

Part V: Installing



Tables of Contents

Part V: Installing	1
Setting up a client workstation	6
Installing the Cloud Pak for Data CLI	8
Installing the OpenShift CLI	9
Collecting required information	10
Obtaining your IBM entitlement API key	10
Determining which components to install	11
Setting up installation environment variables	17
Preparing your cluster	24
Installing Red Hat OpenShift	25
Setting up persistent storage	27
Setting up IBM Spectrum storage	29
Creating Portworx storage classes	31
Setting up NFS storage	39
Configuring your NFS server	40
Setting up dynamic provisioning	42
Setting up Amazon Elastic File System	45
Setting up IBM Cloud File Storage	47
Setting up projects (namespaces)	49
Creating custom SCCs for services	52
Creating the Watson Knowledge Catalog SCC	54
Creating the Db2 SCC	55
Creating the Db2 Warehouse SCC	58
Creating the SCC for embedded Db2 databases	61
Changing required node settings	63
Using the OpenShift Node Tuning Operator to set kernel parameters	69
Updating the global image pull secret	73
Mirroring images to a private container registry	75
Mirroring images directly to the private container registry	77
Mirroring images using an intermediary container registry	78
Configuring an image content source policy	81
Installing the IBM Cloud Pak for Data platform and services	83
Express installations	84
Creating OLM objects	85
Installing components	86
Specialized installations	89
Creating OLM objects	90
Installing components	92
Uninstalling the platform and services	96
Uninstalling the components	96
Uninstalling the OLM objects	97

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

Before you install Cloud Pak for Data, review the information in the [Planning](#) section.

Specifically, ensure that you review the [System requirements](#). You must install the software on a cluster 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.

1. Setting up a client workstation

To install IBM Cloud Pak for Data, you must have a client workstation that can connect to the Red Hat OpenShift Container Platform cluster.

Tip: You can set up multiple client workstations if you want to enable multiple people to work on the installation. The client workstation must be a Windows, Mac OS, or Linux® machine with the following software installed:

- Cloud Pak for Data command-line interface (**cpd-cli**) Version 11.0.0 or later.
- OpenShift command-line interface (**oc**) at a version that is compatible with your cluster.

Options	What to do
You already have a client workstation set up	1. Go to 2. Collecting required information .
You don't have a client workstation set up	1. Review the guidance in Setting up a client workstation . 2. Complete the following tasks to install the required software on the client workstation: a. Installing the IBM Cloud Pak for Data command-line interface b. Installing the OpenShift command-line interface 3. Go to 2. Collecting required information .

2. Collecting required information

To successfully install IBM Cloud Pak for Data, you must have specific information about your environment.

a. Obtaining your IBM entitlement API key

All IBM Cloud Pak for Data images are accessible from the IBM Entitled Registry. The IBM entitlement API key enables you to pull software images from the IBM Entitled Registry, either for installation or for mirroring to a private container registry.

Options	What to do
You already have your API key	1. Go to b. Determining the list of components that you plan to install .
You don't have your API key	1. Complete Obtaining your IBM entitlement API key . 2. Go to b. Determining the list of components that you plan to install .

b. Determining the list of components that you plan to install

IBM Cloud Pak for Data is comprised of numerous components so that you can install the specific services that support your needs. Before you install Cloud Pak for Data, determine which components you need to install.

What to do

What to do
<ol style="list-style-type: none"> Review Determining which components to install to ensure that you: <ul style="list-style-type: none"> Install all the required components Know which tasks you must complete to prepare your cluster (some services have additional pre-installation configuration) Go to c. Collecting information about your cluster that can be used to set up environment variables.

c. Collecting information about your cluster that can be used to set up environment variables

The commands for installing and upgrading IBM Cloud Pak for Data use variables with the format `${VARIABLE_NAME}`. You can create a script to automatically export the appropriate values as environment variables before you run the installation commands. After you source the script, you will be able to copy most install and upgrade commands from the documentation and run them without making any changes.

What to do
<ol style="list-style-type: none"> Complete Setting up installation environment variables. Go to 3. Preparing your cluster.

3. Preparing your cluster

Before you install Cloud Pak for Data, you must prepare your cluster.

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

- [Supported versions of Red Hat OpenShift Container Platform](#)

Cloud Pak for Data can be installed on the following versions of Red Hat OpenShift Container Platform:

- Version 4.6.29 or later fixes
- Version 4.8.0 or later fixes
- Version 4.10.0 or later fixes

Options	What to do
You are running a supported version of OpenShift	<ol style="list-style-type: none"> Go to b. Do you have supported persistent storage on your cluster?
You have an older version of OpenShift	<ol style="list-style-type: none"> Upgrade your cluster. <ul style="list-style-type: none"> If you are using self-managed OpenShift, see the Red Hat OpenShift Container Platform documentation. If you are using managed OpenShift, refer to the documentation for your OpenShift provider. Go to b. Do you have supported persistent storage on your cluster?
You don't have an OpenShift cluster	<ol style="list-style-type: none"> Go to Installing Red Hat OpenShift Container Platform.

b. Do you have supported persistent storage on your cluster?

