Part V: Installing

IBM

Tables of Contents

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

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
     Ensure that you install the software on a system that has sufficient resources and that aligns with the guidance in the [System requirements](#). For example, if you do not follow the specified [disk requirements](#), you can run into out of memory errors.

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 or Version 4.8 cluster.

Options	What to do
You already have an OpenShift 4.6 or 4.8 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
     Available in the IBM Storage Suite for IBM Cloud Paks.
IBM Spectrum® Scale Container Native
     IBM Spectrum Scale Container Native Storage Access Version: 5.1.1.3 or later fixes
     Container Storage Interface Version: 2.3.0 or later fixes
     Available in the IBM Storage Suite for IBM Cloud Paks.
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

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

Options	What to do

You know which projects you plan to use when you install the software	1. Review the guidance in [Setting up projects (namespaces) on Red Hat OpenShift Container Platform](#) to:
	  - Ensure that you have the necessary projects on your cluster
	  - Determine whether you need to create operator groups for the projects
	2. Go to [5. Do you have your API key?](#)

You don't know which projects you plan to use when you install the software	1. Review the guidance in [Setting up projects (namespaces) on Red Hat OpenShift Container Platform](#) to:
	  - Determine which projects you need to create on your cluster
	  - Set up the required 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 private 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 private container registry](#) to ensure you have a private 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.

Options	What to do
You are pulling images from the IBM Entitled Registry	1. Follow the guidance in [Configuring your cluster to pull Cloud Pak for Data images](#) to configure the global image pull secret to include your IBM entitlement API key. 2. Go to [8. Creating catalog sources](#).

Options	What to do

You are pulling images from a private container registry	1. Follow the guidance in [Configuring your cluster to pull Cloud Pak for Data images](#) to:
	    a. Configure the global image pull secret to include the credentials of an account that can pull images from the registry.
	    b. Configure an image content source policy.
	2. Go to [8. Creating catalog sources](#).

8. Creating catalog sources

You must create catalog sources to ensure that your cluster uses the correct software images for your environment.

Options	What to do
You are pulling images from the IBM Entitled Registry	1. Review the guidance in [Creating catalog sources](#) to determine which method is appropriate for your environment. 2. Go to [9. Are the IBM Cloud Pak foundational services already installed on your cluster?](#)
You are pulling images from a private container registry	1. Follow the guidance in [Creating catalog sources for a private container registry](#). 2. Go to [9. Are the IBM Cloud Pak foundational services already installed on your cluster?](#)

9. 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.

For information about supported versions of IBM Cloud Pak foundational services, see the [Cloud Pak for Data platform software requirements](#).

Options	What to do
IBM Cloud Pak foundational services Version 3.15 or later is already installed	1. Go to [10. Creating operator subscriptions](#)
An earlier version of IBM Cloud Pak foundational services is installed	1. Follow the guidance in [Installing IBM Cloud Pak foundational services](#). 2. Go to [10. 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 [10. 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 projects. 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 [10. Creating operator subscriptions](#).

10. 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 [11. Do you plan to install services that require custom SCCs?](#)

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

The following services require custom security context constraints:

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

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 [Custom security context constraints for services](#). 2. Go to [12. Do you plan to install services that require specific node settings?](#)
You don't plan to install any of these services	1. Go to [12. Do you plan to install services that require specific node settings?](#)

12. 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 Discovery
- 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, you have large data sets, or you have slower network speeds	1. Change the appropriate node settings. For details, see Changing required node settings. 2. Go to 13. Do you need to install the scheduling service?
You don't plan to install any of these services	1. Go to 13. Do you need to install the scheduling service?

13. 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 14. Installing Cloud Pak for Data.
You don't plan to install the scheduling service	1. Go to 14. Installing Cloud Pak for Data.

14. 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 15. Completing post-installation tasks.

15. 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 16. Installing services.

16. Installing services

You are ready to install services on your cluster.

1. 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 the common core services**
  A project administrator can uninstall the common core services from IBM Cloud Pak for Data.
- **Uninstalling Cloud Pak for Data**
  A project administrator can uninstall the Cloud Pak for Data control plane.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see Documentation for previous 4.0.x refreshes.

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 persistent storage on your Red Hat OpenShift cluster.
3. [Setting up 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 and configure 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 private 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. [Creating catalog sources](#)
   To ensure that your cluster uses the correct software images, you must create the appropriate catalog sources for your environment.
8. [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.
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).
10. [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 use some custom SCCs.
11. [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 the common core services](#)
- [Uninstalling Cloud Pak for Data](#)

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

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)	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 documentation to install OpenShift:
  - [Version 4.6 documentation](#)

- Version 4.8 documentation

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.
What's next Now that you've installed Red Hat OpenShift, you are ready to complete Setting up projects (namespaces) on Red Hat OpenShift Container Platform.

- > IBM Cloud

Managed OpenShift
To install managed OpenShift, use the IBM Cloud catalog to install Red Hat OpenShift on IBM Cloud. Ensure that you select a supported version of Red Hat OpenShift.
Self-managed OpenShift
To install self-managed OpenShift, contact IBM Software Support.

What's next Now that you've installed Red Hat OpenShift, you are ready to complete Setting up projects (namespaces) on Red Hat OpenShift Container Platform.

- > AWS

Managed OpenShift
To install managed OpenShift, see Red Hat OpenShift Service on AWS (ROSA).
Self-managed OpenShift
To install self-managed OpenShift follow the Red Hat OpenShift Container Platform documentation to install OpenShift on AWS:
- Version 4.6 documentation
- Version 4.8 documentation

What's next Now that you've installed Red Hat OpenShift, you are ready to complete Setting up projects (namespaces) on Red Hat OpenShift Container Platform.

- > Azure

Self-managed OpenShift
To install self-managed OpenShift, follow the Red Hat OpenShift Container Platform documentation to install OpenShift on Azure:
- Version 4.6 documentation
- Version 4.8 documentation

What's next Now that you've installed Red Hat OpenShift, you are ready to complete Setting up projects (namespaces) on Red Hat OpenShift Container Platform.

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

For details, follow the Red Hat OpenShift Container Platform documentation to install OpenShift on Google Cloud:
- Version 4.6 documentation
- Version 4.8 documentation
What's next Now that you've installed Red Hat OpenShift, you are ready to complete Setting up projects (namespaces) on Red Hat OpenShift Container Platform.

Next topic: Setting up shared persistent storage
Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see Documentation for previous 4.0.x refreshes.

Setting up shared persistent storage

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

Important: Before you set up your persistent storage, ensure that you review the available planning information:

- Storage considerations
  - Lists the types of storage that are supported by the platform
  - Compares the features and requirements for each type of storage
  - Identifies the deployment environments that are supported for each type of storage
- Storage requirements
  - Identifies which type of storage are supported by each service
  - Specifies the minimum amount of storage that each service requires.

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.

Storage type	What to do

IBM Spectrum® Scale Container Native	Installation
	    For IBM Spectrum Scale and IBM Spectrum Scale Container Storage Interface, see [Installation](#) in the IBM Spectrum Scale Container Native documentation.
	Post-installation set up
	    Create the IBM Spectrum Scale storage class with parameters. For details, see [Setting up IBM Spectrum Scale Container Native storage](#).

Portworx	Installation
	    To install Portworx Enterprise, see [Install Portworx on OpenShift](#)
	
	Post-installation set up
	    You must configure the required storage classes. For details, see [Creating Portworx storage classes](#)

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 classes](#)**
  If you decide to use Portworx as your storage option, Cloud Pak for Data requires the following storage classes. You can set them up manually.
- **[Setting up IBM Spectrum Scale Container Native storage](#)**
  If you decide to use IBM Spectrum Scale Container Native as your storage option, you will need to create the IBM Spectrum Scale storage class with parameters for use 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.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Setting up Portworx storage classes

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

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

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```
`portworx-watson-assistant-sc`	Watson Assistant	```# Watson Assitant - This was previously named portworx-assitant
cat <<EOF	oc create -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
name: portworx-watson-assistant-sc
parameters:
repl: "3"
priority_io: "high"
io_profile: "db_remote"
block_size: "64k"
disable_io_profile_protection: "1"
allowVolumeExpansion: true
provisioner: kubernetes.io/portworx-volume
reclaimPolicy: Retain
volumeBindingMode: Immediate
EOF```
`portworx-informix-sc`	Informix®	```# Informix
cat <<EOF	oc create -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
name: portworx-informix-sc
provisioner: kubernetes.io/portworx-volume
parameters:
repl: "3"
priority_io: high
io_profile: db
block_size: 2048b
allowVolumeExpansion: true
EOF```

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Setting up IBM Spectrum Scale Container Native storage

If you decide to use IBM Spectrum® Scale Container Native as your storage option, you will need to create the IBM Spectrum Scale storage class with parameters for use with IBM® Cloud Pak for Data.

Create storage class

The permissions field should be set to **777** for read, write, and execute on the local file system.

Example:

```
cat <<EOF |oc apply -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
```

```
    name: ibm-spectrum-scale-sc
provisioner: spectrumscale.csi.ibm.com
parameters:
    volBackendFs: "<file-system-name>"  # Replace with the file system name from the CNSA cluster
    clusterId: "<remote-storage-cluster-ID>"  # Replace with Cluster ID of the remote Spectrum Scale Storage Cluster
    permissions: "777"
reclaimPolicy: Delete
EOF
```

Note: You do not need to be in a specific namespace when you create the storage class. Storage classes are cluster wide, so it can be in any namespace at the time.

Field names	Descriptions

`volBackendFs`	`volBackendFs` is the name of the file system mounted by the IBM Spectrum Scale Container Native cluster. It can be found by running the following command:
	
	`oc exec $(oc get pod -o name -l app.kubernetes.io/name=core \`
	`-n ibm-spectrum-scale |head -1) -n ibm-spectrum-scale \`
	`-c gpfs -- mmremotecluster show`
	
	Replace the file system name from the CNSA cluster.

`clusterID`	`clusterID` is the cluster ID of the Spectrum Scale Storage Cluster Storage Cluster (remote cluster). It can be found by running `mmlscluster` on the Spectrum Scale Storage Cluster.
`permissions`	`permissions` must be set to `777` for read, write, and execute on the local file system.

There are additional parameters that you can change. For details, please see Storage Class for creating fileset-based volumes (remote file system) in the Spectrum Scale documentation.
Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see Documentation for previous 4.0.x refreshes.

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 based on the Red Hat® OpenShift® Container Platform project where they are installed.) If you restrict access to the NFS served to specific UIDs, you might encounter errors when installing or running Cloud Pak for Data.
  For information on determining which UIDs are used, see Service UIDs.

- 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`.

Note: If you are installing Cloud Pak for Data from the IBM® Cloud catalog, the NFS export is automatically set to `no_root_squash`.
However, if you are *manually* installing Cloud Pak for Data on IBM Cloud, you must follow the guidance 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 repository.
3. Open a `bash` shell and change to the deploy directory of the repository.
4. 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
   ```

5. Authorize the provisioner by running the following commands.
   a. Create the required role based access control.
      Important: 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
      ```

6. Edit the deployment.yaml file in the deploy directory to specify the following information:
   - The project (namespace) where the NFS provisioner is deployed.
   - The image that corresponds to your Red Hat OpenShift Container Platform architecture:
     - x86-64: `gcr.io/k8s-staging-sig-storage/nfs-subdir-external-provisioner:amd64-linux-v4.0.2`
     - Power®: `gcr.io/k8s-staging-sig-storage/nfs-subdir-external-provisioner:ppc64le-linux-v4.0.2`
     - s390x: `gcr.io/k8s-staging-sig-storage/nfs-subdir-external-provisioner:s390x-linux-v4.0.2`
   - 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: nfs-provisioner-image     # Specify the appropriate image based on your OpenShift architecture
             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
   ```

7. Deploy the NFS provisioner:

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

8. 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.

9. Create the storage class:

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

10. 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 -f test-pod.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 -f test-claim.yaml
```

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Setting up 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 and configure 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 [Deployment architecture](#).

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

Project	Description
`ibm-common-services` (Recommended name, used in subsequent commands)	**Required for all installations.** The project where IBM Cloud Pak® foundational services is installed. If IBM Cloud Pak foundational services is already installed on your cluster, identify the project where it is installed. If IBM Cloud Pak foundational services is not installed on your cluster, `ibm-common-services` is the recommended project name. `ibm-common-services` 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](#). (Using a different project is not recommended for typical installations.) 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` (Recommended name, used in subsequent commands)	**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` (Sample name, used in subsequent commands)	**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](#).

Project	Description
cpd-instance-tether (Sample name, used in subsequent commands)	**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	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 in 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 Custom security context constraints for services.

cpd-operators	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 Custom security context constraints for services.

cpd-instance	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 Custom security context constraints for services.

cpd-instance-tether	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 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. Create the appropriate projects for your environment.
   Important: Review the guidance in **About this task** to ensure that you create the appropriate projects for your environment.

Project name	Command to create
ibm-common-services	`oc new-project ibm-common-services`
cpd-operators	`oc new-project cpd-operators`
cpd-instance	`oc new-project cpd-instance` `cpd-instance` is a sample name. If you don't want to use this name, replace **cpd-instance** with the appropriate name for your environment. You must also replace this name in subsequent commands.
cpd-instance-tether	`oc new-project cpd-instance-tether` `cpd-instance-tether` is a sample name. If you don't want to use this name, replace **cpd-instance-tether** with the appropriate name for your environment. You must also replace this name in subsequent commands.

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

       What's next Now that you've set up the projects on your cluster, ensure that you have your API key. For details, see Obtaining your IBM entitlement API key.

   - > Specialized installation
     - 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
       ```

     - 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
       ```

       What's next Now that you've set up the projects on your cluster, ensure that you have your API key. For details, see Obtaining your IBM entitlement API key.

Previous topic: Setting up shared persistent storage
Next topic: Obtaining your IBM entitlement API key
Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see Documentation for previous 4.0.x refreshes.

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.

What's next Now that you have your API key, determine whether you will mirror the software images to a private container registry. For details, see Mirroring images to your private container registry.

Previous topic: Setting up projects (namespaces) on Red Hat OpenShift Container Platform
Next topic: Mirroring images to your private container registry
Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see Documentation for previous 4.0.x refreshes.

Mirroring images to your private 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 private 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

Even if these situations do not apply to your environment, you should consider using a private container registry if you want to:

- Run security scans against the software images before you install them on your cluster
- Ensure that you have the same images available for multiple deployments, such as development or test environments and production environments

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 private container 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 private 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.
As part of your private container registry set up, ensure that you follow the guidelines for configuring the registry in Image configuration resources in the Red Hat OpenShift Container Platform documentation.

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 private 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 intermediary container 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 private 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 private 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 private 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 container 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
Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see Documentation for previous 4.0.x refreshes.

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 container registry to mirror the images from the IBM® Entitled Registry.

To obtain fixes, security updates, and new functionality, you must install the latest refresh of the software images, such as the control plane, shared cluster components, and services. The steps in this task assume that you are mirroring the latest software images. If you need older versions of the software, see Operator and operand versions.

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

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 container registry

Before you can mirror the images a private container 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 or directories 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.

Recommended directory name	Notes

Recommended directory name	Notes
$HOME/offline/cpd	Primary directory for Cloud Pak for Data software. This directory is required for all installations. To create this directory, run the following command: `mkdir -p $HOME/offline/cpd`
$HOME/offline/cpfs	Directory for IBM Cloud Pak foundational services software. This directory is recommended in the following situations: • You plan to install multiple IBM Cloud Paks. • IBM Cloud Pak foundational services is not installed on your cluster. To create this directory, run the following command: `mkdir -p $HOME/offline/cpfs`

Best practice: If you have multiple IBM Cloud Paks on your cluster, use a separate directory for the Cloud Pak for Data CASE packages (the packages for the Cloud Pak for Data platform and services).

When you run the command to mirror the images, the command mirrors any updated images in the directory. Using a separate directory ensures that you mirror only the images for Cloud Pak for Data.

2. Set the following environment variables:

Variable	Notes
CASE_REPO_PATH	Required for all installations. This environment variable points to the repository where the CASE packages are hosted. To set this environment variable, run: `export CASE_REPO_PATH=https://github.com/IBM/cloud-pak/raw/master/repo/case`
OFFLINEDIR	Required for all installations. This is the directory that you created to store the Cloud Pak for Data CASE packages. To set this environment variable, run: `export OFFLINEDIR=$HOME/offline/cpd` Replace `$HOME/offline/cpd` with the appropriate value for your environment.
OFFLINEDIR_CPFS	Required only if you created a separate directory for the IBM Cloud Pak foundational services CASE package. This is the directory that you created to store the IBM Cloud Pak foundational services CASE package. To set this environment variable, run: `export OFFLINEDIR_CPFS=$HOME/offline/cpfs` Replace `$HOME/offline/cpfs` with the appropriate value for your environment.

3. Downloading the IBM Cloud Pak for Data platform operator CASE package

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

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cp-datacore \
--version 2.0.10 \
--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-2.0.10.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 private 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 container registry.
   PRIVATE_REGISTRY_PASSWORD
   The password of the user who has the required privileges to *push* images to the private container registry.
   PRIVATE_REGISTRY
   The location of the private container 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-2.0.10.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 Version 3.15 or later is not installed on the cluster.
	`cloudctl case save \`
	`--repo ${CASE_REPO_PATH} \`
	`--case ibm-cp-common-services`
	`\`
	`--version 1.11.0 \`
	`--outputdir`
	`${OFFLINEDIR_CPFS}`

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 \`
	`--repo ${CASE_REPO_PATH} \`
	`--case ibm-cpd-scheduling \`
	`--version 1.3.2 \`
	`--outputdir ${OFFLINEDIR}`

6. Downloading service CASE packages

Decide which [services](#) you plan to install on your cluster and download the appropriate CASE packages.

- › Anaconda Repository for IBM Cloud Pak for Data

  Not applicable. For details, see [Installing Anaconda Repository for IBM Cloud Pak for Data](#).

- › Analytics Engine Powered by Apache Spark

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-analyticsengine \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- › Cognos Analytics

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cognos-analytics-prod \
--version 4.0.8 \
--outputdir ${OFFLINEDIR}
```

- › Cognos Dashboards

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cde \
--version 2.0.6 \
--outputdir ${OFFLINEDIR}
```

- › Data Privacy

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-dp \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- › Data Refinery

  The Data Refinery CASE package is automatically downloaded when you download the CASE package for either Watson Knowledge Catalog or Watson Studio.

- › Data Virtualization

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-dv-case \
--version 1.7.6 \
--outputdir ${OFFLINEDIR}
```

- › DataStage

  Download the appropriate package based on your license:

DataStage Enterprise

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-datastage-enterprise \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

DataStage Enterprise Plus

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-datastage \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

- > Db2

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-db2oltp \
--version 4.0.8 \
--outputdir ${OFFLINEDIR}
```

- > Db2 Big SQL

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-bigsql-case \
--version 7.2.6 \
--outputdir ${OFFLINEDIR}
```

- > Db2 Data Gate

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-datagate-prod \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- > Db2 Data Management Console

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-dmc \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- > Db2 Event Store

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

- > Db2 Warehouse

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-db2wh \
--version 4.0.8 \
--outputdir ${OFFLINEDIR}
```

- > Decision Optimization

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-dods \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- > EDB Postgres

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cpd-edb \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- > Execution Engine for Apache Hadoop

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-hadoop \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

- > Financial Services Workbench

  Not applicable. For details, see the [Financial Services Workbench documentation](#).

- > IBM Match 360 with Watson

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-mdm \
--version 1.0.182 \
--outputdir ${OFFLINEDIR}
```

- > Informix

  Download both of the following CASE packages:

  - ```
    cloudctl case save \
    --repo ${CASE_REPO_PATH} \
    --case ibm-informix-operator \
    --version 4.0.5 \
    --outputdir ${OFFLINEDIR}
    ```

  - ```
    cloudctl case save \
    --repo ${CASE_REPO_PATH} \
    --case ibm-informix-cp4d-operator \
    --version 4.0.5 \
    --outputdir ${OFFLINEDIR}
    ```

- > MongoDB

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cpd-mongodb \
--version 4.0.4 \
--outputdir ${OFFLINEDIR}
```

- > OpenPages

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-openpages \
--version 2.1.3+20220205.003421.82040223 \
--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 \
  --repo ${CASE_REPO_PATH} \
  --case ibm-db2aaservice \
  --version 4.0.8 \
  --outputdir ${OFFLINEDIR}
  ```

- > Planning Analytics

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-planning-analytics \
--version 4.0.60476 \
--outputdir ${OFFLINEDIR}
```

- > Product Master

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-productmaster \
--version 1.0.3 \
--outputdir ${OFFLINEDIR}
```

- > RStudio Server with R 3.6

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-rstudio \
--version 1.0.6 \
--outputdir ${OFFLINEDIR}
```

- > SPSS Modeler

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-spss \
--version 1.0.6 \
--outputdir ${OFFLINEDIR}
```

- > Virtual Data Pipeline

Not applicable. For details, see [Installing Virtual Data Pipeline for Cloud Pak for Data](#).

- > Voice Gateway

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-voice-gateway \
--version 1.0.5 \
--outputdir ${OFFLINEDIR}
```

- > Watson Assistant

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-assistant \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- > Watson Assistant for Voice Interaction

Watson Assistant for Voice Interaction is comprised of the following services:
  - Voice Gateway
  - Watson Assistant
  - Watson Speech to Text
  - Watson Text to Speech

Download the appropriate services for your use case.

- > Watson Discovery

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-discovery \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- > Watson Knowledge Catalog

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-wkc \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- > Watson Knowledge Studio

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-ks \
--version 4.0.5 \
--outputdir ${OFFLINEDIR}
```

- > Watson Machine Learning

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-wml-cpd \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

- > Watson Machine Learning Accelerator

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-wml-accelerator \
--version 2.3.6 \
--outputdir ${OFFLINEDIR}
```

- > Watson OpenScale

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-openscale \
--version 2.5.1 \
--outputdir ${OFFLINEDIR}
```

- > Watson Speech to Text

The same package is used for the Watson Speech to Text service and the Watson Text to Speech service. You only need to download the package once.

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
```

```
--case ibm-watson-speech \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

- > Watson Studio

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-wsl \
--version 2.0.6 \
--outputdir ${OFFLINEDIR}
```

- > Watson Studio Runtimes

  The Watson Studio Runtimes CASE package is automatically downloaded when you download the CASE package for Watson Studio.

- > Watson Text to Speech

  The same package is used for the Watson Speech to Text service and the Watson Text to Speech service. You only need to download the package once.

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-speech \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

7. Mirroring the images to the private container registry

To mirror the images:

1. Some CASE packages include images that require entitlement. If you are not entitled to use those images, you must delete the images:
   a. If you did not purchase a license for EDB Postgres Enterprise, run the following command to remove any `edb-postgres-advanced` images from the list of images to mirror.
   If you purchased a license for EDB Postgres Enterprise, skip this step.

   ```
   sed -i -e '/edb-postgres-advanced/d' ${OFFLINEDIR}/ibm-cloud-native-postgresql-4.0.*-images.csv
   ```

   b. If you downloaded the Informix CASE packages, delete the images that you are not entitled to.
   By default, the `ibm-informix-operator` CASE package includes the images for the following editions of Informix:

   - Developer Edition (free)
   - Workgroup Edition
   - Enterprise Edition
   - Advanced Enterprise Edition

   Run the appropriate commands to delete the images for any editions that you are not entitled to:

   Advanced Enterprise Edition
   If you did not purchase a license for Informix Advanced Enterprise Edition, run the following command to delete the Advanced Enterprise Edition images:

   ```
   sed -i -e '/informix-edition-aee/d' ${OFFLINEDIR}/ibm-informix-operator-4.0.*-images.csv
   ```

   Enterprise Edition
   If you did not purchase a license for Informix Enterprise Edition, run the following command to delete the Enterprise Edition images:

   ```
   sed -i -e '/informix-edition-ee/d' ${OFFLINEDIR}/ibm-informix-operator-4.0.*-images.csv
   ```

   Workgroup Edition
   If you did not purchase a license for Informix Workgroup Edition, run the following command to delete the Workgroup Edition images:

   ```
   sed -i -e '/informix-edition-we/d' ${OFFLINEDIR}/ibm-informix-operator-4.0.*-images.csv
   ```

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. Use the Cloud Pak for Data platform CASE package to mirror all of the images to the private container registry:

   ```
   cloudctl case launch \
     --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.10.tgz \
     --inventory cpdPlatformOperator \
     --action mirror-images \
     --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir
   ${OFFLINEDIR}"
   ```

   Remember: The **mirror-images** command mirrors any changed images in the ${OFFLINEDIR}.

4. If you downloaded the IBM Cloud Pak foundational services CASE package, run the following command to mirror the images to the private container registry:

   ```
   cloudctl case launch \
     --case ${OFFLINEDIR_CPFS}/ibm-cp-common-services-1.11.0.tgz \
     --inventory ibmCommonServiceOperatorSetup \
     --action mirror-images \
     --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD} --inputDir
   ${OFFLINEDIR_CPFS}"
   ```

8. Storing the CASE packages

Best practice: Save the CASE packages in a dedicated location to ensure that you can replicate the configuration at any time or on any cluster.
CASE packages specify dependencies as a range, so it is possible to get a later version of the dependencies if you download the CASE package on different dates.

Saving the downloaded CASE packages prevents any discrepancies.

What's next Now that you've mirrored the images to your private container registry, you are ready to complete Configuring your cluster to pull Cloud Pak for Data images.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see Documentation for previous 4.0.x refreshes.

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.

To obtain fixes, security updates, and new functionality, you must install the latest refresh of the software images, such as the control plane, shared cluster components, and services. The steps in this task assume that you are mirroring the latest software images. If you need older versions of the software, see Operator and operand versions.

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 private container registry:

- 1. Installing the software needed to mirror images
- 2. Setting up your environment to download CASE packages
- 3. Downloading the IBM Cloud Pak for Data platform operator CASE package
- 4. Configuring credentials for mirroring images
- 5. Setting up an intermediary container registry
- 6. Downloading shared cluster component CASE packages
- 7. Downloading service CASE packages
- 8. Mirroring the images to the intermediary container registry
- 9. Setting up a workstation to serve images
- 10. Mirroring images to the private container registry
- 11. Storing the CASE packages

1. Installing the software needed to mirror images

To use an intermediary container 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 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 container 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 container registry

Before you can mirror the images a private container 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 or directories 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.

Recommended directory name	Notes
$HOME/offline/cpd	Primary directory for Cloud Pak for Data software. This directory is required for all installations. To create this directory, run the following command: `mkdir -p $HOME/offline/cpd`
$HOME/offline/cpfs	Directory for IBM Cloud Pak foundational services software. This directory is recommended in the following situations: • You plan to install multiple IBM Cloud Paks. • IBM Cloud Pak foundational services is not installed on your cluster. To create this directory, run the following command: `mkdir -p $HOME/offline/cpfs`

Best practice: If you have multiple IBM Cloud Paks on your cluster, use a separate directory for the Cloud Pak for Data CASE packages (the packages for the Cloud Pak for Data platform and services).
When you run the command to mirror the images, the command mirrors any updated images in the directory. Using a separate directory ensures that you mirror only the images for Cloud Pak for Data.

2. Set the following environment variables:

Variable	Notes
CASE_REPO_PATH	Required for all installations. This environment variable points to the repository where the CASE packages are hosted. To set this environment variable, run: `export CASE_REPO_PATH=https://github.com/IBM/cloud-pak/raw/master/repo/case`

Variable	Notes
`OFFLINEDIR`	Required for all installations. This is the directory that you created to store the Cloud Pak for Data CASE packages. To set this environment variable, run: **`export OFFLINEDIR=$HOME/offline/cpd`** Replace **`$HOME/offline/cpd`** with the appropriate value for your environment.
`OFFLINEDIR_CPFS`	Required only if you created a separate directory for the IBM Cloud Pak foundational services CASE package. This is the directory that you created to store the IBM Cloud Pak foundational services CASE package. To set this environment variable, run: **`export OFFLINEDIR_CPFS=$HOME/offline/cpfs`** Replace **`$HOME/offline/cpfs`** with the appropriate value for your environment.

3. Downloading the IBM Cloud Pak for Data platform operator CASE package

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

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cp-datacore \
--version 2.0.10 \
--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-2.0.10.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_HOST`
   > The location of the intermediary container registry on the local host.

   `PORTABLE_REGISTRY_PORT`
   > 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_HOST=localhost
   export PORTABLE_REGISTRY_PORT=15000
   PORTABLE_REGISTRY="${PORTABLE_REGISTRY_HOST}:${PORTABLE_REGISTRY_PORT}"
   export PORTABLE_REGISTRY
   ```

   c. Run the following command to store the credentials:

   ```
   cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.10.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 container 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-2.0.10.tgz \
   --inventory cpdPlatformOperator \
```

```
    --action init-registry \
    --args "--registry ${PORTABLE_REGISTRY_HOST} --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 intermediary container registry, run the following command:

```
cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.10.tgz \
    --inventory cpdPlatformOperator \
    --action start-registry \
    --args "--port ${PORTABLE_REGISTRY_PORT} --dir ${OFFLINEDIR}/imageregistry --image docker.io/library/registry:2"
```

4. To verify that the intermediary container 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 Version 3.15 or later is not installed on the cluster.
	`cloudctl case save \`
	`--repo ${CASE_REPO_PATH} \`
	`--case ibm-cp-common-services`
	`\`
	`--version 1.11.0 \`
	`--outputdir`
	`${OFFLINEDIR_CPFS}`

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 \`
	`--repo ${CASE_REPO_PATH} \`
	`--case ibm-cpd-scheduling \`
	`--version 1.3.2 \`
	`--outputdir ${OFFLINEDIR}`

7. Downloading service CASE packages

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

- ❯ Anaconda Repository for IBM Cloud Pak for Data

    Not applicable. For details, see Installing Anaconda Repository for IBM Cloud Pak for Data.

- ❯ Analytics Engine Powered by Apache Spark

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-analyticsengine \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- ❯ Cognos Analytics

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cognos-analytics-prod \
--version 4.0.8 \
--outputdir ${OFFLINEDIR}
```

- ❯ Cognos Dashboards

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cde \
--version 2.0.6 \
--outputdir ${OFFLINEDIR}
```

- ❯ Data Privacy

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-dp \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

> ⟩
> Data Refinery

The Data Refinery CASE package is automatically downloaded when you download the CASE package for either Watson Knowledge Catalog or Watson Studio.

> ⟩
> Data Virtualization

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-dv-case \
--version 1.7.6 \
--outputdir ${OFFLINEDIR}
```

> ⟩
> DataStage

Download the appropriate package based on your license:

DataStage Enterprise

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-datastage-enterprise \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

DataStage Enterprise Plus

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-datastage \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

> ⟩
> Db2

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-db2oltp \
--version 4.0.8 \
--outputdir ${OFFLINEDIR}
```

> ⟩
> Db2 Big SQL

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-bigsql-case \
--version 7.2.6 \
--outputdir ${OFFLINEDIR}
```

> ⟩
> Db2 Data Gate

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-datagate-prod \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

> ⟩
> Db2 Data Management Console

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-dmc \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

> ⟩
> Db2 Event Store

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

> ⟩
> Db2 Warehouse

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-db2wh \
--version 4.0.8 \
--outputdir ${OFFLINEDIR}
```

> ⟩
> Decision Optimization

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-dods \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- >

  EDB Postgres

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cpd-edb \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- >

  Execution Engine for Apache Hadoop

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-hadoop \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

- >

  Financial Services Workbench

  Not applicable. For details, see the [Financial Services Workbench documentation](#).

- >

  IBM Match 360 with Watson

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-mdm \
--version 1.0.182 \
--outputdir ${OFFLINEDIR}
```

- >

  Informix

  Download both of the following CASE packages:

  - ```
    cloudctl case save \
    --repo ${CASE_REPO_PATH} \
    --case ibm-informix-operator \
    --version 4.0.5 \
    --outputdir ${OFFLINEDIR}
    ```

  - ```
    cloudctl case save \
    --repo ${CASE_REPO_PATH} \
    --case ibm-informix-cp4d-operator \
    --version 4.0.5 \
    --outputdir ${OFFLINEDIR}
    ```

- >

  MongoDB

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cpd-mongodb \
--version 4.0.4 \
--outputdir ${OFFLINEDIR}
```

- >

  OpenPages

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-openpages \
--version 2.1.3+20220205.003421.82040223 \
--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 \
--repo ${CASE_REPO_PATH} \
--case ibm-db2aaservice \
--version 4.0.8 \
--outputdir ${OFFLINEDIR}
```

- >

  Planning Analytics

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-planning-analytics \
--version 4.0.60476 \
--outputdir ${OFFLINEDIR}
```

- >

  Product Master

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-productmaster \
--version 1.0.3 \
--outputdir ${OFFLINEDIR}
```

- > **RStudio Server with R 3.6**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-rstudio \
--version 1.0.6 \
--outputdir ${OFFLINEDIR}
```

- > **SPSS Modeler**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-spss \
--version 1.0.6 \
--outputdir ${OFFLINEDIR}
```

- > **Virtual Data Pipeline**

  Not applicable. For details, see [Installing Virtual Data Pipeline for Cloud Pak for Data](#).

- > **Voice Gateway**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-voice-gateway \
--version 1.0.5 \
--outputdir ${OFFLINEDIR}
```

- > **Watson Assistant**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-assistant \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- > **Watson Assistant for Voice Interaction**

  Watson Assistant for Voice Interaction is comprised of the following services:
  - Voice Gateway
  - Watson Assistant
  - Watson Speech to Text
  - Watson Text to Speech

  Download the appropriate services for your use case.

- > **Watson Discovery**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-discovery \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- > **Watson Knowledge Catalog**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-wkc \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- > **Watson Knowledge Studio**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-ks \
--version 4.0.5 \
--outputdir ${OFFLINEDIR}
```

- > **Watson Machine Learning**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-wml-cpd \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

- > **Watson Machine Learning Accelerator**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-wml-accelerator \
```

```
--version 2.3.6 \
--outputdir ${OFFLINEDIR}
```

- >
  Watson OpenScale

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-openscale \
--version 2.5.1 \
--outputdir ${OFFLINEDIR}
```

- >
  Watson Speech to Text

The same package is used for the Watson Speech to Text service and the Watson Text to Speech service. You only need to download the package once.

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-speech \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

- >
  Watson Studio

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-wsl \
--version 2.0.6 \
--outputdir ${OFFLINEDIR}
```

- >
  Watson Studio Runtimes

The Watson Studio Runtimes CASE package is automatically downloaded when you download the CASE package for Watson Studio.

- >
  Watson Text to Speech

The same package is used for the Watson Speech to Text service and the Watson Text to Speech service. You only need to download the package once.

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-speech \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

8. Mirroring the images to the intermediary container registry

To mirror the images:

1. Some CASE packages include images that require entitlement. If you are not entitled to use those images, you must delete the images:
   a. If you did not purchase a license for EDB Postgres Enterprise, run the following command to remove any `edb-postgres-advanced` images from the list of images to mirror.
   If you purchased a license for EDB Postgres Enterprise, skip this step.

   ```
   sed -i -e '/edb-postgres-advanced/d' ${OFFLINEDIR}/ibm-cloud-native-postgresql-4.0.*-images.csv
   ```

   b. If you downloaded the Informix CASE packages, delete the images that you are not entitled to.
   By default, the `ibm-informix-operator` CASE package includes the images for the following editions of Informix:

   - Developer Edition (free)
   - Workgroup Edition
   - Enterprise Edition
   - Advanced Enterprise Edition

   Run the appropriate commands to delete the images for any editions that you are not entitled to:

   Advanced Enterprise Edition
   If you did not purchase a license for Informix Advanced Enterprise Edition, run the following command to delete the Advanced Enterprise Edition images:

   ```
   sed -i -e '/informix-edition-aee/d' ${OFFLINEDIR}/ibm-informix-operator-4.0.*-images.csv
   ```

   Enterprise Edition
   If you did not purchase a license for Informix Enterprise Edition, run the following command to delete the Enterprise Edition images:

   ```
   sed -i -e '/informix-edition-ee/d' ${OFFLINEDIR}/ibm-informix-operator-4.0.*-images.csv
   ```

   Workgroup Edition
   If you did not purchase a license for Informix Workgroup Edition, run the following command to delete the Workgroup Edition images:

   ```
   sed -i -e '/informix-edition-we/d' ${OFFLINEDIR}/ibm-informix-operator-4.0.*-images.csv
   ```

2. Use the Cloud Pak for Data platform CASE package to mirror all of the images from the OFFLINEDIR directory to the intermediary container registry:

   ```
   cloudctl case launch \
     --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.10.tgz \
     --inventory cpdPlatformOperator \
     --action mirror-images \
     --args "--registry ${PORTABLE_REGISTRY} --inputDir ${OFFLINEDIR}"
   ```

Remember: The **mirror-images** command mirrors any changed images in the ${OFFLINEDIR}.

3. If you downloaded the IBM Cloud Pak foundational services CASE package, mirror the images from the OFFLINEDIR_CPFS directory to the intermediary container registry:

```
cloudctl case launch \
   --case ${OFFLINEDIR_CPFS}/ibm-cp-common-services-1.11.0.tgz \
   --inventory ibmCommonServiceOperatorSetup \
   --action mirror-images \
   --args "--registry ${PORTABLE_REGISTRY} --inputDir ${OFFLINEDIR_CPFS}"
```

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 container registry

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 miror 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 -o ${OFFLINEDIR}/registry.tar`
	
	• For Podman, run:
	
	`podman save docker.io/library/registry:2 -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 container registry, where you can start the intermediary container 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 -o ${OFFLINEDIR}/registry.tar`
	
	• For Podman, run:
	
	`podman save docker.io/library/registry:2 -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 container registry and where you can start the intermediary container 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 intermediary container 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 container 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 container 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-2.0.10.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 container 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-2.0.10.tgz \
  --inventory cpdPlatformOperator \
  --action init-registry \
  --args "--registry ${PORTABLE_REGISTRY_HOST} --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-2.0.10.tgz \
  --inventory cpdPlatformOperator \
  --action start-registry \
  --args "--port ${PORTABLE_REGISTRY_PORT} --dir ${OFFLINEDIR}/imageregistry --image docker.io/library/registry:2"
```

f. To verify that the intermediary container 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 container 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_HOST
   > The location of the intermediary container registry on the local host.

   PORTABLE_REGISTRY_PORT
   > 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_HOST=localhost
   export PORTABLE_REGISTRY_PORT=15000
   PORTABLE_REGISTRY="${PORTABLE_REGISTRY_HOST}:${PORTABLE_REGISTRY_PORT}"
   export PORTABLE_REGISTRY
   ```

   c. Run the following command to store the credentials:

   ```
   cloudctl case launch \
     --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.10.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 private 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 container registry.

   PRIVATE_REGISTRY_PASSWORD
   > The password of the user who has the required privileges to *push* images to the private container registry.

   PRIVATE_REGISTRY
   > The location of the private container 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-2.0.10.tgz \
     --inventory cpdPlatformOperator \
     --action configure-creds-airgap \
     --args "--registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass ${PRIVATE_REGISTRY_PASSWORD}"
   ```

3. Use the Cloud Pak for Data platform CASE package to mirror all of the images to the private container registry:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.10.tgz \
  --inventory cpdPlatformOperator \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR}"
```

4. If you mirrored the IBM Cloud Pak foundational services images to the intermediary container registry, run the following command to mirror the images to the private container registry:

```
cloudctl case launch \
  --case ${OFFLINEDIR_CPFS}/ibm-cp-common-services-1.11.0.tgz \
  --inventory ibmCommonServiceOperatorSetup \
  --action mirror-images \
  --args "--fromRegistry ${PORTABLE_REGISTRY} --registry ${PRIVATE_REGISTRY} --user ${PRIVATE_REGISTRY_USER} --pass
${PRIVATE_REGISTRY_PASSWORD} --inputDir ${OFFLINEDIR_CPFS}"
```

11. Storing the CASE packages

Best practice: Save the CASE packages in a dedicated location to ensure that you can replicate the configuration at any time or on any cluster.
CASE packages specify dependencies as a range, so it is possible to get a later version of the dependencies if you download the CASE package on different dates.

Saving the downloaded CASE packages prevents any discrepancies.

What's next Now that you've mirrored the images to your private container registry, you are ready to complete Configuring your cluster to pull Cloud Pak for Data images.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see Documentation for previous 4.0.x refreshes.

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

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: When you change the global image pull secret, each node in the cluster is *automatically* restarted so that the Machine Config Operator can apply the changes. This restart process happens one node at a time. The cluster will wait for a node to restart before starting the process on the next node. In some situations, it takes more than 30 minutes for all of the nodes to be restarted. During this process, you might notice that resources are temporarily unavailable.
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

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=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 container registry
	password
	    The password for the specified user.
	registry-location
	    The location of the private container 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
	``

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 container registry
			password
			    The password for the specified user.
			
			b. Add an entry for the container 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 nodes
```

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
worker1                         Ready                     worker   5h48m    v1.20.0
worker2                         Ready                     worker   5h48m    v1.20.0
```

Tip: You can use the `watch oc get nodes` command to monitor the status of the nodes. The command provides an update every 2 seconds. When all of the nodes return `Ready` you can exit the command by pressing `Ctrl+C`.

