Playbook for Verizon Migration of Cloud Pak for Data from NFS to ODF

Runbook Version

Runbook Version: **1.4.0**

Table of Contents

This migration playbook requires the `cpd-migrate.sh` and `cpd-migrate.input.json` (also referenced as input.json) which will also be provided to Verizon alongside this playbook.

Additionally, this playbook is for clusters on a restricted network. Below are the links for offline installations of CPDBR, MTC and OADP.

CPDBR: https://www.ibm.com/docs/en/cloud-paks/cp-data/4.6.x?topic=utility-moving-images-backup-restore-utilities

MTC: https://access.redhat.com/documentation/en-us/openshift_container_platform/4.10/html/migration_toolkit_for_containers/installing-mtc-restricted

MTC rsync fix image: `quay.io/konveyor/mig-controller@sha256:68effcbdddd6a96c3185daea84f8741e6881fb117ca2e70d76957d53ae779c79`

OADP: https://docs.openshift.com/container-platform/4.10/operators/admin/olm-restricted-networks.html#olm-restricted-networks

Cloud pak for data operators backup and restore

OADP installation and setup

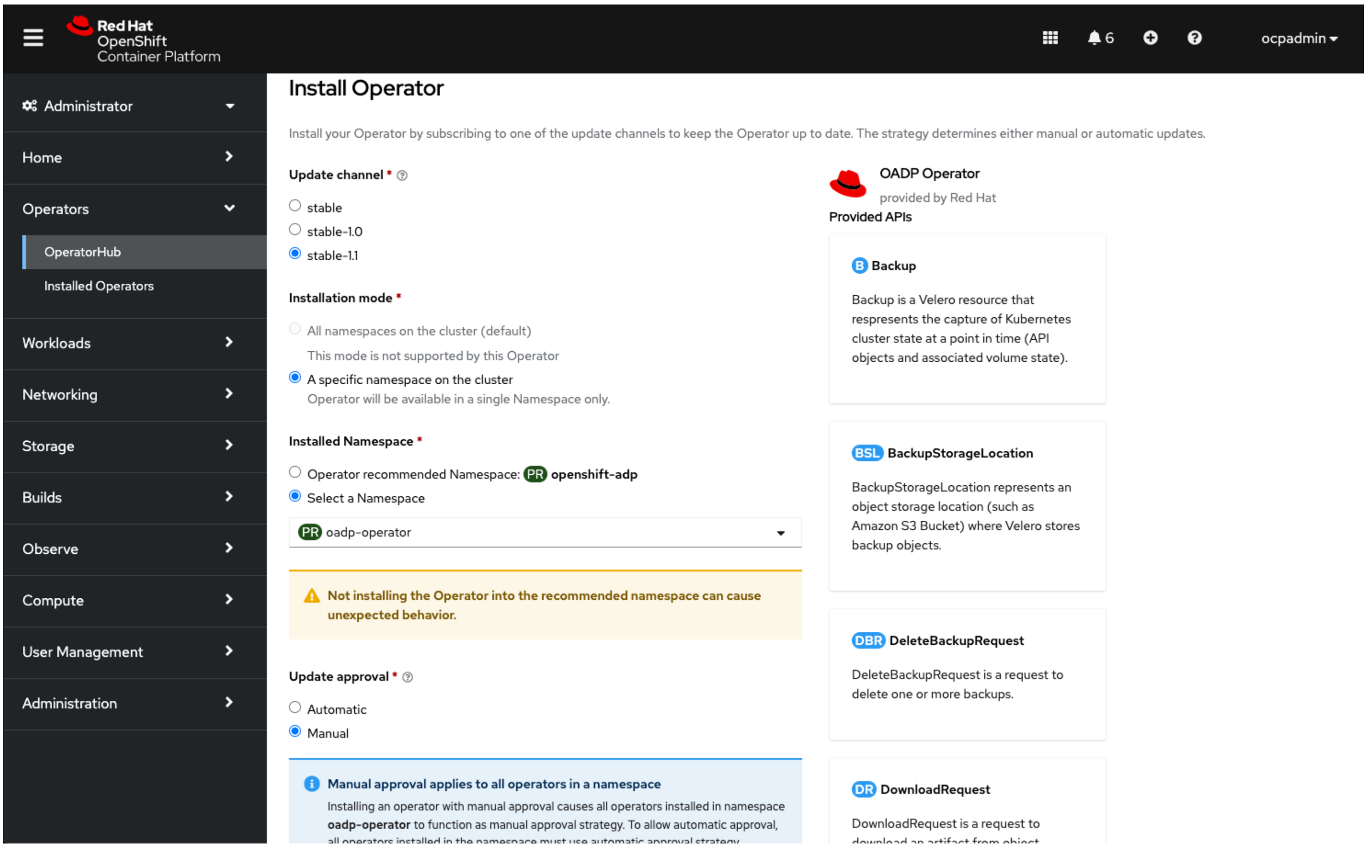
Reference: https://www.ibm.com/docs/en/cloud-paks/cp-data/4.6.x?topic=iobru-installing-cloud-pak-data-oadp-backup-restore-utility-components

This operation needs to be done on both the SOURCE cluster and the TARGET cluster.

Follow steps in the official documentation above which include:

- IMPORTANT: If OADP operator is already installed in oadp-operator namespace, reinstall it to refresh the oadp crds, due to conflict between CPD and MTC versions of OADP that are used.
- Create the oadp-operator project
- Annotate the oadp-operator project so that Restic pods can be scheduled on all nodes
- Install OADP operator
- Create Secret and environment variables for S3 backups
- Create OADP CR

Example of CPD OADP operator installation (v1.1.4):



Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

Update channel *

○ stable
○ stable-1.0
● stable-1.1

Installation mode *

○ All namespaces on the cluster (default)
This mode is not supported by this Operator
● A specific namespace on the cluster
Operator will be available in a single Namespace only.

Installed Namespace *

○ Operator recommended Namespace: PR openshift-adp
● Select a Namespace

PR oadp-operator

Not installing the Operator into the recommended namespace can cause unexpected behavior.

Update approval *

○ Automatic
● Manual

Manual approval applies to all operators in a namespace
Installing an operator with manual approval causes all operators installed in namespace **oadp-operator** to function as manual approval strategy. To allow automatic approval, all operators installed in the namespace must use automatic approval strategy.

OADP Operator
provided by Red Hat

Provided APIs

B **Backup**
Backup is a Velero resource that respresents the capture of Kubernetes cluster state at a point in time (API objects and associated volume state).

BSL **BackupStorageLocation**
BackupStorageLocation represents an object storage location (such as Amazon S3 Bucket) where Velero stores backup objects.

DBR **DeleteBackupRequest**
DeleteBackupRequest is a request to delete one or more backups.

DR **DownloadRequest**
DownloadRequest is a request to download an artifact from object

```
$ oc get csv -n oadp-operator
NAME                       DISPLAY          VERSION   REPLACES
PHASE
oadp-operator.v1.1.4   OADP Operator   1.1.4     oadp-operator.v1.1.3
Succeeded
```

Example of creation of s3 object storage credentials secret:

In this example, the s3 object storage credential secret is called `cloud-credentials`:

```
$ cat credentials-velero-2
[default]
aws_access_key_id = 131f141055654845adf3a7918178xxxx
aws_secret_access_key = xxxx
$ oc create secret generic cloud-credentials \
--namespace oadp-operator \
--from-file cloud=./credentials-velero-2
secret/cloud-credentials created
$
```

Example of creation of OADP CR. In this example it is called `dataprotectionapp-2.yaml`.

```
$ cat dataprotectionapp-2.yaml
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: cpst-dpa
spec:
  configuration:
    velero:
      customPlugins:
      - image: icr.io/cpopen/cpd/cpdbr-velero-plugin:4.0.0-beta1-1-x86_64
        name: cpdbr-velero-plugin
      defaultPlugins:
      - aws
      - openshift
      - csi
      podConfig:
        resourceAllocations:
          limits:
            cpu: "1"
            memory: 1Gi
          requests:
            cpu: 500m
            memory: 256Mi
    restic:
      enable: true
      timeout: 12h
      podConfig:
        resourceAllocations:
          limits:
            cpu: "1"
            memory: 8Gi
          requests:
            cpu: 500m
            memory: 256Mi
          tolerations:
          - key: icp4data
            operator: Exists
            effect: NoSchedule
  backupImages: false
  backupLocations:
    - velero:
        provider: aws
        default: true
        objectStorage:
          bucket: mtc-todd
          prefix: cpst-9c9f-backup
        config:
          region: us-south
          s3ForcePathStyle: "true"
          s3Url: https://s3.us-south.cloud-object-storage.appdomain.cloud
        credential:
          name: cloud-credentials
          key: cloud
$ oc apply -f dataprotectionapp-2.yaml
dataprotectionapplication.oadp.openshift.io/cpst-dpa created
$
```

Create cpd operators configmap on source cluster

Reference: https://www.ibm.com/docs/en/cloud-paks/cp-data/4.6.x?topic=utility-downloading-backup-restore-scripts

Download cpd-operators.sh script:

```
wget https://raw.githubusercontent.com/IBM/cpd-
cli/master/cpdops/files/cpd-operators.sh
```

Note: foundation-namespace and operators namespace are the same for cpd express installations (ibm-common-services by default)

```
cpd-operators.sh backup --foundation-namespace ibm-common-services --
operators-namespace ibm-common-services
```

Verify the configmap is present:

```
oc get configmap cpd-operators -n ibm-common-services
```

Backup cpd operators on source cluster

Reference: https://www.ibm.com/docs/en/cloud-paks/cp-data/4.6.x?topic=obr-scenario-creating-offline-backup-cloud-pak-data-instance-restoring-it-different-cluster#oadp-new-cluster__backup_foundational_operators

Configure cpd-cli oadp to point to the project where the Velero instance is installed:

```
cpd-cli oadp client config set namespace=oadp-operator
```

Trigger a cpd oadp offline backup for cpd-operators namespace:

```
cpd-cli oadp backup create <cpd-ops-backup-name> --include-namespaces ibm-common-services --include-resources='namespaces,operatorgroups,configmaps,scheduling,crd' --skip-hooks --log-level=debug --verbose
```

List backups:

```
cpd-cli oadp backup ls
```

Verify the backup:

```
cpd-cli oadp backup status <cpd-ops-backup-name> --details
```

Restore cpd operators on target cluster

Check catalog sources on the target cluster and clean up any CPD related catalog sources

Before running the CPD operator subscriptions restore, make sure the catalog sources are cleaned of CPD related ones on the target cluster.

Example output, showing no CPD related catalog sources:

```
$ oc get catsrc -n openshift-marketplace
NAME                   DISPLAY               TYPE   PUBLISHER   AGE
certified-operators    Certified Operators   grpc   Red Hat     21d
community-operators    Community Operators   grpc   Red Hat     21d
ibm-operator-catalog   IBM Operator Catalog  grpc   IBM         21d
redhat-marketplace     Red Hat Marketplace   grpc   Red Hat     21d
redhat-operators       Red Hat Operators     grpc   Red Hat     21d
$
```

Verify ibm-common-services namespace does not exist:

```
oc get ns |grep common
```

Configure cpd-cli oadp to point to the project where the Velero instance is installed:

```
cpd-cli oadp client config set namespace=oadp-operator
```

Restore cpd operators crds:

```
cpd-cli oadp restore create <cpd-operators-restore-crds> \
--from-backup=<cpd-ops-backup-name> \
--include-resources='crd' \
--include-cluster-resources=true \
--skip-hooks \
--log-level=debug \
--verbose
```

Verify crd restore operation:

```
cpd-cli oadp restore status <cpd-operators-restore-crds> --details
```

Restore additional resources, such as projects and operator groups, etc:

Restore command:

```
cpd-cli oadp restore create <cpd-operators-restore-remaining> \
--from-backup=<cpd-ops-backup-name>  \
--include-resources='namespaces,operatorgroups,scheduling,crd' \
--include-cluster-resources=true \
--skip-hooks \
--log-level=debug \
--verbose
```

Verify ibm-common-services namespace and resources were created:

```
$ oc get ns |grep ibm-common-services
ibm-common-services                                Active   67s
$ oc get operatorgroups -n ibm-common-services
NAME            AGE
operatorgroup   83s
```

Verify restore operation for remaining resources:

```
cpd-cli oadp restore status <cpd-operators-restore-remaining> --details
```

Restore cpd operators configmaps:

Configmaps before restore:

```
$ oc get cm -n ibm-common-services
NAME                       DATA   AGE
```

```
kube-root-ca.crt            1      2m11s
openshift-service-ca.crt    1      2m11s
```

Restore command:

```
cpd-cli oadp restore create <cpd-operators-restore-cm> \
 --from-backup=<cpd-ops-backup-name> \
 --include-resources='configmaps' \
 --selector 'app=cpd-operators-backup' \
 --skip-hooks \
 --log-level=debug \
 --verbose
```

Configmaps after restore:

```
$ oc get cm -n ibm-common-services
NAME                        DATA   AGE
cpd-operators               18     24s
kube-root-ca.crt            1      3m30s
openshift-service-ca.crt    1      3m30s
```

Verify restore operation for configmaps:

```
cpd-cli oadp restore status <cpd-operators-restore-cm>
```

Restore operator subscriptions Reference: https://www.ibm.com/docs/en/cloud-paks/cp-data/4.6.x?topic=utility-downloading-backup-restore-scripts

Download cpd-operators.sh script:

```
wget https://raw.githubusercontent.com/IBM/cpd-
cli/master/cpdops/files/cpd-operators.sh
```

Run command:

```
cpd-operators.sh restore --foundation-namespace ibm-common-services --
operators-namespace ibm-common-services
```

Verify restore operator subscriptions:

```
$ oc get sub -n ibm-common-services
NAME
PACKAGE                              SOURCE
CHANNEL
cpd-operator
cpd-platform-operator                cpd-platform
v3.8
fdb-kubernetes-operator-v2.4-ibm-fdb-operator-catalog-openshift-
marketplace   fdb-kubernetes-operator              ibm-fdb-operator-
catalog                     v2.4
ibm-cert-manager-operator
ibm-cert-manager-operator            opencloud-operators
v3.23
ibm-common-service-operator
ibm-common-service-operator          opencloud-operators
v3.23
ibm-cpd-ae-operator
analyticsengine-operator             ibm-cpd-ae-operator-catalog
v3.5
ibm-cpd-ccs-operator
ibm-cpd-ccs                          ibm-cpd-ccs-operator-catalog
v6.5
ibm-cpd-datarefinery-operator
ibm-cpd-datarefinery                 ibm-cpd-datarefinery-operator-catalog
v6.5
ibm-cpd-datastage-operator
ibm-cpd-datastage-operator           ibm-cpd-datastage-operator-catalog
v3.5
ibm-cpd-iis-operator
ibm-cpd-iis                          ibm-cpd-iis-operator-catalog
v3.5
ibm-cpd-wkc-operator-catalog-subscription
ibm-cpd-wkc                          ibm-cpd-wkc-operator-catalog
v3.5
ibm-cpd-wml-operator
ibm-cpd-wml-operator                 ibm-cpd-wml-operator-catalog
v3.5
ibm-cpd-ws-operator
ibm-cpd-wsl                          ibm-cpd-ws-operator-catalog
v6.5
ibm-cpd-ws-runtimes-operator
ibm-cpd-ws-runtimes                  ibm-cpd-ws-runtimes-operator-catalog
v6.5
ibm-db2aaservice-cp4d-operator
ibm-db2aaservice-cp4d-operator       ibm-db2aaservice-cp4d-operator-
catalog       v3.2
ibm-db2oltp-cp4d-operator-catalog-subscription
ibm-db2oltp-cp4d-operator            ibm-db2oltp-cp4d-operator-catalog
v3.2
ibm-db2u-operator
db2u-operator                        ibm-db2uoperator-catalog
v3.2
ibm-db2wh-cp4d-operator-catalog-subscription
```

```
ibm-db2wh-cp4d-operator              ibm-db2wh-cp4d-operator-catalog
v3.2
ibm-dmc-operator-subscription
ibm-dmc-operator                     ibm-dmc-operator-catalog
v2.2
ibm-fdb-operator
ibm-opencontent-foundationdb         ibm-fdb-operator-catalog
v2.4
ibm-namespace-scope-operator
ibm-namespace-scope-operator         opencloud-operators
v3.23
ibm-watson-openscale-operator-subscription
ibm-cpd-wos                          ibm-openscale-operator-catalog
v3.5
ibm-zen-operator
ibm-zen-operator                     opencloud-operators
v3.23
manta-adl-operator
manta-adl-operator                   manta-adl-operator-catalog
v1.10
operand-deployment-lifecycle-manager-app
ibm-odlm                             opencloud-operators
v3.23
redis-operator
ibm-cloud-databases-redis-operator   ibm-cloud-databases-redis-operator-
catalog   v1.6
```

```
$ oc get csv -n ibm-common-services
NAME                                           DISPLAY
VERSION   REPLACES                           PHASE
cpd-platform-operator.v3.8.0                   Cloud Pak for Data Platform
Operator                                 3.8.0     cpd-platform-
operator.v3.7.0       Succeeded
db2u-operator.v3.2.0                           IBM Db2
3.2.0                                        Succeeded
fdb-kubernetes-operator.v2.4.6                 FoundationDB Kubernetes
2.4.6                                        Succeeded
ibm-cert-manager-operator.v3.25.2              IBM Cert Manager
3.25.2                                       Succeeded
ibm-cloud-databases-redis.v1.6.6               IBM Operator for Redis
1.6.6     ibm-cloud-databases-redis.v1.6.5   Succeeded
ibm-common-service-operator.v3.23.2            IBM Cloud Pak foundational
services                                  3.23.2
Succeeded
ibm-cpd-ae.v3.5.0                              IBM Analytics Engine
Powered by Apache Spark Service                 3.5.0
Succeeded
ibm-cpd-ccs.v6.5.0                             Common Core Services
6.5.0                                        Succeeded
ibm-cpd-datarefinery.v6.5.0                    IBM Data Refinery
6.5.0                                        Succeeded
```

```
ibm-cpd-datastage-operator.v3.5.0              IBM DataStage
3.5.0                                        Succeeded
ibm-cpd-iis.v1.6.5                             IIS Services
1.6.5                                        Succeeded
ibm-cpd-wkc.v1.6.5                             WKC Services
1.6.5                                        Succeeded
ibm-cpd-wml-operator.v3.5.0                    IBM WML Services
3.5.0                                        Succeeded
ibm-cpd-wos.v3.5.0                             IBM Watson OpenScale
3.5.0                                        Succeeded
ibm-cpd-ws-runtimes.v6.5.0                     Watson Studio Notebook
Runtimes                                      6.5.0
Succeeded
ibm-cpd-wsl.v6.5.0                             Watson Studio
6.5.0                                        Succeeded
ibm-databases-dmc.v2.2.0                       IBM Db2 Data Management
Console                                      2.2.0
Succeeded
ibm-db2aaservice-cp4d-operator.v3.2.0          IBM Db2 as a Service
Operator Extension for IBM Cloud Pak for Data   3.2.0
Succeeded
ibm-db2oltp-cp4d-operator.v3.2.0               IBM® Db2 Operator Extension
for IBM Cloud Pak for Data               3.2.0
Succeeded
ibm-db2wh-cp4d-operator.v3.2.0                 IBM® Db2 Warehouse Operator
Extension for IBM® Cloud Pak for Data    3.2.0
Succeeded
ibm-namespace-scope-operator.v1.17.2           IBM NamespaceScope Operator
1.17.2                                       Succeeded
ibm-opencontent-foundationdb.v2.4.6            IBM Opencontent
FoundationDB                                         2.4.6
Succeeded
ibm-zen-operator.v1.8.3                        IBM Zen Service
1.8.3                                        Succeeded
manta-adl-operator.v1.10.0                     MANTA Automated Data
Lineage                                         1.10.0
Succeeded
operand-deployment-lifecycle-manager.v1.21.2   Operand Deployment
Lifecycle Manager                                 1.21.2
Succeeded
```

Note: The exact subscription and csv installed can vary based on the services installed on source cluster.

Collect an oadp resources dump of the cpd namespace on source cluster

To collect CPD resources from source cluster, perform the following:

```
cpd-cli oadp backup create <src-cpd-instance-resources> \
--include-namespaces <cpd-namespace> \
--exclude-resources='event,event.events.k8s.io,imagetags.openshift.io' \
--include-cluster-resources=true \
--snapshot-volumes=false \
--skip-hooks=true \
--log-level=debug \
--verbose
```

Download the resource list:

```
cpd-cli oadp backup download <src-cpd-instance-resources> -o <src-cpd-
instance-resources>-data.tar.gz
```

Provide the data.tar.gz tarball to IBM. This will be used to help identify all of the resources with storage class references and will be used to build the input.json file, which will be used in a later step to run the cpd-migrate.sh tool.

Example for how to use the cpd resource tarball to identify resources with storage class references:

Extract tarball

```
$ tar -xzf src-cpd-instance-resources-data.tar.gz
$ pwd
/home/admin/cp4d/cpd-cli-linux-EE-12.0.5-61/src-cpd-instance-resources-
data
$ ls -l
total 12
drwxrwxr-x.   2 admin admin   21 May 15 19:44 metadata
drwxrwxr-x. 129 admin admin 8192 May 15 19:44 resources
```

From the extracted oadp tar bundle, we can perform a series of grep commands to narrow in on the subset of k8s resources that are affected with storage class references.

Use helper commands to find potential resources to change.

