Part V: Installing

Tables of Contents

Installing IBM Cloud Pak for Data

A Red Hat® OpenShift® Container Platform cluster administrator and project administrator can work together to prepare the cluster and install IBM® Cloud Pak for Data.

Before you begin

1. Ensure that you review the following information before you install Cloud Pak for Data:
   - Planning
   - System requirements
2. Determine which services you want to install.
   Some of the pre-installation tasks, such as creating catalog source and operator subscriptions, include steps for the services as well as the Cloud Pak for Data platform. If you know which services you plan to install, you can streamline your installation by batching these tasks.
3. Use the following information to ensure that you complete the appropriate tasks for your environment.

1. Do you have an existing Red Hat OpenShift Container Platform cluster?

Cloud Pak for Data is installed on a Red Hat OpenShift Container Platform Version 4.6 cluster.

Options	What to do
You already have an OpenShift 4.6 cluster	1. Go to 3. Do you already have supported persistent storage on your cluster?
You have an older version of OpenShift	1. Upgrade your cluster. For details, see the Red Hat OpenShift Container Platform documentation. 2. Then, go to 3. Do you already have supported persistent storage on your cluster?
You don't have an OpenShift cluster	1. Decide where you want to host your Cloud Pak for Data. Go to 2. Where do you want to host your Cloud Pak for Data installation?

2. Where do you want to host your Cloud Pak for Data installation?

You can deploy Cloud Pak for Data on-premises or on the cloud. Your deployment environment determines how you can install Red Hat OpenShift Container Platform:

Options	What to do
You want to deploy Cloud Pak for Data on-premises	1. Follow the Red Hat OpenShift Container Platform 4.6 documentation to install OpenShift. Additional guidance on setting up OpenShift is available in the IBM Cloud Paks documentation. Alternative: If you don't have existing hardware, you can purchase IBM Cloud Pak for Data System, which comes with Red Hat OpenShift Container Platform and Cloud Pak for Data already installed. 2. Go to 3. Do you already have supported persistent storage on your cluster?
You want to deploy Cloud Pak for Data on cloud	1. Decide which cloud provider you want to use. 2. Decide how you want to install and manage Red Hat OpenShift Container Platform. For details, see Installing Red Hat OpenShift Container Platform. 3. Go to 3. Do you already have supported persistent storage on your cluster?

3. Do you already have supported persistent storage on your cluster?

The Cloud Pak for Data platform supports the following storage:

Red Hat OpenShift Container Storage
    Version: 4.6 or later fixes
Network File System (NFS)
    Version: 4
Portworx
    Version: 2.7.0 or later fixes
IBM Cloud File Storage
    Version: Not applicable

Ensure that you have storage that works with the Services that you plan to install.

Options	What to do
You have the supported storage	1. Review Setting up shared persistent storage to determine whether you need to complete any additional tasks to configure the storage for Cloud Pak for Data. 2. Go to 4. Do you have the required OpenShift projects on your cluster?
You don't have supported storage	1. Decide which storage you want to use. Ensure that you choose storage that works with the services that you plan to install. 2. Follow the guidance in Setting up shared persistent storage for installing and configuring the storage. 3. Go to 4. Do you have the required OpenShift projects on your cluster?

4. Do you have the required OpenShift projects on your cluster?

At a minimum, you must have a project where you will install the Cloud Pak for Data operators and service operators and a project where you will install an instance of Cloud Pak for Data. You might need additional projects depending on whether you want to:

- Separate the IBM Cloud Pak® foundational services operators from the Cloud Pak for Data operators
- Install multiple instances of Cloud Pak for Data on the cluster

1. Review the guidance in Creating projects (namespaces) on Red Hat OpenShift Container Platform to determine whether:
   - You have the necessary projects on your cluster
   - You need to create operator groups for the projects
2. Go to 5. Do you have your API key?

5. Do you have your API key?

The Cloud Pak for Data software images are hosted on the IBM Entitled Registry. To access the images, you must have your IBM entitlement API key.

Options	What to do
You have your API key	1. Go to 6. How are you going to access the required software images?
You don't have your API key	1. Follow the guidance in Obtaining your IBM entitlement API key. 2. Go to 6. How are you going to access the required software images?

6. How are you going to access the required software images?

Cloud Pak for Data images are accessible from the IBM Entitled Registry. In most situations, it is strongly recommended that you mirror the necessary software images from the IBM Entitled Registry to a private container registry.

The only situation in which you might consider pulling images directly from the IBM Entitled Registry is when your cluster is not air-gapped, your network is extremely reliable, and latency is not a concern. However, for predictable and reliable performance, you should mirror the images to a private container registry.

Important: You must mirror the necessary images to your container registry in the following situations:

- Your cluster is air-gapped (also called an offline or disconnected cluster)
- Your cluster uses an *allowlist* to permit direct access by specific sites and the allowlist does not include the IBM Entitled Registry
- Your cluster uses a *blocklist* to prevent direct access by specific sites and the blocklist includes the IBM Entitled Registry

Options	What to do
You are pulling images from the IBM Entitled Registry	1. Go to 7. Configuring your cluster to pull software images.
You are mirroring images to a private container registry	1. Review the guidance in Mirroring images to your container registry to ensure you have a container registry that meets the minimum requirements. 2. Determine how you will mirror the images and complete the appropriate task: • Mirroring images with a bastion node • Mirroring images with an intermediary container registry 3. Go to 7. Configuring your cluster to pull software images.

7. Configuring your cluster to pull software images

You must ensure that your cluster is configured to pull the software images from the appropriate location.

1. Complete the appropriate steps for your environment in Configuring your cluster to pull Cloud Pak for Data images.
2. Go to 8. Are the IBM Cloud Pak foundational services already installed on your cluster?

8. Are the IBM Cloud Pak foundational services already installed on your cluster?

The IBM Cloud Pak foundational services are a prerequisite for Cloud Pak for Data. However, in some situations the IBM Cloud Pak for Data platform operator can automatically install the IBM Cloud Pak foundational services operators and services on the cluster.

Options	What to do
IBM Cloud Pak foundational services is already installed	1. Go to 9. Creating operator subscriptions
IBM Cloud Pak foundational services is not installed and you are using the express installation method	With the express installation method, all of the operators are in the same OpenShift project and the IBM Cloud Pak for Data platform operator can automatically install IBM Cloud Pak foundational services. 1. Go to 9. Creating operator subscriptions.
IBM Cloud Pak foundational services is not installed and you are using the specialized installation method	With the specialized installation method, the IBM Cloud Pak foundational services operators and the Cloud Pak for Data operators are in separate OpenShift project. To ensure IBM Cloud Pak foundational services is installed in the correct project, you must manually install it. 1. Follow the guidance in Installing IBM Cloud Pak foundational services. 2. Go to 9. Creating operator subscriptions.

9. Creating operator subscriptions

An operator subscription tells the cluster where to install a given operator and gives information about the operator to Operator Lifecycle Manager (OLM).

1. Complete the appropriate steps for your environment in Creating operator subscriptions.
2. Go to 10. Do you plan to install services that require custom SCCs?

10. Do you plan to install services that require custom SCCs?

The following services require custom security context constraints:

- Watson™ Knowledge Catalog
- Db2®
- Db2 Warehouse
- Db2 Big SQL
- Data Virtualization
- OpenPages®

Options	What to do
You plan to install one or more of these services	1. Create the appropriate SCCs for your environment. For details, see Creating custom security context constraints for services. 2. Go to 11. Do you plan to install services that require specific node settings?
You don't plan to install any of these services	1. Go to 11. Do you plan to install services that require specific node settings?

11. Do you plan to install services that require specific node settings?

The following services require specific node settings:

- Data Virtualization
- Db2
- Db2 Big SQL
- Db2 Warehouse
- Jupyter Notebooks with Python 3.7 for GPU
- OpenPages
- Watson Knowledge Catalog
- Watson Machine Learning Accelerator
- Watson Studio

You might also need to adjust some node settings if you are working with large data sets or you have slower network speeds.

Options	What to do
You plan to install one or more of these services	1. Change the appropriate node settings. For details, see Changing required node settings. 2. Go to 12. Do you need to install the scheduling service?
You don't plan to install any of these services	1. Go to 12. Do you need to install the scheduling service?

12. Do you need to install the scheduling service?

The scheduling service is required if you plan to install Watson Machine Learning Accelerator.

However, it is strongly recommended that you install the scheduling service so that you can programmatically enforce the quotas that you set on the platform and on individual services.

Options	What to do
You need to install the scheduling service	1. Follow the guidance in Installing the scheduling service. 2. Go to 13. Installing Cloud Pak for Data.
You don't plan to install the scheduling service	1. Go to 13. Installing Cloud Pak for Data.

13. Installing Cloud Pak for Data

Depending on the number of OpenShift projects you created, you can install one or more instances of Cloud Pak for Data on your cluster.

1. Install Cloud Pak for Data.
2. Go to 14. Completing post-installation tasks.

14. Completing post-installation tasks

After you install Cloud Pak for Data, make sure your cluster is secure and complete tasks that will impact how users interact with Cloud Pak for Data, such as configuring SSO or changing the route to the platform.

1. Complete the appropriate tasks for your environment in Post-installation tasks.
2. Go to 15. Installing services.

15. Installing services

You are ready to install services on your cluster. Instructions for installing IBM services are available in Services.

- **Pre-installation tasks**
  Before you install Cloud Pak for Data, complete the following tasks.
- **Installing Cloud Pak for Data**
  When you install IBM Cloud Pak for Data, you update the IBM Cloud Pak for Data platform operator and the IBM Cloud Pak foundational services operator to watch the project where you will install IBM Cloud Pak for Data. Then, you create a custom resource to install Cloud Pak for Data in that project.
- **Post-installation tasks**
  After you install Cloud Pak for Data, complete the following tasks.
- **Uninstalling Cloud Pak for Data**
  A project administrator can uninstall the Cloud Pak for Data control plane.

Pre-installation tasks

Before you install Cloud Pak for Data, complete the following tasks.

Tip: See Installing IBM Cloud Pak for Data for guidance about which tasks you need to complete based on your environment.

1. Installing Red Hat OpenShift Container Platform
   IBM Cloud Pak for Data is deployed on a Red Hat OpenShift Container Platform cluster. If you don't have an existing cluster, complete the appropriate steps to install Red Hat OpenShift on your environment.
2. Setting up shared persistent storage
   Before you can install Cloud Pak for Data, you must set up shared persistent storage on your Red Hat OpenShift cluster.
3. Creating projects (namespaces) on Red Hat OpenShift Container Platform
   Before you install IBM Cloud Pak for Data on Red Hat OpenShift Container Platform, a cluster administrator should create the OpenShift projects (Kubernetes namespaces) where you plan to deploy the Cloud Pak for Data software.
4. Obtaining your IBM entitlement API key
   The IBM entitlement API key enables you to pull software images from the IBM Entitled Registry, either for installation or for mirroring.
5. Mirroring images to your container registry
   IBM Cloud Pak for Data images are accessible from the IBM Entitled Registry. In most situations, it is strongly recommended that you mirror the necessary software images from the IBM Entitled Registry to a private container registry.
6. Configuring your cluster to pull Cloud Pak for Data images
   To ensure that your cluster can pull Cloud Pak for Data software images, you must update your cluster configuration.
7. Installing IBM Cloud Pak foundational services
   IBM Cloud Pak foundational services is a prerequisite for IBM Cloud Pak for Data. IBM Cloud Pak foundational services is installed one time on the cluster and is used by any instances of Cloud Pak for Data or other IBM Cloud Paks that are installed on the cluster.
8. Creating operator subscriptions
   An operator subscription tells the cluster where to install a given operator and gives information about the operator to Operator Lifecycle Manager (OLM).
9. Creating custom security context constraints for services
   Most Cloud Pak for Data services use the `restricted` security context constraint (SCC) that is provided by Red Hat OpenShift Container Platform. However, if you plan to install certain Cloud Pak for Data services, you might need to create some custom SCCs.
10. Changing required node settings
    Some services that run on IBM Cloud Pak for Data require specific settings on the nodes in the cluster. To ensure that the cluster has the required settings for these services, an operating system administrator with `root` privileges must review and adjust the settings on the appropriate nodes in the cluster.

Related reference

- Installing Cloud Pak for Data
- Post-installation tasks
- Uninstalling Cloud Pak for Data

Installing Red Hat OpenShift Container Platform

IBM® Cloud Pak for Data is deployed on a Red Hat® OpenShift® Container Platform cluster. If you don't have an existing cluster, complete the appropriate steps to install Red Hat OpenShift on your environment.

Tip: After you install Red Hat OpenShift Container Platform on your cluster, see Installing IBM Cloud Pak for Data for an overview of the installation flow.

Supported deployment environments

You can deploy Cloud Pak for Data on-premises or on the cloud. Your deployment environment determines how you can install Red Hat OpenShift Container Platform:

- If you deploy Cloud Pak for Data on-premises, you must install a self-managed Red Hat OpenShift Container Platform cluster.
- If you deploy Cloud Pak for Data on cloud, you can choose whether to use a managed or self-managed Red Hat OpenShift Container Platform cluster. However, **managed** OpenShift is not supported on all clouds.

Cloud provider	Managed OpenShift	Self-managed OpenShift
IBM Cloud	Supported (recommended)	Supported
Amazon Web Services (AWS)	Not supported	Supported
Microsoft Azure	Not supported	Supported
Google Cloud	Not supported	Supported

Go to the appropriate section for your deployment environment:

- On-premises
- IBM Cloud
- AWS

- Azure
- Google Cloud

On-premises

You can install a **self-managed** OpenShift cluster on-premises.

Follow the Red Hat OpenShift Container Platform 4.6 documentation to install OpenShift.

Additional guidance on setting up OpenShift is available in the IBM Cloud Paks documentation.

Alternative: If you don't have existing hardware, you can purchase IBM Cloud Pak for Data System, which comes with Red Hat OpenShift Container Platform and Cloud Pak for Data already installed.

IBM Cloud

Managed OpenShift
To install managed OpenShift, you can:

- Install Red Hat OpenShift Container Platform on IBM Cloud.

Self-managed OpenShift
To install self-managed OpenShift, contact IBM Software Support.

AWS

Self-managed OpenShift
To install self-managed OpenShift, you can:

- Install Red Hat OpenShift Container Platform on AWS

Azure

Self-managed OpenShift
To install self-managed OpenShift, you can:

- Install Red Hat OpenShift Container Platform on Azure.

Google Cloud

You can install a **self-managed** OpenShift cluster Google Cloud.

For details, see Red Hat OpenShift Container Platform on Google Cloud.

Next topic: Setting up shared persistent storage

Setting up shared persistent storage

Before you can install Cloud Pak for Data, you must set up shared persistent storage on your Red Hat® OpenShift® cluster.

Tip: For information about supported storage providers, see Storage considerations.
Ensure that the services that you plan to install on Cloud Pak for Data can use the storage that you use. For details, see Compute, memory, and storage requirements.

Select your storage type and complete the steps to set up storage.

Storage type	What to do
Red Hat OpenShift Container Storage	Installation To install OpenShift Container Storage, see the Red Hat OpenShift Container Storage documentation. Post-installation set up No additional set up is required.
Portworx	Installation Determine which version of Portworx you plan to use: • Portworx Essentials for IBM For details, see Setting up Portworx storage • Portworx Enterprise for IBM For details, see Install Portworx on OpenShift Post-installation set up You must configure the required storage classes. For details, see Creating Portworx storage classes

Storage type	What to do
NFS	Installation Refer to the installation documentation for your NFS storage provider. Post-installation set up You must set up dynamic storage and configure your storage. For details, see Setting up NFS storage
IBM Cloud File Storage	Installation When you configure your Red Hat OpenShift cluster, ensure that you select one of the following IBM Cloud File Storage storage classes: • `ibmc-file-gold-gid` • `ibm-file-custom-gold-gid` Post-installation set up No additional configuration is required to use IBM Cloud File Storage. However, you might need to adjust your I/O and storage size settings for production workloads, as indicated in the Storage comparison table.

- **Setting up Portworx storage**
  If you decide to use Portworx for shared persistent storage, you can use your existing Portworx storage or the Portworx Essentials for IBM solution, which is included with IBM Cloud Pak for Data.
- **Setting up NFS storage**
  By default, NFS does not support dynamic storage provisioning. If you plan to use Cloud Pak for Data for persistent storage, you must set up your NFS storage before you install Cloud Pak for Data.

Previous topic: Installing Red Hat OpenShift Container Platform
Next topic: Creating projects (namespaces) on Red Hat OpenShift Container Platform

Setting up Portworx storage

If you decide to use Portworx for shared persistent storage, you can use your existing Portworx storage or the Portworx Essentials for IBM® solution, which is included with IBM Cloud Pak for Data.

Important: Portworx Version 2.7.0 or later is required.

Existing Portworx storage
: Your configuration must support dynamic storage provisioning with `ReadWriteMany` access on the persistent volumes.

Portworx Essentials for IBM storage
: You must download the Portworx Essentials for IBM package from IBM Passport Advantage®. For details, see Licenses and entitlements.
This package includes the files that you need to set up Portworx storage on your cluster.

About this task

Use the following table to determine which tasks you must complete based on your environment:

Task	Existing Portworx storage	Portworx Essentials for IBM storage
Planning for Portworx Essentials for IBM Cloud Pak for Data	Recommended	Required
Installing Portworx Essentials for IBM Cloud Pak for Data	Not applicable	Required
Creating Portworx storage classes	Required	Required

To set up Portworx storage, complete the required tasks for your environment:

- **Planning for Portworx Essentials for IBM Cloud Pak for Data**
  If you plan to use Portworx storage, ensure that you understand the restrictions around the instance of Portworx Essentials that is included with IBM Cloud Pak for Data.
- **Installing Portworx Essentials for IBM Cloud Pak for Data**
  If you plan to use Portworx storage, you must install Portworx before you install the IBM Cloud Pak for Data control plane or any services.
- **Creating Portworx storage classes**
  If you decide to use Portworx as your storage option, Cloud Pak for Data requires the following storage classes. You can create them either manually or automatically.

Planning for Portworx Essentials for IBM Cloud Pak for Data

If you plan to use Portworx storage, ensure that you understand the restrictions around the instance of Portworx Essentials that is included with IBM® Cloud Pak for Data.

Portworx Essentials for IBM Cloud Pak for Data has the following limits:

- A maximum of 128 Virtual Processing Cores per cluster (8 compute nodes).
- A maximum of 500 persistent volumes per cluster.
- A maximum of 5 TB capacity for each persistent volume.

By default, when you install Portworx Essentials for IBM Cloud Pak for Data, Portworx is automatically installed on all of the compute nodes in your cluster, regardless of whether they have a raw disk designated for Portworx storage.

Attention: If you have more compute nodes in your cluster using Portworx Essentials than the maximum eight allowed, even if any of the compute nodes are storageless, the default behavior automatically puts you over the number of nodes that you are entitled to use with the Portworx Essentials license. Consequently, Portworx will fail to start on the additional storage nodes.

Storageless nodes

If a node does not have a storage device but Portworx is installed on the node, the node is considered a *storageless* node. Both storage and storageless nodes count toward the license. For OpenShift® version 4.6, you can label compute nodes as **`px/storageless=true`** before you install Portworx on them to make them storageless. For example, **`oc label nodes node1 node2 px/storageless=true`**.

Clusters with support for multiple storage types

You can prevent Portworx from being installed on specific compute nodes by labeling the nodes with **`px/enabled=false`**. This label enables you to control which nodes are used with Portworx. You should label compute nodes with **`px/enabled=false`** in the following situations:

- The node does not have a raw disk designated for Portworx storage.
- Only some of the services in your cluster use Portworx storage.

The following example illustrates a situation in which you would want to label some of your compute nodes with **`px/enabled=false`**.

You are planning to install several services that support Portworx storage. In addition, you decide to install Db2® Warehouse on multiple nodes. Db2 Warehouse MPP does not support Portworx storage, so the nodes where Db2 Warehouse will be installed do not need to have Portworx installed. Before you install Portworx, label the nodes where you plan to run Db2 Warehouse with **`px/enabled=false`**.

Other considerations

Depending on the number of services that you plan to run in Cloud Pak for Data and the requirements for each service, you might need to adjust the resources in your cluster:

- If all of the services in your cluster use Portworx storage, you can optionally increase the size of your nodes to enable more services to run on your compute nodes. This enables you to stay within the entitled number of nodes.
- If all of the services in your cluster use Portworx storage but you need more than the maximum nodes allowed, you can upgrade your Portworx license.

Installing Portworx Essentials for IBM Cloud Pak for Data

If you plan to use Portworx storage, you must install Portworx before you install the IBM® Cloud Pak for Data control plane or any services.

Before you begin

Required role: To complete this task, you must be a Red Hat® OpenShift® cluster administrator.

Ensure that you have:

- At least 1 TB of raw, unformatted disk on every compute node that is designated for application storage and at least 100 GB of raw unformatted disk for metadata storage. The raw disk must have the same device name on all of the compute nodes.
- On all of the nodes in your cluster, you have the latest version of CRI-O that is available on the Red Hat repository. (Version 1.11.16 or later).
  Important: Ensure that CRI-O is configured according to the information in Software requirements.
- On all of the nodes in your cluster, you have the Podman utility.

About this task

Portworx Essentials for IBM Cloud Pak for Data has the following limits:

- A maximum of 128 Virtual Processing Cores per cluster (8 compute nodes).
- A maximum of 500 persistent volumes per cluster.
- A maximum of 5 TB capacity for each persistent volume.

Contact IBM for additional use licenses.

Procedure

1. Log in to the OpenShift cluster as an administrator:

   ```
   oc login OpenShift_URL:port
   ```

2. On the workstation where you plan to install Portworx Essentials for IBM Cloud Pak for Data from, download the appropriate file from IBM Passport Advantage®:

Cloud Pak for Data Edition	Part number	TAR file
Enterprise Edition	G01KGEN	CP4D_ENT_ED_Portworx_2.7.tgz
Standard Edition	G01KLEN	CP4D_ST_ED_Portworx_2.7.tgz

3. Extract the contents of the TAR file, which contains a cpd-portworx directory of scripts and a cpd-portworx/px-images/px_2.7.0.0-dist.tgz file with all of the Portworx images in it.

   ```
   tar zxvf filename.tgz
   ```

   After you extract the contents of the TAR file, you can delete it from your file system to save space.
4. From the cpd-portworx/px-images directory, run the podman-rm-local-images.sh and process-px-images.sh scripts to upload the Portworx images into your target registry from the TGZ package of Portworx images:

Option	Description

Option	Description

OpenShift v4.x	`cd ./cpd-portworx/px-images`
	`export PODMAN_LOGIN_ARGS="--tls-verify=false"`
	`export PODMAN_PUSH_ARGS="--tls-verify=false"`
	`./podman-rm-local-images.sh`
	`./process-px-images.sh -r $(oc registry info -n openshift-image-registry) -u kubeadmin -p $(oc whoami -t) \`
	`-s kube-system -c podman -t ./px_2.7.0.0-dist.tgz`

Tip: To verify that the images were successfully pushed to the registry, run the following command:

```
oc get imagestreams -n kube-system
```

If the push failed, delete the image registry pod and then retry the push:

```
oc get po -n project
oc delete po image-registry... -n project
```

5. To install Portworx on your cluster, see the following readme:

Option	Description
OpenShift v4.x	See "Install Portworx 2.7.0 on OCP 4.x" in px-install-4.x/README.txt.

6. Verify that Portworx has been deployed correctly:

```
PX_POD=$(kubectl get pods -l name=portworx -n kube-system -o jsonpath='{.items[0].metadata.name}')
kubectl exec $PX_POD -n kube-system -- /opt/pwx/bin/pxctl status
```

If you see the `Status: PX is operational` message, the Portworx deployment succeeded.

What to do next

Complete Creating Portworx storage classes.

Attention: If you uninstall this instance of Portworx, you will lose all of your stored data. To uninstall the Portworx instance, remove all pods and persistent volume claims that mount or refer to Portworx storage. Then delete the associated Portworx projects and enter the following command from the cpd-portworx\px-install-4.x directory:

```
./px-uninstall.sh
```

The script deletes all Portworx services and volumes, and removes all Portworx images.

Creating Portworx storage classes

If you decide to use Portworx as your storage option, Cloud Pak for Data requires the following storage classes. You can create them either manually or automatically.

Before you begin

Required role: To complete this task, you must be a cluster administrator.

Ensure that you have a minimum of 1 TB of raw, unformatted disk on every compute node that is designated for storage. The raw disk must have the same device name on all of the worker nodes.

About this task

If you installed the Portworx Essentials instance that comes free with IBM® Cloud Pak for Data, you can automatically create the storage classes required for Portworx by running the script: **`px-sc.sh`**.

If you are using your own Portworx instance and want to use the px-sc.sh script to automatically create the required storage classes, extract the contents of the Portworx Essentials package that comes for free with Cloud Pak for Data. See Installing Portworx Essentials for IBM Cloud Pak for Data for details.