Alternatively, if you find that the `oc get nodes` command returns `Ready` prematurely, you can use the `oc get mcp` command to get the real-time status of the nodes.

What's next Complete the appropriate task for your environment:

IBM Entitled Registry
    Now that you've updated the global pull secret to include your IBM entitlement API key, you're ready to complete [Creating catalog sources](#).
Private container registry
    Now that you've updated the global pull secret to include the credentials of an account that can pull images from the registry, you're ready to complete *2. Configuring an image content source policy*.

2. Configuring an image content source policy

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 container 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`
	`  - mirrors:`
	`    - ${PRIVATE_REGISTRY}/db2u`
	`    source: icr.io/db2u`
	`EOF`

Options	Image content source policy

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
	  - mirrors:
	    - ${PRIVATE_REGISTRY}/db2u
	    source: icr.io/db2u
	EOF
	``

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

```
oc get imageContentSourcePolicy
```

4. Confirm that the output includes `cloud-pak-for-data-mirror`.
5. Get the status of the nodes:

```
oc get nodes
```

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
worker1         Ready                    worker   5h48m   v1.20.0
worker2         Ready                    worker   5h48m   v1.20.0
```

Tip: You can use the `watch oc get nodes` command to monitor the status of the nodes. The command provides an update every 2 seconds. When all of the nodes return `Ready` you can exit the command by pressing `Ctrl+C`.

Alternatively, if you find that the `oc get nodes` command returns `Ready` prematurely, you can use the `oc get mcp` command to get the real-time status of the nodes.

What's next Now that you've configured the image content source policy, you're ready to complete Creating catalog sources.

Previous topic: Mirroring images to your private container registry
Next topic: Creating catalog sources
Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see Documentation for previous 4.0.x refreshes.

Creating catalog sources

To ensure that your cluster uses the correct software images, you must create the appropriate catalog sources for your environment.

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.)

Choosing the appropriate type of catalog source for your environment

The catalog sources that you create depend on several factors. Answer the following questions to determine which type of catalog source is appropriate for your environment:

1. Where are you pulling software images from?

   Private container registry
   > Follow the guidance in Creating catalog sources for a private container registry.
   IBM® Entitled Registry
   > Go to question 2.

2. Do you want to ensure that a specific version of the software is installed on your cluster?

   Yes, I want to ensure that a specific version of the software is installed on my cluster (Recommended)

Follow the guidance in [Creating catalog sources that pull specific versions of images from the IBM Entitled Registry](#).
No, I want the latest version of the software automatically installed on my cluster
> Restriction: Allowing automatic updates is not recommended in production environments where predictability and stability are important. This option is recommended only for short-term installations, such as proof-of-concept deployments.
> Follow the guidance in [Creating catalog sources that automatically pull the latest images from the IBM Entitled Registry](#).

- **[Creating catalog sources for a private container registry](#)**
  If you are using a *private container registry*, you must create the catalog source for each service that you plan to install.
- **[Creating catalog sources that pull specific versions of images from the IBM Entitled Registry](#)**
  If you are using the IBM Entitled Registry, it is strongly recommended that you use the appropriate CASE packages to create the catalog source for each service that you plan to install. Creating the catalog source for each service enables you to ensure that a specific version of the software is installed on your cluster and that the software is not updated without your knowledge.
- **[Creating catalog sources that automatically pull the latest images from the IBM Entitled Registry](#)**
  If you pull images from the IBM Entitled Registry you can create catalog sources that automatically pull the latest images from the IBM Entitled Registry.

Previous topic: [Configuring your cluster to pull Cloud Pak for Data images](#)
Next topic: [Installing IBM Cloud Pak foundational services](#)
Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Creating catalog sources for a private container registry

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

Important: If you are using the IBM® Entitled Registry, review the guidance in [Creating catalog sources](#) to determine which method you should use to create the catalog sources for your environment.

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 must create the catalog sources for any of the software that you plan to install. To streamline the process, it is recommended that you identify all of the software, including services, that you plan to install before you complete this task. However, if you decide to install an additional service later, you can complete the specific steps required to create the catalog source for the service.

Before you begin

The steps in the **Procedure** section assume that you have the required software and CASE packages on your local file system.

Complete the appropriate steps to ensure that you are able to create the catalog source:

If this statement applies to you	You need to complete the following preparation task
I don't have Python 3 installed on my workstation and I plan to install one or more 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) • Watson™ Knowledge Catalog	[Preparing your system - Installing Python](#)
I don't have the IBM Cloud Pak® CLI (cloudctl) installed on my workstation.	[Preparing your system - Installing IBM Cloud Pak CLI (cloudctl)](#)
I don't have the CASE packages on my local file system, but I can access packages that were already downloaded.	[Preparing your system - Copying the CASE packages](#)
I don't have the CASE packages on my local file system, and I don't have access to packages that were already downloaded.	[Preparing your system - Downloading the CASE packages](#)

If you have the IBM Cloud Pak CLI (cloudctl) and the CASE packages, go to the [Procedure](#) section.

Preparing your system - Installing Python

If you plan to install any of the following services, you must install Python 3 on your workstation:

- Data Virtualization
- Db2
- Db2 Big SQL
- Db2 Warehouse
- OpenPages (required only if you want OpenPages to automatically provision a Db2 database)
- Watson Knowledge Catalog

1. Install the following Python software on the system where you are running the `cloudctl` commands:
   a. Python 3
      To install Python 3, run the following command:

      ```
      yum install -y python3
      alternatives --set python /usr/bin/python3
      ```

b. **PyYAML**
To install **PyYAML**, run the following command:

```
pip3 install PyYAML
```

c. **argparse**
To install **argparse**, run the following command:

```
pip3 install argparse
```

What's next Now that you've installed the Python, you're ready to complete Preparing your system - Installing IBM Cloud Pak CLI (cloudctl).

Preparing your system - Installing IBM Cloud Pak CLI (cloudctl)

The IBM Cloud Pak CLI (cloudctl) enables you to create catalog sources from CASE packages.

If you don't have the IBM Cloud Pak CLI, you must install it on the machine from which you will run the commands to create the catalog sources.

To install the IBM Cloud Pak CLI

1. 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
   ```

2. Extract the contents of the archive file:

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

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

   ```
   chmod 775 cloudctl-architecture
   ```

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

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

5. 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.
What's next Now that you've installed the IBM Cloud Pak CLI, you're ready to obtain the CASE packages. Complete the appropriate task for your environment:

- Preparing your system - Copying the CASE packages
- Preparing your system - Downloading the CASE packages

Preparing your system - Copying the CASE packages

You must use the CASE packages to create the required catalog sources.

If the user who mirrored the images to the private container registry saved the CASE packages to a dedicated location, copy the packages to the machine from which you will run the commands to create the catalog sources.

After you copy the CASE packages to your local file system, you must set the following environment variables so that you can run the commands as written:

1. Set the **OFFLINEDIR** environment variable to the directory where the Cloud Pak for Data CASE packages are stored.
   To set this environment variable, run:

   ```
   export OFFLINEDIR=$HOME/offline/cpd
   ```

   Replace **$HOME/offline/cpd** with the appropriate value for your environment.

2. If IBM Cloud Pak foundational services is not installed, set the **OFFLINEDIR_CPFS** environment variable to the directory where the IBM Cloud Pak foundational services CASE packages are stored.
   To set this environment variable, run:

   ```
   export OFFLINEDIR_CPFS=$HOME/offline/cpfs
   ```

   Replace **$HOME/offline/cpfs** with the appropriate value for your environment.

What's next Now that you've copied the CASE packages, you're ready to create the catalog sources. Follow the steps in the Procedure section.

Preparing your system - Downloading the CASE packages

You must use the CASE packages to create the required catalog sources.

If the user who mirrored the images to the private container registry did not save the CASE packages to a dedicated location, you must download the CASE packages to the machine from which you will run the commands to create the catalog sources.

To download the CASE packages:

1. Set up your environment to download CASE packages:
   a. Identify or create the directory or directories 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.

Recommended directory name	Notes
$HOME/offline/cpd	Primary directory for Cloud Pak for Data software. This directory is required for all installations. To create this directory, run the following command: `mkdir -p $HOME/offline/cpd`
$HOME/offline/cpfs	Directory for IBM Cloud Pak foundational services software. This directory is recommended in the following situations: • You plan to install multiple IBM Cloud Paks. • IBM Cloud Pak foundational services is not installed on your cluster. To create this directory, run the following command: `mkdir -p $HOME/offline/cpfs`

Best practice: If you have multiple IBM Cloud Paks on your cluster, use a separate directory for the Cloud Pak for Data CASE packages (the packages for the Cloud Pak for Data platform and services).

When you run the command to mirror the images, the command mirrors any updated images in the directory. Using a separate directory ensures that you mirror only the images for Cloud Pak for Data.

b. Set the following environment variables:

Variable	Notes
`CASE_REPO_PATH`	Required for all installations. This environment variable points to the repository where the CASE packages are hosted. To set this environment variable, run: `export CASE_REPO_PATH=https://github.com/IBM/cloud-pak/raw/master/repo/case`
`OFFLINEDIR`	Required for all installations. This is the directory that you created to store the Cloud Pak for Data CASE packages. To set this environment variable, run: `export OFFLINEDIR=$HOME/offline/cpd` Replace `$HOME/offline/cpd` with the appropriate value for your environment.
`OFFLINEDIR_CPFS`	Required only if you created a separate directory for the IBM Cloud Pak foundational services CASE package. This is the directory that you created to store the IBM Cloud Pak foundational services CASE package. To set this environment variable, run: `export OFFLINEDIR_CPFS=$HOME/offline/cpfs` Replace `$HOME/offline/cpfs` with the appropriate value for your environment.

2. Download the IBM Cloud Pak for Data platform operator CASE package:

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cp-datacore \
--version 2.0.10 \
--outputdir ${OFFLINEDIR} \
--no-dependency
```

3. If IBM Cloud Pak foundational services is not installed on the cluster, download the IBM Cloud Pak foundational services CASE package:

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cp-common-services \
--version 1.11.0 \
--outputdir ${OFFLINEDIR_CPFS}
```

4. If you plan to install the scheduling service, download the scheduling service CASE package:

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cpd-scheduling \
--version 1.3.2 \
--outputdir ${OFFLINEDIR}
```

5. Download the CASE package for each service that you plan to install on your cluster:
   - ›
     Anaconda Repository for IBM Cloud Pak for Data

     Not applicable. For details, see Installing Anaconda Repository for IBM Cloud Pak for Data.

   - ›
     Analytics Engine Powered by Apache Spark

     ```
     cloudctl case save \
     --repo ${CASE_REPO_PATH} \
     --case ibm-analyticsengine \
     --version 4.0.6 \
     --outputdir ${OFFLINEDIR}
     ```

   - ›
     Cognos Analytics

     ```
     cloudctl case save \
     --repo ${CASE_REPO_PATH} \
     ```

```
--case ibm-cognos-analytics-prod \
--version 4.0.8 \
--outputdir ${OFFLINEDIR}
```

- ❯ Cognos Dashboards

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cde \
--version 2.0.6 \
--outputdir ${OFFLINEDIR}
```

- ❯ Data Privacy

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-dp \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- ❯ Data Refinery

The Data Refinery CASE package is automatically downloaded when you download the CASE package for either Watson Knowledge Catalog or Watson Studio.

- ❯ Data Virtualization

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-dv-case \
--version 1.7.6 \
--outputdir ${OFFLINEDIR}
```

- ❯ DataStage

Download the appropriate package based on your license:

DataStage Enterprise

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-datastage-enterprise \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

DataStage Enterprise Plus

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-datastage \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

- ❯ Db2

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-db2oltp \
--version 4.0.8 \
--outputdir ${OFFLINEDIR}
```

- ❯ Db2 Big SQL

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-bigsql-case \
--version 7.2.6 \
--outputdir ${OFFLINEDIR}
```

- ❯ Db2 Data Gate

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-datagate-prod \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- ❯ Db2 Data Management Console

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-dmc \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- > Db2 Event Store

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

- > Db2 Warehouse

  ```
  cloudctl case save \
  --repo ${CASE_REPO_PATH} \
  --case ibm-db2wh \
  --version 4.0.8 \
  --outputdir ${OFFLINEDIR}
  ```

- > Decision Optimization

  ```
  cloudctl case save \
  --repo ${CASE_REPO_PATH} \
  --case ibm-dods \
  --version 4.0.6 \
  --outputdir ${OFFLINEDIR}
  ```

- > EDB Postgres

  ```
  cloudctl case save \
  --repo ${CASE_REPO_PATH} \
  --case ibm-cpd-edb \
  --version 4.0.6 \
  --outputdir ${OFFLINEDIR}
  ```

- > Execution Engine for Apache Hadoop

  ```
  cloudctl case save \
  --repo ${CASE_REPO_PATH} \
  --case ibm-hadoop \
  --version 4.0.7 \
  --outputdir ${OFFLINEDIR}
  ```

- > Financial Services Workbench

  Not applicable. For details, see the [Financial Services Workbench documentation](#).

- > IBM Match 360 with Watson

  ```
  cloudctl case save \
  --repo ${CASE_REPO_PATH} \
  --case ibm-mdm \
  --version 1.0.182 \
  --outputdir ${OFFLINEDIR}
  ```

- > Informix

  Download both of the following CASE packages:

  - ```
    cloudctl case save \
    --repo ${CASE_REPO_PATH} \
    --case ibm-informix-operator \
    --version 4.0.5 \
    --outputdir ${OFFLINEDIR}
    ```

  - ```
    cloudctl case save \
    --repo ${CASE_REPO_PATH} \
    --case ibm-informix-cp4d-operator \
    --version 4.0.5 \
    --outputdir ${OFFLINEDIR}
    ```

- > MongoDB

  ```
  cloudctl case save \
  --repo ${CASE_REPO_PATH} \
  --case ibm-cpd-mongodb \
  --version 4.0.4 \
  --outputdir ${OFFLINEDIR}
  ```

- > OpenPages

  ```
  cloudctl case save \
  --repo ${CASE_REPO_PATH} \
  --case ibm-openpages \
  --version 2.1.3+20220205.003421.82040223 \
  --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 \
--repo ${CASE_REPO_PATH} \
--case ibm-db2aaservice \
--version 4.0.8 \
--outputdir ${OFFLINEDIR}
```

- > **Planning Analytics**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-planning-analytics \
--version 4.0.60476 \
--outputdir ${OFFLINEDIR}
```

- > **Product Master**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-productmaster \
--version 1.0.3 \
--outputdir ${OFFLINEDIR}
```

- > **RStudio Server with R 3.6**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-rstudio \
--version 1.0.6 \
--outputdir ${OFFLINEDIR}
```

- > **SPSS Modeler**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-spss \
--version 1.0.6 \
--outputdir ${OFFLINEDIR}
```

- > **Virtual Data Pipeline**

  Not applicable. For details, see [Installing Virtual Data Pipeline for Cloud Pak for Data](#).

- > **Voice Gateway**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-voice-gateway \
--version 1.0.5 \
--outputdir ${OFFLINEDIR}
```

- > **Watson Assistant**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-assistant \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- > **Watson Assistant for Voice Interaction**

  Watson Assistant for Voice Interaction is comprised of the following services:
  - Voice Gateway
  - Watson Assistant
  - Watson Speech to Text
  - Watson Text to Speech

  Download the appropriate services for your use case.

- > **Watson Discovery**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-discovery \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- > **Watson Knowledge Catalog**

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-wkc \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- > Watson Knowledge Studio

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-ks \
--version 4.0.5 \
--outputdir ${OFFLINEDIR}
```

- > Watson Machine Learning

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-wml-cpd \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

- > Watson Machine Learning Accelerator

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-wml-accelerator \
--version 2.3.6 \
--outputdir ${OFFLINEDIR}
```

- > Watson OpenScale

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-openscale \
--version 2.5.1 \
--outputdir ${OFFLINEDIR}
```

- > Watson Speech to Text

  The same package is used for the Watson Speech to Text service and the Watson Text to Speech service. You only need to download the package once.

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-speech \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

- > Watson Studio

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-wsl \
--version 2.0.6 \
--outputdir ${OFFLINEDIR}
```

- > Watson Studio Runtimes

  The Watson Studio Runtimes CASE package is automatically downloaded when you download the CASE package for Watson Studio.

- > Watson Text to Speech

  The same package is used for the Watson Speech to Text service and the Watson Text to Speech service. You only need to download the package once.

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-speech \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

What's next Now that you've downloaded the CASE packages, you're ready to create the catalog source. Follow the steps in the Procedure section.

Procedure

To create the catalog sources:

1. Create the IBM Cloud Pak foundational services catalog source.
   Skip this step if a supported version of IBM Cloud Pak foundational services is already installed on your cluster.

   a. Run the following command to create the IBM Cloud Pak foundational services catalog source for the minimum supported version:

```
cloudctl case launch \
  --case ${OFFLINEDIR_CPFS}/ibm-cp-common-services-1.11.0.tgz \
  --inventory ibmCommonServiceOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--registry ${PRIVATE_REGISTRY} --inputDir ${OFFLINEDIR_CPFS} --recursive"
```

   b. Verify that **opencloud-operators** is **READY**:

```
oc get catalogsource -n openshift-marketplace opencloud-operators \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

2. Create the scheduling service catalog source.
   Skip this step if you are not installing the scheduling service.

   a. Run the following command to create the scheduling service catalog source for the latest refresh:

   ```
   cloudctl case launch \
     --case ${OFFLINEDIR}/ibm-cpd-scheduling-1.3.2.tgz \
     --inventory schedulerSetup \
     --namespace openshift-marketplace \
     --action install-catalog \
       --args "--inputDir ${OFFLINEDIR} --recursive"
   ```

   b. Verify that `ibm-cpd-scheduling-catalog` is **READY**:

   ```
   oc get catalogsource -n openshift-marketplace ibm-cpd-scheduling-catalog \
   -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
   ```

   It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

3. Create the IBM Cloud Pak for Data catalog source:
   a. Run the following command to create the IBM Cloud Pak for Data catalog source for the latest refresh:

   ```
   cloudctl case launch \
     --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.10.tgz \
     --inventory cpdPlatformOperator \
     --namespace openshift-marketplace \
     --action install-catalog \
       --args "--inputDir ${OFFLINEDIR} --recursive"
   ```

   b. Verify that `cpd-platform` is **READY**:

   ```
   oc get catalogsource -n openshift-marketplace cpd-platform \
   -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
   ```

   It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

4. 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)
   a. Run the following command to create the Db2U catalog source for the latest refresh:

   ```
   cloudctl case launch \
     --case ${OFFLINEDIR}/ibm-db2uoperator-4.0.9.tgz \
     --inventory db2uOperatorSetup \
     --namespace openshift-marketplace \
     --action install-catalog \
       --args "--inputDir ${OFFLINEDIR} --recursive"
   ```

   b. Verify that `ibm-db2uoperator-catalog` is **READY**:

   ```
   oc get catalogsource -n openshift-marketplace ibm-db2uoperator-catalog \
   -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
   ```

   It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

5. Create the catalog source for *each* service that you mirrored to the private container registry.
   - ⟩ Anaconda Repository for IBM Cloud Pak for Data

     Not applicable. For details, see Installing Anaconda Repository for IBM Cloud Pak for Data.

   - ⟩ Analytics Engine Powered by Apache Spark
     - Run the following command to create the Analytics Engine Powered by Apache Spark catalog source for the latest refresh:

     ```
     cloudctl case launch \
       --case ${OFFLINEDIR}/ibm-analyticsengine-4.0.6.tgz \
       --inventory analyticsengineOperatorSetup \
       --namespace openshift-marketplace \
       --action install-catalog \
         --args "--inputDir ${OFFLINEDIR} --recursive"
     ```

     - Verify that `ibm-cpd-ae-operator-catalog` is **READY**:

     ```
     oc get catalogsource -n openshift-marketplace ibm-cpd-ae-operator-catalog \
     -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
     ```

     It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

   - ⟩ Cognos Analytics
     - Run the following command to create the Cognos Analytics catalog source for the latest refresh:

     ```
     cloudctl case launch \
       --case ${OFFLINEDIR}/ibm-cognos-analytics-prod-4.0.8.tgz \
       --inventory ibmCaOperatorSetup \
       --namespace openshift-marketplace \
     ```

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

- Verify that **ibm-ca-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-ca-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- <button>Cognos Dashboards ›</button>
    - Run the following command to create the Cognos Dashboards catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cde-2.0.6.tgz \
  --inventory cdeOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
     --args "--inputDir ${OFFLINEDIR} --recursive"
```

    - Verify that **ibm-cde-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cde-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- <button>Data Privacy ›</button>
    - Run the following command to create the Data Privacy catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-dp-4.0.6.tgz \
  --inventory dpOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
     --args "--inputDir ${OFFLINEDIR} --recursive"
```

    - Verify that **ibm-cpd-dp-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cpd-dp-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- <button>Data Refinery ›</button>
The catalog source for Data Refinery is automatically created when you create the catalog source for either Watson Knowledge Catalog or Watson Studio.

- <button>Data Virtualization ›</button>
    - This service has a dependency on Db2U. Verify that the Db2U catalog source (**ibm-db2uoperator-catalog**) is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-db2uoperator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

    - Run the following command to create the Data Virtualization catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-dv-case-1.7.6.tgz \
  --inventory dv \
  --namespace openshift-marketplace \
  --action install-catalog \
     --args "--inputDir ${OFFLINEDIR} --recursive"
```

    - Verify that **ibm-dv-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-dv-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- <button>DataStage ›</button>
Create the appropriate catalog source for your environment:

DataStage Enterprise
    - Run the following command to create the DataStage Enterprise catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-datastage-enterprise-4.0.7.tgz \
  --inventory dsOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
     --args "--inputDir ${OFFLINEDIR} --recursive"
```

    - Verify that **ibm-cpd-datastage-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cpd-datastage-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

DataStage Enterprise Plus
- Run the following command to create the DataStage Enterprise Plus catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-datastage-4.0.7.tgz \
  --inventory dsOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

- Verify that **ibm-cpd-datastage-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cpd-datastage-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > Db2

  - This service has a dependency on Db2U. Verify that the Db2U catalog source (**ibm-db2uoperator-catalog**) is **READY**:

  ```
  oc get catalogsource -n openshift-marketplace ibm-db2uoperator-catalog \
  -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
  ```

  - Run the following command to create the Db2 catalog source for the latest refresh:

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-db2oltp-4.0.8.tgz \
    --inventory db2oltpOperatorSetup \
    --namespace openshift-marketplace \
    --action install-catalog \
      --args "--inputDir ${OFFLINEDIR} --recursive"
  ```

  - Verify that **ibm-db2oltp-cp4d-operator-catalog** is **READY**:

  ```
  oc get catalogsource -n openshift-marketplace ibm-db2oltp-cp4d-operator-catalog \
  -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
  ```

  It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > Db2 Big SQL

  - This service has a dependency on Db2U. Verify that the Db2U catalog source (**ibm-db2uoperator-catalog**) is **READY**:

  ```
  oc get catalogsource -n openshift-marketplace ibm-db2uoperator-catalog \
  -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
  ```

  - Run the following command to create the Db2 Big SQL catalog source for the latest refresh:

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-bigsql-case-7.2.6.tgz \
    --inventory bigsql \
    --namespace openshift-marketplace \
    --action install-catalog \
      --args "--inputDir ${OFFLINEDIR} --recursive"
  ```

  - Verify that **ibm-bigsql-operator-catalog** is **READY**:

  ```
  oc get catalogsource -n openshift-marketplace ibm-bigsql-operator-catalog \
  -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
  ```

  It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > Db2 Data Gate

  - Run the following command to create the Db2 Data Gate catalog source for the latest refresh:

  ```
  cloudctl case launch \
    --case ${OFFLINEDIR}/ibm-datagate-prod-4.0.6.tgz \
    --inventory datagateOperatorSetup \
    --namespace openshift-marketplace \
    --action install-catalog \
      --args "--inputDir ${OFFLINEDIR} --recursive"
  ```

  - Verify that **ibm-datagate-operator-catalog** is **READY**:

  ```
  oc get catalogsource -n openshift-marketplace ibm-datagate-operator-catalog \
  -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
  ```

  It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- › **Db2 Data Management Console**
  - Run the following command to create the Db2 Data Management Console catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-dmc-4.0.6.tgz \
      --inventory dmcOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-dmc-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-dmc-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- › **Db2 Event Store**

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

- › **Db2 Warehouse**
  - This service has a dependency on Db2U. Verify that the Db2U catalog source (**ibm-db2uoperator-catalog**) is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-db2uoperator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

  - Run the following command to create the Db2 Warehouse catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-db2wh-4.0.8.tgz \
      --inventory db2whOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-db2wh-cp4d-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-db2wh-cp4d-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- › **Decision Optimization**
  - Run the following command to create the Decision Optimization catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-dods-4.0.6.tgz \
      --inventory dodsOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-cpd-dods-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-cpd-dods-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- › **EDB Postgres**
  - Run the following command to create the EDB Postgres catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-cpd-edb-4.0.6.tgz \
      --inventory ibmCPDEDBSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-cpd-edb-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-cpd-edb-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- › **Execution Engine for Apache Hadoop**
  - Run the following command to create the Execution Engine for Apache Hadoop catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-hadoop-4.0.7.tgz \
      --inventory hadoopSetup \
    ```

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

- Verify that **ibm-cpd-hadoop-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cpd-hadoop-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

  It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- ❯ Financial Services Workbench

  Not applicable. For details, see the [Financial Services Workbench documentation](#).

- ❯ IBM Match 360 with Watson

  - Run the following command to create the IBM Match 360 with Watson catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-mdm-1.0.182.tgz \
  --inventory mdmOperator \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-mdm-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-mdm-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- ❯ Informix

  - Run the following command to create the catalog source for the Informix install operator:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-informix-operator-4.0.5.tgz \
  --inventory ibmInformixOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that the **ibm-informix-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-informix-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

  - Run the following command to create the catalog source for the Informix deployment operator:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-informix-cp4d-operator-4.0.5.tgz \
  --inventory ibmInformixCp4dOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-informix-cp4d-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-informix-cp4d-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- ❯ MongoDB

  - Run the following command to create the MongoDB catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cpd-mongodb-4.0.4.tgz \
  --inventory ibmCPDMongodbSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-cpd-mongodb-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cpd-mongodb-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- ❯ OpenPages

- Run the following command to create the OpenPages catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-openpages-2.1.3+20220205.003421.82040223.tgz \
  --inventory operatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

- Verify that **ibm-cpd-openpages-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cpd-openpages-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

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

Db2U

Verify that the Db2U catalog source (**ibm-db2uoperator-catalog**) is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-db2uoperator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

Db2 as a service
- Run the following command to create the Db2 as a service catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-db2aaservice-4.0.8.tgz \
  --inventory db2aaserviceOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

- Verify that **ibm-db2aaservice-cp4d-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-db2aaservice-cp4d-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- >
  Planning Analytics
  - Run the following command to create the Planning Analytics catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-planning-analytics-4.0.60476.tgz \
  --inventory ibmPlanningAnalyticsOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-planning-analytics-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-planning-analytics-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- >
  Product Master
  - Run the following command to create the Product Master catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-productmaster-1.0.3.tgz \
  --inventory productmasterOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-productmaster-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-productmaster-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- >
  RStudio Server with R 3.6
  - Run the following command to create the RStudio Server with R 3.6 catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-rstudio-1.0.6.tgz \
  --inventory rstudioSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

- Verify that **ibm-cpd-rstudio-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cpd-rstudio-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- >
  SPSS Modeler
  - Run the following command to create the SPSS Modeler catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-spss-1.0.6.tgz \
  --inventory spssSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-cpd-spss-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cpd-spss-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- >
  Virtual Data Pipeline

  Not applicable. For details, see .

- >
  Voice Gateway
  - Run the following command to create the Voice Gateway catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-voice-gateway-1.0.5.tgz \
  --inventory voiceGatewayOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-voice-gateway-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-voice-gateway-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- >
  Watson Assistant
  - Run the following command to create the Watson Assistant catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-watson-assistant-4.0.6.tgz \
  --inventory assistantOperator \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-watson-assistant-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-watson-assistant-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- >
  Watson Assistant for Voice Interaction

  Watson Assistant for Voice Interaction is comprised of the following services:
  - Voice Gateway
  - Watson Assistant
  - Watson Speech to Text
  - Watson Text to Speech

- >
  Watson Discovery
  - Run the following command to create the Watson Discovery catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-watson-discovery-4.0.6.tgz \
  --inventory discoveryOperators \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-watson-discovery-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-watson-discovery-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- › Watson Knowledge Catalog
  - Run the following command to create the Watson Knowledge Catalog catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-wkc-4.0.6.tgz \
      --inventory wkcOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-cpd-wkc-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-cpd-wkc-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- › Watson Knowledge Studio
  - Run the following command to create the Watson Knowledge Studio catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-watson-ks-4.0.5.tgz \
      --inventory wksOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-watson-ks-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-watson-ks-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- › Watson Machine Learning
  - Run the following command to create the Watson Machine Learning catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-wml-cpd-4.0.7.tgz \
      --inventory wmlOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-cpd-wml-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-cpd-wml-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- › Watson Machine Learning Accelerator
  - Run the following command to create the Watson Machine Learning Accelerator catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-wml-accelerator-2.3.6.tgz \
      --inventory wmla_operator_deploy \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-cpd-wml-accelerator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-cpd-wml-accelerator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- › Watson OpenScale
  - Run the following command to create the Watson OpenScale catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-watson-openscale-2.5.1.tgz \
      --inventory ibmWatsonOpenscaleOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-openscale-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-openscale-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- >
  Watson Speech to Text

  The same operator is used for the Watson Speech to Text service and the Watson Text to Speech service. You only need to create the catalog source once.

  - Run the following command to create the Watson Speech to Text catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-watson-speech-4.0.7.tgz \
      --inventory speechOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that `ibm-watson-speech-operator-catalog` is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-watson-speech-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- >
  Watson Studio

  - Run the following command to create the Watson Studio catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-wsl-2.0.6.tgz \
      --inventory wslSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that `ibm-cpd-ws-operator-catalog` is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-cpd-ws-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- >
  Watson Studio Runtimes

  The catalog source for Watson Studio Runtimes is automatically created when you create the catalog source for Watson Studio.

  After you create the catalog source for Watson Studio, verify that the Watson Studio Runtimes catalog source was also created.

  - Verify that `ibm-cpd-ws-runtimes-operator-catalog` is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-cpd-ws-runtimes-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- >
  Watson Text to Speech

  The same operator is used for the Watson Speech to Text service and the Watson Text to Speech service. You only need to create the catalog source once.

  - Run the following command to create the Watson Speech to Text catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-watson-speech-4.0.7.tgz \
      --inventory speechOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that `ibm-watson-speech-operator-catalog` is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-watson-speech-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

What's next Now that you've created the required catalog sources for your environment, you are ready to complete [Installing IBM Cloud Pak foundational services](#).

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Creating catalog sources that pull specific versions of images from the IBM Entitled Registry

If you are using the IBM® Entitled Registry, it is strongly recommended that you use the appropriate CASE packages to create the catalog source for each service that you plan to install. Creating the catalog source for each service enables you to ensure that a specific version of the software is installed on your cluster and that the software is not updated without your knowledge.

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 must create the catalog sources for any of the software that you plan to install. To streamline the process, it is recommended that you identify all of the software, including services, that you plan to install before you complete this task. However, if you decide to install an additional service later, you can complete the specific steps required to create the catalog source for the service.

Before you begin

The steps in the **Procedure** section assume that you have the required software and CASE packages on your local file system.

Complete the following preparation task to ensure that you are able to create the catalog source:

1. Preparing your system - Installing Python
2. Preparing your system - Installing IBM Cloud Pak CLI (cloudctl)
3. Preparing your system - Downloading the CASE packages

Preparing your system - Installing Python

If you plan to install any of the following services, you must install Python 3 on your workstation:

- Data Virtualization
- Db2®
- Db2 Big SQL
- Db2 Warehouse
- OpenPages® (required only if you want OpenPages to automatically provision a Db2 database)
- Watson™ Knowledge Catalog

1. Install the following Python software on the system where you are running the `cloudctl` commands:
   a. Python 3
      To install Python 3, run the following command:

      ```
      yum install -y python3
      alternatives --set python /usr/bin/python3
      ```

   b. **PyYAML**
      To install **PyYAML**, run the following command:

      ```
      pip3 install PyYAML
      ```

   c. **argparse**
      To install **argparse**, run the following command:

      ```
      pip3 install argparse
      ```

What's next Now that you've installed the Python, you're ready to complete Preparing your system - Installing IBM Cloud Pak CLI (cloudctl).

Preparing your system - Installing IBM Cloud Pak CLI (cloudctl)

The IBM Cloud Pak® CLI (cloudctl) enables you to create catalog sources from CASE packages.

If you don't have the IBM Cloud Pak CLI, you must install it on the machine from which you will run the commands to create the catalog sources.

To install the IBM Cloud Pak CLI:

1. 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
   ```

2. Extract the contents of the archive file:

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

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

   ```
   chmod 775 cloudctl-architecture
   ```

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

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

5. 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.
What's next Now that you've installed the IBM Cloud Pak CLI, you're ready to complete Preparing your system - Downloading the CASE packages.

Preparing your system - Downloading the CASE packages

You must download the CASE packages to the machine from which you will run the commands to create the catalog sources.

To download the CASE packages:

1. Set up your environment to download CASE packages:
   a. Identify or create the directory or directories 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.

Recommended directory name	Notes
$HOME/offline/cpd	Primary directory for Cloud Pak for Data software. This directory is required for all installations. To create this directory, run the following command: `mkdir -p $HOME/offline/cpd`
$HOME/offline/cpfs	Directory for IBM Cloud Pak foundational services software. This directory is recommended in the following situations: • You plan to install multiple IBM Cloud Paks. • IBM Cloud Pak foundational services is not installed on your cluster. To create this directory, run the following command: `mkdir -p $HOME/offline/cpfs`

   Best practice: If you have multiple IBM Cloud Paks on your cluster, use a separate directory for the Cloud Pak for Data CASE packages (the packages for the Cloud Pak for Data platform and services).
   When you run the command to mirror the images, the command mirrors any updated images in the directory. Using a separate directory ensures that you mirror only the images for Cloud Pak for Data.

   b. Set the following environment variables:

Variable	Notes
`CASE_REPO_PATH`	Required for all installations. This environment variable points to the repository where the CASE packages are hosted. To set this environment variable, run: `export CASE_REPO_PATH=https://github.com/IBM/cloud-pak/raw/master/repo/case`
`OFFLINEDIR`	Required for all installations. This is the directory that you created to store the Cloud Pak for Data CASE packages. To set this environment variable, run: `export OFFLINEDIR=$HOME/offline/cpd` Replace `$HOME/offline/cpd` with the appropriate value for your environment.
`OFFLINEDIR_CPFS`	Required only if you created a separate directory for the IBM Cloud Pak foundational services CASE package. This is the directory that you created to store the IBM Cloud Pak foundational services CASE package. To set this environment variable, run: `export OFFLINEDIR_CPFS=$HOME/offline/cpfs` Replace `$HOME/offline/cpfs` with the appropriate value for your environment.

2. Download the IBM Cloud Pak for Data platform operator CASE package:

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cp-datacore \
--version 2.0.10 \
--outputdir ${OFFLINEDIR} \
--no-dependency
```

3. If IBM Cloud Pak foundational services is not installed on the cluster, download the IBM Cloud Pak foundational services CASE package:

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cp-common-services \
--version 1.11.0 \
--outputdir ${OFFLINEDIR_CPFS}
```

4. If you plan to install the scheduling service, download the scheduling service CASE package:

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cpd-scheduling \
--version 1.3.2 \
--outputdir ${OFFLINEDIR}
```

5. Download the CASE package for each service that you plan to install on your cluster:

- > Anaconda Repository for IBM Cloud Pak for Data

  Not applicable. For details, see [Installing Anaconda Repository for IBM Cloud Pak for Data](#).

- > Analytics Engine Powered by Apache Spark

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-analyticsengine \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- > Cognos Analytics

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cognos-analytics-prod \
--version 4.0.8 \
--outputdir ${OFFLINEDIR}
```

- > Cognos Dashboards

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cde \
--version 2.0.6 \
--outputdir ${OFFLINEDIR}
```

- > Data Privacy

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-dp \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- > Data Refinery

  The Data Refinery CASE package is automatically downloaded when you download the CASE package for either Watson Knowledge Catalog or Watson Studio.

- > Data Virtualization

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-dv-case \
--version 1.7.6 \
--outputdir ${OFFLINEDIR}
```

- > DataStage

  Download the appropriate package based on your license:

  DataStage Enterprise

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-datastage-enterprise \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

  DataStage Enterprise Plus

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-datastage \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

- > Db2

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-db2oltp \
--version 4.0.8 \
--outputdir ${OFFLINEDIR}
```

- > Db2 Big SQL

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-bigsql-case \
--version 7.2.6 \
--outputdir ${OFFLINEDIR}
```

- >
  Db2 Data Gate

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-datagate-prod \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- >
  Db2 Data Management Console

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-dmc \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- >
  Db2 Event Store

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

- >
  Db2 Warehouse

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-db2wh \
--version 4.0.8 \
--outputdir ${OFFLINEDIR}
```

- >
  Decision Optimization

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-dods \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- >
  EDB Postgres

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-cpd-edb \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- >
  Execution Engine for Apache Hadoop

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-hadoop \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

- >
  Financial Services Workbench

  Not applicable. For details, see the [Financial Services Workbench documentation](#).

- >
  IBM Match 360 with Watson

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-mdm \
--version 1.0.182 \
--outputdir ${OFFLINEDIR}
```

- >
  Informix

  Download both of the following CASE packages:

  - ```
    cloudctl case save \
    --repo ${CASE_REPO_PATH} \
    --case ibm-informix-operator \
    --version 4.0.5 \
    --outputdir ${OFFLINEDIR}
    ```

  - ```
    cloudctl case save \
    --repo ${CASE_REPO_PATH} \
    --case ibm-informix-cp4d-operator \
    --version 4.0.5 \
    --outputdir ${OFFLINEDIR}
    ```

- >
  MongoDB

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
```

```
--case ibm-cpd-mongodb \
--version 4.0.4 \
--outputdir ${OFFLINEDIR}
```

- › OpenPages

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-openpages \
--version 2.1.3+20220205.003421.82040223 \
--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 \
--repo ${CASE_REPO_PATH} \
--case ibm-db2aaservice \
--version 4.0.8 \
--outputdir ${OFFLINEDIR}
```

- › Planning Analytics

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-planning-analytics \
--version 4.0.60476 \
--outputdir ${OFFLINEDIR}
```

- › Product Master

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-productmaster \
--version 1.0.3 \
--outputdir ${OFFLINEDIR}
```

- › RStudio Server with R 3.6

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-rstudio \
--version 1.0.6 \
--outputdir ${OFFLINEDIR}
```

- › SPSS Modeler

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-spss \
--version 1.0.6 \
--outputdir ${OFFLINEDIR}
```

- › Virtual Data Pipeline

Not applicable. For details, see [Installing Virtual Data Pipeline for Cloud Pak for Data](#).

- › Voice Gateway

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-voice-gateway \
--version 1.0.5 \
--outputdir ${OFFLINEDIR}
```

- › Watson Assistant

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-assistant \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- › Watson Assistant for Voice Interaction

Watson Assistant for Voice Interaction is comprised of the following services:
  - Voice Gateway
  - Watson Assistant
  - Watson Speech to Text
  - Watson Text to Speech

Download the appropriate services for your use case.

- >
  Watson Discovery

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-discovery \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- >
  Watson Knowledge Catalog

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-wkc \
--version 4.0.6 \
--outputdir ${OFFLINEDIR}
```

- >
  Watson Knowledge Studio

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-ks \
--version 4.0.5 \
--outputdir ${OFFLINEDIR}
```

- >
  Watson Machine Learning

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-wml-cpd \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

- >
  Watson Machine Learning Accelerator

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-wml-accelerator \
--version 2.3.6 \
--outputdir ${OFFLINEDIR}
```

- >
  Watson OpenScale

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-openscale \
--version 2.5.1 \
--outputdir ${OFFLINEDIR}
```

- >
  Watson Speech to Text

The same package is used for the Watson Speech to Text service and the Watson Text to Speech service. You only need to download the package once.

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-speech \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

- >
  Watson Studio

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-wsl \
--version 2.0.6 \
--outputdir ${OFFLINEDIR}
```

- >
  Watson Studio Runtimes

The Watson Studio Runtimes CASE package is automatically downloaded when you download the CASE package for Watson Studio.

- >
  Watson Text to Speech

The same package is used for the Watson Speech to Text service and the Watson Text to Speech service. You only need to download the package once.

```
cloudctl case save \
--repo ${CASE_REPO_PATH} \
--case ibm-watson-speech \
--version 4.0.7 \
--outputdir ${OFFLINEDIR}
```

What's next Now that you've downloaded the CASE packages, you're ready to create the catalog source. Follow the steps in the section.