- If looking at the cpd-cli oadp backup tar bundle, example grep commands to help find potential resources for patching.
- The best grep command will be to grep on the actual storageclass name(s). This will directly find where the storageclass is referenced.

```
grep -Rl -E "<cpd-storage-class>" ./* | \
grep -vE
"persistentvolumes|persistentvolumeclaims|preferredversion|storageclasses.
storage.k8s.io"
```

- The next grep command is to identify which of those json files from the previous grep might have both RWO and RWX references in them.

```
grep -Rl -E "<cpd-storage-class>" ./* | \
grep -vE
"persistentvolumes|persistentvolumeclaims|preferredversion|storageclasses.
storage.k8s.io" | \
xargs -I{} sh -c "echo ------ {} -------;cat {} | \
jq . | grep -E 'ReadWriteOnce|ReadWriteMany|<cpd-storage-class>'"
```

MTC installation and setup on source and target clusters

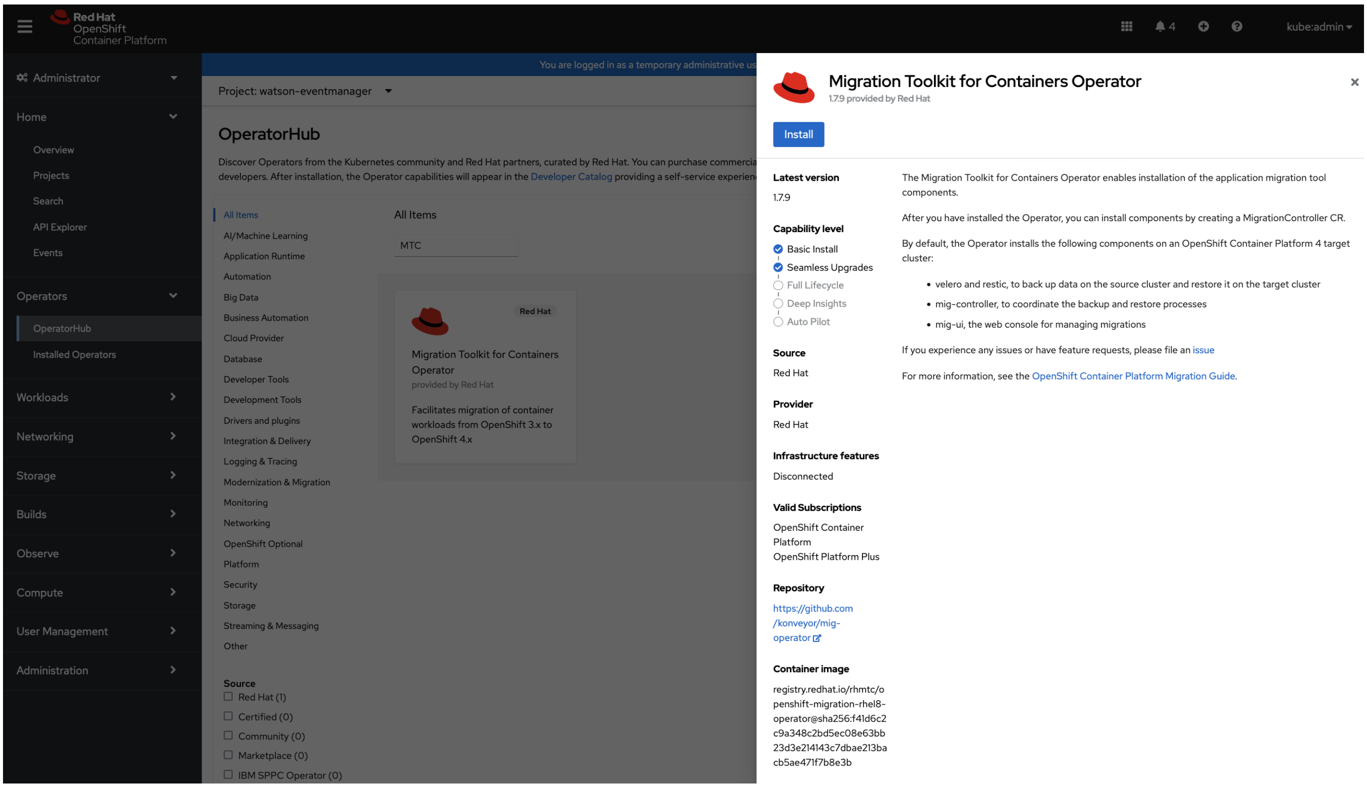
Summary of steps in this section:

- IMPORTANT: If MTC is already installed in openshift-migration namespace, reinstall just the OADP operator (version v1.0.x) in the openshift-migration namespace to refresh the oadp crds, due to conflict between CPD and MTC versions of OADP that are used.
- Install Migration Toolkit for Containers Operator (MTC) (currently v1.7.9) on both source and target clusters
- From source cluster MTC, add target cluster "Add cluster"
- Add replication repository (s3 object store, accessible from both source and target clusters)

Install MTC and create MigrationController on source and target clusters

MTC and MigrationControllor need to be installed and created on both the source cluster and the target cluster.

Install MTC operator from Operator hub:



You are logged in as a temporary administrative us

Project: watson-eventmanager

OperatorHub

Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. You can purchase commercia developers. After installation, the Operator capabilities will appear in the Developer Catalog providing a self-service experien

All Items

MTC

Migration Toolkit for Containers Operator
provided by Red Hat

Facilitates migration of container workloads from OpenShift 3.x to OpenShift 4.x

Migration Toolkit for Containers Operator
1.7.9 provided by Red Hat

Latest version
1.7.9

Capability level
☑ Basic Install
☑ Seamless Upgrades
○ Full Lifecycle
○ Deep Insights
○ Auto Pilot

Source
Red Hat

Provider
Red Hat

Infrastructure features
Disconnected

Valid Subscriptions
OpenShift Container Platform
OpenShift Platform Plus

Repository
https://github.com/konveyor/mig-operator

Container image
registry.redhat.io/rhmtc/openshift-migration-rhel8-operator@sha256:f41d6c2c9a348c2bd5ec08e63bb23d3e214143c7dbae213bacb5ae471f7b8e3b

The Migration Toolkit for Containers Operator enables installation of the application migration tool components.

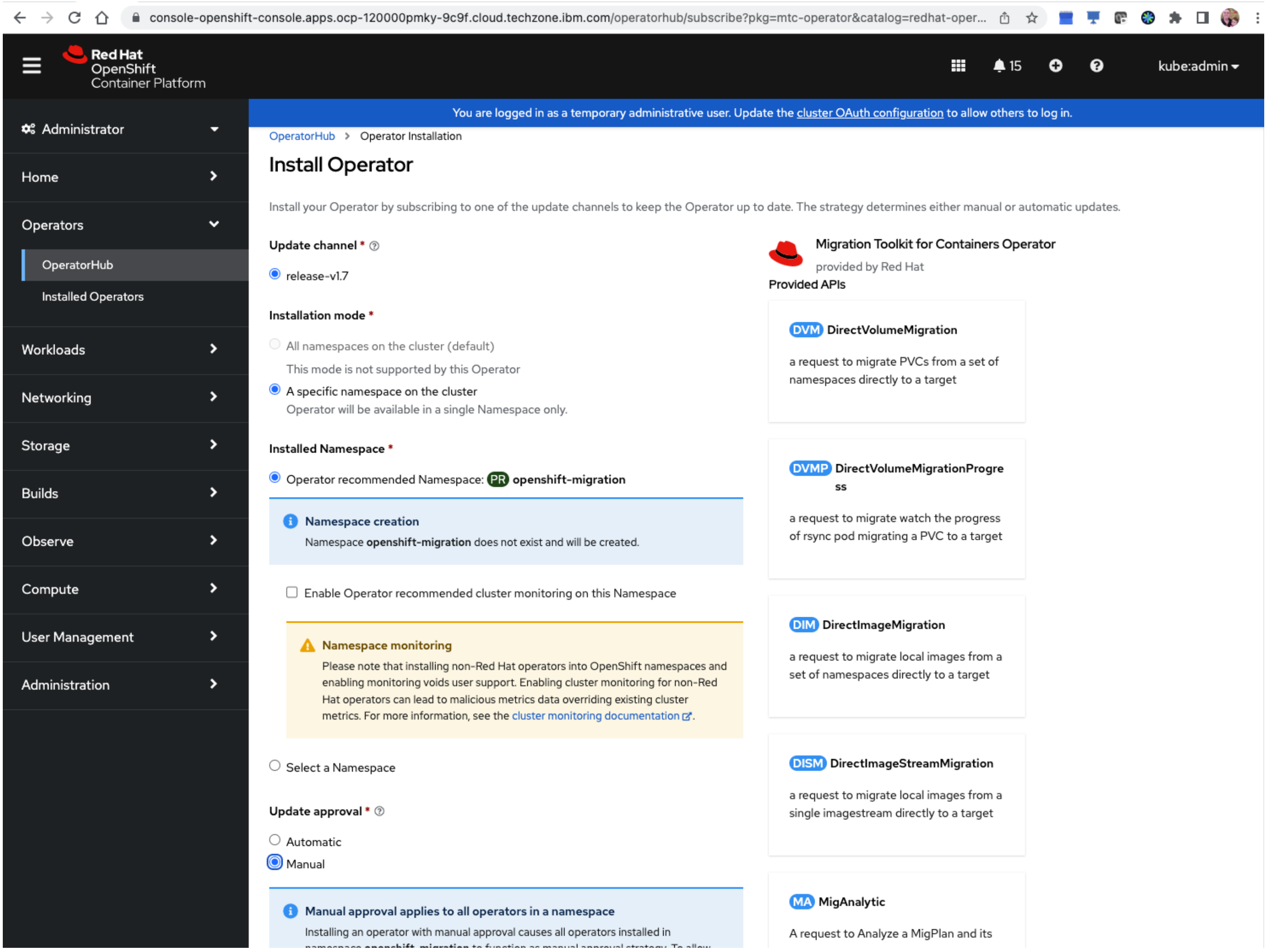
After you have installed the Operator, you can install components by creating a MigrationController CR.

By default, the Operator installs the following components on an OpenShift Container Platform 4 target cluster:

- velero and restic, to back up data on the source cluster and restore it on the target cluster
- mig-controller, to coordinate the backup and restore processes
- mig-ui, the web console for managing migrations

If you experience any issues or have feature requests, please file an issue

For more information, see the OpenShift Container Platform Migration Guide.



You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

OperatorHub > Operator Installation

Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

Update channel *

● release-v1.7

Installation mode *

○ All namespaces on the cluster (default)
This mode is not supported by this Operator

● A specific namespace on the cluster
Operator will be available in a single Namespace only.

Installed Namespace *

● Operator recommended Namespace: PR openshift-migration

Namespace creation
Namespace **openshift-migration** does not exist and will be created.

□ Enable Operator recommended cluster monitoring on this Namespace

Namespace monitoring
Please note that installing non-Red Hat operators into OpenShift namespaces and enabling monitoring voids user support. Enabling cluster monitoring for non-Red Hat operators can lead to malicious metrics data overriding existing cluster metrics. For more information, see the cluster monitoring documentation.

○ Select a Namespace

Update approval *

○ Automatic
● Manual

Manual approval applies to all operators in a namespace
Installing an operator with manual approval causes all operators installed in

Migration Toolkit for Containers Operator
provided by Red Hat

Provided APIs

DVM DirectVolumeMigration
a request to migrate PVCs from a set of namespaces directly to a target

DVMP DirectVolumeMigrationProgress
a request to migrate watch the progress of rsync pod migrating a PVC to a target

DIM DirectImageMigration
a request to migrate local images from a set of namespaces directly to a target

DISM DirectImageStreamMigration
a request to migrate local images from a single imagestream directly to a target

MA MigAnalytic
A request to Analyze a MigPlan and its

After operator is installed, select to create a MigrationController instance.

- Change the view from "Form view" to "YAML view"
- Add the following properties to YAML for known rsync issue under the `spec` section:

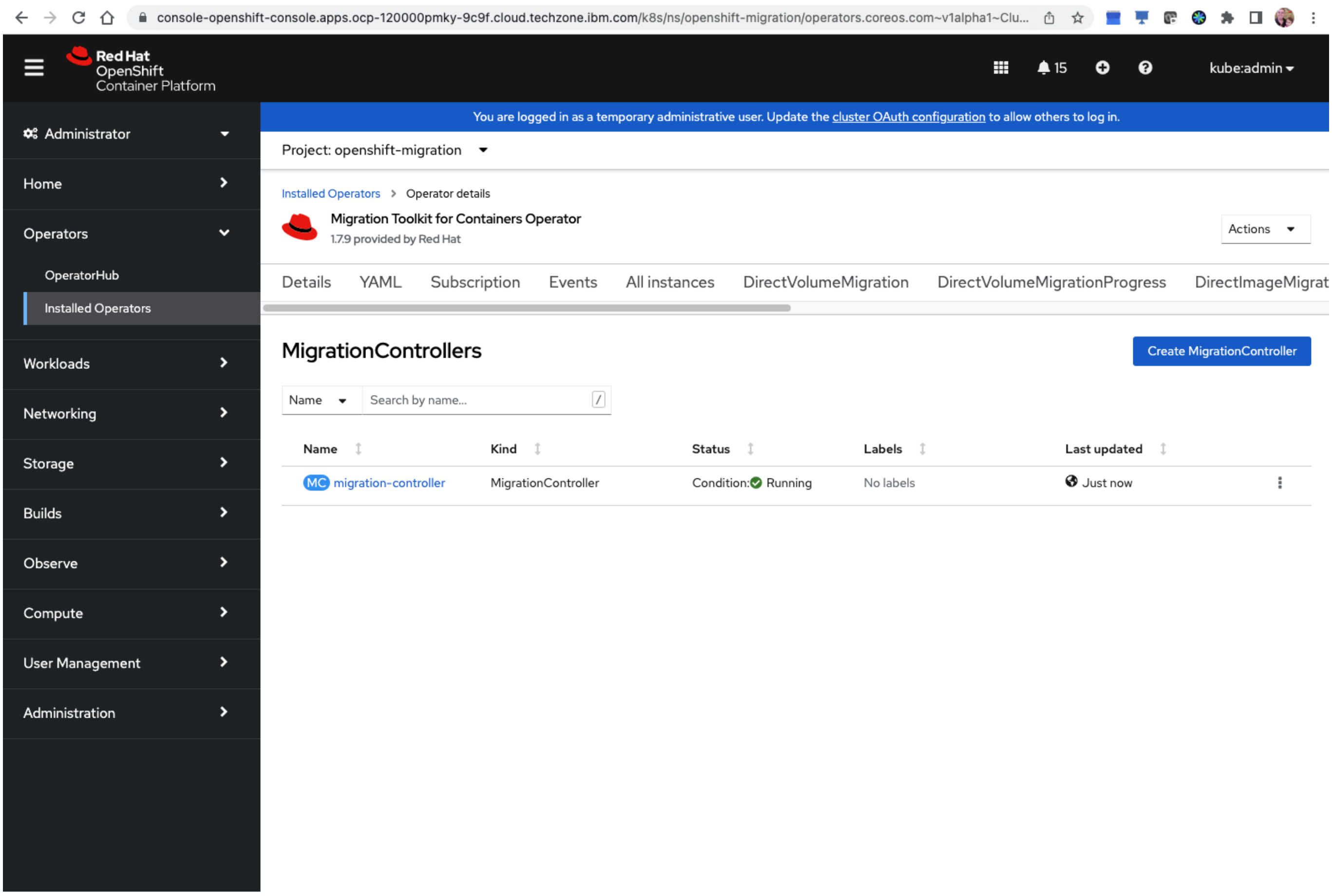
```
  mig_controller_image_fqin: quay.io/konveyor/mig-controller@sha256:68effcbdddd6a96c3185daea84f8741e6881fb117ca2e70d76957d53ae779c79
  migration_rsync_privileged: true
```

Example YAML with updated `spec` section:

```
apiVersion: migration.openshift.io/v1alpha1
kind: MigrationController
metadata:
  creationTimestamp: '2023-05-25T07:55:19Z'
  generation: 2
  managedFields:
    - apiVersion: migration.openshift.io/v1alpha1
      fieldsType: FieldsV1
      fieldsV1:
        'f:spec':
          'f:migration_controller': {}
          'f:migration_log_reader': {}
          'f:cluster_name': {}
          'f:restic_timeout': {}
          'f:migration_rsync_privileged': {}
          'f:mig_pv_limit': {}
          'f:migration_velero': {}
          .: {}
          'f:mig_namespace_limit': {}
          'f:mig_controller_image_fqin': {}
          'f:azure_resource_group': {}
          'f:mig_pod_limit': {}
          'f:migration_ui': {}
          'f:olm_managed': {}
      manager: Mozilla
      operation: Update
      time: '2023-05-25T07:55:19Z'
    - apiVersion: migration.openshift.io/v1alpha1
      fieldsType: FieldsV1
      fieldsV1:
        'f:status':
          .: {}
          'f:conditions': {}
      manager: ansible-operator
      operation: Update
      subresource: status
      time: '2023-05-25T07:55:19Z'
    - apiVersion: migration.openshift.io/v1alpha1
      fieldsType: FieldsV1
      fieldsV1:
        'f:spec':
          'f:version': {}
      manager: OpenAPI-Generator
      operation: Update
      time: '2023-05-25T07:55:30Z'
  name: migration-controller
  namespace: openshift-migration
  resourceVersion: '44253841'
  uid: aa036929-1a16-4e7f-b967-5f0c3ad3bfb6
spec:
  mig_controller_image_fqin: >-
    quay.io/konveyor/mig-controller@sha256:68effcbdddd6a96c3185daea84f8741e6881fb117ca2e70d76957d53ae779c79
  mig_namespace_limit: '10'
  migration_ui: true
  mig_pod_limit: '100'
  migration_controller: true
  migration_log_reader: true
  olm_managed: true
  cluster_name: host
  restic_timeout: 1h
  migration_rsync_privileged: true
  migration_velero: true
```

```
    mig_pv_limit: '100'
    version: 1.7.9
    azure_resource_group: ''
  status:
    conditions:
      - lastTransitionTime: '2023-05-25T07:56:16Z'
        message: ''
        reason: ''
        status: 'False'
        type: Failure
      - ansibleResult:
          changed: 0
          completion: '2023-05-31T21:17:47.506537'
          failures: 0
          ok: 51
          skipped: 22
        lastTransitionTime: '2023-05-25T07:55:19Z'
        message: Awaiting next reconciliation
        reason: Successful
        status: 'True'
        type: Running
      - lastTransitionTime: '2023-05-31T21:17:47Z'
        message: Last reconciliation succeeded
        reason: Successful
        status: 'True'
        type: Successful
```

Click "Create" and ensure MTC status is `Running`.



console-openshift-console.apps.ocp-120000pmky-9c9f.cloud.techzone.ibm.com/k8s/ns/openshift-migration/operators.coreos.com~v1alpha1~Clu...

You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Project: openshift-migration

Installed Operators > Operator details

Migration Toolkit for Containers Operator
1.7.9 provided by Red Hat

MigrationControllers

Name	Kind	Status	Labels	Last updated
MC migration-controller	MigrationController	Condition: ✓ Running	No labels	Just now

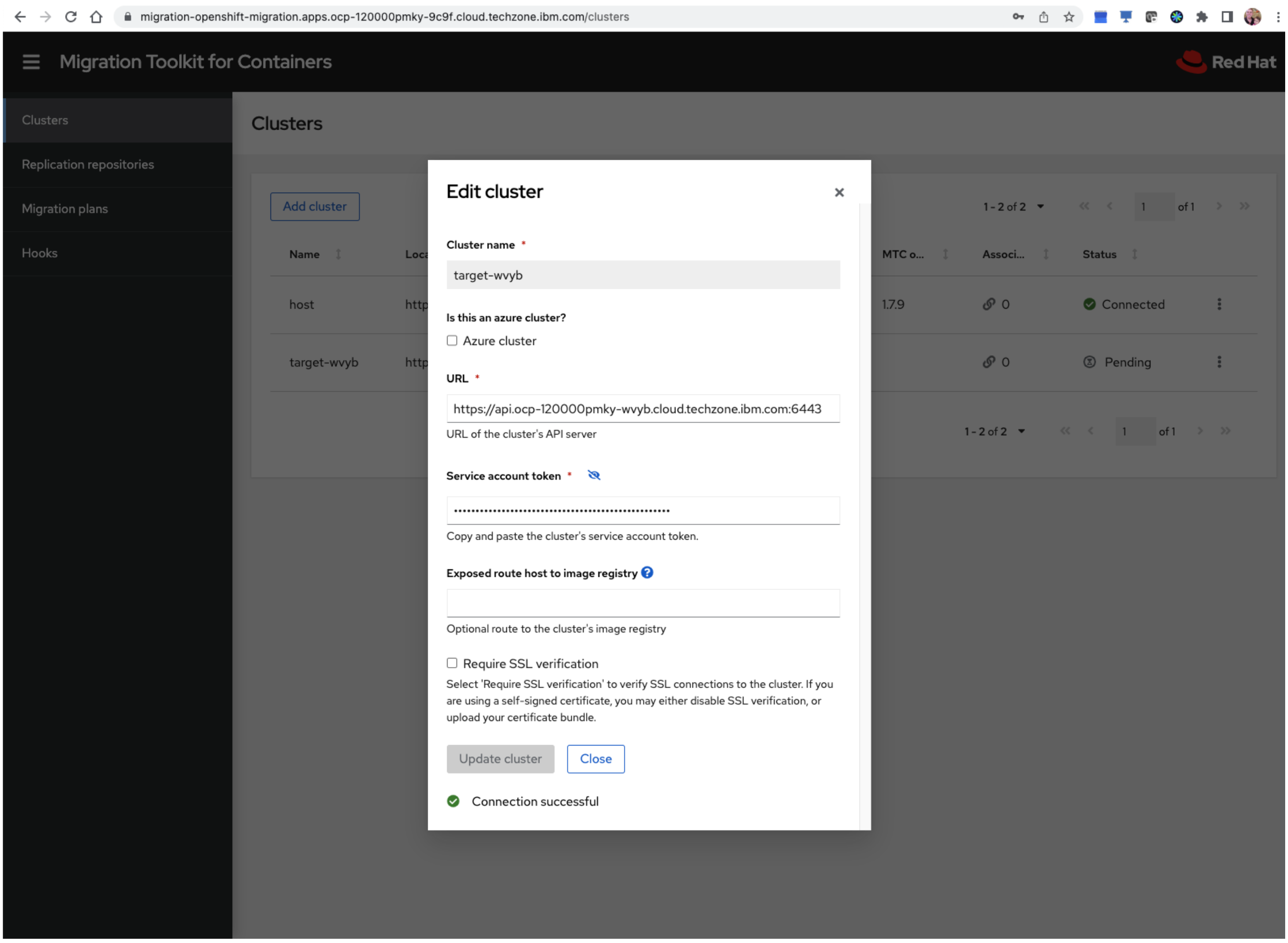
Add target cluster to source cluster MTC:

Launch the migration controller console by clicking on the route created in "openshift-migration" namespace.

From source cluster MTC controller, go to "Clusters" > "Add cluster".

