
2 MigStorage (configuration, CAM cluster): Storage definition
3 MigPlan (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.

4 BackupStorageLocation (configuration, CAM cluster): Location of Velero backup objects
5 VolumeSnapshotLocation (configuration, CAM cluster): Location of Velero volume snapshots
6 MigMigration (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.
Backup (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

Restore (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

Procedure

1. Obtain the CR name:

   ```
   $ oc get <cr> -n openshift-migration
   NAME       AGE
   88435fe0-c9f8-11e9-85e6-5d593ce65e10  6m42s
   ```

   Specify the migration CR you want to view.

2. View the CR:

   ```
   $ oc describe <cr> 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
   ```

   The output is similar to the following examples.

MigMigration example

```
$ oc describe migmigration 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
Name:     88435fe0-c9f8-11e9-85e6-5d593ce65e10
Namespace: openshift-migration
Labels:   <none>
Annotations: touch: 3b48b543-b53e-4e44-9d34-33563f0f8147
API Version: migration.openshift.io/v1alpha1
Kind:     MigMigration
Metadata:
Creation Timestamp: 2019-08-29T01:01:29Z
Generation: 20
Resource Version: 88179
Self Link: /apis/migration.openshift.io/v1alpha1/namespaces/openshift-migration/migmigrations/88435fe0-c9f8-11e9-85e6-5d593ce65e10
UID:       8886de4c-c9f8-11e9-95ad-0205fe66cbb6
Spec:
Mig Plan Ref:
   Name: socks-shop-mig-plan
   Namespace: openshift-migration
Quiesce Pods: true
Stage: false
Status:
```
Conditions:
Category: Advisory
Durable: true
Last Transition Time: 2019-08-29T01:03:40Z
Message: The migration has completed successfully.
Reason: Completed
Status: True
Type: Succeeded
Phase: Completed
Start Timestamp: 2019-08-29T01:01:29Z
Events: <none>

Velero backup CR #2 example (PV data)

apiVersion: velero.io/v1
kind: Backup
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiet-pods: "true"
    openshift.io/migration-registry: 172.30.105.179:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-44dd3bd5-c9f8-11e9-95ad-0205fe66cbb6
  creationTimestamp: "2019-08-29T01:03:15Z"
  generateName: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-
geneneration: 1
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    migration-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    velero.io/storage-location: myrepo-vpzq9
    name: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
    namespace: openshift-migration
    resourceVersion: "87313"
  selfLink: /apis/velero.io/v1/namespaces/openshift-migration/backups/88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
  uid: c80dbbc0-c9f8-11e9-95ad-0205fe66cbb6
spec:
  excludedNamespaces: []
  excludedResources: []
  hooks:
    resources: []
    includeClusterResources: null
  includedNamespaces:
  - sock-shop
  includedResources:
  - persistentvolumes
  - persistentvolumeclaims
  - namespaces
  - imagestreams
  - imagestreamtags
  - secrets
  - configmaps
  - pods
  labelSelector:
    matchLabels:
migration-included-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
storageLocation: myrepo-vpzq9
ttl: 720h0m0s
volumeSnapshotLocations:
  - myrepo-wv6fx
status:
  completionTimestamp: "2019-08-29T01:02:36Z"
  errors: 0
  expiration: "2019-09-28T01:02:35Z"
  phase: Completed
  startTimestamp: "2019-08-29T01:02:35Z"
  validationErrors: null
  version: 1
  volumeSnapshotsAttempted: 0
  volumeSnapshotsCompleted: 0
  warnings: 0

Velero restore CR #2 example (Kubernetes resources)

apiVersion: velero.io/v1
kind: Restore
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.90.187:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-36f54ca7-c925-11e9-825a-06fa9fb68c88
  creationTimestamp: "2019-08-28T00:09:49Z"
  generateName: e13a1b60-c927-11e9-9555-d129df7f3b96-
generation: 3
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: e18252c9-c927-11e9-825a-06fa9fb68c88
    migration-final-restore: e18252c9-c927-11e9-825a-06fa9fb68c88
    name: e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
    namespace: openshift-migration
    resourceVersion: "82329"
    selfLink: /apis/velero.io/v1/namespaces/openshift-migration/restores/e13a1b60-c927-11e9-9555-
d129df7f3b96-gb8nx
    uid: 26983ec0-c928-11e9-825a-06fa9fb68c88
spec:
  backupName: e13a1b60-c927-11e9-9555-d129df7f3b96-sz24f
  excludedNamespaces: null
  excludedResources:
    - nodes
    - events
    - events.events.k8s.io
    - backups.velero.io
    - restores.velero.io
    - resticrepositories.velero.io
  includedNamespaces: null
  includedResources: null
  namespaceMapping: null
  restorePVs: true
status:
3.5.2. 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 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:
   a. 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.
   b. Click Download Selected to download the selected log.

Optionally, you can 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
```

3.5.3. Restic timeout error

If a migration fails because Restic times out, the following error appears in the Velero log:

```
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. You can increase this for large migrations, keeping in mind that a higher value may delay the return of error messages.

**Procedure**

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:

   ```yaml
   spec:
     restic_timeout: 1h
   ```

   1. Valid units are **h** (hours), **m** (minutes), and **s** (seconds), for example, `3h30m15s`.
5. Click **Save**.

### 3.5.4. Manually rolling back a migration

If your application was stopped during a failed migration, you must roll it back manually in order to prevent data corruption in the PV.

This procedure is not required if the application was not stopped during migration, because the original application is still running on the source cluster.

**Procedure**

1. On the target cluster, switch to the migrated project:

   ```bash
   $ oc project <project>
   ```
2. Get the deployed resources:

   ```bash
   $ oc get all
   ```
3. Delete the deployed resources to ensure that the application is not running on the target cluster and accessing data on the PVC:

   ```bash
   $ oc delete <resource_type>
   ```
4. To stop a DaemonSet without deleting it, update the **nodeSelector** in the YAML file:

   ```yaml
   apiVersion: extensions/v1beta1
   kind: DaemonSet
   metadata:
     name: hello-daemonset
   spec:
     selector:
       matchLabels:
   ```
Specify a nodeSelector value that does not exist on any node.

5. 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 the source cluster. You can remove these PVs during rollback.

Specify Recycle or Delete.

6. On the 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>

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

3.5.5. Known issues

This release has the following known issues:

- During migration, the Cluster Application Migration (CAM) tool preserves the following namespace annotations:
  - openshift.io/sa.scc.mcs
  - openshift.io/sa.scc.supplemental-groups
- **openshift.io/sa.scc.uid-range**
  These annotations preserve the UID range, ensuring that the containers retain their file system permissions on the target cluster. There is a risk that the migrated UIDs could duplicate UIDs within an existing or future namespace on the target cluster. (BZ#1748440)

- If an AWS bucket is added to the CAM web console and then deleted, its status remains **True** because the MigStorage CR is not updated. (BZ#1738564)

- Migration fails if the Migration controller is running on a cluster other than the target cluster. The **EnsureCloudSecretPropagated** phase is skipped with a logged warning. (BZ#1757571)

- Most cluster-scoped resources are not yet handled by the CAM tool. If your applications require cluster-scoped resources, you may have to create them manually on the target cluster.

- Incorrect source cluster storage class is displayed when creating the migration plan. (BZ#1777869)

- If a cluster in the CAM web console becomes inaccessible, it blocks attempts to close open migration plans. (BZ#1758269)

- If a migration fails, the migration plan does not retain custom PV settings for quiesced pods. You must manually roll back the migration, delete the migration plan, and create a new migration plan with your PV settings. (BZ#1784899)