Procedure

To create the catalog source:

1. Create the IBM Cloud Pak foundational services catalog source.
   Skip this step if both of the following statements are true:
   - A [supported version](#) IBM Cloud Pak foundational services is already installed on your cluster.
   - The **opencloud-operators** catalog source already exists on your cluster.
     To check whether the **opencloud-operators** catalog source exists, run the following command:

   ```
   oc get catalogsource -n openshift-marketplace opencloud-operators
   ```

   a. Run the following command to create the IBM Cloud Pak foundational services catalog source for the minimum supported version:

   ```
   cloudctl case launch \
      --case ${OFFLINEDIR_CPFS}/ibm-cp-common-services-1.11.0.tgz \
      --inventory ibmCommonServiceOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
         --args "--registry icr.io --inputDir ${OFFLINEDIR_CPFS} --recursive"
   ```

   b. Verify that **opencloud-operators** is **READY**:

   ```
   oc get catalogsource -n openshift-marketplace opencloud-operators \
   -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
   ```

   It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

2. Create the scheduling service catalog source.
   Skip this step if you are not installing the scheduling service.

   a. Run the following command to create the scheduling service catalog source for the latest refresh:

   ```
   cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-cpd-scheduling-1.3.2.tgz \
      --inventory schedulerSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
         --args "--inputDir ${OFFLINEDIR} --recursive"
   ```

   b. Verify that **ibm-cpd-scheduling-catalog** is **READY**:

   ```
   oc get catalogsource -n openshift-marketplace ibm-cpd-scheduling-catalog \
   -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
   ```

   It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

3. Create the IBM Cloud Pak for Data catalog source:
   a. Run the following command to create the IBM Cloud Pak for Data catalog source for the latest refresh:

   ```
   cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-cp-datacore-2.0.10.tgz \
      --inventory cpdPlatformOperator \
      --namespace openshift-marketplace \
      --action install-catalog \
         --args "--inputDir ${OFFLINEDIR} --recursive"
   ```

   b. Verify that **cpd-platform** is **READY**:

   ```
   oc get catalogsource -n openshift-marketplace cpd-platform \
   -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
   ```

   It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

4. 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)
   a. Run the following command to create the Db2U catalog source for the latest refresh:

   ```
   cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-db2uoperator-4.0.9.tgz \
      --inventory db2uOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
         --args "--inputDir ${OFFLINEDIR} --recursive"
   ```

   b. Verify that **ibm-db2uoperator-catalog** is **READY**:

   ```
   oc get catalogsource -n openshift-marketplace ibm-db2uoperator-catalog \
   -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
   ```

   It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

5. Create the catalog source for *each* service that you plan to install.
   - >
     Anaconda Repository for IBM Cloud Pak for Data

     Not applicable. For details, see [Installing Anaconda Repository for IBM Cloud Pak for Data](#).

   - >
     Analytics Engine Powered by Apache Spark

- Run the following command to create the Analytics Engine Powered by Apache Spark catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-analyticsengine-4.0.6.tgz \
  --inventory analyticsengineOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

- Verify that **ibm-cpd-ae-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cpd-ae-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > **Cognos Analytics**
  - Run the following command to create the Cognos Analytics catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cognos-analytics-prod-4.0.8.tgz \
  --inventory ibmCaOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-ca-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-ca-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > **Cognos Dashboards**
  - Run the following command to create the Cognos Dashboards catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-cde-2.0.6.tgz \
  --inventory cdeOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-cde-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cde-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > **Data Privacy**
  - Run the following command to create the Data Privacy catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-dp-4.0.6.tgz \
  --inventory dpOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-cpd-dp-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cpd-dp-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > **Data Refinery**
The catalog source for Data Refinery is automatically created when you create the catalog source for either Watson Knowledge Catalog or Watson Studio.

- > **Data Virtualization**
  - This service has a dependency on Db2U. Verify that the Db2U catalog source (**ibm-db2uoperator-catalog**) is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-db2uoperator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

  - Run the following command to create the Data Virtualization catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-dv-case-1.7.6.tgz \
  --inventory dv \
  --namespace openshift-marketplace \
```

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

- Verify that **ibm-dv-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-dv-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > DataStage

  Create the appropriate catalog source for your environment:

  DataStage Enterprise
  - Run the following command to create the DataStage Enterprise catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-datastage-enterprise-4.0.7.tgz \
      --inventory dsOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-cpd-datastage-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-cpd-datastage-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

  DataStage Enterprise Plus
  - Run the following command to create the DataStage Enterprise Plus catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-datastage-4.0.7.tgz \
      --inventory dsOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-cpd-datastage-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-cpd-datastage-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > Db2

  - This service has a dependency on Db2U. Verify that the Db2U catalog source (**ibm-db2uoperator-catalog**) is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-db2uoperator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

  - Run the following command to create the Db2 catalog source for the latest refresh:
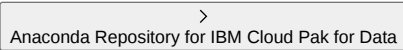
    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-db2oltp-4.0.8.tgz \
      --inventory db2oltpOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-db2oltp-cp4d-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-db2oltp-cp4d-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > Db2 Big SQL

  - This service has a dependency on Db2U. Verify that the Db2U catalog source (**ibm-db2uoperator-catalog**) is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-db2uoperator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

  - Run the following command to create the Db2 Big SQL catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-bigsql-case-7.2.6.tgz \
      --inventory bigsql \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-bigsql-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-bigsql-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- ›
  Db2 Data Gate
  - Run the following command to create the Db2 Data Gate catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-datagate-prod-4.0.6.tgz \
      --inventory datagateOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-datagate-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-datagate-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- ›
  Db2 Data Management Console
  - Run the following command to create the Db2 Data Management Console catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-dmc-4.0.6.tgz \
      --inventory dmcOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-dmc-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-dmc-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- ›
  Db2 Event Store

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

- ›
  Db2 Warehouse
  - This service has a dependency on Db2U. Verify that the Db2U catalog source (**ibm-db2uoperator-catalog**) is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-db2uoperator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

  - Run the following command to create the Db2 Warehouse catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-db2wh-4.0.8.tgz \
      --inventory db2whOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-db2wh-cp4d-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-db2wh-cp4d-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- ›
  Decision Optimization
  - Run the following command to create the Decision Optimization catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-dods-4.0.6.tgz \
      --inventory dodsOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-cpd-dods-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-cpd-dods-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- >
  EDB Postgres
    - Run the following command to create the EDB Postgres catalog source for the latest refresh:

      ```
      cloudctl case launch \
        --case ${OFFLINEDIR}/ibm-cpd-edb-4.0.6.tgz \
        --inventory ibmCPDEDBSetup \
        --namespace openshift-marketplace \
        --action install-catalog \
          --args "--inputDir ${OFFLINEDIR} --recursive"
      ```

    - Verify that **ibm-cpd-edb-operator-catalog** is **READY**:

      ```
      oc get catalogsource -n openshift-marketplace ibm-cpd-edb-operator-catalog \
      -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
      ```

      It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- >
  Execution Engine for Apache Hadoop
    - Run the following command to create the Execution Engine for Apache Hadoop catalog source for the latest refresh:

      ```
      cloudctl case launch \
        --case ${OFFLINEDIR}/ibm-hadoop-4.0.7.tgz \
        --inventory hadoopSetup \
        --namespace openshift-marketplace \
        --action install-catalog \
          --args "--inputDir ${OFFLINEDIR} --recursive"
      ```

    - Verify that **ibm-cpd-hadoop-operator-catalog** is **READY**:

      ```
      oc get catalogsource -n openshift-marketplace ibm-cpd-hadoop-operator-catalog \
      -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
      ```

      It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- >
  Financial Services Workbench

  Not applicable. For details, see the [Financial Services Workbench documentation](#).

- >
  IBM Match 360 with Watson
    - Run the following command to create the IBM Match 360 with Watson catalog source for the latest refresh:

      ```
      cloudctl case launch \
        --case ${OFFLINEDIR}/ibm-mdm-1.0.182.tgz \
        --inventory mdmOperator \
        --namespace openshift-marketplace \
        --action install-catalog \
          --args "--inputDir ${OFFLINEDIR} --recursive"
      ```

    - Verify that **ibm-mdm-operator-catalog** is **READY**:

      ```
      oc get catalogsource -n openshift-marketplace ibm-mdm-operator-catalog \
      -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
      ```

      It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- >
  Informix
    - Run the following command to create the catalog source for the Informix install operator:

      ```
      cloudctl case launch \
        --case ${OFFLINEDIR}/ibm-informix-operator-4.0.5.tgz \
        --inventory ibmInformixOperatorSetup \
        --namespace openshift-marketplace \
        --action install-catalog \
          --args "--inputDir ${OFFLINEDIR} --recursive"
      ```

    - Verify that the **ibm-informix-operator-catalog** is **READY**:

      ```
      oc get catalogsource -n openshift-marketplace ibm-informix-operator-catalog \
      -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
      ```

      It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

    - Run the following command to create the catalog source for the Informix deployment operator:

      ```
      cloudctl case launch \
        --case ${OFFLINEDIR}/ibm-informix-cp4d-operator-4.0.5.tgz \
        --inventory ibmInformixCp4dOperatorSetup \
        --namespace openshift-marketplace \
        --action install-catalog \
          --args "--inputDir ${OFFLINEDIR} --recursive"
      ```

    - Verify that **ibm-informix-cp4d-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-informix-cp4d-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > MongoDB

  - Run the following command to create the MongoDB catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-cpd-mongodb-4.0.4.tgz \
      --inventory ibmCPDMongodbSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-cpd-mongodb-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-cpd-mongodb-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > OpenPages

  - Run the following command to create the OpenPages catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-openpages-2.1.3+20220205.003421.82040223.tgz \
      --inventory operatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-cpd-openpages-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-cpd-openpages-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

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

  Db2U

  Verify that the Db2U catalog source (**ibm-db2uoperator-catalog**) is **READY**:

  ```
  oc get catalogsource -n openshift-marketplace ibm-db2uoperator-catalog \
  -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
  ```

  Db2 as a service

  - Run the following command to create the Db2 as a service catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-db2aaservice-4.0.8.tgz \
      --inventory db2aaserviceOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-db2aaservice-cp4d-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-db2aaservice-cp4d-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > Planning Analytics

  - Run the following command to create the Planning Analytics catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-planning-analytics-4.0.60476.tgz \
      --inventory ibmPlanningAnalyticsOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-planning-analytics-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-planning-analytics-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > Product Master

  - Run the following command to create the Product Master catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-productmaster-1.0.3.tgz \
  --inventory productmasterOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

- Verify that **ibm-productmaster-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-productmaster-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- ⌄ RStudio Server with R 3.6

  - Run the following command to create the RStudio Server with R 3.6 catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-rstudio-1.0.6.tgz \
  --inventory rstudioSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-cpd-rstudio-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cpd-rstudio-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- ⌄ SPSS Modeler

  - Run the following command to create the SPSS Modeler catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-spss-1.0.6.tgz \
  --inventory spssSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-cpd-spss-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cpd-spss-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- ⌄ Virtual Data Pipeline

  Not applicable. For details, see [Installing Virtual Data Pipeline for Cloud Pak for Data](link).

- ⌄ Voice Gateway

  - Run the following command to create the Voice Gateway catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-voice-gateway-1.0.5.tgz \
  --inventory voiceGatewayOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-voice-gateway-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-voice-gateway-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- ⌄ Watson Assistant

  - Run the following command to create the Watson Assistant catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-watson-assistant-4.0.6.tgz \
  --inventory assistantOperator \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-watson-assistant-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-watson-assistant-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > Watson Assistant for Voice Interaction

Watson Assistant for Voice Interaction is comprised of the following services:
- Voice Gateway
- Watson Assistant
- Watson Speech to Text
- Watson Text to Speech

- > Watson Discovery

  - Run the following command to create the Watson Discovery catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-watson-discovery-4.0.6.tgz \
      --inventory discoveryOperators \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-watson-discovery-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-watson-discovery-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > Watson Knowledge Catalog

  - Run the following command to create the Watson Knowledge Catalog catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-wkc-4.0.6.tgz \
      --inventory wkcOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-cpd-wkc-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-cpd-wkc-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > Watson Knowledge Studio

  - Run the following command to create the Watson Knowledge Studio catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-watson-ks-4.0.5.tgz \
      --inventory wksOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-watson-ks-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-watson-ks-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > Watson Machine Learning

  - Run the following command to create the Watson Machine Learning catalog source for the latest refresh:

    ```
    cloudctl case launch \
      --case ${OFFLINEDIR}/ibm-wml-cpd-4.0.7.tgz \
      --inventory wmlOperatorSetup \
      --namespace openshift-marketplace \
      --action install-catalog \
        --args "--inputDir ${OFFLINEDIR} --recursive"
    ```

  - Verify that **ibm-cpd-wml-operator-catalog** is **READY**:

    ```
    oc get catalogsource -n openshift-marketplace ibm-cpd-wml-operator-catalog \
    -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
    ```

    It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- > Watson Machine Learning Accelerator
  - Run the following command to create the Watson Machine Learning Accelerator catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wml-accelerator-2.3.6.tgz \
  --inventory wmla_operator_deploy \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

- Verify that **ibm-cpd-wml-accelerator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cpd-wml-accelerator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- **⟩ Watson OpenScale**

  - Run the following command to create the Watson OpenScale catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-watson-openscale-2.5.1.tgz \
  --inventory ibmWatsonOpenscaleOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-openscale-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-openscale-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- **⟩ Watson Speech to Text**

  The same operator is used for the Watson Speech to Text service and the Watson Text to Speech service. You only need to create the catalog source once.

  - Run the following command to create the Watson Speech to Text catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-watson-speech-4.0.7.tgz \
  --inventory speechOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-watson-speech-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-watson-speech-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- **⟩ Watson Studio**

  - Run the following command to create the Watson Studio catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-wsl-2.0.6.tgz \
  --inventory wslSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

  - Verify that **ibm-cpd-ws-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cpd-ws-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- **⟩ Watson Studio Runtimes**

  The catalog source for Watson Studio Runtimes is automatically created when you create the catalog source for Watson Studio.

  After you create the catalog source for Watson Studio, verify that the Watson Studio Runtimes catalog source was also created.

  - Verify that **ibm-cpd-ws-runtimes-operator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-cpd-ws-runtimes-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

- **⟩ Watson Text to Speech**

  The same operator is used for the Watson Speech to Text service and the Watson Text to Speech service. You only need to create the catalog source once.

- Run the following command to create the Watson Speech to Text catalog source for the latest refresh:

```
cloudctl case launch \
  --case ${OFFLINEDIR}/ibm-watson-speech-4.0.7.tgz \
  --inventory speechOperatorSetup \
  --namespace openshift-marketplace \
  --action install-catalog \
    --args "--inputDir ${OFFLINEDIR} --recursive"
```

- Verify that `ibm-watson-speech-operator-catalog` is `READY`:

```
oc get catalogsource -n openshift-marketplace ibm-watson-speech-operator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return `READY`, wait a few minutes and try to verify the status again.

What's next Now that you've created the required catalog sources for your environment, you are ready to complete [Installing IBM Cloud Pak foundational services](#).

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Creating catalog sources that automatically pull the latest images from the IBM Entitled Registry

If you pull images from the IBM® Entitled Registry you can create catalog sources that automatically pull the latest images from the IBM Entitled Registry.

Attention: Automatically pulling the latest images from the IBM Entitled Registry is not recommended for deployments where predictability and stability are important, such as production environments. This option is recommended only for short-term installations, such as proof-of-concept deployments.
If you choose to automatically pull the latest images from the IBM Entitled Registry, the software on your cluster could be updated at any time. This can be especially problematic if newer and incompatible versions of dependencies are automatically installed on your cluster.

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.

To create catalog sources that pull the latest images from the IBM Entitled Registry:

1. Create the catalog source for the IBM Operator Catalog.
   This catalog source is used by :
   - IBM Cloud Pak® foundational services
   - IBM Cloud Pak for Data platform operator
   - Service operators
   a. Check whether the IBM Operator Catalog already exists on your cluster:

   ```
   oc get catalogsource -n openshift-marketplace
   ```

   Review the output to determine whether there is an entry called `ibm-operator-catalog`.

   b. If the IBM Operator Catalog does not exist, create it:

   ```
   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:
         interval: 45m
   EOF
   ```

   c. Verify that the IBM Operator Catalog was successfully created:

   ```
   oc get catalogsource -n openshift-marketplace
   ```

   Review the output to ensure that there is an entry called `ibm-operator-catalog`.

   d. Verify that `ibm-operator-catalog` is `READY`:

   ```
   oc get catalogsource -n openshift-marketplace ibm-operator-catalog \
   -o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
   ```

   It might take several minutes before the catalog source is ready. If the command does not return `READY`, wait a few minutes and try to verify the status again.

2. 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)

a. Check whether the IBM Db2U Catalog already exists on your cluster:

```
oc get catalogsource -n openshift-marketplace
```

Review the output to determine whether there is an entry called **ibm-db2uoperator-catalog**.

b. If the IBM Db2U Catalog does not exist, create it:

```
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
```

c. Verify that the IBM Db2U Catalog was successfully created:

```
oc get catalogsource -n openshift-marketplace
```

Review the output to ensure that there is an entry called **ibm-db2uoperator-catalog**.

d. Verify that **ibm-db2uoperator-catalog** is **READY**:

```
oc get catalogsource -n openshift-marketplace ibm-db2uoperator-catalog \
-o jsonpath='{.status.connectionState.lastObservedState} {"\n"}'
```

It might take several minutes before the catalog source is ready. If the command does not return **READY**, wait a few minutes and try to verify the status again.

What's next Now that you've created the required catalog sources for your environment, you are ready to complete Installing IBM Cloud Pak foundational services.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see Documentation for previous 4.0.x refreshes.

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 Version 3.15 or later is installed, you can skip this task.
- If an earlier version of IBM Cloud Pak foundational services is installed, follow the guidance in the IBM Cloud Pak foundational services documentation:
  - If you are pulling images directly from the IBM Entitled Registry, follow the guidance in Upgrading foundational services from an operator release
  - If you are pulling images from a private container registry, follow the guidance in Upgrading foundational services from an operator release in an airgap environment.
- If IBM Cloud Pak foundational services is not installed, pick the appropriate action based on your environment:

Express installations
      If all of the operators will be installed in the same project, you can skip this task.
      The IBM Cloud Pak for Data platform operator will automatically install IBM Cloud Pak foundational services in the **ibm-common-services** project.

Specialized installations
      If you plan to install the IBM Cloud Pak for Data operators in a separate project from the IBM Cloud Pak foundational services services, 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 private 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.
3. The required catalog sources exist. For details, see Creating catalog sources.

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. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

   - ` > ` 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
     ```

   - ` > ` IBM Entitled Registry with catalog sources that pull specific versions of images

     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
     ```

   - ` > ` IBM Entitled Registry with a catalog source for the IBM Operator Catalog

     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
     ```

   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.v3.13.0            IBM Cloud Pak foundational services     3.13.0     ibm-common-service-oper.
   ibm-namespace-scope-operator.v1.6.0            IBM NamespaceScope Operator             1.6.0      ibm-namespace-scope-ope.
   operand-deployment-lifecycle-manager.v1.10.0   Operand Deployment Lifecycle Manager    1.10.0     operand-deployment-life.
   ```

   Note: The preceding output is a *sample* and is intended to show the format of the output. Your installation might show a newer version of the IBM Cloud Pak foundational services. Ensure that the version of IBM Cloud Pak foundational services that is displayed in the output matches the version that is listed in Operator and operand versions.

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  YYYY-MM-DDTHH:MM:SSZ
```

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 individual foundational services

Automatically installed services

The IBM Cloud Pak for Data platform operator *automatically* installs the following foundational services if they are not already installed on the cluster:

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.

Recommended services

If you are installing any of the following services, it is recommended that you install the License Service:

- Voice Gateway
- Watson™ Assistant
- Watson Discovery
- Watson Knowledge Studio
- Watson Speech to Text
- Watson Text to Speech

To install the License Service

1. Submit the following operand request to install the License Service operator in the project where you plan to install the Cloud Pak for Data software.
   The command assumes that the following statements are true:
   - You will install Cloud Pak for Data in the `cpd-instance` project. If you plan to install Cloud Pak for Data in a different project, update the `namespace` parameter with the appropriate project for your environment.
   - You installed IBM Cloud Pak foundational servicesin the `ibm-common-services` project. If you installed IBM Cloud Pak foundational services in a different project, update the `registryNamespace` parameter with the appropriate project for your environment.

```
cat <<EOF |oc apply -f -
apiVersion: operator.ibm.com/v1alpha1
kind: OperandRequest
metadata:
  name: common-service-license
  namespace: cpd-instance      # Replace with the project where you will install Cloud Pak for Data
spec:
  requests:
  - operands:
      - name: ibm-licensing-operator
        bindings:
          public-api-upload:
            secret: ibm-licensing-upload-token
            configmap: ibm-licensing-upload-config
    registry: common-service
    registryNamespace: ibm-common-services       # Replace with the project where Cloud Pak foundational services are
installed
EOF
```

2. Validate that the operator was successfully created.
   a. Run the following command to confirm that the operand request was created:

   ```
   oc get opreq common-service-license -n ibm-common-services \
   -o jsonpath='{.status.phase}  {"\n"}'
   ```

   Verify that the command returns `Running`. If the command returns `Initialized` or `Installing`, wait several minutes and run the command again.

   b. Run the following command to confirm that the operator is installed:

   ```
   oc get pod -n ibm-common-services -l app.kubernetes.io/name=ibm-licensing \
   -o jsonpath='{.items[0].status.phase} {"\n"}'
   ```

   Verify that the command returns `Running`.

Other services
> To install additional foundational services, 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](#).

What's next Now that you've installed IBM Cloud Pak foundational services, you're ready to complete [Creating operator subscriptions](#).

Previous topic: [Creating catalog sources](#)
Next topic: [Creating operator subscriptions](#)
Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

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

For an in-depth description of operators, see the [Red Hat® OpenShift®: Operators Framework video](#) from Red Hat.

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. Enabling services to use namespace scoping with third-party operators](#)
- [5. 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. Red Hat OpenShift Container Platform supports the following install plans:

Automatic
> If you specify `installPlanApproval: Automatic`, OpenShift 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 initial request to create the operator
> - Subsequent requests to update the operator to a newer version.

Note: An operator *manages* software. Upgrading the operator does not necessarily impact the version of the software that is running on the cluster. In most cases, the software will not be automatically upgraded even after the operator is upgraded. However, there are exceptions to this rule:

- The software does not support a `version` entry in the custom resource. For example, the Voice Gateway service does not support a `version` entry in the custom resource, which means that the service is automatically upgraded when you install a newer version of the operator on the cluster.
- You choose an automatic install plan when you install or upgrade the services on your cluster. When you choose an automatic upgrade plan, you remove the `version` entry from the custom resource.

Review the documentation for the [services](#) that you plan to install to determine whether:

- The services support a `version` entry
- The services support the automatic upgrade

Use the following guidance to choose an install plan:

Private container registry
> When you pull images from a private container registry, you must mirror the images to the registry and update the catalog source before you can upgrade the software.
>
> - Use the automatic install plan to simplify the process of managing your environment.
> - Use the manual install plan to ensure that all aspects of the software remain at the same version until you are ready to upgrade the software.

IBM® Entitled Registry

The guidance for the IBM Entitled Registry depends on the type of catalog source that you created:

Catalog source that pulls a specific version of the software from the IBM Entitled Registry
When you create catalog sources that pull a specific version of the software from the IBM Entitled Registry, you must update the catalog source before you can upgrade the software.

- Use the automatic install plan to simplify the process of managing your environment.
- Use the manual install plan to ensure that all aspects of the software remain at the same version until you are ready to upgrade the software.

Catalog source that automatically pulls the latest version of the software from the IBM Entitled Registry
The IBM Operator Catalog automatically pulls the latest version of the software from the IBM Entitled Registry. The IBM Operator Catalog is refreshed any time any IBM operator is released. OLM handles any updates that it finds in the IBM Operator Catalog (`ibm-operator-catalog`) automatically.
With this configuration, the manual install plan reduces, but does not eliminate, the likelihood that the software will be updated without your knowledge.
Remember: If a service does not include or support a `version` entry in the custom resource, the service will be automatically upgraded when the operator is upgraded.

Note: All of the operator subscriptions in this topic specify the automatic install plan (`installPlanApproval: Automatic`). If you want to use the manual install plan, update the subscription to use `installPlanApproval: Manual` before you create the operator subscription on your environment.
If you set `installPlanApproval: Manual`, you should also complete [Specifying the install plan for operators that are automatically installed by Operand Deployment Lifecycle Manager](#) after you install the Cloud Pak for Data control plane.

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. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

- > Private container registry
  1. Create the following operator subscription.

     If you did not install IBM Cloud Pak® foundational services in `ibm-common-services`, update the value in the `namespace: ibm-common-services` entry.

     ```
     cat <<EOF |oc apply -f -
     apiVersion: operators.coreos.com/v1alpha1
     kind: Subscription
     metadata:
       name: ibm-cpd-scheduling-catalog-subscription
       namespace: ibm-common-services     # Specify the project that contains the Cloud Pak foundational services operators
     spec:
       channel: v1.3
       installPlanApproval: Automatic
       name: ibm-cpd-scheduling-operator
       source: ibm-cpd-scheduling-catalog
       sourceNamespace: openshift-marketplace
     EOF
     ```

  2. Validate that the operator was successfully created.
     a. Run the following command to confirm that the subscription was triggered:

        ```
        oc get sub -n ibm-common-services ibm-cpd-scheduling-catalog-subscription \
        -o jsonpath='{.status.installedCSV} {"\n"}'
        ```

        Verify that the command returns `ibm-cpd-scheduling-operator.v1.3.2`.

     b. Run the following command to confirm that the cluster service version (CSV) is ready:

        ```
        oc get csv -n ibm-common-services ibm-cpd-scheduling-operator.v1.3.2 \
        -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
        ```

        Verify that the command returns `Succeeded : install strategy completed with no errors`.

     c. Run the following command to confirm that the operator is ready:

        ```
        oc get deployments -n ibm-common-services -l olm.owner="ibm-cpd-scheduling-operator.v1.3.2" \
        -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
        ```

        Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

  Remember: If you created the scheduling service operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve the update request for `ibm-cpd-scheduling-operator`.

- > IBM Entitled Registry with catalog sources that pull specific versions of images
  1. Create the following operator subscription.

     If you did not install IBM Cloud Pak foundational services in `ibm-common-services`, update the value in the `namespace: ibm-common-services` entry.

     ```
     cat <<EOF |oc apply -f -
     apiVersion: operators.coreos.com/v1alpha1
     kind: Subscription
     metadata:
       name: ibm-cpd-scheduling-catalog-subscription
       namespace: ibm-common-services     # Specify the project that contains the Cloud Pak foundational services operators
     spec:
     ```

```
    channel: v1.3
    installPlanApproval: Automatic
    name: ibm-cpd-scheduling-operator
    source: ibm-cpd-scheduling-catalog
    sourceNamespace: openshift-marketplace
EOF
```

2. Validate that the operator was successfully created.
   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n ibm-common-services ibm-cpd-scheduling-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-cpd-scheduling-operator.v1.3.2**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n ibm-common-services ibm-cpd-scheduling-operator.v1.3.2 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n ibm-common-services -l olm.owner="ibm-cpd-scheduling-operator.v1.3.2" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Remember: If you created the scheduling service operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve the update request for **ibm-cpd-scheduling-operator**.

- > IBM Entitled Registry with a catalog source for the IBM Operator Catalog

  1. Create the following operator subscription.

     If you did not install IBM Cloud Pak foundational services in **ibm-common-services**, update the value in the **namespace: ibm-common-services** entry.

     ```
     cat <<EOF |oc apply -f -
     apiVersion: operators.coreos.com/v1alpha1
     kind: Subscription
     metadata:
       name: ibm-cpd-scheduling-catalog-subscription
       namespace: ibm-common-services     # Specify the project that contains the Cloud Pak foundational services operators
     spec:
       channel: v1.3
       installPlanApproval: Automatic
       name: ibm-cpd-scheduling-operator
       source: ibm-operator-catalog
       sourceNamespace: openshift-marketplace
     EOF
     ```

  2. Validate that the operator was successfully created.
     a. Run the following command to confirm that the subscription was triggered:

     ```
     oc get sub -n ibm-common-services ibm-cpd-scheduling-catalog-subscription \
     -o jsonpath='{.status.installedCSV} {"\n"}'
     ```

     Verify that the command returns **ibm-cpd-scheduling-operator.v1.3.2**.

     b. Run the following command to confirm that the cluster service version (CSV) is ready:

     ```
     oc get csv -n ibm-common-services ibm-cpd-scheduling-operator.v1.3.2 \
     -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
     ```

     Verify that the command returns **Succeeded : install strategy completed with no errors**.

     c. Run the following command to confirm that the operator is ready:

     ```
     oc get deployments -n ibm-common-services -l olm.owner="ibm-cpd-scheduling-operator.v1.3.2" \
     -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
     ```

     Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Remember: If you created the scheduling service operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve the update request for **ibm-cpd-scheduling-operator**.

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

Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

- > Private container registry

  1. Create the following operator subscription.
     Ensure that you update the **namespace: ibm-common-services|cpd-operators** entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

     ```
     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 where you want to install the Cloud Pak for
Data platform operator
spec:
  channel: v2.0
  installPlanApproval: Automatic
  name: cpd-platform-operator
  source: cpd-platform
  sourceNamespace: openshift-marketplace
EOF
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

      ```
      oc get sub -n operator-project cpd-operator \
      -o jsonpath='{.status.installedCSV} {"\n"}'
      ```

      Verify that the command returns `cpd-platform-operator.v2.0.6`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

      ```
      oc get csv -n operator-project cpd-platform-operator.v2.0.6 \
      -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
      ```

      Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

      ```
      oc get deployments -n operator-project -l olm.owner="cpd-platform-operator.v2.0.6" \
      -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
      ```

      Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

3. 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
   ```

Important: If you created the IBM Cloud Pak for Data platform operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
   - `cpd-platform-operator`
   - `ibm-cert-manager-operator` *
   - `ibm-common-service-operator` *
   - `ibm-namespace-scope-operator` *
   - `ibm-zen-operator` *
   - `operand-deployment-lifecycle-manager` *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- ⟩
  IBM Entitled Registry with catalog sources that pull specific versions of images

  1. Create the following operator subscription.
     Ensure that you update the `namespace: ibm-common-services|cpd-operators`
     entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

     ```
     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 where you want to install the Cloud Pak for
     Data platform operator
     spec:
       channel: v2.0
       installPlanApproval: Automatic
       name: cpd-platform-operator
       source: cpd-platform
       sourceNamespace: openshift-marketplace
     EOF
     ```

  2. Validate that the operator was successfully created.
     For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project cpd-operator \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns **cpd-platform-operator.v2.0.6**.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project cpd-platform-operator.v2.0.6 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns **Succeeded : install strategy completed with no errors**.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="cpd-platform-operator.v2.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

3. 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
```

Important: If you created the IBM Cloud Pak for Data platform operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:

- **cpd-platform-operator**
- **ibm-cert-manager-operator** *
- **ibm-common-service-operator** *
- **ibm-namespace-scope-operator** *
- **ibm-zen-operator** *
- **operand-deployment-lifecycle-manager** *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- ⟩
  IBM Entitled Registry with a catalog source for the IBM Operator Catalog

  1. Create the following operator subscription.
     Ensure that you update the **namespace:**
     ibm-common-services|cpd-operators
     entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

     ```
     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 where you want to install the Cloud Pak for
     Data platform operator
     spec:
       channel: v2.0
       installPlanApproval: Automatic
       name: cpd-platform-operator
       source: ibm-operator-catalog
       sourceNamespace: openshift-marketplace
     EOF
     ```

  2. Validate that the operator was successfully created.
     For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

     a. Run the following command to confirm that the subscription was triggered:

     ```
     oc get sub -n operator-project cpd-operator \
     -o jsonpath='{.status.installedCSV} {"\n"}'
     ```

     Verify that the command returns **cpd-platform-operator.v2.0.6**.

     b. Run the following command to confirm that the cluster service version (CSV) is ready:

     ```
     oc get csv -n operator-project cpd-platform-operator.v2.0.6 \
     -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
     ```

     Verify that the command returns **Succeeded : install strategy completed with no errors**.

     c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="cpd-platform-operator.v2.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

3. 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
```

Important: If you created the IBM Cloud Pak for Data platform operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `cpd-platform-operator`
- `ibm-cert-manager-operator` *
- `ibm-common-service-operator` *
- `ibm-namespace-scope-operator` *
- `ibm-zen-operator` *
- `operand-deployment-lifecycle-manager` *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

4. Enabling services to use namespace scoping with third-party operators

Some services have dependencies on third-party operators, such as open source operators. By default, the service has no control over the namespace scope of these third-party operators.

However, you can enable the `csvInjector` option in the **IBM NamespaceScope Operator** to ensure that the required third-party operators use the same namespace scope as the service that requires them. (Setting the `csvInjector` option enables a service to augment the cluster service version (CSV) of a third-party operator.)

To ensure that the service can update the CSV, you must enable the `csvInjector` option before you create the operator subscription for the service.

This setting is required for the following services:

- Data Virtualization
- Db2® Data Management Console
- IBM Match 360 with Watson™
- Watson Assistant
- Watson Assistant for Voice Interaction
- Watson Discovery
- Watson Speech to Text
- Watson Text to Speech

To update the **IBM NamespaceScope Operator**, run the appropriate command for your environment:

- **⟩**
  Express installations

  Run the following command to update the **IBM NamespaceScope Operator** in the **ibm-common-services** project:

  ```
  oc patch NamespaceScope common-service \
  -n ibm-common-services \
  --type=merge \
  --patch='{"spec": {"csvInjector": {"enable": true} } }'
  ```

- **⟩**
  Specialized installations

  Run the following command to update the **IBM NamespaceScope Operator** in the **cpd-operators** project:

  ```
  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:
    csvInjector:
      enable: true
    namespaceMembers:
    - cpd-operators                   # (Default) Replace with the Cloud Pak for Data platform operator project name
  EOF
  ```

5. Creating an operator subscription for services

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

- **Anaconda Repository for IBM Cloud Pak for Data**

  Not applicable. For details, see [Installing Anaconda Repository for IBM Cloud Pak for Data](#).

- **Analytics Engine Powered by Apache Spark**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - > **Private container registry**

    1. Create the following operator subscription.
       Ensure that you update the `namespace:` `ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       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
       ```

    2. Validate that the operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

       a. Run the following command to confirm that the subscription was triggered:

          ```
          oc get sub -n operator-project ibm-cpd-ae-operator-subscription \
          -o jsonpath='{.status.installedCSV} {"\n"}'
          ```

          Verify that the command returns `ibm-cpd-ae.v1.0.6`.

       b. Run the following command to confirm that the cluster service version (CSV) is ready:

          ```
          oc get csv -n operator-project ibm-cpd-ae.v1.0.6 \
          -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
          ```

          Verify that the command returns `Succeeded : install strategy completed with no errors`.

       c. Run the following command to confirm that the operator is ready:

          ```
          oc get deployments -n operator-project -l olm.owner="ibm-cpd-ae.v1.0.6" \
          -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
          ```

          Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

    Remember: If you created the Analytics Engine Powered by Apache Spark operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve the update request for `analyticsengine-operator`.

  - > **IBM Entitled Registry with catalog sources that pull specific versions of images**

    1. Create the following operator subscription.
       Ensure that you update the `namespace:` `ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       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
       ```

    2. Validate that the operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-cpd-ae-operator-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns **ibm-cpd-ae.v1.0.6**.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-cpd-ae.v1.0.6 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns **Succeeded : install strategy completed with no errors**.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-ae.v1.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Remember: If you created the Analytics Engine Powered by Apache Spark operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve the update request for **analyticsengine-operator**.

- ⌄

  IBM Entitled Registry with a catalog source for the IBM Operator Catalog

  1. Create the following operator subscription.
  Ensure that you update the **namespace: ibm-common-services|cpd-operators** entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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
```

  2. Validate that the operator was successfully created.
  For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

      a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-cpd-ae-operator-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

      Verify that the command returns **ibm-cpd-ae.v1.0.6**.

      b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-cpd-ae.v1.0.6 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

      Verify that the command returns **Succeeded : install strategy completed with no errors**.

      c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-ae.v1.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

      Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Remember: If you created the Analytics Engine Powered by Apache Spark operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve the update request for **analyticsengine-operator**.

- **Cognos® Analytics**

Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

- ⌄

  Private container registry

  1. Create the following operator subscription.
  Ensure that you update the **namespace: ibm-common-services|cpd-operators** entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v4.0
  name: ibm-ca-operator
  installPlanApproval: Automatic
  source: ibm-ca-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-ca-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-ca-operator.v4.0.6**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-ca-operator.v4.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-ca-operator.v4.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the Cognos Analytics operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **ibm-ca-operator**
- **ibm-cpd-ccs-operator** *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○ >
  IBM Entitled Registry with catalog sources that pull specific versions of images

   1. Create the following operator subscription.
      Ensure that you update the **namespace: ibm-common-services|cpd-operators**
      entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

      ```
      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: v4.0
        name: ibm-ca-operator
        installPlanApproval: Automatic
        source: ibm-ca-operator-catalog
        sourceNamespace: openshift-marketplace
      EOF
      ```

   2. Validate that the operator was successfully created.
      For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

      a. Run the following command to confirm that the subscription was triggered:

      ```
      oc get sub -n operator-project ibm-ca-operator-catalog-subscription \
      -o jsonpath='{.status.installedCSV} {"\n"}'
      ```

      Verify that the command returns **ibm-ca-operator.v4.0.6**.

      b. Run the following command to confirm that the cluster service version (CSV) is ready:

      ```
      oc get csv -n operator-project ibm-ca-operator.v4.0.6 \
      -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
      ```

      Verify that the command returns **Succeeded : install strategy completed with no errors**.

      c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-ca-operator.v4.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to 1. If the command returns 0, wait for the deployment to become available.