Add target cluster access details. For Service account token, you can execute the following commands in the target cluster. Please find the correct secret in your environment by executing `oc -n openshift-migration get secret|grep velero-token` and substite it in the command below:

```
oc -n openshift-migration get secret velero-token-XXXXX -
ojsonpath='{.data.token}'|base64 -d
```



migration-openshift-migration.apps.ocp-120000pmky-9c9f.cloud.techzone.ibm.com/clusters
Migration Toolkit for Containers
Red Hat
Clusters
Replication repositories
Migration plans
Hooks
Add cluster
Update cluster
Close

Clusters

1 - 2 of 2 · 1 of 1

Name	Loc…	MTC o…	Associ…	Status
host	http…	1.7.9	0	Connected
target-wvyb	http…		0	Pending

Edit cluster

Cluster name *: target-wvyb

Is this an azure cluster?
☐ Azure cluster

URL *: https://api.ocp-120000pmky-wvyb.cloud.techzone.ibm.com:6443
URL of the cluster's API server

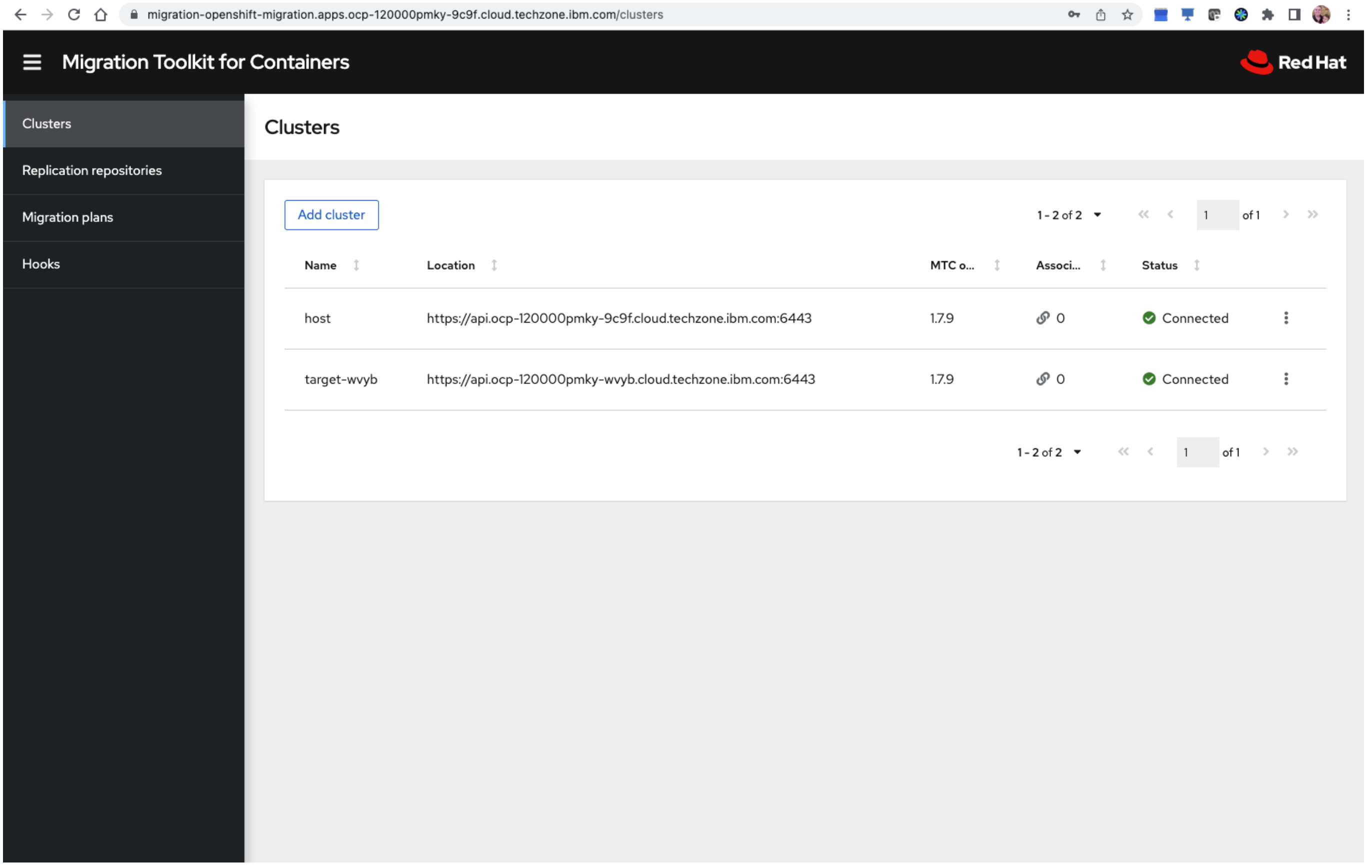
Service account token *: ••
Copy and paste the cluster's service account token.

Exposed route host to image registry
Optional route to the cluster's image registry

☐ Require SSL verification
Select 'Require SSL verification' to verify SSL connections to the cluster. If you are using a self-signed certificate, you may either disable SSL verification, or upload your certificate bundle.

✓ Connection successful

Next, verify source and target clusters are available and connected.

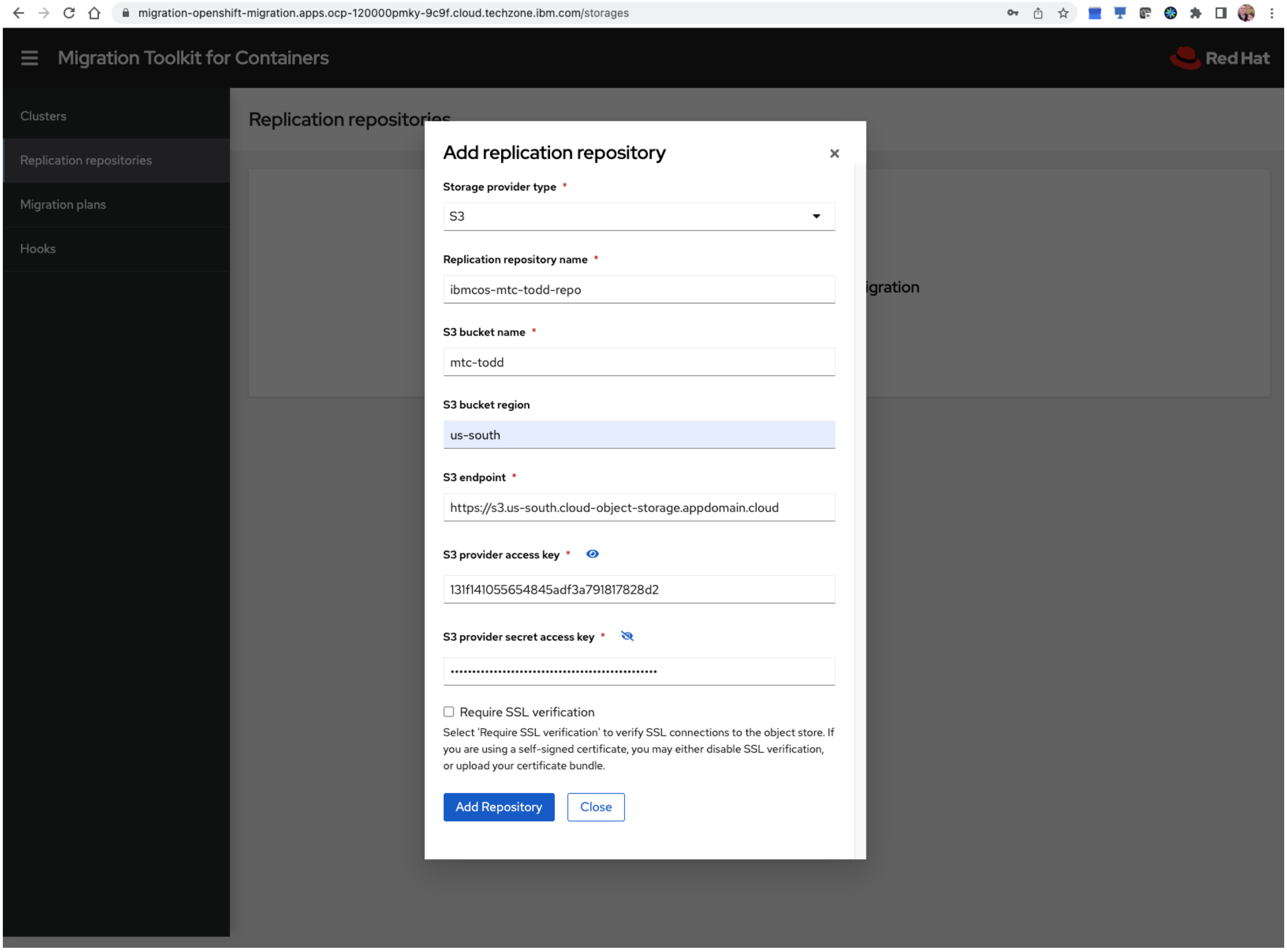


migration-openshift-migration.apps.ocp-120000pmky-9c9f.cloud.techzone.ibm.com/clusters
Migration Toolkit for Containers
Red Hat
Clusters
Replication repositories
Migration plans
Hooks
Clusters
Add cluster
1 - 2 of 2
1
of 1

Name	Location	MTC o...	Associ...	Status
host	https://api.ocp-120000pmky-9c9f.cloud.techzone.ibm.com:6443	1.7.9	0	Connected
target-wvyb	https://api.ocp-120000pmky-wvyb.cloud.techzone.ibm.com:6443	1.7.9	0	Connected

Add replication repository (access to s3 object store):

From source cluster MTC controller. Go to "Replication repositories" > "Add replication repository" Add the s3 object store access details and click "Add repository" and verify it is successfully connected.


migration-openshift-migration.apps.ocp-120000pmky-9c9f.cloud.techzone.ibm.com/storages
Migration Toolkit for Containers
Red Hat
Clusters
Replication repositories
Migration plans
Hooks
Replication repositories
igration

Add replication repository

Storage provider type *
S3

Replication repository name *
ibmcos-mtc-todd-repo

S3 bucket name *
mtc-todd

S3 bucket region
us-south

S3 endpoint *
https://s3.us-south.cloud-object-storage.appdomain.cloud

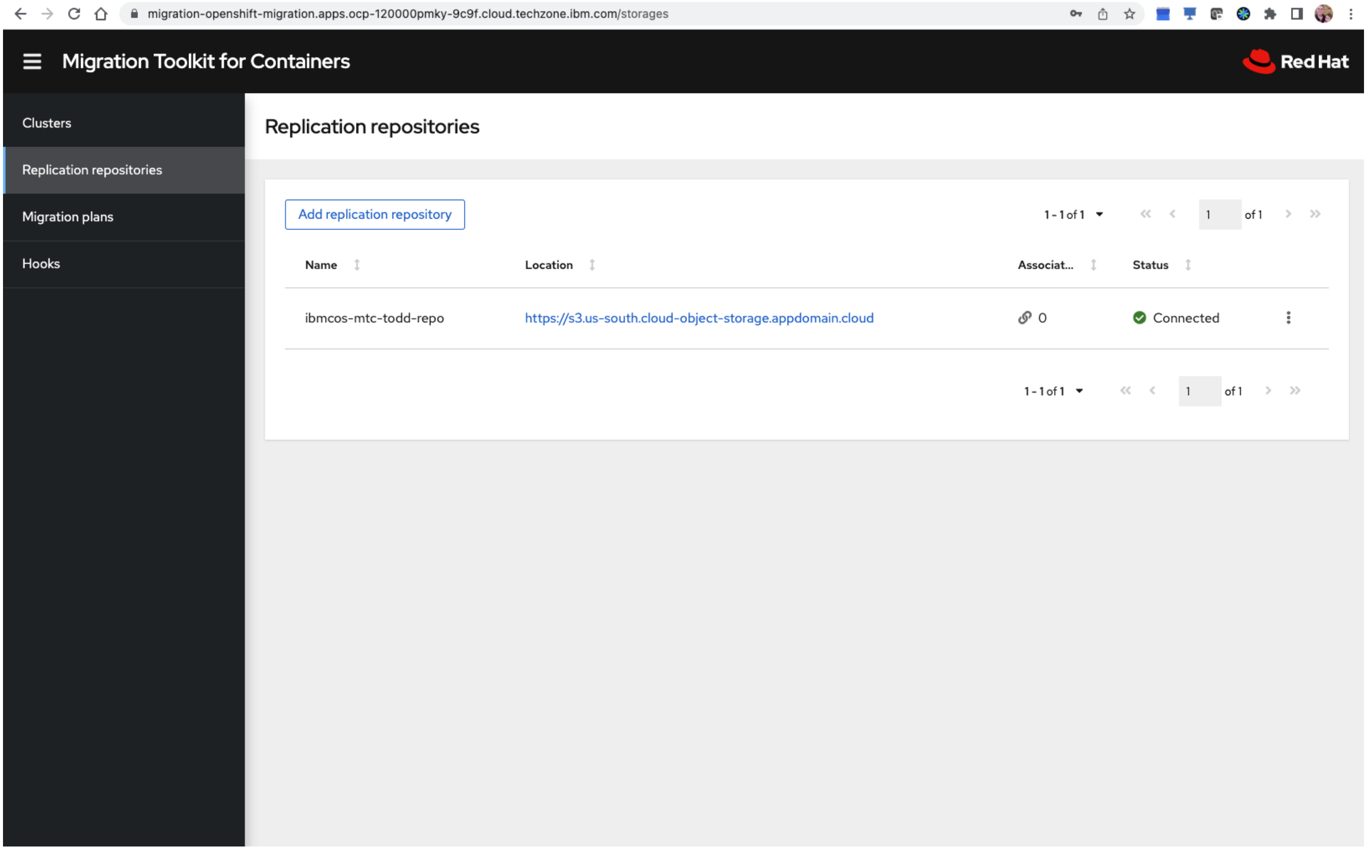
S3 provider access key *
131f141055654845adf3a791817828d2

S3 provider secret access key *
••

□ Require SSL verification

Select 'Require SSL verification' to verify SSL connections to the object store. If you are using a self-signed certificate, you may either disable SSL verification, or upload your certificate bundle.

Repo should now be accessible:

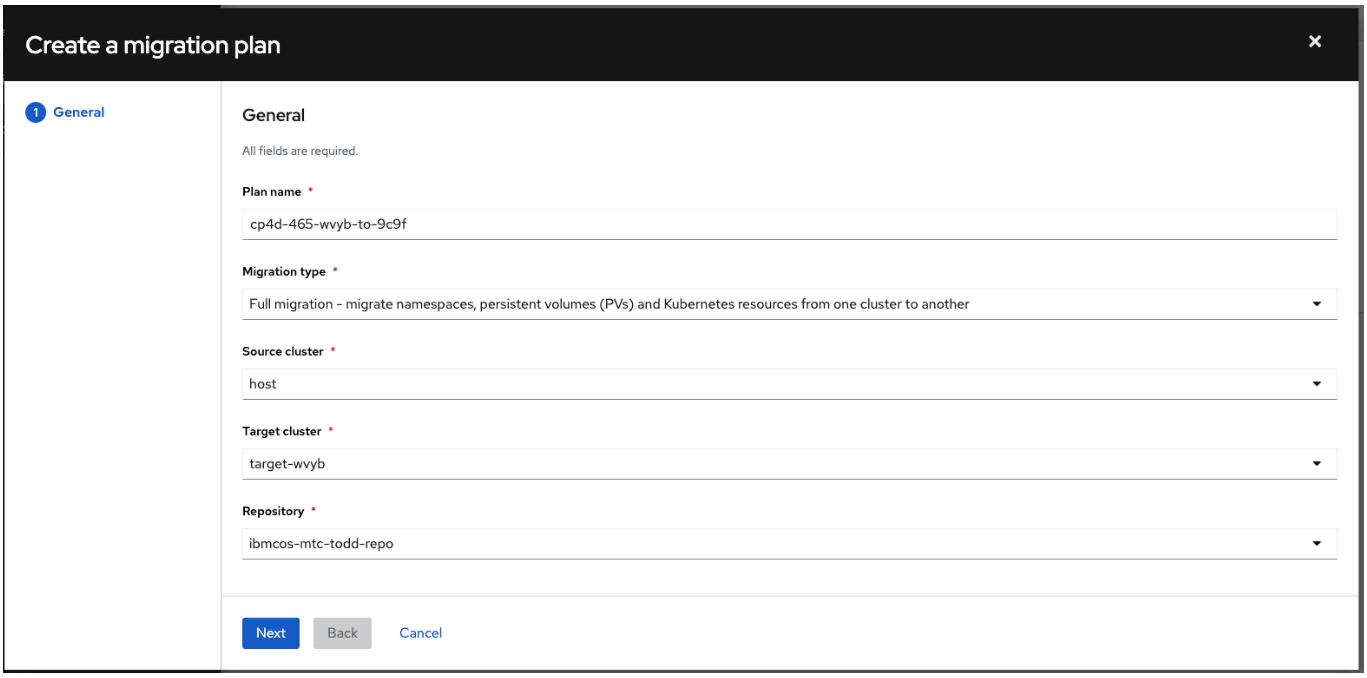

migration-openshift-migration.apps.ocp-120000pmky-9c9f.cloud.techzone.ibm.com/storages
Migration Toolkit for Containers
Red Hat
Clusters
Replication repositories
Migration plans
Hooks
Add replication repository
1 - 1 of 1
1 of 1

Replication repositories

Name	Location	Associat...	Status
ibmcos-mtc-todd-repo	https://s3.us-south.cloud-object-storage.appdomain.cloud	0	✓ Connected

MTC Migration plan creation

From source MTC migration console, select "Migration Plans" > "Add migration plan" Select "Full migration" migration type. Complete the remainder of the "General" section of the migration plan.



Create a migration plan

General

All fields are required.

Plan name *: cp4d-465-wvyb-to-9c9f

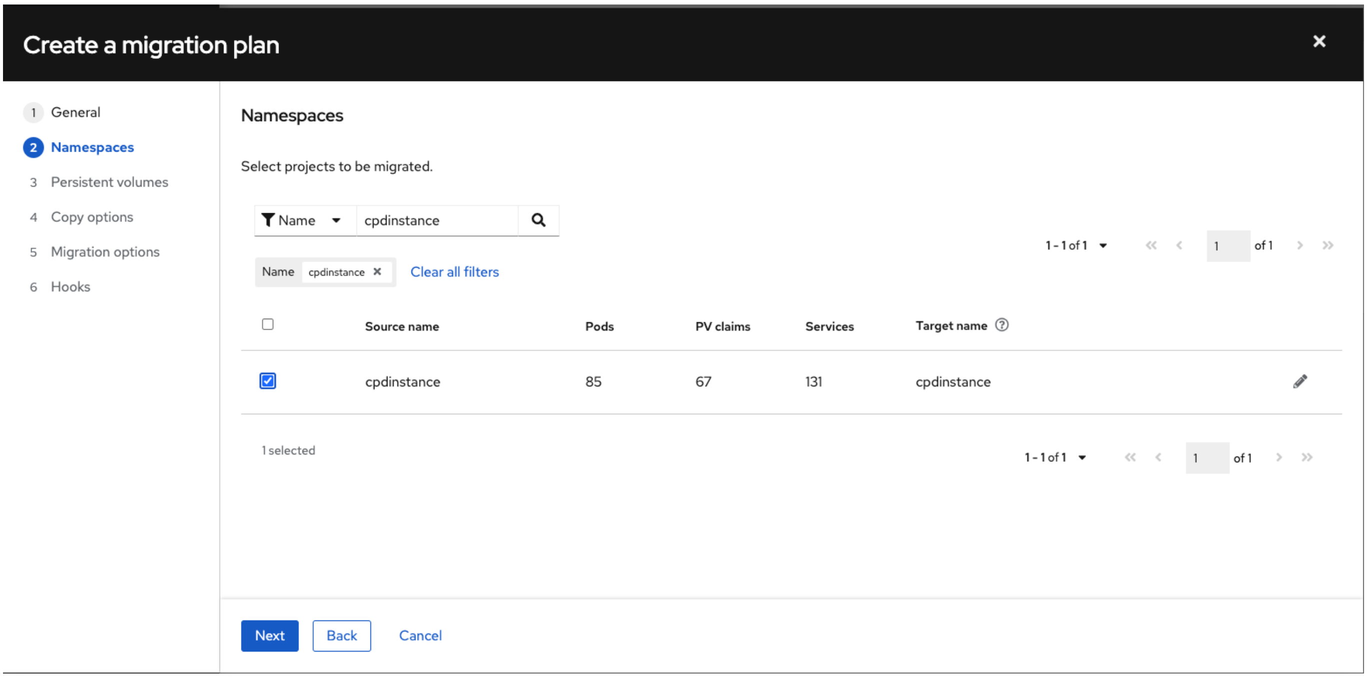
Migration type *: Full migration - migrate namespaces, persistent volumes (PVs) and Kubernetes resources from one cluster to another

Source cluster *: host

Target cluster *: target-wvyb

Repository *: ibmcos-mtc-todd-repo

Namespaces: Select your Cloud Pak for Data namespace.



Create a migration plan

Namespaces

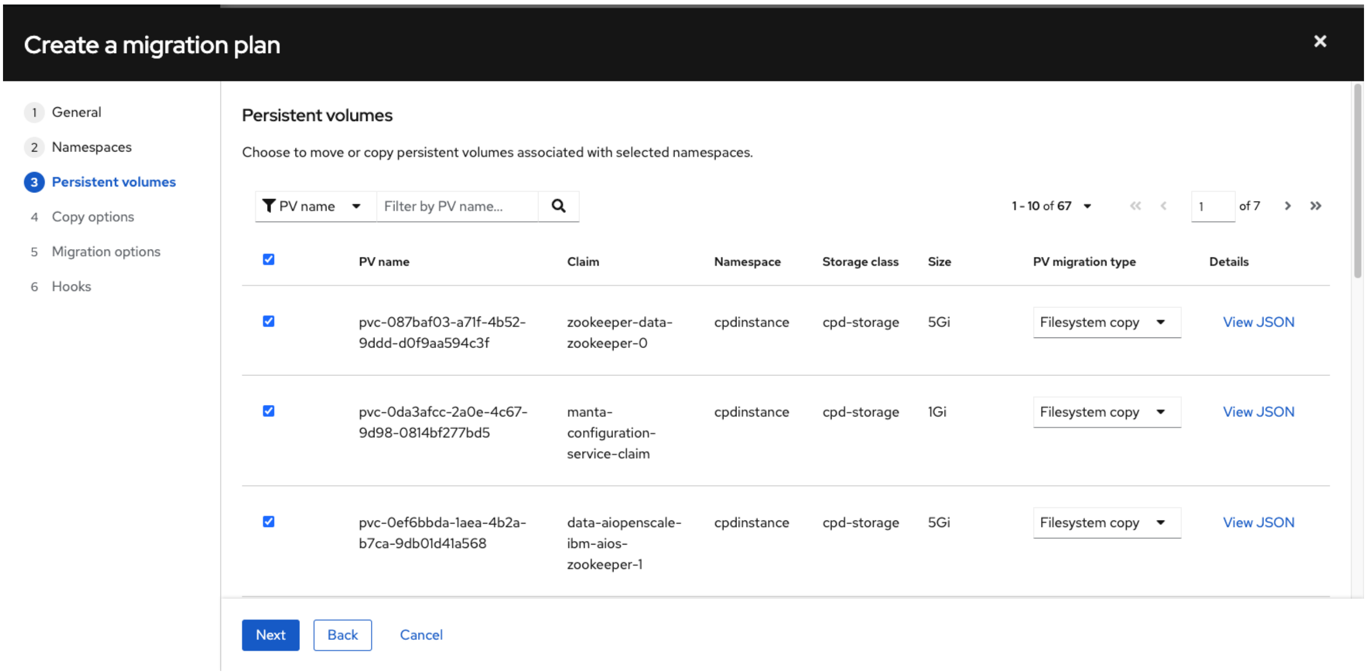
Select projects to be migrated.