If you opt not to use the px-sc.sh script, you must manually create the following Portworx storage classes that are required for Cloud Pak for Data:

Storage class	Storage type	Storage class definitions

`portworx-couchdb-sc`	CouchDB	`# CouchDB (Implemented application-level redundancy)`
		`cat <<EOF | oc create -f -`
		`kind: StorageClass`
		`apiVersion: storage.k8s.io/v1`
		`metadata:`
		`  name: portworx-couchdb-sc`
		`provisioner: kubernetes.io/portworx-volume`
		`parameters:`
		`  repl: "3"`
		`  priority_io: "high"`
		`  io_profile: "db_remote"`
		`  disable_io_profile_protection: "1"`
		`allowVolumeExpansion: true`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

Storage class	Storage type	Storage class definitions

portworx-elastic-sc	Elastic Search	`# ElasticSearch (Implemented application-level redundancy)`
		`cat <<EOF | oc create -f -`
		`kind: StorageClass`
		`apiVersion: storage.k8s.io/v1`
		`metadata:`
		`  name: portworx-elastic-sc`
		`provisioner: kubernetes.io/portworx-volume`
		`parameters:`
		`  repl: "2"`
		`  priority_io: "high"`
		`  io_profile: "db_remote"`
		`  disable_io_profile_protection: "1"`
		`allowVolumeExpansion: true`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

portworx-solr-sc	Solr	`# Solr`
		`cat <<EOF | oc create -f -`
		`kind: StorageClass`
		`apiVersion: storage.k8s.io/v1`
		`metadata:`
		`  name: portworx-solr-sc`
		`provisioner: kubernetes.io/portworx-volume`
		`parameters:`
		`  repl: "3"`
		`  priority_io: "high"`
		`  io_profile: "db_remote"`
		`  disable_io_profile_protection: "1"`
		`allowVolumeExpansion: true`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

portworx-cassandra-sc	Cassandra	`# Cassandra`
		`cat <<EOF | oc create -f -`
		`kind: StorageClass`
		`apiVersion: storage.k8s.io/v1`
		`metadata:`
		`  name: portworx-cassandra-sc`
		`provisioner: kubernetes.io/portworx-volume`
		`parameters:`
		`  repl: "3"`
		`  priority_io: "high"`
		`  io_profile: "db_remote"`
		`  disable_io_profile_protection: "1"`
		`allowVolumeExpansion: true`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

portworx-kafka-sc	Apache Kafka	`# Kafka`
		`cat <<EOF | oc create -f -`
		`kind: StorageClass`
		`apiVersion: storage.k8s.io/v1`
		`metadata:`
		`  name: portworx-kafka-sc`
		`provisioner: kubernetes.io/portworx-volume`
		`parameters:`
		`  repl: "3"`
		`  priority_io: "high"`
		`  io_profile: "db_remote"`
		`  disable_io_profile_protection: "1"`
		`allowVolumeExpansion: true`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

portworx-metastoredb-sc	Metastore	`# metastoredb:`
		`cat <<EOF | oc create -f -`
		`apiVersion: storage.k8s.io/v1`
		`kind: StorageClass`
		`metadata:`
		`  name: portworx-metastoredb-sc`
		`parameters:`
		`  priority_io: high`
		`  io_profile: db_remote`
		`  repl: "3"`
		`  disable_io_profile_protection: "1"`
		`allowVolumeExpansion: true`
		`provisioner: kubernetes.io/portworx-volume`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

Storage class	Storage type	Storage class definitions

portworx-rwx-gp3-sc	GP3 replica 3	`# General Purpose, 3 Replicas - Default SC for other applications`
		`# without specific SC defined and with RWX volume access mode - New Install`
		`cat <<EOF | oc create -f -`
		`apiVersion: storage.k8s.io/v1`
		`kind: StorageClass`
		`metadata:`
		`  name: portworx-rwx-gp3-sc`
		`parameters:`
		`  priority_io: high`
		`  repl: "3"`
		`  sharedv4: "true"`
		`  io_profile: db_remote`
		`  disable_io_profile_protection: "1"`
		`allowVolumeExpansion: true`
		`provisioner: kubernetes.io/portworx-volume`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

portworx-shared-gp3	GP3 replica 3	`# General Purpose, 3 Replicas [Default for other applications without`
		`# specific SC defined and with RWX volume access mode] - SC portworx-shared-gp3 for upgrade purposes`
		`cat <<EOF | oc create -f -`
		`apiVersion: storage.k8s.io/v1`
		`kind: StorageClass`
		`metadata:`
		`  name: portworx-shared-gp3`
		`parameters:`
		`  priority_io: high`
		`  repl: "3"`
		`  sharedv4: "true"`
		`  io_profile: db_remote`
		`  disable_io_profile_protection: "1"`
		`allowVolumeExpansion: true`
		`provisioner: kubernetes.io/portworx-volume`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

portworx-rwx-gp2-sc	GP2 replica 2	`# General Purpose, 2 Replicas RWX volumes`
		`cat <<EOF | oc create -f -`
		`apiVersion: storage.k8s.io/v1`
		`kind: StorageClass`
		`metadata:`
		`  name: portworx-rwx-gp2-sc`
		`parameters:`
		`  priority_io: high`
		`  repl: "2"`
		`  sharedv4: "true"`
		`  io_profile: db_remote`
		`  disable_io_profile_protection: "1"`
		`allowVolumeExpansion: true`
		`provisioner: kubernetes.io/portworx-volume`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

portworx-dv-shared-gp	Shared DV replica 1	`# DV - Single replica`
		`cat <<EOF | oc create -f -`
		`allowVolumeExpansion: true`
		`apiVersion: storage.k8s.io/v1`
		`kind: StorageClass`
		`metadata:`
		`  name: portworx-dv-shared-gp`
		`parameters:`
		`  block_size: 4096b`
		`  priority_io: high`
		`  repl: "1"`
		`  shared: "true"`
		`provisioner: kubernetes.io/portworx-volume`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

portworx-dv-shared-gp3	Shared DV GP3 replica 3	`# DV - three replicas`
		`cat <<EOF | oc create -f -`
		`allowVolumeExpansion: true`
		`apiVersion: storage.k8s.io/v1`
		`kind: StorageClass`
		`metadata:`
		`  name: portworx-dv-shared-gp3`
		`parameters:`
		`  block_size: 4096b`
		`  priority_io: high`
		`  repl: "3"`
		`  shared: "true"`
		`provisioner: kubernetes.io/portworx-volume`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

Storage class	Storage type	Storage class definitions

portworx-shared-gp-allow	Streams	# Streams
		cat <<EOF | oc create -f -
		allowVolumeExpansion: true
		apiVersion: storage.k8s.io/v1
		kind: StorageClass
		metadata:
		  name: portworx-shared-gp-allow
		parameters:
		  priority_io: high
		  repl: "3"
		  io_profile: "cms"
		provisioner: kubernetes.io/portworx-volume
		reclaimPolicy: Delete
		volumeBindingMode: Immediate
		EOF

portworx-rwx-gp-sc	GP replica 1	# General Purpose, 1 Replica - RWX volumes for TESTING ONLY.
		cat <<EOF | oc create -f -
		kind: StorageClass
		apiVersion: storage.k8s.io/v1
		metadata:
		  name: portworx-rwx-gp-sc
		provisioner: kubernetes.io/portworx-volume
		parameters:
		  repl: "1"
		  priority_io: "high"
		  sharedv4: "true"
		  io_profile: db_remote
		  disable_io_profile_protection: "1"
		allowVolumeExpansion: true
		volumeBindingMode: Immediate
		reclaimPolicy: Delete
		EOF

portworx-shared-gp	Shared GP high IOPS	
		# General Purpose, 3 Replicas - RWX volumes - placeholder SC portworx-shared-gp for upgrade purposes
		cat <<EOF | oc create -f -
		apiVersion: storage.k8s.io/v1
		kind: StorageClass
		metadata:
		  name: portworx-shared-gp
		parameters:
		  priority_io: high
		  repl: "3"
		  sharedv4: "true"
		  io_profile: db_remote
		  disable_io_profile_protection: "1"
		allowVolumeExpansion: true
		provisioner: kubernetes.io/portworx-volume
		reclaimPolicy: Retain
		volumeBindingMode: Immediate
		EOF

portworx-gp3-sc	GP3 replica 3	# General Purpose, 3 Replicas RWO volumes rabbitmq and redis-ha - New Install
		cat <<EOF | oc create -f -
		apiVersion: storage.k8s.io/v1
		kind: StorageClass
		metadata:
		  name: portworx-gp3-sc
		parameters:
		  priority_io: high
		  repl: "3"
		  io_profile: "db_remote"
		  disable_io_profile_protection: "1"
		allowVolumeExpansion: true
		provisioner: kubernetes.io/portworx-volume
		reclaimPolicy: Retain
		volumeBindingMode: Immediate
		EOF

portworx-nonshared-gp2	GP2 nonshared throughput optimized	# General Purpose, 3 Replicas RWO volumes rabbitmq and redis-ha - placeholder SC portworx-nonshared-gp2 for upgrade purposes
		cat <<EOF | oc create -f -
		apiVersion: storage.k8s.io/v1
		kind: StorageClass
		metadata:
		  name: portworx-nonshared-gp2
		parameters:
		  priority_io: high
		  repl: "3"
		  io_profile: "db_remote"
		  disable_io_profile_protection: "1"
		allowVolumeExpansion: true
		provisioner: kubernetes.io/portworx-volume
		reclaimPolicy: Retain
		volumeBindingMode: Immediate
		EOF

Storage class	Storage type	Storage class definitions

`portworx-shared-gp1`	Shared GP high iops	`#Shared gp high iops:`
		`cat <<EOF | oc create -f -`
		`apiVersion: storage.k8s.io/v1`
		`kind: StorageClass`
		`metadata:`
		`  name: portworx-shared-gp1`
		`parameters:`
		`  priority_io: high`
		`  repl: "1"`
		`  sharedv4: "true"`
		`allowVolumeExpansion: true`
		`provisioner: kubernetes.io/portworx-volume`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

January 2021 release or later `portworx-db-gp`	DB GP replica 1 for MongoDB	`# gp db`
		`cat <<EOF | oc create -f -`
		`apiVersion: storage.k8s.io/v1`
		`kind: StorageClass`
		`metadata:`
		`  name: portworx-db-gp`
		`parameters:`
		`  io_profile: "db_remote"`
		`  repl: "1"`
		`  disable_io_profile_protection: "1"`
		`allowVolumeExpansion: true`
		`provisioner: kubernetes.io/portworx-volume`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

`portworx-db-gp2-sc`	DB GP2 replica 3	`# General Purpose for Databases, 2 Replicas - MongoDB - (Implemented application-level redundancy)`
		`cat <<EOF | oc create -f -`
		`apiVersion: storage.k8s.io/v1`
		`kind: StorageClass`
		`metadata:`
		`  name: portworx-db-gp2-sc`
		`parameters:`
		`  priority_io: "high"`
		`  io_profile: "db_remote"`
		`  repl: "2"`
		`  disable_io_profile_protection: "1"`
		`allowVolumeExpansion: true`
		`provisioner: kubernetes.io/portworx-volume`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

`portworx-db-gp3-sc`	DB GP3 replica 3	`# General Purpose for Databases, 3 Replicas`
		`cat <<EOF | oc create -f -`
		`apiVersion: storage.k8s.io/v1`
		`kind: StorageClass`
		`metadata:`
		`  name: portworx-db-gp3-sc`
		`parameters:`
		`  io_profile: "db_remote"`
		`  repl: "3"`
		`  priority_io: "high"`
		`  disable_io_profile_protection: "1"`
		`allowVolumeExpansion: true`
		`provisioner: kubernetes.io/portworx-volume`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

`portworx-db2-rwx-sc`	Db2® and Db2 Warehouse (System and Backup Storage) and Data Virtualization	`# DB2 RWX shared volumes for System Storage, backup storage, future load storage, and future diagnostic logs storage`
		`cat <<EOF | oc create -f -`
		`allowVolumeExpansion: true`
		`apiVersion: storage.k8s.io/v1`
		`kind: StorageClass`
		`metadata:`
		`  name: portworx-db2-rwx-sc`
		`parameters:`
		`  io_profile: cms`
		`  block_size: 4096b`
		`  nfs_v4: "true"`
		`  repl: "3"`
		`  sharedv4: "true"`
		`  priority_io: high`
		`provisioner: kubernetes.io/portworx-volume`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

Storage class	Storage type	Storage class definitions

`portworx-db2-rwo-sc`	Db2 andDb2 Warehouse (User Storage)
	Watson™ Knowledge Catalog Db2 Metastore
		`# Db2 RWO volumes SC for user storage, future transaction logs storage, future archive/mirrors logs storage. This is also used for WKC DB2 Metastore`
		`cat <<EOF | oc create -f -`
		`allowVolumeExpansion: true`
		`apiVersion: storage.k8s.io/v1`
		`kind: StorageClass`
		`metadata:`
		`  name: portworx-db2-rwo-sc`
		`parameters:`
		`  block_size: 4096b`
		`  io_profile: db_remote`
		`  priority_io: high`
		`  repl: "3"`
		`  sharedv4: "false"`
		`  disable_io_profile_protection: "1"`
		`provisioner: kubernetes.io/portworx-volume`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

`portworx-db2-sc`	Watson Knowledge Catalog Db2 Metastore (Upgrade)	`# WKC DB2 Metastore - SC portworx-db2-sc for upgrade purposes`
		`cat <<EOF | oc create -f -`
		`allowVolumeExpansion: true`
		`apiVersion: storage.k8s.io/v1`
		`kind: StorageClass`
		`metadata:`
		`  name: portworx-db2-sc`
		`parameters:`
		`  io_profile: "db_remote"`
		`  priority_io: high`
		`  repl: "3"`
		`  disable_io_profile_protection: "1"`
		`provisioner: kubernetes.io/portworx-volume`
		`reclaimPolicy: Retain`
		`volumeBindingMode: Immediate`
		`EOF`

Setting up NFS storage

By default, NFS does not support dynamic storage provisioning. If you plan to use Cloud Pak for Data for persistent storage, you must set up your NFS storage before you install Cloud Pak for Data.

Supported storage topology

If you use NFS storage, you can use one of following cluster configurations:

- NFS on a dedicated node in the same VLAN as the cluster (recommended)
- An external NFS server
  If you select this option, configure the server based on your availability requirements and ensure that you have a sufficiently fast network connection (at least 1 GB) to reduce latency and ensure performance.

Configuration requirements

Ensure that the following statements are true:

- All of the nodes in the cluster must have access to mount the NFS server.
- All of the nodes in the cluster must have read/write access to the NFS server.
- Containerized processes must have read/write access to the NFS server.
  Important: Containerized processes create files that are owned by various UIDs. (In Cloud Pak for Data, most services use long UIDs between 1000320900 and 1000361000.) If you restrict access to the NFS served to specific UIDs, you might encounter errors when installing or running Cloud Pak for Data.
- If you use NFS as the storage for a database service, ensure that the storage has sufficient throughput. For details, see the appropriate topic for your environment:
  - Db2®: Requirements for Db2 on SELinux
  - Db2 Warehouse: Requirements for Db2 Warehouse on SELinux

Setting the NFS export

Ensure that the NFS export is set to **`no_root_squash`**.

If you are using NFS on IBM® Cloud, follow the instructions in Implementing **`no_root_squash`** for NFS.

Configuring dynamic storage

By default, Red Hat® OpenShift® does not include a provisioner plug-in to create an NFS storage class. To dynamically provision NFS storage, use the Kubernetes NFS-Client Provisioner, which is available from the Kubernetes SIGs organization on GitHub.

Permissions you need for this task
You must be a cluster administrator.

Important: The following steps assume you have an existing NFS server. Ensure that you know how to connect to your NFS server. At a minimum, you must have the hostname of the server.

To configure dynamic storage:

1. Ensure that your NFS server is accessible from your Red Hat OpenShift Container Platform cluster.
2. Clone the https://github.com/kubernetes-sigs/nfs-subdir-external-provisioner/tree/master/deploy repository.
3. Download all of the files in the deploy directory (in the Kubernetes NFS-Client Provisioner repository).
4. Open a **bash** shell and change to the deploy directory of the repository.
5. Log in to your Red Hat OpenShift Container Platform cluster as a user with sufficient permissions to complete the task:

   ```
   oc login OpenShift_URL:port
   ```

6. Authorize the provisioner by running the following commands.
   a. Create the required role based access control.
   If you plan to deploy the NFS provisioner to a project other than the **default** project, you must replace each instance of **default** in the rbac.yaml file before you run this command.

   ```
   oc create -f rbac.yaml
   ```

   b. Add the **nfs-client-provisioner** security context constraint to the **system** service account.
   If you plan to deploy the NFS provisioner to a project other than the **default** project, replace **default** in the following command.

   ```
   oc adm policy add-scc-to-user hostmount-anyuid system:serviceaccount:default:nfs-client-provisioner
   ```

7. Edit the deployment.yaml file to specify the following information:
   - The project (namespace) where the NFS provisioner is deployed
   - The hostname of your NFS server.
   - The path where you want to dynamically provision storage on your NFS server.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nfs-client-provisioner
  labels:
    app: nfs-client-provisioner
  namespace: default         # Specify the namespace where the NFS provisioner is deployed
spec:
  replicas: 1
  strategy:
    type: Recreate
  selector:
    matchLabels:
      app: nfs-client-provisioner
  template:
    metadata:
      labels:
        app: nfs-client-provisioner
    spec:
      serviceAccountName: nfs-client-provisioner
      containers:
        - name: nfs-client-provisioner
          image: quay.io/external_storage/nfs-client-provisioner:latest
          volumeMounts:
            - name: nfs-client-root
              mountPath: /persistentvolumes
          env:
            - name: PROVISIONER_NAME
              value: nfs-storage
            - name: NFS_SERVER
              value: MyNFSHostname        # Specify the host name of your NFS server
            - name: NFS_PATH
              value: /nfs/cpshare/        # Specify the path where you want to provision storage
      volumes:
        - name: nfs-client-root
          nfs:
            server: MyNFSHostname         # Specify the host name of your NFS server
            path: /nfs/cpshare/           # Specify the path where you want to provision storage
```

8. Deploy the NFS provisioner:

   ```
   oc create -f deployment.yaml
   ```

9. Edit the class.yaml file to specify the names of the storage classes that you want to create. The following example includes the recommended **managed-nfs-storage** storage class:

   ```
   apiVersion: storage.k8s.io/v1
   kind: StorageClass
   metadata:
     name: managed-nfs-storage        # Recommended storage class name
   provisioner: nfs-client-provisioner        # This name must match the value you specified in the deployment.yaml
   parameters:
     archiveOnDelete: "false"
   ```

   For a complete list of parameters, see Deploying your storage class in the NFS provisioner documentation.

10. Create the storage class:

    ```
    oc create -f class.yaml
    ```

11. Verify that the NFS provisioner is running correctly:
    a. Create a test persistent volume claim (PVC).
    Note: The test-claim.yaml file uses the **managed-nfs-storage** storage class.

    ```
    oc create -f test-claim.yaml
    ```

b. On your NFS server, verify that the share directory, which you specified in the deployment.yaml file, contains a file called SUCCESS.
c. Remove the test PVC:

```
oc delete -f test-pod.yaml test-claim.yaml
```

Creating projects (namespaces) on Red Hat OpenShift Container Platform

Before you install IBM® Cloud Pak for Data on Red Hat® OpenShift® Container Platform, a cluster administrator should create the OpenShift projects (Kubernetes namespaces) where you plan to deploy the Cloud Pak for Data software.

Permissions you need for this task
You must be a cluster administrator.
When you need to complete this task
You must complete this task the first time you install Cloud Pak for Data.
You might need to complete this task if you decide to install additional instances of Cloud Pak for Data on your cluster or decide to deploy a service in a tethered namespace.

About this task

For information on supported project configurations, see Architecture for IBM Cloud Pak for Data.

Use the following table to determine which projects (namespaces) you need to create.

Project	Description
`ibm-common-services` project	Required for all installations. The default project where IBM Cloud Pak® foundational services is installed. **`ibm-common-services`** is the recommended name and is used in various installation commands. If you want to install IBM Cloud Pak foundational services in a different project, you must create configmap. For details, see Installing IBM Cloud Pak foundational services in a custom namespace. If IBM Cloud Pak foundational services is already installed on your cluster, identify the project where it is installed. Additional software that might be installed in this project Depending on the software that you plan to install and the installation method that you use, the following software might also be installed in the **`ibm-common-services`** project: • The IBM Cloud Pak for Data scheduling service If you need to install the scheduling service, it is recommended that you install it in the same project as IBM Cloud Pak foundational services. • The IBM Cloud Pak for Data platform operator If you decide to use the express installation method, the IBM Cloud Pak for Data platform operator will be installed in this project. • IBM Cloud Pak for Data service operators If you decide to use the express installation method, the service operators will be installed in this project.
`cpd-operators` project	Required for specialized installations. **`cpd-operators`** is the recommended name and is used in various installation commands. In a specialized installation, the IBM Cloud Pak foundational services operators are installed in the **`ibm-common-services`** project and the Cloud Pak for Data operators are installed in a separate project (typically **`cpd-operators`**). Each project has a dedicated: • Operator group, which specifies the **`OwnNamespace`** installation mode • **`NamespaceScope Operator`**, which allows the operators in the project to manage operators and service workloads in specific projects In this way, you can specify different settings for the IBM Cloud Pak foundational services and for the Cloud Pak for Data operators.
`cpd-instance` projects	At least one project is required for all installations. The project where the Cloud Pak for Data control plane is installed. (The Cloud Pak for Data control plane is installed in a *separate* project from the operators.) If you plan to install multiple install multiple instances of Cloud Pak for Data, you must create one project for each instance. ***`cpd-instance`*** is an example. You can use any project name. ***`cpd-instance`*** is used as a placeholder in various installation commands. Most services are installed in the same project as the Cloud Pak for Data control plane. Review the documentation for the services that you plan to deploy to determine whether you must create any additional projects. For details, see Services.
`cpd-instance-tether` projects	Required or supported for some services. A few services can be installed in tethered projects. A *tethered project* is managed by the Cloud Pak for Data control plane but is otherwise isolated from Cloud Pak for Data and the other services that are installed in that project. ***`cpd-instance-tether`*** is an example. You can use any project name. ***`cpd-instance-tether`*** is used as a placeholder in various installation commands. For information on which services can be installed in tethered projects, see Multitenancy support. If you want to install a service in a tethered project, you must create the tethered project before you install the service.

After you decide which projects you need to create, review the following information to ensure that you understand the security considerations that you need to take into account:

Project	Security considerations
ibm-common-services project	Operator group The **ibm-common-services** project uses the **OwnNamespace** installation mode. See the Procedure after this table for information on creating the operator group. Namespace scope The **ibm-common-services** project needs to be able to watch the project or projects where Cloud Pak for Data is deployed. IBM Cloud Pak foundational services includes the **IBM NamespaceScope Operator**, which allows the operators in the **ibm-common-services** project to manage operators and service workloads in specific projects. When you install Cloud Pak for Data or create a tethered namespace, you submit an operand request to grant permission to the operators in the **ibm-common-services** project to watch over the project (for example ***cpd-instance*** or ***cpd-instance-tether***). By default, the **IBM NamespaceScope Operator** has *cluster permissions* so that role binding projections can be completed automatically. However, you can optionally remove the cluster permissions from the **IBM NamespaceScope Operator** and manually authorize the projections. For details, see Authorizing foundational services to perform operations on workloads in a namespace. SCCs Follow the guidance Security context constraints (SCCs) in the IBM Cloud Pak foundational services documentation. Express installations only The Cloud Pak for Data control plane and most Cloud Pak for Data services use the **restricted** SCC. However, a few services require custom SCCs. For details, see Creating custom security context constraints for services.
cpd-operators project	Operator group The **cpd-operators** project uses the **OwnNamespace** installation mode. See the Procedure after this table for information on creating the operator group. Namespace scope The **cpd-operators** project needs to be able to watch the project or projects where Cloud Pak for Data is deployed. When you prepare your cluster, you create an operator subscription for the **IBM NamespaceScope Operator** in the **cpd-operators** project. The **IBM NamespaceScope Operator** allows the operators in the **cpd-operators** project to manage operators and service workloads in specific projects. When you install Cloud Pak for Data or create a tethered namespace, you submit an operand request to grant permission to the operators in the **cpd-operators** project to watch over the project (for example ***cpd-instance*** or ***cpd-instance-tether***). By default, the **IBM NamespaceScope Operator** has *cluster permissions* so that role binding projections can be completed automatically. However, you can optionally remove the cluster permissions from the **IBM NamespaceScope Operator** and manually authorize the projections. For details, see Authorizing foundational services to perform operations on workloads in a namespace. SCCs The Cloud Pak for Data control plane and most Cloud Pak for Data services use the **restricted** SCC. However, a few services require custom SCCs. For details, see Creating custom security context constraints for services.
cpd-instance projects	Operator group Not applicable. Namespace scope Not applicable. SCCs The Cloud Pak for Data control plane and most Cloud Pak for Data services use the **restricted** SCC. However, a few services require custom SCCs. For details, see Creating custom security context constraints for services.
cpd-instance-tether projects	Operator group Not applicable. Namespace scope Not applicable. SCCs The Cloud Pak for Data control plane and most Cloud Pak for Data services use the **restricted** SCC. However, a few services require custom SCCs. For details, see Creating custom security context constraints for services.

Procedure

To create the necessary projects for your environment:

1. Log in to your Red Hat OpenShift Container Platform as a cluster administrator:

   ```
   oc login OpenShift:port
   ```

2. Run the following command to create a project:

   ```
   oc new-project project-name
   ```

   Repeat this step for *each* project that you need to create.

3. Create the appropriate operator groups based on the type of installation method you are using:

Installation method	Operator group contents

Installation method	Operator group contents

Express installation	a. If IBM Cloud Pak foundational services is not installed, create the operator group for the IBM Cloud Pak foundational services project. The following example uses the recommended project name (**ibm-common-services**):
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha2`
	`kind: OperatorGroup`
	`metadata:`
	`  name: operatorgroup`
	`  namespace: ibm-common-services`
	`spec:`
	`  targetNamespaces:`
	`  - ibm-common-services`
	`EOF`

Specialized installation	a. If IBM Cloud Pak foundational services is not installed, create the operator group for the IBM Cloud Pak foundational services project. The following example uses the recommended project name (**ibm-common-services**):
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha2`
	`kind: OperatorGroup`
	`metadata:`
	`  name: operatorgroup`
	`  namespace: ibm-common-services`
	`spec:`
	`  targetNamespaces:`
	`  - ibm-common-services`
	`EOF`
	
	b. Create the operator group for the IBM Cloud Pak for Data platform operator project. The following example uses the recommended project name (**cpd-operators**):
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha2`
	`kind: OperatorGroup`
	`metadata:`
	`  name: operatorgroup`
	`  namespace: cpd-operators`
	`spec:`
	`  targetNamespaces:`
	`  - cpd-operators`
	`EOF`

Previous topic: Setting up shared persistent storage
Next topic: Obtaining your IBM entitlement API key

Obtaining your IBM entitlement API key

The IBM entitlement API key enables you to pull software images from the IBM® Entitled Registry, either for installation or for mirroring.

All Cloud Pak for Data images are accessible from the IBM Entitled Registry. You must decide whether you will install the images directly from the IBM Entitled Registry or whether you will mirror the images to your local container registry.

IBM entitlement API key

You must have your IBM entitlement API key to access images in the IBM Entitled Registry.

After you purchase Cloud Pak for Data, an entitlement API key for the software is associated with your My IBM account. You need this key to complete the Cloud Pak for Data installation. To obtain the entitlement key, complete the following steps:

1. Log in to Container software library on My IBM with the IBM ID and password that are associated with the entitled software.
2. On the Get entitlement key tab, select Copy key to copy the entitlement key to the clipboard.
3. Save the API key in a text file.

Previous topic: Creating projects (namespaces) on Red Hat OpenShift Container Platform
Next topic: Mirroring images to your container registry

Mirroring images to your container registry

IBM® Cloud Pak for Data images are accessible from the IBM Entitled Registry. In most situations, it is strongly recommended that you mirror the necessary software images from the IBM Entitled Registry to a private container registry.

Important: You must mirror the necessary images to your container registry in the following situations:

- Your cluster is air-gapped (also called an offline or disconnected cluster)
- Your cluster uses an *allowlist* to permit direct access by specific sites and the allowlist does not include the IBM Entitled Registry
- Your cluster uses a *blocklist* to prevent direct access by specific sites and the blocklist includes the IBM Entitled Registry

The only situation in which you might consider pulling images directly from the IBM Entitled Registry is when your cluster is not air-gapped, your network is extremely reliable, and latency is not a concern. However, for predictable and reliable performance, you should mirror the images to a private container registry.

Setting up a private container registry

For details about which container registries you can use with Red Hat® OpenShift® Container Platform, see Registry options in the Red Hat OpenShift Container Platform documentation.

Your private container registry must meet the following requirements:

- Support the Docker Image Manifest Version 2, Schema 2
- Allow path separators in image names
- Be in close proximity to your Red Hat OpenShift Container Platform cluster

In addition, the registry must be accessible from all of the nodes in the cluster and all of the nodes must have permission to push to and pull from the container registry.

Restriction: You cannot use the integrated OpenShift Container Platform registry. It does not support multi-architecture images and is not compliant with the Docker Image Manifest Version 2, Schema 2.

Image prefixes

IBM Cloud Pak software uses the following prefixes to identify images:

Tag	Used for
`cp.icr.io/cp`	Images that are pulled from the IBM Entitled Registry that require an entitlement key to download. Most of the IBM Cloud Pak for Data software uses this tag.
`icr.io/cpopen`	Publicly available images that are provided by IBM and that don't require an entitlement key to download. The IBM Cloud Pak for Data operators use this tag.
`quay.io/opencloudio`	IBM open source images that are available on quay.io. The IBM Cloud Pak® foundational services software uses this tag.

Ensure that:

- Your private container registry is configured to allow these prefixes
- The credentials that you will use to push images to the private container registry can push images with these prefixes

Methods for mirroring images

There are several ways that you can mirror images from the IBM Entitled Registry to your private container registry. Choose the most appropriate method for your environment:

Method	Description	Connected clusters	Air-gapped clusters
Portable compute device	**Example:** A laptop that you can move behind your firewall is a portable compute device. High-level process using a portable compute device: 1. Create an intermediary container registry on a portable compute device that is connected to the internet. 2. From the portable compute device, mirror images from the IBM Entitled Registry to the intermediary container registry. 3. Bring the device behind your firewall and mirror the images from the intermediary container registry to the container registry that is accessible from the Red Hat OpenShift Container Platform cluster. For the full process, see Mirroring images with an intermediary container registry.		✓
File transfer	**Example:** You can either use a portable storage device, such as a USB drive, or use `scp` or `sftp` to move images behind your firewall. High-level process using a file transfer: 1. Create an intermediary container registry. If you are using a portable storage device, create the registry on the storage device. 2. From a workstation that can connect to the internet and the intermediary container registry, mirror the images from the IBM Entitled Registry to the intermediary container registry. 3. Move the files and or the storage device behind your firewall. 4. Set up a workstation behind the firewall to mirror the images to the container registry that is accessible from the Red Hat OpenShift Container Platform cluster. For the full process, see Mirroring images with an intermediary container registry.		✓
Bastion node	**Example:** A server with access to both the public internet and the container registry that is accessible from the Red Hat OpenShift Container Platform cluster. High-level process using a bastion node: 1. From the bastion node, replicate the images from the IBM Entitled Registry to the container registry that is accessible from the Red Hat OpenShift Container Platform cluster. For the full process, see Mirroring images with a bastion node.	✓	✓

Mirroring images to a private container registry

Complete the appropriate task for your environment:

- **Mirroring images with a bastion node**
  If your Red Hat OpenShift Container Platform cluster is air-gapped, you must mirror the software images that you need to a private container registry that is accessible from the cluster. You can use a bastion node that is connected to the internet and to the private registry to mirror the images from the IBM Entitled Registry.

- **Mirroring images with an intermediary container registry**
  If your Red Hat OpenShift Container Platform cluster is air-gapped, you must mirror the software images that you need to a private container registry that is accessible from the cluster. You can use an intermediary container registry to mirror the images from the IBM Entitled Registry to a private container registry.

Previous topic: Obtaining your IBM entitlement API key
Next topic: Configuring your cluster to pull Cloud Pak for Data images

Mirroring images with a bastion node

If your Red Hat® OpenShift® Container Platform cluster is air-gapped, you must mirror the software images that you need to a private container registry that is accessible from the cluster. You can use a bastion node that is connected to the internet and to the private registry to mirror the images from the IBM® Entitled Registry.

Important: Use a Linux x86-64 system with Red Hat Enterprise Linux® to mirror the images.
The system must be able to access the following sites:

- Red Hat Quay.io (https://quay.io:443)
- GitHub (https://github.com)
  If your company does not permit access to GitHub, contact IBM Support for assistance.

- IBM Entitled Registry (http://icr.io:443)
  To validate that you can connect, run the following command:

  ```
  curl -v https://icr.io
  ```

  The command should return the following message:

  ```
  * Connected to icr.io (169.60.98.86) port 443 (#0)
  ```

Procedure

Complete the following tasks to mirror the images to your container registry:

- 1. Downloading and installing the software needed to mirror images
- 2. Setting up your environment to download CASE packages
- 3. Downloading the Cloud Pak for Data CASE package
- 4. Configuring credentials for mirroring images
- 5. Downloading shared cluster component CASE packages
- 6. Downloading service CASE packages
- 7. Mirroring the images to the private registry

1. Downloading and installing the software needed to mirror images

To use a connected bastion node, you must install the following software on the system:

Prerequisite	Purpose
OpenShift CLI	Required to interact with your Red Hat OpenShift Container Platform cluster.
IBM Cloud Pak® CLI (cloudctl)	Required to download images from the IBM Entitled Registry.
`httpd-tools`	Required to run the IBM Cloud Pak CLI (cloudctl).
`skopeo` Version 1.2.0 or later	Required to run the IBM Cloud Pak CLI (cloudctl).

To install the prerequisite software:

1. To install the OpenShift CLI, see Getting started with the OpenShift CLI in the Red Hat documentation.
2. To install the IBM Cloud Pak CLI (cloudctl):
   a. Download the **`cloudctl`** software from the IBM/cloud-pak-cli repository on GitHub. Ensure that you download the appropriate package for your workstation:

      ```
      cloudctl-operating-system-architecture.tar.gz
      ```

   b. Extract the contents of the archive file:

      ```
      tar -xzf archive-name
      ```

   c. Change to the directory where you extracted the file and make the file executable:

      ```
      chmod 775 cloudctl-architecture
      ```

   d. Move the file to the /usr/local/bin directory:

      ```
      mv cloudctl-architecture /usr/local/bin/cloudctl
      ```

   e. Confirm that the IBM Cloud Pak CLI (cloudctl) is installed:

      ```
      cloudctl --help
      ```

   Tip: Additional guidance for validating the archive file is available in the IBM/cloud-pak-cli repository.
3. To install **`httpd-tools`**, run the following command:

   ```
   yum install httpd-tools
   ```

4. To install **`skopeo`**, see Installing from packages in the **`skopeo`** repository on GitHub.

2. Setting up your environment to download CASE packages

A Container Application Software for Enterprises (CASE) package is an archive file that describes a containerized component of Cloud Pak for Data.

There are CASE packages for:

- IBM Cloud Pak foundational services
- IBM Cloud Pak for Data control plane
- Each IBM Cloud Pak for Data service
- Software dependencies for the control plane and services

Each CASE package includes:

- Metadata about the component
- An inventory of the container images that are required to deploy the component
- References to any software dependencies
- The scripts needed to mirror the images to a private registry

Before you can mirror the images a private registry, you must download the CASE packages for the software that you plan to install.

To set up your environment:

1. Identify or create the directory where you want to store the CASE packages on the system.
   Important: Keep the following requirements in mind:
   - You must have sufficient storage in the directory.
   - You must use a persistent directory. Using a persistent directory prevents you from transferring files more than once. Additionally, if you use a persistent directory, you can run the mirror process multiple times or on a schedule.

   For example, you could create a directory called offline:

   ```
   mkdir -p $HOME/offline
   ```

2. Set the following environment variables:

   ```
   export OFFLINEDIR=$HOME/offline
   export CASE_REPO_PATH=https://github.com/IBM/cloud-pak/raw/master/repo/case
   ```

   OFFLINEDIR is the directory that you created to store CASE packages. Replace **$HOME/offline** with the appropriate value for your environment.

3. Downloading the Cloud Pak for Data CASE package

1. Run the following command to download the IBM Cloud Pak for Data platform operator package:

   ```
   cloudctl case save \
     --case ${CASE_REPO_PATH}/ibm-cp-datacore-2.0.1.tgz \
     --outputdir ${OFFLINEDIR} \
     --no-dependency
   ```

4. Configuring credentials for mirroring images

The IBM Cloud Pak CLI (cloudctl) includes an action called **configure-cred-airgap**. Run the appropriate commands to store the credentials that you will need to mirror images to the private container registry. The command stores the credentials to the following file on your local file system: $HOME/.airgap/secrets.

To configure the credentials that you need to mirror software images:

1. Store the IBM Entitled Registry credentials by running the following command:

   ```
   cloudctl case launch \
     --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.1.tgz \
     --inventory cpdPlatformOperator \
     --action configure-creds-airgap \
     --args "--registry cp.icr.io --user cp --pass entitlement-key --inputDir ${OFFLINEDIR}"
   ```

   Replace *entitlement-key* with your entitlement key. For details, see IBM entitlement API key.

2. Store the private container registry credentials:
   a. Work with your container registry administrator to identify the values for the following parameters:

      PRIVATE_REGISTRY_USER
      The username of a user who has the required privileges to *push* images to the private registry.
      PRIVATE_REGISTRY_PASSWORD
      The password of the user who has the required privileges to *push* images to the private registry.
      PRIVATE_REGISTRY
      The location of the private registry.

   b. Set environment variables for the parameters:

      ```
      export PRIVATE_REGISTRY_USER=username
      export PRIVATE_REGISTRY_PASSWORD=password
      export PRIVATE_REGISTRY=private-registry-location
      ```

   c. Run the following command to store the credentials:

      ```
      cloudctl case launch \
        --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.1.tgz \
        --inventory cpdPlatformOperator \
        --action configure-creds-airgap \
        --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD}"
      ```

5. Downloading shared cluster component CASE packages

Determine which shared cluster components you need to install on your cluster and download the appropriate CASE packages.

Shared cluster component	CASE download command

IBM Cloud Pak foundational services
Download this package if IBM Cloud Pak foundational services is not installed on the cluster.
	`cloudctl case save \`
	`--case ${CASE_REPO_PATH}/ibm-cp-common-services-1.4.1.tgz \`
	`--outputdir ${OFFLINEDIR}`

Scheduling service
Download this package if you plan to install Watson™ Machine Learning Accelerator or if you want to use the quota enforcement feature.
	`cloudctl case save \`
	`--case ${CASE_REPO_PATH}/ibm-cpd-scheduling-1.2.1.tgz \`
	`--outputdir ${OFFLINEDIR}`

6. Downloading service CASE packages

Decide which services you plan to install on your cluster and download the appropriate CASE packages.

- Analytics Engine Powered by Apache Spark

  ```
  cloudctl case save \
  --case ${CASE_REPO_PATH}/ibm-analyticsengine-4.0.0.tgz \
  --outputdir ${OFFLINEDIR}
  ```

- Cognos Analytics

  ```
  cloudctl case save \
  --case ${CASE_REPO_PATH}/ibm-cognos-analytics-prod-4.0.0.tgz \
  --outputdir ${OFFLINEDIR}
  ```

- Cognos Dashboards

  ```
  cloudctl case save \
  --case ${CASE_REPO_PATH}/ibm-cde-2.0.0.tgz \
  --outputdir ${OFFLINEDIR}
  ```

- Data Refinery

  ```
  cloudctl case save \
  --case ${CASE_REPO_PATH}/ibm-datarefinery-1.0.0.tgz \
  --outputdir ${OFFLINEDIR}
  ```

- Data Virtualization

  ```
  cloudctl case save \
  --case ${CASE_REPO_PATH}/ibm-dv-case-1.7.0.tgz \
  --outputdir ${OFFLINEDIR}
  ```

- DataStage

  Download the appropriate package based on your license:

  DataStage Enterprise

  ```
  cloudctl case save \
  --case ${CASE_REPO_PATH}/ibm-datastage-enterprise-4.0.0.tgz \
  --outputdir ${OFFLINEDIR}
  ```

  DataStage Enterprise Plus

  ```
  cloudctl case save \
  --case ${CASE_REPO_PATH}/ibm-datastage-4.0.0.tgz \
  --outputdir ${OFFLINEDIR}
  ```

- Db2

  ```
  cloudctl case save \
  --case ${CASE_REPO_PATH}/ibm-db2oltp-4.0.0.tgz \
  --outputdir ${OFFLINEDIR}
  ```

- Db2 Big SQL

  ```
  cloudctl case save \
  --case ${CASE_REPO_PATH}/ibm-bigsql-case-7.2.0.tgz \
  --outputdir ${OFFLINEDIR}
  ```

- Db2 Data Gate

  ```
  cloudctl case save \
  --case ${CASE_REPO_PATH}/ibm-datagate-prod-4.0.0.tgz \
  --outputdir ${OFFLINEDIR}
  ```

- Db2 Data Management Console

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-dmc-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Db2 Event Store

Not applicable. Contact IBM Software support if you plan to install this service.

- Db2 Warehouse

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-db2wh-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Decision Optimization

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-dods-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- EDB Postgres

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-cpd-edb-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Execution Engine for Apache Hadoop

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-hadoop-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Financial Services Workbench

Not applicable.

- IBM Match 360 with Watson

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-mdm-1.0.1.tgz \
--outputdir ${OFFLINEDIR}
```

- Jupyter Notebooks with Python 3.7 for GPU

The same package is used for the Jupyter Notebooks with Python 3.7 for GPU service and the Jupyter Notebooks with R 3.6 service. You only need to download the package once.

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-wsl-runtimes-1.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Jupyter Notebooks with R 3.6

The same package is used for the Jupyter Notebooks with Python 3.7 for GPU service and the Jupyter Notebooks with R 3.6 service. You only need to download the package once.

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-wsl-runtimes-1.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- MongoDB

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-cpd-mongodb-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- OpenPages

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-openpages-2.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

If you want OpenPages to automatically provision a Db2 database, you must also download the following package: :

Db2 as a service

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-db2aaservice-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Planning Analytics

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-planning-analytics-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Product Master

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-productmaster-1.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- RStudio Server with R 3.6

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-rstudio-1.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- SPSS Modeler

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-spss-1.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Virtual Data Pipeline

Not applicable

- Watson Knowledge Catalog

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-wkc-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Watson Machine Learning

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-wml-cpd-4.0.1.tgz \
--outputdir ${OFFLINEDIR}
```

- Watson Machine Learning Accelerator

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-wml-accelerator-2.3.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Watson OpenScale

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-watson-openscale-2.0.1.tgz \
--outputdir ${OFFLINEDIR}
```

- Watson Studio

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-wsl-2.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

7. Mirroring the images to the private registry

To mirror the images:

1. Mirror the Cloud Pak for Data control plane images to the private container registry:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.1.tgz \
  --inventory cpdPlatformOperator \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir
${OFFLINEDIR}"
```

2. Mirror the images for *each* of the shared cluster components that you downloaded:

Shared cluster component	Image mirroring command

IBM Cloud Pak foundational services	`cloudctl case launch \`
	`  --case ${OFFLINEDIR}/ibm-cp-common-services-1.4.1.tgz \`
	`  --inventory ibmCommonServiceOperatorSetup \`
	`  --action mirror-images \`
	`  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"`

Shared cluster component	Image mirroring command

Scheduling service	`cloudctl case launch \`
	`  --case ${OFFLINEDIR}/ibm-cpd-scheduling-1.2.1.tgz \`
	`  --inventory schedulerSetup \`
	`  --action mirror-images \`
	`  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"`

3. Mirror the images for *each* of the services that you downloaded.

- Analytics Engine Powered by Apache Spark

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-analyticsengine-4.0.0.tgz \
  --inventory analyticsengineOperatorSetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Cognos Analytics

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cognos-analytics-prod-4.0.0.tgz \
  --inventory ibmCaOperatorSetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Cognos Dashboards

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cde-2.0.0.tgz \
  --inventory cdeOperatorSetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Data Refinery

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-datarefinery-1.0.0.tgz \
  --inventory datarefinerySetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Data Virtualization

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-dv-case-1.7.0.tgz \
  --inventory dv \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- DataStage

  Mirror the images for the package that you downloaded

  DataStage Enterprise

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-datastage-enterprise-4.0.0.tgz \
    --inventory datastageOperatorSetup \
    --action mirror-images \
    --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
  inputDir ${OFFLINEDIR}"
  ```

  DataStage Enterprise Plus

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-datastage-4.0.0.tgz \
    --inventory datastageOperatorSetup \
    --action mirror-images \
    --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
  inputDir ${OFFLINEDIR}"
  ```

- Db2

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-db2oltp-4.0.0.tgz \
  --inventory db2oltpOperatorSetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Db2 Big SQL

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-bigsql-case-7.2.0.tgz \
  --inventory bigsql \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Db2 Data Gate

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-datagate-prod-4.0.0.tgz \
  --inventory datagateOperatorSetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Db2 Data Management Console

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-dmc-4.0.0.tgz \
  --inventory dmcOperatorSetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Db2 Event Store

Not applicable. Contact IBM Software support if you plan to install this service.

- Db2 Warehouse

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-db2wh-4.0.0.tgz \
  --inventory db2whOperatorSetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Decision Optimization

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-dods-4.0.0.tgz \
  --inventory dodsOperatorSetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- EDB Postgres

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cpd-edb-4.0.0.tgz \
  --inventory ibmCPDEDBSetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Execution Engine for Apache Hadoop

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-hadoop-4.0.0.tgz \
  --inventory hadoopSetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Financial Services Workbench

Not applicable.

- IBM Match 360 with Watson

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-mdm-1.0.1.tgz \
  --inventory mdmOperator \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Jupyter Notebooks with Python 3.7 for GPU

The same package is used for the Jupyter Notebooks with Python 3.7 for GPU service and the Jupyter Notebooks with Python 3.7 for GPU service. You only need to mirror the images once.

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wsl-runtimes-1.0.0.tgz \
  --inventory runtimesOperatorSetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Jupyter Notebooks with R 3.6

  The same package is used for the Jupyter Notebooks with Python 3.7 for GPU service and the Jupyter Notebooks with Python 3.7 for GPU service. You only need to mirror the images once.

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-wsl-runtimes-1.0.0.tgz \
    --inventory runtimesOperatorSetup \
    --action mirror-images \
    --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
  inputDir ${OFFLINEDIR}"
  ```

- MongoDB

  Mirror the images for the package that you downloaded

  MongoDB

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-cpd-mongodb-4.0.0.tgz \
    --inventory ibmCPDMongodbSetup \
    --action mirror-images \
    --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
  inputDir ${OFFLINEDIR}"
  ```

  MongoDB

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-cpd-mongodb-4.0.0.tgz \
    --inventory ibmMongodbEnterpriseSetup \
    --action mirror-images \
    --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
  inputDir ${OFFLINEDIR}"
  ```

- OpenPages

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-openpages-2.0.0.tgz \
    --inventory operatorSetup \
    --action mirror-images \
    --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
  inputDir ${OFFLINEDIR}"
  ```

  If you want OpenPages to automatically provision a Db2 database, you must mirror the following images:

  Db2 as a service

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-db2aaservice-4.0.0.tgz \
    --inventory db2aaserviceOperatorSetup \
    --action mirror-images \
    --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
  inputDir ${OFFLINEDIR}"
  ```

- Planning Analytics

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-planning-analytics-4.0.0.tgz \
    --inventory ibmPlanningAnalyticsOperatorSetup \
    --action mirror-images \
    --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
  inputDir ${OFFLINEDIR}"
  ```

- Product Master

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-productmaster-1.0.0.tgz \
    --inventory productmasterOperatorSetup \
    --action mirror-images \
    --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
  inputDir ${OFFLINEDIR}"
  ```

- RStudio Server with R 3.6

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-rstudio-1.0.0.tgz \
    --inventory rstudioSetup \
    --action mirror-images \
    --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
  inputDir ${OFFLINEDIR}"
  ```

- SPSS Modeler

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-spss-1.0.0.tgz \
  --inventory spssSetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Virtual Data Pipeline

Not applicable

- Watson Knowledge Catalog

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wkc-4.0.0.tgz \
  --inventory wkcOperatorSetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Watson Machine Learning

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wml-cpd-4.0.1.tgz \
  --inventory wmlOperatorSetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Watson Machine Learning Accelerator

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wml-accelerator-2.3.0.tgz \
  --inventory wmla_operator_deploy \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Watson OpenScale

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-watson-openscale-2.0.1.tgz \
  --inventory ibmWatsonOpenscaleOperatorSetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Watson Studio

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wsl-2.0.0.tgz \
  --inventory wslSetup \
  --action mirror-images \
  --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

Mirroring images with an intermediary container registry

If your Red Hat® OpenShift® Container Platform cluster is air-gapped, you must mirror the software images that you need to a private container registry that is accessible from the cluster. You can use an intermediary container registry to mirror the images from the IBM® Entitled Registry to a private container registry.

Important: Use a Linux x86-64 system with Red Hat Enterprise Linux® to mirror the images.
The system must be able to access the following sites:

- Red Hat Quay.io (https://quay.io:443)
- GitHub (https://github.com)
  If your company does not permit access to GitHub, contact IBM Support for assistance.
- IBM Entitled Registry (http://icr.io:443)
  To validate that you can connect, run the following command:

  ```
  curl -v https://icr.io
  ```

  The command should return the following message:

  ```
  * Connected to icr.io (169.60.98.86) port 443 (#0)
  ```

Procedure

Complete the following tasks to mirror the images to your container registry:

1. Installing the software needed to mirror images

To use an intermediary registry, you must install the following software on the system:

Prerequisite	Purpose
One of the following container client tools: • Docker • Podman	Required to set up an in intermediary container registry for the images that you download.
OpenSSL Version 1.1.1 or later	Required to generate TLS certificates and keys for use with the intermediary registry.
OpenShift CLI	Required to interact with your Red Hat OpenShift Container Platform cluster.
IBM Cloud Pak® CLI (cloudctl)	Required to download images from the IBM Entitled Registry.
`httpd-tools`	Required to run the IBM Cloud Pak CLI (cloudctl).
`skopeo` Version 1.2.0 or later	Required to run the IBM Cloud Pak CLI (cloudctl).

To install the prerequisite software:

1. Install the container client tool of your choice:
   - To install Docker, run the following commands:

     ```
     yum check-update
     yum install docker
     ```

   - To install Podman, see the Podman installation instructions on the Podman site.
2. To install OpenSSL, see Downloads on the OpenSSL site.
3. To install the OpenShift CLI, see Getting started with the OpenShift CLI in the Red Hat documentation.
4. To install the IBM Cloud Pak CLI (cloudctl):
   a. Download the **`cloudctl`** software from the IBM/cloud-pak-cli repository on GitHub. Ensure that you download the appropriate package for your workstation:

      ```
      cloudctl-operating-system-architecture.tar.gz
      ```

   b. Extract the contents of the archive file:

      ```
      tar -xzf archive-name
      ```

   c. Change to the directory where you extracted the file and make the file executable:

      ```
      chmod 775 cloudctl-architecture
      ```

   d. Move the file to the /usr/local/bin directory:

      ```
      mv cloudctl-architecture /usr/local/bin/cloudctl
      ```

   e. Confirm that the IBM Cloud Pak CLI (cloudctl) is installed:

      ```
      cloudctl --help
      ```

   Tip: Additional guidance for validating the archive file is available in the IBM/cloud-pak-cli repository.
5. To install **`httpd-tools`**, run the following command:

   ```
   yum install httpd-tools
   ```

6. To install **`skopeo`**, see Installing from packages in the **`skopeo`** repository on GitHub.

2. Setting up your environment to download CASE packages

A Container Application Software for Enterprises (CASE) package is an archive file that describes a containerized component of Cloud Pak for Data.

There are CASE packages for:

- IBM Cloud Pak foundational services
- IBM Cloud Pak for Data control plane
- Each IBM Cloud Pak for Data service
- Software dependencies for the control plane and services

Each CASE package includes:

- Metadata about the component
- An inventory of the container images that are required to deploy the component
- References to any software dependencies

- The scripts needed to mirror the images to a private registry

Before you can mirror the images a private registry, you must download the CASE packages for the software that you plan to install.

To set up your environment:

1. Identify or create the directory where you want to store the CASE packages on the system.
   Important: Keep the following requirements in mind:
   - You must have sufficient storage in the directory. You must have sufficient storage for both the CASE packages and the software images that you need to mirror.
   - You must use a persistent directory. Using a persistent directory prevents you from transferring files more than once.
     Additionally, if you use a persistent directory, you can run the mirror process multiple times or on a schedule.

   For example, you could create a directory called offline:

   ```
   mkdir -p $HOME/offline
   ```

2. Set the following environment variables:

   ```
   export OFFLINEDIR=$HOME/offline
   export CASE_REPO_PATH=https://github.com/IBM/cloud-pak/raw/master/repo/case
   ```

   OFFLINEDIR is the directory that you created to store CASE packages. Replace **$HOME/offline** with the appropriate value for your environment.

3. Downloading the Cloud Pak for Data CASE package

1. Run the following command to download the IBM Cloud Pak for Data platform operator package:

   ```
   cloudctl case save \
     --case ${CASE_REPO_PATH}/ibm-cp-datacore-2.0.1.tgz \
     --outputdir ${OFFLINEDIR} \
     --no-dependency
   ```

4. Configuring credentials for mirroring images

The IBM Cloud Pak CLI (cloudctl) includes an action called **configure-cred-airgap**. Run the appropriate commands to store the credentials that you will need to mirror images to the private container registry. The command stores the credentials to the following file on your local file system: $HOME/.airgap/secrets.

To configure the credentials that you need to mirror software images:

1. Store the IBM Entitled Registry credentials by running the following command:

   ```
   cloudctl case launch \
     --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.1.tgz \
     --inventory cpdPlatformOperator \
     --action configure-creds-airgap \
     --args "--registry cp.icr.io --user cp --pass entitlement-key --inputDir ${OFFLINEDIR}"
   ```

   Replace *entitlement-key* with your entitlement key. For details, see IBM entitlement API key.

2. Store the intermediary container registry credentials:
   a. Determine the values that you want to use for the following parameters:

      PORTABLE_REGISTRY_USER
      The username that you want to use to authenticate to the intermediary container registry.
      PORTABLE_REGISTRY_PASS
      The password that you want to use to authenticate to the intermediary container registry.
      PORTABLE_REGISTRY
      The location of the intermediary registry on the local host. For example: **localhost:5000**.
      Pick a port that is not currently in use. To avoid conflicts with system ports, choose a port greater than **1024**.

   b. Set the environment variable for the parameters:

      ```
      export PORTABLE_REGISTRY_USER=username
      export PORTABLE_REGISTRY_PASSWORD=password
      export PORTABLE_REGISTRY=localhost:port_number
      ```

   c. Run the following command to store the credentials:

      ```
      cloudctl case launch \
        --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.1.tgz \
        --inventory cpdPlatformOperator \
        --action configure-creds-airgap \
        --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD}"
      ```

5. Setting up an intermediary container registry

You must create an intermediary container registry where you can mirror the images before you can mirror them to the private registry.

To create an intermediary container registry:

1. To initialize the intermediary container registry, run the following command:

   ```
   cloudctl case launch \
     --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.1.tgz \
     --inventory cpdPlatformOperator \
     --action init-registry \
   ```

```
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --dir
${OFFLINEDIR}/imageregistry"
```

2. Set the following environment variable to ensure that the IBM Cloud Pak CLI (cloudctl) uses `skopeo` rather than `oc mirror`. The `oc mirror` command can have trouble pulling images from binary repositories.

```
export USE_SKOPEO=true
```

3. To start the registry, run the following command:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.1.tgz \
  --inventory cpdPlatformOperator \
  --action start-registry \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --dir
${OFFLINEDIR}/imageregistry"
```

4. To verify that the registry is running, run the following commands:

Container client	Commands
Docker	List the containers by running: `docker ps`
Podman	List the containers by running: `podman ps`

6. Downloading shared cluster component CASE packages

Determine which shared cluster components you need to install on your cluster and download the appropriate CASE packages.

Shared cluster component	CASE download command

IBM Cloud Pak foundational services
Download this package if IBM Cloud Pak foundational services is not installed on the cluster.
	`cloudctl case save \`
	`--case ${CASE_REPO_PATH}/ibm-cp-common-services-1.4.1.tgz \`
	`--outputdir ${OFFLINEDIR}`

Scheduling service
Download this package if you plan to install Watson™ Machine Learning Accelerator or if you want to use the quota enforcement feature.
	`cloudctl case save \`
	`--case ${CASE_REPO_PATH}/ibm-cpd-scheduling-1.2.1.tgz \`
	`--outputdir ${OFFLINEDIR}`

7. Downloading service CASE packages

Decide which services you plan to install on your cluster and download the appropriate CASE packages.

- Analytics Engine Powered by Apache Spark

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-analyticsengine-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Cognos Analytics

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-cognos-analytics-prod-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Cognos Dashboards

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-cde-2.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Data Refinery

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-datarefinery-1.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Data Virtualization

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-dv-case-1.7.0.tgz \
--outputdir ${OFFLINEDIR}
```

- DataStage

  Download the appropriate package based on your license:

  DataStage Enterprise

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-datastage-enterprise-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

DataStage Enterprise Plus

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-datastage-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Db2

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-db2oltp-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Db2 Big SQL

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-bigsql-case-7.2.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Db2 Data Gate

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-datagate-prod-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Db2 Data Management Console

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-dmc-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Db2 Event Store

Not applicable. Contact IBM Software support if you plan to install this service.

- Db2 Warehouse

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-db2wh-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Decision Optimization

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-dods-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- EDB Postgres

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-cpd-edb-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Execution Engine for Apache Hadoop

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-hadoop-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Financial Services Workbench

Not applicable.

- IBM Match 360 with Watson

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-mdm-1.0.1.tgz \
--outputdir ${OFFLINEDIR}
```

- Jupyter Notebooks with Python 3.7 for GPU

The same package is used for the Jupyter Notebooks with Python 3.7 for GPU service and the Jupyter Notebooks with R 3.6 service. You only need to download the package once.