Important: If you created the Cognos Analytics operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- `ibm-ca-operator`
- `ibm-cpd-ccs-operator` *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○ >
IBM Entitled Registry with a catalog source for the IBM Operator Catalog

1. Create the following operator subscription.
   Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v4.0
  name: ibm-ca-operator
  installPlanApproval: Automatic
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:
   ```
   oc get sub -n operator-project ibm-ca-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```
   Verify that the command returns `ibm-ca-operator.v4.0.6`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:
   ```
   oc get csv -n operator-project ibm-ca-operator.v4.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```
   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:
   ```
   oc get deployments -n operator-project -l olm.owner="ibm-ca-operator.v4.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```
   Verify that the command returns an integer greater than or equal to 1. If the command returns 0, wait for the deployment to become available.

Important: If you created the Cognos Analytics operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- `ibm-ca-operator`
- `ibm-cpd-ccs-operator` *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Cognos Dashboards**

Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

○ >
Private container registry

1. Create the following operator subscription.
   Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-cde-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-cpd-cde.v1.0.6**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-cde.v1.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-cpd-cde.v1.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the Cognos Dashboards operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **ibm-cde-operator**
- **ibm-cpd-ccs-operator** *
* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

>
IBM Entitled Registry with catalog sources that pull specific versions of images

1. Create the following operator subscription.
   Ensure that you update the **namespace:**
   ibm-common-services|cpd-operators
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   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
   ```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-cde-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-cpd-cde.v1.0.6**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-cde.v1.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-cpd-cde.v1.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the Cognos Dashboards operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **ibm-cde-operator**
- **ibm-cpd-ccs-operator** *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- ⟩

  IBM Entitled Registry with a catalog source for the IBM Operator Catalog

  1. Create the following operator subscription.
     Ensure that you update the **namespace:**
     ibm-common-services|cpd-operators
     entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

     ```
     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
     ```

  2. Validate that the operator was successfully created.
     For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace (-n)** argument.

     a. Run the following command to confirm that the subscription was triggered:

        ```
        oc get sub -n operator-project ibm-cde-operator-subscription \
        -o jsonpath='{.status.installedCSV} {"\n"}'
        ```

        Verify that the command returns **ibm-cpd-cde.v1.0.6**.

     b. Run the following command to confirm that the cluster service version (CSV) is ready:

        ```
        oc get csv -n operator-project ibm-cpd-cde.v1.0.6 \
        -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
        ```

        Verify that the command returns **Succeeded : install strategy completed with no errors**.

     c. Run the following command to confirm that the operator is ready:

        ```
        oc get deployments -n operator-project -l olm.owner="ibm-cpd-cde.v1.0.6" \
        -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
        ```

        Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

  Important: If you created the Cognos Dashboards operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
  - **ibm-cde-operator**
  - **ibm-cpd-ccs-operator** *

  * An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Data Privacy**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - ⟩

    Private container registry

    1. Create the following operator subscription.
       Ensure that you update the **namespace:**
       ibm-common-services|cpd-operators
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       cat <<EOF |oc apply -f -
       apiVersion: operators.coreos.com/v1alpha1
       kind: Subscription
       metadata:
         labels:
           app.kubernetes.io/instance:  ibm-cpd-dp-operator-catalog-subscription
           app.kubernetes.io/managed-by: ibm-cpd-dp-operator
           app.kubernetes.io/name:  ibm-cpd-dp-operator-catalog-subscription
         namespace: ibm-common-services|cpd-operators    # Pick the project that contains the Cloud Pak for Data
       operator
         name: ibm-cpd-dp-operator-catalog-subscription
       ```

```
spec:
    channel: v1.0
    installPlanApproval: Automatic
    name: ibm-cpd-dp
    source: ibm-cpd-dp-operator-catalog
    sourceNamespace: openshift-marketplace
EOF
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-cpd-dp-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-cpd-dp.v1.0.6`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-dp.v1.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-cpd-dp.v1.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the Data Privacy operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- `analyticsengine-operator` *
- `ibm-cpd-dp`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- >
  IBM Entitled Registry with catalog sources that pull specific versions of images
  1. Create the following operator subscription.
     Ensure that you update the `namespace: ibm-common-services|cpd-operators`
     entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

     ```
     cat <<EOF |oc apply -f -
     apiVersion: operators.coreos.com/v1alpha1
     kind: Subscription
     metadata:
       labels:
         app.kubernetes.io/instance:  ibm-cpd-dp-operator-catalog-subscription
         app.kubernetes.io/managed-by: ibm-cpd-dp-operator
         app.kubernetes.io/name:  ibm-cpd-dp-operator-catalog-subscription
       namespace: ibm-common-services|cpd-operators     # Pick the project that contains the Cloud Pak for Data
     operator
       name: ibm-cpd-dp-operator-catalog-subscription
     spec:
         channel: v1.0
         installPlanApproval: Automatic
         name: ibm-cpd-dp
         source: ibm-cpd-dp-operator-catalog
         sourceNamespace: openshift-marketplace
     EOF
     ```

  2. Validate that the operator was successfully created.
     For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

     a. Run the following command to confirm that the subscription was triggered:

     ```
     oc get sub -n operator-project ibm-cpd-dp-operator-catalog-subscription \
     -o jsonpath='{.status.installedCSV} {"\n"}'
     ```

     Verify that the command returns `ibm-cpd-dp.v1.0.6`.

     b. Run the following command to confirm that the cluster service version (CSV) is ready:

     ```
     oc get csv -n operator-project ibm-cpd-dp.v1.0.6 \
     -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
     ```

     Verify that the command returns `Succeeded : install strategy completed with no errors`.

     c. Run the following command to confirm that the operator is ready:

     ```
     oc get deployments -n operator-project -l olm.owner="ibm-cpd-dp.v1.0.6" \
     -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
     ```

     Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the Data Privacy operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `analyticsengine-operator` *
- `ibm-cpd-dp`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- > IBM Entitled Registry with a catalog source for the IBM Operator Catalog

    1. Create the following operator subscription.
       Ensure that you update the `namespace:` *`ibm-common-services|cpd-operators`* entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       cat <<EOF |oc apply -f -
       apiVersion: operators.coreos.com/v1alpha1
       kind: Subscription
       metadata:
         labels:
           app.kubernetes.io/instance:  ibm-cpd-dp-operator-catalog-subscription
           app.kubernetes.io/managed-by: ibm-cpd-dp-operator
           app.kubernetes.io/name:  ibm-cpd-dp-operator-catalog-subscription
         namespace: ibm-common-services|cpd-operators    # Pick the project that contains the Cloud Pak for Data
       operator
         name: ibm-cpd-dp-operator-catalog-subscription
       spec:
         channel: v1.0
         installPlanApproval: Automatic
         name: ibm-cpd-dp
         source: ibm-operator-catalog
         sourceNamespace: openshift-marketplace
       EOF
       ```

    2. Validate that the operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

         a. Run the following command to confirm that the subscription was triggered:

            ```
            oc get sub -n operator-project ibm-cpd-dp-operator-catalog-subscription \
            -o jsonpath='{.status.installedCSV} {"\n"}'
            ```

            Verify that the command returns `ibm-cpd-dp.v1.0.6`.

         b. Run the following command to confirm that the cluster service version (CSV) is ready:

            ```
            oc get csv -n operator-project ibm-cpd-dp.v1.0.6 \
            -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
            ```

            Verify that the command returns `Succeeded : install strategy completed with no errors`.

         c. Run the following command to confirm that the operator is ready:

            ```
            oc get deployments -n operator-project -l olm.owner="ibm-cpd-dp.v1.0.6" \
            -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
            ```

            Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

    Important: If you created the Data Privacy operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

    - `analyticsengine-operator` *
    - `ibm-cpd-dp`

    * An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Data Refinery**

  The operator subscription for Data Refinery is automatically created when you create the operator subscription for either Watson Knowledge Catalog or Watson Studio.

- **Data Virtualization**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

    - > Private container registry

        1. Create the Db2U operator subscription.
           Ensure that you update the `namespace:` *`ibm-common-services|cpd-operators`* entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

           ```
           cat <<EOF |oc apply -f -
           apiVersion: operators.coreos.com/v1alpha1
           kind: Subscription
           metadata:
             name: ibm-db2u-operator
             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. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

      a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-db2u-operator \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

      Verify that the command returns `db2u-operator.v1.1.11`.

      b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project db2u-operator.v1.1.11 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

      Verify that the command returns `Succeeded : install strategy completed with no errors`.

      c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="db2u-operator.v1.1.11" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

      Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

3. Create the Data Virtualization operator subscription.
   Ensure that you update the `namespace:`
   `ibm-common-services|cpd-operators`
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v1.7
  installPlanApproval: Automatic
  name: ibm-dv-operator
  source: ibm-dv-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

4. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

      a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-dv-operator-catalog-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

      Verify that the command returns `ibm-dv-operator.v1.7.6`.

      b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-dv-operator.v1.7.6 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

      Verify that the command returns `Succeeded : install strategy completed with no errors`.

      c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-dv-operator.v1.7.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

      Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for Data Virtualization with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- `db2u-operator` *
- `ibm-cpd-ccs-operator` *
- `ibm-dmc-operator` *
- `ibm-dv-operator`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

    ◦
> IBM Entitled Registry with catalog sources that pull specific versions of images

      1. Create the Db2U operator subscription.
      Ensure that you update the `namespace:`
      `ibm-common-services|cpd-operators`

entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-db2u-operator
  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. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-db2u-operator \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **db2u-operator.v1.1.11**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project db2u-operator.v1.1.11 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="db2u-operator.v1.1.11" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

3. Create the Data Virtualization operator subscription.
   Ensure that you update the **namespace: ibm-common-services|cpd-operators**
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v1.7
  installPlanApproval: Automatic
  name: ibm-dv-operator
  source: ibm-dv-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

4. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-dv-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-dv-operator.v1.7.6**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-dv-operator.v1.7.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-dv-operator.v1.7.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the preceding operators for Data Virtualization with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:

- **db2u-operator** *
- **ibm-cpd-ccs-operator** *
- **ibm-dmc-operator** *
- **ibm-dv-operator**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

-      ⟩
       IBM Entitled Registry with a catalog source for the IBM Operator Catalog

1. Create the Db2U operator subscription.
   Ensure that you update the **namespace:**
   ibm-common-services|cpd-operators
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   cat <<EOF |oc apply -f -
   apiVersion: operators.coreos.com/v1alpha1
   kind: Subscription
   metadata:
     name: ibm-db2u-operator
     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. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-db2u-operator \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **db2u-operator.v1.1.11**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project db2u-operator.v1.1.11 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="db2u-operator.v1.1.11" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

3. Create the Data Virtualization operator subscription.
   Ensure that you update the **namespace:**
   ibm-common-services|cpd-operators
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   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: v1.7
     installPlanApproval: Automatic
     name: ibm-dv-operator
     source: ibm-operator-catalog
     sourceNamespace: openshift-marketplace
   EOF
   ```

4. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-dv-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-dv-operator.v1.7.6**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-dv-operator.v1.7.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-dv-operator.v1.7.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the preceding operators for Data Virtualization with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- **db2u-operator** *
- **ibm-cpd-ccs-operator** *
- **ibm-dmc-operator** *
- **ibm-dv-operator**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **DataStage®**

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

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - > **Private container registry**

    1. Create the following operator subscription.
       Ensure that you update the **namespace:** *ibm-common-services|cpd-operators* entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       cat <<EOF |oc apply -f -
       apiVersion: operators.coreos.com/v1alpha1
       kind: Subscription
       metadata:
         name: ibm-cpd-datastage-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-cpd-datastage-operator
         source: ibm-cpd-datastage-operator-catalog
         sourceNamespace: openshift-marketplace
       EOF
       ```

    2. Validate that the operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

       a. Run the following command to confirm that the subscription was triggered:

          ```
          oc get sub -n operator-project ibm-cpd-datastage-operator-subscription \
          -o jsonpath='{.status.installedCSV} {"\n"}'
          ```

          Verify that the command returns **ibm-cpd-datastage-operator.v1.0.4**.

       b. Run the following command to confirm that the cluster service version (CSV) is ready:

          ```
          oc get csv -n operator-project ibm-cpd-datastage-operator.v1.0.4 \
          -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
          ```

          Verify that the command returns **Succeeded : install strategy completed with no errors**.

       c. Run the following command to confirm that the operator is ready:

          ```
          oc get deployments -n operator-project -l olm.owner="ibm-cpd-datastage-operator.v1.0.4" \
          -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
          ```

          Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

  Important: If you created the DataStage operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
  - **ibm-cpd-ccs-operator** *
  - **ibm-cpd-datastage-operator**

  * An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

  - > **IBM Entitled Registry with catalog sources that pull specific versions of images**

    1. Create the following operator subscription.
       Ensure that you update the **namespace:** *ibm-common-services|cpd-operators* entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       cat <<EOF |oc apply -f -
       apiVersion: operators.coreos.com/v1alpha1
       kind: Subscription
       metadata:
         name: ibm-cpd-datastage-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-cpd-datastage-operator
       ```

```
      source: ibm-cpd-datastage-operator-catalog
      sourceNamespace: openshift-marketplace
EOF
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-cpd-datastage-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-cpd-datastage-operator.v1.0.4`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-datastage-operator.v1.0.4 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-cpd-datastage-operator.v1.0.4" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the DataStage operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
   - `ibm-cpd-ccs-operator` *
   - `ibm-cpd-datastage-operator`
* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

   ○ >
   [IBM Entitled Registry with a catalog source for the IBM Operator Catalog]

   1. Create the following operator subscription.
      Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

      ```
      cat <<EOF |oc apply -f -
      apiVersion: operators.coreos.com/v1alpha1
      kind: Subscription
      metadata:
        name: ibm-cpd-datastage-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-cpd-datastage-operator
        source: ibm-operator-catalog
        sourceNamespace: openshift-marketplace
      EOF
      ```

   2. Validate that the operator was successfully created.
      For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

      a. Run the following command to confirm that the subscription was triggered:

      ```
      oc get sub -n operator-project ibm-cpd-datastage-operator-subscription \
      -o jsonpath='{.status.installedCSV} {"\n"}'
      ```

      Verify that the command returns `ibm-cpd-datastage-operator.v1.0.4`.

      b. Run the following command to confirm that the cluster service version (CSV) is ready:

      ```
      oc get csv -n operator-project ibm-cpd-datastage-operator.v1.0.4 \
      -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
      ```

      Verify that the command returns `Succeeded : install strategy completed with no errors`.

      c. Run the following command to confirm that the operator is ready:

      ```
      oc get deployments -n operator-project -l olm.owner="ibm-cpd-datastage-operator.v1.0.4" \
      -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
      ```

      Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the DataStage operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
   - `ibm-cpd-ccs-operator` *
   - `ibm-cpd-datastage-operator`
* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Db2**

Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

- > Private container registry

  1. **Specialized installations only.** By default, IBM Cloud Pak foundational services expects that the Db2U operator (`ibm-db2u-operator`) is installed in the `ibm-common-services` project. However, if you are using the *specialized* installation architecture (where the IBM Cloud Pak foundational services operators and Cloud Pak for Data operators are in separate OpenShift projects), you must edit the IBM Cloud Pak foundational services operand registry to point to the project where the Cloud Pak for Data operators are installed:

     a. Run the following command to edit the IBM Cloud Pak foundational services operand registry.
        The following command assumes that the IBM Cloud Pak foundational services are installed in the `ibm-common-services` project.

     ```
     oc edit operandRegistry common-service \
     --namespace ibm-common-services
     ```

     b. Change the value of the `namespace` entry for the `ibm-db2u-operator`. Specify the OpenShift project where the Cloud Pak for Data operators are installed, for example, `cpd-operators`:

     ```
     - channel: v1.1
         installPlanApproval: Automatic
         name: ibm-db2u-operator
         namespace: cpd-operators      # Specify the project where the Cloud Pak for Data operators are installed
         packageName: db2u-operator
         scope: public
         sourceName: ibm-db2uoperator-catalog
         sourceNamespace: openshift-marketplace
     ```

     c. Save your changes and exit. For example, if you are using `vi`, hit `esc` and enter `:wq`.

  2. Create the following operator subscription.
     Ensure that you update the `namespace: ibm-common-services|cpd-operators`
     entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

     ```
     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
     ```

  3. Validate that the operator was successfully created.
     For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

     a. Run the following command to confirm that the subscription was triggered:

     ```
     oc get sub -n operator-project ibm-db2oltp-cp4d-operator-catalog-subscription \
     -o jsonpath='{.status.installedCSV} {"\n"}'
     ```

     Verify that the command returns `ibm-db2oltp-cp4d-operator.v1.0.8`.

     b. Run the following command to confirm that the cluster service version (CSV) is ready:

     ```
     oc get csv -n operator-project ibm-db2oltp-cp4d-operator.v1.0.8 \
     -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
     ```

     Verify that the command returns `Succeeded : install strategy completed with no errors`.

     c. Run the following command to confirm that the operator is ready:

     ```
     oc get deployments -n operator-project -l olm.owner="ibm-db2oltp-cp4d-operator.v1.0.8" \
     -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
     ```

     Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

  Important: If you created preceding operators for Db2 with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
  - `db2u-operator` *
  - `ibm-db2oltp-cp4d-operator`

  * An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- > IBM Entitled Registry with catalog sources that pull specific versions of images

  1. **Specialized installations only.** By default, IBM Cloud Pak foundational services expects that the Db2U operator (`ibm-db2u-operator`) is installed in the `ibm-common-services` project. However, if you are using the *specialized* installation architecture (where the IBM Cloud Pak foundational services operators and Cloud Pak for Data operators are in separate OpenShift projects), you must edit the IBM Cloud Pak foundational services operand registry to point to the project where the Cloud Pak for Data operators are installed:

     a. Run the following command to edit the IBM Cloud Pak foundational services operand registry.
        The following command assumes that the IBM Cloud Pak foundational services are installed in the `ibm-common-services` project.

```
oc edit operandRegistry common-service \
--namespace ibm-common-services
```

b. Change the value of the **namespace** entry for the **ibm-db2u-operator**. Specify the OpenShift project where the Cloud Pak for Data operators are installed, for example, **cpd-operators**:

```
- channel: v1.1
    installPlanApproval: Automatic
    name: ibm-db2u-operator
    namespace: cpd-operators     # Specify the project where the Cloud Pak for Data operators are installed
    packageName: db2u-operator
    scope: public
    sourceName: ibm-db2uoperator-catalog
    sourceNamespace: openshift-marketplace
```

c. Save your changes and exit. For example, if you are using **vi**, hit esc and enter :wq.

2. Create the following operator subscription.

Ensure that you update the **namespace:** *ibm-common-services|cpd-operators* entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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
```

3. Validate that the operator was successfully created.

For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-db2oltp-cp4d-operator-catalog-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns **ibm-db2oltp-cp4d-operator.v1.0.8**.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-db2oltp-cp4d-operator.v1.0.8 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns **Succeeded : install strategy completed with no errors**.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-db2oltp-cp4d-operator.v1.0.8" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created preceding operators for Db2 with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **db2u-operator** *
- **ibm-db2oltp-cp4d-operator**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

o   ⟩

> IBM Entitled Registry with a catalog source for the IBM Operator Catalog

1. **Specialized installations only.** By default, IBM Cloud Pak foundational services expects that the Db2U operator (**ibm-db2u-operator**) is installed in the **ibm-common-services** project. However, if you are using the *specialized* installation architecture (where the IBM Cloud Pak foundational services operators and Cloud Pak for Data operators are in separate OpenShift projects), you must edit the IBM Cloud Pak foundational services operand registry to point to the project where the Cloud Pak for Data operators are installed:

a. Run the following command to edit the IBM Cloud Pak foundational services operand registry.

The following command assumes that the IBM Cloud Pak foundational services are installed in the **ibm-common-services** project.

```
oc edit operandRegistry common-service \
--namespace ibm-common-services
```

b. Change the value of the **namespace** entry for the **ibm-db2u-operator**. Specify the OpenShift project where the Cloud Pak for Data operators are installed, for example, **cpd-operators**:

```
- channel: v1.1
    installPlanApproval: Automatic
    name: ibm-db2u-operator
    namespace: cpd-operators     # Specify the project where the Cloud Pak for Data operators are installed
    packageName: db2u-operator
    scope: public
    sourceName: ibm-db2uoperator-catalog
    sourceNamespace: openshift-marketplace
```

c. Save your changes and exit. For example, if you are using `vi`, hit `esc` and enter `:wq`.

2. Create the following operator subscription.

Ensure that you update the **`namespace:`** *`ibm-common-services`*`|`*`cpd-operators`* entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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
```

3. Validate that the operator was successfully created.

For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **`ibm-common-services`** or **`cpd-operators`**) for the **`--namespace`** (**`-n`**) argument.

  a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-db2oltp-cp4d-operator-catalog-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

  Verify that the command returns **`ibm-db2oltp-cp4d-operator.v1.0.8`**.

  b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-db2oltp-cp4d-operator.v1.0.8 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

  Verify that the command returns **`Succeeded : install strategy completed with no errors`**.

  c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-db2oltp-cp4d-operator.v1.0.8" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

  Verify that the command returns an integer greater than or equal to **`1`**. If the command returns **`0`**, wait for the deployment to become available.

Important: If you created preceding operators for Db2 with the manual install plan (**`installPlanApproval: Manual`**), a cluster administrator must approve any pending update requests for the following operators:
- **`db2u-operator`** *
- **`ibm-db2oltp-cp4d-operator`**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Db2 Big SQL**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  ○ ⌐ › ⌐
    Private container registry

    1. Create the Db2U operator subscription.

    Ensure that you update the **`namespace:`** *`ibm-common-services`*`|`*`cpd-operators`* entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-db2u-operator
  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. Validate that the operator was successfully created.

    For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **`ibm-common-services`** or **`cpd-operators`**) for the **`--namespace`** (**`-n`**) argument.

      a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-db2u-operator \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

      Verify that the command returns **`db2u-operator.v1.1.11`**.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project db2u-operator.v1.1.11 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns **Succeeded : install strategy completed with no errors**.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="db2u-operator.v1.1.11" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

3. Create the Db2 Big SQL operator subscription.
Ensure that you update the **namespace:**
ibm-common-services|cpd-operators
entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v7.2
  installPlanApproval: Automatic
  name: ibm-bigsql-operator
  source: ibm-bigsql-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

4. Validate that the operator was successfully created.
For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the --namespace (**-n**) argument.

a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-bigsql-operator-catalog-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns **ibm-bigsql-operator.v7.2.6**.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-bigsql-operator.v7.2.6 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns **Succeeded : install strategy completed with no errors**.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-bigsql-operator.v7.2.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the preceding operators for Db2 Big SQL with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:

- **db2u-operator** *
- **ibm-bigsql-operator**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○
>
IBM Entitled Registry with catalog sources that pull specific versions of images

1. Create the Db2U operator subscription.
Ensure that you update the **namespace:**
ibm-common-services|cpd-operators
entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-db2u-operator
  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. Validate that the operator was successfully created.
For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the --namespace (**-n**) argument.

a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-db2u-operator \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns **db2u-operator.v1.1.11**.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project db2u-operator.v1.1.11 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns **Succeeded : install strategy completed with no errors**.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="db2u-operator.v1.1.11" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

3. Create the Db2 Big SQL operator subscription.
Ensure that you update the **namespace:**
ibm-common-services|cpd-operators
entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v7.2
  installPlanApproval: Automatic
  name: ibm-bigsql-operator
  source: ibm-bigsql-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

4. Validate that the operator was successfully created.
For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-bigsql-operator-catalog-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns **ibm-bigsql-operator.v7.2.6**.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-bigsql-operator.v7.2.6 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns **Succeeded : install strategy completed with no errors**.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-bigsql-operator.v7.2.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the preceding operators for Db2 Big SQL with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **db2u-operator** *
- **ibm-bigsql-operator**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

o
>
IBM Entitled Registry with a catalog source for the IBM Operator Catalog

1. Create the Db2U operator subscription.
Ensure that you update the **namespace:**
ibm-common-services|cpd-operators
entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-db2u-operator
  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. Validate that the operator was successfully created.

   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-db2u-operator \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **db2u-operator.v1.1.11**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project db2u-operator.v1.1.11 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="db2u-operator.v1.1.11" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

3. Create the Db2 Big SQL operator subscription.

   Ensure that you update the **namespace:**
   ibm-common-services|cpd-operators
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   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: v7.2
     installPlanApproval: Automatic
     name: ibm-bigsql-operator
     source: ibm-operator-catalog
     sourceNamespace: openshift-marketplace
   EOF
   ```

4. Validate that the operator was successfully created.

   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-bigsql-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-bigsql-operator.v7.2.6**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-bigsql-operator.v7.2.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-bigsql-operator.v7.2.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the preceding operators for Db2 Big SQL with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- **db2u-operator** *
- **ibm-bigsql-operator**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Db2 Data Gate**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - > Private container registry

    1. Create the following operator subscription.

       Ensure that you update the **namespace:**
       ibm-common-services|cpd-operators
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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|cpd-operators    # Pick the project that contains the Cloud Pak for Data
operator
spec:
    channel: v2.0
    installPlanApproval: Automatic
    name: ibm-datagate-operator
    source: ibm-datagate-operator-catalog
    sourceNamespace: openshift-marketplace
EOF
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-datagate-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-datagate-operator.v2.0.6`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-datagate-operator.v2.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-datagate-operator.v2.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Remember: If you created the Db2 Data Gate operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve the update request for `ibm-datagate-operator`.

o   >
   IBM Entitled Registry with catalog sources that pull specific versions of images

   1. Create the following operator subscription.
      Ensure that you update the `namespace:` `ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   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|cpd-operators    # Pick the project that contains the Cloud Pak for Data
   operator
   spec:
       channel: v2.0
       installPlanApproval: Automatic
       name: ibm-datagate-operator
       source: ibm-datagate-operator-catalog
       sourceNamespace: openshift-marketplace
   EOF
   ```

   2. Validate that the operator was successfully created.
      For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

      a. Run the following command to confirm that the subscription was triggered:

      ```
      oc get sub -n operator-project ibm-datagate-operator-subscription \
      -o jsonpath='{.status.installedCSV} {"\n"}'
      ```

      Verify that the command returns `ibm-datagate-operator.v2.0.6`.

      b. Run the following command to confirm that the cluster service version (CSV) is ready:

      ```
      oc get csv -n operator-project ibm-datagate-operator.v2.0.6 \
      -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
      ```

      Verify that the command returns `Succeeded : install strategy completed with no errors`.

      c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-datagate-operator.v2.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Remember: If you created the Db2 Data Gate operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve the update request for **ibm-datagate-operator**.

- > IBM Entitled Registry with a catalog source for the IBM Operator Catalog

  1. Create the following operator subscription.
     Ensure that you update the **namespace:**
     ibm-common-services|cpd-operators
     entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

     ```
     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|cpd-operators    # Pick the project that contains the Cloud Pak for Data
     operator
     spec:
         channel: v2.0
         installPlanApproval: Automatic
         name: ibm-datagate-operator
         source: ibm-operator-catalog
         sourceNamespace: openshift-marketplace
     EOF
     ```

  2. Validate that the operator was successfully created.
     For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

     a. Run the following command to confirm that the subscription was triggered:

        ```
        oc get sub -n operator-project ibm-datagate-operator-subscription \
        -o jsonpath='{.status.installedCSV} {"\n"}'
        ```

        Verify that the command returns **ibm-datagate-operator.v2.0.6**.

     b. Run the following command to confirm that the cluster service version (CSV) is ready:

        ```
        oc get csv -n operator-project ibm-datagate-operator.v2.0.6 \
        -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
        ```

        Verify that the command returns **Succeeded : install strategy completed with no errors**.

     c. Run the following command to confirm that the operator is ready:

        ```
        oc get deployments -n operator-project -l olm.owner="ibm-datagate-operator.v2.0.6" \
        -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
        ```

        Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

  Remember: If you created the Db2 Data Gate operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve the update request for **ibm-datagate-operator**.

- **Db2 Data Management Console**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - > Private container registry

    1. Create the following operator subscription.
       Ensure that you update the **namespace:**
       ibm-common-services|cpd-operators
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       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
       ```

    2. Validate that the operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-dmc-operator-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns **ibm-databases-dmc.v1.0.5**.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-databases-dmc.v1.0.5 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns **Succeeded : install strategy completed with no errors**.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-databases-dmc.v1.0.5" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the Db2 Data Management Console operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **ibm-dmc-operator**
- **ibm-cloud-databases-redis-operator** *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○ >
IBM Entitled Registry with catalog sources that pull specific versions of images

1. Create the following operator subscription.
Ensure that you update the **namespace:**
ibm-common-services|cpd-operators
entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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
```

2. Validate that the operator was successfully created.
For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-dmc-operator-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns **ibm-databases-dmc.v1.0.5**.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-databases-dmc.v1.0.5 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns **Succeeded : install strategy completed with no errors**.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-databases-dmc.v1.0.5" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the Db2 Data Management Console operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **ibm-dmc-operator**
- **ibm-cloud-databases-redis-operator** *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○ >
IBM Entitled Registry with a catalog source for the IBM Operator Catalog

1. Create the following operator subscription.
Ensure that you update the **namespace:**
ibm-common-services|cpd-operators
entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-dmc-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-databases-dmc.v1.0.5**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-databases-dmc.v1.0.5 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-databases-dmc.v1.0.5" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

   Important: If you created the Db2 Data Management Console operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
   - **ibm-dmc-operator**
   - **ibm-cloud-databases-redis-operator** *
   * An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **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. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - **> Private container registry**

    1. **Specialized installations only.** By default, IBM Cloud Pak foundational services expects that the Db2U operator (**ibm-db2u-operator**) is installed in the **ibm-common-services** project. However, if you are using the *specialized* installation architecture (where the IBM Cloud Pak foundational services operators and Cloud Pak for Data operators are in separate OpenShift projects), you must edit the IBM Cloud Pak foundational services operand registry to point to the project where the Cloud Pak for Data operators are installed:
       a. Run the following command to edit the IBM Cloud Pak foundational services operand registry.
          The following command assumes that the IBM Cloud Pak foundational services are installed in the **ibm-common-services** project.

          ```
          oc edit operandRegistry common-service \
          --namespace ibm-common-services
          ```

       b. Change the value of the **namespace** entry for the **ibm-db2u-operator**. Specify the OpenShift project where the Cloud Pak for Data operators are installed, for example, **cpd-operators**:

          ```
           - channel: v1.1
             installPlanApproval: Automatic
             name: ibm-db2u-operator
             namespace: cpd-operators      # Specify the project where the Cloud Pak for Data operators are installed
             packageName: db2u-operator
             scope: public
             sourceName: ibm-db2uoperator-catalog
             sourceNamespace: openshift-marketplace
          ```

       c. Save your changes and exit. For example, if you are using **vi**, hit esc and enter :wq.
    2. Create the following operator subscription.
       Ensure that you update the **namespace:**
       ibm-common-services|cpd-operators
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       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
```

3. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-db2wh-cp4d-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-db2wh-cp4d-operator.v1.0.8**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-db2wh-cp4d-operator.v1.0.8 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-db2wh-cp4d-operator.v1.0.8" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created preceding operators for Db2 Warehouse with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **db2u-operator** *
- **ibm-db2wh-cp4d-operator**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○ >
　IBM Entitled Registry with catalog sources that pull specific versions of images

1. **Specialized installations only.** By default, IBM Cloud Pak foundational services expects that the Db2U operator (**ibm-db2u-operator**) is installed in the **ibm-common-services** project. However, if you are using the *specialized* installation architecture (where the IBM Cloud Pak foundational services operators and Cloud Pak for Data operators are in separate OpenShift projects), you must edit the IBM Cloud Pak foundational services operand registry to point to the project where the Cloud Pak for Data operators are installed:
   a. Run the following command to edit the IBM Cloud Pak foundational services operand registry.
   The following command assumes that the IBM Cloud Pak foundational services are installed in the **ibm-common-services** project.

   ```
   oc edit operandRegistry common-service \
   --namespace ibm-common-services
   ```

   b. Change the value of the **namespace** entry for the **ibm-db2u-operator**. Specify the OpenShift project where the Cloud Pak for Data operators are installed, for example, **cpd-operators**:

   ```
   - channel: v1.1
       installPlanApproval: Automatic
       name: ibm-db2u-operator
       namespace: cpd-operators     # Specify the project where the Cloud Pak for Data operators are installed
       packageName: db2u-operator
       scope: public
       sourceName: ibm-db2uoperator-catalog
       sourceNamespace: openshift-marketplace
   ```

   c. Save your changes and exit. For example, if you are using **vi**, hit esc and enter :wq.
2. Create the following operator subscription.
   Ensure that you update the **namespace: ibm-common-services|cpd-operators** entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   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
   ```

3. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-db2wh-cp4d-operator-catalog-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns `ibm-db2wh-cp4d-operator.v1.0.8`.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-db2wh-cp4d-operator.v1.0.8 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns `Succeeded : install strategy completed with no errors`.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-db2wh-cp4d-operator.v1.0.8" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created preceding operators for Db2 Warehouse with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- `db2u-operator` *
- `ibm-db2wh-cp4d-operator`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○ | > |
  | IBM Entitled Registry with a catalog source for the IBM Operator Catalog |

1. **Specialized installations only.** By default, IBM Cloud Pak foundational services expects that the Db2U operator (`ibm-db2u-operator`) is installed in the `ibm-common-services` project. However, if you are using the *specialized* installation architecture (where the IBM Cloud Pak foundational services operators and Cloud Pak for Data operators are in separate OpenShift projects), you must edit the IBM Cloud Pak foundational services operand registry to point to the project where the Cloud Pak for Data operators are installed:

   a. Run the following command to edit the IBM Cloud Pak foundational services operand registry.
   The following command assumes that the IBM Cloud Pak foundational services are installed in the `ibm-common-services` project.

   ```
   oc edit operandRegistry common-service \
   --namespace ibm-common-services
   ```

   b. Change the value of the `namespace` entry for the `ibm-db2u-operator`. Specify the OpenShift project where the Cloud Pak for Data operators are installed, for example, `cpd-operators`:

   ```
    - channel: v1.1
       installPlanApproval: Automatic
       name: ibm-db2u-operator
       namespace: cpd-operators     # Specify the project where the Cloud Pak for Data operators are installed
       packageName: db2u-operator
       scope: public
       sourceName: ibm-db2uoperator-catalog
       sourceNamespace: openshift-marketplace
   ```

   c. Save your changes and exit. For example, if you are using `vi`, hit `esc` and enter `:wq`.

2. Create the following operator subscription.
   Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   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
   ```

3. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-db2wh-cp4d-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-db2wh-cp4d-operator.v1.0.8`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-db2wh-cp4d-operator.v1.0.8 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-db2wh-cp4d-operator.v1.0.8" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to 1. If the command returns 0, wait for the deployment to become available.

Important: If you created preceding operators for Db2 Warehouse with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- `db2u-operator` *
- `ibm-db2wh-cp4d-operator`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Decision Optimization**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - > Private container registry

    1. Create the following operator subscription.
       Ensure that you update the **namespace:**
       ibm-common-services|cpd-operators
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       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: v4.0
           installPlanApproval: Automatic
           name: ibm-cpd-dods
           source: ibm-cpd-dods-operator-catalog
           sourceNamespace: openshift-marketplace
       EOF
       ```

    2. Validate that the operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

       a. Run the following command to confirm that the subscription was triggered:

          ```
          oc get sub -n operator-project ibm-cpd-dods-operator-catalog-subscription \
          -o jsonpath='{.status.installedCSV} {"\n"}'
          ```

          Verify that the command returns **ibm-cpd-dods.v4.0.6**.

       b. Run the following command to confirm that the cluster service version (CSV) is ready:

          ```
          oc get csv -n operator-project ibm-cpd-dods.v4.0.6 \
          -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
          ```

          Verify that the command returns **Succeeded : install strategy completed with no errors**.

       c. Run the following command to confirm that the operator is ready:

          ```
          oc get deployments -n operator-project -l olm.owner="ibm-cpd-dods.v4.0.6" \
          -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
          ```

          Verify that the command returns an integer greater than or equal to 1. If the command returns 0, wait for the deployment to become available.

    Important: If you created the Decision Optimization operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
    - `ibm-cpd-ccs-operator` *
    - `ibm-cpd-dods`
    - `ibm-cpd-wml-operator` *
    - `ibm-cpd-wsl` *

    * An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

  - > IBM Entitled Registry with catalog sources that pull specific versions of images

    1. Create the following operator subscription.
       Ensure that you update the **namespace:**
       ibm-common-services|cpd-operators
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       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: v4.0
    installPlanApproval: Automatic
    name: ibm-cpd-dods
    source: ibm-cpd-dods-operator-catalog
    sourceNamespace: openshift-marketplace
EOF
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-cpd-dods-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-cpd-dods.v4.0.6**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-dods.v4.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-cpd-dods.v4.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the Decision Optimization operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **ibm-cpd-ccs-operator** *
- **ibm-cpd-dods**
- **ibm-cpd-wml-operator** *
- **ibm-cpd-wsl** *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○ >
   IBM Entitled Registry with a catalog source for the IBM Operator Catalog

   1. Create the following operator subscription.
      Ensure that you update the **namespace:**
      ibm-common-services|cpd-operators
      entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   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: v4.0
       installPlanApproval: Automatic
       name: ibm-cpd-dods
       source: ibm-operator-catalog
       sourceNamespace: openshift-marketplace
   EOF
   ```

   2. Validate that the operator was successfully created.
      For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

      a. Run the following command to confirm that the subscription was triggered:

      ```
      oc get sub -n operator-project ibm-cpd-dods-operator-catalog-subscription \
      -o jsonpath='{.status.installedCSV} {"\n"}'
      ```

      Verify that the command returns **ibm-cpd-dods.v4.0.6**.

      b. Run the following command to confirm that the cluster service version (CSV) is ready:

      ```
      oc get csv -n operator-project ibm-cpd-dods.v4.0.6 \
      -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
      ```

      Verify that the command returns **Succeeded : install strategy completed with no errors**.

      c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-dods.v4.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the Decision Optimization operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:

- **ibm-cpd-ccs-operator** *
- **ibm-cpd-dods**
- **ibm-cpd-wml-operator** *
- **ibm-cpd-wsl** *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **EDB Postgres**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - ⌄

    Private container registry

    1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the **ibm-common-services** project.
       Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.
       If you did not install IBM Cloud Pak foundational services in **ibm-common-services**, update the value in the **namespace: ibm-common-services** entry.

       ```
       cat <<EOF |oc apply -f -
       apiVersion: operator.ibm.com/v1alpha1
       kind: OperandRequest
       metadata:
         name: common-service-edb
         namespace: ibm-common-services
       spec:
         requests:
           - operands:
               - name: cloud-native-postgresql
             registry: common-service
       EOF
       ```

    2. Validate that the operator was successfully installed.
       a. Run the following command to confirm that the operand request was created:

          ```
          oc get opreq common-service-edb -n ibm-common-services \
          -o jsonpath='{.status.phase}  {"\n"}'
          ```

          Verify that the command returns **Running**. If the command returns **Initialized** or **Installing**, wait several minutes and run the command again.

       b. Run the following command to confirm that the operator is installed:

          ```
          oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
          -o jsonpath='{.items[0].status.phase} {"\n"}'
          ```

          Verify that the command returns **Running**.

    3. Create the EDB Postgres operator subscription.
       Ensure that you update the **namespace: ibm-common-services|cpd-operators** entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       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
       spec:
         installPlanApproval: Automatic
         channel: v4.0
         name: ibm-cpd-edb
         source: ibm-cpd-edb-operator-catalog
         sourceNamespace: openshift-marketplace
       EOF
       ```

    4. Validate that the operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

       a. Run the following command to confirm that the subscription was triggered:

          ```
          oc get sub -n operator-project ibm-cpd-edb-operator-catalog-subscription \
          -o jsonpath='{.status.installedCSV} {"\n"}'
          ```

          Verify that the command returns **ibm-cpd-edb.v4.0.6**.

       b. Run the following command to confirm that the cluster service version (CSV) is ready:

          ```
          oc get csv -n operator-project ibm-cpd-edb.v4.0.6 \
          -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
          ```

Verify that the command returns `Succeeded : install strategy completed with no errors`.

    c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-edb.v4.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for EDB Postgres with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `cloud-native-postgresql` *
- `ibm-cpd-edb`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○
>
IBM Entitled Registry with catalog sources that pull specific versions of images

1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the `ibm-common-services` project.
   Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.
   If you did not install IBM Cloud Pak foundational services in `ibm-common-services`, update the value in the `namespace: ibm-common-services` entry.

```
cat <<EOF |oc apply -f -
apiVersion: operator.ibm.com/v1alpha1
kind: OperandRequest
metadata:
  name: common-service-edb
  namespace: ibm-common-services
spec:
  requests:
    - operands:
        - name: cloud-native-postgresql
      registry: common-service
EOF
```

2. Validate that the operator was successfully installed.

    a. Run the following command to confirm that the operand request was created:

```
oc get opreq common-service-edb -n ibm-common-services \
-o jsonpath='{.status.phase}  {"\n"}'
```

Verify that the command returns `Running`. If the command returns `Initialized` or `Installing`, wait several minutes and run the command again.

    b. Run the following command to confirm that the operator is installed:

```
oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
-o jsonpath='{.items[0].status.phase} {"\n"}'
```

Verify that the command returns `Running`.

3. Create the EDB Postgres operator subscription.
   Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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
spec:
  installPlanApproval: Automatic
  channel: v4.0
  name: ibm-cpd-edb
  source: ibm-cpd-edb-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

4. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

    a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-cpd-edb-operator-catalog-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns `ibm-cpd-edb.v4.0.6`.

    b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-cpd-edb.v4.0.6 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns `Succeeded : install strategy completed with no errors`.

    c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-edb.v4.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to 1. If the command returns 0, wait for the deployment to become available.

Important: If you created the preceding operators for EDB Postgres with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `cloud-native-postgresql` *
- `ibm-cpd-edb`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○ >
  IBM Entitled Registry with a catalog source for the IBM Operator Catalog

1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the `ibm-common-services` project.

   Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.

   If you did not install IBM Cloud Pak foundational services in `ibm-common-services`, update the value in the `namespace: ibm-common-services` entry.

   ```
   cat <<EOF |oc apply -f -
   apiVersion: operator.ibm.com/v1alpha1
   kind: OperandRequest
   metadata:
     name: common-service-edb
     namespace: ibm-common-services
   spec:
     requests:
       - operands:
           - name: cloud-native-postgresql
         registry: common-service
   EOF
   ```

2. Validate that the operator was successfully installed.

   a. Run the following command to confirm that the operand request was created:

      ```
      oc get opreq common-service-edb -n ibm-common-services \
      -o jsonpath='{.status.phase}  {"\n"}'
      ```

      Verify that the command returns `Running`. If the command returns `Initialized` or `Installing`, wait several minutes and run the command again.

   b. Run the following command to confirm that the operator is installed:

      ```
      oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
      -o jsonpath='{.items[0].status.phase} {"\n"}'
      ```

      Verify that the command returns `Running`.