Name: cpdinstance

□	Source name	Pods	PV claims	Services	Target name
☑	cpdinstance	85	67	131	cpdinstance

1 selected

Persistent volumes: For PVs, after persistent volumes are discovered, the MTC UI will have all PVs selected, with PV Migration Type as "Filesystem copy" by default. Leave the defaults and click "Next".



Create a migration plan

Persistent volumes

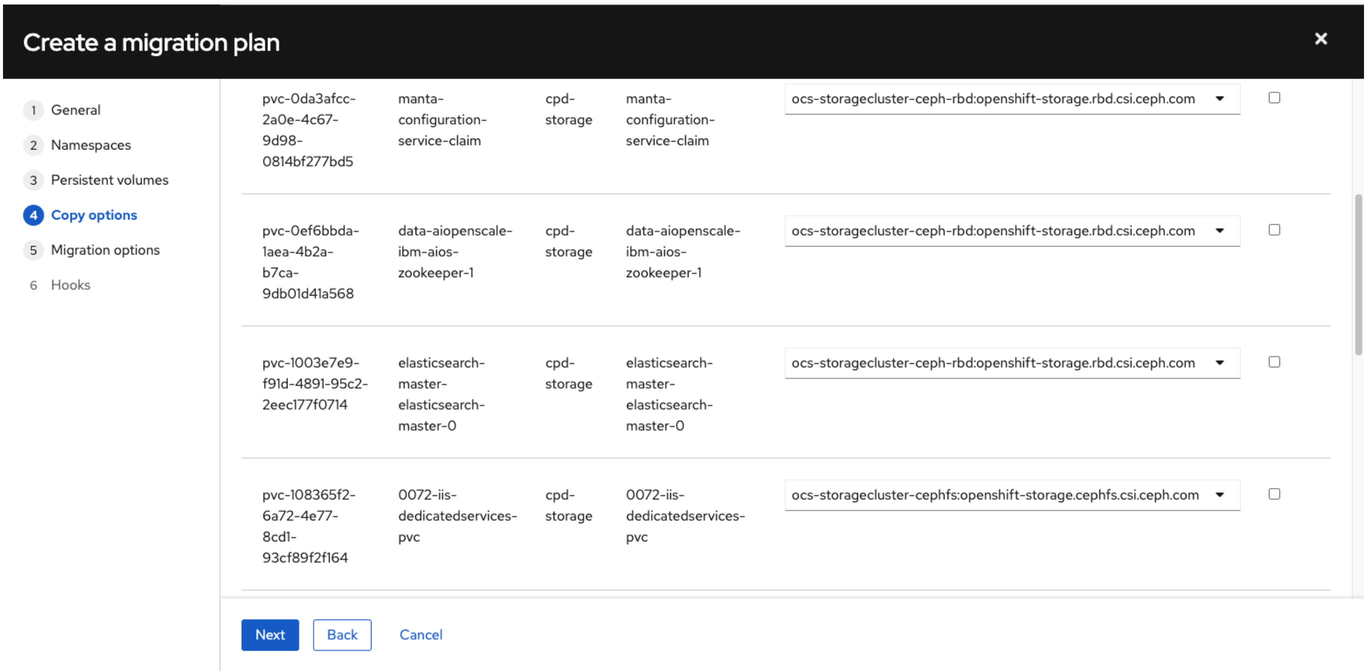
Choose to move or copy persistent volumes associated with selected namespaces.

☑	PV name	Claim	Namespace	Storage class	Size	PV migration type	Details
☑	pvc-087baf03-a71f-4b52-9ddd-d0f9aa594c3f	zookeeper-data-zookeeper-0	cpdinstance	cpd-storage	5Gi	Filesystem copy	View JSON
☑	pvc-0da3afcc-2a0e-4c67-9d98-0814bf277bd5	manta-configuration-service-claim	cpdinstance	cpd-storage	1Gi	Filesystem copy	View JSON
☑	pvc-0ef6bbda-1aea-4b2a-b7ca-9db01d41a568	data-aiopenscale-ibm-aios-zookeeper-1	cpdinstance	cpd-storage	5Gi	Filesystem copy	View JSON

Copy options: Leave all of the default values and click "Next" without making any changes. In general, you would want to select the "file" storage class for RWX volumes and the "block" storage class for RWO volumes.

```
For ODF, it will be:
    RWX - file sc: ocs-storagecluster-cephfs
    RWO - block sc: ocs-storagecluster-ceph-rbd
```

However, MTC detects if the PVC is RWX vs RWO and attempts to select the correct storage class on the target cluster. Example screenshot:

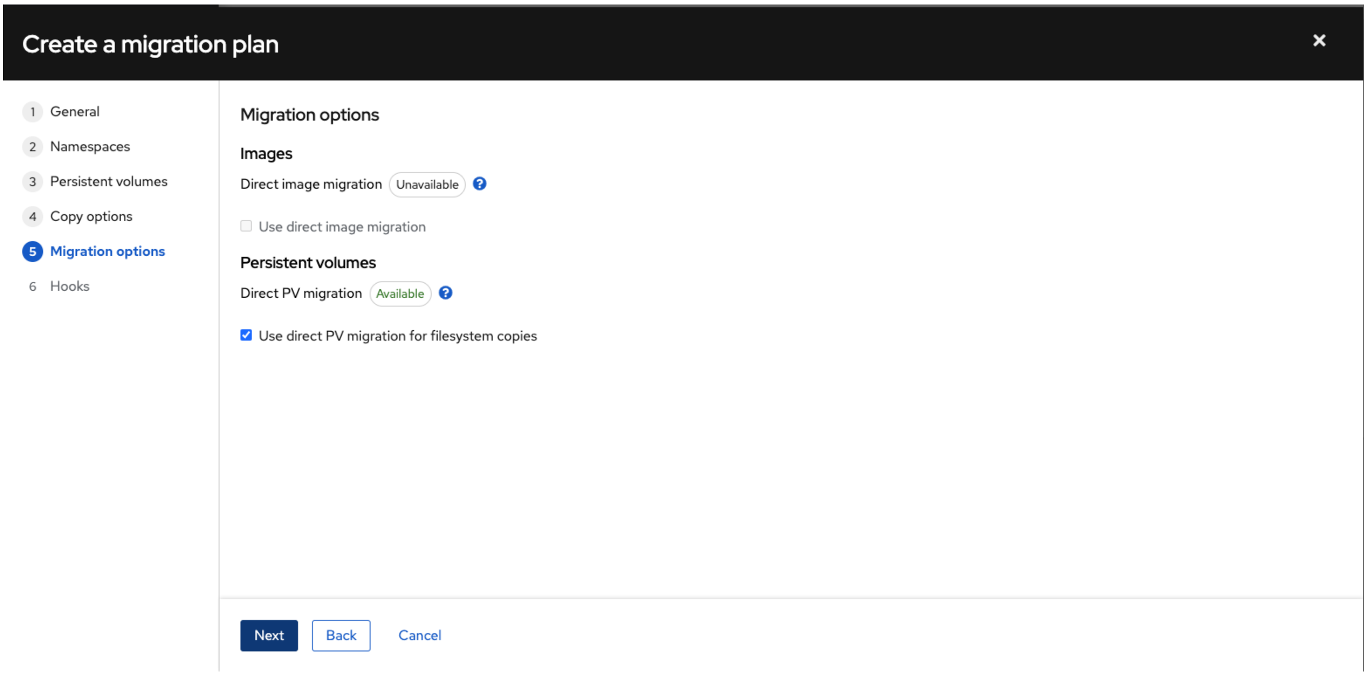


Create a migration plan

pvc-0da3afcc-2a0e-4c67-9d98-0814bf277bd5	manta-configuration-service-claim	cpd-storage	manta-configuration-service-claim	ocs-storagecluster-ceph-rbd:openshift-storage.rbd.csi.ceph.com	☐
pvc-0ef6bbda-1aea-4b2a-b7ca-9db01d41a568	data-aiopenscale-ibm-aios-zookeeper-1	cpd-storage	data-aiopenscale-ibm-aios-zookeeper-1	ocs-storagecluster-ceph-rbd:openshift-storage.rbd.csi.ceph.com	☐
pvc-1003e7e9-f91d-4891-95c2-2eec177f0714	elasticsearch-master-elasticsearch-master-0	cpd-storage	elasticsearch-master-elasticsearch-master-0	ocs-storagecluster-ceph-rbd:openshift-storage.rbd.csi.ceph.com	☐
pvc-108365f2-6a72-4e77-8cd1-93cf89f2f164	0072-iis-dedicatedservices-pvc	cpd-storage	0072-iis-dedicatedservices-pvc	ocs-storagecluster-cephfs:openshift-storage.cephfs.csi.ceph.com	☐

Migration options:

For Persistent volumes, the option "Use direct PV migration for filesystem copies" option has the following effect: If selected, then it will create an rsync server on the target cluster and perform rsync copies of the PV data from source to target. If not selected, then it will perform an indirect copy, where the data is pushed to s3 object storage location via velero/restic. Then pulled from the s3 object store to the PV. The direct method is faster, since it performs a direct copy, as long as the two clusters have network connectivity to one another.



Create a migration plan

1 General
2 Namespaces
3 Persistent volumes
4 Copy options
5 Migration options
6 Hooks

Migration options

Images

Direct image migration Unavailable

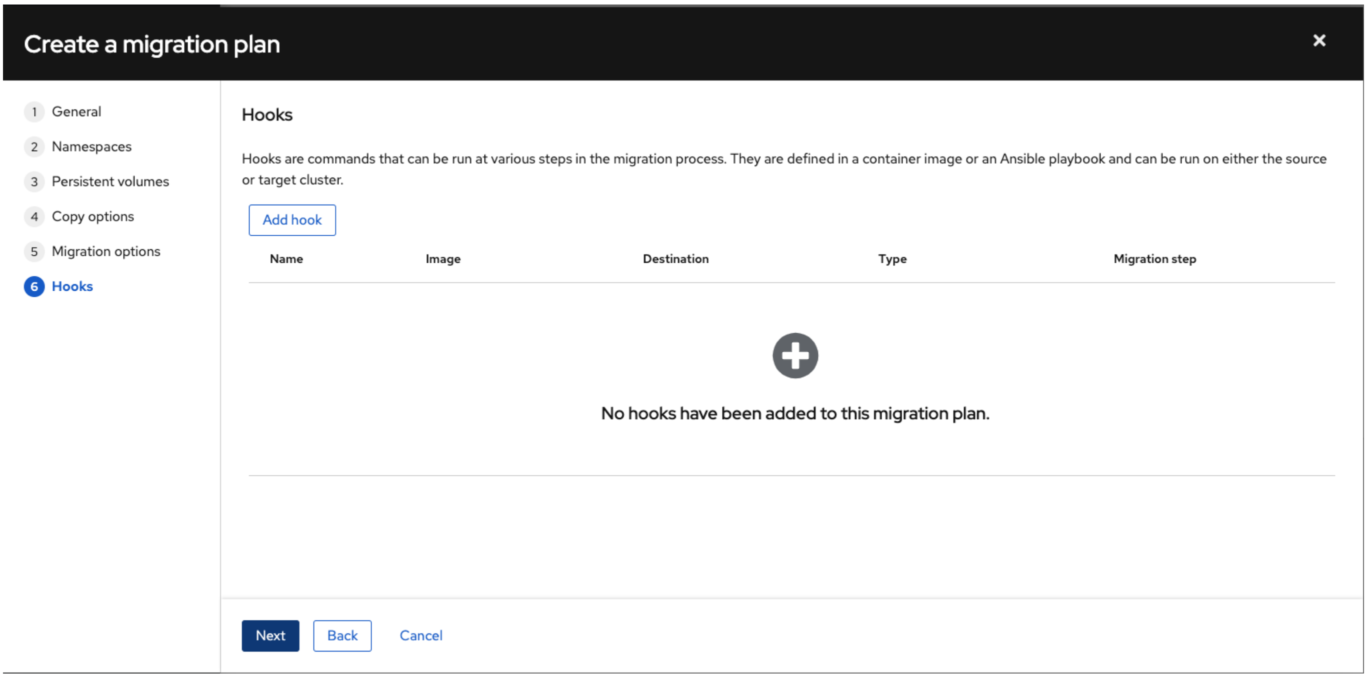
□ Use direct image migration

Persistent volumes

Direct PV migration Available

☑ Use direct PV migration for filesystem copies

Hooks: For CPD migration, no hooks are used. Click "Next".



Create a migration plan

1 General
2 Namespaces
3 Persistent volumes
4 Copy options
5 Migration options
6 Hooks

Hooks

Hooks are commands that can be run at various steps in the migration process. They are defined in a container image or an Ansible playbook and can be run on either the source or target cluster.

Name	Image	Destination	Type	Migration step

No hooks have been added to this migration plan.

Migration plan validation:

The migration plan validation might display the following known warning, which indicates that it cannot calculate how much capacity for each PVC has been used in the current state. So just make sure that you have not exceeded the capacity of any of the PVCs, otherwise the data transfer will fail. But as long as the applications honor the PVC limit size, this will not be a problem.

> Failed to compute PV resizing data for the following volumes. PV resizing will be disabled for these volumes and the migration may fail if the volumes are full or their requested and actual capacities differ in the source cluster.

Migration Plan PVC storage class validation

After the migration plan is created, you can see the migplan CR:

Example command:

```
$ oc get migplan -n openshift-migration
NAME                    READY   SOURCE   TARGET        STORAGE
AGE
cp4d-465-wvyb-to-9c9f   True    host     target-wvyb   ibmcos-mtc-todd-
repo   34m
$
```

Run the following command against the migration plan to see if the PVCs have the right matching of storage class to access mode: Command:

```
oc get migplan -n openshift-migration <migplan-name> -o json | jq -r
'.spec.persistentVolumes[] | "[PVC:" + .pvc.name + "][Access mode:" +
.pvc.accessModes[0] + "][New storage class:" + .selection.storageClass +
"]"'
```

Compare the output one by one to validate it is as expected.

Below are grep commands to help validate quickly:

- Validate that all RWX have cephfs storage class (you want no grep matches):

```
oc get migplan -n openshift-migration <migplan-name> -o json | jq -r
'.spec.persistentVolumes[] | "[PVC:" + .pvc.name + "][Access mode:" +
.pvc.accessModes[0] + "][New storage class:" + .selection.storageClass +
"]"' |grep ReadWriteMany |grep -v cephfs
```

- Validate that all RWO have ceph-rbd storage class (you want no grep matches):

```
oc get migplan -n openshift-migration <migplan-name> -o json | jq -r
'.spec.persistentVolumes[] | "[PVC:" + .pvc.name + "][Access mode:" +
.pvc.accessModes[0] + "][New storage class:" + .selection.storageClass +
"]"' |grep ReadWriteOnce |grep -v ceph-rbd
```

- Example output:

```
$ oc get migplan -n openshift-migration <migplan-name> -o json | jq -r
'.spec.persistentVolumes[] | "[PVC:" + .pvc.name + "][Access mode:" +
.pvc.accessModes[0] + "][New storage class:" + .selection.storageClass +
"]"' |grep ReadWriteMany |grep -v cephfs
$

$ oc get migplan -n openshift-migration <migplan-name> -o json | jq -r
'.spec.persistentVolumes[] | "[PVC:" + .pvc.name + "][Access mode:" +
.pvc.accessModes[0] + "][New storage class:" + .selection.storageClass +
"]"' |grep ReadWriteOnce |grep -v ceph-rbd
$
```

If the target storage classes for any of the PVCs do not match what you intended, then you will need to edit the Migration Plan and adjust the target storage classes. This can be accomplished by one of these methods:

- Go to the MTC console, Migration plans. Select the Migration Plan and edit.
- Edit the Migration Plan from cli: `oc edit migplan -n openshift-migration <migplan-name>`

Shutdown Cloud Pak for Data instance on source cluster

Set oadp prehooks pre-reqs

Reference: https://www.ibm.com/docs/en/cloud-paks/cp-data/4.6.x?topic=backup-prerequisite-tasks

DB2 instances:

Obtain all the db2uclusters in the namespace

```
oc -n <cpd-namespace> get db2ucluster
```

For each of the db2ucluster displayed in the command above execute the following command:

```
oc -n <cpd-instance> label db2ucluster <DB2UCLUSTER> db2u/cpdbr=db2u --
overwrite
```

Verify each db2ucluster has the label `db2u/cpdbr=db2u`:

```
oc -n <cpd-instance> get db2ucluster --show-labels
```

Scale down CPD services by running backup prehooks

Example command:

```
cpd-cli oadp backup prehooks --include-namespaces <cpd-namespace> --log-
level=debug --verbose
```

Check the state of the deployments and statefulsets in the CPD instance namespace by running the following commands:

```
oc get deploy -n <cpd-namespace>
oc get sts -n <cpd-namespace>
oc get pod -n <cpd-namespace> --no-headers |grep -v Completed |wc -l
```

Run backup prehooks:

```
cpd-cli oadp backup prehooks --include-namespaces <cpd-namespace> --log-
level=debug --verbose
```

Verify successful completion of backup prehooks by ensuring output for command above includes these messages:

```
configmap pre-backup hooks completed
backup prehooks command completed
```

After running the backup prehooks, check for the remaining deployments/statefulsets/pods running:

Example:

```
$ oc get deploy -n <cpd-namespace> |grep -v "0/0"
NAME                                         READY   UP-TO-DATE
AVAILABLE   AGE
ibm-nginx                                    2/2     2            2
16h
manta-admin-gui                              1/1     1            1
13h
manta-artemis                                1/1     1            1
13h
manta-configuration-service                  1/1     1            1
13h
manta-dataflow                               1/1     1            1
13h
manta-flow-agent                             1/1     1            1
13h
$
$ oc get sts -n <cpd-namespace> |grep -v "0/0"
```

```
NAME                             READY   AGE
c-db2oltp-iis-db2u               1/1     13h
c-db2oltp-wkc-db2u               1/1     13h
$
$ oc get pod -n <cpd-namespace> |grep -v Completed
NAME                                                      READY   STATUS
RESTARTS      AGE
c-db2oltp-iis-db2u-0                                      1/1     Running
0             13h
c-db2oltp-wkc-db2u-0                                      1/1     Running
0             13h
ibm-nginx-69dd4dc867-9vrdz                                1/1     Running
0             16h
ibm-nginx-69dd4dc867-lc7pf                                1/1     Running
0             16h
manta-admin-gui-5bcd558778-bq9mz                          1/1     Running
0             13h
manta-artemis-8f4b5d55f-xmjfk                             1/1     Running
0             13h
manta-configuration-service-5d87768b87-49bsw              1/1     Running
0             13h
manta-dataflow-56577f6f75-kxtss                           1/1     Running
0             13h
manta-flow-agent-5dfcb4864b-rfd2n                         1/1     Running
0             13h
wkc-foundationdb-cluster-cluster-controller-1             2/2     Running
0             14h
wkc-foundationdb-cluster-log-1                            2/2     Running
0             14h
wkc-foundationdb-cluster-log-2                            2/2     Running
0             14h
wkc-foundationdb-cluster-storage-1                        2/2     Running
0             14h
wkc-foundationdb-cluster-storage-2                        2/2     Running
1 (13h ago)   14h
wkc-foundationdb-cluster-storage-3                        2/2     Running
0             14h
$
```

Scale down remaining services and resources running in CPD namespace.

For the remaining pods running in the CPD instance namespace, they will have to be shutdown manually. The following is a list of known services/deployments/statefulsets that have to be shutdown manually, and the process to do so.

nginx:

```
oc get deploy/ibm-nginx -n <cpd-namespace>  #retain this output for the
resume
oc scale --replicas=0 deploy/ibm-nginx -n <cpd-namespace>
```

db2uclusters statefulsets:

```
oc get sts -n <cpd-namespace> |grep -E "NAME|db2"  #retain this output for the resume
oc scale --replicas=0 sts/c-db2oltp-iis-db2u -n <cpd-namespace>
oc scale --replicas=0 sts/c-db2oltp-wkc-db2u -n <cpd-namespace>

## Get sts for your db2wh instance
oc get sts -licpdsupport/addOnId=db2wh -n <cpd-namespace> #retain this output for the resume, you will see two sts (one for db2u and one for etcd)
oc scale --replicas=0 sts/<db2wh-db2u-sts> -n <cpd-namespace>
oc scale --replicas=0 sts/<db2wh-etc-sts> -n <cpd-namespace>
```

Note: There may be additional db2u related statefulsets that need to be scaled down. Scale them down similar to the above commands.