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-wsl-runtimes-1.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- Jupyter Notebooks with R 3.6

The same package is used for the Jupyter Notebooks with Python 3.7 for GPU service and the Jupyter Notebooks with R 3.6 service. You only need to download the package once.

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-wsl-runtimes-1.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- > MongoDB

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-cpd-mongodb-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- > OpenPages

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-openpages-2.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

If you want OpenPages to automatically provision a Db2 database, you must also download the following package: :

Db2 as a service

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-db2aaservice-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- > Planning Analytics

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-planning-analytics-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- > Product Master

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-productmaster-1.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- > RStudio Server with R 3.6

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-rstudio-1.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- > SPSS Modeler

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-spss-1.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- > Virtual Data Pipeline

Not applicable

- > Watson Knowledge Catalog

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-wkc-4.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

- > Watson Machine Learning

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-wml-cpd-4.0.1.tgz \
--outputdir ${OFFLINEDIR}
```

- > Watson Machine Learning Accelerator

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-wml-accelerator-2.3.0.tgz \
--outputdir ${OFFLINEDIR}
```

- > Watson OpenScale

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-watson-openscale-2.0.1.tgz \
--outputdir ${OFFLINEDIR}
```

- > Watson Studio

```
cloudctl case save \
--case ${CASE_REPO_PATH}/ibm-wsl-2.0.0.tgz \
--outputdir ${OFFLINEDIR}
```

8. Mirroring the images to the intermediary registry

To mirror the images:

1. Mirror the Cloud Pak for Data control plane images to the intermediary container registry:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.1.tgz \
  --inventory cpdPlatformOperator \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --inputDir ${OFFLINEDIR}"
```

2. Mirror the images for *each* of the shared cluster components that you downloaded:

Shared cluster component	Image mirroring command

IBM Cloud Pak foundational services	`cloudctl case launch \`
	`  --case ${OFFLINEDIR}/ibm-cp-common-services-1.4.1.tgz \`
	`  --inventory ibmCommonServiceOperatorSetup \`
	`  --action mirror-images \`
	`  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"`

Scheduling service	`cloudctl case launch \`
	`  --case ${OFFLINEDIR}/ibm-cpd-scheduling-1.2.1.tgz \`
	`  --inventory schedulerSetup \`
	`  --action mirror-images \`
	`  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"`

3. Mirror the images for *each* of the services that you downloaded:

- Analytics Engine Powered by Apache Spark

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-analyticsengine-4.0.0.tgz \
  --inventory analyticsengineOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Cognos Analytics

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cognos-analytics-prod-4.0.0.tgz \
  --inventory ibmCaOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Cognos Dashboards

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cde-2.0.0.tgz \
  --inventory cdeOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Data Refinery

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-datarefinery-1.0.0.tgz \
  --inventory datarefinerySetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Data Virtualization

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-dv-case-1.7.0.tgz \
  --inventory dv \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- DataStage

  Mirror the images for the package that you downloaded

  DataStage Enterprise

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-datastage-enterprise-4.0.0.tgz \
  --inventory datastageOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD}
--inputDir ${OFFLINEDIR}"
```

DataStage Enterprise Plus

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-datastage-4.0.0.tgz \
  --inventory datastageOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD}
--inputDir ${OFFLINEDIR}"
```

- Db2

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-db2oltp-4.0.0.tgz \
  --inventory db2oltpOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Db2 Big SQL

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-bigsql-case-7.2.0.tgz \
  --inventory bigsql \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Db2 Data Gate

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-datagate-prod-4.0.0.tgz \
  --inventory datagateOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Db2 Data Management Console

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-dmc-4.0.0.tgz \
  --inventory dmcOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Db2 Event Store

Not applicable. Contact IBM Software support if you plan to install this service.

- Db2 Warehouse

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-db2wh-4.0.0.tgz \
  --inventory db2whOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Decision Optimization

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-dods-4.0.0.tgz \
  --inventory dodsOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- EDB Postgres

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cpd-edb-4.0.0.tgz \
  --inventory ibmCPDEDBSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Execution Engine for Apache Hadoop

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-hadoop-4.0.0.tgz \
  --inventory hadoopSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Financial Services Workbench

  Not applicable.

- IBM Match 360 with Watson

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-mdm-1.0.1.tgz \
  --inventory mdmOperator \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Jupyter Notebooks with Python 3.7 for GPU

  The same package is used for the Jupyter Notebooks with Python 3.7 for GPU service and the Jupyter Notebooks with Python 3.7 for GPU service. You only need to mirror the images once.

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wsl-runtimes-1.0.0.tgz \
  --inventory runtimesOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Jupyter Notebooks with R 3.6

  The same package is used for the Jupyter Notebooks with Python 3.7 for GPU service and the Jupyter Notebooks with Python 3.7 for GPU service. You only need to mirror the images once.

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wsl-runtimes-1.0.0.tgz \
  --inventory runtimesOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- MongoDB

  Mirror the images for the package that you downloaded

  MongoDB

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cpd-mongodb-4.0.0.tgz \
  --inventory ibmCPDMongodbSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD}
--inputDir ${OFFLINEDIR}"
```

  MongoDB

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cpd-mongodb-4.0.0.tgz \
  --inventory ibmMongodbEnterpriseSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD}
--inputDir ${OFFLINEDIR}"
```

- OpenPages

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-openpages-2.0.0.tgz \
  --inventory operatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

  If you want OpenPages to automatically provision a Db2 database, you must mirror the following images:

  Db2 as a service

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-db2aaservice-4.0.0.tgz \
  --inventory db2aaserviceOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD}
--inputDir ${OFFLINEDIR}"
```

- Planning Analytics

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-planning-analytics-4.0.0.tgz \
  --inventory ibmPlanningAnalyticsOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Product Master

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-productmaster-1.0.0.tgz \
  --inventory productmasterOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- RStudio Server with R 3.6

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-rstudio-1.0.0.tgz \
  --inventory rstudioSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- SPSS Modeler

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-spss-1.0.0.tgz \
  --inventory spssSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Virtual Data Pipeline

Not applicable

- Watson Knowledge Catalog

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wkc-4.0.0.tgz \
  --inventory wkcOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Watson Machine Learning

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wml-cpd-4.0.1.tgz \
  --inventory wmlOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Watson Machine Learning Accelerator

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wml-accelerator-2.3.0.tgz \
  --inventory wmla_operator_deploy \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Watson OpenScale

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-watson-openscale-2.0.1.tgz \
  --inventory ibmWatsonOpenscaleOperatorSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

- Watson Studio

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wsl-2.0.0.tgz \
  --inventory wslSetup \
  --action mirror-images \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --
inputDir ${OFFLINEDIR}"
```

4. Determine the appropriate action based on the mirroring method that you are using:

Method	Next steps
Portable compute device	a. Move the portable compute device behind your firewall or connect the device to your private network. b. Complete 10. Mirroring images to the private registry

Method	Next steps

File transfer with a portable storage device	a. Save a copy of the software that you downloaded when you completed 1. Installing the software needed to mirror images to the ${OFFLINEDIR} on the portable storage device.
	b. Save the Docker registry image:
	• For Docker, run:
	`docker save docker.io/library/registry:2.6 -o ${OFFLINEDIR}/registry.tar`
	• For Podmand, run:
	`podman save docker.io/library/registry:2.6 -o ${OFFLINEDIR}/registry.tar`
	c. Disconnect the portable storage device and move the device behind your firewall.
	d. Identify a workstation from which you can access the private registry, where you can start the intermediary registry, and where you can attach the portable storage device.
	e. Set an environment variable for the directory where the portable storage device is attached. For example:
	`export OFFLINEDIR=$HOME/offline`
	f. Complete 9. Setting up a workstation to serve images.

File transfer with `scp` or `sftp`	a. Save a copy of the software that you downloaded when you completed 1. Installing the software needed to mirror images to the ${OFFLINEDIR}.
	b. Save the Docker registry image:
	• For Docker, run:
	`docker save docker.io/library/registry:2.6 -o ${OFFLINEDIR}/registry.tar`
	• For Podmand, run:
	`podman save docker.io/library/registry:2.6 -o ${OFFLINEDIR}/registry.tar`
	c. Archive the contents off the `$OFFLINE` directory:
	`tar -cvzf archive-file-name -c ${OFFLINEDIR}`
	d. Identify a workstation from which you can access the private registry and where you can start the intermediary registry.
	e. Copy the archive file to the workstation using `scp` or `sftp`.
	f. Identify or create the directory where you want to serve the images. For example:
	`mkdir -p $HOME/offline`
	g. Set an environment variable for this directory:
	`export OFFLINEDIR=$HOME/offline`
	h. Extract the contents of the archive:
	`tar -xvf archive-file-name -c ${OFFLINEDIR}`
	i. Complete 9. Setting up a workstation to serve images

9. Setting up a workstation to serve images

Skip this step if you are using a portable compute device.

Note: If you have an existing registry on this workstation that you set up using the IBM Cloud Pak CLI (cloudctl), you can re-use your setup and go directly to 10. Mirroring images to the private registry.
If you are transferring the files inside your firewall, you must set up a workstation to serve the images so that you can transfer them to the private container registry.

To set up the workstation:

1. Install the software that you installed in 1. Installing the software needed to mirror images.
   Remember: For any software that you downloaded, you should have a copy of the software in the ${OFFLINEDIR}.
2. Use the IBM Cloud Pak for Data platform operator package to set up the intermediary registry on the cluster node:
   a. Set the following environment variables for the IBM Cloud Pak for Data platform operator:

   ```
   export CASE_ARCHIVE=ibm-cp-datacore-2.0.1.tgz
   export CASE_INVENTORY_SETUP=cpdPlatformOperator
   ```

   b. Set environment variables for the intermediary container registry credentials. For example, reuse the credentials that you used when you set up the intermediary registry:

   ```
   export PORTABLE_REGISTRY_USER=username
   export PORTABLE_REGISTRY_PASSWORD=password
   export PORTABLE_REGISTRY=localhost:port_number
   ```

   c. Initialize the intermediary container registry:

   ```
   cloudctl case launch \
     --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.1.tgz \
     --inventory cpdPlatformOperator \
     --action init-registry \
     --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --dir
   ${OFFLINEDIR}/imageregistry"
   ```

   d. Set the following environment variable to ensure that the IBM Cloud Pak CLI (cloudctl) uses `skopeo` rather than `oc mirror`. The `oc mirror` command can have trouble pulling images from binary repositories.

   ```
   export USE_SKOPEO=true
   ```

e. Start the intermediary container registry:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.1.tgz \
  --inventory cpdPlatformOperator \
  --action start-registry \
  --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD} --dir
${OFFLINEDIR}/imageregistry"
```

f. To verify that the registry is running, run the following commands:

Container client	Commands
Docker	List the containers by running: `docker ps`
Podman	List the containers by running: `podman ps`

10. Mirroring images to the private registry

To mirror the images:

1. Store the intermediary container registry credentials:
   a. Determine the values that you want to use for the following parameters:

   PORTABLE_REGISTRY_USER
   The username that you want to use to authenticate to the intermediary container registry.
   PORTABLE_REGISTRY_PASS
   The password that you want to use to authenticate to the intermediary container registry.
   PORTABLE_REGISTRY
   The location of the intermediary registry on the local host. For example: `localhost:5000`.
   Pick a port that is not currently in use. To avoid conflicts with system ports, choose a port greater than `1024`.

   b. Set the environment variable for the parameters:

   ```
   export PORTABLE_REGISTRY_USER=username
   export PORTABLE_REGISTRY_PASSWORD=password
   export PORTABLE_REGISTRY=localhost:port_number
   ```

   c. Run the following command to store the credentials:

   ```
   cloudctl case launch \
     --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.1.tgz \
     --inventory cpdPlatformOperator \
     --action configure-creds-airgap \
     --args "--registry ${PORTABLE_REGISTRY} --user ${PORTABLE_REGISTRY_USER} --pass ${PORTABLE_REGISTRY_PASSWORD}"
   ```

2. Store the private container registry credentials:
   a. Work with your container registry administrator to identify the values for the following parameters:

   PRIVATE_REGISTRY_USER
   The username of a user who has the required privileges to *push* images to the private registry.
   PRIVATE_REGISTRY_PASSWORD
   The password of the user who has the required privileges to *push* images to the private registry.
   PRIVATE_REGISTRY
   The location of the private registry.

   b. Set environment variables for the parameters:

   ```
   export PRIVATE_REGISTRY_USER=username
   export PRIVATE_REGISTRY_PASSWORD=password
   export PRIVATE_REGISTRY=private-registry-location
   ```

   c. Run the following command to store the credentials:

   ```
   cloudctl case launch \
     --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.1.tgz \
     --inventory cpdPlatformOperator \
     --action configure-creds-airgap \
     --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD}"
   ```

3. Mirror the Cloud Pak for Data control plane images to the private container registry:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.1.tgz \
  --inventory cpdPlatformOperator \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

4. Mirror the images for *each* of the shared cluster components that you downloaded:

Shared cluster component	Image mirroring command

IBM Cloud Pak foundational services	`cloudctl case launch \`
	`  --case ${OFFLINEDIR}/ibm-cp-common-services-1.4.1.tgz \`
	`  --inventory ibmCommonServiceOperatorSetup \`
	`  --action mirror-images \`
	`  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"`

Shared cluster component	Image mirroring command

Scheduling service	`cloudctl case launch \`
	`  --case ${OFFLINEDIR}/ibm-cpd-scheduling-1.2.1.tgz \`
	`  --inventory schedulerSetup \`
	`  --action mirror-images \`
	`  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"`

5. Mirror the images for *each* of the services that you downloaded.
   - Analytics Engine Powered by Apache Spark

     ```
     cloudctl case launch \
       --case ${OFFLINEDIR}/ibm-analyticsengine-4.0.0.tgz \
       --inventory analyticsengineOperatorSetup \
       --action mirror-images \
       --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
     ```

   - Cognos Analytics

     ```
     cloudctl case launch \
       --case ${OFFLINEDIR}/ibm-cognos-analytics-prod-4.0.0.tgz \
       --inventory ibmCaOperatorSetup \
       --action mirror-images \
       --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
     ```

   - Cognos Dashboards

     ```
     cloudctl case launch \
       --case ${OFFLINEDIR}/ibm-cde-2.0.0.tgz \
       --inventory cdeOperatorSetup \
       --action mirror-images \
       --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
     ```

   - Data Refinery

     ```
     cloudctl case launch \
       --case ${OFFLINEDIR}/ibm-datarefinery-1.0.0.tgz \
       --inventory datarefinerySetup \
       --action mirror-images \
       --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
     ```

   - Data Virtualization

     ```
     cloudctl case launch \
       --case ${OFFLINEDIR}/ibm-dv-case-1.7.0.tgz \
       --inventory dv \
       --action mirror-images \
       --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
     ```

   - DataStage

     Mirror the images for the package that you downloaded

     DataStage Enterprise

     ```
     cloudctl case launch \
       --case ${OFFLINEDIR}/ibm-datastage-enterprise-4.0.0.tgz \
       --inventory datastageOperatorSetup \
       --action mirror-images \
       --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
     ```

     DataStage Enterprise Plus

     ```
     cloudctl case launch \
       --case ${OFFLINEDIR}/ibm-datastage-4.0.0.tgz \
       --inventory datastageOperatorSetup \
       --action mirror-images \
       --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
     ```

   - Db2

     ```
     cloudctl case launch \
       --case ${OFFLINEDIR}/ibm-db2oltp-4.0.0.tgz \
       --inventory db2oltpOperatorSetup \
       --action mirror-images \
       --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
     ```

- Db2 Big SQL

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-bigsql-case-7.2.0.tgz \
  --inventory bigsql \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- Db2 Data Gate

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-datagate-prod-4.0.0.tgz \
  --inventory datagateOperatorSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- Db2 Data Management Console

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-dmc-4.0.0.tgz \
  --inventory dmcOperatorSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- Db2 Event Store

Not applicable. Contact IBM Software support if you plan to install this service.

- Db2 Warehouse

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-db2wh-4.0.0.tgz \
  --inventory db2whOperatorSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- Decision Optimization

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-dods-4.0.0.tgz \
  --inventory dodsOperatorSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- EDB Postgres

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cpd-edb-4.0.0.tgz \
  --inventory ibmCPDEDBSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- Execution Engine for Apache Hadoop

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-hadoop-4.0.0.tgz \
  --inventory hadoopSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- Financial Services Workbench

Not applicable.

- IBM Match 360 with Watson

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-mdm-1.0.1.tgz \
  --inventory mdmOperator \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- Jupyter Notebooks with Python 3.7 for GPU

The same package is used for the Jupyter Notebooks with Python 3.7 for GPU service and the Jupyter Notebooks with Python 3.7 for GPU service. You only need to mirror the images once.

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wsl-runtimes-1.0.0.tgz \
  --inventory runtimesOperatorSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- Jupyter Notebooks with R 3.6

  The same package is used for the Jupyter Notebooks with Python 3.7 for GPU service and the Jupyter Notebooks with Python 3.7 for GPU service. You only need to mirror the images once.

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-wsl-runtimes-1.0.0.tgz \
    --inventory runtimesOperatorSetup \
    --action mirror-images \
    --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
  ${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
  ```

- MongoDB

  Mirror the images for the package that you downloaded

  MongoDB

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-cpd-mongodb-4.0.0.tgz \
    --inventory ibmCPDMongodbSetup \
    --action mirror-images \
    --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --
  pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
  ```

  MongoDB

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-cpd-mongodb-4.0.0.tgz \
    --inventory ibmMongodbEnterpriseSetup \
    --action mirror-images \
    --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --
  pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
  ```

- OpenPages

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-openpages-2.0.0.tgz \
  --inventory operatorSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

If you want OpenPages to automatically provision a Db2 database, you must mirror the following images:

Db2 as a service

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-db2aaservice-4.0.0.tgz \
  --inventory db2aaserviceOperatorSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --
pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- Planning Analytics

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-planning-analytics-4.0.0.tgz \
  --inventory ibmPlanningAnalyticsOperatorSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- Product Master

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-productmaster-1.0.0.tgz \
  --inventory productmasterOperatorSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- RStudio Server with R 3.6

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-rstudio-1.0.0.tgz \
  --inventory rstudioSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- SPSS Modeler

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-spss-1.0.0.tgz \
  --inventory spssSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- Virtual Data Pipeline

Not applicable

- Watson Knowledge Catalog

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wkc-4.0.0.tgz \
  --inventory wkcOperatorSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- Watson Machine Learning

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wml-cpd-4.0.1.tgz \
  --inventory wmlOperatorSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- Watson Machine Learning Accelerator

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wml-accelerator-2.3.0.tgz \
  --inventory wmla_operator_deploy \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- Watson OpenScale

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-watson-openscale-2.0.1.tgz \
  --inventory ibmWatsonOpenscaleOperatorSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

- Watson Studio

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wsl-2.0.0.tgz \
  --inventory wslSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

Configuring your cluster to pull Cloud Pak for Data images

To ensure that your cluster can pull Cloud Pak for Data software images, you must update your cluster configuration.

Permissions you need for this task
: You must be a cluster administrator.

When you need to complete this task
: You must complete this task the first time you install Cloud Pak for Data.

The tasks that you must complete depend on whether your cluster pulls images directly from the IBM® Entitled Registry or from a private container registry.

Task	IBM Entitled Registry	Private container registry
1. Configuring the global image pull secret	Required	Required
2. Configuring an image content source policy	Not applicable	Required
3. Creating the catalog source	Required	Required

1. Configuring the global image pull secret

The global image pull secret ensures that your cluster has the necessary credentials to pull images.

The credentials that you need to specify depend on where you want to pull images from:

IBM Entitled Registry
: If you are pulling images from the IBM Entitled Registry, the global image pull secret must contain your IBM entitlement API key.

Private container registry
: If you are pulling images from a private container registry, the global image pull secret must contain the credentials of an account that can pull images from the registry.

If you have already configured the global image pull secret with the necessary credentials, you can skip this task.

Important: Changing the global image pull secret will *automatically* restart each node in the cluster so that the Machine Config Operator can apply the changes, so you might notice that resources are temporarily unavailable. However, this process happens one node at a time. The cluster will wait for the node to restart before starting the process on the next node.
If your deployment is on IBM Cloud, you must *manually* reload the worker nodes in your cluster for the changes to take effect.

To configure the global image pull secret:

1. Determine whether there is an existing global image pull secret:

   ```
   oc extract secret/pull-secret -n openshift-config
   ```

   This command generates a JSON file called .dockerconfigjson in the current directory.

2. Take the appropriate action based on the contents of the .dockerconfigjson file:

Pull secret status	Image content source policy

The file is empty	a. Set the following environment variables based on the container registry that OpenShift® is going to pull from:
	
	IBM Entitled Registry
	
	`export REGISTRY_USER=docker-username=cp`
	`export REGISTRY_PASSWORD=`***`entitlement-key`***
	`export REGISTRY_SERVER=cp.icr.io`
	
	Replace *entitlement-key* with your entitlement key. For details, see IBM entitlement API key.
	
	Private container registry
	
	`export REGISTRY_USER=`***`username`***
	`export REGISTRY_PASSWORD=`***`password`***
	`export REGISTRY_SERVER=`***`registry-location`***
	
	Replace the following values:
	
	username
	The username of a user that can pull images from the private registry
	password
	The password for the specified user.
	registry-location
	The location of the private registry. For example, `private-registry.example.com`.
	
	b. Run the following command to create the pull secret:
	
	`oc create secret docker-registry \`
	`--docker-server=${REGISTRY_SERVER} \`
	`--docker-username=${REGISTRY_USER} \`
	`--docker-password=${REGISTRY_PASSWORD} \`
	`--docker-email=${REGISTRY_USER} \`
	`-n openshift-config pull-secret`

Pull secret status	Image content source policy

There is an existing pull secret	a. Encode the username and password using Base64 encoding:
	
	IBM Entitled Registry
	
	`echo -n "cp:entitlement-key" | base64 -w0`
	
	Replace *entitlement-key* with your entitlement key. For details, see IBM entitlement API key.
	
	Private container registry
	
	`echo -n "username:password" | base64 -w0`
	
	Replace the following values:
	
	username
	The username of a user that can pull images from the private registry
	password
	The password for the specified user.
	
	b. Add an entry for the registry to the `auths` section in the JSON file. In the following example, `1` is the new entry and `2` is the existing entry:
	
	`{ "auths":{ 1 "registry-location":{ "auth":"base64-encoded-credentials", "email":"not-used" }, 2 "myregistry.example.com":{ "auth":"b3Blb=", "email":"not-used" } } }`
	
	Replace the following values:
	
	registry-location
	If you are pulling images from the IBM Entitled Registry, the value is `cp.icr.io`.
	If you are pulling images from a private container registry, specify the location of the private container registry. For example, `private-registry.example.com`.
	
	base64-encoded-credentials
	The encoded credentials that you generated in the previous step. For example, `cmVnX3VzZXJuYW1lOnJlZ19wYXNzd29yZAo=`.
	
	c. Apply the new configuration:
	
	`oc set data secret/pull-secret -n openshift-config --from-file=.dockerconfigjson=.dockerconfigjson`

Important: For deployments on IBM Cloud, you must reload the worker nodes in your cluster for the changes to take effect. For details, see Adding a private registry to the global pull secret.
If have a VPC Gen2 cluster and you use Portworx storage, see Portworx storage limitations before you reload your worker nodes.

3. Get the status of the nodes:

   `oc get node`

   Wait until all the nodes are `Ready` before you proceed to the next step. For example, if you see `Ready,SchedulingDisabled`, wait for the process to complete:

```
NAME                         STATUS                     ROLES    AGE     VERSION
master0                      Ready                      master   5h57m   v1.20.0
master1                      Ready                      master   5h57m   v1.20.0
master2                      Ready                      master   5h57m   v1.20.0
worker0                      Ready,SchedulingDisabled   worker   5h48m   v1.20.0
worker                       Ready                      worker   5h48m   v1.20.0
worker2                      Ready                      worker   5h48m   v1.20.0
```

2. Configuring an image content source policy

If you are pulling images directly from the IBM Entitled Registry on a connected cluster, you can skip this step.

If you mirrored images to a private container registry, you must tell your cluster where to find the software images. (For more information how Red Hat® OpenShift Container Platform locates images from an mirrored repository, see Configuring image registry repository mirroring in the Red Hat OpenShift Container Platform documentation.)

Important: This process will temporarily disable scheduling on each node in the cluster, so you might notice that resources are temporarily unavailable. However, this process happens on one node at a time. The cluster will temporarily disable scheduling on a node, apply the configuration change, and then re-enable scheduling before starting the process on the next node.
To configure an image content source policy:

1. Set the following environment variable to point to the location of the private registry:

   `export PRIVATE_REGISTRY=private-registry-location`

2. Create an image content source policy. The contents of the policy depend on whether you have an existing policy for IBM Cloud Pak® foundational services.

Options	Image content source policy

IBM Cloud Pak foundational services is already installed on the cluster	If IBM Cloud Pak foundational services is already installed, it is likely that you already have an image content source policy for `quay.io/opencloudio`. Therefore, you do not need to create a mirroring policy for those images.
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operator.openshift.io/v1alpha1`
	`kind: ImageContentSourcePolicy`
	`metadata:`
	`  name: cloud-pak-for-data-mirror`
	`spec:`
	`  repositoryDigestMirrors:`
	`  - mirrors:`
	`  - ${PRIVATE_REGISTRY}/cp`
	`  source: cp.icr.io/cp`
	`  - mirrors:`
	`  - ${PRIVATE_REGISTRY}/cp/cpd`
	`  source: cp.icr.io/cp/cpd`
	`  - mirrors:`
	`  - ${PRIVATE_REGISTRY}/cpopen`
	`  source: icr.io/cpopen`
	`EOF`

IBM Cloud Pak foundational services is not installed on the cluster	If IBM Cloud Pak foundational services is not installed, it is unlikely that you have an image content source policy for `quay.io/opencloudio`, so you should create a mirroring policy for those images.
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operator.openshift.io/v1alpha1`
	`kind: ImageContentSourcePolicy`
	`metadata:`
	`  name: cloud-pak-for-data-mirror`
	`spec:`
	`  repositoryDigestMirrors:`
	`  - mirrors:`
	`  - ${PRIVATE_REGISTRY}/opencloudio`
	`  source: quay.io/opencloudio`
	`  - mirrors:`
	`  - ${PRIVATE_REGISTRY}/cp`
	`  source: cp.icr.io/cp`
	`  - mirrors:`
	`  - ${PRIVATE_REGISTRY}/cp/cpd`
	`  source: cp.icr.io/cp/cpd`
	`  - mirrors:`
	`  - ${PRIVATE_REGISTRY}/cpopen`
	`  source: icr.io/cpopen`
	`EOF`

3. Verify that the image content source policy was created:

   ```
   oc get imageContentSourcePolicy
   ```

4. Get the status of the nodes:

   ```
   oc get node
   ```

   Wait until all the nodes are `Ready` before you proceed to the next step. For example, if you see `Ready,SchedulingDisabled`, wait for the process to complete:

   ```
   NAME                          STATUS                     ROLES    AGE     VERSION
   master0                       Ready                      master   5h57m   v1.20.0
   master1                       Ready                      master   5h57m   v1.20.0
   master2                       Ready                      master   5h57m   v1.20.0
   worker0                       Ready,SchedulingDisabled   worker   5h48m   v1.20.0
   worker                        Ready                      worker   5h48m   v1.20.0
   worker2                       Ready                      worker   5h48m   v1.20.0
   ```

3. Creating the catalog source

Operator Lifecycle Manager (OLM) uses an *Operator catalog* to discover and install Operators and their dependencies.

A *catalog source* is a repository of cluster service versions (CSVs), custom resource definitions (CRDs), and packages that comprise an application. To ensure that OLM can use the Cloud Pak for Data operators to install the software, you must create the appropriate catalog sources for your environment. (For more information about these terms, see the Operator Framework glossary of common terms in the Red Hat OpenShift Container Platform documentation.)