3. Create the EDB Postgres operator subscription.

   Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   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
   spec:
     installPlanApproval: Automatic
     channel: v4.0
     name: ibm-cpd-edb
     source: ibm-operator-catalog
     sourceNamespace: openshift-marketplace
   EOF
   ```

4. Validate that the operator was successfully created.

   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

      ```
      oc get sub -n operator-project ibm-cpd-edb-operator-catalog-subscription \
      -o jsonpath='{.status.installedCSV} {"\n"}'
      ```

      Verify that the command returns `ibm-cpd-edb.v4.0.6`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

      ```
      oc get csv -n operator-project ibm-cpd-edb.v4.0.6 \
      -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
      ```

      Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

      ```
      oc get deployments -n operator-project -l olm.owner="ibm-cpd-edb.v4.0.6" \
      -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
      ```

Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for EDB Postgres with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- `cloud-native-postgresql` *
- `ibm-cpd-edb`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Execution Engine for Apache Hadoop**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - > **Private container registry**

    1. Create the following operator subscription.
       Ensure that you update the `namespace:` `ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       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
       ```

    2. Validate that the operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

       a. Run the following command to confirm that the subscription was triggered:

       ```
       oc get sub -n operator-project ibm-cpd-hadoop-operator-catalog-subscription \
       -o jsonpath='{.status.installedCSV} {"\n"}'
       ```

       Verify that the command returns `ibm-cpd-hadoop.v1.0.6`.

       b. Run the following command to confirm that the cluster service version (CSV) is ready:

       ```
       oc get csv -n operator-project ibm-cpd-hadoop.v1.0.6 \
       -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
       ```

       Verify that the command returns `Succeeded : install strategy completed with no errors`.

       c. Run the following command to confirm that the operator is ready:

       ```
       oc get deployments -n operator-project -l olm.owner="ibm-cpd-hadoop.v1.0.6" \
       -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
       ```

       Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

    Important: If you created the Execution Engine for Apache Hadoop operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
    - `ibm-cpd-hadoop`
    - `ibm-cpd-wsl` *

    * An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

  - > **IBM Entitled Registry with catalog sources that pull specific versions of images**

    1. Create the following operator subscription.
       Ensure that you update the `namespace:` `ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       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
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-cpd-hadoop-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-cpd-hadoop.v1.0.6**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-hadoop.v1.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-cpd-hadoop.v1.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the Execution Engine for Apache Hadoop operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **ibm-cpd-hadoop**
- **ibm-cpd-wsl** *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○ > IBM Entitled Registry with a catalog source for the IBM Operator Catalog

1. Create the following operator subscription.
   Ensure that you update the **namespace: ibm-common-services|cpd-operators** entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   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
   ```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-cpd-hadoop-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-cpd-hadoop.v1.0.6**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-hadoop.v1.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-cpd-hadoop.v1.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the Execution Engine for Apache Hadoop operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `ibm-cpd-hadoop`
- `ibm-cpd-wsl` *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Financial Services Workbench**

  Not applicable. For details, see the [Financial Services Workbench documentation](#).

- **IBM Match 360 with Watson**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - ⌄ Private container registry

    1. Create the following operator subscription.
       Ensure that you update the `namespace:`
       `ibm-common-services|cpd-operators`
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

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

    2. Validate that the operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

       a. Run the following command to confirm that the subscription was triggered:

          ```
          oc get sub -n operator-project ibm-mdm-operator-subscription \
          -o jsonpath='{.status.installedCSV} {"\n"}'
          ```

          Verify that the command returns `ibm-mdm.v1.1.188`.

       b. Run the following command to confirm that the cluster service version (CSV) is ready:

          ```
          oc get csv -n operator-project ibm-mdm.v1.1.188 \
          -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
          ```

          Verify that the command returns `Succeeded : install strategy completed with no errors`.

       c. Run the following command to confirm that the operator is ready:

          ```
          oc get deployments -n operator-project -l olm.owner="ibm-mdm.v1.1.188" \
          -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
          ```

          Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

    Important: If you created the IBM Match 360 with Watson operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

    - `fdb-kubernetes-operator` *
    - `ibm-cloud-databases-redis-operator` *
    - `ibm-cpd-ccs-operator` *
    - `ibm-elasticsearch-operator` *
    - `ibm-mdm`
    - `ibm-opencontent-foundationdb` *
    - `ibm-rabbitmq-operator` *

    * An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

  - ⌄ IBM Entitled Registry with catalog sources that pull specific versions of images

    1. Create the following operator subscription.
       Ensure that you update the `namespace:`
       `ibm-common-services|cpd-operators`
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       cat <<EOF |oc apply -f -
       apiVersion: operators.coreos.com/v1alpha1
       kind: Subscription
       ```

```
metadata:
  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
  name: ibm-mdm-operator-subscription
  namespace: ibm-common-services|cpd-operators      # Pick the project that contains the Cloud Pak for Data
operator
spec:
  channel: v1.1
  installPlanApproval: Automatic
  name: ibm-mdm
  source: ibm-mdm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-mdm-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-mdm.v1.1.188`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-mdm.v1.1.188 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-mdm.v1.1.188" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the IBM Match 360 with Watson operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `fdb-kubernetes-operator` *
- `ibm-cloud-databases-redis-operator` *
- `ibm-cpd-ccs-operator` *
- `ibm-elasticsearch-operator` *
- `ibm-mdm`
- `ibm-opencontent-foundationdb` *
- `ibm-rabbitmq-operator` *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

o
>
[IBM Entitled Registry with a catalog source for the IBM Operator Catalog]

1. Create the following operator subscription.
   Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

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

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-mdm-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-mdm.v1.1.188`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-mdm.v1.1.188 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns `Succeeded : install strategy completed with no errors`.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-mdm.v1.1.188" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the IBM Match 360 with Watson operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `fdb-kubernetes-operator` *
- `ibm-cloud-databases-redis-operator` *
- `ibm-cpd-ccs-operator` *
- `ibm-elasticsearch-operator` *
- `ibm-mdm`
- `ibm-opencontent-foundationdb` *
- `ibm-rabbitmq-operator` *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Informix®**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - > Private container registry

    1. Create the following operator subscription for the install operator.
       Ensure that you update the `namespace:`
       `ibm-common-services|cpd-operators`
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       cat <<EOF |oc apply -f -
       apiVersion: operators.coreos.com/v1alpha1
       kind: Subscription
       metadata:
         name: ibm-informix-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-informix-operator
         source: ibm-informix-operator-catalog
         sourceNamespace: openshift-marketplace
       EOF
       ```

    2. Validate that the install operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

       a. Run the following command to confirm that the subscription was triggered:

       ```
       oc get sub -n operator-project ibm-informix-operator-catalog-subscription \
       -o jsonpath='{.status.installedCSV} {"\n"}'
       ```

       Verify that the command returns `ibm-informix-operator.v4.1.0`.

       b. Run the following command to confirm that the cluster service version (CSV) is ready:

       ```
       oc get csv -n operator-project ibm-informix-operator.v4.1.0 \
       -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
       ```

       Verify that the command returns `Succeeded : install strategy completed with no errors`.

       c. Run the following command to confirm that the operator is ready:

       ```
       oc get deployments -n operator-project -l olm.owner="ibm-informix-operator.v4.1.0" \
       -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
       ```

       Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

    3. Create the following operator subscription for the deployment operator.
       Ensure that you update the `namespace:`
       `ibm-common-services|cpd-operators`
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       cat <<EOF |oc apply -f -
       apiVersion: operators.coreos.com/v1alpha1
       kind: Subscription
       metadata:
         name: ibm-informix-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
         installPlanApproval: Automatic
       ```

```
  name: ibm-informix-cp4d-operator
  source: ibm-informix-cp4d-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

4. Validate that the deployment operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-informix-cp4d-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-informix-cp4d-operator.v4.1.0**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-informix-cp4d-operator.v4.1.0 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-informix-cp4d-operator.v4.1.0" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the preceding operators for Informix with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **ibm-informix-operator**
- **ibm-informix-cp4d-operator**

○ >
IBM Entitled Registry with catalog sources that pull specific versions of images

1. Create the following operator subscription for the install operator.
   Ensure that you update the **namespace:**
   ibm-common-services|cpd-operators
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   cat <<EOF |oc apply -f -
   apiVersion: operators.coreos.com/v1alpha1
   kind: Subscription
   metadata:
     name: ibm-informix-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-informix-operator
     source: ibm-informix-operator-catalog
     sourceNamespace: openshift-marketplace
   EOF
   ```

2. Validate that the install operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-informix-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-informix-operator.v4.1.0**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-informix-operator.v4.1.0 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-informix-operator.v4.1.0" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

3. Create the following operator subscription for the deployment operator.
   Ensure that you update the **namespace:**
   ibm-common-services|cpd-operators
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   cat <<EOF |oc apply -f -
   apiVersion: operators.coreos.com/v1alpha1
   kind: Subscription
   metadata:
     name: ibm-informix-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
  installPlanApproval: Automatic
  name: ibm-informix-cp4d-operator
  source: ibm-informix-cp4d-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

4. Validate that the deployment operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-informix-cp4d-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-informix-cp4d-operator.v4.1.0**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-informix-cp4d-operator.v4.1.0 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-informix-cp4d-operator.v4.1.0" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

   Important: If you created the preceding operators for Informix with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
   - **ibm-informix-operator**
   - **ibm-informix-cp4d-operator**

   ○ ❯ IBM Entitled Registry with a catalog source for the IBM Operator Catalog

   1. Create the following operator subscription for the install operator.
      Ensure that you update the **namespace: ibm-common-services|cpd-operators** entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

      ```
      cat <<EOF |oc apply -f -
      apiVersion: operators.coreos.com/v1alpha1
      kind: Subscription
      metadata:
        name: ibm-informix-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-informix-operator
        source: ibm-operator-catalog
        sourceNamespace: openshift-marketplace
      EOF
      ```

   2. Validate that the install operator was successfully created.
      For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

      a. Run the following command to confirm that the subscription was triggered:

      ```
      oc get sub -n operator-project ibm-informix-operator-catalog-subscription \
      -o jsonpath='{.status.installedCSV} {"\n"}'
      ```

      Verify that the command returns **ibm-informix-operator.v4.1.0**.

      b. Run the following command to confirm that the cluster service version (CSV) is ready:

      ```
      oc get csv -n operator-project ibm-informix-operator.v4.1.0 \
      -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
      ```

      Verify that the command returns **Succeeded : install strategy completed with no errors**.

      c. Run the following command to confirm that the operator is ready:

      ```
      oc get deployments -n operator-project -l olm.owner="ibm-informix-operator.v4.1.0" \
      -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
      ```

      Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

   3. Create the following operator subscription for the deployment operator.
      Ensure that you update the **namespace: ibm-common-services|cpd-operators** entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

      ```
      cat <<EOF |oc apply -f -
      apiVersion: operators.coreos.com/v1alpha1
      ```

```
kind: Subscription
metadata:
  name: ibm-informix-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
  installPlanApproval: Automatic
  name: ibm-informix-cp4d-operator
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

4. Validate that the deployment operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-informix-cp4d-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-informix-cp4d-operator.v4.1.0**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-informix-cp4d-operator.v4.1.0 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-informix-cp4d-operator.v4.1.0" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the preceding operators for Informix with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
   ▪ **ibm-informix-operator**
   ▪ **ibm-informix-cp4d-operator**

- **MongoDB**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  ○ 〉 Private container registry

    1. Create the MongoDB Enterprise (third-party) operator subscription.
       Ensure that you update the **namespace: ibm-common-services|cpd-operators**
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       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. Validate that the operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

       a. Run the following command to confirm that the subscription was triggered:

       ```
       oc get sub -n operator-project ibm-mongodb-enterprise-catalog-subscription \
       -o jsonpath='{.status.installedCSV} {"\n"}'
       ```

       Verify that the command returns **mongodb-enterprise.v1.13.0**.

       b. Run the following command to confirm that the cluster service version (CSV) is ready:

       ```
       oc get csv -n operator-project mongodb-enterprise.v1.13.0 \
       -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
       ```

       Verify that the command returns **Succeeded : install strategy completed with no errors**.

       c. Run the following command to confirm that the operator is ready:

       ```
       oc get deployments -n operator-project -l olm.owner="mongodb-enterprise.v1.13.0" \
       -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
       ```

Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

3. Create the MongoDB (Cloud Pak for Data) operator subscription.
Ensure that you update the `namespace:`
`ibm-common-services|cpd-operators`
entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v4.0
  name: ibm-cpd-mongodb
  installPlanApproval: Automatic
  source: ibm-cpd-mongodb-catalog
  sourceNamespace: openshift-marketplace
EOF
```

4. Validate that the operator was successfully created.
For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-cpd-mongodb-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-cpd-mongodb.v4.0.4`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-mongodb.v4.0.4 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-cpd-mongodb.v4.0.4" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for MongoDB with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- `mongodb-enterprise`
- `ibm-cpd-mongodb`

- > IBM Entitled Registry with catalog sources that pull specific versions of images

  1. Create the MongoDB Enterprise (third-party) operator subscription.
  Ensure that you update the `namespace:`
  `ibm-common-services|cpd-operators`
  entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

  ```
  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. Validate that the operator was successfully created.
  For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

     a. Run the following command to confirm that the subscription was triggered:

     ```
     oc get sub -n operator-project ibm-mongodb-enterprise-catalog-subscription \
     -o jsonpath='{.status.installedCSV} {"\n"}'
     ```

     Verify that the command returns `mongodb-enterprise.v1.13.0`.

     b. Run the following command to confirm that the cluster service version (CSV) is ready:

     ```
     oc get csv -n operator-project mongodb-enterprise.v1.13.0 \
     -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
     ```

     Verify that the command returns `Succeeded : install strategy completed with no errors`.

     c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="mongodb-enterprise.v1.13.0" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to 1. If the command returns 0, wait for the deployment to become available.

3. Create the MongoDB (Cloud Pak for Data) operator subscription.
   Ensure that you update the `namespace:`
   `ibm-common-services|cpd-operators`
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v4.0
  name: ibm-cpd-mongodb
  installPlanApproval: Automatic
  source: ibm-cpd-mongodb-catalog
  sourceNamespace: openshift-marketplace
EOF
```

4. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-cpd-mongodb-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-cpd-mongodb.v4.0.4`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-mongodb.v4.0.4 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-cpd-mongodb.v4.0.4" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to 1. If the command returns 0, wait for the deployment to become available.

Important: If you created the preceding operators for MongoDB with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- **mongodb-enterprise**
- **ibm-cpd-mongodb**

○
　　　　　　　　　 ＞
　IBM Entitled Registry with a catalog source for the IBM Operator Catalog

1. Create the MongoDB Enterprise (third-party) operator subscription.
   Ensure that you update the `namespace:`
   `ibm-common-services|cpd-operators`
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-mongodb-enterprise-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `mongodb-enterprise.v1.13.0`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project mongodb-enterprise.v1.13.0 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

Verify that the command returns `Succeeded : install strategy completed with no errors`.

    c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="mongodb-enterprise.v1.13.0" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

3. Create the MongoDB (Cloud Pak for Data) operator subscription.
Ensure that you update the `namespace: ibm-common-services|cpd-operators`
entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v4.0
  name: ibm-cpd-mongodb
  installPlanApproval: Automatic
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

4. Validate that the operator was successfully created.
For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

    a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-cpd-mongodb-catalog-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns `ibm-cpd-mongodb.v4.0.4`.

    b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-cpd-mongodb.v4.0.4 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns `Succeeded : install strategy completed with no errors`.

    c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-mongodb.v4.0.4" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for MongoDB with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- `mongodb-enterprise`
- `ibm-cpd-mongodb`

- **OpenPages®**

Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

    ◦   ❭   Private container registry

        1. Create the following operator subscription.
Ensure that you update the `namespace: ibm-common-services|cpd-operators`
entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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
```

        2. Validate that the operator was successfully created.
For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

            a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-cpd-openpages-operator \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns **ibm-cpd-openpages-operator.v8.204.3**.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-cpd-openpages-operator.v8.204.3 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns **Succeeded : install strategy completed with no errors**.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-openpages-operator.v8.204.3" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the OpenPages operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **ibm-db2aaservice-cp4d-operator** *
- **db2u-operator** *
- **ibm-cpd-openpages-operator**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

    >
    IBM Entitled Registry with catalog sources that pull specific versions of images

1. Create the following operator subscription.
   Ensure that you update the **namespace:**
   ibm-common-services|cpd-operators
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-cpd-openpages-operator \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns **ibm-cpd-openpages-operator.v8.204.3**.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-cpd-openpages-operator.v8.204.3 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns **Succeeded : install strategy completed with no errors**.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-openpages-operator.v8.204.3" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the OpenPages operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **ibm-db2aaservice-cp4d-operator** *
- **db2u-operator** *
- **ibm-cpd-openpages-operator**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

    >
    IBM Entitled Registry with a catalog source for the IBM Operator Catalog

1. Create the following operator subscription.
   Ensure that you update the **namespace:**
   ibm-common-services|cpd-operators
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-cpd-openpages-operator \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-cpd-openpages-operator.v8.204.3**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-openpages-operator.v8.204.3 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-cpd-openpages-operator.v8.204.3" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the OpenPages operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **ibm-db2aaservice-cp4d-operator** *
- **db2u-operator** *
- **ibm-cpd-openpages-operator**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Planning Analytics**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  o [> Private container registry]

  1. Create the following operator subscription.
     Ensure that you update the **namespace:**
     ibm-common-services|cpd-operators
     entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

     ```
     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: v4.0
       name: ibm-planning-analytics-operator
       installPlanApproval: Automatic
       source: ibm-planning-analytics-operator-catalog
       sourceNamespace: openshift-marketplace
     EOF
     ```

  2. Validate that the operator was successfully created.
     For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

     a. Run the following command to confirm that the subscription was triggered:

     ```
     oc get sub -n operator-project ibm-planning-analytics-subscription \
     -o jsonpath='{.status.installedCSV} {"\n"}'
     ```

     Verify that the command returns **ibm-planning-analytics-operator.v4.0.6**.

     b. Run the following command to confirm that the cluster service version (CSV) is ready:

     ```
     oc get csv -n operator-project ibm-planning-analytics-operator.v4.0.6 \
     -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
     ```

     Verify that the command returns **Succeeded : install strategy completed with no errors**.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-planning-analytics-operator.v4.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Remember: If you created the Planning Analytics operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve the update request for `ibm-planning-analytics-operator`.

- ○ 〉
  IBM Entitled Registry with catalog sources that pull specific versions of images

  1. Create the following operator subscription.
     Ensure that you update the `namespace:`
     `ibm-common-services|cpd-operators`
     entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

     ```
     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: v4.0
       name: ibm-planning-analytics-operator
       installPlanApproval: Automatic
       source: ibm-planning-analytics-operator-catalog
       sourceNamespace: openshift-marketplace
     EOF
     ```

  2. Validate that the operator was successfully created.
     For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

     a. Run the following command to confirm that the subscription was triggered:

     ```
     oc get sub -n operator-project ibm-planning-analytics-subscription \
     -o jsonpath='{.status.installedCSV} {"\n"}'
     ```

     Verify that the command returns `ibm-planning-analytics-operator.v4.0.6`.

     b. Run the following command to confirm that the cluster service version (CSV) is ready:

     ```
     oc get csv -n operator-project ibm-planning-analytics-operator.v4.0.6 \
     -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
     ```

     Verify that the command returns `Succeeded : install strategy completed with no errors`.

     c. Run the following command to confirm that the operator is ready:

     ```
     oc get deployments -n operator-project -l olm.owner="ibm-planning-analytics-operator.v4.0.6" \
     -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
     ```

     Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Remember: If you created the Planning Analytics operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve the update request for `ibm-planning-analytics-operator`.

- ○ 〉
  IBM Entitled Registry with a catalog source for the IBM Operator Catalog

  1. Create the following operator subscription.
     Ensure that you update the `namespace:`
     `ibm-common-services|cpd-operators`
     entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

     ```
     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: v4.0
       name: ibm-planning-analytics-operator
       installPlanApproval: Automatic
       source: ibm-operator-catalog
       sourceNamespace: openshift-marketplace
     EOF
     ```

  2. Validate that the operator was successfully created.
     For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

     a. Run the following command to confirm that the subscription was triggered:

     ```
     oc get sub -n operator-project ibm-planning-analytics-subscription \
     -o jsonpath='{.status.installedCSV} {"\n"}'
     ```

     Verify that the command returns `ibm-planning-analytics-operator.v4.0.6`.

     b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-planning-analytics-operator.v4.0.6 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns `Succeeded : install strategy completed with no errors`.

    c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-planning-analytics-operator.v4.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Remember: If you created the Planning Analytics operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve the update request for `ibm-planning-analytics-operator`.

- **Product Master**

Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

   ○ | **＞** |
      Private container registry

      1. Create the following operator subscription.
        Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v1.0
  name: ibm-cpd-productmaster
  installPlanApproval: Automatic
  source: ibm-productmaster-catalog
  sourceNamespace: openshift-marketplace
EOF
```

      2. Validate that the operator was successfully created.
        For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

        a. Run the following command to confirm that the subscription was triggered:

```
oc get sub --namespace operator-project ibm-productmaster-catalog-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

        Verify that the command returns `ibm-cpd-productmaster.v1.0.3`.

        b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-cpd-productmaster.v1.0.3 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

        Verify that the command returns `Succeeded : install strategy completed with no errors`.

        c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-productmaster.v1.0.3" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

        Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Remember: If you created the Product Master operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve the update request for `ibm-cpd-productmaster`.

   ○ | **＞** |
      IBM Entitled Registry with catalog sources that pull specific versions of images

      1. Create the following operator subscription.
        Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v1.0
  name: ibm-cpd-productmaster
  installPlanApproval: Automatic
  source: ibm-productmaster-catalog
  sourceNamespace: openshift-marketplace
EOF
```

      2. Validate that the operator was successfully created.

For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

    a. Run the following command to confirm that the subscription was triggered:

```
oc get sub --namespace operator-project ibm-productmaster-catalog-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

    Verify that the command returns `ibm-cpd-productmaster.v1.0.3`.

    b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-cpd-productmaster.v1.0.3 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

    Verify that the command returns `Succeeded : install strategy completed with no errors`.

    c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-productmaster.v1.0.3" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

    Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Remember: If you created the Product Master operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve the update request for `ibm-cpd-productmaster`.

○   ›   IBM Entitled Registry with a catalog source for the IBM Operator Catalog

1. Create the following operator subscription.
   Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v1.0
  name: ibm-cpd-productmaster
  installPlanApproval: Automatic
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

    a. Run the following command to confirm that the subscription was triggered:

```
oc get sub --namespace operator-project ibm-productmaster-catalog-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

    Verify that the command returns `ibm-cpd-productmaster.v1.0.3`.

    b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-cpd-productmaster.v1.0.3 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

    Verify that the command returns `Succeeded : install strategy completed with no errors`.

    c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-productmaster.v1.0.3" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

    Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Remember: If you created the Product Master operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve the update request for `ibm-cpd-productmaster`.

- **RStudio® Server with R 3.6**

Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

○   ›   Private container registry

1. Create the following operator subscription.
   Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-cpd-rstudio-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-cpd-rstudio.v1.0.6**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-rstudio.v1.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-cpd-rstudio.v1.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the RStudio Server with R 3.6 operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **ibm-cpd-ccs-operator** *
- **ibm-cpd-rstudio**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○ > IBM Entitled Registry with catalog sources that pull specific versions of images

   1. Create the following operator subscription.
      Ensure that you update the **namespace: ibm-common-services|cpd-operators**
      entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   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
   ```

   2. Validate that the operator was successfully created.
      For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

      a. Run the following command to confirm that the subscription was triggered:

      ```
      oc get sub -n operator-project ibm-cpd-rstudio-operator-catalog-subscription \
      -o jsonpath='{.status.installedCSV} {"\n"}'
      ```

      Verify that the command returns **ibm-cpd-rstudio.v1.0.6**.

      b. Run the following command to confirm that the cluster service version (CSV) is ready:

      ```
      oc get csv -n operator-project ibm-cpd-rstudio.v1.0.6 \
      -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
      ```

      Verify that the command returns **Succeeded : install strategy completed with no errors**.

      c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-rstudio.v1.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the RStudio Server with R 3.6 operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `ibm-cpd-ccs-operator` *
- `ibm-cpd-rstudio`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○ 〉 IBM Entitled Registry with a catalog source for the IBM Operator Catalog

1. Create the following operator subscription.

   Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   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
   ```

2. Validate that the operator was successfully created.

   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-cpd-rstudio-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-cpd-rstudio.v1.0.6`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-rstudio.v1.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-cpd-rstudio.v1.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the RStudio Server with R 3.6 operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `ibm-cpd-ccs-operator` *
- `ibm-cpd-rstudio`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **SPSS® Modeler**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  ○ 〉 Private container registry

  1. Create the following operator subscription.

     Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

     ```
     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
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

       a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n ibm-common-services ibm-cpd-spss-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

       Verify that the command returns **ibm-cpd-spss.v1.0.6**.

       b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n ibm-common-services ibm-cpd-spss.v1.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

       Verify that the command returns **Succeeded : install strategy completed with no errors**.

       c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n ibm-common-services -l olm.owner="ibm-cpd-spss.v1.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

       Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the SPSS Modeler operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **ibm-cpd-spss**
- **ibm-cpd-wsl** *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

    ◦   ❯

> IBM Entitled Registry with catalog sources that pull specific versions of images

1. Create the following operator subscription.
   Ensure that you update the **namespace: ibm-common-services|cpd-operators**
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   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
   ```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

       a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n ibm-common-services ibm-cpd-spss-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

       Verify that the command returns **ibm-cpd-spss.v1.0.6**.

       b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n ibm-common-services ibm-cpd-spss.v1.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

       Verify that the command returns **Succeeded : install strategy completed with no errors**.

       c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n ibm-common-services -l olm.owner="ibm-cpd-spss.v1.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the SPSS Modeler operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `ibm-cpd-spss`
- `ibm-cpd-wsl` *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- > IBM Entitled Registry with a catalog source for the IBM Operator Catalog

    1. Create the following operator subscription.
       Ensure that you update the `namespace:`
       `ibm-common-services|cpd-operators`
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       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
       ```

    2. Validate that the operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

        a. Run the following command to confirm that the subscription was triggered:

           ```
           oc get sub -n ibm-common-services ibm-cpd-spss-operator-catalog-subscription \
           -o jsonpath='{.status.installedCSV} {"\n"}'
           ```

           Verify that the command returns `ibm-cpd-spss.v1.0.6`.

        b. Run the following command to confirm that the cluster service version (CSV) is ready:

           ```
           oc get csv -n ibm-common-services ibm-cpd-spss.v1.0.6 \
           -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
           ```

           Verify that the command returns `Succeeded : install strategy completed with no errors`.

        c. Run the following command to confirm that the operator is ready:

           ```
           oc get deployments -n ibm-common-services -l olm.owner="ibm-cpd-spss.v1.0.6" \
           -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
           ```

           Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

    Important: If you created the SPSS Modeler operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

    - `ibm-cpd-spss`
    - `ibm-cpd-wsl` *

    * An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Virtual Data Pipeline**

  Not applicable. For details, see [Installing Virtual Data Pipeline for Cloud Pak for Data](#).

- **Voice Gateway**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

    - > Private container registry

        1. Create the following operator subscription.
           Ensure that you update the `namespace:`
           `ibm-common-services|cpd-operators`
           entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

           ```
           cat <<EOF |oc apply -f -
           apiVersion: operators.coreos.com/v1alpha1
           kind: Subscription
           metadata:
             name: ibm-voice-gateway-operator-subscription
             namespace: ibm-common-services|cpd-operators     # Pick the project that contains the Cloud Pak for Data
           operator
           ```

```
spec:
  channel: v1.0
  name: ibm-voice-gateway-operator
  source: ibm-voice-gateway-operator-catalog
  sourceNamespace: openshift-marketplace
  installPlanApproval: Automatic
EOF
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-voice-gateway-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-voice-gateway-operator.v1.0.5`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-voice-gateway-operator.v1.0.5 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-voice-gateway-operator.v1.0.5" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Remember: If you created the Voice Gateway operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve the update request for `ibm-voice-gateway-operator`.

○ >
   [IBM Entitled Registry with catalog sources that pull specific versions of images]

   1. Create the following operator subscription.
      Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

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

   2. Validate that the operator was successfully created.
      For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

      a. Run the following command to confirm that the subscription was triggered:

      ```
      oc get sub -n operator-project ibm-voice-gateway-operator-subscription \
      -o jsonpath='{.status.installedCSV} {"\n"}'
      ```

      Verify that the command returns `ibm-voice-gateway-operator.v1.0.5`.

      b. Run the following command to confirm that the cluster service version (CSV) is ready:

      ```
      oc get csv -n operator-project ibm-voice-gateway-operator.v1.0.5 \
      -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
      ```

      Verify that the command returns `Succeeded : install strategy completed with no errors`.

      c. Run the following command to confirm that the operator is ready:

      ```
      oc get deployments -n operator-project -l olm.owner="ibm-voice-gateway-operator.v1.0.5" \
      -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
      ```

      Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Remember: If you created the Voice Gateway operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve the update request for `ibm-voice-gateway-operator`.

○ >
   [IBM Entitled Registry with a catalog source for the IBM Operator Catalog]

   1. Create the following operator subscription.
      Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

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

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-voice-gateway-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-voice-gateway-operator.v1.0.5**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-voice-gateway-operator.v1.0.5 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-voice-gateway-operator.v1.0.5" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

   Remember: If you created the Voice Gateway operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve the update request for **ibm-voice-gateway-operator**.

- **Watson Assistant**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  ○ > Private container registry

    1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the **ibm-common-services** project.
       Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.
       If you did not install IBM Cloud Pak foundational services in **ibm-common-services**, update the value in the **namespace: ibm-common-services** entry.

       ```
       cat <<EOF |oc apply -f -
       apiVersion: operator.ibm.com/v1alpha1
       kind: OperandRequest
       metadata:
         name: common-service-edb
         namespace: ibm-common-services
       spec:
         requests:
           - operands:
               - name: cloud-native-postgresql
             registry: common-service
       EOF
       ```

    2. Validate that the operator was successfully installed.
       a. Run the following command to confirm that the operand request was created:

       ```
       oc get opreq common-service-edb -n ibm-common-services \
       -o jsonpath='{.status.phase}  {"\n"}'
       ```

       Verify that the command returns **Running**. If the command returns **Initialized** or **Installing**, wait several minutes and run the command again.

       b. Run the following command to confirm that the operator is installed:

       ```
       oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
       -o jsonpath='{.items[0].status.phase} {"\n"}'
       ```

       Verify that the command returns **Running**.

    3. Create the following operator subscription.
       Ensure that you update the **namespace: ibm-common-services|cpd-operators** entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       cat <<EOF |oc apply -f -
       apiVersion: operators.coreos.com/v1alpha1
       kind: Subscription
       ```

```
metadata:
  name: ibm-watson-assistant-operator-subscription
  namespace: ibm-common-services|cpd-operators     # Pick the project that contains the Cloud Pak for Data
operator
spec:
  channel: v4.0
  name: ibm-watson-assistant-operator
  source: ibm-watson-assistant-operator-catalog
  sourceNamespace: openshift-marketplace
  installPlanApproval: Automatic
EOF
```

4. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-watson-assistant-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-watson-assistant-operator.v4.0.6**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-watson-assistant-operator.v4.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-watson-assistant-operator.v4.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the preceding operators for Watson Assistant with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **cloud-native-postgresql** *
- **cp4d-audit-webhook-operator**
- **ibm-cloud-databases-redis-operator** *
- **ibm-elasticsearch-operator** *
- **ibm-etcd-operator** *
- **ibm-minio-operator** *
- **ibm-model-train-dwf-operator**
- **ibm-rabbitmq-operator** *
- **ibm-watson-assistant-operator**
- **ibm-watson-gateway-operator** *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- ○ >
   IBM Entitled Registry with catalog sources that pull specific versions of images

   1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the **ibm-common-services** project.
      Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.
      If you did not install IBM Cloud Pak foundational services in **ibm-common-services**, update the value in the **namespace: ibm-common-services** entry.

      ```
      cat <<EOF |oc apply -f -
      apiVersion: operator.ibm.com/v1alpha1
      kind: OperandRequest
      metadata:
        name: common-service-edb
        namespace: ibm-common-services
      spec:
        requests:
          - operands:
              - name: cloud-native-postgresql
            registry: common-service
      EOF
      ```

   2. Validate that the operator was successfully installed.
      a. Run the following command to confirm that the operand request was created:

      ```
      oc get opreq common-service-edb -n ibm-common-services \
      -o jsonpath='{.status.phase}   {"\n"}'
      ```

      Verify that the command returns **Running**. If the command returns **Initialized** or **Installing**, wait several minutes and run the command again.

      b. Run the following command to confirm that the operator is installed:

      ```
      oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
      -o jsonpath='{.items[0].status.phase} {"\n"}'
      ```

      Verify that the command returns **Running**.

   3. Create the following operator subscription.

Ensure that you update the **namespace:**
ibm-common-services|cpd-operators
entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-watson-assistant-operator-subscription
  namespace: ibm-common-services|cpd-operators    # Pick the project that contains the Cloud Pak for Data
operator
spec:
  channel: v4.0
  name: ibm-watson-assistant-operator
  source: ibm-watson-assistant-operator-catalog
  sourceNamespace: openshift-marketplace
  installPlanApproval: Automatic
EOF
```

4. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-watson-assistant-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-watson-assistant-operator.v4.0.6**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-watson-assistant-operator.v4.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-watson-assistant-operator.v4.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the preceding operators for Watson Assistant with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **cloud-native-postgresql** *
- **cp4d-audit-webhook-operator**
- **ibm-cloud-databases-redis-operator** *
- **ibm-elasticsearch-operator** *
- **ibm-etcd-operator** *
- **ibm-minio-operator** *
- **ibm-model-train-dwf-operator**
- **ibm-rabbitmq-operator** *
- **ibm-watson-assistant-operator**
- **ibm-watson-gateway-operator** *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○
> IBM Entitled Registry with a catalog source for the IBM Operator Catalog

1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the **ibm-common-services** project.
   Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.
   If you did not install IBM Cloud Pak foundational services in **ibm-common-services**, update the value in the **namespace: ibm-common-services** entry.

```
cat <<EOF |oc apply -f -
apiVersion: operator.ibm.com/v1alpha1
kind: OperandRequest
metadata:
  name: common-service-edb
  namespace: ibm-common-services
spec:
  requests:
    - operands:
        - name: cloud-native-postgresql
      registry: common-service
EOF
```

2. Validate that the operator was successfully installed.
   a. Run the following command to confirm that the operand request was created:

   ```
   oc get opreq common-service-edb -n ibm-common-services \
   -o jsonpath='{.status.phase}  {"\n"}'
   ```

   Verify that the command returns **Running**. If the command returns **Initialized** or **Installing**, wait several minutes and run the command again.

   b. Run the following command to confirm that the operator is installed:

```
oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
-o jsonpath='{.items[0].status.phase} {"\n"}'
```

Verify that the command returns `Running`.

3. Create the following operator subscription.
   Ensure that you update the `namespace:`
   `ibm-common-services|cpd-operators`
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-watson-assistant-operator-subscription
  namespace: ibm-common-services|cpd-operators     # Pick the project that contains the Cloud Pak for Data
operator
spec:
  channel: v4.0
  name: ibm-watson-assistant-operator
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
  installPlanApproval: Automatic
EOF
```

4. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-watson-assistant-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-watson-assistant-operator.v4.0.6`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-watson-assistant-operator.v4.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-watson-assistant-operator.v4.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for Watson Assistant with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
   - `cloud-native-postgresql` *
   - `cp4d-audit-webhook-operator`
   - `ibm-cloud-databases-redis-operator` *
   - `ibm-elasticsearch-operator` *
   - `ibm-etcd-operator` *
   - `ibm-minio-operator` *
   - `ibm-model-train-dwf-operator`
   - `ibm-rabbitmq-operator` *
   - `ibm-watson-assistant-operator`
   - `ibm-watson-gateway-operator` *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Watson Assistant for Voice Interaction**

  Watson Assistant for Voice Interaction is comprised of the following services:
  - Voice Gateway
  - Watson Assistant
  - Watson Speech to Text
  - Watson Text to Speech
- **Watson Discovery**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - >
    Private container registry

    1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the **ibm-common-services** project.
       Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.
       If you did not install IBM Cloud Pak foundational services in **ibm-common-services**, update the value in the **namespace: ibm-common-services** entry.

       ```
       cat <<EOF |oc apply -f -
       apiVersion: operator.ibm.com/v1alpha1
       kind: OperandRequest
       metadata:
       ```

```
    name: common-service-edb
    namespace: ibm-common-services
spec:
  requests:
    - operands:
        - name: cloud-native-postgresql
      registry: common-service
EOF
```

2. Validate that the operator was successfully installed.

   a. Run the following command to confirm that the operand request was created:

   ```
   oc get opreq common-service-edb -n ibm-common-services \
   -o jsonpath='{.status.phase}  {"\n"}'
   ```

   Verify that the command returns `Running`. If the command returns `Initialized` or `Installing`, wait several minutes and run the command again.

   b. Run the following command to confirm that the operator is installed:

   ```
   oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
   -o jsonpath='{.items[0].status.phase}  {"\n"}'
   ```

   Verify that the command returns `Running`.

3. Create the following operator subscription.
   Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   cat <<EOF |oc apply -f -
   apiVersion: operators.coreos.com/v1alpha1
   kind: Subscription
   metadata:
     labels:
       app.kubernetes.io/instance: ibm-watson-discovery-operator-subscription
       app.kubernetes.io/managed-by: ibm-watson-discovery-operator
       app.kubernetes.io/name: ibm-watson-discovery-operator-subscription
     name: ibm-watson-discovery-operator-subscription
     namespace: ibm-common-services|cpd-operators    # Pick the project that contains the Cloud Pak for Data operator
   spec:
     channel: v4.0
     name: ibm-watson-discovery-operator
     source: ibm-watson-discovery-operator-catalog
     sourceNamespace: openshift-marketplace
     installPlanApproval: Automatic
   EOF
   ```

4. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-watson-discovery-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-watson-discovery-operator.v4.0.6`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-watson-discovery-operator.v4.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-watson-discovery-operator.v4.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for Watson Discovery with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `cloud-native-postgresql` *
- `ibm-elasticsearch-operator` *
- `ibm-etcd-operator` *
- `ibm-minio-operator` *
- `ibm-model-train-classic-operator`
- `ibm-rabbitmq-operator` *
- `ibm-watson-gateway-operator` *
- `ibm-watson-discovery-operator`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

o
> IBM Entitled Registry with catalog sources that pull specific versions of images

   1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the `ibm-common-services` project.

Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.

If you did not install IBM Cloud Pak foundational services in `ibm-common-services`, update the value in the `namespace: ibm-common-services` entry.

```
cat <<EOF |oc apply -f -
apiVersion: operator.ibm.com/v1alpha1
kind: OperandRequest
metadata:
  name: common-service-edb
  namespace: ibm-common-services
spec:
  requests:
    - operands:
        - name: cloud-native-postgresql
      registry: common-service
EOF
```

2. Validate that the operator was successfully installed.

   a. Run the following command to confirm that the operand request was created:

   ```
   oc get opreq common-service-edb -n ibm-common-services \
   -o jsonpath='{.status.phase}  {"\n"}'
   ```

   Verify that the command returns `Running`. If the command returns `Initialized` or `Installing`, wait several minutes and run the command again.

   b. Run the following command to confirm that the operator is installed:

   ```
   oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
   -o jsonpath='{.items[0].status.phase} {"\n"}'
   ```

   Verify that the command returns `Running`.