Foundationdb (FDB)

```
oc patch -p '{"spec":{"shutdown":"true"}}' --type=merge fdbcluster wkc-foundationdb-cluster -n <cpd-namespace>
```

MANTA (IBM internal reference: https://github.ibm.com/manta/manta-adl-operator#scaling)

```
oc patch -p '{"spec":{"replicas":0}}' --type=merge mantaflow mantaflow-wkc -n <cpd-namespace>
```

Openscale

Note: Even though the openscale deployments and statefulsets are shutdown during the backup prehooks, the service itself is not shutdown and not in maintenance mode. That means that it could be reconciled and started up again when not desired on the target cluster after the cpd operators are restored. To prevent this, we need to manually shut down the openscale service on the source cluster.

```
oc patch -p '{"spec":{"shutdown":"true"}}' --type=merge woservice aiopenscale -n <cpd-namespace>
oc patch -p '{"spec":{"ignoreForMaintenance": true}}' --type=merge woservice aiopenscale -n <cpd-namespace>
```

Datastage deploy and statefulset

```
## Record this output for the resume operations:
oc get deploy,sts -n <cpd-namespace>|grep ds-px-default
oc scale --replicas=0 deploy/ds-px-default-ibm-datastage-px-runtime -n
<cpd-namespace>
oc scale --replicas=0 sts/ds-px-default-ibm-datastage-px-compute -n <cpd-
namespace>
```

Example Output for scale down and shutdown commands above:

nginx

```
$ oc get deploy ibm-nginx -n cpd
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
ibm-nginx   2/2     2            2           16h
$ oc scale --replicas=0 deploy/ibm-nginx
deployment.apps/ibm-nginx scaled
$ oc get deploy ibm-nginx -n cpd
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
ibm-nginx   0/0     0            0           16h
$ oc get pod -n cpd |grep -v Completed |grep -E "NAME|nginx"
NAME                                                      READY   STATUS
RESTARTS      AGE
$
```

db2ucluster statefulsets

```
$ oc get db2ucluster -n cpd
NAME                     STATE   MAINTENANCESTATE   AGE
db2oltp-iis              Ready   InMaintenance      2d17h
db2oltp-wkc              Ready   InMaintenance      8d
db2wh-1686277435040922   Ready   InMaintenance      13h

$ oc get sts -n cpd |grep -v "0/0"
NAME                                     READY   AGE
c-db2oltp-iis-db2u                       1/1     2d17h
c-db2oltp-wkc-db2u                       1/1     8d
c-db2wh-1686277435040922-db2u            1/1     13h
c-db2wh-1686277435040922-etcd            1/1     13h
ds-px-default-ibm-datastage-px-compute   2/2     8d

$ oc  scale --replicas=0 sts/c-db2oltp-iis-db2u -n cpd
statefulset.apps/c-db2oltp-iis-db2u scaled
$ oc scale --replicas=0 sts/c-db2oltp-wkc-db2u -n cpd
statefulset.apps/c-db2oltp-wkc-db2u scaled
$ oc scale --replicas=0 sts/c-db2wh-1686277435040922-db2u -n cpd
statefulset.apps/c-db2wh-1686277435040922-db2u scaled
$ oc scale --replicas=0 sts/c-db2wh-1686277435040922-etcd -n cpd
statefulset.apps/c-db2wh-1686277435040922-etcd scaled
```

```
$ oc get sts -n cpd |grep -E "NAME|db2u"
NAME                              READY   AGE
c-db2oltp-iis-db2u                0/0     14h
c-db2oltp-wkc-db2u                0/0     14h
c-db2wh-1686277435040922-db2u            0/0     13h
c-db2wh-1686277435040922-etcd            0/0     13h

$ oc get pod -n cpd |grep -v Completed |grep -E "NAME|db2"
NAME                                                     READY   STATUS
RESTARTS      AGE
$
```

Foundationdb shutdown

```
$ oc describe fdbcluster wkc-foundationdb-cluster -n cpd |grep "
foundationdb.opencontent.ibm.com/backup-trigger"
              foundationdb.opencontent.ibm.com/backup-trigger: pre-backup
$
$ oc patch -p '{"spec":{"shutdown":"true"}}' --type=merge fdbcluster wkc-
foundationdb-cluster -n cpd
fdbcluster.foundationdb.opencontent.ibm.com/wkc-foundationdb-cluster
patched
$
$ oc get fdbcluster wkc-foundationdb-cluster -n cpd -o yaml |grep shutdown
  shutdown: "true"
  shutdownStatus: ' '
$

$ oc get pod -n cpd |grep -E "NAME|fdb|found"
NAME                                                     READY   STATUS
RESTARTS      AGE
wkc-foundationdb-cluster-cluster-controller-1            2/2
Terminating   0             14h
wkc-foundationdb-cluster-log-1                           2/2
Terminating   0             14h
wkc-foundationdb-cluster-log-2                           2/2
Terminating   0             14h
wkc-foundationdb-cluster-storage-1                       0/2
Terminating   0             14h
wkc-foundationdb-cluster-storage-2                       2/2
Terminating   1 (14h ago)   14h
wkc-foundationdb-cluster-storage-3                       2/2
Terminating   0             14h
$
$ oc get fdbcluster wkc-foundationdb-cluster -n cpd -o yaml |grep shutdown
  shutdown: "true"
  shutdownStatus: shutdown
$
$ oc get pod -n cpd |grep -E "NAME|fdb|found"
NAME                                                     READY   STATUS
RESTARTS   AGE
$
```

MANTA scaledown replica

```
$ oc get pod -n cpd |grep -v Completed
NAME                                                       READY   STATUS
RESTARTS   AGE
manta-admin-gui-5bcd558778-bq9mz                           1/1     Running
0          13h
manta-artemis-8f4b5d55f-xmjfk                              1/1     Running
0          13h
manta-configuration-service-5d87768b87-49bsw               1/1     Running
0          13h
manta-dataflow-56577f6f75-kxtss                            1/1     Running
0          13h
manta-flow-agent-5dfcb4864b-rfd2n                          1/1     Running
0          13h
$
$ oc get mantaflow -n cpd
NAME            AGE
mantaflow-wkc   13h
$ oc get mantaflow -n cpd mantaflow-wkc -o yaml |grep replicas
  replicas: 1
$
$ oc patch -p '{"spec":{"replicas":0}}' --type=merge mantaflow mantaflow-
wkc -n cpd
mantaflow.adl.getmanta.com/mantaflow-wkc patched
$ oc get mantaflow mantaflow-wkc -o yaml |grep replicas
  replicas: 0
$
$ oc get pod -n cpd |grep -E "NAME|manta"
NAME                                                       READY   STATUS
RESTARTS   AGE
$
```

Openscale shutdown:

Shutdown openscale:

```
$ oc -n cpd get woservice
NAME                        TYPE              STORAGE   SCALECONFIG
PHASE   RECONCILED   STATUS
aiopenscale                 service                     small
Ready   4.6.5        Completed
openscale-defaultinstance   serviceInstance             small
Ready   4.6.5        Completed
$
$ oc get woservice aiopenscale -o json |jq .spec.shutdown
"false"
$
```

```
$ oc patch -p '{"spec":{"shutdown":"true"}}' --type=merge woservice
aiopenscale -n cpd
woservice.wos.cpd.ibm.com/aiopenscale patched
$ oc -n cpd get woservice aiopenscale -o json |jq -y .spec.shutdown
'true'
$
```

Place it in maintenance mode:

```
$ oc patch -p '{"spec":{"ignoreForMaintenance": true}}' --type=merge
woservice aiopenscale -n cpd
$ oc get woservice
NAME                        TYPE              STORAGE   SCALECONFIG
PHASE          RECONCILED   STATUS
aiopenscale                 service                     small
InMaintenance   4.6.5        InMaintenance
openscale-defaultinstance   serviceInstance             small
shutdown        4.6.5        shutdown
$ oc get woservice aiopenscale -o yaml
apiVersion: wos.cpd.ibm.com/v1
kind: WOService
metadata:
  creationTimestamp: "2023-05-19T05:34:44Z"
  finalizers:
  - wos.cpd.ibm.com/finalizer
  generation: 5
  name: aiopenscale
  namespace: cpdinstance
  resourceVersion: "13630612"
  uid: 18596e99-8ffc-47f4-9068-a0d2d3fd8542
spec:
  acceptRollback: false
  blockStorageClass: cpd-storage
  fileStorageClass: cpd-storage
  ignoreForMaintenance: true
  license:
    accept: true
    license: Enterprise
  scaleConfig: small
  shutdown: "true"
  type: service
  version: 4.6.5
status:
  conditions:
  - ansibleResult:
      changed: 1
      completion: 2023-05-19T18:49:06.575661
      failures: 0
      ok: 2
      skipped: 0
    lastTransitionTime: "2023-05-19T18:49:01Z"
    message: Awaiting next reconciliation
    reason: Successful
    status: "True"
    type: Running
  - lastTransitionTime: "2023-05-19T18:49:06Z"
    message: Last reconciliation succeeded
    reason: Successful
    status: "True"
    type: Successful
  - lastTransitionTime: "2023-05-19T18:49:01Z"
    message: ""
    reason: ""
    status: "False"
    type: Failure
  phase: InMaintenance
  versions:
    reconciled: 4.6.5
  wosBuildNumber: "105"
  wosStatus: InMaintenance
$
```

Datastage

```
$ oc scale --replicas=0 deploy/ds-px-default-ibm-datastage-px-runtime -n
cpd
deploy.apps/ds-px-default-ibm-datastage-px-runtime scaled
$ oc scale --replicas=0 sts/ds-px-default-ibm-datastage-px-compute -n cpd
statefulset.apps/ds-px-default-ibm-datastage-px-compute scaled
```

Validate all CPD services and pods are now shutdown:

Ensure all pods in the CPD instance namespace are now shutdown:

```
$ oc get pod -n <cpd-namespace> |grep -v Completed
NAME                                                      READY   STATUS
RESTARTS   AGE
$
```

Ensure all CPD services are in maintenance mode:

```
for cpdcrd in $(oc get crd |grep cpd.ibm.com |awk '{print $1}');do echo --
- $cpdcrd ---;oc get $cpdcrd -o name |xargs -I{} sh -c "oc get {} -o yaml
|grep -i maintenance";done
```

Example:

```
$ for cpdcrd in $(oc get crd |grep cpd.ibm.com |awk '{print $1}');do echo
--- $cpdcrd ---;oc get $cpdcrd -o name |xargs -I{} sh -c "oc get {} -o
yaml |grep -i maintenance";done
--- analyticsengines.ae.cpd.ibm.com ---
  ignoreForMaintenance: true
  analyticsengineStatus: InMaintenance
--- ccs.ccs.cpd.ibm.com ---
  ignoreForMaintenance: true
  ccsStatus: InMaintenance
--- datarefinery.datarefinery.cpd.ibm.com ---
  ignoreForMaintenance: true
  datarefineryStatus: InMaintenance
--- datastages.ds.cpd.ibm.com ---
--- db2aaserviceservices.databases.cpd.ibm.com ---
  ignoreForMaintenance: true
  db2aaserviceStatus: InMaintenance
--- db2oltpservices.databases.cpd.ibm.com ---
  ignoreForMaintenance: true
  db2oltpStatus: InMaintenance
--- db2whservices.databases.cpd.ibm.com ---
  ignoreForMaintenance: true
  db2whStatus: InMaintenance
--- ibmcpds.cpd.ibm.com ---
--- iis.iis.cpd.ibm.com ---
  ignoreForMaintenance: true
  iisStatus: InMaintenance
--- notebookruntimes.ws.cpd.ibm.com ---
  ignoreForMaintenance: true
  runtimeStatus: InMaintenance
--- pxruntimes.ds.cpd.ibm.com ---
--- ug.wkc.cpd.ibm.com ---
  ignoreForMaintenance: true
  ugStatus: InMaintenance
--- wkc.wkc.cpd.ibm.com ---
  ignoreForMaintenance: true
  wkcStatus: InMaintenance
--- wmlbases.wml.cpd.ibm.com ---
  ignoreForMaintenance: true
  wmlStatus: InMaintenance
--- woservices.wos.cpd.ibm.com ---
  ignoreForMaintenance: true
  phase: InMaintenance
  wosStatus: InMaintenance
  ignoreForMaintenance: false
--- ws.ws.cpd.ibm.com ---
  ignoreForMaintenance: true
  wsStatus: InMaintenance
--- zenextensions.zen.cpd.ibm.com ---
--- zenservices.zen.cpd.ibm.com ---
  ignoreForMaintenance: true
  zenStatus: InMaintenance
```

MTC migration Stage to target cluster

The migration Stage will create the corresponding namespace on the target cluster. Then it will create the PVs and PVCs and copy the current data to the target cluster PVs, using whichever copy options were selected in the migration plan.

Note: The “Stage” process is concurrent and can be performed while the CPD services are running. No application resources will be scaled down or shutdown in this phase.

On the target cluster, verify that the cpd namespace to be migrated does not exist yet: Example command:

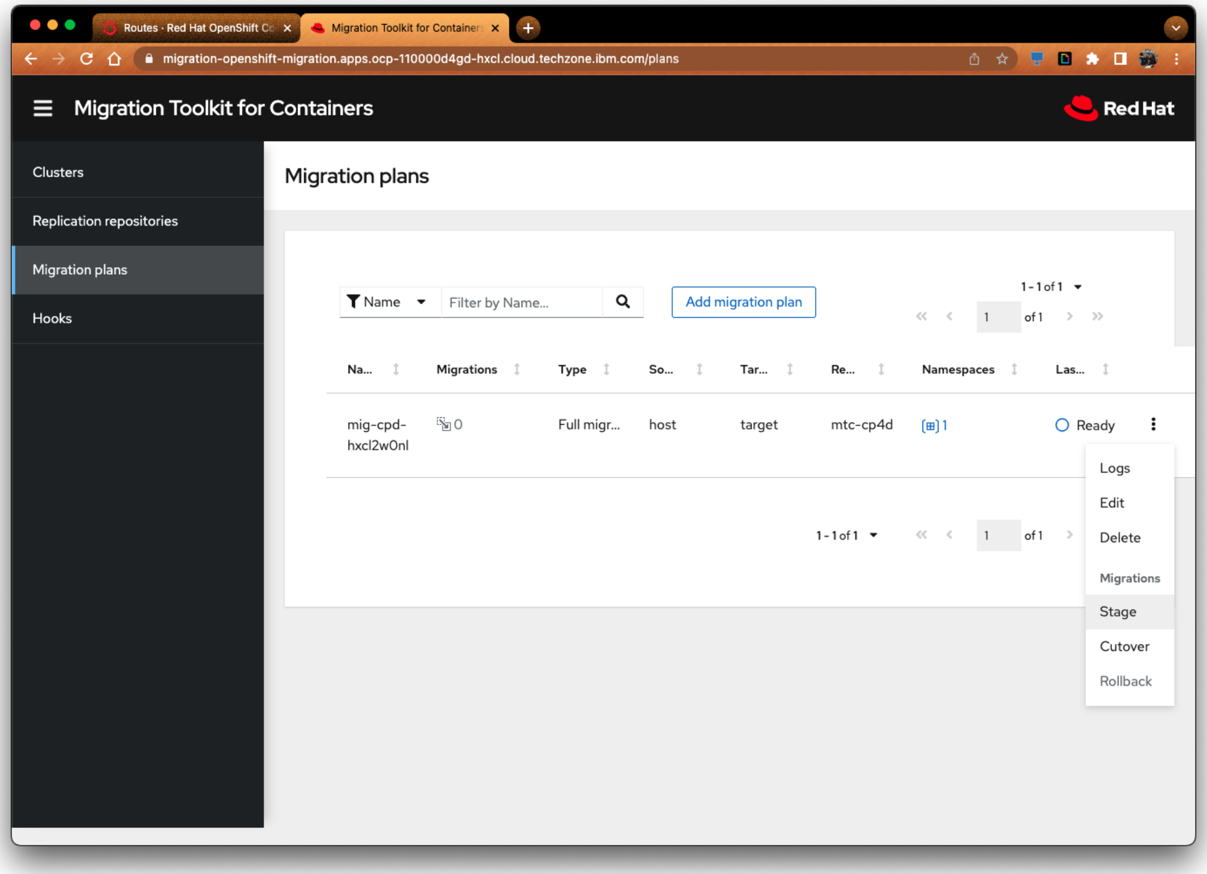
```
$ oc get ns |grep -v open
NAME                                               STATUS   AGE
default                                            Active   13d
ibm-common-services                                Active   5d21h
kube-node-lease                                    Active   13d
kube-public                                        Active   13d
kube-system                                        Active   13d
oadp-operator                                      Active   5d22h
turbo                                              Active   13d
```

Verify that the PVs do not exist on the target cluster:

```
$ oc get pv
NAME                                       CAPACITY   ACCESS MODES
RECLAIM POLICY   STATUS   CLAIM
STORAGECLASS                            REASON   AGE
local-pv-18e3632b                          700Gi      RWO
Delete           Bound    openshift-storage/ocs-deviceset-2-data-0sls6n
localblock                                       18d
local-pv-1f601e15                          700Gi      RWO
Delete           Bound    openshift-storage/ocs-deviceset-1-data-0hxdkm
localblock                                       18d
local-pv-d647fb26                          700Gi      RWO
Delete           Bound    openshift-storage/ocs-deviceset-0-data-099z84
localblock                                       18d
pvc-22cad882-f36a-41d1-8efb-e954859e86d8   1Gi        RWO
Delete           Bound    busybox-6y5e/busybox-pvc
ocs-storagecluster-cephfs-alt-cluster            18d
pvc-8c4d2bab-3815-42f8-bd23-004db1913711   50Gi       RWO
Delete           Bound    openshift-storage/db-noobaa-db-pg-0
ocs-storagecluster-ceph-rbd                      18d
pvc-bceec5e0-75bd-4e35-8652-b106cbf00cc1   5Gi        RWX
Delete           Bound    busybox-migrate/busybox-cephfs-pvc
ocs-storagecluster-cephfs-alt-cluster            10d
pvc-dd1054a8-f7a9-4500-82ef-4172eddbe0fe   1Gi        RWO
Delete           Bound    busybox-6y5e/busybox-pvc-new
ocs-storagecluster-cephfs-alt-cluster            18d
```

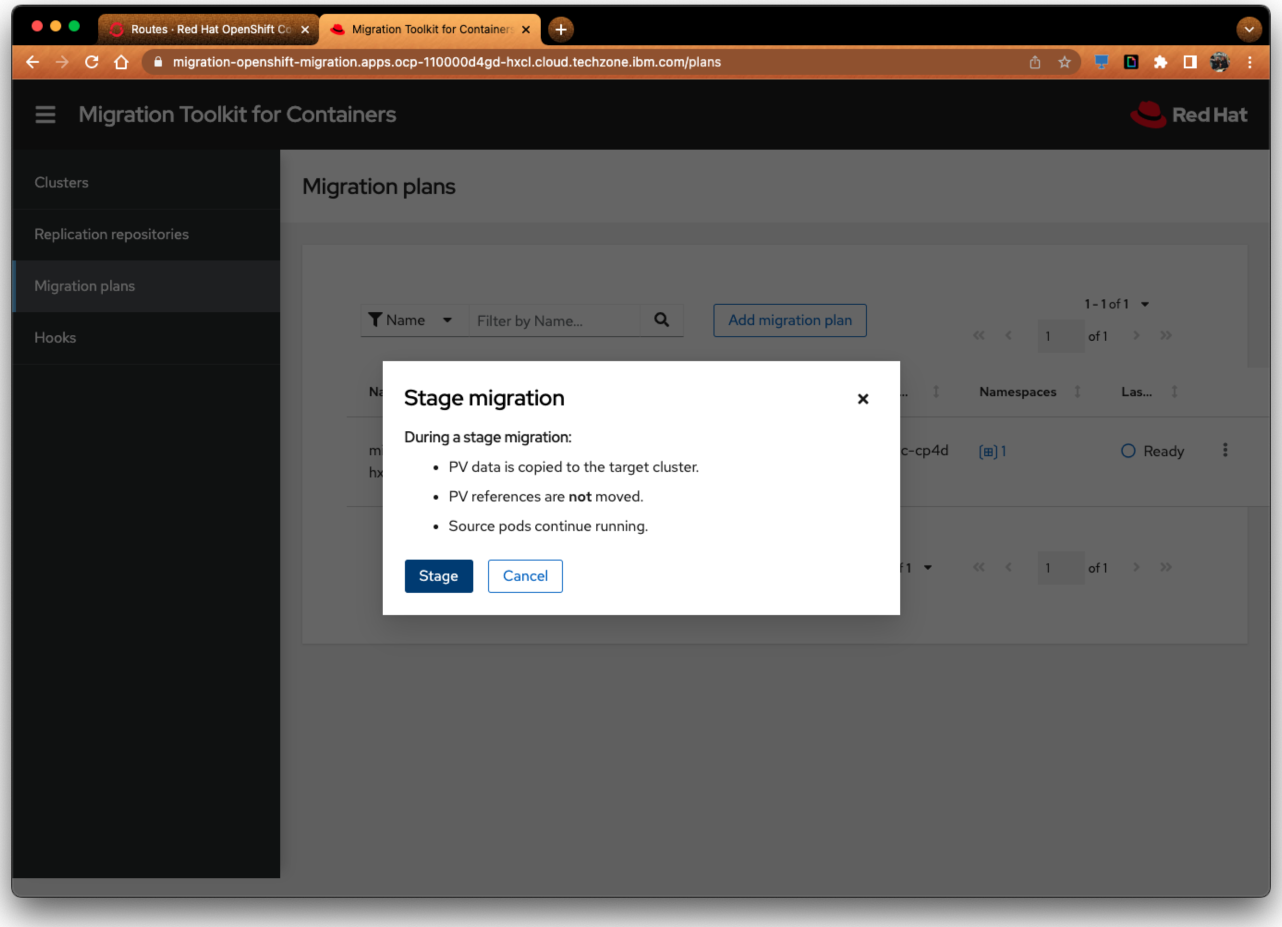
Perform Migration Stage:

- From MTC console on source cluster, go to "Migration plans". For the CPD migration plan, go to the three dots on the right side and select "Stage". This will create the PVs and PVCs in the cpd namespace on the target cluster.