To create the catalog source, complete the appropriate steps for your environment:

Image location	Required catalog source

IBM Entitled Registry	If you are pulling images from the IBM Entitled Registry, create the following catalog sources:
	
	1. Create the IBM Operator catalog source.
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: CatalogSource`
	`metadata:`
	`  name: ibm-operator-catalog`
	`  namespace: openshift-marketplace`
	`spec:`
	`  displayName: "IBM Operator Catalog"`
	`  publisher: IBM`
	`  sourceType: grpc`
	`  image: icr.io/cpopen/ibm-operator-catalog:latest`
	`  updateStrategy:`
	`  registryPoll:`

Image location	Required catalog source

```
      interval: 45m
EOF
```

This catalog source is used for :

- IBM Cloud Pak foundational services
- IBM Cloud Pak for Data platform operator
- Service operators

2. Create the following catalog source to ensure that dependencies can be installed:

```
cat <<EOF |oc apply -f -
---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-cpd-ccs-operator-catalog
  namespace: openshift-marketplace
spec:
  sourceType: grpc
  image: icr.io/cpopen/ibm-cpd-ccs-operator-
catalog@sha256:34854b0b5684d670cf1624d01e659e9900f4206987242b453ee917b32b79f5b7
  imagePullPolicy: Always
  displayName: CPD Common Core Services
  publisher: IBM

---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-cpd-datarefinery-operator-catalog
  namespace: openshift-marketplace
spec:
  sourceType: grpc
  image: icr.io/cpopen/ibm-cpd-datarefinery-operator-
catalog@sha256:27c6b458244a7c8d12da72a18811d797a1bef19dadf84b38cedf6461fe53643a
  imagePullPolicy: Always
  displayName: Cloud Pak for Data IBM DataRefinery
  publisher: IBM

---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-db2aaservice-cp4d-operator-catalog
  namespace: openshift-marketplace
spec:
  sourceType: grpc
  image: icr.io/cpopen/ibm-db2aaservice-cp4d-operator-
catalog@sha256:a0d9b6c314193795ec1918e4227ede916743381285b719b3d8cfb05c35fec071
  imagePullPolicy: Always
  displayName: IBM Db2aaservice CP4D Catalog
  publisher: IBM

---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-cpd-iis-operator-catalog
  namespace: openshift-marketplace
spec:
  sourceType: grpc
  image: icr.io/cpopen/ibm-cpd-iis-operator-
catalog@sha256:3ad952987b2f4d921459b0d3bad8e30a7ddae9e0c5beb407b98cf3c09713efcc
  imagePullPolicy: Always
  displayName: CPD IBM Information Server
  publisher: IBM

---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-cpd-wml-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: Cloud Pak for Data Watson Machine Learning
  publisher: IBM
  sourceType: grpc
  imagePullPolicy: Always
  image: icr.io/cpopen/ibm-cpd-wml-operator-
catalog@sha256:d2da8a2573c0241b5c53af4d875dbfbf988484768caec2e4e6231417828cb192
  updateStrategy:
    registryPoll:
      interval: 45m

---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-cpd-ws-operator-catalog
  namespace: openshift-marketplace
spec:
  sourceType: grpc
  image: icr.io/cpopen/ibm-cpd-ws-operator-
catalog@sha256:bf6b42df3d8cee32740d3273154986b28dedbf03349116fba39974dc29610521
```

Image location	Required catalog source

```
  imagePullPolicy: Always
  displayName: CPD IBM Watson Studio
  publisher: IBM

---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: opencontent-elasticsearch-dev-catalog
  namespace: openshift-marketplace
spec:
  sourceType: grpc
  image: icr.io/cpopen/opencontent-elasticsearch-operator-
catalog@sha256:bc284b8c2754af2eba81bb1edf6daa59dc823bf7a81fe91710c603f563a9a724
  displayName: IBM Opencontent Elasticsearch Catalog
  publisher: CloudpakOpen
  updateStrategy:
    registryPoll:
      interval: 45m

---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-rabbitmq-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM RabbitMQ operator Catalog
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/opencontent-rabbitmq-operator-
catalog@sha256:c3b14816eabc04bcdd5c653eaf6e0824adb020ca45d81d57059f50c80f22964f
  updateStrategy:
    registryPoll:
      interval: 45m

---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-cloud-databases-redis-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-cloud-databases-redis-operator-catalog
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-cloud-databases-redis-
catalog@sha256:980e4182ec20a01a93f3c18310e0aa5346dc299c551bd8aca070ddf2a5bf9ca5

---
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-cpd-ws-runtimes-operator-catalog
  namespace: openshift-marketplace
spec:
  sourceType: grpc
  image: icr.io/cpopen/ibm-cpd-ws-runtimes-operator-
catalog@sha256:c1faf293456261f418e01795eecd4fe8b48cc1e8b37631fb6433fad261b74ea4
  imagePullPolicy: Always
  displayName: CPD Watson Studio Runtimes
  publisher: IBM
EOF
```

3. Create the Db2U catalog source if you plan to install one of the following services:
   - Data Virtualization
   - Db2®
   - Db2 Big SQL
   - Db2 Warehouse
   - OpenPages® (required only if you want OpenPages to automatically provision a Db2 database)

```
cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-db2uoperator-catalog
  namespace: openshift-marketplace
spec:
  sourceType: grpc
  image: docker.io/ibmcom/ibm-db2uoperator-catalog:latest
  imagePullPolicy: Always
  displayName: IBM Db2U Catalog
  publisher: IBM
  updateStrategy:
    registryPoll:
      interval: 45m
EOF
```

Image location	Required catalog source

Private container registry	Recommendation: It is strongly recommended that you use the IBM Cloud Pak CLI (cloudctl) to complete this task. The IBM Cloud Pak CLI includes a script that can automatically generate the catalog source. If you'd prefer not to permit the IBM Cloud Pak CLI to create the catalog source, you can include the `--dry-run` option before the arguments (`--args`).
	The `--dry-run` option shows you the equivalent `oc` commands to run to achieve the same effect.
	
	The following steps do not include the `--dry-run` option.
	
	The following steps assume that you have the CASE packages on you local file system from mirroring the images to your private container registry.
	
	If you are running the commands on a different machine, you must download the necessary packages before you create the catalog source:
	
	• IBM Cloud Pak foundational services CASE package (Skip this download if you already have a catalog source for IBM Cloud Pak foundational services)
	• IBM Cloud Pak foundational services CASE package (Skip this download if you are not installing the scheduling service.)
	• IBM Cloud Pak for Data CASE package
	• Service CASE packages
	
	If you are pulling images from a private container registry, create the following catalog sources:
	
	1. Create the IBM Cloud Pak foundational services catalog source. (Skip this step if you already have a catalog source for IBM Cloud Pak foundational services.)
	`cloudctl case launch \`
	`  --case ${OFFLINEDIR}/ibm-cp-common-services-1.4.1.tgz \`
	`  --inventory ibmCommonServiceOperatorSetup \`
	`  --namespace openshift-marketplace \`
	`  --action install-catalog \`
	`  --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"`
	2. Create the Scheduling service catalog source. (Skip this step if you are not installing the scheduling service.)
	`cloudctl case launch \`
	`  --case ${OFFLINEDIR}/ibm-cpd-scheduling-1.2.1.tgz \`
	`  --inventory schedulerSetup \`
	`  --namespace openshift-marketplace \`
	`  --action install-catalog \`
	`  --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"`
	3. Create the IBM Cloud Pak for Data catalog source:
	`cloudctl case launch \`
	`  --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.1.tgz \`
	`  --inventory cpdPlatformOperator \`
	`  --namespace openshift-marketplace \`
	`  --action install-catalog \`
	`  --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"`
	4. Create the catalog source for *each* service that you mirrored to the private container registry. For details, see Service catalog source.

Service catalog source

If you are using a private container registry, create the catalog source for each service that you plan to install.

- Analytics Engine Powered by Apache Spark

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-analyticsengine-4.0.0.tgz \
  --inventory analyticsengineOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Cognos Analytics

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cognos-analytics-prod-4.0.0.tgz \
  --inventory ibmCaOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Cognos Dashboards

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cde-2.0.0.tgz \
  --inventory cdeOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Data Refinery

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-datarefinery-1.0.0.tgz \
  --inventory datarefinerySetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Data Virtualization

  Create the catalog source for *both* of the following operators:

  Db2U

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-db2uoperator-4.0.0.tgz \
    --inventory db2uOperatorSetup \
    --namespace openshift-marketplace \
    --action install-catalog \
      --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
  ```

  Data Virtualization

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-dv-case-1.7.0.tgz \
    --inventory dv \
    --namespace openshift-marketplace \
    --action install-catalog \
      --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
  ```

  After you create the catalog source for Data Virtualization, you must edit the catalog source to correct the private registry URL. You must complete the following steps:

  1. Run the following command:

     ```
     oc edit catalogsource -n openshift-marketplace ibm-dv-operator-catalog
     ```

  2. Edit the line `image: ${registry}/ibm-cpd-dv-operator-catalog@sha256:${sha}` , adding `cpopen` between the registry and the image name.
  3. Verify your change looks like the following example>

     ```
     image: ${registry}/cpopen/ibm-cpd-dv-operator-catalog@sha256:${sha}
     ```

- DataStage

  Create the appropriate catalog source for your environment:

  DataStage Enterprise

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-datastage-enterprise-4.0.0.tgz \
    --inventory datastageOperatorSetup \
    --namespace openshift-marketplace \
    --action install-catalog \
      --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
  ```

  DataStage Enterprise Plus

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-datastage-4.0.0.tgz \
    --inventory datastageOperatorSetup \
    --namespace openshift-marketplace \
    --action install-catalog \
      --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
  ```

- Db2

  Create the catalog source for *both* of the following operators:

  Db2U

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-db2uoperator-4.0.0.tgz \
    --inventory db2uOperatorSetup \
    --namespace openshift-marketplace \
    --action install-catalog \
      --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
  ```

  Db2

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-db2oltp-4.0.0.tgz \
    --inventory db2oltpOperatorSetup \
    --namespace openshift-marketplace \
    --action install-catalog \
      --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
  ```

- Db2 Big SQL

  Create the catalog source for *both* of the following operators:

  Db2U

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-db2uoperator-4.0.0.tgz \
    --inventory db2uOperatorSetup \
  ```

```
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

Db2 Big SQL

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-bigsql-case-7.2.0.tgz \
  --inventory bigsql \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Db2 Data Gate

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-datagate-prod-4.0.0.tgz \
  --inventory datagateOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Db2 Data Management Console

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-dmc-4.0.0.tgz \
  --inventory dmcOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Db2 Event Store

Not applicable. Contact IBM Software support if you plan to install this service.

- Db2 Warehouse

Create the catalog source for *both* of the following operators:

Db2U

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-db2uoperator-4.0.0.tgz \
  --inventory db2uOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

Db2 Warehouse

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-db2wh-4.0.0.tgz \
  --inventory db2whOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Decision Optimization

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-dods-4.0.0.tgz \
  --inventory dodsOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- EDB Postgres

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cpd-edb-4.0.0.tgz \
  --inventory ibmCPDEDBSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Execution Engine for Apache Hadoop

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-hadoop-4.0.0.tgz \
  --inventory hadoopSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Financial Services Workbench

Not applicable.

- IBM Match 360 with Watson

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-mdm-1.0.1.tgz \
  --inventory mdmOperator \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Jupyter Notebooks with Python 3.7 for GPU

The same operator is used for the Jupyter Notebooks with Python 3.7 for GPU service and the Jupyter Notebooks with R 3.6 service. You only need to create the catalog source once.

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wsl-runtimes-1.0.0.tgz \
  --inventory runtimesOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Jupyter Notebooks with R 3.6

The same operator is used for the Jupyter Notebooks with Python 3.7 for GPU service and the Jupyter Notebooks with R 3.6 service. You only need to create the catalog source once.

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wsl-runtimes-1.0.0.tgz \
  --inventory runtimesOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- MongoDB

Create the appropriate catalog source for your environment:

MongoDB

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cpd-mongodb-4.0.0.tgz \
  --inventory ibmCPDMongodbSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

MongoDB

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cpd-mongodb-4.0.0.tgz \
  --inventory ibmMongodbEnterpriseSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- OpenPages

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-openpages-2.0.0.tgz \
  --inventory operatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

If you want OpenPages to automatically provision a Db2 database, you must also create the following catalog sources:

Db2U

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-db2uoperator-4.0.0.tgz \
  --inventory db2uOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

Db2 as a service

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-db2aaservice-4.0.0.tgz \
  --inventory db2aaserviceOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Planning Analytics

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-planning-analytics-4.0.0.tgz \
  --inventory ibmPlanningAnalyticsOperatorSetup \
  --namespace openshift-marketplace \
```

```
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Product Master

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-productmaster-1.0.0.tgz \
  --inventory productmasterOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- RStudio Server with R 3.6

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-rstudio-1.0.0.tgz \
  --inventory rstudioSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- SPSS Modeler

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-spss-1.0.0.tgz \
  --inventory spssSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Virtual Data Pipeline

Not applicable

- Watson Knowledge Catalog

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wkc-4.0.0.tgz \
  --inventory wkcOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Watson Machine Learning

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wml-cpd-4.0.1.tgz \
  --inventory wmlOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Watson Machine Learning Accelerator

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wml-accelerator-2.3.0.tgz \
  --inventory wmla_operator_deploy \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Watson OpenScale

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-watson-openscale-2.0.1.tgz \
  --inventory ibmWatsonOpenscaleOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

- Watson Studio

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wsl-2.0.0.tgz \
  --inventory wslSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR} --recursive"
```

Previous topic: Mirroring images to your container registry
Next topic: Installing IBM Cloud Pak foundational services

Installing IBM Cloud Pak foundational services

IBM Cloud Pak® foundational services is a prerequisite for IBM® Cloud Pak for Data. IBM Cloud Pak foundational services is installed one time on the cluster and is used by any instances of Cloud Pak for Data or other IBM Cloud Paks that are installed on the cluster.

Permissions you need for this task
: You must be a cluster administrator.

When you need to complete this task
: Use the following guidance to determine if you need to complete this task:

- If IBM Cloud Pak foundational services is already installed, you can skip this task.
- If you are running an express installation of IBM Cloud Pak for Data, you can skip this task.
- If you are running a specialized installation of IBM Cloud Pak for Data and IBM Cloud Pak foundational services is not installed, you must complete this task.

Before you begin

Verify that you completed these tasks before you install IBM Cloud Pak foundational services:

1. For environments that use a private container registry, such as air-gapped environments, the IBM Cloud Pak foundational services images are mirrored to the private container registry. For details, see Mirroring images to your container registry.
2. The cluster is configured to pull the software images. For details, see Configuring your cluster to pull Cloud Pak for Data images

If you do not complete these steps, the IBM Cloud Pak foundational services installation will fail.

Procedure

To install IBM Cloud Pak foundational services:

1. Create the appropriate operator subscription for your environment:

Image location	Required catalog source

Publicly available registry (`quay.io/opencloudio`)	If you are pulling images from the publicly available registry, create the following operator subscription:
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-common-service-operator`
	`  namespace: ibm-common-services`
	`spec:`
	`  channel: v3`
	`  installPlanApproval: Automatic`
	`  name: ibm-common-service-operator`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Private container registry	If you are pulling images from a private container registry, create the following operator subscription:
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-common-service-operator`
	`  namespace: ibm-common-services`
	`spec:`
	`  channel: v3`
	`  installPlanApproval: Automatic`
	`  name: ibm-common-service-operator`
	`  source: opencloud-operators`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

When you create the operator subscription:

- The IBM Cloud Pak foundational services operator installs the `Operand Deployment Lifecycle Manager` operator and the `IBM NamespaceScope` operator in the `ibm-common-services` project.
- The IBM Cloud Pak foundational services operator creates the `CommonService` custom resource.
- The `Operand Deployment Lifecycle Manager` operator creates the `OperandRegistry`, `OperandConfig`, and the `OperatorBindInfo` instances in the `ibm-common-services` project.

Verifying the installation

Verify the status of the operators by running the following commands:

1. Verify the status of `ibm-common-service-operator`:

   ```
   oc --namespace ibm-common-services get csv
   ```

   If you installed IBM Cloud Pak foundational services in a different project, replace *`ibm-common-services`* with the correct project name.

   The command should return output with the following format:

   ```
   NAME                                    DISPLAY                                VERSION   REPLACES
   ibm-common-service-operator.v3.8        IBM Cloud Pak foundational services    3.8       ibm-common-service-opera
   ```

2. Verify that the custom resource definitions were created:

```
oc get crd | grep operandrequest
```

The command should return output with the following format:

```
NAME                                CREATED AT
operandrequests.operator.ibm.com    2021-06-23T10:10:22Z
```

3. Confirm that IBM Cloud Pak foundational services API resources are available:

```
oc api-resources --api-group operator.ibm.com
```

The command should return output similar to the following:

```
NAME                SHORTNAMES   APIGROUP            NAMESPACED   KIND
commonservices                   operator.ibm.com    true         CommonService
namespacescopes     nss          operator.ibm.com    true         NamespaceScope
operandbindinfos    opbi         operator.ibm.com    true         OperandBindInfo
operandconfigs      opcon        operator.ibm.com    true         OperandConfig
operandregistries   opreg        operator.ibm.com    true         OperandRegistry
operandrequests     opreq        operator.ibm.com    true         OperandRequest
podpresets                       operator.ibm.com    true         PodPreset
```

Installing the foundational services

The IBM Cloud Pak for Data platform operator automatically installs the following foundational services:

Certificate management service
: The IBM Cloud Pak for Data platform operator requires the Certificate management service (`ibm-cert-manager-operator`). If the Certificate management service is not installed, the IBM Cloud Pak for Data platform operator automatically installs the service.

Identity and Access Management Service (IAM Service)
: The IAM Service (`ibm-iam-operator`) is required if you decide to integrate with the IAM Service. If the IAM Service is not installed, the IBM Cloud Pak for Data platform operator automatically installs the service.

Administration hub
: The Administration hub (`ibm-commonui-operator`) is required if you decide to integrate with the IAM Service. If the Administration hub is not installed, the IBM Cloud Pak for Data platform operator automatically installs the service.

If you want to install additional foundational services, such as the License Service, you must manually install them. For details, see Installing foundational services in your cluster in the IBM Cloud Pak foundational services documentation.

Additional considerations

By default, the **`IBM NamespaceScope Operator`** that is installed with IBM Cloud Pak foundational services has *cluster permissions* so that role binding projections can be completed automatically.

You can optionally remove the cluster permissions from the **`IBM NamespaceScope Operator`** and manually authorize the projections. For details, see Authorizing foundational services to perform operations on workloads in a namespace.

Previous topic: Configuring your cluster to pull Cloud Pak for Data images
Next topic: Creating operator subscriptions

Creating operator subscriptions

An operator subscription tells the cluster where to install a given operator and gives information about the operator to Operator Lifecycle Manager (OLM).

When you create an operator subscription, OLM gets the cluster service version (CSV) for the operator. The CSV describes the operator, and OLM uses the CSV to:

- Introduce the custom resource definition (CRD) if it doesn't exist
- Set up the operator's service accounts
- Start up the operator deployment

Procedure

Complete the following tasks to create the relevant operator subscriptions on your cluster:

- 1. Choosing an install plan
- 2. Creating an operator subscription for the scheduling service
- 3. Creating an operator subscription for the IBM Cloud Pak for Data platform operator
- 4. Creating an operator subscription for services

1. Choosing an install plan

When you create an operator subscription, you specify the install plan for the operator; you can specify whether you want to manually or automatically upgrade the operator.

Automatic
: If you specify **`installPlanApproval: Automatic`**, Red Hat® OpenShift® Container Platform will automatically load newer versions of the operator if they are available. For example, if you mirror images to a private container registry and you set the install plan to automatic, Red Hat OpenShift Container Platform will automatically use the latest version of the operator that is available in the private container registry.

Manual

If you specify `installPlanApproval: Manual`, OLM creates an update request when a newer version of an operator is available. A cluster administrator must manually approve the update request to update the operator to the newer version.

Best practice: It is recommended that you use the automatic install plan.
Upgrading the operator does not impact the version of the software that is running on the cluster. It updates only the operator.

An operator *manages* the software.

For an in-depth description of operators, see the Red Hat OpenShift: Operators Framework video from Red Hat.

All of the operator subscriptions in this topic specify the automatic install plan (`installPlanApproval: Automatic`).

2. Creating an operator subscription for the scheduling service

If you don't plan to install the scheduling service, you can skip this step.

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

IBM® Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  annotations:`
	`  labels:`
	`  operators.coreos.com/ibm-cpd-scheduling-operator.aks: ""`
	`  velero.io/exclude-from-backup: "true"`
	`  name: ibm-cpd-scheduling-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: alpha`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-scheduling-operator`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  annotations:`
	`  labels:`
	`  operators.coreos.com/ibm-cpd-scheduling-operator.aks: ""`
	`  velero.io/exclude-from-backup: "true"`
	`  name: ibm-cpd-scheduling-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: alpha`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-scheduling-operator`
	`  source: ibm-cpd-scheduling-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

3. Creating an operator subscription for the IBM Cloud Pak for Data platform operator

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscriptions:
	
	1. Create the Cloud Pak for Data operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	``
	cat <<EOF |oc apply -f -
	apiVersion: operators.coreos.com/v1alpha1
	kind: Subscription
	metadata:
	  name: cpd-operator
	  namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator
	spec:
	  channel: stable-v1
	  installPlanApproval: Automatic
	  name: cpd-platform-operator
	  source: ibm-operator-catalog
	  sourceNamespace: openshift-marketplace
	EOF
	``
	
	2. If you are running a specialized installation (installing the IBM Cloud Pak® for Data platform operator and the IBM Cloud Pak foundational services in separate projects), create an operator subscription for the `IBM NamespaceScope Operator` in the IBM Cloud Pak for Data platform operator project:
	
	``
	cat <<EOF |oc apply -f -
	apiVersion: operators.coreos.com/v1alpha1
	kind: Subscription
	metadata:
	  name: ibm-namespace-scope-operator
	  namespace: cpd-operators
	spec:
	  channel: v3
	  installPlanApproval: Automatic
	  name: ibm-namespace-scope-operator
	  source: ibm-operator-catalog
	  sourceNamespace: openshift-marketplace
	EOF
	``

Private container registry	If you are pulling images from a private container registry, create the following operator subscriptions:
	
	1. Create the Cloud Pak for Data operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	``
	cat <<EOF |oc apply -f -
	apiVersion: operators.coreos.com/v1alpha1
	kind: Subscription
	metadata:
	  name: cpd-operator
	  namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator
	spec:
	  channel: stable-v1
	  installPlanApproval: Automatic
	  name: cpd-platform-operator
	  source: cpd-platform
	  sourceNamespace: openshift-marketplace
	EOF
	``
	
	2. If you are running a specialized installation (installing the IBM Cloud Pak for Data platform operator and the IBM Cloud Pak foundational services in separate projects), create an operator subscription for the `IBM NamespaceScope Operator` in the IBM Cloud Pak for Data platform operator project:
	
	``
	cat <<EOF |oc apply -f -
	apiVersion: operators.coreos.com/v1alpha1
	kind: Subscription
	metadata:
	  name: ibm-namespace-scope-operator
	  namespace: cpd-operators
	spec:
	  channel: v3
	  installPlanApproval: Automatic
	  name: ibm-namespace-scope-operator
	  source: opencloud-operators
	  sourceNamespace: openshift-marketplace
	EOF
	``

4. Creating an operator subscription for services

Create the operator subscription for each service that you plan to install.

- Analytics Engine Powered by Apache Spark

  Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

Image location	Required subscriptions
IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** labels:** ** app.kubernetes.io/instance: ibm-cpd-ae-operator-subscription** ** app.kubernetes.io/managed-by: ibm-cpd-ae-operator** ** app.kubernetes.io/name: ibm-cpd-ae-operator-subscription** ** name: ibm-cpd-ae-operator-subscription** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** **spec:** ** channel: stable-v1** ** installPlanApproval: Automatic** ** name: analyticsengine-operator** ** source: ibm-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**
Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** labels:** ** app.kubernetes.io/instance: ibm-cpd-ae-operator-subscription** ** app.kubernetes.io/managed-by: ibm-cpd-ae-operator** ** app.kubernetes.io/name: ibm-cpd-ae-operator-subscription** ** name: ibm-cpd-ae-operator-subscription** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** **spec:** ** channel: stable-v1** ** installPlanApproval: Automatic** ** name: analyticsengine-operator** ** source: ibm-cpd-ae-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**

- > Cognos Analytics

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions
IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** name: ibm-ca-operator-catalog-subscription** ** labels:** ** app.kubernetes.io/instance: ibm-ca-operator** ** app.kubernetes.io/managed-by: ibm-ca-operator** ** app.kubernetes.io/name: ibm-ca-operator** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** **spec:** ** channel: v4** ** name: ibm-ca-operator** ** installPlanApproval: Automatic** ** source: ibm-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**
Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** name: ibm-ca-operator-catalog-subscription** ** labels:** ** app.kubernetes.io/instance: ibm-ca-operator** ** app.kubernetes.io/managed-by: ibm-ca-operator** ** app.kubernetes.io/name: ibm-ca-operator** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** **spec:** ** channel: v4** ** name: ibm-ca-operator** ** installPlanApproval: Automatic** ** source: ibm-ca-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**

- > Cognos Dashboards

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-cde-operator-subscription`
	`  app.kubernetes.io/managed-by: ibm-cde-operator`
	`  app.kubernetes.io/name: ibm-cde-operator-subscription`
	`  name: ibm-cde-operator-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-cde-operator`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-cde-operator-subscription`
	`  app.kubernetes.io/managed-by: ibm-cde-operator`
	`  app.kubernetes.io/name: ibm-cde-operator-subscription`
	`  name: ibm-cde-operator-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-cde-operator`
	`  source: ibm-cde-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- > Data Refinery

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-cpd-datarefinery-operator-catalog-subscription`
	`  app.kubernetes.io/managed-by: ibm-cpd-datarefinery-operator`
	`  app.kubernetes.io/name: ibm-cpd-datarefinery-operator-catalog-subscription`
	`  name: ibm-cpd-datarefinery-operator-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-datarefinery`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-cpd-datarefinery-operator-catalog-subscription`
	`  app.kubernetes.io/managed-by: ibm-cpd-datarefinery-operator`
	`  app.kubernetes.io/name: ibm-cpd-datarefinery-operator-catalog-subscription`
	`  name: ibm-cpd-datarefinery-operator-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-datarefinery`
	`  source: ibm-cpd-datarefinery-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- Data Virtualization

  Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-dv-operator-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-dv-operator`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-dv-operator-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-dv-operator`
	`  source: ibm-dv-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- DataStage

  The operator subscription is the same for DataStage Enterprise or DataStage Enterprise Plus.

  Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-datastage-operator`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-datastage-operator`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-datastage-operator`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-datastage-operator`
	`  source: ibm-datastage-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- Db2

  Create the appropriate operator subscriptions for your environment:

Image location	Required subscriptions

<table>
<tr><th>Image location</th><th>Required subscriptions</th></tr>
<tr><td>IBM Entitled Registry</td><td>If you are pulling images from the publicly available registry, create the following operator subscriptions:
Create the Db2U operator subscription. (Ensure that you update the namespace parameter to specify the correct Red Hat OpenShift project.)

cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-db2uoperator-catalog-subscription
  namespace: ibm-common-services|cpd-operators   # Pick the project that contains the Cloud Pak for Data operator
spec:
  channel: v1.1
  name: db2u-operator
  installPlanApproval: Automatic
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF

Create the Db2 operator subscription. (Ensure that you update the namespace parameter to specify the correct Red Hat OpenShift project.)

cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-db2oltp-cp4d-operator-catalog-subscription
  namespace: ibm-common-services|cpd-operators   # Pick the project that contains the Cloud Pak for Data operator
spec:
  channel: v1.0
  name: ibm-db2oltp-cp4d-operator
  installPlanApproval: Automatic
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF</td></tr>
<tr><td>Private container registry</td><td>If you are pulling images from a private container registry, create the following operator subscriptions:
Create the Db2U operator subscription. (Ensure that you update the namespace parameter to specify the correct Red Hat OpenShift project.)

cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-db2uoperator-catalog-subscription
  namespace: ibm-common-services|cpd-operators   # Pick the project that contains the Cloud Pak for Data operator
spec:
  channel: v1.1
  name: db2u-operator
  installPlanApproval: Automatic
  source: ibm-db2uoperator-catalog
  sourceNamespace: openshift-marketplace
EOF

Create the Db2 operator subscription. (Ensure that you update the namespace parameter to specify the correct Red Hat OpenShift project.)

cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-db2oltp-cp4d-operator-catalog-subscription
  namespace: ibm-common-services|cpd-operators   # Pick the project that contains the Cloud Pak for Data operator
spec:
  channel: v1.0
  name: ibm-db2oltp-cp4d-operator
  installPlanApproval: Automatic
  source: ibm-db2oltp-cp4d-operator-catalog
  sourceNamespace: openshift-marketplace
EOF</td></tr>
</table>

- > Db2 Big SQL

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

Image location	Required subscriptions
IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** name: ibm-bigsql-operator-catalog-subscription** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** **spec:** ** channel: v1.0** ** installPlanApproval: Automatic** ** name: ibm-bigsql-operator** ** source: ibm-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**
Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** name: ibm-bigsql-operator-catalog-subscription** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** **spec:** ** channel: v1.0** ** installPlanApproval: Automatic** ** name: ibm-bigsql-operator** ** source: ibm-bigsql-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**

- > Db2 Data Gate

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions
IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** labels:** ** app.kubernetes.io/instance: ibm-datagate-operator-subscription** ** app.kubernetes.io/managed-by: ibm-datagate-operator** ** app.kubernetes.io/name: ibm-datagate-operator-subscription** ** name: ibm-datagate-operator-subscription** ** namespace: ibm-common-services** **spec:** ** channel: alpha** ** installPlanApproval: Automatic** ** name: ibm-datagate-operator** ** source: ibm-datagate-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**
Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** labels:** ** app.kubernetes.io/instance: ibm-datagate-operator-subscription** ** app.kubernetes.io/managed-by: ibm-datagate-operator** ** app.kubernetes.io/name: ibm-datagate-operator-subscription** ** name: ibm-datagate-operator-subscription** ** namespace: ibm-common-services** **spec:** ** channel: alpha** ** installPlanApproval: Automatic** ** name: ibm-datagate-operator** ** source: ibm-datagate-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**

- > Db2 Data Management Console

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

Image location	Required subscriptions
IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** name: ibm-dmc-operator-subscription** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** **spec:** ** channel: v1.0** ** installPlanApproval: Automatic** ** name: ibm-dmc-operator** ** source: ibm-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**
Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** name: ibm-dmc-operator-subscription** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** **spec:** ** channel: v1.0** ** installPlanApproval: Automatic** ** name: ibm-dmc-operator** ** source: ibm-dmc-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**

- > Db2 Event Store

Not applicable. Contact IBM Software support if you plan to install this service.

- > Db2 Warehouse

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions
IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscriptions: 1. Create the Db2U operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** name: ibm-db2uoperator-catalog-subscription** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** **spec:** ** channel: v1.1** ** name: db2u-operator** ** installPlanApproval: Automatic** ** source: ibm-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF** 2. Create the Db2 Warehouse operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** name: ibm-db2wh-cp4d-operator-catalog-subscription** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** **spec:** ** channel: v1.0** ** name: ibm-db2wh-cp4d-operator** ** installPlanApproval: Automatic** ** source: ibm-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**

Image location	Required subscriptions

Private container registry	If you are pulling images from a private container registry, create the following operator subscriptions:
	1. Create the Db2U operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.)
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-db2uoperator-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.1`
	`  name: db2u-operator`
	`  installPlanApproval: Automatic`
	`  source: ibm-db2uoperator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`
	2. Create the Db2 Warehouse operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.)
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-db2wh-cp4d-operator-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  name: ibm-db2wh-cp4d-operator`
	`  installPlanApproval: Automatic`
	`  source: ibm-db2wh-cp4d-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- > Decision Optimization

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.)
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-cpd-dods-operator-catalog-subscription`
	`  app.kubernetes.io/managed-by: ibm-cpd-dods-operator`
	`  app.kubernetes.io/name: ibm-cpd-dods-operator-catalog-subscription`
	`  name: ibm-cpd-dods-operator-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: alpha`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-dods`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.)
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-cpd-dods-operator-catalog-subscription`
	`  app.kubernetes.io/managed-by: ibm-cpd-dods-operator`
	`  app.kubernetes.io/name: ibm-cpd-dods-operator-catalog-subscription`
	`  name: ibm-cpd-dods-operator-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: alpha`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-dods`
	`  source: ibm-cpd-dods-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- > EDB Postgres

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

<table>
<tr><th>Image location</th><th>Required subscriptions</th></tr>
<tr><td>IBM Entitled Registry</td><td>If you are pulling images from the publicly available registry, create the following operator subscriptions:
Create the PostgreSQL operator subscription. (Ensure that you update the namespace parameter to specify the correct Red Hat OpenShift project.)

<pre>cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: cloud-native-postgresql-catalog-subscription
  namespace: ibm-common-services|cpd-operators   # Pick the project that contains the Cloud Pak for Data operator
spec:
  channel: stable
  name: cloud-native-postgresql
  installPlanApproval: Automatic
  source: cloud-native-postgresql-catalog
  sourceNamespace: openshift-marketplace
EOF</pre>
Create the EDB Postgres operator subscription. (Ensure that you update the namespace parameter to specify the correct Red Hat OpenShift project.)

<pre>cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-cpd-edb-operator-catalog-subscription
  namespace: ibm-common-services|cpd-operators   # Pick the project that contains the Cloud Pak for Data operator
spec:
  installPlanApproval: Automatic
  channel: stable
  name: ibm-cpd-edb
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF</pre></td></tr>
<tr><td>Private container registry</td><td>If you are pulling images from a private container registry, create the following operator subscriptions:
Create the PostgreSQL operator subscription. (Ensure that you update the namespace parameter to specify the correct Red Hat OpenShift project.)

<pre>cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: cloud-native-postgresql-catalog-subscription
  namespace: ibm-common-services|cpd-operators   # Pick the project that contains the Cloud Pak for Data operator
spec:
  channel: stable
  name: cloud-native-postgresql
  installPlanApproval: Automatic
  source: cloud-native-postgresql-catalog
  sourceNamespace: openshift-marketplace
EOF</pre>
Create the EDB Postgres operator subscription. (Ensure that you update the namespace parameter to specify the correct Red Hat OpenShift project.)

<pre>cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-cpd-edb-operator-catalog-subscription
  namespace: ibm-common-services|cpd-operators   # Pick the project that contains the Cloud Pak for Data operator
spec:
  installPlanApproval: Automatic
  channel: stable
  name: ibm-cpd-edb
  source: ibm-cpd-edb-operator-catalog
  sourceNamespace: openshift-marketplace
EOF</pre></td></tr>
</table>

- Execution Engine for Apache Hadoop

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

Image location	Required subscriptions
IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** labels:** ** app.kubernetes.io/instance: ibm-cpd-hadoop-operator-catalog-subscription** ** app.kubernetes.io/managed-by: ibm-cpd-hadoop-operator** ** app.kubernetes.io/name: ibm-cpd-hadoop-operator-catalog-subscription** ** name: ibm-cpd-hadoop-operator-catalog-subscription** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** **spec:** ** channel: v1.0** ** installPlanApproval: Automatic** ** name: ibm-cpd-hadoop** ** source: ibm-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**
Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** labels:** ** app.kubernetes.io/instance: ibm-cpd-hadoop-operator-catalog-subscription** ** app.kubernetes.io/managed-by: ibm-cpd-hadoop-operator** ** app.kubernetes.io/name: ibm-cpd-hadoop-operator-catalog-subscription** ** name: ibm-cpd-hadoop-operator-catalog-subscription** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** **spec:** ** channel: v1.0** ** installPlanApproval: Automatic** ** name: ibm-cpd-hadoop** ** source: ibm-cpd-hadoop-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**

- > Financial Services Workbench

  Not applicable.

- > IBM Match 360 with Watson

  Create the appropriate operator subscription for your environment:

Image location	Required subscriptions
IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** name: ibm-mdm-operator-subscription** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** ** labels:** ** app.kubernetes.io/instance: ibm-mdm-operator-subscription** ** app.kubernetes.io/managed-by: ibm-mdm-operator** ** app.kubernetes.io/name: ibm-mdm-operator-subscription** **spec:** ** channel: alpha** ** installPlanApproval: Automatic** ** name: ibm-mdm** ** source: ibm-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**

Image location	Required subscriptions

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-mdm-operator-subscription`
	`  namespace: ibm-common-services|cpd-operators    # Pick the project that contains the Cloud Pak for Data operator`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-mdm-operator-subscription`
	`  app.kubernetes.io/managed-by: ibm-mdm-operator`
	`  app.kubernetes.io/name: ibm-mdm-operator-subscription`
	`spec:`
	`  channel: alpha`
	`  installPlanApproval: Automatic`
	`  name: ibm-mdm`
	`  source: ibm-mdm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- Jupyter Notebooks with Python 3.7 for GPU

The same operator is used for the Jupyter Notebooks with Python 3.7 for GPU service and the Jupyter Notebooks with R 3.6 service. You only need to create this subscription once.

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-cpd-ws-runtimes-operator-catalog-subscription`
	`  app.kubernetes.io/managed-by: ibm-cpd-ws-runtimes-operator`
	`  app.kubernetes.io/name: ibm-cpd-ws-runtimes-operator`
	`  name: ibm-cpd-ws-runtimes-operator`
	`  namespace: ibm-common-services|cpd-operators    # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-ws-runtimes`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-cpd-ws-runtimes-operator-catalog-subscription`
	`  app.kubernetes.io/managed-by: ibm-cpd-ws-runtimes-operator`
	`  app.kubernetes.io/name: ibm-cpd-ws-runtimes-operator`
	`  name: ibm-cpd-ws-runtimes-operator`
	`  namespace: ibm-common-services|cpd-operators    # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-ws-runtimes`
	`  source: ibm-cpd-ws-runtimes-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- Jupyter Notebooks with R 3.6

The same operator is used for the Jupyter Notebooks with Python 3.7 for GPU service and the Jupyter Notebooks with R 3.6 service. You only need to create this subscription once.

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-cpd-ws-runtimes-operator-catalog-subscription`
	`  app.kubernetes.io/managed-by: ibm-cpd-ws-runtimes-operator`
	`  app.kubernetes.io/name: ibm-cpd-ws-runtimes-operator`
	`  name: ibm-cpd-ws-runtimes-operator`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-ws-runtimes`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-cpd-ws-runtimes-operator-catalog-subscription`
	`  app.kubernetes.io/managed-by: ibm-cpd-ws-runtimes-operator`
	`  app.kubernetes.io/name: ibm-cpd-ws-runtimes-operator`
	`  name: ibm-cpd-ws-runtimes-operator`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-ws-runtimes`
	`  source: ibm-cpd-ws-runtimes-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- > MongoDB

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscriptions:
	1. Create the MongoDB Enterprise (third-party) operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-mongodb-enterprise-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: stable`
	`  name: mongodb-enterprise`
	`  installPlanApproval: Automatic`
	`  source: ibm-mongodb-enterprise-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`
	
	2. Create the MongoDB (Cloud Pak for Data) operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-cpd-mongodb-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: stable`
	`  name: ibm-cpd-mongodb`
	`  installPlanApproval: Automatic`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Image location	Required subscriptions

Private container registry	If you are pulling images from a private container registry, create the following operator subscriptions:
	1. Create the MongoDB Enterprise (third-party) operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-mongodb-enterprise-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: stable`
	`  name: mongodb-enterprise`
	`  installPlanApproval: Automatic`
	`  source: ibm-mongodb-enterprise-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`
	
	2. Create the MongoDB (Cloud Pak for Data) operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-cpd-mongodb-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: stable`
	`  name: ibm-cpd-mongodb`
	`  installPlanApproval: Automatic`
	`  source: ibm-cpd-mongodb-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- > OpenPages

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-cpd-openpages-operator`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-openpages-operator`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`
	
	If you want OpenPages to automatically provision a Db2 database, you must also create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-db2aaservice-catalog-subscription`
	`  generation: 1`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  name: ibm-db2aaservice-cp4d-operator`
	`  installPlanApproval: Automatic`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`  startingCSV: ibm-db2aaservice-cp4d-operator.v1.0.0`
	`EOF`

Image location	Required subscriptions

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-cpd-openpages-operator`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-openpages-operator`
	`  source: ibm-cpd-openpages-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`
	
	If you want OpenPages to automatically provision a Db2 database, you must also create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-db2aaservice-catalog-subscription`
	`  generation: 1`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  name: ibm-db2aaservice-cp4d-operator`
	`  installPlanApproval: Automatic`
	`  source: ibm-db2aaservice-cp4d-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`  startingCSV: ibm-db2aaservice-cp4d-operator.v1.0.0`
	`EOF`

- › Planning Analytics

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-planning-analytics-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  name: ibm-planning-analytics-operator`
	`  installPlanApproval: Automatic`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-planning-analytics-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  name: ibm-planning-analytics-operator`
	`  installPlanApproval: Automatic`
	`  source: ibm-planning-analytics-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- › Product Master

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-productmaster-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: alpha`
	`  name: ibm-cpd-productmaster`
	`  installPlanApproval: Automatic`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-productmaster-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: alpha`
	`  name: ibm-cpd-productmaster`
	`  installPlanApproval: Automatic`
	`  source: ibm-productmaster-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- RStudio Server with R 3.6

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-cpd-rstudio-operator-catalog-subscription`
	`  app.kubernetes.io/managed-by: ibm-cpd-rstudio-operator`
	`  app.kubernetes.io/name: ibm-cpd-rstudio-operator-catalog-subscription`
	`  name: ibm-cpd-rstudio-operator-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-rstudio`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-cpd-rstudio-operator-catalog-subscription`
	`  app.kubernetes.io/managed-by: ibm-cpd-rstudio-operator`
	`  app.kubernetes.io/name: ibm-cpd-rstudio-operator-catalog-subscription`
	`  name: ibm-cpd-rstudio-operator-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-rstudio`
	`  source: ibm-cpd-rstudio-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- SPSS Modeler

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

Image location	Required subscriptions
IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** labels:** ** app.kubernetes.io/instance: ibm-cpd-spss-operator-catalog-subscription** ** app.kubernetes.io/managed-by: ibm-cpd-spss-operator** ** app.kubernetes.io/name: ibm-cpd-spss-operator-catalog-subscription** ** name: ibm-cpd-spss-operator-catalog-subscription** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** **spec:** ** channel: v1.0** ** installPlanApproval: Automatic** ** name: ibm-cpd-spss** ** source: ibm-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**
Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** labels:** ** app.kubernetes.io/instance: ibm-cpd-spss-operator-catalog-subscription** ** app.kubernetes.io/managed-by: ibm-cpd-spss-operator** ** app.kubernetes.io/name: ibm-cpd-spss-operator-catalog-subscription** ** name: ibm-cpd-spss-operator-catalog-subscription** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** **spec:** ** channel: v1.0** ** installPlanApproval: Automatic** ** name: ibm-cpd-spss** ** source: ibm-cpd-spss-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**

- > Virtual Data Pipeline

  Not applicable

- > Watson Knowledge Catalog

  Create the appropriate operator subscription for your environment:

Image location	Required subscriptions
IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** labels:** ** app.kubernetes.io/instance: ibm-cpd-wkc-operator-catalog-subscription** ** app.kubernetes.io/managed-by: ibm-cpd-wkc-operator** ** app.kubernetes.io/name: ibm-cpd-wkc-operator-catalog-subscription** ** name: ibm-cpd-wkc-operator-catalog-subscription** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** **spec:** ** channel: v1.0** ** installPlanApproval: Automatic** ** name: ibm-cpd-wkc** ** source: ibm-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**

Image location	Required subscriptions

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  labels:`
	`  app.kubernetes.io/instance:  ibm-cpd-wkc-operator-catalog-subscription`
	`  app.kubernetes.io/managed-by: ibm-cpd-wkc-operator`
	`  app.kubernetes.io/name:  ibm-cpd-wkc-operator-catalog-subscription`
	`  name: ibm-cpd-wkc-operator-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v1.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-wkc`
	`  source: ibm-cpd-wkc-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- Watson Machine Learning

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-cpd-wml-operator-subscription`
	`  app.kubernetes.io/managed-by: ibm-cpd-wml-operator`
	`  app.kubernetes.io/name: ibm-cpd-wml-operator-subscription`
	`  name: ibm-cpd-wml-operator-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: alpha`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-wml-operator`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-cpd-wml-operator-subscription`
	`  app.kubernetes.io/managed-by: ibm-cpd-wml-operator`
	`  app.kubernetes.io/name: ibm-cpd-wml-operator-subscription`
	`  name: ibm-cpd-wml-operator-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: alpha`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-wml-operator`
	`  source: ibm-cpd-wml-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- Watson Machine Learning Accelerator

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-cpd-wml-accelerator-operator`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  name: ibm-cpd-wml-accelerator-operator`
	`  channel: WML-Accelerator-2.3`
	`  installPlanApproval: Automatic`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Image location	Required subscriptions

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-cpd-wml-accelerator-operator`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  name: ibm-cpd-wml-accelerator-operator`
	`  channel: WML-Accelerator-2.3`
	`  installPlanApproval: Automatic`
	`  source: ibm-cpd-wml-accelerator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- > Watson OpenScale

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-watson-openscale-operator-subscription`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-watson-openscale-operator-subscription`
	`  app.kubernetes.io/managed-by: ibm-watson-openscale-operator`
	`  app.kubernetes.io/name: ibm-watson-openscale-operator-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: alpha`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-wos`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  name: ibm-watson-openscale-operator-subscription`
	`  labels:`
	`  app.kubernetes.io/instance: ibm-watson-openscale-operator-subscription`
	`  app.kubernetes.io/managed-by: ibm-watson-openscale-operator`
	`  app.kubernetes.io/name: ibm-watson-openscale-operator-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: alpha`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-wos`
	`  source: ibm-openscale-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

- > Watson Studio

Create the appropriate operator subscription for your environment:

Image location	Required subscriptions

IBM Entitled Registry	If you are pulling images from the publicly available registry, create the following operator subscription. (Ensure that you update the `namespace` parameter to specify the correct Red Hat OpenShift project.)
	
	`cat <<EOF |oc apply -f -`
	`apiVersion: operators.coreos.com/v1alpha1`
	`kind: Subscription`
	`metadata:`
	`  annotations:`
	`  name: ibm-cpd-ws-operator-catalog-subscription`
	`  namespace: ibm-common-services|cpd-operators  # Pick the project that contains the Cloud Pak for Data operator`
	`spec:`
	`  channel: v2.0`
	`  installPlanApproval: Automatic`
	`  name: ibm-cpd-wsl`
	`  source: ibm-operator-catalog`
	`  sourceNamespace: openshift-marketplace`
	`EOF`

Image location	Required subscriptions
Private container registry	If you are pulling images from a private container registry, create the following operator subscription. (Ensure that you update the **namespace** parameter to specify the correct Red Hat OpenShift project.) **cat <<EOF |oc apply -f -** **apiVersion: operators.coreos.com/v1alpha1** **kind: Subscription** **metadata:** ** annotations:** ** name: ibm-cpd-ws-operator-catalog-subscription** ** namespace: ibm-common-services|cpd-operators # Pick the project that contains the Cloud Pak for Data operator** **spec:** ** channel: v2.0** ** installPlanApproval: Automatic** ** name: ibm-cpd-wsl** ** source: ibm-cpd-ws-operator-catalog** ** sourceNamespace: openshift-marketplace** **EOF**

Previous topic: Installing IBM Cloud Pak foundational services
Next topic: Creating custom security context constraints for services

Creating custom security context constraints for services

Most Cloud Pak for Data services use the **restricted** security context constraint (SCC) that is provided by Red Hat® OpenShift® Container Platform. However, if you plan to install certain Cloud Pak for Data services, you might need to create some custom SCCs.

OpenShift provides SCCs that control the actions that a pod can perform and what it can access. OpenShift provides a set of predefined SCCs that can be used, modified, or extended by any administrator. By default, the execution of any container is granted access to the restricted SCC and only the capabilities that are defined by that SCC. For more information, see Managing security context constraints in the Red Hat OpenShift Container Platform documentation.

When you install Cloud Pak for Data services, the default service account is associated with the restricted SCC. Cloud Pak for Data does not support the use of privileged SCCs in OpenShift. However, some Cloud Pak for Data services might require custom SCCs, for example to support IPCs. For more information, see Security context constraints in the IBM® Cloud Platform Common Services documentation.

The following Cloud Pak for Data services use custom SCCs:

- Watson™ Knowledge Catalog
- Db2®
- Db2 Warehouse
- Db2 Big SQL
- Data Virtualization
- OpenPages®

The SCCs are created only one time per cluster. If you have multiple copies of Cloud Pak for Data installed in different namespaces, you must create these SCCs one time for the cluster.

When you install Watson Knowledge Catalog, you must create the custom SCCs manually.

When you install Db2, the Db2 operator creates the custom SCC, service accounts, roles, and role bindings. Db2 Warehouse, Db2 Big SQL, Data Virtualization, and OpenPages use the SCC capabilities in Db2.

For more information about basic security features in Cloud Pak for Data, see Basic security features on Red Hat OpenShift Container Platform.

- **Creating custom security context constraints in Watson Knowledge Catalog**
  Watson Knowledge Catalog requires the use of a custom security context constraint (SCC).
- **Security context constraints in Db2**
  Db2 requires the use of custom security context constraints (SCCs), which are created automatically when you install Db2.

Previous topic: Creating operator subscriptions
Next topic: Changing required node settings

Creating custom security context constraints in Watson Knowledge Catalog

Watson Knowledge Catalog requires the use of a custom security context constraint (SCC).

Custom SCCs in Watson Knowledge Catalog

Administrators can use security context constraints to control permissions for pods on their Red Hat OpenShift cluster. These permissions include actions that a pod can perform and what resources it can access. For Watson Knowledge Catalog, you must create a custom SCC.

Creating custom SCCs in Watson Knowledge Catalog

To create the Watson Knowledge Catalog SCC, complete the following steps:

1. Define the SCC in the file wkc-iis-scc.yaml, as follows:

```
allowHostDirVolumePlugin: false
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: true
allowPrivilegedContainer: false
allowedCapabilities: null
apiVersion: security.openshift.io/v1
defaultAddCapabilities: null
fsGroup:
  type: RunAsAny
kind: SecurityContextConstraints
metadata:
  annotations:
    kubernetes.io/description: WKC/IIS provides all features of the restricted SCC
      but runs as user 10032.
  name: wkc-iis-scc
readOnlyRootFilesystem: false
requiredDropCapabilities:
- KILL
- MKNOD
- SETUID
- SETGID
runAsUser:
  type: MustRunAs
  uid: 10032
seLinuxContext:
  type: MustRunAs
supplementalGroups:
  type: RunAsAny
volumes:
- configMap
- downwardAPI
- emptyDir
- persistentVolumeClaim
- projected
- secret
users:
- system:serviceaccount:{{ namespace }}:wkc-iis-sa
```

- Replace `{{ namespace }}` with the value for the actual namespace where Watson Knowledge Catalog is to be installed.
- If the custom SCC (wkc-iis-scc) already exists in the environment, delete the custom SCC that already exists and create a new custom SCC by using the YAML file from this step. Use the following command to delete the custom SCC: `oc delete scc wkc-iis-scc`

2. Run `oc create` to create the file:

```
$ oc create -f <yaml_file_name.yaml>
```

3. Run the following command to verify that the SCC was created:

```
$ oc get scc wkc-iis-scc
```

For more information about SCCs, see Red Hat - Managing Security Context Constraints.

Security context constraints in Db2

Db2 requires the use of custom security context constraints (SCCs), which are created automatically when you install Db2.

SYS_RESOURCE
: Allows manipulation of reservations, memory allocations, and resource limits. Maximum memory allocation is still constrained by the memory cgroup (memcg) limit, which cannot be overridden by this sys-capability. The Db2 database engine needs this sys-capability to increase the resource limits (IE.ulimits).

IPC_OWNER
: Bypasses permission checks for operations on IPC objects. Even when the IPC kernel parameters are set to maximum values on the hosts/worker nodes, the Db2 engine still tries to dynamically throttle those values. This system capability is provided in addition to sharing IPC namespace with the host.

SYS_NICE
: Allows changing process priorities. Because each container has its own PID namespace, this capability applies to that container only. The Db2 database engine relies on process thread prioritization to ensure that Work Load Management (WLM) and Fast Communications Manager (FCM) processing is prioritized over generic agent work.

CHOWN
: Necessary to run **chown** to change ownership of files/directories in persistent volumes.

DAC_OVERRIDE
: Bypasses permission checks for file read, write, and execute.

FSETID
: Prevents the clearing of the setuid and setgid mode bits when a file is modified.

FOWNER
: Bypasses permission checks on operations that normally require the file system UID of the process to match the UID of the file (for example, chmod(2), utime(2)), excluding those operations that are covered by CAP_DAC_OVERRIDE and CAP_DAC_READ_SEARCH.

SETGID
: Necessary to run Db2 engine processes with escalated group privileges.

SETUID
: Necessary to run Db2 engine processes with escalated user privileges.

SETFCAP
: Used to set capabilities on files.

SETPCAP
: Used to set capabilities on processes.

SYS_CHROOT
: Necessary to use the **chroot** command.

KILL
: Bypasses permission checks for sending signals. Necessary for signal handling during process management.

AUDIT_WRITE
: Required to write records to the kernel auditing log when SELinux is enabled.

Changing required node settings

Some services that run on IBM® Cloud Pak for Data require specific settings on the nodes in the cluster. To ensure that the cluster has the required settings for these services, an operating system administrator with `root` privileges must review and adjust the settings on the appropriate nodes in the cluster.

Machine Config Operator
: The Machine Config Operator is a cluster-level operator that you can use to manage the operating system and keep the cluster up to date and configured.
For more information, see Using MachineConfig objects to configure nodes.

Node Tuning Operator
: You can use the Node Tuning Operator to manage node-level tuning.
On Red Hat® OpenShift®, you can use the Node Tuning Operator to manage node-level profiles. For more information, see Using the Node Tuning Operator.

Node settings for services

The following table shows the node settings that require changes for some services, with links to instructions for changing the settings.

Node settings	Services that require changes to the setting	Environments	Instructions
HAProxy timeout settings for the load balancer	• Watson Knowledge Catalog • OpenPages® • Also recommended if you are working with large data sets or you have slower network speeds.	All environments	• Load balancer timeout settings
CRI-O container settings	• Watson Knowledge Catalog • Watson Studio	All environments except IBM Cloud	• CRI-O container settings
Kernel parameter settings	• Watson Knowledge Catalog • Watson Studio • Db2® • Db2 Warehouse • Db2 Big SQL • Data Virtualization	All environments	• Kernel parameter settings • Deploying Db2 with limited privileges • Deploying Db2 Warehouse with limited privileges • Updating kernel semaphore settings - Db2 Big SQL • Preparing to install the service (Data Virtualization)
GPU settings	• Jupyter Notebooks with Python 3.7 for GPU • Watson Machine Learning Accelerator (requires that the NVIDIA GPU Operator is installed)	All environments	• GPU node settings • Installing the NVIDIA GPU Operator

Load balancer timeout settings

To prevent connections from being closed before processes complete, you might need to adjust the timeout settings on your load balancer node. The recommended timeout is at least 5 minutes. In some situations, you might need to set the timeout even higher. For more information about timeout settings in Watson Knowledge Catalog, see Processes time out before completing.

This setting is required if you plan to install the Watson Knowledge Catalog service or the OpenPages service. However, this setting is also recommended if you are working with large data sets or you have slower network speeds.

The following steps assume that you are using HAProxy. If you are using a different load balancer, see the documentation for your load balancer.

On premises or private cloud

1. On the load balancer node, check the HAProxy timeout settings in the /etc/haproxy/haproxy.cfg file.
The recommended values are at least:

```
timeout client          300s
timeout server          300s
```

2. If the timeout values are less than 300 seconds (5 minutes), update the values:
   - To change the `timeout client` setting, enter the following command:

     ```
     sed -i -e "/timeout client/s/ [0-9].*/ 5m/" /etc/haproxy/haproxy.cfg
     ```

   - To change the `timeout server` setting, enter the following command:

     ```
     sed -i -e "/timeout server/s/ [0-9].*/ 5m/" /etc/haproxy/haproxy.cfg
     ```

3. Run the following command to apply the changes that you made to the HAProxy configuration:

```
systemctl restart haproxy
```

On IBM Cloud

If you are setting HAProxy timeout settings for Cloud Pak for Data on IBM Cloud, you can configure route timeouts by using the **oc annotate** command.

1. Use the following command to set the server-side timeout for the HAProxy route to 360 seconds:

   ```
   oc annotate route zen-cpd --overwrite haproxy.router.openshift.io/timeout=360s
   ```

   If you don't provide the units, **ms** is the default.

2. Optionally, customize other route-specific settings. For more information, see Route-specific annotations.

Note: On a Virtual Private Cloud (VPC) Gen2 cluster, the load balancer timeout is set to 30s by default. If you use the **annotate** command to set the timeout value greater than 50s, it will be set to 50s. You cannot customize the timeout value to be greater than 50s. The server might time out during long running transactions. For more information, see Connection timeouts.

CRI-O container settings

To ensure that services can run correctly, you must adjust values in the CRI-O container settings to specify the maximum number of processes and the maximum number of open files.

These settings are required if you are using the CRI-O container runtime, which is the default on the OpenShift Container Platform.

Note: You do not need to adjust these settings on IBM Cloud.
To change CRI-O settings, you modify the contents of the **crio.conf** file and pass those updates to your nodes as a machine config.

1. Obtain a copy of the existing crio.conf file from a node. For example, run the following command, replacing *$node* with one of the worker nodes:

   ```
   scp core@$node:/etc/crio/crio.conf /tmp/crio.conf
   ```

   If the crio.conf file doesn't exist in the path /etc/crio/crio.conf, use the path /etc/crio/crio.conf.d/00-default instead.

   If you don't have the access by using the **scp** command, ask your cluster administrator for the crio.conf file.

   Make sure that you obtain the latest version of the crio.conf file. You can verify that the file is the latest version by running the **oc get mcp** command and verifying that the worker node is not being updated (UPDATING = False).

2. In the crio.conf file, make the following changes in the **[crio.runtime]** section (uncomment the lines if necessary):
   - To set the maximum number of open files, change the default_ulimits setting to at least **66560**, as follows:

     ```
     ......
     [crio.runtime]
     default_ulimits = [
             "nofile=66560:66560"
     ]
     ......
     ```

   - To set the maximum number of processes, change the pids_limit setting to at least **12288**, as follows:

     ```
     ......
     # Maximum number of processes allowed in a container.
     pids_limit = 12288
     ......
     ```

3. Create a **machineconfig** object YAML file, as follows, and apply it.

   ```
   cat << EOF | oc apply -f -
   apiVersion: machineconfiguration.openshift.io/v1
   kind: MachineConfig
   metadata:
   labels:
     machineconfiguration.openshift.io/role: worker
   name: 99-worker-cp4d-crio-conf
   spec:
   config:
     ignition:
       version: 3.1.0
     storage:
       files:
       - contents:
           source: data:text/plain;charset=utf-8;base64,SPECIFY-CONTENTS-of-/tmp/crio_encoded.txt-HERE
         filesystem: root
         mode: 0644
         path: /etc/crio/crio.conf
   EOF
   ```

4. Monitor all of the nodes to ensure that the changes are applied, by using the following command:

   ```
   watch oc get nodes
   ```

   You can also use the following command to confirm that the MachineConfig sync is complete:

   ```
   watch oc get mcp
   ```

Kernel parameter settings

To ensure that certain services can run correctly, you must verify the kernel parameters. These settings are required for all deployments; however, they depend on the machine RAM size and the OS page size. The following steps assume that you have worker nodes with 64 GB of RAM on an x86 platform with a 4 K OS page size. If the worker nodes have 128 GB of RAM each, you must double the values for **kernel.shmmax** and **kernel.shmall**.

- Virtual memory limit (**vm.max_map_count**)
- Message limits (**kernel.msgmax**, **kernel.msgmnb**, and **kernel.msgmni**)
- Shared memory limits (**kernel.shmmax**, **kernel.shmall**, and **kernel.shmmni**)
  The following settings are recommended:
  - **kernel.shmmni**: 256 * *<size of RAM in GB>*
  - **kernel.shmmax**: *<size of RAM in bytes>*
  - **kernel.shmall**: 2 * *<size of RAM in the default OS system page size>*
- Semaphore limits (**kernel.sem**)
  As of Red Hat Enterprise Linux® version 7.8 and Red Hat Enterprise Linux version 8.1, the **kernel.shmmni**, **kernel.msgmni**, and **kernel.semmni** settings in **kernel.sem** must be set to 32768. If the boot parameter **ipcmni_extend** is specified, then the maximum value is 8388608 while the minimum value is 32768. Use 256 * *<size of RAM in GB>* to calculate possible values for **kernel.shmmni** and **kernel.semmni**. Use 1024 * *<size of RAM in GB>* to calculate a possible value for **kernel.msgmni**. For more information, see On RHEL servers, changing the semaphore value fails with a message "setting key "kernel.sem": Numerical result out of range".

  - The **kernel.sem** value for SEMMNS must be 1024000 for Watson Knowledge Catalog service.
  - The **kernel.sem** value for SEMOPM must be at least 100 for Data Virtualization service.

For more information about changing Kernel node settings for Db2, see Deploying Db2 with limited privileges and for Db2 Warehouse, see Updating kernel semaphore settings - Db2 Big SQL.
Use the Node Tuning Operator to change the Kernel parameter settings. The following steps affect all services and all worker nodes on the cluster. You might need to manage node-level profiles for each worker node in the cluster based on the services that are installed. You can limit node tuning to specific nodes. For more information, see Managing nodes.

1. Create a custom node-level tune with the following content.
   Important: If your current settings are less than the recommendations, adjust the settings. The following command assumes that you have worker nodes with 64 GB of RAM.

   ```
   cat <<EOF |oc apply -f -
   apiVersion: tuned.openshift.io/v1
   kind: Tuned
   metadata:
     name: cp4d-wkc-ipc
     namespace: openshift-cluster-node-tuning-operator
   spec:
     profile:
     - name: cp4d-wkc-ipc
       data: |
         [main]
         summary=Tune IPC Kernel parameters on OpenShift Worker Nodes running WKC Pods
         [sysctl]
         kernel.shmall = 33554432
         kernel.shmmax = 68719476736
         kernel.shmmni = 32768
         kernel.sem = 250 1024000 100 32768
         kernel.msgmax = 65536
         kernel.msgmnb = 65536
         kernel.msgmni = 32768
         vm.max_map_count = 262144
     recommend:
     - match:
       - label: node-role.kubernetes.io/worker
       priority: 10
       profile: cp4d-wkc-ipc
   EOF
   ```

2. Configure **kubelet** to allow Db2U to make **syscalls** as needed:
   a. Update all of the nodes to use a custom **KubletConfig**:

      ```
      cat << EOF | oc apply -f -
      apiVersion: machineconfiguration.openshift.io/v1
      kind: KubeletConfig
      metadata:
        name: db2u-kubelet
      spec:
        machineConfigPoolSelector:
          matchLabels:
            db2u-kubelet: sysctl
        kubeletConfig:
          allowedUnsafeSysctls:
            - "kernel.msg*"
            - "kernel.shm*"
            - "kernel.sem"
      EOF
      ```

   b. Update the label on the **machineconfigpool**:

      ```
      oc label machineconfigpool worker db2u-kubelet=sysctl
      ```

   c. Wait for the cluster to reboot. Then, run the following command to verify that the **machineconfigpool** is updated:

      ```
      oc get machineconfigpool
      ```

      The command should return output with the following format:

      ```
      NAME     CONFIG   UPDATED   UPDATING   DEGRADED   MACHINECOUNT   READYMACHINECOUNT   UPDATEDMACHINECOUNT   DEGRADEDMACHIN
      master   master   True      False      False      3              3                   3                     0
      worker   worker   False     True       False      5              1                   1                     0
      ```

      Wait until all of the worker nodes are updated and ready.

Previous topic: Creating custom security context constraints for services

Setting up the scheduling service on your cluster

The scheduling service is a cluster-wide pod scheduling service that you can install on your IBM® Cloud Pak for Data cluster.

Important: If you plan to install the Watson™ Machine Learning Accelerator service on your cluster, you must install the scheduling service.

Overview of the scheduling service

The scheduling service offers enhancements over the default Kubernetes scheduler, including:

Quota enforcement
: This feature enables you to programmatically enforce the quotas that you set for Cloud Pak for Data or for various Cloud Pak for Data services.
For details on quota enforcement, see Managing the platform.

Co-scheduling of pods
: This feature is provided for the Watson Machine Learning Accelerator service.
Parallel and AI workloads can co-schedule pods to

  - Guarantee that all pods can start
  - Remove resource deadlock
  - Enable workloads to grow and shrink
  - Support reclaiming pods in the event of resource contention

GPU sharing
: This feature is provided for the Watson Machine Learning Accelerator service.
The scheduling service allows competing groups to share GPUs, which improves GPU utilization. Sharing policies govern how to resolve resource contention.

- **Installing the scheduling service**
  A Red Hat® OpenShift® cluster administrator can install the scheduling service in the IBM Cloud Pak® foundational services operator namespace.

Installing the scheduling service

A Red Hat® OpenShift® cluster administrator can install the scheduling service in the IBM Cloud Pak® foundational services operator namespace.

Permissions you need for this task
: You must be a cluster administrator.

When you need to complete this task
: If you plan to install the Watson™ Machine Learning Accelerator service, you must install the scheduling service.
In general, it is strongly recommended that you install the scheduling service, because it enables you to programmatically enforce the quotas that you set for the Cloud Pak for Data control plane or various Cloud Pak for Data services. For details on quota enforcement, see Monitoring the platform.

Information you need to complete this task

- The scheduling service is installed in the same project as the IBM Cloud Pak foundational services, typically `ibm-common-services`.
- The scheduling service uses the following storage classes. If you don't use these storage classes on your cluster, ensure that you have a storage class with an equivalent definition:
  - **OpenShift Container Storage:** `ocs-storagecluster-cephfs`
  - **NFS:** `managed-nfs-storage`
  - **Portworx:** `portworx-shared-gp3`

Before you begin

Verify that you completed these tasks before you install the scheduling service.

1. The `ibm-common-services` namespace exists and has the required operator group. For details, see Creating projects (namespaces) on Red Hat OpenShift Container Platform.
2. For environments that use a private container registry, such as air-gapped environments, the scheduling service images are mirrored to the private container registry. For details, see Mirroring images to your container registry.
3. The cluster is configured to pull the software images. For details, see Configuring your cluster to pull Cloud Pak for Data images.
4. The scheduling service operator subscription exists. For details, see Creating operator subscriptions.

If these tasks are not complete, the scheduling service installation will fail.

Procedure

Creating the required cluster role binding

1. Log in to Red Hat OpenShift Container Platform as a user with sufficient permissions to complete the task:

   ```
   oc login OpenShift_URL:port
   ```

2. Run the following command to add the `ibm-cpd-scheduling-operator-kube-sched-crb` cluster role binding to the `system` service account.
   The following command uses the recommended project for the IBM Cloud Pak foundational services (`ibm-common-services`). If you installed IBM Cloud Pak foundational services in a different project, edit the command to specify the correct project.

3. ```
   oc adm policy add-cluster-role-to-user \
   system:kube-scheduler system:serviceaccount:ibm-common-services:ibm-cpd-scheduling-operator \
   --rolebinding-name=ibm-cpd-scheduling-operator-kube-sched-crb-ibm-common-services
   ```

Installing the scheduling service

To install the scheduling service:

1. Log in to Red Hat OpenShift Container Platform as a user with sufficient permissions to complete the task:

   ```
   oc login OpenShift_URL:port
   ```

2. Create a **Scheduling** custom resource to install the scheduling service.
   The recommended storage class names are described in Setting up shared persistent storage.

   Create a custom resource with the following format.

   ```
   cat <<EOF |oc apply -f -
   apiVersion: scheduler.spectrumcomputing.ibm.com/v1
   kind: Scheduling
   metadata:
     labels:
       release: cpd-scheduler
       velero.io/exclude-from-backup: "true"
     name: ibm-cpd-scheduler
   spec:
     appVersion: 1.2.1
     version: 1.2.1
     cluster:
       pvc:
         dynamicStorage: true
         size: 10G
     license:
       accept: true
       license: Enterprise|Standard        # Specify the license you purchased
     registry: cp.icr.io/cp/cpd
     releasename: ibm-cpd-scheduler
     storageClass: storage-class-name      # See the guidance in "Information you need to complete this task"
     scheduler:
       image: ibm-cpd-scheduler
       imagePullPolicy: Always
       replicas: 1
       resources:
         limits:
           cpu: "1"
           memory: 4G
         requests:
           cpu: "1"
           memory: 4G
     agent:
       image: ibm-cpd-scheduler-agent
       imagePullPolicy: Always
       resources:
         limits:
           cpu: 200m
           memory: 750M
         requests:
           cpu: 200m
           memory: 750M
     mwebhook:
       image: ibm-cpd-scheduler-mutate-webhook
       imagePullPolicy: Always
       replicas: 1
       resources:
         limits:
           cpu: 200m
           memory: 1G
         requests:
           cpu: 200m
           memory: 1G
     vwebhook:
       image: ibm-cpd-scheduler-webhook
       imagePullPolicy: Always
       replicas: 1
       resources:
         limits:
           cpu: 200m
           memory: 1G
         requests:
           cpu: 200m
           memory: 1G
   EOF
   ```

   When you create the custom resource, the scheduling service operator installs scheduling service.

Verifying the installation

When you create the custom resource, the scheduling service operator processes the contents of the custom resource and starts up the microservices that comprise scheduling service, including the **Scheduling**. (The **Scheduling** is defined by the **ibm-cpd-scheduler**.) The scheduling service is installed when the **Scheduling** status is `Completed`.
   ```

To check the status of the installation:

1. Change to the project where you installed the scheduling service:

   ```
   oc project ibm-common-services
   ```

2. Get the status of the scheduling service (`ibm-cpd-scheduler`):

   ```
   oc get scheduling -o jsonpath='{.items[0].status.cpd-schedulingStatus} {"\n"}'
   ```

   The scheduling service is ready when the command returns `Completed`.

What to do next

The scheduling service is ready to use. The Cloud Pak for Data control plane and Watson Machine Learning Accelerator will automatically integrate with the scheduling service when they are installed.

You can now install Cloud Pak for Data.

Installing Cloud Pak for Data

When you install IBM® Cloud Pak for Data, you update the IBM Cloud Pak® for Data platform operator and the IBM Cloud Pak foundational services operator to watch the project where you will install IBM Cloud Pak for Data. Then, you create a custom resource to install Cloud Pak for Data in that project.

Permissions you need for this task
: You must be either:

- A cluster administrator
- An administrator of the following projects:
  - The IBM Cloud Pak foundational services project (`ibm-common-services`)
  - The IBM Cloud Pak for Data platform operator project (`cpd-operators` or `ibm-common-services`)
  - The project where you plan to install Cloud Pak for Data

When you need to complete this task
: You must complete this task each time you want to install an instance of Cloud Pak for Data on your cluster.

Before you begin

Ensure that a cluster administrator completed the required pre-installation tasks for your environment. Specifically, verify that a cluster administrator completed the following tasks:

1. If you are using the specialized installation method, Ensure that IBM Cloud Pak foundational services is installed. For details, see Installing IBM Cloud Pak foundational services.
2. For environments that use a private container registry, such as air-gapped environments, the Cloud Pak for Data software images are mirrored to the private container registry. For details, see Mirroring images to your container registry.
3. The cluster is configured to pull the software images. For details, see Configuring your cluster to pull Cloud Pak for Data images.
4. The Cloud Pak for Data operator subscription and the `IBM Namespace Scope Operator` subscription exist. For details, see Creating operator subscriptions.

If you do not complete these steps, the Cloud Pak for Data installation will fail.

Procedure

To install Cloud Pak for Data:

1. Log in to the Red Hat® OpenShift® Container Platform as a user with sufficient permissions to complete the task:

   ```
   oc login OpenShift_URL:port
   ```

2. Enable the IBM Cloud Pak for Data platform operator and the IBM Cloud Pak foundational services operator to watch the project where you will install IBM Cloud Pak for Data:

   - > Express installations

     Create an operand request to grant permission to the IBM Cloud Pak for Data platform operator and the IBM Cloud Pak foundational services operator to manage the project where you plan to install Cloud Pak for Data:

     ```
     cat <<EOF |oc apply -f -
     apiVersion: operator.ibm.com/v1alpha1
     kind: OperandRequest
     metadata:
       name: empty-request
       namespace: cpd-instance        # Replace with the project where you will install Cloud Pak for Data
     spec:
       requests: []
     EOF
     ```

   - > Specialized installations

     Update the `IBM NamespaceScope Operator` in the Cloud Pak for Data operators project to watch the project where you plan to install Cloud Pak for Data.

     Edit the `namespaceMembers` list to add the project where you plan to install Cloud Pak for Data. For example, if you plan to install Cloud Pak for Data in the `cpd-instance` project, add that project to the list:

```
cat <<EOF |oc apply -f -
apiVersion: operator.ibm.com/v1
kind: NamespaceScope
metadata:
  name: cpd-operators
  namespace: cpd-operators        # (Default) Replace with the Cloud Pak for Data platform operator project name
spec:
  namespaceMembers:
  - cpd-operators                 # (Default) Replace with the Cloud Pak for Data platform operator project name
  - cpd-instance                  # Replace with the project where you will install Cloud Pak for Data
EOF
```

3. Create a custom resource to install Cloud Pak for Data.
   Create a custom resource with the following format:

```
cat <<EOF |oc apply -f -
apiVersion: cpd.ibm.com/v1
kind: Ibmcpd
metadata:
  name: ibmcpd-cr                                 # This is the recommended name, but you can change it
  namespace: cpd-instance                         # Replace with the project where you will install Cloud Pak for Data
spec:
  license:
    accept: true
    license: Enterprise|Standard                  # Specify the Cloud Pak for Data license you purchased
  storageClass: RWX-storage-class                 # Replace with the name of a RWX storage class
  zenCoreMetadbStorageClass: RWO-storage-class    # (Recommended) Replace with the name of a RWO storage class
  version: "4.0.1"
EOF
```

   Best practice: The **zenCoreMetadbStorageClass** setting is optional but strongly recommended for reliability. Specify the ReadWriteOnce (RWO) storage class to use for Cloud Pak for Data metadata storage. Ideally, this storage class points to block storage.
   If you do not specify this setting, the storage class that you specified for the **storageClass** value is used. If you do not want to specify this setting, remove or comment out this line.

Verifying the installation

When you create the custom resource, the IBM Cloud Pak for Data platform operator processes the contents of the custom resource and starts up the microservices that comprise the Cloud Pak for Data control plane, including the **zenservice**, which is defined by the **lite-cr** custom resource definition.

To check the status of the installation:

1. Change to the project where you installed Cloud Pak for Data. For example:

   ```
   oc project cpd-instance
   ```

2. Get the status of the control plane (**lite-cr**):

   ```
   oc get ZenService lite-cr -o jsonpath="{.status.zenStatus}{'\n'}"
   ```

   The Cloud Pak for Data control plane is ready when the command returns `Completed`.

3. Get the URL of the Cloud Pak for Data web client:

   ```
   oc get ZenService lite-cr -o jsonpath="{.status.url}{'\n'}"
   ```

   The URL has the following format:

   ```
   https://cpd-namespace.apps.OCP-default-domain
   ```

4. Get the initial password for the **admin** user:

   ```
   oc extract secret/admin-user-details --keys=initial_admin_password --to=-
   ```

   Important: Save the output of this command so that you can log in to the web client. It is strongly recommended that you change the initial password the first time that you log in to the web client.

Related reference

- Pre-installation tasks
- Post-installation tasks
- Uninstalling Cloud Pak for Data

Post-installation tasks

After you install Cloud Pak for Data, complete the following tasks.

- **Integrating with the IAM Service**
  By default, IBM Cloud Pak for Data user records are stored in an internal repository database. However, it is strongly recommended that you use an enterprise-grade password management solution, such as single sign-on (SSO) or LDAP.
- **Creating a custom route to the platform**
  After you install IBM Cloud Pak for Data, you can optionally customize the route by which users access the Cloud Pak for Data web client.
- **Securing communication ports**
  To ensure secure transmission of network traffic to and from the Cloud Pak for Data cluster, you need to configure the communication ports used by the network.

- **Setting up the Cloud Pak for Data web client**
  After you install Cloud Pak for Data, you can configure the web client to add users and set up email notifications.

Related reference

- Pre-installation tasks
- Installing Cloud Pak for Data
- Uninstalling Cloud Pak for Data

Integrating with the IAM Service

By default, IBM® Cloud Pak for Data user records are stored in an internal repository database. However, it is strongly recommended that you use an enterprise-grade password management solution, such as single sign-on (SSO) or LDAP.

If you use LDAP, you can choose between the following options:

Mechanism	Benefits	Drawbacks
LDAP integration provided by Cloud Pak for Data	You can use LDAP with or without SAML SSO. You can choose the level of integration with the LDAP server. You can use LDAP to: • Validate users' credentials • Manage access to the platform	You can connect to a single LDAP server from each instance of Cloud Pak for Data. The LDAP configuration cannot be shared across Cloud Pak for Data instances or used by any other IBM Cloud Paks on the cluster.
LDAP integration provided by the Identity and Access Management Service (IAM Service) in IBM Cloud Pak® foundational services	You can connect to multiple LDAP servers, and the connections can be used by multiple instances of Cloud Pak for Data or other IBM Cloud Paks on the cluster.	Do not use this method if you have multiple LDAP servers that must be isolated from each other. For example, you maintain two instances of Cloud Pak for Data for different groups of users. Each group of users is managed by a different LDAP server, and you don't want the users to be able to see information about users in the other LDAP server.

To use the LDAP integration provided by Cloud Pak for Data, see Connecting to your identity provider.

Permissions you need for this task

You must be either:

- A cluster administrator
- An administrator of the following projects:
  - The project where IBM Cloud Pak foundational services is installed (`ibm-common-services`)
  - The project where the IBM Cloud Pak for Data platform operator is installed (either `ibm-common-services` or `cpd-operators`)
  - The project where Cloud Pak for Data is installed

When you need to complete this task

If you want to use the LDAP integration provided by the IAM Service, you must integrate Cloud Pak for Data with the IAM Service before you onboard users or create user groups.

When you integrate with the IAM Service, you delegate all authentication to the IAM Service. If you onboard users before you integrate with the IAM Service, existing users might not be able to log in to Cloud Pak for Data.

About this task

Important: Integrating with the IAM Service is irreversible.
Contact IBM Software support to reset Cloud Pak for Data to the previous state.

Procedure

1. Log in to Red Hat® OpenShift® Container Platform as a user with sufficient permissions to complete the task:

   ```
   oc login OpenShift_URL:port
   ```

2. Modify the `ZenService` custom resource to add the `iamIntegration: true` entry:
   a. Run the following command to get the name of the platform custom resource:

      ```
      oc get Ibmcpd -n Cloud-Pak-for-Data-project
      ```

      By default, the custom resource name is `ibmcpd-cr`.

   b. Run the following command to edit the platform custom resource:

      ```
      oc edit Ibmcpd custom-resource-name
      ```

   c. Add the `iamIntegration: true` entry to the custom resource:

      ```
      apiVersion: cpd.ibm.com/v1
      kind: Ibmcpd
      metadata:
        name: ibmcpd-cr
        namespace: cpd-instance                          # The project where Cloud Pak for Data is installed
      spec:
        csNamespace: ibm-common-services
        version: 4.0.0
      ```

```
  license:
    accept: true
    license: Enterprise
  storageClass: RWX-storage-class                    # The RWX storage class you specified during installation
  zenCoreMetaDbStorageClass: RWO-storage-class       # The RWO storage class you specified during installation
  cloudpakfordata: true
  iamIntegration: true
```

Note: In the preceding example, the cluster uses custom storage class names. Your custom resource file might use the **storageVendor** setting instead of the **storageClass** and **zenCoreMetaDbStorageClass** settings.

d. Save your changes to the **ZenService** custom resource. For example, `:wq`

3. Check the status of the **ZenService** custom resource:

```
oc get ZenService custom-resource-name -o jsonpath="{.status}"
```

The command triggers a reconciliation in the **Zen** operator:

```
{"conditions":[{"lastTransitionTime":"2021-06-20T01:05:55Z","message":"Running reconciliation",
"reason":"Running","status":"True","type":"Running"}],"url":"cloud-pak-for-data-URL",
"zenOperatorBuildNumber":"zen operator build 305","zenStatus":"InProgress"}
```

It might take up to 20 minutes for the process to complete if the IAM Service needs to be started and configured.

Tip: You can follow the logs generated by the **Zen** operator pod.
The location of the **Zen** operator pod depends on whether IBM Cloud Pak foundational services and the IBM Cloud Pak for Data platform operator are installed in the same project (**ibm-common-services**) or separate projects (**cpd-operators**).