3. Create the following operator subscription.

   Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  labels:
    app.kubernetes.io/instance: ibm-watson-discovery-operator-subscription
    app.kubernetes.io/managed-by: ibm-watson-discovery-operator
    app.kubernetes.io/name: ibm-watson-discovery-operator-subscription
  name: ibm-watson-discovery-operator-subscription
  namespace: ibm-common-services|cpd-operators     # Pick the project that contains the Cloud Pak for Data
operator
spec:
  channel: v4.0
  name: ibm-watson-discovery-operator
  source: ibm-watson-discovery-operator-catalog
  sourceNamespace: openshift-marketplace
  installPlanApproval: Automatic
EOF
```

4. Validate that the operator was successfully created.

   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace (-n)` argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-watson-discovery-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-watson-discovery-operator.v4.0.6`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-watson-discovery-operator.v4.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-watson-discovery-operator.v4.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for Watson Discovery with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `cloud-native-postgresql` *
- `ibm-elasticsearch-operator` *
- `ibm-etcd-operator` *
- `ibm-minio-operator` *
- `ibm-model-train-classic-operator`
- `ibm-rabbitmq-operator` *
- `ibm-watson-gateway-operator` *

- **ibm-watson-discovery-operator**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- >
  IBM Entitled Registry with a catalog source for the IBM Operator Catalog

  1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the **ibm-common-services** project.
     Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.
     If you did not install IBM Cloud Pak foundational services in **ibm-common-services**, update the value in the **namespace: ibm-common-services** entry.

     ```
     cat <<EOF |oc apply -f -
     apiVersion: operator.ibm.com/v1alpha1
     kind: OperandRequest
     metadata:
       name: common-service-edb
       namespace: ibm-common-services
     spec:
       requests:
         - operands:
             - name: cloud-native-postgresql
           registry: common-service
     EOF
     ```

  2. Validate that the operator was successfully installed.
     a. Run the following command to confirm that the operand request was created:

        ```
        oc get opreq common-service-edb -n ibm-common-services \
        -o jsonpath='{.status.phase}   {"\n"}'
        ```

        Verify that the command returns **Running**. If the command returns **Initialized** or **Installing**, wait several minutes and run the command again.

     b. Run the following command to confirm that the operator is installed:

        ```
        oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
        -o jsonpath='{.items[0].status.phase} {"\n"}'
        ```

        Verify that the command returns **Running**.

  3. Create the following operator subscription.
     Ensure that you update the **namespace: ibm-common-services|cpd-operators** entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

     ```
     cat <<EOF |oc apply -f -
     apiVersion: operators.coreos.com/v1alpha1
     kind: Subscription
     metadata:
       labels:
         app.kubernetes.io/instance: ibm-watson-discovery-operator-subscription
         app.kubernetes.io/managed-by: ibm-watson-discovery-operator
         app.kubernetes.io/name: ibm-watson-discovery-operator-subscription
       name: ibm-watson-discovery-operator-subscription
       namespace: ibm-common-services|cpd-operators    # Pick the project that contains the Cloud Pak for Data
     operator
     spec:
       channel: v4.0
       name: ibm-watson-discovery-operator
       source: ibm-operator-catalog
       sourceNamespace: openshift-marketplace
       installPlanApproval: Automatic
     EOF
     ```

  4. Validate that the operator was successfully created.
     For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

     a. Run the following command to confirm that the subscription was triggered:

        ```
        oc get sub -n operator-project ibm-watson-discovery-operator-subscription \
        -o jsonpath='{.status.installedCSV} {"\n"}'
        ```

        Verify that the command returns **ibm-watson-discovery-operator.v4.0.6**.

     b. Run the following command to confirm that the cluster service version (CSV) is ready:

        ```
        oc get csv -n operator-project ibm-watson-discovery-operator.v4.0.6 \
        -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
        ```

        Verify that the command returns **Succeeded : install strategy completed with no errors**.

     c. Run the following command to confirm that the operator is ready:

        ```
        oc get deployments -n operator-project -l olm.owner="ibm-watson-discovery-operator.v4.0.6" \
        -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
        ```

        Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the preceding operators for Watson Discovery with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:

- cloud-native-postgresql *
- ibm-elasticsearch-operator *
- ibm-etcd-operator *
- ibm-minio-operator *
- ibm-model-train-classic-operator
- ibm-rabbitmq-operator *
- ibm-watson-gateway-operator *
- ibm-watson-discovery-operator

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Watson Knowledge Catalog**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - > Private container registry

    1. **Specialized installations only.** By default, IBM Cloud Pak foundational services expects that the Db2U operator (**ibm-db2u-operator**) is installed in the **ibm-common-services** project. However, if you are using the *specialized* installation architecture (where the IBM Cloud Pak foundational services operators and Cloud Pak for Data operators are in separate OpenShift projects), you must edit the IBM Cloud Pak foundational services operand registry to point to the project where the Cloud Pak for Data operators are installed:
       a. Run the following command to edit the IBM Cloud Pak foundational services operand registry.
          The following command assumes that the IBM Cloud Pak foundational services are installed in the **ibm-common-services** project.

          ```
          oc edit operandRegistry common-service \
          --namespace ibm-common-services
          ```

       b. Change the value of the **namespace** entry for the **ibm-db2u-operator**. Specify the OpenShift project where the Cloud Pak for Data operators are installed, for example, **cpd-operators**:

          ```
          - channel: v1.1
            installPlanApproval: Automatic
            name: ibm-db2u-operator
            namespace: cpd-operators        # Specify the project where the Cloud Pak for Data operators are installed
            packageName: db2u-operator
            scope: public
            sourceName: ibm-db2uoperator-catalog
            sourceNamespace: openshift-marketplace
          ```

       c. Save your changes and exit. For example, if you are using **vi**, hit esc and enter :wq.
    2. Create the Watson Knowledge Catalog operator subscription.
       Ensure that you update the **namespace:**
       ibm-common-services|cpd-operators
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       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
       ```

    3. Validate that the operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

       a. Run the following command to confirm that the subscription was triggered:

          ```
          oc get sub -n operator-project ibm-cpd-wkc-operator-catalog-subscription \
          -o jsonpath='{.status.installedCSV} {"\n"}'
          ```

          Verify that the command returns **ibm-cpd-wkc.v1.0.6**.

       b. Run the following command to confirm that the cluster service version (CSV) is ready:

          ```
          oc get csv -n operator-project ibm-cpd-wkc.v1.0.6 \
          -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
          ```

          Verify that the command returns **Succeeded : install strategy completed with no errors**.

       c. Run the following command to confirm that the operator is ready:

          ```
          oc get deployments -n operator-project -l olm.owner="ibm-cpd-wkc.v1.0.6" \
          -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
          ```

          Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the preceding operators for Watson Knowledge Catalog with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `db2u-operator` *
- `ibm-cpd-ccs-operator` *
- `ibm-cpd-datarefinery-operator`
- `ibm-db2aaservice-cp4d-operator` *
- `ibm-cpd-iis-operator`
- `ibm-cpd-wkc`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

-     >

  IBM Entitled Registry with catalog sources that pull specific versions of images

  1. **Specialized installations only.** By default, IBM Cloud Pak foundational services expects that the Db2U operator (`ibm-db2u-operator`) is installed in the `ibm-common-services` project. However, if you are using the *specialized* installation architecture (where the IBM Cloud Pak foundational services operators and Cloud Pak for Data operators are in separate OpenShift projects), you must edit the IBM Cloud Pak foundational services operand registry to point to the project where the Cloud Pak for Data operators are installed:

     a. Run the following command to edit the IBM Cloud Pak foundational services operand registry.
        The following command assumes that the IBM Cloud Pak foundational services are installed in the `ibm-common-services` project.

        ```
        oc edit operandRegistry common-service \
        --namespace ibm-common-services
        ```

     b. Change the value of the `namespace` entry for the `ibm-db2u-operator`. Specify the OpenShift project where the Cloud Pak for Data operators are installed, for example, `cpd-operators`:

        ```
        - channel: v1.1
            installPlanApproval: Automatic
            name: ibm-db2u-operator
            namespace: cpd-operators      # Specify the project where the Cloud Pak for Data operators are installed
            packageName: db2u-operator
            scope: public
            sourceName: ibm-db2uoperator-catalog
            sourceNamespace: openshift-marketplace
        ```

     c. Save your changes and exit. For example, if you are using `vi`, hit esc and enter `:wq`.

  2. Create the Watson Knowledge Catalog operator subscription.
     Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

     ```
     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
     ```

  3. Validate that the operator was successfully created.
     For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

     a. Run the following command to confirm that the subscription was triggered:

        ```
        oc get sub -n operator-project ibm-cpd-wkc-operator-catalog-subscription \
        -o jsonpath='{.status.installedCSV} {"\n"}'
        ```

        Verify that the command returns `ibm-cpd-wkc.v1.0.6`.

     b. Run the following command to confirm that the cluster service version (CSV) is ready:

        ```
        oc get csv -n operator-project ibm-cpd-wkc.v1.0.6 \
        -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
        ```

        Verify that the command returns `Succeeded : install strategy completed with no errors`.

     c. Run the following command to confirm that the operator is ready:

        ```
        oc get deployments -n operator-project -l olm.owner="ibm-cpd-wkc.v1.0.6" \
        -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
        ```

        Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for Watson Knowledge Catalog with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `db2u-operator` *
- `ibm-cpd-ccs-operator` *
- `ibm-cpd-datarefinery-operator`

- **ibm-db2aaservice-cp4d-operator** *
- **ibm-cpd-iis-operator**
- **ibm-cpd-wkc**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- ⟩
  IBM Entitled Registry with a catalog source for the IBM Operator Catalog

  1. **Specialized installations only.** By default, IBM Cloud Pak foundational services expects that the Db2U operator (**ibm-db2u-operator**) is installed in the **ibm-common-services** project. However, if you are using the *specialized* installation architecture (where the IBM Cloud Pak foundational services operators and Cloud Pak for Data operators are in separate OpenShift projects), you must edit the IBM Cloud Pak foundational services operand registry to point to the project where the Cloud Pak for Data operators are installed:

     a. Run the following command to edit the IBM Cloud Pak foundational services operand registry.
        The following command assumes that the IBM Cloud Pak foundational services are installed in the **ibm-common-services** project.

        ```
        oc edit operandRegistry common-service \
        --namespace ibm-common-services
        ```

     b. Change the value of the **namespace** entry for the **ibm-db2u-operator**. Specify the OpenShift project where the Cloud Pak for Data operators are installed, for example, **cpd-operators**:

        ```
        - channel: v1.1
          installPlanApproval: Automatic
          name: ibm-db2u-operator
          namespace: cpd-operators     # Specify the project where the Cloud Pak for Data operators are installed
          packageName: db2u-operator
          scope: public
          sourceName: ibm-db2uoperator-catalog
          sourceNamespace: openshift-marketplace
        ```

     c. Save your changes and exit. For example, if you are using **vi**, hit esc and enter :wq.
  2. Create the Watson Knowledge Catalog operator subscription.
     Ensure that you update the **namespace:**
     ibm-common-services|cpd-operators
     entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

     ```
     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
     ```

  3. Validate that the operator was successfully created.
     For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

     a. Run the following command to confirm that the subscription was triggered:

        ```
        oc get sub -n operator-project ibm-cpd-wkc-operator-catalog-subscription \
        -o jsonpath='{.status.installedCSV} {"\n"}'
        ```

        Verify that the command returns **ibm-cpd-wkc.v1.0.6**.

     b. Run the following command to confirm that the cluster service version (CSV) is ready:

        ```
        oc get csv -n operator-project ibm-cpd-wkc.v1.0.6 \
        -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
        ```

        Verify that the command returns **Succeeded : install strategy completed with no errors**.

     c. Run the following command to confirm that the operator is ready:

        ```
        oc get deployments -n operator-project -l olm.owner="ibm-cpd-wkc.v1.0.6" \
        -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
        ```

        Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the preceding operators for Watson Knowledge Catalog with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **db2u-operator** *
- **ibm-cpd-ccs-operator** *
- **ibm-cpd-datarefinery-operator**
- **ibm-db2aaservice-cp4d-operator** *
- **ibm-cpd-iis-operator**
- **ibm-cpd-wkc**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Watson Knowledge Studio**

Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

- > Private container registry

  1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the `ibm-common-services` project.
     Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.
     If you did not install IBM Cloud Pak foundational services in `ibm-common-services`, update the value in the `namespace: ibm-common-services` entry.

     ```
     cat <<EOF |oc apply -f -
     apiVersion: operator.ibm.com/v1alpha1
     kind: OperandRequest
     metadata:
       name: common-service-edb
       namespace: ibm-common-services
     spec:
       requests:
         - operands:
             - name: cloud-native-postgresql
           registry: common-service
     EOF
     ```

  2. Validate that the operator was successfully installed.
     a. Run the following command to confirm that the operand request was created:

        ```
        oc get opreq common-service-edb -n ibm-common-services \
        -o jsonpath='{.status.phase}  {"\n"}'
        ```

        Verify that the command returns `Running`. If the command returns `Initialized` or `Installing`, wait several minutes and run the command again.

     b. Run the following command to confirm that the operator is installed:

        ```
        oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
        -o jsonpath='{.items[0].status.phase} {"\n"}'
        ```

        Verify that the command returns `Running`.

  3. Create the following operator subscription.
     Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

     ```
     cat <<EOF |oc apply -f -
     apiVersion: operators.coreos.com/v1alpha1
     kind: Subscription
     metadata:
       labels:
         app.kubernetes.io/instance: ibm-watson-ks-operator-subscription
         app.kubernetes.io/managed-by: ibm-watson-ks-operator
         app.kubernetes.io/name: ibm-watson-ks-operator-subscription
       name: ibm-watson-ks-operator-subscription
       namespace: ibm-common-services|cpd-operators     # Pick the project that contains the Cloud Pak for Data
     operator
     spec:
       channel: v4.0
       name: ibm-watson-ks-operator
       source: ibm-watson-ks-operator-catalog
       sourceNamespace: openshift-marketplace
       installPlanApproval: Automatic
     EOF
     ```

  4. Validate that the operator was successfully created.
     For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

     a. Run the following command to confirm that the subscription was triggered:

        ```
        oc get sub -n operator-project ibm-watson-ks-operator-subscription \
        -o jsonpath='{.status.installedCSV} {"\n"}'
        ```

        Verify that the command returns `ibm-watson-ks-operator.v4.0.5`.

     b. Run the following command to confirm that the cluster service version (CSV) is ready:

        ```
        oc get csv -n operator-project ibm-watson-ks-operator.v4.0.5 \
        -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
        ```

        Verify that the command returns `Succeeded : install strategy completed with no errors`.

     c. Run the following command to confirm that the operator is ready:

        ```
        oc get deployments -n operator-project -l olm.owner="ibm-watson-ks-operator.v4.0.5" \
        -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
        ```

        Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for Watson Knowledge Studio with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `cloud-native-postgresql`
- `ibm-etcd-operator` *
- `ibm-minio-operator` *
- `ibm-watson-gateway-operator` *
- `ibm-watson-ks-operator`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- >
  IBM Entitled Registry with catalog sources that pull specific versions of images

  1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the `ibm-common-services` project.
     Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.
     If you did not install IBM Cloud Pak foundational services in `ibm-common-services`, update the value in the `namespace: ibm-common-services` entry.

     ```
     cat <<EOF |oc apply -f -
     apiVersion: operator.ibm.com/v1alpha1
     kind: OperandRequest
     metadata:
       name: common-service-edb
       namespace: ibm-common-services
     spec:
       requests:
         - operands:
             - name: cloud-native-postgresql
           registry: common-service
     EOF
     ```

  2. Validate that the operator was successfully installed.
     a. Run the following command to confirm that the operand request was created:

     ```
     oc get opreq common-service-edb -n ibm-common-services \
     -o jsonpath='{.status.phase}   {"\n"}'
     ```

     Verify that the command returns `Running`. If the command returns `Initialized` or `Installing`, wait several minutes and run the command again.

     b. Run the following command to confirm that the operator is installed:

     ```
     oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
     -o jsonpath='{.items[0].status.phase} {"\n"}'
     ```

     Verify that the command returns `Running`.

  3. Create the following operator subscription.
     Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

     ```
     cat <<EOF |oc apply -f -
     apiVersion: operators.coreos.com/v1alpha1
     kind: Subscription
     metadata:
       labels:
         app.kubernetes.io/instance: ibm-watson-ks-operator-subscription
         app.kubernetes.io/managed-by: ibm-watson-ks-operator
         app.kubernetes.io/name: ibm-watson-ks-operator-subscription
       name: ibm-watson-ks-operator-subscription
       namespace: ibm-common-services|cpd-operators     # Pick the project that contains the Cloud Pak for Data
     operator
     spec:
       channel: v4.0
       name: ibm-watson-ks-operator
       source: ibm-watson-ks-operator-catalog
       sourceNamespace: openshift-marketplace
       installPlanApproval: Automatic
     EOF
     ```

  4. Validate that the operator was successfully created.
     For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

     a. Run the following command to confirm that the subscription was triggered:

     ```
     oc get sub -n operator-project ibm-watson-ks-operator-subscription \
     -o jsonpath='{.status.installedCSV} {"\n"}'
     ```

     Verify that the command returns `ibm-watson-ks-operator.v4.0.5`.

     b. Run the following command to confirm that the cluster service version (CSV) is ready:

     ```
     oc get csv -n operator-project ibm-watson-ks-operator.v4.0.5 \
     -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
     ```

     Verify that the command returns `Succeeded : install strategy completed with no errors`.

     c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-watson-ks-operator.v4.0.5" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to 1. If the command returns 0, wait for the deployment to become available.

Important: If you created the preceding operators for Watson Knowledge Studio with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `cloud-native-postgresql`
- `ibm-etcd-operator` *
- `ibm-minio-operator` *
- `ibm-watson-gateway-operator` *
- `ibm-watson-ks-operator`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

o 〉
   ┌───┐
   │ IBM Entitled Registry with a catalog source for the IBM Operator Catalog │
   └───┘

1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the `ibm-common-services` project.

   Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.

   If you did not install IBM Cloud Pak foundational services in `ibm-common-services`, update the value in the `namespace: ibm-common-services` entry.

   ```
   cat <<EOF |oc apply -f -
   apiVersion: operator.ibm.com/v1alpha1
   kind: OperandRequest
   metadata:
     name: common-service-edb
     namespace: ibm-common-services
   spec:
     requests:
       - operands:
           - name: cloud-native-postgresql
         registry: common-service
   EOF
   ```

2. Validate that the operator was successfully installed.

   a. Run the following command to confirm that the operand request was created:

      ```
      oc get opreq common-service-edb -n ibm-common-services \
      -o jsonpath='{.status.phase}  {"\n"}'
      ```

      Verify that the command returns `Running`. If the command returns `Initialized` or `Installing`, wait several minutes and run the command again.

   b. Run the following command to confirm that the operator is installed:

      ```
      oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
      -o jsonpath='{.items[0].status.phase} {"\n"}'
      ```

      Verify that the command returns `Running`.

3. Create the following operator subscription.

   Ensure that you update the `namespace: ibm-common-services|cpd-operators` entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   cat <<EOF |oc apply -f -
   apiVersion: operators.coreos.com/v1alpha1
   kind: Subscription
   metadata:
     labels:
       app.kubernetes.io/instance: ibm-watson-ks-operator-subscription
       app.kubernetes.io/managed-by: ibm-watson-ks-operator
       app.kubernetes.io/name: ibm-watson-ks-operator-subscription
     name: ibm-watson-ks-operator-subscription
     namespace: ibm-common-services|cpd-operators     # Pick the project that contains the Cloud Pak for Data
   operator
   spec:
     channel: v4.0
     name: ibm-watson-ks-operator
     source: ibm-operator-catalog
     sourceNamespace: openshift-marketplace
     installPlanApproval: Automatic
   EOF
   ```

4. Validate that the operator was successfully created.

   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the --namespace (-n) argument.

   a. Run the following command to confirm that the subscription was triggered:

      ```
      oc get sub -n operator-project ibm-watson-ks-operator-subscription \
      -o jsonpath='{.status.installedCSV} {"\n"}'
      ```

      Verify that the command returns `ibm-watson-ks-operator.v4.0.5`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

      ```
      oc get csv -n operator-project ibm-watson-ks-operator.v4.0.5 \
      -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
      ```

Verify that the command returns `Succeeded : install strategy completed with no errors`.

    c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-watson-ks-operator.v4.0.5" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for Watson Knowledge Studio with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- `cloud-native-postgresql`
- `ibm-etcd-operator` *
- `ibm-minio-operator` *
- `ibm-watson-gateway-operator` *
- `ibm-watson-ks-operator`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Watson Machine Learning**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - > Private container registry

    1. Create the following operator subscription.
       Ensure that you update the `namespace:`
       `ibm-common-services|cpd-operators`
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

       ```
       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: v1.1
           installPlanApproval: Automatic
           name: ibm-cpd-wml-operator
           source: ibm-cpd-wml-operator-catalog
           sourceNamespace: openshift-marketplace
       EOF
       ```

    2. Validate that the operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

           a. Run the following command to confirm that the subscription was triggered:

       ```
       oc get sub -n operator-project ibm-cpd-wml-operator-subscription \
       -o jsonpath='{.status.installedCSV} {"\n"}'
       ```

       Verify that the command returns `ibm-cpd-wml-operator.v1.1.5`.

           b. Run the following command to confirm that the cluster service version (CSV) is ready:

       ```
       oc get csv -n operator-project ibm-cpd-wml-operator.v1.1.5 \
       -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
       ```

       Verify that the command returns `Succeeded : install strategy completed with no errors`.

           c. Run the following command to confirm that the operator is ready:

       ```
       oc get deployments -n operator-project -l olm.owner="ibm-cpd-wml-operator.v1.1.5" \
       -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
       ```

       Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

    Important: If you created the Watson Machine Learning operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
    - `ibm-cpd-ccs-operator` *
    - `ibm-cpd-wml-operator` *

    * An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

  - > IBM Entitled Registry with catalog sources that pull specific versions of images

    1. Create the following operator subscription.
       Ensure that you update the `namespace:`
       `ibm-common-services|cpd-operators`
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v1.1
    installPlanApproval: Automatic
    name: ibm-cpd-wml-operator
    source: ibm-cpd-wml-operator-catalog
    sourceNamespace: openshift-marketplace
EOF
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-cpd-wml-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-cpd-wml-operator.v1.1.5`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-wml-operator.v1.1.5 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-cpd-wml-operator.v1.1.5" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the Watson Machine Learning operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- `ibm-cpd-ccs-operator` *
- `ibm-cpd-wml-operator` *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

o   [>]
    IBM Entitled Registry with a catalog source for the IBM Operator Catalog

1. Create the following operator subscription.
   Ensure that you update the `namespace:`
   `ibm-common-services|cpd-operators`
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v1.1
    installPlanApproval: Automatic
    name: ibm-cpd-wml-operator
    source: ibm-operator-catalog
    sourceNamespace: openshift-marketplace
EOF
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-cpd-wml-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-cpd-wml-operator.v1.1.5`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-wml-operator.v1.1.5 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

Verify that the command returns `Succeeded : install strategy completed with no errors`.

    c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-wml-operator.v1.1.5" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the Watson Machine Learning operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- `ibm-cpd-ccs-operator` *
- `ibm-cpd-wml-operator` *

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Watson Machine Learning Accelerator**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - > Private container registry

    1. Create the following operator subscription.
       Ensure that you update the `namespace:`
       `ibm-common-services|cpd-operators`
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

    ```
    cat <<EOF |oc apply -f -
    apiVersion: operators.coreos.com/v1alpha1
    kind: Subscription
    metadata:
      name: ibm-cpd-wml-accelerator-operator
      namespace: ibm-common-services    # The project that contains the Cloud Pak for Data operator
    spec:
      name: ibm-cpd-wml-accelerator-operator
      channel: v1.0
      installPlanApproval: Automatic
      source: ibm-cpd-wml-accelerator-catalog
      sourceNamespace: openshift-marketplace
    EOF
    ```

    2. Validate that the operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (`cpd-operators`) for the `--namespace` (`-n`) argument.

           a. Run the following command to confirm that the subscription was triggered:

       ```
       oc get sub -n operator-project ibm-cpd-wml-accelerator-operator \
       -o jsonpath='{.status.installedCSV} {"\n"}'
       ```

       Verify that the command returns `ibm-cpd-wml-accelerator.v1.0.6`.

           b. Run the following command to confirm that the cluster service version (CSV) is ready:

       ```
       oc get csv -n operator-project ibm-cpd-wml-accelerator.v1.0.6 \
       -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
       ```

       Verify that the command returns `Succeeded : install strategy completed with no errors`.

           c. Run the following command to confirm that the operator is ready:

       ```
       oc get deployments -n operator-project -l olm.owner="ibm-cpd-wml-accelerator.v1.0.6" \
       -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
       ```

       Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

    Remember: If you created the Watson Machine Learning Accelerator operator with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve the update request for `ibm-cpd-wml-accelerator-operator`.

  - > IBM Entitled Registry with catalog sources that pull specific versions of images

    1. Create the following operator subscription.
       Ensure that you update the `namespace:`
       `ibm-common-services|cpd-operators`
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

    ```
    cat <<EOF |oc apply -f -
    apiVersion: operators.coreos.com/v1alpha1
    kind: Subscription
    metadata:
      name: ibm-cpd-wml-accelerator-operator
      namespace: ibm-common-services    # The project that contains the Cloud Pak for Data operator
    spec:
      name: ibm-cpd-wml-accelerator-operator
      channel: v1.0
      installPlanApproval: Automatic
      source: ibm-cpd-wml-accelerator-catalog
      sourceNamespace: openshift-marketplace
    EOF
    ```

    2. Validate that the operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (`cpd-operators`) for the `--namespace` (`-n`) argument.

a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-cpd-wml-accelerator-operator \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns **ibm-cpd-wml-accelerator.v1.0.6**.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-cpd-wml-accelerator.v1.0.6 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns **Succeeded : install strategy completed with no errors**.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-wml-accelerator.v1.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Remember: If you created the Watson Machine Learning Accelerator operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve the update request for **ibm-cpd-wml-accelerator-operator**.

o >
| IBM Entitled Registry with a catalog source for the IBM Operator Catalog |

1. Create the following operator subscription.
   Ensure that you update the **namespace: ibm-common-services|cpd-operators** entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v1.0
  installPlanApproval: Automatic
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (**cpd-operators**) for the **--namespace** (**-n**) argument.

a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-cpd-wml-accelerator-operator \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns **ibm-cpd-wml-accelerator.v1.0.6**.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-cpd-wml-accelerator.v1.0.6 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns **Succeeded : install strategy completed with no errors**.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-wml-accelerator.v1.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Remember: If you created the Watson Machine Learning Accelerator operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve the update request for **ibm-cpd-wml-accelerator-operator**.

- **Watson OpenScale**

Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

o >
| Private container registry |

1. Create the following operator subscription.
   Ensure that you update the **namespace: ibm-common-services|cpd-operators** entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v1.5
  installPlanApproval: Automatic
  name: ibm-cpd-wos
  source: ibm-openscale-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-watson-openscale-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-cpd-wos.v1.5.1**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-wos.v1.5.1 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-cpd-wos.v1.5.1" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

   Remember: If you created the Watson OpenScale operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve the update request for **ibm-cpd-wos**.

   ○ >
   IBM Entitled Registry with catalog sources that pull specific versions of images

   1. Create the following operator subscription.
      Ensure that you update the **namespace: ibm-common-services|cpd-operators** entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

   ```
   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: v1.5
     installPlanApproval: Automatic
     name: ibm-cpd-wos
     source: ibm-openscale-operator-catalog
     sourceNamespace: openshift-marketplace
   EOF
   ```

   2. Validate that the operator was successfully created.
      For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

      a. Run the following command to confirm that the subscription was triggered:

      ```
      oc get sub -n operator-project ibm-watson-openscale-operator-subscription \
      -o jsonpath='{.status.installedCSV} {"\n"}'
      ```

      Verify that the command returns **ibm-cpd-wos.v1.5.1**.

      b. Run the following command to confirm that the cluster service version (CSV) is ready:

      ```
      oc get csv -n operator-project ibm-cpd-wos.v1.5.1 \
      -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
      ```

      Verify that the command returns **Succeeded : install strategy completed with no errors**.

      c. Run the following command to confirm that the operator is ready:

      ```
      oc get deployments -n operator-project -l olm.owner="ibm-cpd-wos.v1.5.1" \
      -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
      ```

      Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

   Remember: If you created the Watson OpenScale operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve the update request for **ibm-cpd-wos**.

   ○ >
   IBM Entitled Registry with a catalog source for the IBM Operator Catalog

1. Create the following operator subscription.
   Ensure that you update the **namespace:**
   ibm-common-services|cpd-operators
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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: v1.5
  installPlanApproval: Automatic
  name: ibm-cpd-wos
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace (-n)** argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-watson-openscale-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-cpd-wos.v1.5.1**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-wos.v1.5.1 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-cpd-wos.v1.5.1" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

   Remember: If you created the Watson OpenScale operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve the update request for **ibm-cpd-wos**.

- **Watson Speech to Text**

  Important: The same operator subscription is used for Watson Speech to Text and Watson Text to Speech. If you plan to install both services, you only need to create the subscription once.
  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - > Private container registry

    1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the **ibm-common-services** project.
       Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.
       If you did not install IBM Cloud Pak foundational services in **ibm-common-services**, update the value in the **namespace: ibm-common-services** entry.

    ```
    cat <<EOF |oc apply -f -
    apiVersion: operator.ibm.com/v1alpha1
    kind: OperandRequest
    metadata:
      name: common-service-edb
      namespace: ibm-common-services
    spec:
      requests:
        - operands:
            - name: cloud-native-postgresql
          registry: common-service
    EOF
    ```

    2. Validate that the operator was successfully installed.
       a. Run the following command to confirm that the operand request was created:

       ```
       oc get opreq common-service-edb -n ibm-common-services \
       -o jsonpath='{.status.phase}   {"\n"}'
       ```

       Verify that the command returns **Running**. If the command returns **Initialized** or **Installing**, wait several minutes and run the command again.

       b. Run the following command to confirm that the operator is installed:

```
oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
-o jsonpath='{.items[0].status.phase} {"\n"}'
```

Verify that the command returns `Running`.

3. Create the following operator subscription.
   Ensure that you update the `namespace:`
   `ibm-common-services|cpd-operators`
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-watson-speech-operator-subscription
  namespace: ibm-common-services|cpd-operators        # Pick the project that contains the Cloud Pak for Data
operator
spec:
  channel: v4.0
  name: ibm-watson-speech-operator
  source: ibm-watson-speech-operator-catalog
  sourceNamespace: openshift-marketplace
  installPlanApproval: Automatic
EOF
```

4. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-watson-speech-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-watson-speech-operator.v4.0.6`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-watson-speech-operator.v4.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-watson-speech-operator.v4.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for the Watson Speech services with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- `cloud-native-postgresql` *
- `ibm-minio-operator` *
- `ibm-rabbitmq-operator` *
- `ibm-watson-gateway-operator` *
- `ibm-watson-speech-operator`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○ | >
   IBM Entitled Registry with catalog sources that pull specific versions of images |

1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the `ibm-common-services` project.
   Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.
   If you did not install IBM Cloud Pak foundational services in `ibm-common-services`, update the value in the `namespace:`
   `ibm-common-services` entry.

```
cat <<EOF |oc apply -f -
apiVersion: operator.ibm.com/v1alpha1
kind: OperandRequest
metadata:
  name: common-service-edb
  namespace: ibm-common-services
spec:
  requests:
    - operands:
        - name: cloud-native-postgresql
      registry: common-service
EOF
```

2. Validate that the operator was successfully installed.
   a. Run the following command to confirm that the operand request was created:

   ```
   oc get opreq common-service-edb -n ibm-common-services \
   -o jsonpath='{.status.phase}  {"\n"}'
   ```

   Verify that the command returns `Running`. If the command returns `Initialized` or `Installing`, wait several minutes and run the command again.

   b. Run the following command to confirm that the operator is installed:

```
oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
-o jsonpath='{.items[0].status.phase} {"\n"}'
```

Verify that the command returns `Running`.

3. Create the following operator subscription.
   Ensure that you update the `namespace:`
   `ibm-common-services|cpd-operators`
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-watson-speech-operator-subscription
  namespace: ibm-common-services|cpd-operators      # Pick the project that contains the Cloud Pak for Data
operator
spec:
  channel: v4.0
  name: ibm-watson-speech-operator
  source: ibm-watson-speech-operator-catalog
  sourceNamespace: openshift-marketplace
  installPlanApproval: Automatic
EOF
```

4. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-watson-speech-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-watson-speech-operator.v4.0.6`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-watson-speech-operator.v4.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-watson-speech-operator.v4.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for the Watson Speech services with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:

- `cloud-native-postgresql` *
- `ibm-minio-operator` *
- `ibm-rabbitmq-operator` *
- `ibm-watson-gateway-operator` *
- `ibm-watson-speech-operator`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

o
>
 ┌──┐
 │ IBM Entitled Registry with a catalog source for the IBM Operator Catalog │
 └──┘

1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the `ibm-common-services` project.
   Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.
   If you did not install IBM Cloud Pak foundational services in `ibm-common-services`, update the value in the `namespace: ibm-common-services` entry.

```
cat <<EOF |oc apply -f -
apiVersion: operator.ibm.com/v1alpha1
kind: OperandRequest
metadata:
  name: common-service-edb
  namespace: ibm-common-services
spec:
  requests:
    - operands:
        - name: cloud-native-postgresql
      registry: common-service
EOF
```

2. Validate that the operator was successfully installed.
   a. Run the following command to confirm that the operand request was created:

   ```
   oc get opreq common-service-edb -n ibm-common-services \
   -o jsonpath='{.status.phase}  {"\n"}'
   ```

   Verify that the command returns `Running`. If the command returns `Initialized` or `Installing`, wait several minutes and run the command again.

   b. Run the following command to confirm that the operator is installed:

```
oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
-o jsonpath='{.items[0].status.phase} {"\n"}'
```

Verify that the command returns `Running`.

3. Create the following operator subscription.
   Ensure that you update the `namespace:`
   `ibm-common-services|cpd-operators`
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-watson-speech-operator-subscription
  namespace: ibm-common-services|cpd-operators       # Pick the project that contains the Cloud Pak for Data
operator
spec:
  channel: v4.0
  name: ibm-watson-speech-operator
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
  installPlanApproval: Automatic
EOF
```

4. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-watson-speech-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-watson-speech-operator.v4.0.6`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-watson-speech-operator.v4.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-watson-speech-operator.v4.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for the Watson Speech services with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
   - `cloud-native-postgresql` *
   - `ibm-minio-operator` *
   - `ibm-rabbitmq-operator` *
   - `ibm-watson-gateway-operator` *
   - `ibm-watson-speech-operator`
   * An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Watson Studio**

  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - **> Private container registry**

    1. Create the following operator subscription.
       Ensure that you update the `namespace:`
       `ibm-common-services|cpd-operators`
       entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

    ```
    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
    ```

    2. Validate that the operator was successfully created.
       For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-cpd-ws-operator-catalog-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns **ibm-cpd-wsl.v2.0.6**.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-cpd-wsl.v2.0.6 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns **Succeeded : install strategy completed with no errors**.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-wsl.v2.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the Watson Studio operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **ibm-cpd-ccs-operator** *
- **ibm-cpd-datarefinery-operator** *
- **ibm-cpd-wsl** *
- **ibm-cpd-ws-runtimes**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○ >
   IBM Entitled Registry with catalog sources that pull specific versions of images

1. Create the following operator subscription.
   Ensure that you update the **namespace:**
   ibm-common-services|cpd-operators
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-cpd-ws-operator-catalog-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns **ibm-cpd-wsl.v2.0.6**.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-cpd-wsl.v2.0.6 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns **Succeeded : install strategy completed with no errors**.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-cpd-wsl.v2.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the Watson Studio operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **ibm-cpd-ccs-operator** *
- **ibm-cpd-datarefinery-operator** *
- **ibm-cpd-wsl** *
- **ibm-cpd-ws-runtimes**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○ >
   IBM Entitled Registry with a catalog source for the IBM Operator Catalog

1. Create the following operator subscription.

Ensure that you update the **namespace:**
ibm-common-services|cpd-operators
entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
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
```

2. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either **ibm-common-services** or **cpd-operators**) for the **--namespace** (**-n**) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-cpd-ws-operator-catalog-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns **ibm-cpd-wsl.v2.0.6**.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-cpd-wsl.v2.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns **Succeeded : install strategy completed with no errors**.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-cpd-wsl.v2.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to **1**. If the command returns **0**, wait for the deployment to become available.

Important: If you created the Watson Studio operator with the manual install plan (**installPlanApproval: Manual**), a cluster administrator must approve any pending update requests for the following operators:
- **ibm-cpd-ccs-operator** *
- **ibm-cpd-datarefinery-operator** *
- **ibm-cpd-wsl** *
- **ibm-cpd-ws-runtimes**

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

- **Watson Studio Runtimes**

  The operator subscription for Watson Studio Runtimes is automatically created when you create the operator subscription for Watson Studio.

- **Watson Text to Speech**

  Important: The same operator subscription is used for Watson Speech to Text and Watson Text to Speech. If you plan to install both services, you only need to create the subscription once.
  Create the appropriate operator subscription for your environment. The catalog that the operator subscription points to depends on the type of catalog source that you created and the location from which the cluster pulls images.

  - >
    Private container registry

    1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the **ibm-common-services** project.
       Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.
       If you did not install IBM Cloud Pak foundational services in **ibm-common-services**, update the value in the **namespace: ibm-common-services** entry.

       ```
       cat <<EOF |oc apply -f -
       apiVersion: operator.ibm.com/v1alpha1
       kind: OperandRequest
       metadata:
         name: common-service-edb
         namespace: ibm-common-services
       spec:
         requests:
           - operands:
               - name: cloud-native-postgresql
             registry: common-service
       EOF
       ```

    2. Validate that the operator was successfully installed.
       a. Run the following command to confirm that the operand request was created:

       ```
       oc get opreq common-service-edb -n ibm-common-services \
       -o jsonpath='{.status.phase}   {"\n"}'
       ```

Verify that the command returns `Running`. If the command returns `Initialized` or `Installing`, wait several minutes and run the command again.

b. Run the following command to confirm that the operator is installed:

```
oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
-o jsonpath='{.items[0].status.phase} {"\n"}'
```

Verify that the command returns `Running`.

3. Create the following operator subscription.
   Ensure that you update the `namespace:`
   `ibm-common-services|cpd-operators`
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-watson-speech-operator-subscription
  namespace: ibm-common-services|cpd-operators      # Pick the project that contains the Cloud Pak for Data
operator
spec:
  channel: v4.0
  name: ibm-watson-speech-operator
  source: ibm-watson-speech-operator-catalog
  sourceNamespace: openshift-marketplace
  installPlanApproval: Automatic
EOF
```

4. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-watson-speech-operator-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

   Verify that the command returns `ibm-watson-speech-operator.v4.0.6`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-watson-speech-operator.v4.0.6 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-watson-speech-operator.v4.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

   Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for the Watson Speech services with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- `cloud-native-postgresql` *
- `ibm-minio-operator` *
- `ibm-rabbitmq-operator` *
- `ibm-watson-gateway-operator` *
- `ibm-watson-speech-operator`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○ [>]
  [IBM Entitled Registry with catalog sources that pull specific versions of images]

1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the `ibm-common-services` project.
   Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.
   If you did not install IBM Cloud Pak foundational services in `ibm-common-services`, update the value in the `namespace:`
   `ibm-common-services` entry.

```
cat <<EOF |oc apply -f -
apiVersion: operator.ibm.com/v1alpha1
kind: OperandRequest
metadata:
  name: common-service-edb
  namespace: ibm-common-services
spec:
  requests:
    - operands:
        - name: cloud-native-postgresql
      registry: common-service
EOF
```

2. Validate that the operator was successfully installed.
   a. Run the following command to confirm that the operand request was created:

```
oc get opreq common-service-edb -n ibm-common-services \
-o jsonpath='{.status.phase}  {"\n"}'
```

Verify that the command returns `Running`. If the command returns `Initialized` or `Installing`, wait several minutes and run the command again.

b. Run the following command to confirm that the operator is installed:

```
oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
-o jsonpath='{.items[0].status.phase} {"\n"}'
```

Verify that the command returns `Running`.

3. Create the following operator subscription.
   Ensure that you update the `namespace:`
   `ibm-common-services|cpd-operators`
   entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-watson-speech-operator-subscription
  namespace: ibm-common-services|cpd-operators      # Pick the project that contains the Cloud Pak for Data
operator
spec:
  channel: v4.0
  name: ibm-watson-speech-operator
  source: ibm-watson-speech-operator-catalog
  sourceNamespace: openshift-marketplace
  installPlanApproval: Automatic
EOF
```

4. Validate that the operator was successfully created.
   For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

   a. Run the following command to confirm that the subscription was triggered:

   ```
   oc get sub -n operator-project ibm-watson-speech-operator-subscription \
   -o jsonpath='{.status.installedCSV} {"\n"}'
   ```

   Verify that the command returns `ibm-watson-speech-operator.v4.0.6`.

   b. Run the following command to confirm that the cluster service version (CSV) is ready:

   ```
   oc get csv -n operator-project ibm-watson-speech-operator.v4.0.6 \
   -o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
   ```

   Verify that the command returns `Succeeded : install strategy completed with no errors`.

   c. Run the following command to confirm that the operator is ready:

   ```
   oc get deployments -n operator-project -l olm.owner="ibm-watson-speech-operator.v4.0.6" \
   -o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
   ```

   Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for the Watson Speech services with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- `cloud-native-postgresql` *
- `ibm-minio-operator` *
- `ibm-rabbitmq-operator` *
- `ibm-watson-gateway-operator` *
- `ibm-watson-speech-operator`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

○ 〉
   IBM Entitled Registry with a catalog source for the IBM Operator Catalog

1. Submit the following operand request to install the Cloud Native PostgreSQL operator in the `ibm-common-services` project.
   Important: If you installed the Cloud Native PostgreSQL operator as a cluster-scoped operator for use outside of Cloud Pak for Data, you can skip this step.
   If you did not install IBM Cloud Pak foundational services in `ibm-common-services`, update the value in the `namespace:`
   `ibm-common-services` entry.

```
cat <<EOF |oc apply -f -
apiVersion: operator.ibm.com/v1alpha1
kind: OperandRequest
metadata:
  name: common-service-edb
  namespace: ibm-common-services
spec:
  requests:
    - operands:
        - name: cloud-native-postgresql
      registry: common-service
EOF
```

2. Validate that the operator was successfully installed.
   a. Run the following command to confirm that the operand request was created:

   ```
   oc get opreq common-service-edb -n ibm-common-services \
   -o jsonpath='{.status.phase}   {"\n"}'
   ```

Verify that the command returns `Running`. If the command returns `Initialized` or `Installing`, wait several minutes and run the command again.

b. Run the following command to confirm that the operator is installed:

```
oc get pod -n ibm-common-services -l app.kubernetes.io/name=cloud-native-postgresql \
-o jsonpath='{.items[0].status.phase} {"\n"}'
```

Verify that the command returns `Running`.