Routes · Red Hat OpenShift Co
Migration Toolkit for Containers
migration-openshift-migration.apps.ocp-110000d4gd-hxcl.cloud.techzone.ibm.com/plans
Migration Toolkit for Containers
Red Hat
Clusters
Replication repositories
Migration plans
Hooks
Migration plans
Name
Filter by Name...
Add migration plan
1 - 1 of 1
1
of 1
Logs
Edit
Delete
Migrations
Stage
Cutover
Rollback

Na...	Migrations	Type	So...	Tar...	Re...	Namespaces	Las...
mig-cpd-hxcl2w0nl	0	Full migr...	host	target	mtc-cp4d	1	Ready

Click `Stage`

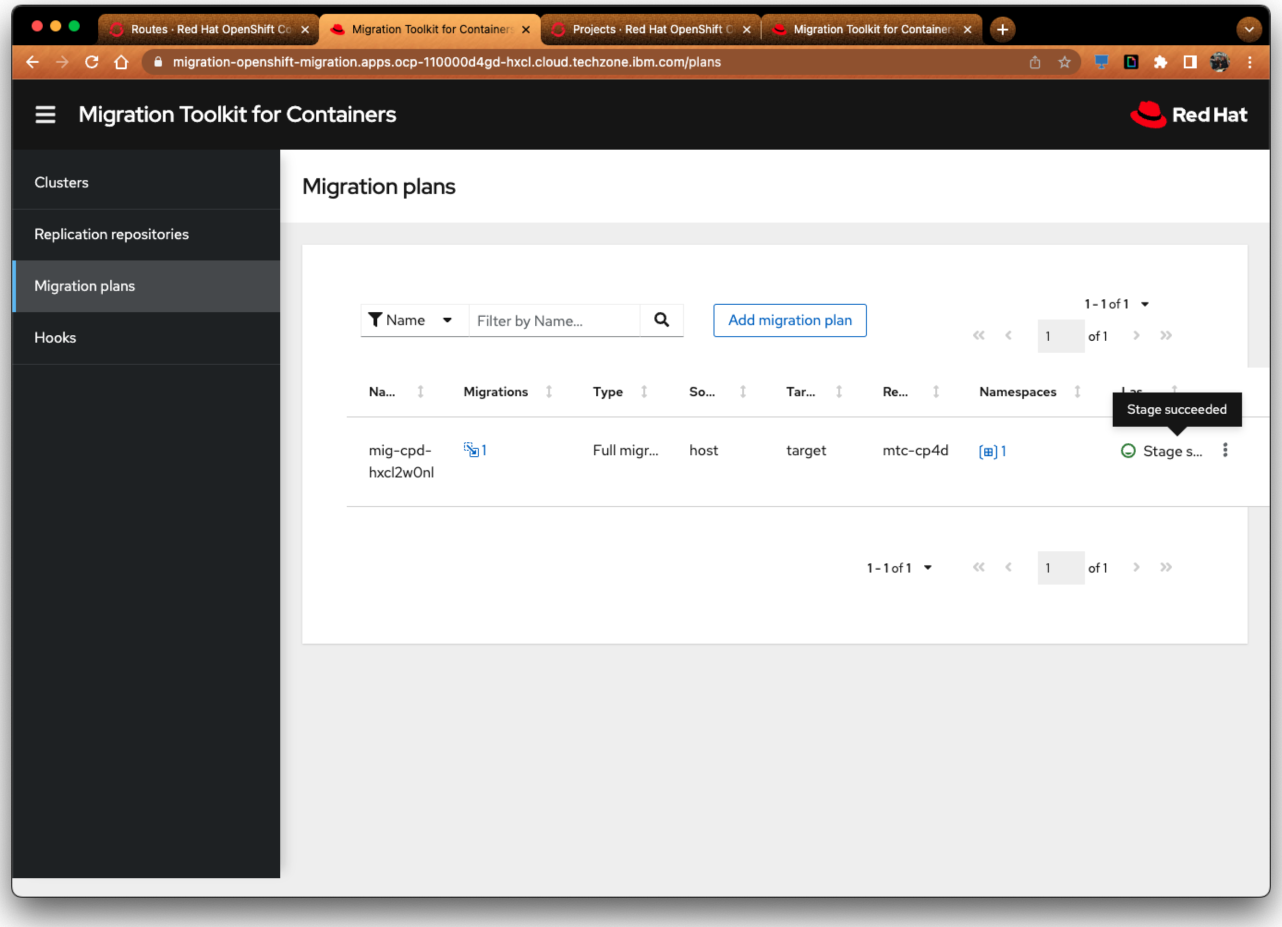


Stage migration

During a stage migration:

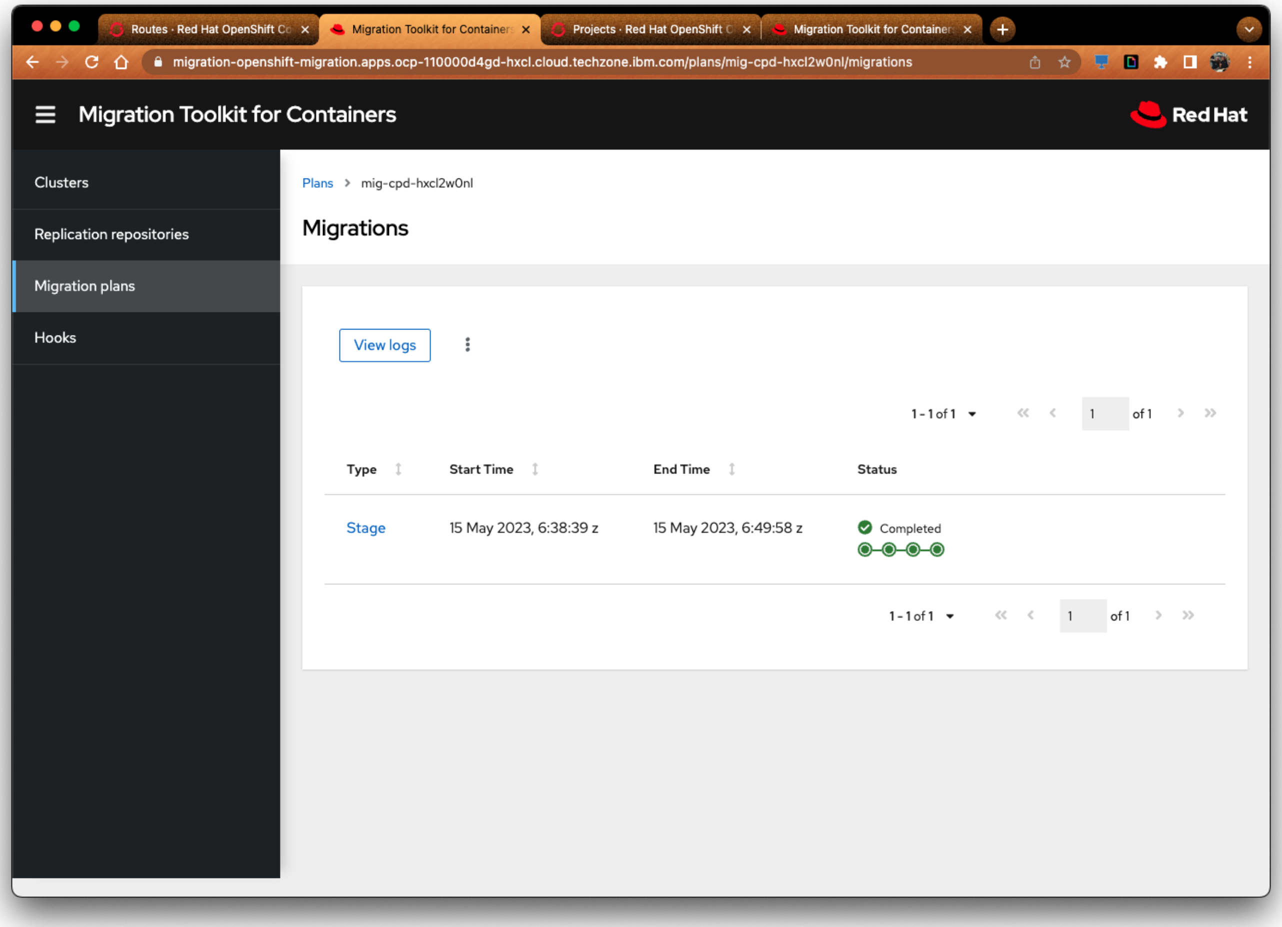
- PV data is copied to the target cluster.
- PV references are **not** moved.
- Source pods continue running.

Ensure Stage completes successfully with no warnings:


Routes · Red Hat OpenShift Co
Migration Toolkit for Containers
Projects · Red Hat OpenShift C
Migration Toolkit for Containers
migration-openshift-migration.apps.ocp-110000d4gd-hxcl.cloud.techzone.ibm.com/plans
Migration Toolkit for Containers
Red Hat
Clusters
Replication repositories
Migration plans
Hooks
Name
Filter by Name...
Add migration plan
1 - 1 of 1
1
of 1
Stage succeeded

Migration plans

Na...	Migrations	Type	So...	Tar...	Re...	Namespaces	Las...
mig-cpd-hxcl2wOnl	1	Full migr...	host	target	mtc-cp4d	1	Stage s...


Routes · Red Hat OpenShift Co
Migration Toolkit for Container
Projects · Red Hat OpenShift C
Migration Toolkit for Container
migration-openshift-migration.apps.ocp-110000d4gd-hxcl.cloud.techzone.ibm.com/plans/mig-cpd-hxcl2w0nl/migrations
Migration Toolkit for Containers
Red Hat
Clusters
Replication repositories
Migration plans
Hooks
Plans
mig-cpd-hxcl2w0nl
Migrations
View logs
1 - 1 of 1
of 1

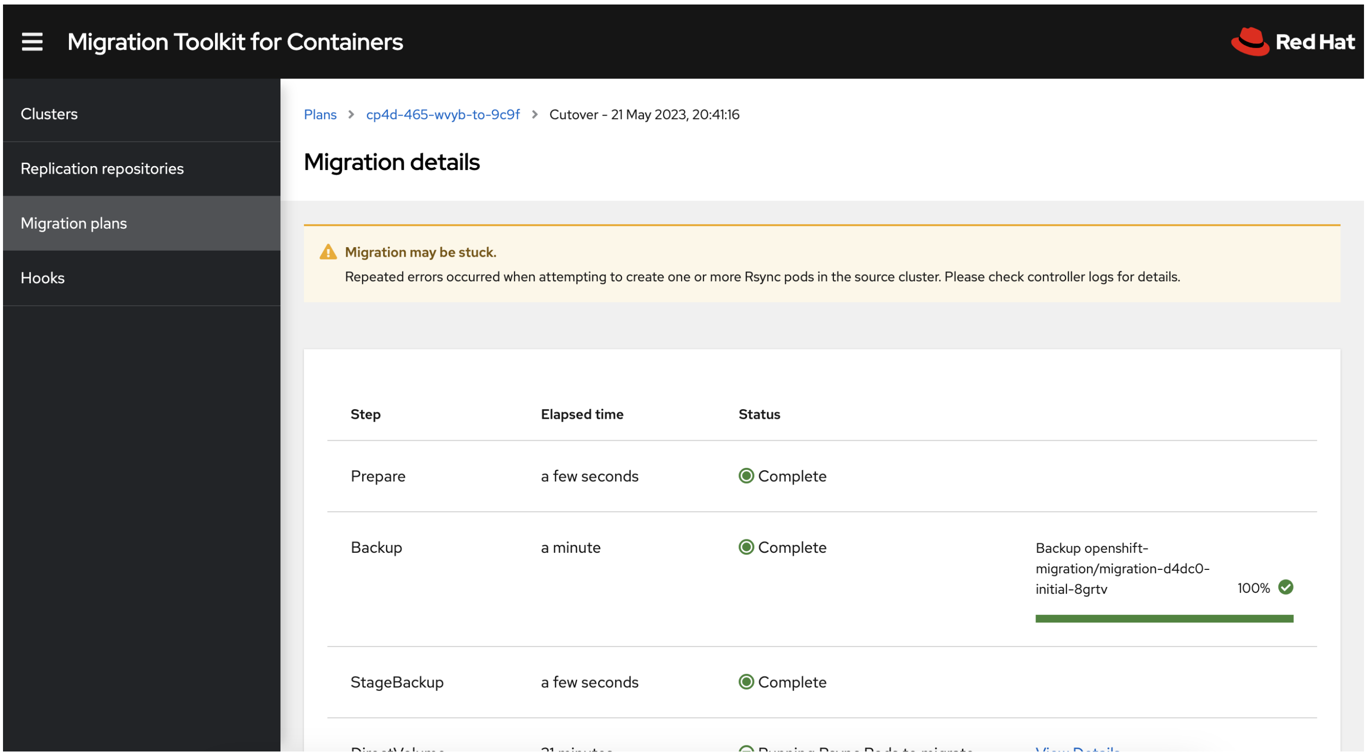
Type	Start Time	End Time	Status
Stage	15 May 2023, 6:38:39 z	15 May 2023, 6:49:58 z	Completed

During the Stage process, the `cpd` namespace, PVs and PVCs will be created and the `rsync-server` pod will be started.

Verify Cloud Pak for Data namespace was created.

```
oc get ns <cpd-namespace>
```

Note: During the Stage process, you may see an error indicating "Migration may be stuck." creating the rsync pods. Be patient in this phase, as the rsync pods will usually create successfully and the migration activity will complete.



Plans > cp4d-465-wvyb-to-9c9f > Cutover - 21 May 2023, 20:41:16

Migration details

Migration may be stuck.
Repeated errors occurred when attempting to create one or more Rsync pods in the source cluster. Please check controller logs for details.

Step	Elapsed time	Status	
Prepare	a few seconds	Complete	
Backup	a minute	Complete	Backup openshift-migration/migration-d4dc0-initial-8grtv 100%
StageBackup	a few seconds	Complete	
DirectVolume	21 minutes	Running Rsync Pods to migrate	View Details

If the Stage process fails, check velero status of the failed velero "migration" job. Example: `velero describe restore migration-xxx-final-xxx -n openshift-migration`

Target cluster migration steps

Verify PVs are created on target cluster and verify PV total matches original total before migration.

```
oc get pv |grep <cpd-namespace>
oc get pv |grep <cpd-namespace> | wc -l
```

Verify PVCs are created on target cluster.

```
oc get pvc -n <cpd-namespace>
```

Verify `rsync-server` started by running the command below. Note: The rsync server on the target cluster will be removed when the PV rsync actions are complete.

```
$ oc get pods -n <cpd-namespace>| grep rysnc
NAME           READY   STATUS    RESTARTS   AGE
rsync-server   2/2     Running   0          30s
```

Additionally, `rsync` pods will start and complete on the source cluster. Example output:

```
$ oc get pod -n <cpd-namespace> | grep rsync
rsync-22qck                                                  0/2
Completed          0          35s
```

```
rsync-2hh4r                                               0/2
Completed           0          2m13s
rsync-4f486                                               0/2
Completed           0          43s
rsync-7hwrf                                               0/2
Completed           0          96s
rsync-7xhh2                                               0/2
Completed           0          87s
rsync-8wbwv                                               0/2
Completed           0          2m4s
rsync-jb5gj                                               0/2
Completed           0          10s
rsync-m7ks6                                               0/2
Completed           0          77s
rsync-mnp7t                                               0/2
Completed           0          26s
rsync-pdsbn                                               0/2
Completed           0          114s
rsync-q9lcq                                               0/2
ContainerCreating   0          3s
rsync-rmbmm                                               0/2
Completed           0          19s
rsync-rn7zm                                               0/2
Completed           0          60s
rsync-tcdjf                                               0/2
Completed           0          105s
rsync-vbzfz                                               0/2
Completed           0          51s
rsync-xmm5k                                               0/2
Completed           0          69s
```

MTC Migration Cutover to target cluster

Before cutover, validate again that no pods are running in source cluster and target cluster. This is needed to ensure that everything is stopped on source cluster for the cutover, so when the cutover is performed, MTC does not attempt to resume any operations. The services will be resumed manually for CPD in later steps.

Source cluster example:

```
$ oc get pod -n <cpd-namespace> |grep -v Completed
NAME                                                      READY   STATUS
RESTARTS   AGE
$
```

Target cluster example:

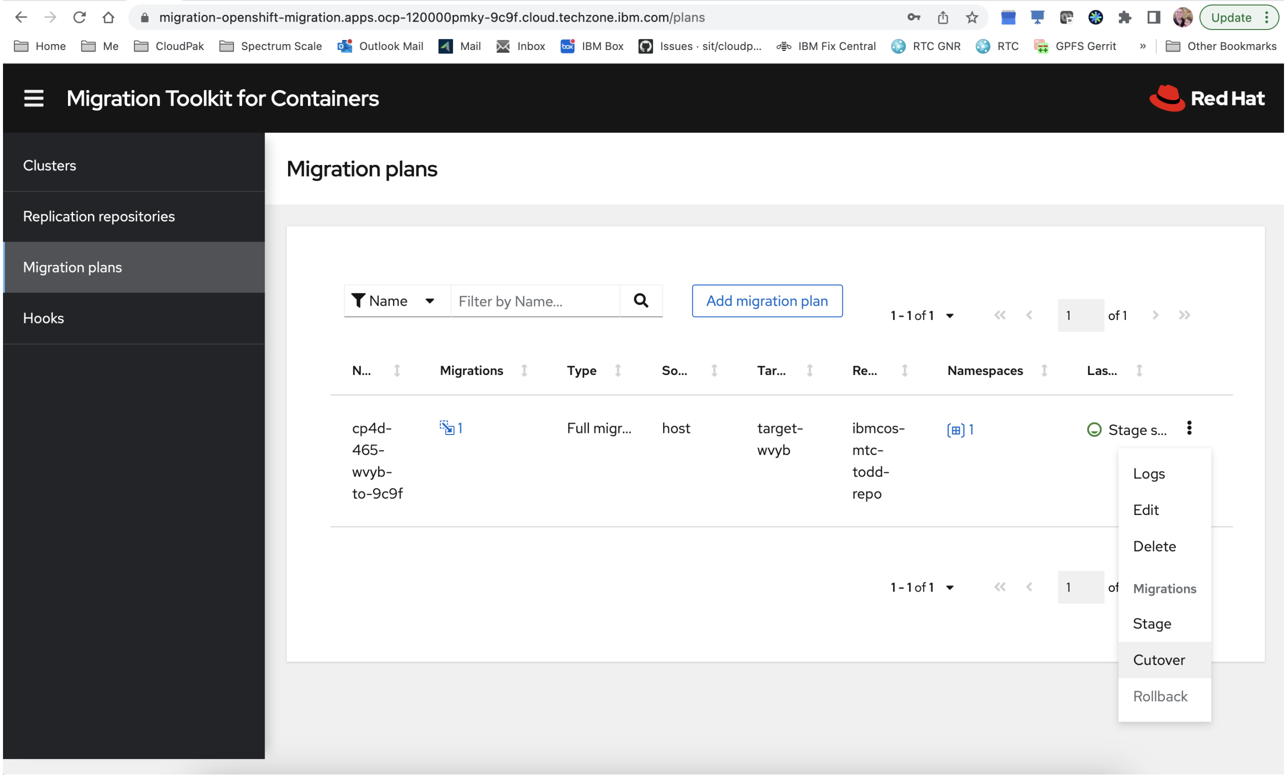
```
$ oc get pod -n <cpd-namespace>
No resources found in cpdinstance namespace.
$
```

Perform the MTC Cutover from source cluster MTC UI:

Gather target cluster resources before cutover by running the following command:

```
$ NSPACE="<cpd-namespace>";oc get $(oc api-resources --namespaced=true --
verbs=list -o name | awk '{printf "%s%s",sep,$0;sep=","}')  --ignore-not-
found -n ${NSPACE} |grep -v packagemanifest
```

- From Migration Controller, go to "Migration plans". For the CPD migration plan, select the three dots on right hand side and select "Cutover".

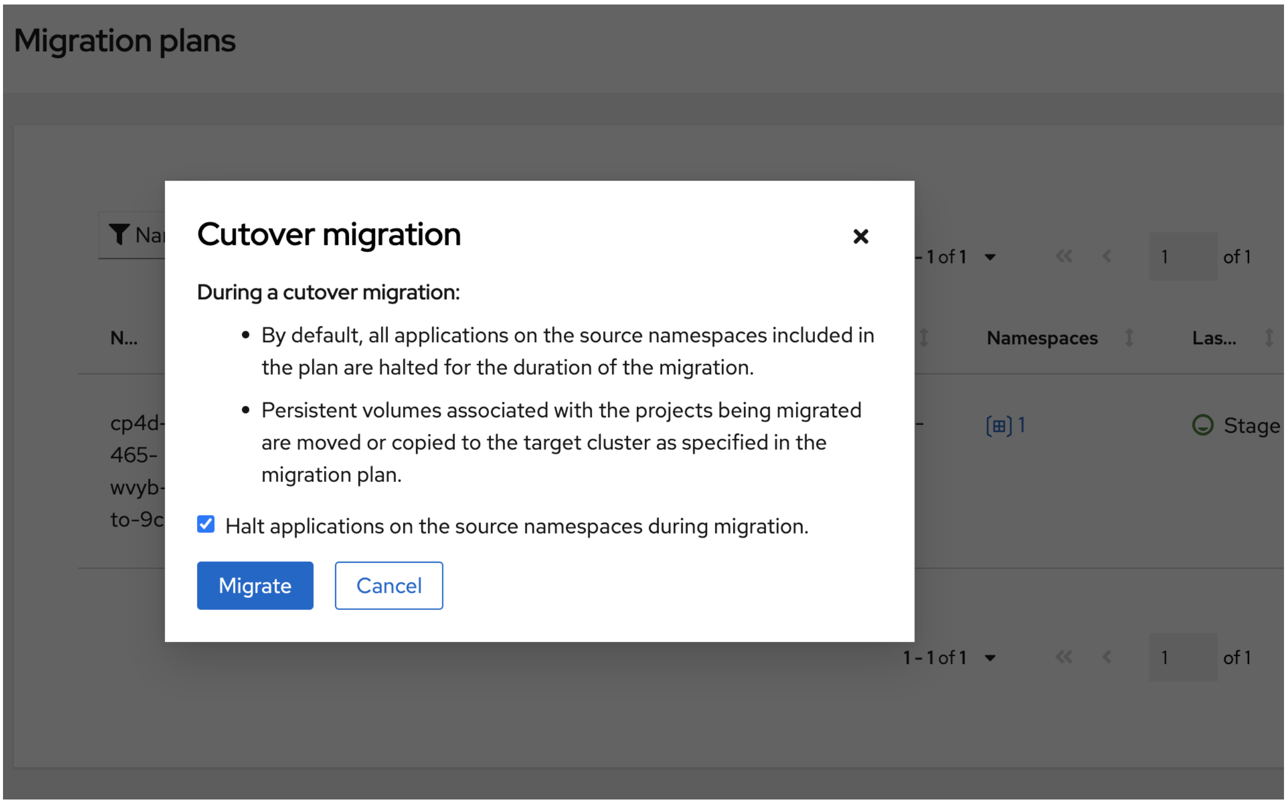


migration-openshift-migration.apps.ocp-120000pmky-9c9f.cloud.techzone.ibm.com/plans

Migration plans

N...	Migrations	Type	So...	Tar...	Re...	Namespaces	Las...
cp4d-465-wvyb-to-9c9f	1	Full migr...	host	target-wvyb	ibmcos-mtc-todd-repo	1	Stage s...