```
oc logs -n project-name $(oc get pod -n project-name -l name=ibm-zen-operator -o jsonpath='{.items[0].metadata.name'}) -f
```

4. Wait for the **ZenService** custom resource to return the following status:

```
{"conditions":[{"ansibleResult":{"changed":22,"completion":"2021-06-12T06:57:56.861621","failures":0,
"ok":288,"skipped":324},"lastTransitionTime":"2021-06-20T01:05:55Z","message":"Awaiting next reconciliation",
"reason":"Successful","status":"True","type":"Running"}],"url":"cloud-pak-for-data-URL",
"zenOperatorBuildNumber":"zen operator build 305","zenStatus":"Completed"}
```

5. Confirm that the IAM Service is set up:
   a. Go to the Cloud Pak for Data web client.
      Tip: If you don't know the URL, you can run the following command to get the route to the web client:

      ```
      oc get ZenService lite-cr -o jsonpath="{.status.url}{'\n'}"
      ```

   b. Verify that the login page includes the following options:
      - Enterprise LDAP
      - OpenShift authentication
      - IBM provided credentials (admin only)
6. Get the initial password for the **admin** user from the IAM Service:

```
oc extract -n ibm-common-services secret/platform-auth-idp-credentials --keys=admin_password --to=-
```

Creating a custom route to the platform

After you install IBM® Cloud Pak for Data, you can optionally customize the route by which users access the Cloud Pak for Data web client.

Before you begin

Required permissions: To complete this task, you must be an administrator of the project (namespace) where Cloud Pak for Data is installed.

About this task

In Red Hat® OpenShift®, a route is how you expose a service. A route is an externally reachable hostname, such as **https://www.ibm.com**.

By default, the route to the Cloud Pak for Data web client has the following format:

https://cpd-*namespace*.apps.*OCP-default-domain*/zen/

You can create a custom route to override the default URL.

Procedure

1. Log in to your Red Hat OpenShift cluster as a project administrator:

   ```
   oc login OpenShift_URL:Port
   ```

2. Change to the project where Cloud Pak for Data is installed:

   ```
   oc project project_name
   ```

3. Run the following command to create the new route:

   ```
   oc create route passthrough route_name --hostname hostname --service=ibm-nginx-svc
   ```

   Replace *route_name* with the label you want to use for the route, and replace *hostname* with the FQDN you want to assign to the route. The following example command creates a route called **mycompany** with a FQDN of **mycompany.com**:

```
oc create route passthrough mycompany --hostname mycompany.com --service=ibm-nginix-svc
```

4. Run the following command to annotate the route:

   ```
   oc annotate route route_name haproxy.router.openshift.io/balance=roundrobin
   ```

   Replace *route_name* with the value that you specified in the preceding step.

   Users can now access the Cloud Pak for Data through the route that you created.
5. Run the following command to ensure that any HTTP requests are automatically redirected to HTTPS:

   ```
   oc patch route route_name -p "{\"spec\":{\"tls\":{\"insecureEdgeTerminationPolicy\":\"Redirect\"}}}"
   ```

Securing communication ports

To ensure secure transmission of network traffic to and from the Cloud Pak for Data cluster, you need to configure the communication ports used by the network.

Cluster ports

The primary port is what the Red Hat® OpenShift® router exposes. See Configuring and managing cluster networking for Red Hat OpenShift Container Platform 4.6 for details.

Ports for services

When you provision a new service or integration on your Cloud Pak for Data cluster, the services might require connections to be made from outside the cluster. For example, you might require connections when you access databases, or run data virtualization through an ODBC/JDBC connection. If the service or integration requires connections to be made to the cluster, locate the port numbers from each service's Details page and open those network ports. Each port is TCP, and randomly allocated between the 30000-32767 range.

Setting up the Cloud Pak for Data web client

After you install Cloud Pak for Data, you can configure the web client to add users and set up email notifications.

Before you begin

You will need the web client URL provided by the Cloud Pak for Data installation.

Tip: The web client URL is an OpenShift® route, which can be viewed by your cluster administrator by entering the `oc get routes` command. See Temporarily disabling the route to the platform for details on manually disabling or creating one.

Procedure

To configure the Cloud Pak for Data web client:

1. Sign in to the Cloud Pak for Data web client as the default administrator. The default user name is `admin`, and the default password is `password`.
2. Change the password for the `admin` user:
   a. From the menu, click Administer > User management.
   b. On the Users page, select the admin user and select Edit user from the actions menu.
   c. Change the password and specify an email address.

What to do next

It is strongly recommended that you complete the following tasks before you give users access to the web client:

- Using a custom TLS certificate for HTTPS connections
- Configuring single sign-on
- Connecting to your LDAP server
- Displaying a terms and conditions prompt
- Enabling email notifications
- Enabling users to access the web client from platform-generated emails

- **Using a custom TLS certificate for HTTPS connections**
  The Cloud Pak for Data installation includes a self-signed TLS certificate that can be used to enable HTTPS connections. By default, this certificate is untrusted by all HTTPS clients. However, you can replace the default certificate with your own TLS certificate.
- **Configuring single sign-on**
  You can use Security Assertion Markup Language (SAML) for single sign-on (SSO) to the IBM Cloud Pak for Data web client.
- **Changing shared credentials settings**
  A Red Hat OpenShift Container Platform project administrator can optionally disable the option to use shared credentials in connections in platform connections, projects, and catalogs.
- **Setting the idle session timeout**
  You can adjust the idle session timeout for IBM Cloud Pak for Data in accordance with your security and compliance requirements. If a user leaves their session idle in a web browser for the specified length of time, the user is automatically logged out of the web client.
- **Restricting the list of storage classes that are available to an instance of Cloud Pak for Data**
  After you install IBM Cloud Pak for Data, you can optionally restrict the list of storage classes that end users can see and select in the web client. By default, users

can see all of the storage classes that are defined on the cluster. However, you might want to prevent users from selecting certain storage classes when creating new storage volumes or deploying service instances.

- **Displaying a terms and conditions prompt**
  If you need users to accept terms and conditions before they use the web client, you can enable a dialog that prompts users to accept the terms and conditions before they can log in to the web client. For example, you might need to enable the prompt to comply with the Federal Information Security Management Act (FISMA) regulations.
- **Enabling email notifications**
  You can configure a connection to your SMTP server so that Cloud Pak for Data can send email to users.
- **Enabling users to access the web client from platform-generated emails**
  Some services in IBM Cloud Pak for Data generate notifications. For example, collaborators in an analytics project get a notification when assets or new collaborators are added to the project. If you configure a connection to your SMTP server, users can receive these notifications through email. To ensure that these emails include active links to the web client, you must add the `URL_PREFIX` for your deployment to the Cloud Pak for Data product-configmap.

Using a custom TLS certificate for HTTPS connections

The Cloud Pak for Data installation includes a self-signed TLS certificate that can be used to enable HTTPS connections. By default, this certificate is untrusted by all HTTPS clients. However, you can replace the default certificate with your own TLS certificate.

IBM® Cloud Pak for Data exposes one HTTPS port as the primary access point for the web client and for API requests. On Red Hat® OpenShift®, the port is exposed as an OpenShift route.

Before you begin

Required permissions

To complete this task, you must have one of the following roles:

- Red Hat OpenShift cluster administrator
- Red Hat OpenShift project administrator on the project where Cloud Pak for Data is installed

To complete this task, you must have your own certificate and private key file that meet the following requirements:

- Both files are in PEM format.
- The certificate is named cert.crt.
  The certificate can be a bundle that contains your server, intermediates, and root certificates concatenated (in the proper order) into one file. The necessary certificates must be enabled as trusted certificates on the clients that connect to the cluster.
- The private key is named cert.key.

Procedure

To replace the default TLS certificate with your custom TLS certificate:

1. Place the cert.crt and cert.key files in the same directory on your local file system.
2. Change to the directory where the files are located.
3. Connect to your OpenShift cluster:

   ```
   oc login OpenShift_URL:port
   ```

4. Set the context to the project where Cloud Pak for Data is deployed:

   ```
   oc project Project_name
   ```

5. Create a secret to store your certificate files:

   ```
   oc create secret generic external-tls-secret --from-file=cert.crt=./cert.crt --from-file=cert.key=./cert.key --dry-run -o
   yaml | oc apply -f -
   ```

   Important: Do not change the name of the secret. You must use the name `external-tls-secret`.
   Wait for the command to return a message that the secret was created:

   ```
   secret/external-tls-secret created
   ```

   Then, wait another minute to ensure that `kubelet` has sufficient time to detect where the secret will be used and to mount the secret to the `ibm-nginx` pods.

6. Reload `ibm-nginx`:

   ```
   for i in `oc get pods | grep ibm-nginx |  cut -f1 -d\ `; do oc exec ${i} -- /scripts/reload.sh; done
   ```

   The output should be similar to the following output:

   ```
   reloading nginx conf
   Setting up ssl certificate files...
    Custom ssl certificate files were found. Processing them...
   lrwxrwxrwx. 1 1000321000 root 50 DATE-AND-TIME /nginx_data/defaults.d/external-server.conf ->
   /nginx_data/defaults.d/external-server.active.conf
   nginx: the configuration file /usr/local/openresty/nginx/conf/nginx.conf syntax is ok
   nginx: configuration file /usr/local/openresty/nginx/conf/nginx.conf test is successful
   TIMESTAMP [notice] 76#76: signal process started
   reloading nginx conf
   Setting up ssl certificate files...
   Custom ssl certificate files were found. Processing them...
   lrwxrwxrwx. 1 1000321000 root 50 DATE-AND-TIME /nginx_data/defaults.d/external-server.conf ->
   /nginx_data/defaults.d/external-server.active.conf
   nginx: the configuration file /usr/local/openresty/nginx/conf/nginx.conf syntax is ok
   ```

```
nginx: configuration file /usr/local/openresty/nginx/conf/nginx.conf test is successful
TIMESTAMP [notice] 76#76: signal process started
reloading nginx conf
Setting up ssl certificate files...
Custom ssl certificate files were found. Processing them...
lrwxrwxrwx. 1 1000321000 root 50 DATE-AND-TIME /nginx_data/defaults.d/external-server.conf ->
/nginx_data/defaults.d/external-server.active.conf
nginx: the configuration file /usr/local/openresty/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /usr/local/openresty/nginx/conf/nginx.conf test is successful
TIMESTAMP [notice] 76#76: signal process started
```

Verify that the certificate files were found.

Configuring single sign-on

You can use Security Assertion Markup Language (SAML) for single sign-on (SSO) to the IBM® Cloud Pak for Data web client.

Before you begin

You must have an existing SAML identity provider (IdP). Work with your IdP administrator to gather the following information:

Parameter	Description	Value
entryPoint	The URL of the login page for your identity provider.	
fieldToAuthenticate	The name of the parameter you use to authenticate with the identity provider, such as `emailAddress` or `username`. If you plan to use LDAP and SAML, ensure that you use the same attribute to identify users. This parameter should have the same value as the User search field in your LDAP configuration.	
spCert	The private key used to sign SAML requests to the identity provider. The certificate corresponding to this key needs to be set when you register Cloud Pak for Data with your identity provider so that the SAML requests can be verified by your identity provider. If you do not specify a certificate, the requests won't be signed.	Remove the "BEGIN PRIVATE KEY" and "END PRIVATE KEY" lines and provide the private key as a single line.
idpCert	The certificate provided by the identity provider to verify SAML responses from the identity provider.	Remove the "BEGIN CERTIFICATE" and "END CERTIFICATE" lines and provide the certificate as a single line.
issuer	The name that you want to use to register Cloud Pak for Data with your identity provider. If you do not specify a value, the default (`ibm_privatecloud`) is used.	
identifierFormat	The format of requests from Cloud Pak for Data to the identity provider. The format must be supported by the identity provider. If you do not specify a format, the default format (`urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress`) is used	
callbackUrl	An approved URL (that you set with the SAML identity provider) to redirect users to after they successfully authenticate using SSO. For example, to redirect successfully authenticated users to the Cloud Pak for Data landing page, you can specify `https://cluster/auth/login/sso/callback`.	
disableRequestedAuthnContext	A boolean parameter for AD FS client authentication. If set the true, the authentication context is disabled so that the IDP determines the method of authentication. If you do not specify a value, then the default is false.	

About this task

To configure SSO, you must specify information about your Identity Provider in a configuration file. Use the preceding table to gather the required information that you need to supply in the configuration file.

Important: It is strongly recommended that you complete this task before you add users to Cloud Pak for Data. If you have already added users to Cloud Pak for Data you must re-add the users with their SAML ID to enable them to use SSO.

Procedure

1. Log in to your Red Hat® OpenShift® cluster as a project administrator:

   ```
   oc login OpenShift_URL:port
   ```

2. Enable SAML by running the following command:

   ```
   oc exec -it -n namespace \
   $(oc get pod -n namespace -l component=usermgmt | tail -1 | cut -f1 -d\ ) \
   -- bash -c "vi /user-home/_global_/config/saml/samlConfig.json"
   ```

   Replace *namespace* with the namespace where Cloud Pak for Data is deployed.

3. In the samlConfig.json file, specify the appropriate values for your environment.

   ```
   {
     "entryPoint": "",
     "fieldToAuthenticate": "",
     "spCert": "",
     "idpCert": "",
     "issuer": "",
     "identifierFormat": "",
   ```

```
  "callbackUrl": ""
}
```

4. Save your changes to samlConfig.json.
   a. Press `Esc`.
   b. Press `:`.
   c. Enter `:x`.
   d. Press `Enter`.
5. Run the following command to delete the **usermgmt** pods:

```
oc delete pods -l component=usermgmt
```

What to do next

Wait several minutes before you attempt to log in to the web client. The instructions restart the **usermgmt** pods. If the pods are not running, you will not be able to log in.

If you previously added users to Cloud Pak for Data, you must re-add the users with their SAML ID to enable them to use SSO. To add users:

1. Go directly to the web client log in page by appending the following path to your Cloud Pak for Data URL: **/auth/login/zen-login.html**.
2. Log in to the web client as the admin user or another administrator with user management permissions.
3. Add users with their SAML IDs. For details, see Managing users.

Disabling SAML

Procedure

1. Disable SAML by running the following command:

```
oc exec -it -n namespace \
$(oc get pod -n namespace -l component=usermgmt | tail -1 | cut -f1 -d\ ) \
-- bash -c "rm /user-home/_global_/config/saml/samlConfig.json"
```

   Replace *namespace* with the namespace where Cloud Pak for Data is deployed.

2. Run the following command to delete the **usermgmt** pods:

```
oc delete pods -l component=usermgmt
```

Changing shared credentials settings

A Red Hat® OpenShift® Container Platform project administrator can optionally disable the option to use shared credentials in connections in platform connections, projects, and catalogs.

About this task

When a connection is created with shared credentials, all users access the connection with the same credentials. However, because the credentials are shared, it is difficult to audit access to the connection, to identify the source of data loss, or identify the source of a security breach.

Permissions you need for this task
: You must be a Red Hat OpenShift Container Platform project or cluster administrator.

When you need to complete this task
: If individual accountability is required, especially by industry-specific regulations that your organization must comply with, it is recommended that you disable shared credentials on the platform.
However, this setting is only applicable if the Cloud Pak for Data common core services are installed. If you can see Data > Platform connections in the navigation, you should determine whether you want to allow the use of shared credentials or whether you want to require each user to provide personal credentials.

  If you want to require each user to provide personal credentials, disable the option to use shared credentials.

Procedure

1. Log in to your Red Hat OpenShift Container Platform cluster as a user with sufficient permissions to complete the task:

```
oc login OpenShift_URL:port
```

2. Change to the project where Cloud Pak for Data is installed:

```
oc project Project_name
```

3. Run the following command to edit the Cloud Pak for Data config-wdp-connect-connection file:

```
oc edit configmap config-wdp-connect-connection
```

4. Change the value of allow-shared-credentials parameter from `true` to `false`.

```
allow-shared-credentials:false
```

5. Save your changes to the config-wdp-connect-connection file.
   For example, if you are using **vi**, enter:

```
:wq
```

Setting the idle session timeout

You can adjust the idle session timeout for IBM® Cloud Pak for Data in accordance with your security and compliance requirements. If a user leaves their session idle in a web browser for the specified length of time, the user is automatically logged out of the web client.

Before you begin

Required permissions

To complete this task, you must have one of the following roles:

- Red Hat® OpenShift® cluster administrator
- Red Hat OpenShift project administrator on the project where Cloud Pak for Data is installed

About this task

By default, Cloud Pak for Data logs users out after 12 hours. You can edit the Cloud Pak for Data `product-configmap` to adjust:

The length of time until a user's session expires (`TOKEN_EXPIRY_TIME`).

The default is 12 hours.

If you set `TOKEN_EXPIRY_TIME: "1"`, a user's session will expire in after 1 hour of inactivity. If you set `TOKEN_EXPIRY_TIME: "0.5"`, a user's session will expire after 30 minutes of inactivity. When the user leaves their session idle for the specified length of time, the user is automatically logged out of the web client.

It is recommended that you set the value between `0.1` and `1`.

The length of time that a user has to refresh their session (`TOKEN_REFRESH_PERIOD`).

The default is 12 hours.

If you set `TOKEN_REFRESH_PERIOD: "1"` and the user's session does not expire, the user's session is automatically refreshed during this 60 minute period. The session is extended based on the value that is set for the `TOKEN_EXPIRY_TIME` parameter. However, after the token refresh period passes, the user must log back into the web client when their current session expires.

It is recommended that you set the value between `1` and `24`.

If you don't want to allow users to extend their sessions, set the value of the `TOKEN_REFRESH_PERIOD` parameter to a value less than the value of the `TOKEN_EXPIRY_TIME` parameter.

For example, as an administrator, you configure:

```
TOKEN_EXPIRY_TIME: "0.5"
TOKEN_REFRESH_PERIOD: "2"
```

If a user starts work at 8 AM and logs in to the web client, the user must be active in the web session within 30 minutes for their token to be refreshed:

- If the user stops using the web client at 8:10 and attempts to use the web client again until 8:41, the user must re-authenticate to the web client because their session expired.
- If the user remains active in their session and their token refreshes at 9:59 AM, their session will last until 10:29 AM. However, when the session expires at 10:29, the user must re-authenticate to the web client because the token refresh period expired.

Procedure

1. Log in to your OpenShift cluster:

   ```
   oc login OpenShift_URL:port
   ```

2. Change to the project where Cloud Pak for Data is deployed:

   ```
   oc project Project
   ```

3. Run the following command to edit the Cloud Pak for Data `product-configmap`:

   ```
   oc edit configmap product-configmap
   ```

4. Add an entry for the `TOKEN_EXPIRY_TIME` parameter to the `data` section of the `product-configmap` file. For example:

   ```
   data:
     ...
     TOKEN_EXPIRY_TIME: "1"
     ...
   ```

5. Add an entry for the `TOKEN_REFRESH_PERIOD` parameter to the `data` section of the `product-configmap` file. For example:

   ```
   data:
     ...
     TOKEN_REFRESH_PERIOD: "1"
     ...
   ```

6. Save your changes to the product-configmap file.
   For example, if you are using `vi`, enter:

   ```
   :wq
   ```

7. You must restart the `usermgmt` pods for the changes to take effect. To restart the pods, run the following command:

   ```
   oc delete pod -l component=usermgmt
   ```

Restricting the list of storage classes that are available to an instance of Cloud Pak for Data

After you install IBM® Cloud Pak for Data, you can optionally restrict the list of storage classes that end users can see and select in the web client. By default, users can see all of the storage classes that are defined on the cluster. However, you might want to prevent users from selecting certain storage classes when creating new storage volumes or deploying service instances.

Before you begin

Required permissions: To complete this task, you must be an administrator of the project (namespace) where Cloud Pak for Data is installed.

Work with your cluster administrator to determine which storage classes to display in the web client. As part of this discussion, you might need to consider which services you plan to deploy on this instance of Cloud Pak for Data.

About this task

You can restrict the list of storage classes by creating a list of allowed storage classes in the Cloud Pak for Data product-configmap file.

Procedure

1. Log in to your Red Hat® OpenShift® cluster as a project administrator:

   ```
   oc login OpenShift_URL:Port
   ```

2. Change to the project where Cloud Pak for Data is installed:

   ```
   oc project Project_name
   ```

3. Run the following command to edit the Cloud Pak for Data `product-configmap`:

   ```
   oc edit configmap product-configmap
   ```

4. Add an entry for the `ALLOWED_STORAGE_CLASSES` parameter to the `data` section of the `product-configmap` file. For example:

   ```
   data:
     ...
    ALLOWED_STORAGE_CLASSES: "storage-class-1, storage-class-2, storage-class-3"
     ...
   ```

   Specify multiple storage classes as a comma separated list.

5. Save your changes to the product-configmap file.
   For example, if you are using `vi`, enter:

   ```
   :wq
   ```

Displaying a terms and conditions prompt

If you need users to accept terms and conditions before they use the web client, you can enable a dialog that prompts users to accept the terms and conditions before they can log in to the web client. For example, you might need to enable the prompt to comply with the Federal Information Security Management Act (FISMA) regulations.

About this task

When you configure the web client to display a terms and conditions prompt, you must specify the following information:

- The header text for the dialog
- The terms and conditions that the user must accept
- The prompt that the user must click to acknowledge that they accept the terms and conditions

Procedure

1. Log in to your Red Hat® OpenShift® cluster as a project administrator:

   ```
   oc login OpenShift_URL:port
   ```

2. Create the login-dialog.json configuration file:

Property	Description
enabled	Set enabled to true to enable the dialog. For example: `"enabled": true,` Valid values: true or false

Property	Description
headerText	Specify the text to display as the header in the dialog. For example: `"headerText": "Terms of use",` Use standard JSON string format.
dialogText	Specify the terms and conditions that the user must agree to before they can access the web client. For example: `"dialogText": "The terms and conditions of use that your user must accept.",` Use standard JSON string format.
acceptText	Specify the text that the user must click to acknowledge that they agree to the terms and conditions. For example: `"acceptText": "I understand and accept the terms",` Use standard JSON string format.

3. Copy the login-dialog.json file into the config directory:

   ```
   oc cp login-dialog.json $(oc get pod -n namespace -l component=usermgmt | tail -1 | cut -f1 -d\ ):/user-home/_global_/config/
   ```

   Replace *namespace* with your project namespace.

Enabling email notifications

You can configure a connection to your SMTP server so that Cloud Pak for Data can send email to users.

About this task

To send emails to users, one of the following services must be installed:

- Watson™ Studio
- Watson Knowledge Catalog

If neither service is installed, the SMTP configuration is not used.

Procedure

To enable Cloud Pak for Data to send email:

1. Log in to the web client as an administrator.
2. From the menu, select Administer > Configure platform.
3. On the SMTP settings page, specify the following information:
   - Your SMTP mail server address.
   - The port number of your SMTP server.
     Important: If you specify a secure port, you must select Use TLS connection so that Cloud Pak for Data can communicate with your SMTP server.
   - Specify the appropriate SMTP credentials for your environment:

Method of sending communications	SMTP server requires authentication	SMTP server does not require authentication
My SMTP server uses a mailer daemon to send communications	You must specify the following fields: • SMTP username • SMTP password	You don't need to specify any fields. However, if you want to override the mailer daemon, you can specify a From account.
My SMTP server uses a default account to send communications	You must specify the following fields: • SMTP username • SMTP password • From account	You must specify the following fields: • From account

4. Click Save. If your SMTP configuration is successful, you will receive a confirmation email.
   - If you specified a From account when you configured the connection to your SMTP server, the confirmation email is sent to the account specified in the From account field.
   - If you did not specify a From account when you configured a connection to your SMTP server, the confirmation email is sent to the account specified in the SMTP username field.

Results

Depending on your configuration, notification emails are sent from one of the following accounts:

- If you specified a From account when you configured the connection to your SMTP server, notifications are sent from the account specified in the From account field.
- If you did not specify a From account when you configured a connection to your SMTP server, notifications are sent from the mailer daemon.

Enabling users to access the web client from platform-generated emails

Some services in IBM® Cloud Pak for Data generate notifications. For example, collaborators in an analytics project get a notification when assets or new collaborators are added to the project. If you configure a connection to your SMTP server, users can receive these notifications through email. To ensure that these emails include active links to the web client, you must add the `URL_PREFIX` for your deployment to the Cloud Pak for Data product-configmap.

About this task

A Red Hat® OpenShift® project (namespace) administrator can edit the Cloud Pak for Data product-configmap to specify the `URL_PREFIX` for your deployment.

The `URL_PREFIX` is the domain name at the beginning of your deployment URL. For example, if your deployment of Cloud Pak for Data is accessible from `https://domain.my.company.com/zen`, your domain name is `domain.my.company.com`. Do not include the protocol in the value that you specify.

If you use the default port, `443`, you do not need to specify the port number in the value for the `URL_PREFIX` parameter. However, if you use a non-standard port, include it in the `URL_PREFIX`. For example, if you use port `31843`, your entry would be:

```
URL_PREFIX: domain.my.company.com:31843
```

Procedure

To enable users to access the web client from platform-generated emails:

1. Log in to your Red Hat OpenShift cluster as a project administrator:

   ```
   oc login OpenShift_URL:port
   ```

2. Change to the project where you installed Cloud Pak for Data:

   ```
   oc project Project
   ```

3. Run the following command to edit the Cloud Pak for Data product-configmap:

   ```
   oc edit cm product-configmap
   ```

4. Add an entry for the `URL_PREFIX` parameter to the `data` section of the product-configmap file. For example:

   ```
   data:
     ...
     SHOW_USER_APPROVAL: "false"
     URL_PREFIX: domain.my.company.com
     ...
   ```

5. Save your changes to the product-configmap file.
   For example, if you are using `vi`, enter:

   ```
   :wq
   ```

   The changes are automatically applied to the platform.

Uninstalling Cloud Pak for Data

A project administrator can uninstall the Cloud Pak for Data control plane.

Permissions you need for this task:
To complete this task, you must be an administrator of the OpenShift® project (Kubernetesnamespace) where:

- The Cloud Pak for Data control plane is installed
- The Cloud Pak for Data operators are installed, either `ibm-common-services` or `cpd-operators`

Procedure

1. Uninstalling dependent services
2. Uninstalling Cloud Pak for Data
3. Uninstalling the operators
   Note: Complete this step only if you want to completely remove the Cloud Pak for Data software from your cluster.

Uninstalling dependent services

All services depend on the Cloud Pak for Data control plane.

Ensure that you uninstall any services that are installed in the same project (or tethered projects) as the control plane before you uninstall the control plane.

For details, see the appropriate service documentation in Services.

Uninstalling Cloud Pak for Data

When you need to complete this task
Complete this task when you want to remove a running instance of the Cloud Pak for Data.
If you installed multiple instances of Cloud Pak for Data on the cluster, you must complete this task for each instance of Cloud Pak for Data that you want to uninstall.

If you plan to uninstall the Cloud Pak for Data operators, you must uninstall all instances of Cloud Pak for Data *before* you uninstall the operators.

To remove Cloud Pak for Data:

1. Log in to your Red Hat® OpenShift cluster as a user with sufficient permissions to complete the task:

   ```
   oc login OpenShift_URL:port
   ```

2. Change to the project where the Cloud Pak for Data control plane is deployed:

   ```
   oc project project_name
   ```

3. Get the name of the custom resource:

   ```
   oc get Ibmcpd -n project_name
   ```

4. Delete the `Ibmcpd` custom resource to remove the Cloud Pak for Data platform:

   ```
   oc delete Ibmcpd custom-resource-name -n project_name
   ```

5. Delete the `zenservice` custom resource to remove the control plane:

   ```
   oc delete zenservice lite-cr -n project_name
   ```

6. Run the following command to verify that the resources that were created by Cloud Pak for Data and the control plane are deleted:

   ```
   oc get all -l "app.kubernetes.io/name in (0020-zen-base, 0015-setup, 0010-infra)"
   ```

Uninstalling the operators

When you need to complete this task
Complete this task only if you want to completely remove Cloud Pak for Data from your cluster.
Complete this task *after* you uninstall the Cloud Pak for Data control plane.

To uninstall the Cloud Pak for Data operators:

1. Log in to your Red Hat OpenShift cluster as a user with sufficient permissions to complete the task:

   ```
   oc login OpenShift_URL:port
   ```

2. Change to the project where Cloud Pak for Data was deployed:

   ```
   oc project project_name
   ```

3. Remove the `zen` operator:

   ```
   oc delete operandrequest zen-service
   ```

   This command removes the `zen` operator from the instance project and from the project where the IBM Cloud Pak® for Data platform operator is installed (either `ibm-common-services` or `cpd-operators`).

4. Delete the IBM Cloud Pak for Data platform operator:
   a. Delete the subscription to the operator:

      ```
      oc delete subscription cpd-operator -n ibm-common-services|cpd-operators
      ```

   b. Delete the CSV for the operator:

      ```
      oc delete csv cpd-platform-operator.v2.0.0 -n ibm-common-services|cpd-operators
      ```

5. Verify that the IBM Cloud Pak for Data platform operator deployment was removed:

   ```
   oc get deploy cpd-platform-operator-manager
   ```

   The command should return the following message:

   ```
   Error from server (NotFound): deployments.apps "cpd-platform-operator-manager" not found
   ```

Uninstalling IBM Cloud Pak foundational services

If you don't have other IBM® Cloud Paks on your cluster, you might want to uninstall IBM Cloud Pak foundational services after you uninstall Cloud Pak for Data. For details, see:

- For connected clusters, see Uninstalling foundational services.
- For air-gapped clusters, see Uninstalling IBM Cloud Pak foundational services in an airgap environment.

Related reference

- Pre-installation tasks
- Installing Cloud Pak for Data
- Post-installation tasks