3. Create the following operator subscription.
Ensure that you update the `namespace:`
`ibm-common-services|cpd-operators`
entry to specify the Red Hat OpenShift Container Platform project where you want to install the Cloud Pak for Data software operators.

```
cat <<EOF |oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-watson-speech-operator-subscription
  namespace: ibm-common-services|cpd-operators     # Pick the project that contains the Cloud Pak for Data
operator
spec:
  channel: v4.0
  name: ibm-watson-speech-operator
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
  installPlanApproval: Automatic
EOF
```

4. Validate that the operator was successfully created.
For each command, ensure that you specify the appropriate Red Hat OpenShift project (either `ibm-common-services` or `cpd-operators`) for the `--namespace` (`-n`) argument.

a. Run the following command to confirm that the subscription was triggered:

```
oc get sub -n operator-project ibm-watson-speech-operator-subscription \
-o jsonpath='{.status.installedCSV} {"\n"}'
```

Verify that the command returns `ibm-watson-speech-operator.v4.0.6`.

b. Run the following command to confirm that the cluster service version (CSV) is ready:

```
oc get csv -n operator-project ibm-watson-speech-operator.v4.0.6 \
-o jsonpath='{ .status.phase } : { .status.message} {"\n"}'
```

Verify that the command returns `Succeeded : install strategy completed with no errors`.

c. Run the following command to confirm that the operator is ready:

```
oc get deployments -n operator-project -l olm.owner="ibm-watson-speech-operator.v4.0.6" \
-o jsonpath="{.items[0].status.availableReplicas} {'\n'}"
```

Verify that the command returns an integer greater than or equal to `1`. If the command returns `0`, wait for the deployment to become available.

Important: If you created the preceding operators for the Watson Speech services with the manual install plan (`installPlanApproval: Manual`), a cluster administrator must approve any pending update requests for the following operators:
- `cloud-native-postgresql` *
- `ibm-minio-operator` *
- `ibm-rabbitmq-operator` *
- `ibm-watson-gateway-operator` *
- `ibm-watson-speech-operator`

* An asterisk indicates that the operator is used by multiple services. If you do not see an update request for this operator, it might have been approved as part of the approval for another service.

Previous topic: Installing IBM Cloud Pak foundational services
Next topic: Custom security context constraints for services
Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see Documentation for previous 4.0.x refreshes.

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 use some custom SCCs.

OpenShift provides a set of predefined SCCs that control the actions that a pod can perform and what it can access. These SCCs 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:

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

If you plan to install Watson Knowledge Catalog, you must create the custom SCCs manually. If you have multiple copies of Cloud Pak for Data installed in different namespaces, you must create these SCCs one time for the cluster.

If you plan to install Db2 or Db2 Warehouse, the Db2 operator creates the custom SCC, service accounts, roles, and role bindings.

The Db2 Big SQL and Data Virtualization services embed Db2 and use the Db2 custom SCC.

The OpenPages service embeds Db2, but the custom SCC is used only if the OpenPages service instance is provisioned by using the internal database option. When the OpenPages service is provisioned by using a database outside the cluster, the custom SCC is not required.

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

- **Creating the custom security context constraint for Watson Knowledge Catalog**
  The Watson Knowledge Catalog service requires the use of a custom security context constraint (SCC).
- **Security context constraints for 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
Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see Documentation for previous 4.0.x refreshes.

Creating the custom security context constraint for Watson Knowledge Catalog

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

An SCC controls the permissions that pods have on the Red Hat® OpenShift® Container Platform cluster.

Watson Knowledge Catalog requires a custom SCC to run.

To create the SCC, complete the following steps:

1. Run the following command to create an SCC named `wkc-iis-scc`.
   The command assumes that you plan to install Watson Knowledge Catalog in the `cpd-instance` project. If you plan to install Watson Knowledge Catalog in a different project, replace `cpd-instance` in `-system:serviceaccount:cpd-instance:wkc-iis-sa` with the appropriate project for your environment.

```
cat <<EOF |oc apply -f -
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:cpd-instance:wkc-iis-sa
    EOF
```

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

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

3. Create the SCC cluster role for **wkc-iis-scc**:

   ```
   oc create clusterrole system:openshift:scc:wkc-iis-scc \
   --verb=use \
   --resource=scc \
   --resource-name=wkc-iis-scc
   ```

4. Assign the **wkc-iis-sa** service account to the SCC cluster role.
   The command assumes that you plan to install Watson Knowledge Catalog in the **cpd-instance** project. If you plan to install Watson Knowledge Catalog in a different project, replace **cpd-instance** in the **serviceaccount** parameter with the appropriate project for your environment.

   ```
   oc create rolebinding wkc-iis-scc-rb \
   --namespace cpd-instance \
   --clusterrole=system:openshift:scc:wkc-iis-scc \
   --serviceaccount=cpd-instance:wkc-iis-sa
   ```

   Replace **cpd-instance** with the name of the Red Hat OpenShift project where you plan to install Watson Knowledge Catalog.

5. Confirm that the **wkc-iis-sa** service account can use the **wkc-iis-scc** SCC:

   ```
   oc adm policy who-can use scc wkc-iis-scc \
   --namespace cpd-instance | grep "wkc-iis-sa"
   ```

For more information about SCCs, see the Red Hat OpenShift documentation:

- Version 4.6: [Managing Security Context Constraints](#)
- Version 4.8: [Managing Security Context Constraints](#)

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Security context constraints for 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.

- **[Manually creating an SCC, service account, role, and role binding](#)**
  If you do not want Db2 or Db2 Warehouse to automatically create custom Security Context Constraints (SCC) when you deploy the service on Cloud Pak for Data, you can manually create the SCC, service account, role, and role binding within Red Hat OpenShift.
- **[Specifying a custom service account, SCC, role, and role binding before deploying Db2 or Db2 Warehouse](#)**
  If you are using a custom service account, security context constraints (SCC), role, and role binding for the Db2 or Db2 Warehouse service, you need to take steps

outside of the Cloud Pak for Data web console to prepare for deploying Db2 or Db2 Warehouse.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Manually creating an SCC, service account, role, and role binding

If you do not want Db2® or Db2 Warehouse to automatically create custom Security Context Constraints (SCC) when you deploy the service on Cloud Pak for Data, you can manually create the SCC, service account, role, and role binding within Red Hat® OpenShift®.

Before you begin

Define the following parameters in OpenShift:

- SCC_NAME
- PROJECT
- SERVICE_ACCOUNT
- ROLE_NAME
- ROLEBINDING_NAME

When you run the commands below, the values that you assign to these parameters will automatically be used in the commands in place of `${SCC_NAME}`, `${PROJECT}`, `${SERVICE_ACCOUNT}, ${ROLE_NAME}, and ${ROLEBINDING_NAME}`.

For more information, see [Managing security context constraints](#) and [Understanding and creating service accounts](#) in the OpenShift documentation.

About this task

Creating the SCC and service account prior to deploying Db2 or Db2 Warehouse gives you full control over the security specifications. The following procedure shows you how to create the required objects in two ways:

- Privileged account without enabling unsafe sysctls
- Limited privileged account with unsafe sysctls enabled

Procedure

Run one of the following commands to create either a privileged account or unprivileged account:

- **Privileged account without enabling unsafe sysctls**

```
oc apply -n ${PROJECT} -f - <<EOF
apiVersion: v1
kind: ServiceAccount
metadata:
  name: ${SERVICE_ACCOUNT}
  namespace: ${PROJECT}
EOF

oc apply -n ${PROJECT} -f - <<EOF
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: ${ROLE_NAME}
  namespace: ${PROJECT}
rules:
- apiGroups:
  - ""
  resources:
  - endpoints
  - pods
  verbs:
  - get
  - patch
  - update
- apiGroups:
  - apps
  resources:
  - statefulsets
  - deployments
  - replicasets
  verbs:
  - get
  - list
- apiGroups:
  - ""
  resources:
  - configmaps
  verbs:
  - get
  - patch
  - watch
  - list
  - update
```

```
    - apiGroups:
      - ""
      resources:
      - secrets
      verbs:
      - get
      - create
      - update
    - apiGroups:
      - db2u.databases.ibm.com
      resources:
      - recipes
      verbs:
      - watch
      - get
      - update
      - create
      - patch
      - list
      - delete
    - apiGroups:
      - db2u.databases.ibm.com
      resources:
      - buckets
      verbs:
      - patch
    - apiGroups:
      - db2u.databases.ibm.com
      resources:
      - backups
      verbs:
      - patch
      - delete
      - list
    - apiGroups:
      - db2u.databases.ibm.com
      resources:
      - formations
      verbs:
      - get
    - apiGroups:
      - ""
      resources:
      - pods/exec
      verbs:
      - create
    - apiGroups:
      - ""
      resources:
      - pods
      verbs:
      - watch
      - list
      - get
    - apiGroups:
      - ""
      resources:
      - services
      verbs:
      - watch
      - list
      - get
EOF

oc apply -n ${PROJECT} -f - <<EOF
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: ${ROLEBINDING_NAME}
  namespace: ${PROJECT}
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: ${ROLE_NAME}
subjects:
- kind: ServiceAccount
  name: ${SERVICE_ACCOUNT}
  namespace: ${PROJECT}
EOF

oc apply -n ${PROJECT} -f - <<EOF
allowHostDirVolumePlugin: true
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: true
allowPrivilegedContainer: true
allowedCapabilities: []
apiVersion: security.openshift.io/v1
defaultAddCapabilities: null
fsGroup:
  ranges:
  - max: 1000
    min: 1000
  type: MustRunAs
```

```
      groups: []
      kind: SecurityContextConstraints
      metadata:
        name: ${SCC_NAME}
      priority: null
      readOnlyRootFilesystem: false
      requiredDropCapabilities:
      - KILL
      - SETUID
      - SETGID
      - MKNOD
      - ALL
      runAsUser:
        type: RunAsAny
      seLinuxContext:
        type: MustRunAs
      supplementalGroups:
        type: RunAsAny
      users:
      - system:serviceaccount:${PROJECT}:${SERVICE_ACCOUNT}
      volumes:
      - configMap
      - downwardAPI
      - emptyDir
      - hostPath
      - persistentVolumeClaim
      - projected
      - secret
      EOF
```

- **Limited privileged account with unsafe sysctls enabled**

```
oc apply -n ${PROJECT} -f - <<EOF
apiVersion: v1
kind: ServiceAccount
metadata:
  name: ${SERVICE_ACCOUNT}
  namespace: ${PROJECT}
EOF
```

```
oc apply -n ${PROJECT} -f - <<EOF
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: ${ROLE_NAME}
  namespace: zen
rules:
- apiGroups:
  - ""
  resources:
  - endpoints
  - pods
  verbs:
  - get
  - patch
  - update
- apiGroups:
  - apps
  resources:
  - statefulsets
  - deployments
  - replicasets
  verbs:
  - get
  - list
- apiGroups:
  - ""
  resources:
  - configmaps
  verbs:
  - get
  - patch
  - watch
  - list
  - update
- apiGroups:
  - ""
  resources:
  - secrets
  verbs:
  - get
  - create
  - update
- apiGroups:
  - db2u.databases.ibm.com
  resources:
  - recipes
  verbs:
  - watch
  - get
  - update
  - create
  - patch
  - list
  - delete
- apiGroups:
```

```
    - db2u.databases.ibm.com
  resources:
  - buckets
  verbs:
  - patch
- apiGroups:
  - db2u.databases.ibm.com
  resources:
  - backups
  verbs:
  - patch
  - delete
  - list
- apiGroups:
  - db2u.databases.ibm.com
  resources:
  - formations
  verbs:
  - get
- apiGroups:
  - ""
  resources:
  - pods/exec
  verbs:
  - create
- apiGroups:
  - ""
  resources:
  - pods
  verbs:
  - watch
  - list
  - get
- apiGroups:
  - ""
  resources:
  - services
  verbs:
  - watch
  - list
  - get
EOF

oc apply -n ${PROJECT} -f - <<EOF
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: ${ROLEBINDING_NAME}
  namespace: ${PROJECT}
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: ${ROLE_NAME}
subjects:
- kind: ServiceAccount
  name: ${SERVICE_ACCOUNT}
  namespace: ${PROJECT}
EOF


oc apply -n ${PROJECT} -f - <<EOF
allowHostDirVolumePlugin: false
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: false
allowPrivilegedContainer: false
allowedCapabilities: []
allowedUnsafeSysctls:
- kernel.shmmni
- kernel.shmmax
- kernel.shmall
- kernel.sem
- kernel.msgmni
- kernel.msgmax
- kernel.msgmnb
apiVersion: security.openshift.io/v1
defaultAddCapabilities: null
fsGroup:
  ranges:
  - max: 1000
    min: 1000
  type: MustRunAs
groups: []
kind: SecurityContextConstraints
metadata:
  name: zen-c-db2u-oltp-restr-scc
priority: null
readOnlyRootFilesystem: false
requiredDropCapabilities:
- KILL
- SETUID
- SETGID
- MKNOD
- ALL
```

```
        runAsUser:
          type: MustRunAs
          uid: 500
        seLinuxContext:
          type: MustRunAs
        supplementalGroups:
          type: RunAsAny
        users:
        - system:serviceaccount:${PROJECT}:${SERVICE_ACCOUNT}
        volumes:
        - configMap
        - downwardAPI
        - emptyDir
        - persistentVolumeClaim
        - projected
        - secret
        EOF
```

What to do next

[Deploy Db2 or Db2 Warehouse with the custom service account, SCC, role, and role binding](#) that you created.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Specifying a custom service account, SCC, role, and role binding before deploying Db2 or Db2 Warehouse

If you are using a custom service account, security context constraints (SCC), role, and role binding for the Db2® or Db2 Warehouse service, you need to take steps outside of the Cloud Pak for Data web console to prepare for deploying Db2 or Db2 Warehouse.

Procedure

1. Set the Db2 or Db2 Warehouse operator replica to 0 by running the following command:

   ```
   oc patch deployment ibm-db2oltp-cp4d-operator-controller-manager --patch '{"spec": {"replicas": 0}}' -n operator_namespace
   ```

   Where *operator_namespace* is the Red Hat® OpenShift® project (namespace) that contains the Db2 or Db2 Warehouse operator.

2. Open the db2u-json-cm `ConfigMap` object in edit mode:

   ```
   oc edit cm db2u-json-cm -n=namespace
   ```

3. Type /service-account to locate the `service-account` section of the `ConfigMap`.
4. Under the `ibm-db2oltp.json` data option, change the service account name from `db2u` to the name of the custom service account that you created in [Manually creating an SCC, service account, role, and role binding](#).
   The entry that you change should look similar to the following example:

   ```
   \"service-account\": {\n        \"name\": \"db2u\"\n    }
   ```

5. Save the change and exit the `ConfigMap`.
6. Delete the `zen-database-core` pod in order for the `ConfigMap` changes to be re-mounted to the volume.

   ```
   oc delete po $(oc get po -n=namespace | grep zen-database-core | awk {'print $1'}) -n=namespace
   ```

   Note: If you reinstall or upgrade the Db2 or Db2 Warehouse service, you must perform these steps again.
7. [Deploy the Db2 service](#) or [Deploy the Db2 Warehouse service](#).
8. Set the Db2 or Db2 Warehouse operator replica to 1 by running the following command:

   ```
   oc patch deployment ibm-db2oltp-cp4d-operator-controller-manager --patch '{"spec": {"replicas": 1}}' -n cpd_operator_namespace
   ```

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

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.

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 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	- OpenPages® - Watson™ Discovery - Watson Knowledge Catalog - 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	- Data Virtualization - Db2® - Db2 Big SQL - Db2 Warehouse - Watson Discovery - Watson Knowledge Catalog - Watson Studio	All environments except IBM Cloud	CRI-O container settings
Kernel parameter settings	- Data Virtualization - Db2 - Db2 Big SQL - Db2 Warehouse - Watson Knowledge Catalog - Watson Studio	All environments	- Kernel parameter settings - Deploying Db2 with limited privileges - Deploying Db2 Warehouse with limited privileges - Using the Red Hat® OpenShift® Node Tuning Operator to set kernel parameters - Updating kernel semaphore settings - Db2 Big SQL - Preparing to install the service (Data Virtualization)
Power settings			Power settings
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. If you are using HAProxy, the load balancer node is usually the OpenShift cluster public node.

The recommended timeout is at least 5 minutes (300 seconds).

This setting is required if you plan to install the Watson Knowledge Catalog service or the OpenPages service. For more information about timeout settings in Watson Knowledge Catalog, see Processes time out before completing, and for timeout settings in OpenPages, see Setting up the cluster for OpenPages.

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 load balancer other than HAProxy, see the documentation for your load balancer for information about how to configure the timeout.

Changing timeout settings 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
   ```

Changing timeout settings 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 descriptor files.

These settings are required for the CRI-O CRI-O container runtime on the OpenShift Container Platform.

Note: If you install Cloud Pak for Data on IBM Cloud, the CRI-O container settings are automatically applied to your cluster as part of the installation. You do not need to manually change these settings.
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 worker node. For example, run the following command, replacing *$node* with one of the worker nodes. You can obtain the worker nodes by using the `oc get nodes` command.

   ```
   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 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.

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.
   Note: If you are using Cloud Pak for Data on OpenShift Container Platform version 4.6, the ignition version is 3.1.0. If you are using Cloud Pak for Data on OpenShift Container Platform version 4.8, change the ignition version to 3.2.0.

   ```
   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,$(cat /tmp/crio.conf | base64 -w0)
           filesystem: root
           mode: 0644
           overwrite: true
           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

Enabling unsafe sysctls

Configure `kubelet` to allow Db2U to make unsafe sysctl calls for Db2 to manage required memory settings. For more information, see Enabling unsafe sysctls in the OpenShift documentation.
Note: This procedure applies to on-premises and private cloud deployments of Cloud Pak for Data. It does not apply to Cloud Pak for Data deployments on IBM Cloud.

1. 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
```

2. Update the label on the `machineconfigpool`:

```
oc label machineconfigpool worker db2u-kubelet=sysctl
```

3. Wait for the cluster to restart and 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   DEGRADEDMACHINECOUNT
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.

Changing kernel parameter settings on IBM Cloud

Note: If you install Cloud Pak for Data and services on IBM Cloud by using the IBM Cloud Catalog, the kernel parameter settings are automatically applied to your cluster as part of the installation. You do not need to manually change these settings.
If you do not use the IBM Cloud Catalog to install Cloud Pak for Data and services on IBM Cloud, you must manually change the kernel parameter settings by applying a custom Kubernetes daemon set. For more information, see Modifying default worker node settings to optimize performance in the IBM Cloud documentation. Update the values in the daemon set based on the recommended settings for Cloud Pak for Data. For more information, see Kernel parameter requirements (Linux).

Changing kernel parameter settings in environments that do not support `kubelet` settings

Some environments do not support setting `kubelet` settings. In these cases you can verify the kernel parameters to ensure that certain services can run correctly. You can use the Red Hat OpenShift Node Tuning Operator to calculate the correct kernel parameters. For more information, see Using the Red Hat OpenShift Node Tuning Operator to set kernel parameters.

Power settings

On Power® Systems, you must complete the following steps to change the simultaneous multithreading (SMT) settings and set the kernel argument slub_max_order to 0 for small core, Kernel-based Virtual Machine (KVM) capable (LC922, IC922, AC922) systems, and big core, PowerVM® capable systems (L922, E950, E980, S922).
Note: You need to set the kernel argument slub_max_order to **0** only if your OpenShift Container Platform version is earlier than 4.8. Remove the kernel argument setting from the YAML file if your OpenShift Container Platform version is 4.8 or later.

1. Label all small core KVM capable worker nodes that are *not* running Db2 Warehouse workloads to SMT=**2**. For example:

```
oc label node <node> SMT=2 --overwrite
```

2. Label all small core KVM capable worker nodes that *are* running Db2 Warehouse workloads to SMT=**4**. For example:

```
oc label node <node> SMT=4 --overwrite
```

3. Label all big core PowerVM capable worker nodes that are *not* running Db2 Warehouse workloads to SMT=**4**. For example:

```
oc label node <node> SMT=4 --overwrite
```

4. Label all big core PowerVM capable worker nodes that *are* running Db2 Warehouse workloads to SMT=**8**. For example:

```
oc label node <node> SMT=8 --overwrite
```

5. Create a YAML file, smt.yaml, with the following content:

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 99-worker-smt
spec:
  kernelArguments:
  - slub_max_order=0
  config:
    ignition:
      version: 3.1.0
    storage:
      files:
      - contents:
          source: data:text/plain;charset=utf-
```
```
8;base64,IyEvYmluL2Jhc2gKKZXhwb3J0IFBBVEg9L3Jvb3QvLmxvY2FsL2Jpbjovcm9vdC9iaW46L3NiaW46L2JpbjovdXNyL2xvY2FsL3NiaW46L3Vzci9sb
2NhbC9iaW46L3Vzci9zYmluOi91c3IvYmluCmV4cG9ydCBLVUJFQ09ORklHPS92YXIvbGliL2t1YmVsZXQva3ViZWNvbmZpZwpDT1JFUFM9JCgvYmluL2xzY3B
1IHwgL2Jpbi9hd2sgLUYICCgJDEgfiAvXkNvcmVcKHNcKSBwZXIgc29ja2V0JC8ge3ByaW50ICQyfSd8L2Jpbi94YXJncykKU09DS0VUUz0kKC9iaW4vbHNjc
HUgfCAvYmluL2F3ayAtRjogJyAkMSB+IC9eU29ja2V0XChzXCkkLyB7cHJpbnQgJDJ9J3wvYmluL3hhcmdzKQpsZXQgVE9UQUxDT1JFUz0kQO9SRVBTKiRTT0N
LRVRTCk1BWFRIUkVBRFM9JCgvYmluL2xzY3B1IHwgL2Jpbi9hd2sgLUY6ICcgJDEgfiAvXkNQVVwoc1wpJC8ge3ByaW50ICQyfSd8L2Jpbi94YXJncykKbGV0I
E1BWFNNVD0kTUFYVEhSRUFEUy8kVE9UQUxDT1JFVwpDVVJSRU5U01UPSQoL2Jpbi9sc2NwdSB8IC9iaW4vYXdrIC1GOiAnICQxIH4gL15UaHJlYWRcKHNcKSB
wZXIgY29yZSQvIHtwcmludCAkMn0nfC9iaW4veGFyZ3MpCgpTTVRMQUJFTD0kKC9iaW4vb2MgZ2V0IG5vZGUgJEhPU1ROQU1FIC1MIFNNVCAtLW5vLWhlYWRlc
nMgfC9iaW4vYXdrICd7cHJpbnQgQJDZ9JykKaWYgW1sgLW4gJFNNVExBQkVMIF1dCiAgdGhlbgogICAgY2FzZSAkU01UEFCRUwgaW4KICAgICAgMSkgVEFSR0V
UU01UPTEKICAgIDs7CiAgICAgIDIpIFRBUkdFVFNNVD0yCiAgICA7OwogICAgICA0KSBUQVJHRVRTTVQ9NAogICAgOzsKICAgICAgOCkgVEFSR0VUU01UPTgKI
CAgIDs7CiAgICAgICopIFRBUkdFVFNNVD0kQ1VSUkVOVFNNVCA7IGVjaG8g8gIlNNVCB2YWx1ZSBtdXN0IG1lIDEsIDIsIDQsIG9yIDggYW5kIHNtYWxsZXIgdGh
hbiBNYXhpbXVtIFNNVC4iCiAgICA7OwogICAgZXNhYwogIGVsc2UKICAgIFRBUkdFVFNNVD0kTUFYU01UCmZpCgpDVVJSRU5UU01UPSQoL2Jpbi9sc2NwdSB8I
C9iaW4vYXdrIC1GOiAnICQxIH4gL15UaHJlYWRcKHNcKSBwZXIgY29yZSQvIHtwcmludCAkMn0nfC9iaW4veGFyZ3MpCgppZiBbWyAkQ1VSUkVOVFNNVCAtbmU
gJFRBUkdFVFNNVCBdXQogIHRoZW4KICAgIElOSVRPTllRIUkVBRD0wCiAgICBJTkllUT0ZGVhSRUFEPSRUQVJHRVRTTVQKICAgIGlmIFtbICRNQVhTTVQgLWdlI
CRUQVJHRVRTTVQgLXV0KICAgICAgdGhlbgogICAgICAgIHdoaWxlIFtbICRJTkllUT05USFJFQUQgLWx0ICRNQVhSSFJFQURTIF1dCiAgICAgICAgZG8KICAgICA
```

```
gICAgIE9OVEhSRUFEPSRJTklUT05USFJFQUQKICAgICAgICAgIE9GRlRIUkVVBRD0kSU5JVE9GRlRIUkVVBRAoKICAgICAgICAgIHdoaWxlIFtbICRTUTlRIUkVVBR
CAtbHQgJE9GRlRIUkVVBRCBdXQogICAgICAgICAgZG8KICAgICAgICAgICAgICAgICAgICAgICAgICL2Jpbi9lY2hvIDEgPiAvc3lzL2RldmljZXMvc3lzdGVtL2NwdS9jHUkT05USFJ
FQUQvb25saW5lCiAgICAgICAgICAgIGxldCBPT05USFJFUkVVBRD0kT05USFJFQUQrMQogICAgICAgICAgZG9uZQogICAgICAgICAgbGV0IElOSVRPRT1RIUkVVBRD0kS
U5JVE9OVEhSRUFEKyRNQVhPThTTVQKICAgICAgICAgIHdoaWxlIFtbICRPRkZUSFJFQUQgLWx0ICRJTklUT05USFJFQUQgXV0KICAgICAgICAgICAgICAgICAgICAgIGRvCiAgICAgICA
gICAgIC9iaW4vZWNvbobyAwID4gL3N5cy9kZXZpY2VzL3N5c3Rlb9S9jcHUvY3B1JE9GRlRIUkVVBRC9vbmxpbmUKICAgICAgICAgICAgbGV0IE9GRlRIUkVVBRD0kT
0ZGVEhSRUFEKzEKICAgICAgICAgIGRvbmUKICAgICAgICAgIGxldCBJTklUT0ZGVEhSRUFEPSRJTklUT0ZGVEhSRUFEKyRNQVhPThTTVQKICAgICAgICAgIGkb251CiA
gICAgIGVsc2UKICAgICAgICBlY2hvICJUYXJnZXQgU01UIG11c3QgYmUgc21hbGxlciBvciBlcXVhbCB0aGFuIE1heGltdW0gU01UIHN1cHBvcnRlZCIKICAgI
GZpCmZp
```
```
              verification: {}
            filesystem: root
            mode: 0755
            overwrite: true
            path: /usr/local/bin/powersmt
    systemd:
      units:
        - name: smt.service
          enabled: true
          contents: |
            [Unit]
            Description=Set SMT
            After=network-online.target
            Before= crio.service
            [Service]
            Type=oneshot
            RemainAfterExit=yes
            ExecStart=/usr/local/bin/powersmt
            [Install]
            WantedBy=multi-user.target
```

6. Run the **oc create** command to apply the changes.

   Note: You must ensure that the cluster master nodes (or control plane) are in Ready status before you issue this command.

   `oc create -f smt.yaml`

   Your worker nodes will perform a rolling reboot action to update the kernel argument slub_max_order and set the labeled SMT level.
   Note:
   - All the worker nodes are rebooted after the command is issued. The `slub_max_order=0` kernel argument and the specified SMT level are applied to all the worker nodes after the reboot completes. The SMT level on the worker nodes that are not labeled will be set to the default value.
   - After this process is done, if the SMT level on a particular worker node needs to be changed, you must label that worker node with the desired SMT level and manually reboot it.

- **[Using the Red Hat OpenShift Node Tuning Operator to set kernel parameters](#)**
  You can use the Red Hat OpenShift Node Tuning Operator to set IPC kernel parameters that are required to deploy Db2 or Db2 Warehouse on Cloud Pak for Data.

Previous topic: [Custom security context constraints for services](#)
Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Using the Red Hat OpenShift Node Tuning Operator to set kernel parameters

You can use the Red Hat® OpenShift® Node Tuning Operator to set IPC kernel parameters that are required to deploy Db2® or Db2 Warehouse on Cloud Pak for Data.

Before you begin

1. Disable the automatic IPC tuning mechanism by following the steps in [Configuring Db2 Warehouse to disable automatic setting of kernel parameters](#).
2. Decide whether to use [dedicated nodes](#). With dedicated deployments, you can control which cluster nodes the database pods can be scheduled on. When you use dedicated notes, you can limit node tuning to the dedicated nodes.

About this task

The Node Tuning Operator helps you manage node-level tuning by orchestrating the tuned daemon. Tuned is a system tuning service for Linux®. The core of Tuned are profiles, which tune your system for different use cases. In addition to static application of system settings, Tuned can also monitor your system and optimize the performance on-demand based on the profile that is applied.

Tuned is distributed with a number of predefined profiles. However, it is also possible to modify the rules defined for each profile and customize how and what to tune. Tuned supports various types of system configuration such as sysctl, sysfs, and kernel boot parameters. For more information, see [Monitoring and managing system status and performance](#) and [The Tuned Project](#).

The Node Tuning Operator provides a unified management interface to users of node-level **sysctls** and gives more flexibility to add custom tuning.

The operator manages the containerized tuned daemon for Red Hat OpenShift Container Platform as a Kubernetes DaemonSet. It ensures the custom tuning specification is passed to all containerized tuned daemons that run in the cluster in the format that the daemons understand. The daemons run on all nodes in the cluster, one per node.

The Node Tuning Operator is part of a standard Red Hat OpenShift Container Platform installation. For full documentation, see [Using the Node Tuning Operator](#).

Procedure

You can employ the Node Tuning Operator either by a creating a custom resource definition (CRD) file that is based on the YAML file that is provided here, or by using the provided sample shell script. The CRD method requires you to manually compute all required IPC kernel parameters; the shell script enables you to generate a YAML file that you can install, deploy, and run on the target OpenShift cluster.

- **To create a Custom Resource Definition file**
  The following sample YAML file describes the basic structure that is needed to create the CRD for a Node Tuning Operator instance that can tune IPC kernel parameters.

  Important: The following sample file is for Db2 databases. If you are deploying Db2 Warehouse, replace database-db2oltp with database-db2wh .

```
apiVersion: tuned.openshift.io/v1
kind: Tuned
metadata:
  name: db2u-ipc-tune
  namespace: openshift-cluster-node-tuning-operator
spec:
  profile:
  - name: openshift-db2u-ipc
    data: |
      [main]
      summary=Tune IPC Kernel parameters on OpenShift nodes running Db2U engine PODs
      include=openshift-node

      [sysctl]
      kernel.shmmni = <shmmni>
      kernel.shmmax = <shmmax>
      kernel.shmall = <shmall>
      kernel.sem = <SEMMSL> <SEMMNS> <SEMOPM> <SEMMNI>
      kernel.msgmni = <msgmni>
      kernel.msgmax = <msgmax>
      kernel.msgmnb = <msgmnb>

  recommend:
  - match:
    - label: node-role.kubernetes.io/worker
    - label: icp4data
      value:  database-db2oltp
    priority: 10
    profile: openshift-db2u-ipc
```

  - You need to compute the values for the IPC kernel parameters that are denoted by **<**
    ... > based on the formulas in <u>Kernel parameter requirements (Linux)</u>.
  - **Important:** Use the `memory.resource` limit that you plan to apply to the deployment as `size of RAM` if your Kubernetes worker node pool is heterogeneous.
  - The inheritance option `include=openshift-node` is included to implement the inheritance chain `openshift-db2u-ipc <- openshift-node <- openshift <- virtual-host`. You inject our IPC sysctl changes on top of the OpenShift pre-installed tuned profile settings for worker nodes.
  - The match label `icp4data` and the corresponding value is only required for dedicated deployments. In that case the IPC kernel tuning is applied only on the labeled worker nodes.

  Save the finalized CRD into a YAML file, for example as `/tmp/Db2UnodeTuningCRD.yaml` on the OpenShift cluster master node. Log in to the same cluster with a user ID that has the cluster admin role and run the following command to create the CR:

```
oc create -f /tmp/Db2UnodeTuningCRD.yaml
```

  It might take a minute or so for the CRD to be created and the custom IPC tuned profile to become active and applied on the worker nodes.

- **To use the sample shell script**
  Instead of manually computing the required IPC kernel parameters and then generating and installing the Node Tuning Operator CRD, the following sample shell script can be used to:

  - Generate a YAML file that you can install and deploy and run on the target OpenShift cluster.
  - Delete the CRD and clean up deployed tuned profiles.

  The sample assumes that the script is saved as /root/script/crtNodeTuneCRD.sh.

```
#!/bin/bash

# Compute IPC kernel parameters as per IBM Documentation topic
# https://www.ibm.com/support/knowledgecenter/SSEPGG_11.1.0/com.ibm.db2.luw.qb.server.doc/doc/c0057140.html
# and generate the Node Tuning Operator CRD yaml.

tuned_crd_yaml="/tmp/Db2UnodeTuningCRD.yaml"
mem_limit_Gi=0
node_label=""
cr_name="db2u-ipc-tune"
cr_profile_name="openshift-db2u-ipc"
cr_namespace="openshift-cluster-node-tuning-operator"
create_cr="false"
delete_cr="false"

usage() {
    cat <<-USAGE #| fmt
    Usage: $0 [OPTIONS] [arg]

    OPTIONS:
    =======
    * -m|--mem-limit mem_limit   : The memory.limit (Gi) to be applied to Db2U deployment.
    * [-l|--label node_label]    : The node label to use for dedicated Cp4D deployments.
    * [-f|--file yaml_output]    : The NodeTuningOperator CRD YAML output file. Default /tmp/Db2UnodeTuningCRD.yaml.
    * [-c|--create]              : Create the NodeTuningOperator CR ${cr_name} using the generated CRD yaml file.
    * [-d|--delete]              : Delete the NodeTuningOperator CR ${cr_name}.
    * [-h|--help]                : Display the help text of the script.
USAGE
}

[[ $# -lt 1 ]] && { usage && exit 1; }
```

```
while [[ $# -gt 0 ]]; do
    case "$1" in
        -f|--file) shift; tuned_crd_yaml=$1
        ;;
        -m|--mem-limit) shift; mem_limit_Gi=$1
        ;;
        -l|--label) shift; node_label=$1
        ;;
        -c|--create) create_cr="true"
        ;;
        -d|--delete) delete_cr="true"
        ;;
        -h|--help) usage && exit 0
        ;;
        *) usage && exit 1
        ;;
    esac
    shift
done

((ram_in_BYTES=mem_limit_Gi * 1073741824))
((ram_GB=ram_in_BYTES / (1024 * 1024 * 1024)))
((IPCMNI_LIMIT=32 * 1024))
tr ' ' '\n' < /proc/cmdline | grep -q ipcmni_extend && ((IPCMNI_LIMIT=8 * 1024 * 1024))

#
### =============== functions ================ ###
#
# Compute the required kernel IPC parameter values
compute_kernel_ipc_params() {
    local PAGESZ=$(getconf PAGESIZE)

    # Global vars
    ((shmmni=256 * ram_GB))
    shmmax=${ram_in_BYTES}
    ((shmall=2 * (ram_in_BYTES / PAGESZ)))
    ((msgmni=1024 * ram_GB))
    msgmax=65536
    msgmnb=${msgmax}
    SEMMSL=250
    SEMMNS=256000
    SEMOPM=32
    SEMMNI=${shmmni}

    # RH bugzilla https://access.redhat.com/solutions/4968021. Limit SEMMNI, shmmni and msgmni to the max
    # supported by the Linux kernel -- 32k (default) or 8M if kernel boot parameter 'ipcmni_extend' is set.
    ((SEMMNI=SEMMNI < IPCMNI_LIMIT ? SEMMNI : IPCMNI_LIMIT))
    ((shmmni=shmmni < IPCMNI_LIMIT ? shmmni : IPCMNI_LIMIT))
    ((msgmni=msgmni < IPCMNI_LIMIT ? msgmni : IPCMNI_LIMIT))
}

# Generate NodeTuning Operator YAML file
gen_tuned_crd_yaml() {
    # Generate YAML file for NodeTuning CRD and save as ${tuned_crd_yaml}
    cat <<-EOF > ${tuned_crd_yaml}
apiVersion: tuned.openshift.io/v1
kind: Tuned
metadata:
  name: ${cr_name}
  namespace: ${cr_namespace}
spec:
  profile:
  - name: ${cr_profile_name}
    data: |
      [main]
      summary=Tune IPC Kernel parameters on OpenShift nodes running Db2U engine PODs
      include=openshift-node

      [sysctl]
      kernel.shmmni = ${shmmni}
      kernel.shmmax = ${shmmax}
      kernel.shmall = ${shmall}
      kernel.sem = ${SEMMSL} ${SEMMNS} ${SEMOPM} ${SEMMNI}
      kernel.msgmni = ${msgmni}
      kernel.msgmax = ${msgmax}
      kernel.msgmnb = ${msgmnb}

  recommend:
  - match:
    - label: node-role.kubernetes.io/worker
EOF

    # Add the optional dedicated label into match array
    if [[ -n "${node_label}" ]]; then
        cat <<-EOF >> ${tuned_crd_yaml}
    - label: icp4data
      value: ${node_label}
EOF
    fi

    # Add the priority and profile keys
    cat <<-EOF >> ${tuned_crd_yaml}
    priority: 10
    profile: ${cr_profile_name}
EOF
```

```
    [[ "${create_cr}" == "true" ]] && return
    cat <<-MSG
    =============================================================================
    * Successfully generated the Node Tuning Operator Custom Resource Definition as
      ${tuned_crd_yaml} YAML with Db2U specific IPC sysctl settings.

    * Please run 'oc create -f ${tuned_crd_yaml}' on the master node to
      create the Node Tuning Operator CR to apply those customized sysctl values.
    =============================================================================
    MSG
}

create_tuned_cr() {
    echo "Creating the Node Tuning Operator Custom Resource for Db2U IPC kernel parameter tuning ..."
    oc create -f ${tuned_crd_yaml}
    sleep 2

    # List the NodeTuning CR and describe
    oc -n ${cr_namespace} get Tuned/${cr_name}
    echo ""

    echo "The CRD of the Node Tuning Operator deployed"
    echo "-----------------------------------------"
    oc -n ${cr_namespace} describe Tuned/${cr_name}
    echo ""
}

delete_tuned_cr() {
    echo "Deleting the Node Tuning Operator Custom Resource used for Db2U IPC kernel parameter tuning ..."
    oc -n ${cr_namespace} get Tuned/${cr_name} --no-headers -ojsonpath='{.kind}' | grep -iq tuned || \
        { echo "No matching CR found ..." && exit 0; }
    oc -n ${cr_namespace} delete Tuned/${cr_name}
    echo ""
    sleep 2

    # Get the list of containerized tuned PODs (DaemonSet) deployed on the cluster
    local tuned_pods=( $(oc -n ${cr_namespace} get po --selector openshift-app=tuned --no-headers -
ojsonpath='{.items[*].metadata.name}') )
    # Remove the tuned profile directory deployed on those PODs
    for p in "${tuned_pods[@]}"; do
        echo "Removing the installed tuned profile ${cr_profile_name} on POD: $p"
        oc -n ${cr_namespace} exec -it $p -- bash -c "rm -fr /etc/tuned/${cr_profile_name}"
    done
    echo ""
}

#
### ================= Main ==================== ###
#

[[ "${delete_cr}" == "true" ]] && { delete_tuned_cr && exit 0; }

compute_kernel_ipc_params

gen_tuned_crd_yaml

[[ "${create_cr}" == "true" ]] && create_tuned_cr
```

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`.

Before you begin

Ensure that the cluster meets the minimum requirements for installing the scheduling service. For details, see System requirements.

Verify that you completed the required Pre-installation tasks for your environment. Specifically, verify that:

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 catalog source exists. For details, see Creating catalog sources.
5. 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

Complete the following tasks to install the scheduling service:

1. Installing the scheduling service
2. Verifying the installation
3. What to do next

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
     namespace: ibm-common-services      # Specify the namespace where IBM Cloud Pak foundational services is installed.
   spec:
     version: 1.3.2
     license:
       accept: true
     registry: cp.icr.io/cp/cpd
     releasename: ibm-cpd-scheduler
   EOF
   ```

   When you create the custom resource, the scheduling service operator installs the 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 the scheduling service, including `Scheduling`. (The `Scheduling` microservice is defined by the `ibm-cpd-scheduler` custom resource.) 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.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see Documentation for previous 4.0.x refreshes.

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.
Information you need to complete this task

- The Cloud Pak for Data control plane needs only the `restricted` security context constraint (SCC).
- The Cloud Pak for Data control plane 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`
    - **IBM Spectrum® Scale Container Native:** `ibm-spectrum-scale-sc`
    - **NFS:** `managed-nfs-storage`
    - **Portworx:** `portworx-shared-gp3`
    - **IBM Cloud File Storage:** `ibmc-file-gold-gid` or `ibm-file-custom-gold-gid`

Important: The steps in this task assume that you are installing the latest software images.

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. The project where you plan to install Cloud Pak for Data exists. For details, see Setting up projects (namespaces) on Red Hat OpenShift Container Platform.
3. 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 private container registry.
4. The cluster is configured to pull the software images. For details, see Configuring your cluster to pull Cloud Pak for Data images.
5. The Cloud Pak for Data platform operator catalog source exists. For details, see Creating catalog sources.
6. The Cloud Pak for Data platform operator subscription and the `IBM NamespaceScope Operator` subscription exist. For details, see Creating operator subscriptions.

If you do not complete these steps, the Cloud Pak for Data installation will fail.
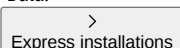
Procedure

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.