- A warning is displayed that applications will be halted. But this is not an issue, since we manually halted all applications prior to cutover. Make sure `Halt Applications...` option is selected and click `Migrate`



Migration plans

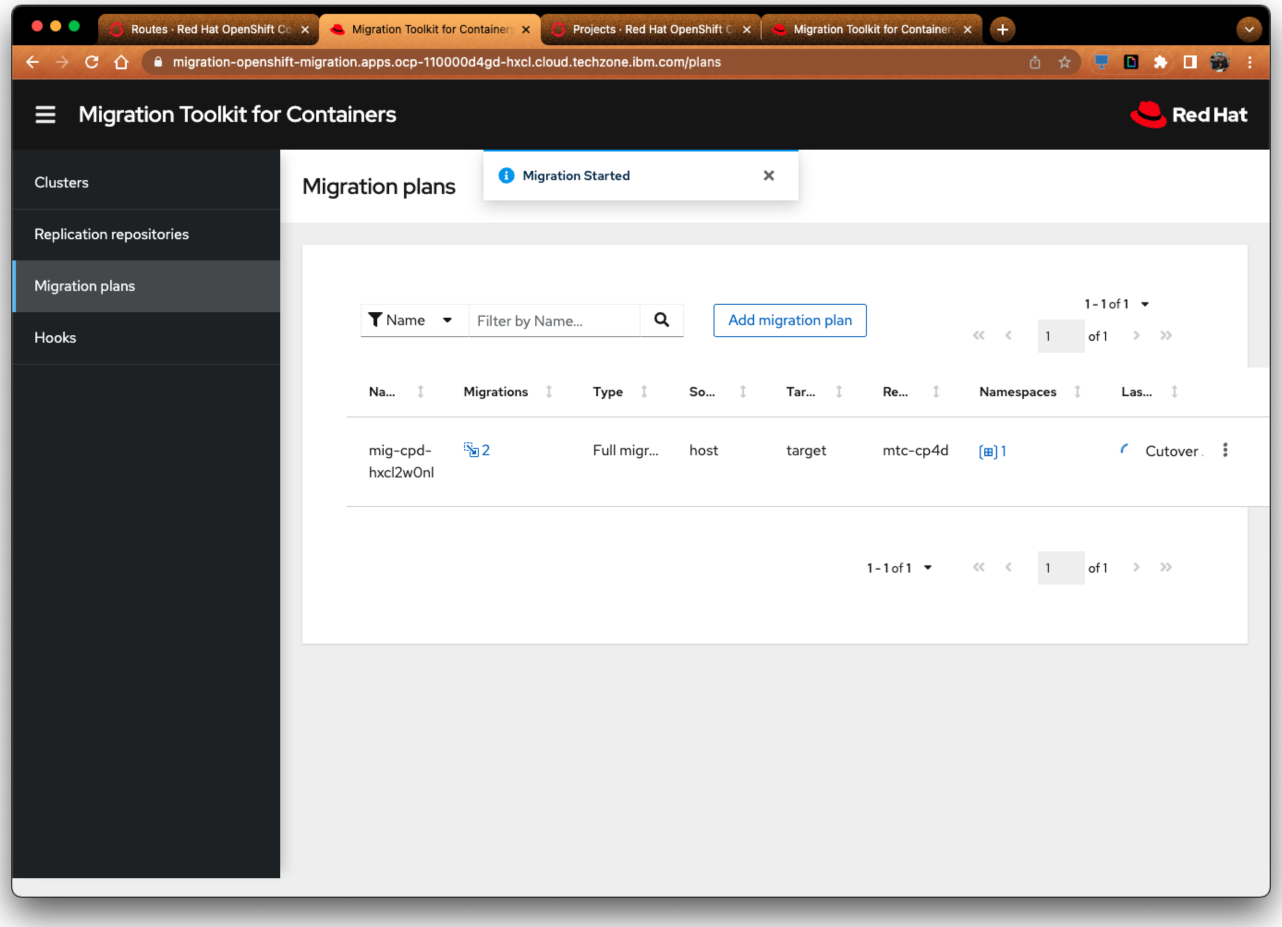
Cutover migration

During a cutover migration:

- By default, all applications on the source namespaces included in the plan are halted for the duration of the migration.
- Persistent volumes associated with the projects being migrated are moved or copied to the target cluster as specified in the migration plan.

☑ Halt applications on the source namespaces during migration.

Migration cutover started

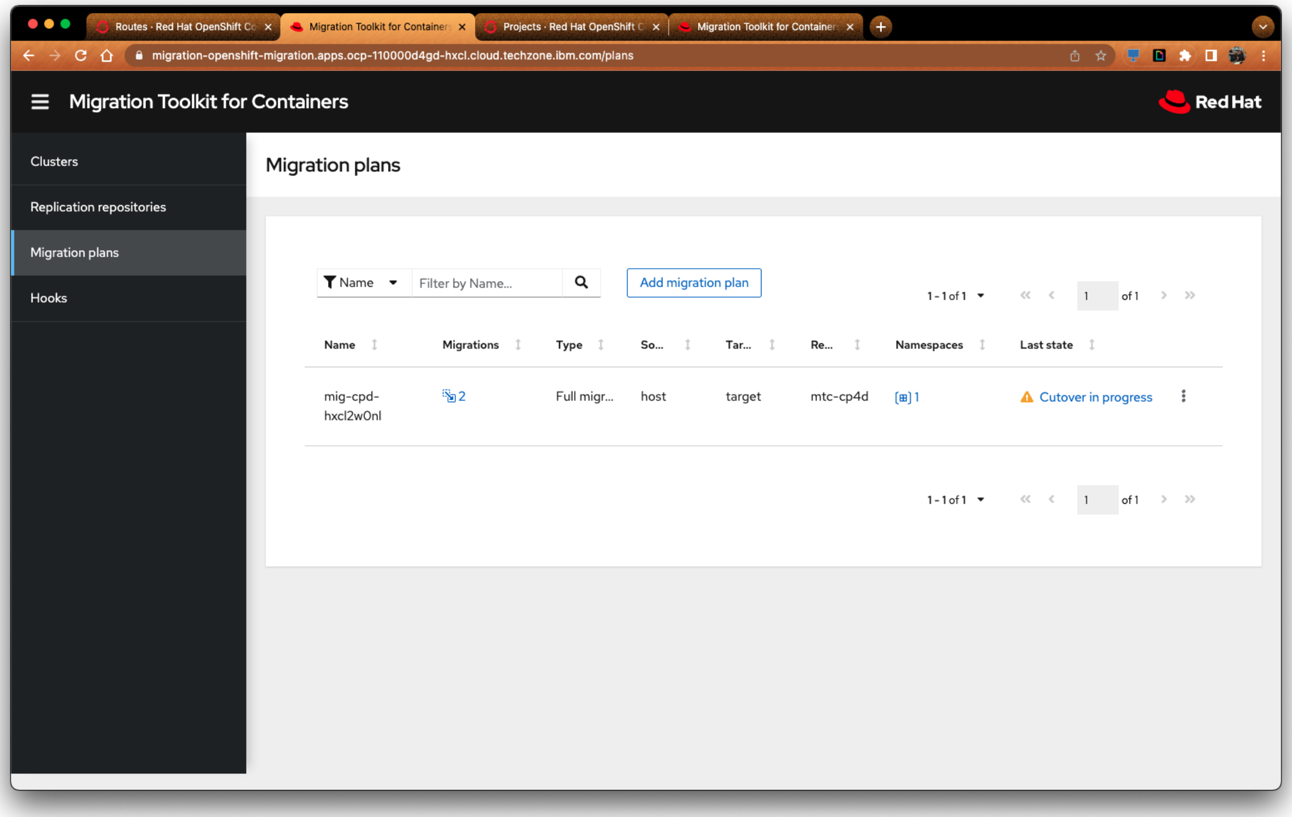


migration-openshift-migration.apps.ocp-110000d4gd-hxcl.cloud.techzone.ibm.com/plans

Migration plans

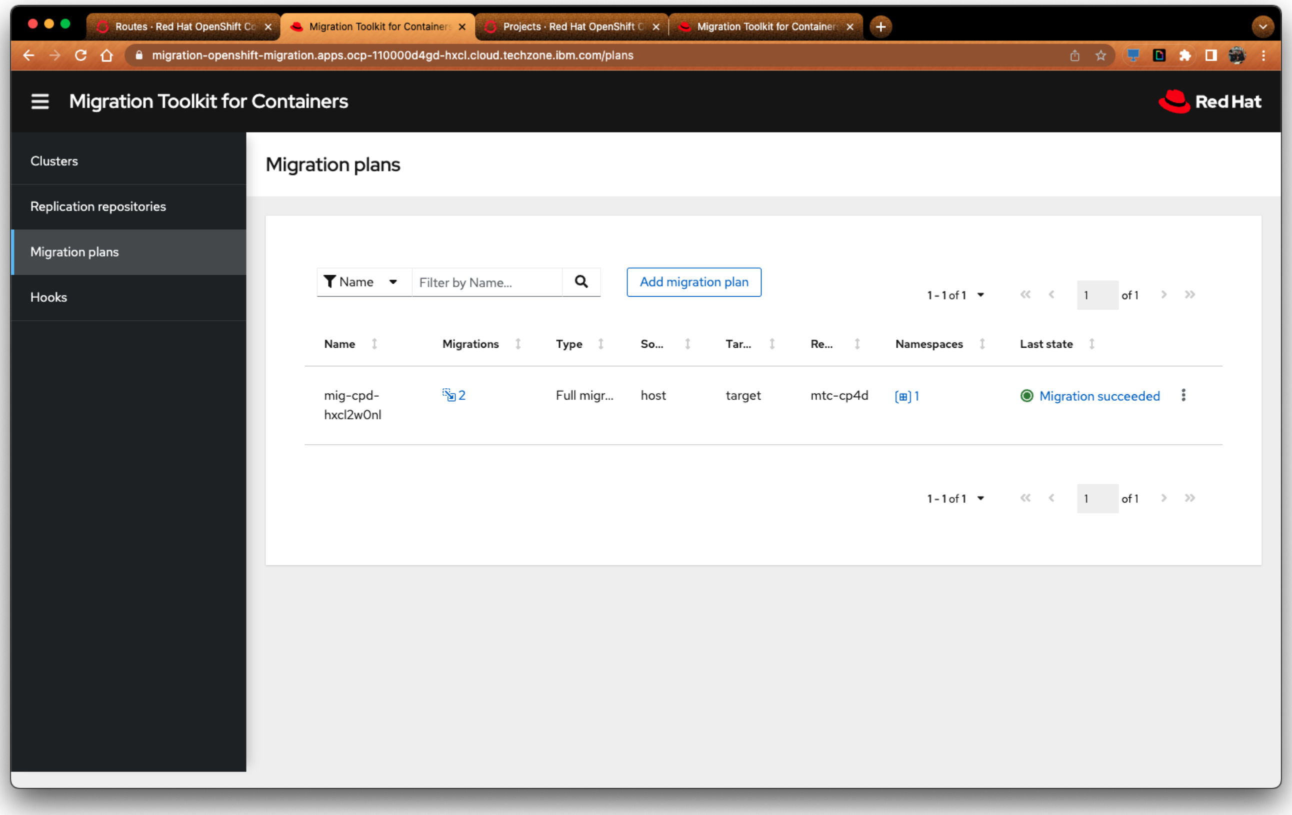
Migration Started

Na...	Migrations	Type	So...	Tar...	Re...	Namespaces	Las...
mig-cpd-hxcl2w0nl	2	Full migr...	host	target	mtc-cp4d	1	Cutover


Routes · Red Hat OpenShift Co
Migration Toolkit for Containers
Projects · Red Hat OpenShift C
Migration Toolkit for Containers
migration-openshift-migration.apps.ocp-110000d4gd-hxcl.cloud.techzone.ibm.com/plans
Migration Toolkit for Containers
Red Hat
Clusters
Replication repositories
Migration plans
Hooks
Migration plans
Name
Filter by Name...
Add migration plan
1 - 1 of 1
1
of 1

Name	Migrations	Type	So...	Tar...	Re...	Namespaces	Last state
mig-cpd-hxcl2w0nl	2	Full migr...	host	target	mtc-cp4d	1	⚠ Cutover in progress

Migration completed:


Routes · Red Hat OpenShift Co
Migration Toolkit for Containers
Projects · Red Hat OpenShift C
Migration Toolkit for Containers
migration-openshift-migration.apps.ocp-110000d4gd-hxcl.cloud.techzone.ibm.com/plans
Migration Toolkit for Containers
Red Hat
Clusters
Replication repositories
Migration plans
Hooks
Migration plans
Name
Filter by Name...
Add migration plan
1 - 1 of 1
1
of 1

Name	Migrations	Type	So...	Tar...	Re...	Namespaces	Last state
mig-cpd-hxcl2w0nl	2	Full migr...	host	target	mtc-cp4d	1	● Migration succeeded

During the migration cutover, an `rsync-server` pod will be created on the target cluster and `rsync` pods will start on the source cluster (similar to migration stage).

rsync-server pod on target cluster

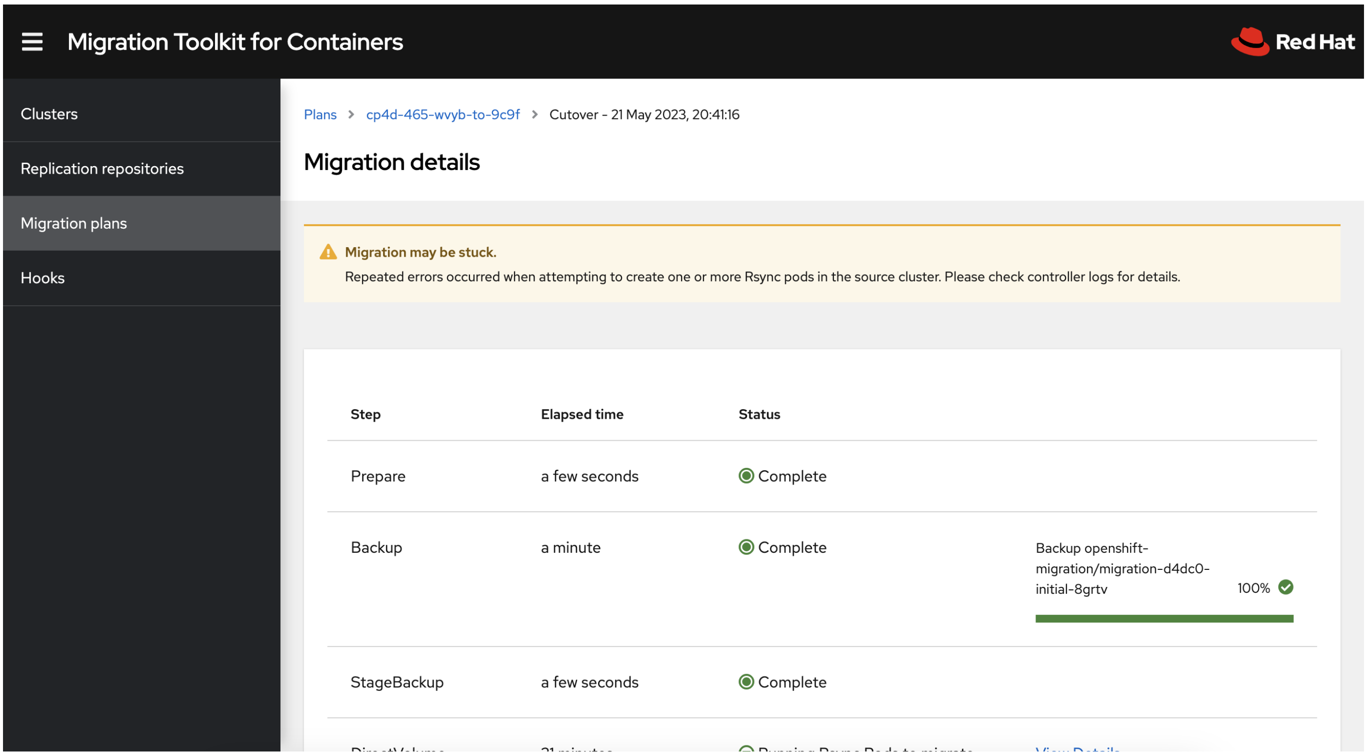
```
$ oc get pod -n <cpd-namespace> | grep rsync
NAME           READY   STATUS    RESTARTS   AGE
rsync-server   2/2     Running   0          8m16s
```

rsync pods on source cluster

```
$ oc get pod -n <cpd-namespace> | grep rsync
rsync-4q5gc                                               0/2
Completed   0          89s
rsync-4rbsk                                               0/2
Completed   0          32s
rsync-6vs6g                                               0/2
Completed   0          3m10s
rsync-9jjt7                                               0/2
Completed   0          2m9s
rsync-bf6tl                                               0/2
Completed   0          4m14s
rsync-c547g                                               0/2
Completed   0          3m52s
rsync-cgwqh                                               0/2
Completed   0          109s
rsync-dkhxx                                               0/2
Completed   0          2m49s
rsync-hvd5m                                               0/2
Completed   0          7m45s
rsync-lbxvl                                               0/2
Completed   0          4m36s
rsync-lzdcn                                               0/2
Completed   0          6m32s
rsync-m2g2p                                               0/2
Completed   0          8m2s
rsync-m7wm4                                               0/2
Completed   0          6m56s
rsync-ncbwl                                               0/2
Completed   0          3m31s
rsync-nfn7f                                               0/2
Completed   0          4m59s
rsync-pprkq                                               0/2
Completed   0          2m29s
rsync-qzkrm                                               0/2
Completed   0          6m8s
rsync-rdc54                                               0/2
Completed   0          51s
rsync-rm9mh                                               0/2
Completed   0          5m45s
rsync-sdzck                                               0/2
Completed   0          7m20s
rsync-tw8j7                                               0/2
```

```
Completed   0          70s
rsync-w54bc                                            0/2
Completed   0          13s
rsync-xxxvz                                            0/2
Completed   0          5m22s
```

Note: During the Cutover process, you may see an error indicating "Migration may be stuck." creating the rsync pods. Be patient in this phase, as the rsync pods will usually create successfully and the migration activity will complete.


Migration Toolkit for Containers
Red Hat
Clusters
Replication repositories
Migration plans
Hooks
Plans > cp4d-465-wvyb-to-9c9f > Cutover - 21 May 2023, 20:41:16

Migration details

Migration may be stuck.
Repeated errors occurred when attempting to create one or more Rsync pods in the source cluster. Please check controller logs for details.

Step	Elapsed time	Status	
Prepare	a few seconds	Complete	
Backup	a minute	Complete	Backup openshift-migration/migration-d4dc0-initial-8grtv 100% ✓
StageBackup	a few seconds	Complete	
DirectVolume	21 minutes	Running Rsync Pods to migrate	View Details

If the Cutover process fails, check velero status of the failed velero "migration" job. Example: `velero describe restore migration-xxx-final-xxx -n openshift-migration`

Migration Cutover Validation on Target cluster

Gather resources in CPD namespace:

```
$ oc get zenservice -n <cpd-namespace>
NAME      AGE
lite-cr   40m
$ oc get wkc -n <cpd-namespace>
NAME     VERSION   RECONCILED   STATUS   AGE
wkc-cr   4.6.5                           40m
$ oc get db2ucluster -n <cpd-namespace>
NAME           STATE   MAINTENANCESTATE   AGE
db2oltp-wkc                               40m

$ oc get ws
NAME    VERSION   RECONCILED   STATUS   AGE
ws-cr   6.5.0                           41m

$ oc get sts -n <cpd-namespace>
```

```
NAME                               READY   AGE
aiopenscale-ibm-aios-etcd          0/0     42m
aiopenscale-ibm-aios-kafka         0/0     42m
aiopenscale-ibm-aios-redis         0/0     42m
aiopenscale-ibm-aios-zookeeper     0/0     42m
c-db2oltp-wkc-db2u                 0/0     42m
dsx-influxdb                       0/0     42m
elasticsearch-master               0/0     42m
rabbitmq-ha                        0/0     42m
redis-ha-server                    0/0     42m
wdp-couchdb                        0/0     42m
wml-cpd-etcd                       0/0     42m
wml-deployment-agent               0/0     42m
zen-metastoredb                    0/0     42m
```

No pods found running in CPD namespace

```
$ oc get pod -n <cpd-namespace>
No resources found in cpd namespace.
```

Statefulsets will still point to the old storage class (Source storage class used in this example is called `cpd-storage`)

Example command:

```
$ oc get sts -n <cpd-namespace> c-db2oltp-wkc-db2u -o yaml | grep <cpd-storage-class>
storageClassName: cpd-storage
```

Migration cutover completed successfully.

Patch and update new storage class references on target cluster

Description of why patching is required:

MTC migration does not scour and patch all k8s resources for storage class references. It will only change the PV/PVC references, then copy over all of the existing k8s resources from the source namespace. However, any CR resources/etc that internally have references to storage classes and PVC claims for storage classes need to be adjusted to fit the new target cluster's storage classes.

Patch all kubernetes resources in the CPD namespace, by changing all references of old storage class(es) to new storage class(es).

- Find all resources with matches for the old storage class. Then strategically edit/change the corresponding k8s resource and change the referenced storage class.

- **Note:** You must make sure you pick the correct RWO or RWX storage class. Do not find/replace indiscriminately.

Discover all resources in the cpd instance namespace with storage class references

If an oadp dump of the cpd namespace was collected in the earlier step **"Collect an oadp resources dump of the cpd namespace on source cluster"**, that can be used to discover the resources that will need to be patched ahead of time. Do not attempt to collect an oadp resources dump if MTC was installed after cpd oadp was installed due to conflicts with oadp versions.

Alternatively, the resources can be queried live on either the source or target cluster.

Run the following command on the target cluster to look through all resources in the cpd namespace to find references to the old storage class, which will need to be patched/replaced with the new storage class(es). (For this command, replace `<cpd-namespace>` and `<source-storage-class>` with your values.)

```
NSPACE="<cpd-namespace>";SOURCE_SC="<source-storage-class>";for cr in $(oc
get $(oc api-resources --namespaced=true --verbs=list -o name | grep -vE
"packagemanifests|events|endpointslice|persistentvolumeclaim" | awk
'{printf "%s%s",sep,$0;sep=","}') --ignore-not-found -n ${NSPACE} -o
name);do cr_grep=$(oc get $cr -oyaml |grep -E "${SOURCE_SC}" |grep -v "
{\"apiVersion\":\"");if [[ -n "$cr_grep" ]];then echo $cr;fi;done
```

Example:

```
$ NSPACE="cpdinstance";SOURCE_SC="cpd-storage";for cr in $(oc get $(oc
api-resources --namespaced=true --verbs=list -o name | grep -vE
"packagemanifests|events|endpointslice|persistentvolumeclaim" | awk
'{printf "%s%s",sep,$0;sep=","}') --ignore-not-found -n ${NSPACE} -o
name);do cr_grep=$(oc get $cr -oyaml |grep -E "${SOURCE_SC}" |grep -v "
{\"apiVersion\":\"");if [[ -n "$cr_grep" ]];then echo $cr;fi;done
configmap/ibm-cpp-config
configmap/iis-db2u-config
configmap/wdp-profiling-iae-config
configmap/wkc-db2u-config
configmap/zen-lite-operation-configmap
mantaflow.adl.getmanta.com/mantaflow-wkc
analyticsengine.ae.cpd.ibm.com/analyticsengine-sample
statefulset.apps/aiopenscale-ibm-aios-etcd
statefulset.apps/aiopenscale-ibm-aios-kafka
statefulset.apps/aiopenscale-ibm-aios-zookeeper
statefulset.apps/c-db2oltp-iis-db2u
statefulset.apps/c-db2oltp-wkc-db2u
statefulset.apps/dsx-influxdb
statefulset.apps/elasticsearch-master
statefulset.apps/kafka
statefulset.apps/rabbitmq-ha
statefulset.apps/redis-ha-server
statefulset.apps/solr
statefulset.apps/wdp-couchdb
statefulset.apps/zen-metastoredb
statefulset.apps/zookeeper
foundationdbcluster.apps.foundationdb.org/wkc-foundationdb-cluster
ccs.ccs.cpd.ibm.com/ccs-cr
ibmcpd.cpd.ibm.com/ibmcpd-cr
db2oltpservice.databases.cpd.ibm.com/db2oltp-cr
datarefinery.datarefinery.cpd.ibm.com/datarefinery-sample
db2ucluster.db2u.databases.ibm.com/db2oltp-iis
db2ucluster.db2u.databases.ibm.com/db2oltp-wkc
formation.db2u.databases.ibm.com/db2oltp-iis
formation.db2u.databases.ibm.com/db2oltp-wkc
fdbcluster.foundationdb.opencontent.ibm.com/wkc-foundationdb-cluster
iis.iis.cpd.ibm.com/iis-cr
ug.wkc.cpd.ibm.com/ug-cr
wkc.wkc.cpd.ibm.com/wkc-cr
woservice.wos.cpd.ibm.com/aiopenscale
woservice.wos.cpd.ibm.com/openscale-defaultinstance
zenservice.zen.cpd.ibm.com/lite-cr
$
```

Execute the cpd-migrate.sh script to patch the storage class references

Prepare input.json file:

Before running the cpd-migrate.sh script, prepare the input.json file that will be fed into the cpd-migrate.sh script. This input.json file is expected to have matching json entries for every resource found in the discovery, from the previous step.

The creation of this input.json file was completed by the IBM team and it is called `cpd-migrate.input.json`. It has been provided for use with the `cpd-migrate.sh` script.

An example of the format of the input.json file:

```
{
  "analyticsengines.ae.cpd.ibm.com": {
    "analyticsengine-sample": [
      {
        "action": "patch",
        "type": "json-path",
        "sourcepath": ".spec.storageClass",
        "sourcevalue": "--source-file-storage-class",
        "targetvalue": "--target-file-storage-class"
      }
    ]
  },
  "statefulsets.apps": {
    "aiopenscale-ibm-aios-etcd": [
      {
        "action": "delete-create",
        "type": "json-path",
        "sourcepath":
".spec.volumeClaimTemplates[0].spec.storageClassName",
        "sourcevalue": "--source-block-storage-class",
        "targetvalue": "--target-block-storage-class"
      }
    ],
    "zen-metastoredb": [
      {
        "action": "delete-create",
        "type": "json-path",
        "sourcepath":
".spec.volumeClaimTemplates[0].spec.storageClassName",
        "sourcevalue": "--source-block-storage-class",
        "targetvalue": "--target-block-storage-class"
      }
    ]
  }
}
```

Run the cpd-migrate.sh script in preview mode:

Run the cpd-migrate.sh in preview mode, which will output json files with the updated storage class references. The resulting json files should be inspected to validate that the new storage class references are accurate and contains valid content to be applied to patch the corresponding resource.

Command (assumes new storage classes are default ODF storage class names):

```
$ ./cpd-migrate.sh restore \
--input-file cpd-migrate.input.json \
--cpd-operand-namespace <cpd-namespace> \
--source-block-storage-class <source-cluster-nfs-sc> \
--target-block-storage-class ocs-storagecluster-ceph-rbd \
--source-file-storage-class <source-cluster-nfs-sc> \
--target-file-storage-class ocs-storagecluster-cephfs \
--output-directory /tmp/cpd-migrate \
--preview
```

Example command:

```
./cpd-migrate.sh restore --input-file cpd-migrate.input.json --cpd-
operand-namespace cpdinstance --source-block-storage-class cpd-storage --
target-block-storage-class ocs-storagecluster-ceph-rbd --source-file-
storage-class cpd-storage --target-file-storage-class ocs-storagecluster-
cephfs --preview --output-directory ./preview
```

Example output-directory contents with resulting json files:

```
$ ls -1
cpdinstance.analyticsengines.ae.cpd.ibm.com.analyticsengine-sample.json
cpdinstance.ccs.ccs.cpd.ibm.com.ccs-cr.json
cpdinstance.configmaps.wdp-profiling-iae-config.json
cpdinstance.configmaps.wkc-db2u-config.json
cpdinstance.configmaps.zen-lite-operation-configmap.json
cpdinstance.datarefinery.datarefinery.cpd.ibm.com.datarefinery-sample.json
cpdinstance.db2uclusters.db2u.databases.ibm.com.db2oltp-iis.json
cpdinstance.db2uclusters.db2u.databases.ibm.com.db2oltp-wkc.json
cpdinstance.fdbclusters.foundationdb.opencontent.ibm.com.wkc-foundationdb-cluster.json
cpdinstance.formations.db2u.databases.ibm.com.db2oltp-iis.json
cpdinstance.formations.db2u.databases.ibm.com.db2oltp-wkc.json
cpdinstance.foundationdbclusters.apps.foundationdb.org.wkc-foundationdb-cluster.json
cpdinstance.ibmcpds.cpd.ibm.com.ibmcpd-cr.json
cpdinstance.iis.iis.cpd.ibm.com.iis-cr.json
cpdinstance.mantaflows.adl.getmanta.com.mantaflow-wkc.json
cpdinstance.statefulsets.apps.aiopenscale-ibm-aios-etcd.json
cpdinstance.statefulsets.apps.aiopenscale-ibm-aios-kafka.json
cpdinstance.statefulsets.apps.aiopenscale-ibm-aios-zookeeper.json
cpdinstance.statefulsets.apps.c-db2oltp-iis-db2u.json
cpdinstance.statefulsets.apps.c-db2oltp-wkc-db2u.json
cpdinstance.statefulsets.apps.dsx-influxdb.json
cpdinstance.statefulsets.apps.elasticsearch-master.json
cpdinstance.statefulsets.apps.kafka.json
cpdinstance.statefulsets.apps.rabbitmq-ha.json
cpdinstance.statefulsets.apps.redis-ha-server.json
cpdinstance.statefulsets.apps.solr.json
cpdinstance.statefulsets.apps.wdp-couchdb.json
cpdinstance.statefulsets.apps.zen-metastoredb.json
cpdinstance.statefulsets.apps.zookeeper.json
cpdinstance.ug.wkc.cpd.ibm.com.ug-cr.json
cpdinstance.wkc.wkc.cpd.ibm.com.wkc-cr.json
cpdinstance.woservices.wos.cpd.ibm.com.aiopenscale.json
cpdinstance.woservices.wos.cpd.ibm.com.openscale-defaultinstance.json
cpdinstance.zenservices.zen.cpd.ibm.com.lite-cr.json
```

Run the cpd-migrate.sh script to perform the resource updates:

After running cpd-migrate.sh in preview mode, and inspecting and validating the resulting json files, run the cpd-migrate.sh script without the preview option. This will take action on the kubernetes resources in the actual cluster, updating the resources with storage class references in the cpd instance namespace.

Note: There will be a small subset of resources that will have actions of "ignore" or "manual". This means that the cpd-migrate.sh tool will not take action on those resources.

- For a resource with action of "ignore", that indicates that no action is required and the existing reference to the old storage class can remain.

Example of "ignore" action:

```
---------------------------------------------------
Time: 2023-05-31T06:05:41.543+0000 level=info - patch-resource:
cpdinstance, configmaps, ibm-cpp-config, [{"action":"ignore","type":"json-
path","sourcepath":".data.storageclass.default","sourcevalue":"--source-
block-storage-class","targetvalue":""},{"action":"ignore","type":"json-
path","sourcepath":".data.storageclass.list","sourcevalue":"--source-
block-storage-class","targetvalue":""}]

Time: 2023-05-31T06:05:43.186+0000 level=info - Migrate Action:
{"action":"ignore","type":"json-
path","sourcepath":".data.storageclass.default","sourcevalue":"--source-
block-storage-class","targetvalue":""}
Time: 2023-05-31T06:05:43.497+0000 level=warning - Ignore
.data.storageclass.default: "cpd-storage"

Time: 2023-05-31T06:05:43.500+0000 level=info - Migrate Action:
{"action":"ignore","type":"json-
path","sourcepath":".data.storageclass.list","sourcevalue":"--source-
block-storage-class","targetvalue":""}
Time: 2023-05-31T06:05:43.809+0000 level=warning - Ignore
.data.storageclass.list: "cpd-storage"

Time: 2023-05-31T06:05:43.812+0000 level=info - configmaps ibm-cpp-config
- No Change
---------------------------------------------------
```

- For a resource with action of "manual", that indicates that the cpd-migrate.sh tool is not prepared to patch that resource. The user must manually edit/change the resource (i.e. `oc edit <kind>/<name>`; search for the source storage class name; replace with appropriate target storage class name).

Example of "manual" action:

```
---------------------------------------------------
Time: 2023-05-31T06:05:39.575+0000 level=info - patch-resource:
cpdinstance, configmaps, db2oltp-1685056932270995-db2oltp-cm,
[{"action":"manual","type":"string","sourcepath":".data.genkeys.sh","sourc
evalue":"--source-file-storage-class","targetvalue":"--target-file-
storage-class"}]

Time: 2023-05-31T06:05:41.176+0000 level=info - Migrate Action:
{"action":"manual","type":"string","sourcepath":".data.genkeys.sh","source
value":"--source-file-storage-class","targetvalue":"--target-file-storage-
class"}
Time: 2023-05-31T06:05:41.492+0000 level=warning - Manual Edit Required
.data.genkeys.sh: "cpd-storage"

Time: 2023-05-31T06:05:41.494+0000 level=info - configmaps db2oltp-
1685056932270995-db2oltp-cm - No Change
---------------------------------------------------
```

Command (assumes new storage classes are default ODF storage class names):

```
$ ./cpd-migrate.sh restore \
--input-file cpd-migrate.input.json \
--cpd-operand-namespace <cpd-namespace> \
--source-block-storage-class <source-cluster-nfs-sc> \
--target-block-storage-class ocs-storagecluster-ceph-rbd \
--source-file-storage-class <source-cluster-nfs-sc> \
--target-file-storage-class ocs-storagecluster-cephfs \
--output-directory /tmp/cpd-migrate
```

Validate patching completed successfully

After the patching activities, validate that the cpd namespace has no more references to the source cluster's storage class. With the exception of the resources that the cpd-migrate.sh tool identified as "ignore" or "manual", which need to be handled manually.

Run the following command on the target cluster to query for any remaining references to the old storage class, and handle appropriately:

```
NSPACE="<cpd-namespace>";SOURCE_SC="<source-storage-class>";for cr in $(oc
get $(oc api-resources --namespaced=true --verbs=list -o name | grep -vE
"packagemanifests|events|endpointslice|persistentvolumeclaim" | awk
'{printf "%s%s",sep,$0;sep=","}') --ignore-not-found -n ${NSPACE} -o
name);do cr_grep=$(oc get $cr -oyaml |grep -E "${SOURCE_SC}" |grep -v "
{\"apiVersion\":\"");if [[ -n "$cr_grep" ]];then echo $cr;fi;done
```

Start Cloud Pak for Data instance on target cluster

Prepare target cluster for Cloud Pak for Data

Before resuming CPD and services on the target cluster, make sure it has been prepared for CPD. Create custom SCCs and change required node settings. Reference: https://www.ibm.com/docs/en/cloud-paks/cp-data/4.6.x?topic=installing-preparing-your-cluster

Examples:

- Db2 kubeletconfig: `cpd-cli manage apply-db2-kubelet --openshift-type=self-managed`
- WKC SCC: `cpd-cli manage apply-scc --cpd_instance_ns=<cpd-namespace> --components=wkc`
- CRI-O settings: `cpd-cli manage apply-crio --openshift-type=self-managed`

Start services on the target cluster that were manually shutdown on source cluster

For the services and resources that were manually shutdown in the **"Scale down remaining services and resources running in CPD namespace"** step, startup those services now.

- Examples: nginx deployment, db2ucluster statefulsets, foundationdb, MANTA, openscale, etc.

nginx deployment (scale back up to 2):

```
oc scale --replicas=2 deploy/ibm-nginx -n <cpd-namespace>
```

MANTA scale up replicas (back to 1):

```
oc patch -p '{"spec":{"replicas":1}}' --type=merge mantaflow mantaflow-wkc
-n <cpd-namespace>
```

Openscale startup:

```
oc patch -p '{"spec":{"shutdown":"false"}}' --type=merge woservice
aiopenscale -n <cpd-namespace>
oc patch -p '{"spec":{"ignoreForMaintenance": false}}' --type=merge
woservice aiopenscale -n <cpd-namespace>
```

Foundationdb startup:

```
oc patch -p '{"spec":{"shutdown":"false"}}' --type=merge fdbcluster wkc-
foundationdb-cluster -n <cpd-namespace>
```

db2ucluster statefulsets (scale back up to 1):

```
oc scale --replicas=1 sts/c-db2oltp-iis-db2u -n <cpd-namespace>
oc scale --replicas=1 sts/c-db2oltp-wkc-db2u -n <cpd-namespace>

oc scale --replicas=1 sts/<db2wh-db2u-sts> -n <cpd-namespace>
oc scale --replicas=1 sts/<db2wh-etc-sts> -n <cpd-namespace>
```

Note: If there were additional db2u related statefulsets that were scaled down, scale them up here to the same replica number as it was on the source cluster.

Datastage deploy and statefulset

```
## Record this output for the resume operations:
oc get deploy,sts -n <cpd-namespace>|grep ds-px-default
oc scale --replicas=1 deploy/ds-px-default-ibm-datastage-px-runtime -n
<cpd-namespace>
oc scale --replicas=1 sts/ds-px-default-ibm-datastage-px-compute -n <cpd-
namespace>
```

Remaining resources: If there were additional services, deployments, statefulsets, or other that were manually shutdown on the source cluster, resume them here.

Examples of starting the resources:

Scale up nginx deployment

```
$ oc get deploy -n cpd |grep -E "NAME|nginx"
NAME                                         READY   UP-TO-DATE
AVAILABLE   AGE
aiopenscale-ibm-aios-nginx                   0/0     0            0
2d13h
ibm-nginx                                    0/0     0            0
2d13h
ibm-nginx-tester                             0/0     0            0
2d13h
spark-hb-nginx                               0/0     0            0
2d13h
$ oc scale --replicas=2 deploy/ibm-nginx -n cpd
deployment.apps/ibm-nginx scaled
$ oc get deploy ibm-nginx -n cpd
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
ibm-nginx   2/2     2            2           2d13h
$ oc get pod -n cpd
NAME                         READY   STATUS    RESTARTS   AGE
ibm-nginx-7cd879b767-fjgn4   1/1     Running   0          4m47s
ibm-nginx-7cd879b767-fn22k   1/1     Running   0          4m47s
```

Scale up db2u statefulsets

```
$ oc -n cpd get sts -n cpd |grep -E "NAME|db2"
NAME                                     READY   AGE
c-db2oltp-iis-db2u                       0/0     2d18h
c-db2oltp-wkc-db2u                       0/0     8d
c-db2wh-1686277435040922-db2u            0/0     15h
c-db2wh-1686277435040922-etcd            0/0     15h

$ oc scale --replicas=1 sts/c-db2oltp-wkc-db2u -n cpd
statefulset.apps/c-db2oltp-wkc-db2u scaled
$ oc scale --replicas=1 sts/c-db2oltp-iis-db2u -n cpd
statefulset.apps/c-db2oltp-iis-db2u scaled
$ oc scale --replicas=1 sts/c-db2wh-1686277435040922-db2u -n cpd
statefulset.apps/c-db2wh-1686277435040922-db2u scaled
```

```
$ oc scale --replicas=1 sts/c-db2wh-1686277435040922-etcd -n cpd
statefulset.apps/c-db2wh-1686277435040922-etcd scaled

$ oc -n cpd get pods|grep db2
c-db2oltp-iis-db2u-0                                        1/1     Running
0             44h
c-db2oltp-iis-instdb-sx2hm                                  0/1
Completed   0             2d19h
c-db2oltp-wkc-db2u-0                                        1/1     Running
0             44h
c-db2wh-1686277435040922-db2u-0                             1/1     Running
0             15h
c-db2wh-1686277435040922-etcd-0                             1/1     Running
0             15h
```

Start foundationdb

Check foundationdb CR state from CPD namespace:

```
$ oc -n cpd get fdbcluster wkc-foundationdb-cluster -o yaml |grep -E
"ceph|shutdown"

    pvcStorageClass: ocs-storagecluster-cephfs
            storageClassName: ocs-storagecluster-ceph-rbd
            storageClassName: ocs-storagecluster-ceph-rbd
  shutdown: "true"
  shutdownStatus: shutdown
```

Start foundationdb-cluster CR from CPD namespace:

```
$ oc -n cpd get fdbcluster wkc-foundationdb-cluster -o json |jq
'.spec.shutdown'
"true"
$ oc -n cpd edit fdbcluster wkc-foundationdb-cluster
fdbcluster.foundationdb.opencontent.ibm.com/wkc-foundationdb-cluster
edited
$ oc -n cpd get fdbcluster wkc-foundationdb-cluster -o json |jq
'.spec.shutdown'
"false"
$

$ oc get pods -n cpd
NAME                                                  READY   STATUS    RESTARTS
AGE
c-db2oltp-wkc-db2u-0                                  1/1     Running   0
35m
ibm-nginx-7cd879b767-fjgn4                            1/1     Running   0
45m
ibm-nginx-7cd879b767-fn22k                            1/1     Running   0
```

```
45m
wkc-foundationdb-cluster-cluster-controller-1   2/2     Running   0
18m
wkc-foundationdb-cluster-log-1                  2/2     Running   0
18m
wkc-foundationdb-cluster-log-2                  2/2     Running   0
18m
wkc-foundationdb-cluster-log-3                  2/2     Running   0
18m
wkc-foundationdb-cluster-log-4                  2/2     Running   0
18m
wkc-foundationdb-cluster-storage-1              2/2     Running   0
18m
wkc-foundationdb-cluster-storage-2              2/2     Running   0
18m
wkc-foundationdb-cluster-storage-3              2/2     Running   0
18m
wkc-foundationdb-cluster-storage-4              2/2     Running   0
18m
wkc-foundationdb-cluster-storage-5              2/2     Running   0
18m
wkc-foundationdb-cluster-storage-6              2/2     Running   0
18m
```

Start MANTA

From CPD namespace, set replicas to 1:

```
$ oc -n cpd get mantaflow mantaflow-wkc -o json |jq '.spec.replicas'
0
$ oc -n cpd edit mantaflow mantaflow-wkc
mantaflow.adl.getmanta.com/mantaflow-wkc edited
$ oc -n cpd get mantaflow mantaflow-wkc -o json |jq '.spec.replicas'
1
```

Start Openscale:

From CPD namespace, set shutdown=false in the two openscale CR instances:

```
$ oc patch WOService  openscale-defaultinstance -n cpd --type merge --
patch '{"spec": {"shutdown":"false"}}'
woservice.wos.cpd.ibm.com/openscale-defaultinstance patched
$ oc -n cpd patch WOService aiopenscale -n cpd --type merge --patch
'{"spec": {"shutdown": "false"}}'
woservice.wos.cpd.ibm.com/aiopenscale patched
$ oc -n cpd get WOService  openscale-defaultinstance -o json | jq
'.spec.shutdown'
"false"
$ oc -n cpd get WOService aiopenscale -o json | jq '.spec.shutdown'
```

```
"false"
$
```

Datastage

```
$ oc scale --replicas=1 deploy/ds-px-default-ibm-datastage-px-runtime -n
cpd
deploy.apps/ds-px-default-ibm-datastage-px-runtime scaled
$ oc scale --replicas=1 sts/ds-px-default-ibm-datastage-px-compute -n cpd
statefulset.apps/ds-px-default-ibm-datastage-px-compute scaled
```

Resume CPD service operations with restore posthooks

The cpd-cli oadp restore posthooks will take all of the CPD services out of maintenance mode and scale back up all of the deployments, statefulsets, etc.

Example Command:

```
cpd-cli oadp restore posthooks --include-namespaces <cpd-namespace> --log-
level=debug --verbose --scale-wait-timeout
```

Run the following commands to verify CPD instance is running:

```
$ oc get pods -n <cpd-namespace>
$ oc get route -n <cpd-namespace>
$ oc get pvc -n <cpd-namespace>
$ oc get sts -n <cpd-namespace>
$ oc get deployments -n <cpd-namespace>
```

Get CR status:

```
$ cpd-cli manage get-cr-status --cpd_instance_ns=<cpd-namespace>
$ oc get ccs -n <cpd-namespace>
$ oc get wkc wkc-cr -o yaml -n <cpd-namespace>
```

Validate that all installed CPD services resume to Completed status:

```
cpd-cli manage get-cr-status --cpd_instance_ns=<cpd-namespace>
```

After the command completes successfully, the migration restore is now complete!