        For example, if you plan to install Cloud Pak for Data in the `cpd-instance` project, specify that project in the `namespace` entry:

```
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:
  csvInjector:                       # This setting is required for some services. Do not delete this line if you
specified it when you created operator subscriptions.
    enable: true                     # This setting is required for some services. Do not delete this line if you
specified it when you created operator subscriptions.
  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. Follow the appropriate guidance for your environment:

   Important: By creating an **Ibmcpd** custom resource with **spec.license.accept: true**, you are accepting the license terms for Cloud Pak for Data. You can find links to the relevant licenses in Cloud Pak for Data License Information.

   - > The cluster uses the recommended storage class names on Red Hat OpenShift Container Storage

     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: 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
  storageVendor: ocs
EOF
```

   - > The cluster uses the recommended storage class names on Portworx

     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: 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
  storageVendor: portworx
EOF
```

   - > The cluster uses IBM Spectrum Scale Container Native storage

     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: 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: ibm-spectrum-scale-sc
EOF
```

- > The cluster uses NFS storage

  The recommended storage class names are described in [Setting up shared persistent storage](#).

  ```
  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, such as
  managed-nfs-storage
  EOF
  ```

- > The cluster uses custom storage classes

  If your cluster uses storage class names other than those described in [Setting up shared persistent storage](#), you must tell Cloud Pak for Data what storage class names to use.

  ```
  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 # Replace with the name of a RWO storage class that points to
  block storage
  EOF
  ```

  Important:

  Configuring metadata storage

  If your cluster does not use the recommended storage classes, you must configure metadata storage for Cloud Pak for Data by specifying a storage class for the `zenCoreMetadbStorageClass` setting. The storage class must be ReadWriteOnce (RWO) and point to block storage, such as:
  - `ocs-storagecluster-ceph-rbd` on Red Hat OpenShift Container Storage
  - `portworx-metastoredb-sc` on Portworx

Verifying the installation

When you create the `Ibmcpd` custom resource, the IBM Cloud Pak for Data platform operator processes the contents of the custom resource and starts up the microservices that comprise Cloud Pak for Data, including the Cloud Pak for Data control plane.

The Cloud Pak for Data control plane is defined by the `ZenSerivce lite-cr` custom resource.

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 installation:
   a. Run the following command to determine whether the `ibmcpd-cr` has been created:

   ```
   oc get Ibmcpd ibmcpd-cr -o jsonpath="{.status.controlPlaneStatus}{'\n'}"
   ```

Output	What to do next
InProgress	Wait a few minutes. Then, run the command again.
Completed	Go to step 2b.
Failed	Go to step 2b.

   b. Run the following command to determine whether the control plane is ready:

   ```
   oc get ZenService lite-cr -o jsonpath="{.status.zenStatus}{'\n'}"
   ```

Output	What to do next
InProgress	Wait a few minutes. Then, run the command again. It can take up to 90 minutes for the command to return `Completed`. If the command still has not returned `Completed` after 90 minutes, contact IBM Software Support.
Completed	The Cloud Pak for Data control plane is ready when the command returns `Completed`. Go to step 3.
Failed	Contact IBM Software Support.

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.

Choosing an upgrade plan for the Cloud Pak for Data control plane

Automatic upgrade (recommended)
By default, the Cloud Pak for Data control plane will be automatically upgraded when you install a newer version of IBM Cloud Pak foundational services on the cluster.
If you want to continue using the automatic upgrade plan, no additional action is required.

Note: For this release of Cloud Pak for Data, the default version of the control plane is 4.4.0. If you install the minimum required version of IBM Cloud Pak foundational services (Version 3.15), the control plane will remain at 4.4.0. If you install a later release of IBM Cloud Pak foundational services, you might see a later version of the control plane on your cluster.
Manual upgrade
If you want to manually upgrade the Cloud Pak for Data control plane when you install a newer version of IBM Cloud Pak foundational services, you can optionally *pin* the installation to a specific version.
To change to the manual upgrade plan:

1. Update the `ZenService` custom resource.
For a list of operand versions supported by the `Zen` operator, see [Cloud Pak for Data operator and operand versions](#).

For example, to pin the installation at 4.4.0, run the following command:

```
oc patch ZenService lite-cr \
--namespace <cpd-instance> \
--type=merge \
--patch '{"spec": {"version":"4.4.0"}}'
```

What to do next

Ensure that you complete the appropriate [Post-installation tasks](#) for your environment.

For example:

- Replace the self-signed TLS certificate with your own TLS certificate. For details, see [Using a custom TLS certificate for HTTPS connections to the platform](#).
- If you created operators with the manual installation plan, review [Specifying the install plan for operators that are automatically installed by Operand Deployment Lifecycle Manager](#).
- If you want to change the route to the platform, follow the guidance in [Changing the route to the platform](#).

Related reference

- [Pre-installation tasks](#)
- [Post-installation tasks](#)
- [Uninstalling the common core services](#)
- [Uninstalling Cloud Pak for Data](#)

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Post-installation tasks

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

- **[Specifying the install plan for operators that are automatically installed by Operand Deployment Lifecycle Manager](#)**
When you create a Cloud Pak for Data software operator subscription, you can specify whether Red Hat OpenShift automatically loads newer versions of the operator or whether it creates an update request that must be approved by a cluster administrator. However, any operators that are automatically installed by another Cloud Pak for Data software operator are created with the *automatic* install plan.
- **[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.
- **[Making monitoring data highly available](#)**
IBM Cloud Pak for Data monitoring data is stored in an embedded InfluxDB database. By default, only one replica of the database deployed. However, if you want to ensure that the monitoring data is highly available, you can increase the number of replicas.
- **[Changing the route to the platform](#)**
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`. IBM Cloud Pak for Data supports two different methods for changing the route to the platform.
- **[Configuring an external route to the Flight Server](#)**
The Flight Server is a data connection service that enables assets, such as notebooks, to interact with various data sources without calling the REST APIs for the data sources. By default, the Flight Server is only available to the IBM Cloud Pak for Data instance where the Flight Server is running. However, a Red Hat OpenShift Container Platform project administrator can create an external route to the Flight Server to enable other applications to interact with it.

- **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 the common core services
- Uninstalling Cloud Pak for Data

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see Documentation for previous 4.0.x refreshes.

Specifying the install plan for operators that are automatically installed by Operand Deployment Lifecycle Manager

When you create a Cloud Pak for Data software operator subscription, you can specify whether Red Hat® OpenShift® automatically loads newer versions of the operator or whether it creates an update request that must be approved by a cluster administrator. However, any operators that are automatically installed by another Cloud Pak for Data software operator are created with the *automatic* install plan.

For example, if you install Watson™ Studio, Watson Studio automatically installs the following dependencies if they are not already available:

- Watson Studio Runtimes
- Common core services
- Data Refinery

By default, the operators for these dependencies are created with the automatic install plan (`installPlanApproval: Automatic`).

If you want to use the manual install plan to ensure that all aspects of the software remain at the same version until you are ready to upgrade the software, you can edit the `ZenService` custom resource to add the `installPlanApproval` parameter. This parameter ensures that any dependent operator subscriptions that are created by the Operand Deployment Lifecycle Manager are created with the specified install plan. (If you do not set this parameter, the automatic install plan is used.)

Permissions you need for this task
You must be either:

- A cluster administrator
- An administrator of the Red Hat OpenShift project (namespace) where Cloud Pak for Data is installed.

When you need to complete this task
It is recommended that you complete this task after you install the Cloud Pak for Data control plane but before you install any services.
However, you can also complete this task after you install services. In this situation, the automatically created operators will continue to use the automatic install plan unless you complete one of the following tasks:

- Upgrade the Cloud Pak for Data services.
  When you upgrade the service, Operand Deployment Lifecycle Manager changes the install plan on the operators that are created for the dependencies.

- Manually edit the operand registries to change the install plan.
  When you edit the operand registry, the Operand Deployment Lifecycle Manager changes the install plan on the operators that are created for the dependencies.

Procedure

To specify the install plan for automatically installed operators:

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 `installPlanApproval` parameter to the `ZenService` custom resource and set the install plan to Manual.
   The command assumes that Cloud Pak for Data is installed in the `cpd-instance` project. If Cloud Pak for Data is installed in a different project, update the `--namespace` parameter with the appropriate project for your environment.

   ```
   oc patch ZenService lite-cr \
   --namespace cpd-instance \
   --type=merge \
   --patch '{"spec": {"installPlanApproval":"Manual"}}'
   ```

3. Check the status of the `ZenService` custom resource.
   The command assumes that Cloud Pak for Data is installed in the `cpd-instance` project. If Cloud Pak for Data is installed in a different project, update the `--namespace` parameter with the appropriate project for your environment.

   ```
   oc get ZenService lite-cr \
   --namespace cpd-instance \
   -o jsonpath="{.status}"
   ```

   Wait for the command to return `Completed`.

What to do next

The next step depends on your environment:

- If you completed this task after you installed Cloud Pak for Data but before you installed any services, Operand Deployment Lifecycle Manager will use the install plan when it creates dependent operators. You can install the appropriate [services](#) on your cluster.
- If you completed this task after you installed services, choose the appropriate action for your environment:
  - If you need to update the services on your cluster, follow the guidance for upgrading the [services](#) on your cluster.
  - If you don't plan to update the services soon and you want the change to take effect immediately, edit the operand registries to set `installPlanApproval: Manual`.

If you need to edit the operand registries for the changes to take effect:

1. Run the following command to determine the location of all of the operand registries on the cluster:

   `oc get operandRegistry -A`

   The command returns output with the following format:

   - > **Express installations**

     ```
     NAMESPACE             NAME                           AGE    PHASE     CREATED AT
     ibm-common-services   common-service                 37d    Running   2021-10-15T05:17:36Z
     ibm-common-services   ibm-cpd-ccs-registry           29d    Running   2021-10-23T16:25:06Z
     ibm-common-services   ibm-cpd-datarefinery-registry  28d    Running   2021-10-24T00:35:59Z
     ibm-common-services   ibm-cpd-ws-runtimes-registry   28d    Running   2021-10-24T00:46:30Z
     ```

   - > **Specialized installations**

     ```
     NAMESPACE             NAME                           AGE    PHASE     CREATED AT
     ibm-common-services   common-service                 37d    Running   2021-10-15T05:17:36Z
     cpd-operators         ibm-cpd-ccs-registry           29d    Running   2021-10-23T16:25:06Z
     cpd-operators         ibm-cpd-datarefinery-registry  28d    Running   2021-10-24T00:35:59Z
     cpd-operators         ibm-cpd-ws-runtimes-registry   28d    Running   2021-10-24T00:46:30Z
     ```

2. Run the appropriate commands based on your environment:

   - > **Express installations**

     - Edit the operand registries in the `ibm-common-services` project:

       `oc edit operandRegistry -n ibm-common-services`

     - For each entry in the `operators` section, ensure that the operator has `installPlanApproval: Manual` set. If an entry does not have the `installPlanApproval` parameter, add it.
       For example:

       ```
       - channel: v1.0
           installPlanApproval: Manual
         name: ibm-cpd-ccs-operator
         namespace: ibm-common-services
         packageName: ibm-cpd-ccs
         scope: public
         sourceNamespace: openshift-marketplace
       ```

     - Save your changes and exit. For example, if you are using `vi`, hit esc and enter :wq.

   - > **Specialized installations**

     - Edit the operand registries in the `cpd-operators` project:

       `oc edit operandRegistry -n cpd-operators`

     - For each entry in the `operators` section, ensure that the operator has `installPlanApproval: Manual` set. If an entry does not have the `installPlanApproval` parameter, add it.
       For example:

       ```
       - channel: v1.0
           installPlanApproval: Manual
         name: ibm-cpd-ccs-operator
         namespace: cpd-operators
         packageName: ibm-cpd-ccs
         scope: public
         sourceNamespace: openshift-marketplace
       ```

     - Save your changes and exit. For example, if you are using `vi`, hit esc and enter :wq.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

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 <u>Connecting to your identity provider</u>.

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 `Ibmcpd` 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 ibmcpd-cr
      ```

   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 `Ibmcpd` custom resource. For example, if you are using `vi`, hit `esc` and enter `: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=-
```

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Making monitoring data highly available

IBM® Cloud Pak for Data monitoring data is stored in an embedded InfluxDB database. By default, only one replica of the database deployed. However, if you want to ensure that the monitoring data is highly available, you can increase the number of replicas.

Permissions you need to complete this task
   You must be an administrator on the Red Hat® OpenShift® Container Platform project where Cloud Pak for Data is installed.
When you need to complete this task
   It is recommended that you make the monitoring data highly available if you expect to run large workloads or if you plan to install multiple services. In large or complex deployments, having timely access to monitoring data is critical. [Monitoring your deployment](#) can help you prevent unexpected outages.
   You can complete this task at any time. However, it is recommended that you complete this task before you make Cloud Pak for Data available to end users.

About this task

Having multiple replicas of the monitoring data improves reliability and availability. In some situations, it can also reduce latency.

Note: [Scaling the Cloud Pak for Data control plane](#) does not have any impact on the InfluxDB database. Similarly, you can make the monitoring data highly available without scaling the Cloud Pak for Data control plane.

Procedure

To increase the number of InfluxDB replicas:

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. Edit the `ZenService` custom resource. For example:

```
oc edit ZenService lite-cr --namespace cpd-instance
```

3. Add the `zen_monitoring_scale_config` property to the `spec` section of the custom resource.

```
...
spec:
  zen_monitoring_scale_config: medium
```

4. Save your changes to the `ZenService` custom resource. For example, if you are using `vi`, hit `esc` and enter `:wq`.
5. Check the status of the `ZenService` custom resource:

```
oc get ZenService lite-cr \
--namespace cpd-instance \
-o jsonpath="{.status}"
```

Wait for the command to return `Completed`.

When you set `zen_monitoring_scale_config: medium`, a the `Zen` operator starts a second replica of the InfluxDB database and an instance of the InfluxDB relay service, which ensures that the replicas contain the same information.

Release This information applies to Refresh 6 of Version 4.0.

Changing the route to the platform

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**. IBM® Cloud Pak for Data supports two different methods for changing the route to the platform.

About this task

By default, the route to the platform has the following structure:

https://cpd-*namespace*.apps.*OCP-default-domain*

Cloud Pak for Data supports the following methods for changing the route to the platform:

Overriding the hostname in the default route
> You can edit the **ZenService** custom resource to override the default hostname.

Creating a custom route
> You can use the **oc create route** command to create an additional route to the platform.

You can use these methods independently or in tandem depending on your needs.

Complete the appropriate tasks for your environment:

- **Overriding the hostname in the default route**
  After you install IBM Cloud Pak for Data, you can optionally override the hostname that is used in the default route.
- **Creating a custom route to the platform**
  After you install IBM Cloud Pak for Data, you can optionally create an alternative route by which users access the Cloud Pak for Data web client.

Release This information applies to Refresh 6 of Version 4.0.

Overriding the hostname in the default route

After you install IBM® Cloud Pak for Data, you can optionally override the hostname that is used in the default route.

Permissions you need to complete this task
> You must be either:

- A cluster administrator
- An administrator of the Red Hat® OpenShift® project (namespace) where Cloud Pak for Data is installed.

When you need to complete this task
> This task is optional. However, if you plan to override the hostname in the default route, you must complete this task before you give users access to the platform. If you change the hostname after you give users the link, the users won't be able to access the platform until you give them the updated link.

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*

You can edit the **ZenService** custom resource to override the hostname that is used in the default route.

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. Run the following command to add the **zenCustomRoute.route_host** parameter to the **ZenService** custom resource and set the hostname to use for the platform.
   The command assumes that Cloud Pak for Data is installed in the **cpd-instance** project. If Cloud Pak for Data is installed in a different project, update the **--namespace** parameter with the appropriate project for your environment.

   Ensure that you replace **mycpdhost.myapps.mydomain.org** with the fully qualified hostname that you want to use.

   ```
   oc patch ZenService lite-cr \
   --namespace cpd-instance \
   --type=merge \
   --patch '{"spec": {"zenCustomRoute": {"route_host":" mycpdhost.myapps.mydomain.org "}}}'
   ```

3. Check the status of the **ZenService** custom resource.

The command assumes that Cloud Pak for Data is installed in the `cpd-instance` project. If Cloud Pak for Data is installed in a different project, update the `--namespace` parameter with the appropriate project for your environment.

```
oc get ZenService lite-cr \
--namespace cpd-instance \
-o jsonpath="{.status}"
```

Wait for the command to return `Completed`.

Results

When you add the hostname to the `ZenService` custom resource, the `Zen` operator updates:

- The route (ingress hostname) of the Cloud Pak for Data control plane.
- Any entries related to the client hostname in the IAM Service registration, if applicable.
- The `URL_PREFIX` in the Cloud Pak for Data `product-configmap`.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Creating a custom route to the platform

After you install IBM® Cloud Pak for Data, you can optionally create an alternative route by which users access the Cloud Pak for Data web client.

Permissions you need to complete this task
    You must be either:

- A cluster administrator
- An administrator of the Red Hat® OpenShift® project (namespace) where Cloud Pak for Data is installed.

When you need to complete this task
    This task is optional. However, if you plan to create a custom route to the platform, it is recommended that you complete this task before you give users access to the platform. This ensures that users have the link that you want them to use to access the platform.

Before you begin

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`

You can create a custom route that users can use to access the Cloud Pak for Data platform.

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. 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.
   - 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-nginx-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\"}}}"`

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Configuring an external route to the Flight Server

The Flight Server is a data connection service that enables assets, such as notebooks, to interact with various data sources without calling the REST APIs for the data sources. By default, the Flight Server is only available to the IBM® Cloud Pak for Data instance where the Flight Server is running. However, a Red Hat® OpenShift® Container Platform project administrator can create an external route to the Flight Server to enable other applications to interact with it.

Permissions you need for this task
You must be either:

- A cluster administrator
- An administrator of the Red Hat OpenShift project (namespace) where Cloud Pak for Data is installed.

About this task

Common core services The Flight Server is available only if the common core services are installed.

Remember: The common core services are not a separately installable component. They are automatically installed if you install a service that has a dependency on them. For information about which services install the common core services, see Service software requirements.

Procedure

To create an external route to the Flight Server:

1. Log in to your Red Hat OpenShift Container Platform 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 and the Flight Server are installed.

    `oc project project-name`

3. Run the following command to create the route:

    `oc create route passthrough route-name --service=wdp-connect-flight`

    Replace *route_name* with the label you want to use for the route

    If you want to use a specific hostname, you can include the `--hostname hostname` argument in the command. For example:

    `oc create route passthrough route-name --hostname hostname --service=wdp-connect-flight`

4. Run the following command to get the URL:

    `oc get route route-name`

    Use this URL to access the Flight Server from external applications.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see Documentation for previous 4.0.x refreshes.

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.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see Documentation for previous 4.0.x refreshes.

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. You can optionally create a custom route. For details, see [Creating a custom route to the platform](#).

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 to the platform](#)
- [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 to the platform](#)**
  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.
- **[Using a CA certificate to connect to internal servers from the platform](#)**
  If you want to enable the IBM Cloud Pak for Data platform to use your company's CA certificate to validate certificates from your internal servers, you must create a secret that contains the CA certificate. Additionally, if your internal servers use an SSL certificate that is signed using your company's CA certificate, you must create this secret to enable the platform to connect to the servers.
- **[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.
- **[Enabling users to create NFS storage volumes](#)**
  To enable users to connect to remote NFS storage volumes from Cloud Pak for Data, a cluster administrator must create a cluster role and cluster role binding to grant the `meta-api` the appropriate permissions to provision persistent volumes.
- **[Enabling users to connect to external SMB storage volumes](#)**
  To enable users to connect to remote SMB storage volumes from IBM Cloud Pak for Data a cluster administrator must install the SMB CSI Driver for Kubernetes to enable the cluster to communicate with your remote SMB volumes. Then the administrator must create a cluster role and cluster role binding to grant the `meta-api` the appropriate permissions to provision persistent volumes.
- **[Configuring vault usage](#)**
  Cloud Pak for Data includes an internal vault that you can use to store secrets. You can also connect to external vaults where you already store sensitive information as secrets. After you install Cloud Pak for Data, use the following tasks to align the vault configuration with your security policies and practices.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Using a custom TLS certificate for HTTPS connections to the platform

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.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Using a CA certificate to connect to internal servers from the platform

If you want to enable the IBM® Cloud Pak for Data platform to use your company's CA certificate to validate certificates from your internal servers, you must create a secret that contains the CA certificate. Additionally, if your internal servers use an SSL certificate that is signed using your company's CA certificate, you must create this secret to enable the platform to connect to the servers.

After you create the secret, the CA certificate is used in the following connections:

- Platform connections
- Connections created in an analytics project
- Connections created in a catalog

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 file in PEM format.

Procedure

To add a CA certificate to the platform:

1. Place the PEM file on your local file system.
2. Change to the directory where the files are located.
3. Log in to Red Hat OpenShift Container Platform as a user with sufficient permissions to complete the task.

   `oc login OpenShift_URL:port`

4. Set the context to the project where Cloud Pak for Data is deployed, for example `cpd-instance`

   `oc project cpd-instance`

5. Create a secret named `connection-ca-certs` to store the CA certificate.
   Important: The secret must be named `connection-ca-certs`. If you use a different name, the platform will not be able to locate the secret.

   `oc create secret generic connection-ca-certs --from-file=my-cert.pem=./my-cert.crt`

   Wait for the command to return a message that the secret was created:

   `secret/connection-ca-certs created`

6. If the following pods are running on your cluster, restart them so that they can use the `connection-ca-certs` secret:

   `wdp-connect` pods
   > Run the following command to restart any `wdp-connect` pods in the project:
   >
   > `for i in `oc get pods | grep wdp-connect- | cut -f1 -d\ `; do oc delete po ${i}; done`
   >
   > If there are any `wdp-connect` pods in the project, the command should return output with the following format:
   >
   > ```
   > pod "wdp-connect-connector-6f968c97ff-4lcxc" deleted
   > pod "wdp-connect-connection-77879f485d-f7cfl" deleted
   > pod "wdp-connect-flight-6b75df645b-ts4rg" deleted
   > ```

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

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.	

Parameter	Description	Value
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
```

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

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`, hit `esc` and enter:

```
:wq
```

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

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`, hit `esc` and 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
   ```

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

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**, hit `esc` and enter:

   `:wq`

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

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
headerText	Specify the text to display as the header in the dialog. For example: `"headerText": "Terms of use",` Use standard JSON string format.

Property	Description
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.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see <u>Documentation for previous 4.0.x refreshes</u>.

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.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see <u>Documentation for previous 4.0.x refreshes</u>.

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`, hit esc and enter:

   `:wq`

   The changes are automatically applied to the platform.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Enabling users to create NFS storage volumes

To enable users to connect to remote NFS storage volumes from Cloud Pak for Data, a cluster administrator must create a cluster role and cluster role binding to grant the `meta-api` the appropriate permissions to provision persistent volumes.

About this task

Permissions you need for this task
   You must be a cluster administrator.
When you need to complete this task
   Complete this task only if you want to enable Cloud Pak for Data users to connect to storage volumes on a remote NFS server.
   Note: The External NFS option is always visible on the New storage volume page. However, if you do not complete this task, users cannot connect to NFS storage volumes from the New storage volume page or through the Volumes API.

Procedure

To enable users to connect to a remote NFS storage volume:

1. Log on to Red Hat® OpenShift® Container Platform as a user with sufficient permissions to complete this task:

   `oc login OpenShift_URL:port`

2. Run the following command to create the required cluster role and cluster role binding for the `meta-api`:

Tip: The same role binding is required to use remote SMB storage volumes. If you already completed [Enabling users to connect to external SMB storage volumes](#), this role binding already exists.

```
cat <<EOF | oc apply -f -
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  # "namespace" omitted. ClusterRoles are not scoped to a namespace.
  name: ibm-zen-volumes-cluster-role
rules:
- apiGroups: [""]
  #
  # at the HTTP level, the name of the resource for accessing Secret
  # objects is "secrets"
  resources: ["persistentvolumes"]
  verbs: ["create", "get", "list", "patch", "update", "watch", "delete", "use"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: ibm-zen-volumes-cluster-role-binding
subjects:
- kind: ServiceAccount
  name: ibm-zen-operator-serviceaccount
  namespace: ibm-common-services    # The namespace where the IBM Cloud Pak foundational services are installed
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: ibm-zen-volumes-cluster-role
EOF
```

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Enabling users to connect to external SMB storage volumes

To enable users to connect to remote SMB storage volumes from IBM® Cloud Pak for Data a cluster administrator must install the SMB CSI Driver for Kubernetes to enable the cluster to communicate with your remote SMB volumes. Then the administrator must create a cluster role and cluster role binding to grant the `meta-api` the appropriate permissions to provision persistent volumes.

About this task

Permissions you need for this task
    You must be a cluster administrator.
When you need to complete this task
    Complete this task only if you want to enable Cloud Pak for Data users to connect to storage volumes on an SMB file share server.
    Note: The External SMB option is always visible on the New storage volume page. However, if you do not complete this task, users cannot connect to SMB storage volumes from the New storage volume page or through the Volumes API.

Procedure

To enable users to connect to a remote SMB storage volume:

1. Log on to Red Hat® OpenShift® Container Platform as a user with sufficient permissions to complete this task:

   ```
   oc login OpenShift_URL:port
   ```

2. Install the [SMB CSI Driver for Kubernetes](#) (`csi-smb-driver`) on the cluster. For details, see [the readme](#) for the driver.
   The `csi-smb-driver` is provided by the Kubernetes CSI organization, which provides Kubernetes specific Constainer-Storage-Interface (CSI) components.

3. Run the following command to create the required cluster role and cluster role binding for the `meta-api`.
   Tip: The same role binding is required to use external NFS storage volumes. If you already completed [Enabling users to create NFS storage volumes](#), this role binding already exists.

   ```
   cat <<EOF | oc apply -f -
   ---
   apiVersion: rbac.authorization.k8s.io/v1
   kind: ClusterRole
   metadata:
     # "namespace" omitted. ClusterRoles are not scoped to a namespace.
     name: ibm-zen-volumes-cluster-role
   rules:
   - apiGroups: [""]
     #
     # at the HTTP level, the name of the resource for accessing Secret
     # objects is "secrets"
     resources: ["persistentvolumes"]
     verbs: ["create", "get", "list", "patch", "update", "watch", "delete", "use"]
   ---
   apiVersion: rbac.authorization.k8s.io/v1
   kind: ClusterRoleBinding
   metadata:
     name: ibm-zen-volumes-cluster-role-binding
   ```

```
    subjects:
    - kind: ServiceAccount
      name: ibm-zen-operator-serviceaccount
      namespace: ibm-common-services      # The namespace where the IBM Cloud Pak foundational services are installed
    roleRef:
      apiGroup: rbac.authorization.k8s.io
      kind: ClusterRole
      name: ibm-zen-volumes-cluster-role
    EOF
```

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Configuring vault usage

Cloud Pak for Data includes an internal vault that you can use to store secrets. You can also connect to external vaults where you already store sensitive information as secrets. After you install Cloud Pak for Data, use the following tasks to align the vault configuration with your security policies and practices.

- **Enabling vaults for the Cloud Pak for Data web client**
  A Red Hat® OpenShift® Container Platform project administrator can enable the vaults interface in the Cloud Pak for Data web client. The vaults interface provides options for adding secrets to the internal vault and for connecting to external vaults such as CyberArk or HashiCorp.
- **Disabling the internal vault for the Cloud Pak for Data web client**
  A Red Hat OpenShift Container Platform project administrator can optionally disable the Cloud Pak for Data internal vault for the Cloud Pak for Data web client.
- **Requiring users to use secrets for credentials when creating connections**
  When a user creates a connection, they can provide their credentials by entering them directly or by specifying a secret. A Red Hat OpenShift administrator can configure Cloud Pak for Data to enforce the exclusive use of secrets from an external vault (such as CyberArk or HashiCorp).
- **Disabling secret sharing for Cloud Pak for Data vaults**
  A Red Hat OpenShift Container Platform project administrator can optionally disable secret sharing for both internal and external Cloud Pak for Data vaults. Disabling secret sharing ensures that secret owners cannot share their vault secrets with other users.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Enabling vaults for the Cloud Pak for Data web client

A Red Hat® OpenShift® Container Platform project administrator can enable the vaults interface in the Cloud Pak for Data web client. The vaults interface provides options for adding secrets to the internal vault and for connecting to external vaults such as CyberArk or HashiCorp.

About this task

By default, the internal vault is available only through the [Credentials and Secrets API](#). If you want to enable users to connect to external vaults, or to use the web client to add secrets to the internal vault, you must update the Cloud Pak for Data deployment to enable the vaults interface.

Connections that use credentials from a vault might not work for all services. The following services support connections that use secrets from vaults:

- Data Virtualization
- Decision Optimization
- Execution Engine for Apache Hadoop
- OpenPages®
- Watson™ Knowledge Catalog
- Watson Studio Runtimes

Permissions that 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
   The ensure the use of secrets, you should complete this task immediately after Cloud Pak for Data is installed.

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 add `zenVaultEnabled:true` to the `Ibmcpd` custom resource:

   ```
   oc patch Ibmcpd ibmcpd-cr \
   --namespace <cpd-instance>
   --type=merge \
   --patch '{"spec": {"zenVaultEnabled":"true"}}'
   ```

4. Confirm that the `zenVaultEnabled:true` is added:

```
oc get Ibmcpd ibmcpd-cr -o jsonpath="{.spec.zenVaultEnabled}{'\n'}"
```

The command returns **true** when the entry is added.

5. Confirm that the IBM Cloud Pak® for Data platform operator reconciliation loop completes:

```
oc get Ibmcpd ibmcpd-cr -o jsonpath="{.status.controlPlaneStatus}{'\n'}"
```

Wait for the command to return **Completed**.

6. Restart the watcher pod.

```
oc delete pods -l component=zen-watcher
```

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Disabling the internal vault for the Cloud Pak for Data web client

A Red Hat® OpenShift® Container Platform project administrator can optionally disable the Cloud Pak for Data internal vault for the Cloud Pak for Data web client.

About this task

If you plan to use vaults to store sensitive data, it is strongly recommended that you use an enterprise-grade vault solution, such as CyberArk or HashiCorp.

After you [enable the web client to connect to external vaults](#), you can optionally disable the internal vault to ensure that users use secrets from your existing vault.

You can disable the internal vault, for example, if you want to restrict the use of vault to an organization's approved vault vendors only. Disabling the internal vault occurs after product installation.

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
        You can complete this task anytime after Cloud Pak for Data is installed.

Important: When you disable the internal vault, all of the secrets that are stored in the vault are also deleted.

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 product-configmap file:

```
oc edit configmap product-configmap
```

4. Change the value of VAULT_DISABLE_INTERNAL_VAULT parameter to `true`.

```
VAULT_DISABLE_INTERNAL_VAULT:true
```

5. Save your changes to the product-configmap file. For example, if you are using **vi**, hit `esc` and enter `:wq`.
6. Restart the watcher pod.

```
oc delete pods -l component=zen-watcher
```

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Requiring users to use secrets for credentials when creating connections

When a user creates a connection, they can provide their credentials by entering them directly or by specifying a secret. A Red Hat® OpenShift® administrator can configure Cloud Pak for Data to enforce the exclusive use of secrets from an external vault (such as CyberArk or HashiCorp).

Permissions that you need for this task
        You must have the following permissions to restrict user access to connections with external vault secrets.

    • Administrator of the Red Hat OpenShift project (namespace) where Cloud Pak for Data is installed.

When you need to complete this task
        You can complete this task anytime after Cloud Pak for Data is installed.

About this task

A Red Hat OpenShift project (namespace) administrator can edit the config-wdp-connect-connection configuration map to set `allow-only-vaulted-credentials` to `true`.

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 config-wdp-connect-connection file.

   ```
   oc edit configmap config-wdp-connect-connection
   ```

4. Change the allow-only-vaulted-credentials parameter value to `true` (the default value is `false`).

   ```
   allow-only-vaulted-credentials:true
   ```

5. Save your changes and exit. For example, if you are using `vi`, hit esc and enter `:wq`.
6. Delete the relevant pods:

   ```
   oc delete pods -l app=wdp-connect-connection
   oc delete pods -l app=wdp-connect-connector
   ```

7. Verify that the pods return and are running:

   ```
   oc get pods -l app=wdp-connect-connection
   oc get pods -l app=wdp-connect-connector
   ```

Results

Cloud Pak for Data is configured for the exclusive use of external vault secrets for connections.
Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Disabling secret sharing for Cloud Pak for Data vaults

A Red Hat® OpenShift® Container Platform project administrator can optionally disable secret sharing for both internal and external Cloud Pak for Data vaults. Disabling secret sharing ensures that secret owners cannot share their vault secrets with other users.

About this task

After you [enable the web client to connect to a vault](#), you can optionally disable secret sharing for the vaults.

You can disable secret sharing for the internal or external vault, for example, if you want to ensure that secret owners cannot share their vault secrets with other users. Disabling secret sharing for the internal or external vault occurs after product installation and after the web client is configured to connect to a vault.

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
    You can complete this task anytime after Cloud Pak for Data is installed, and after [enabling the web client to connect to a vault](#).

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 update the Cloud Pak for Data product-configmap file:

   ```
   oc patch cm product-configmap \
   --type=merge \
   --patch '{"data": {"VAULT_DISABLE_SHARING_SECRET": "true"}}'
   ```

4. Check the updated VAULT_DISABLE_SHARING_SECRET value.

   ```
   oc get cm product-configmap -o jsonpath="{.data.VAULT_DISABLE_SHARING_SECRET}{'\n'}"
   ```

   When the `VAULT_DISABLE_SHARING_SECRET` equals true, secret owners with share secret permission are not allowed to share their secrets with other users.
5. Save your changes to the product-configmap file. For example, if you are using `vi`, hit esc and enter `:wq`.

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see [Documentation for previous 4.0.x refreshes](#).

Uninstalling the common core services

A project administrator can uninstall the common core services from IBM® Cloud Pak for Data.

When you need to complete this task
You should uninstall the common core services if you plan to uninstall Cloud Pak for Data.
Attention: Do not uninstall the common core services if:

- You are using other services that also have a dependency on the common core services. For details, see Service software requirements.
- You plan to re-install a service that has a dependency on the common core services.

If you uninstall the common core services, the data that is associated with the common core services is deleted and cannot be recovered.

Contact IBM Software Support if you are unsure whether you should complete this task.

Procedure

The common core services are automatically installed by services that require them, such as Watson™ Studio and Watson Knowledge Catalog. However, when you uninstall these services, the common core services are *not* uninstalled. The common core services remain on the cluster to ensure that they are available to any services that use them.

Complete the following tasks to uninstall the common core services:

1. Uninstalling dependent services
2. Uninstalling common core services
3. Uninstalling the operator

Uninstalling dependent services

When you need to complete this task
Complete this task when you want to remove the common core services.

The following services install the common core services. Ensure that you uninstall these services before you uninstall the common core services.

- Cognos® Analytics
- Cognos Dashboards
- Watson Knowledge Catalog
- Watson Machine Learning
- Watson Studio

To identify all services that depend on the common core services, run the following command:

```
oc get operandrequest -n project-name | grep ccs
```

Uninstalling the common core services

To uninstall the common core services:

1. Delete the common core services custom resource in the OpenShift® project, where the common core services are installed. Use the following command:

   ```
   oc delete ccs -n project-name ccs-cr
   ```

2. Confirm that the common core services were deleted by using the following command:

   ```
   oc get ccs -n project-name
   ```

   That command should return the following results:

   ```
   No resources found in project-name namespace.
   ```

3. Locate the common core services operandrequest by using the following command:

   ```
   oc get operandrequest -n project-name | grep ccs
   ```

4. Delete the common core services operandrequest by using the following command:

   ```
   oc delete operandrequest -n project-name <ccs oprandrequest name>
   ```

Uninstalling the operator

When you need to complete this task
Complete this task only if you want to completely remove the common core services.
Complete this task *after* you uninstall the common core services.

Before you begin

Confirm that there are no common core services custom resources and common core services operandrequests all namespaces in the cluster.

For each command, ensure that you specify the appropriate Red Hat® OpenShift project, either `ibm-common-services` or `cpd-operators` for the `--namespace` argument.

For the variable, use `operator-project`.

1. Delete the common core services subscription using the following command:

   ```
   oc delete sub ibm-cpd-ccs-operator -n <ccs operator namespace>
   ```

2. Delete the common core services catalog source.

   ```
   oc delete catsrc ibm-cpd-ccs-operator-catalog -n openshift-marketplace
   ```

Related reference

- [Pre-installation tasks](#)
- [Installing Cloud Pak for Data](#)
- [Post-installation tasks](#)
- [Uninstalling Cloud Pak for Data](#)

Uninstalling the scheduling service

A cluster administrator can uninstall the scheduling service.

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

Procedure

Complete the following tasks to uninstall scheduling service:

1. [Uninstalling dependent services](#)
2. [Uninstalling the service](#)
3. [Restarting pending pods](#)
4. [Uninstalling the operator](#)
   Note: Complete this step only if you want to completely remove the scheduling service software from your cluster.

Uninstalling dependent services

The following service requires the scheduling service:

- [Watson™ Machine Learning Accelerator](#)

Ensure that you uninstall this service before you uninstall the scheduling service.

Uninstalling the service

When you need to complete this task
Complete this task when you want to remove the scheduling service and revert your cluster to the default scheduler.

Note: If you uninstall the scheduling service, the quota enforcement feature will not be available in Cloud Pak for Data.
When you delete the `Scheduling` custom resource, the `Scheduling` operand is uninstalled. (The operand is the running instance of the scheduling service.)

Uninstalling the scheduling service will remove all of the data that is associated with this instance of the scheduling service.

To uninstall the 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. Delete the scheduling service custom resource.
   The following command assumes that the scheduling service is installed in the default `ibm-common-services` project:

   ```
   oc delete Scheduling ibm-cpd-scheduler \
    --namespace ibm-common-services
   ```

3. Run the following command to verify that the pods that are associated with the scheduling service operand have been stopped.

   ```
   oc get po -n ibm-common-services | grep ^ibm-cpd-sched
   ```

   The command should return only the operator pod (`ibm-cpd-scheduling-operator-*`). If the command returns multiple pods, wait several minutes before running the command again.

Restarting pending pods

After you uninstall the scheduling service, you must delete any pending pods that are waiting to be scheduled by the scheduling service. When the pods restart, they will be scheduled by the default scheduler.

To restart the pending pods:

1. Run the following command to see which projects have pods that are under the control of the scheduling service:

```
oc get po -A --no-headers | while read namespace name ready status misc; do
  schedulerName=`oc get po -n$namespace $name -o=jsonpath="{.spec.schedulerName}"`
  if [ "$schedulerName" == "ibm-cpd-scheduler" -a "$status" == "Pending" ]; then
    echo "$namespace"
  fi
done | sort -u
```

The command returns a list of projects.

2. For each project returned by the preceding command, complete the following steps:
   a. Change to the project:

   ```
   oc project project-name
   ```

   b. Run the command to list the pending pods in the project that are still under the control of the scheduling service:

   ```
   oc get po --no-headers | while read name ready status misc; do
     schedulerName=`oc get po $name -o=jsonpath='{.spec.schedulerName}'`
     if [ "$schedulerName" == "ibm-cpd-scheduler" -a $status == "Pending" ]; then
       echo "Pod $name is pending, and schedulerName is set to ibm-cpd-scheduler"
     fi
   done
   ```

   The command returns a list of pods. If the command does not return any pods, there are no pending pods that need to be restarted in this project.

   c. Run the following command to delete the pods:

   ```
   oc get po --no-headers | while read name ready status misc; do
     schedulerName=`oc get po $name -o=jsonpath='{.spec.schedulerName}'`
     if [ "$schedulerName" == "ibm-cpd-scheduler" -a $status == "Pending" ]; then
       oc delete po $name
     fi
   done
   ```

   d. Repeat the preceding steps for each project that was returned in the output for step 1.

Uninstalling the operator

When you need to complete this task
    Complete this task only if you want to completely remove the scheduling service from your cluster.
    Complete this task *after* you uninstall the service.

To uninstall the operator:

1. Delete the scheduling service operator subscription.
   The following command assumes that the scheduling service is installed in the default **ibm-common-services** project:

   ```
   oc delete sub ibm-cpd-scheduling-catalog-subscription \
   --namespace ibm-common-services
   ```

2. Delete the scheduling service cluster service version (CSV).
   The following command assumes that the scheduling service is installed in the default **ibm-common-services** project:

   ```
   oc delete -n ibm-common-services csv \
   -l operators.coreos.com/ibm-cpd-scheduling-operator.ibm-common-services=""
   ```

Release This information applies to Refresh 6 of Version 4.0.

For information about previous releases, see <u>Documentation for previous 4.0.x refreshes</u>.

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. <u>Uninstalling dependent services</u>
2. <u>Uninstalling Cloud Pak for Data</u>
3. <u>Uninstalling the operators</u>
   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 <u>Services</u>.

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. 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 the Cloud Pak for Data operators are installed:

   `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 <u>Uninstalling foundational services</u>.
- For air-gapped clusters, see <u>Uninstalling IBM Cloud Pak foundational services in an airgap environment</u>.

Related reference

- <u>Pre-installation tasks</u>
- <u>Installing Cloud Pak for Data</u>
- <u>Post-installation tasks</u>

- [Uninstalling the common core services](#)