- [Supported storage for the Cloud Pak for Data platform](#)

The Cloud Pak for Data platform supports the following storage:

Storage option	Version	Notes
OpenShift Data Foundation (formerly called OpenShift Container Storage)	Version: 4.6 or later fixes	Available in the IBM Storage Suite for IBM Cloud® Paks
OpenShift Data Foundation as a Service	Not applicable	Contact IBM Support for assistance.
IBM Spectrum® Fusion	Version 2.2.0 or later fixes	Available in either: <ul style="list-style-type: none"> IBM Spectrum Fusion IBM Storage Suite for IBM Cloud Paks

Storage option	Version	Notes
IBM Spectrum Scale Container Native (with IBM Spectrum Scale Container Storage Interface)	Version 5.1.3.x or later fixes CSI Version 2.5.x or later fixes	Available in either: <ul style="list-style-type: none"> IBM Spectrum Fusion IBM Storage Suite for IBM Cloud Paks
Portworx	Version 2.7.0 or later fixes	
NFS	Version 3 or 4 The latest version is recommend.	
Amazon Elastic Block Store (EBS)	Not applicable	Your environment must also include EFS storage.
Amazon Elastic File System (EFS)	Not applicable	It is recommended that you use both EBS and EFS storage.
IBM Cloud Block Storage	Not applicable	Your environment must also include IBM Cloud File Storage.
IBM Cloud File Storage	Not applicable	It is recommended that you use both IBM Cloud Block Storage and IBM Cloud File Storage storage.

Options	What to do
You have supported storage	<ol style="list-style-type: none"> Ensure that you have storage that works with the services that you plan to install. Review Setting up persistent storage to determine whether you need to complete any additional tasks to configure the storage for Cloud Pak for Data. Go to c. Do you have the required OpenShift projects on your cluster?
You don't have supported storage	<ol style="list-style-type: none"> Decide which storage you want to use. Ensure that you choose storage that works with the services that you plan to install. Follow the guidance in Setting up persistent storage for installing and configuring the storage. Go to c. Do you have the required OpenShift projects on your cluster?

c. 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 a project where you will install an instance of Cloud Pak for Data. You will need additional projects if you want to:

- Separate the Cloud Pak for Data operators from the IBM Cloud Pak® foundational services operators
- Install multiple instances of Cloud Pak for Data
- Deploy service instances or workloads in tethered projects

For details, see [Supported project \(namespace\) configurations](#).

Options	What to do
You know which projects you plan to use when you install the software	<ol style="list-style-type: none"> Review the guidance in Setting up projects (namespaces) on Red Hat OpenShift Container Platform to: <ul style="list-style-type: none"> • Ensure that you have the necessary projects on your cluster • Determine whether you need to label any projects • Set up tethered projects, if applicable Go to d. Do you plan to install any services that require custom SCCs?
You don't know which projects you plan to use when you install the software	<ol style="list-style-type: none"> 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 and then create the appropriate projects. Go to d. Do you plan to install any services that require custom SCCs?

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

- >
 Services that require custom SCCs

If you plan to install any of the following Cloud Pak for Data services, you must create the appropriate custom SCCs:

Service	Required SCCs
Data Virtualization	Data Virtualization embeds an instance of Db2®, which requires a custom SCC. This SCC is used only by the instance of Data Virtualization that embeds the Db2 database. For details, see Creating the custom security context constraint for embedded Db2 databases .
Db2	Db2 requires a custom SCC. By default, the SCC is created automatically; however, you can choose to create the SCC manually. For details, see Creating the custom security context constraint for Db2 .
Db2 Big SQL	Db2 Big SQL embeds an instance of Db2, which requires a custom SCC. This SCC is used only by the instance of Db2 Big SQL that embeds the Db2 database. For details, see Creating the custom security context constraint for embedded Db2 databases .
Db2 Warehouse	Db2 Warehouse requires a custom SCC. By default, the SCC is created automatically; however, you can choose to create the SCC manually. For details, see Creating the custom security context constraint for Db2 Warehouse .
OpenPages®	The OpenPages service can optionally embed an instance of Db2. If you chose to use an embedded instance of Db2, OpenPages requires a custom SCC for the Db2 database. This SCC is used only by the instance of OpenPages that embeds the Db2 database. For details, see Creating the custom security context constraint for embedded Db2 databases . If you choose to use an external database, the custom SCC is not required.
Watson™ Knowledge Catalog	Watson Knowledge Catalog requires two custom SCCs: <ul style="list-style-type: none"> ◦ An SCC for Watson Knowledge Catalog. You must create this SCC manually. For details, see Creating the custom security context constraint for Watson Knowledge Catalog. ◦ An SCC for the instance of Db2 that is embedded in Watson Knowledge Catalog. This SCC is used only by the instance of Watson Knowledge Catalog that embeds the Db2 database. For details, see Creating the custom security context constraint for embedded Db2 databases. <p>If you install Data Privacy, the service uses the Watson Knowledge Catalog SCC.</p>

Options	What to do
You plan to install one or more of these services	<ol style="list-style-type: none"> 1. Create the appropriate SCCs for your environment. For details, see Creating custom security context constraints for services. 2. Go to e. Do you plan to install any services that require specific node settings?
You don't plan to install any of these services	<ol style="list-style-type: none"> 1. Go to e. Do you plan to install any services that require specific node settings?

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

- >
 Services that require node settings

Node setting	Services that require changes to the setting
Load balancer timeout settings	<ul style="list-style-type: none"> ◦ Db2 Data Gate ◦ OpenPages ◦ Watson Discovery ◦ Watson Knowledge Catalog ◦ Also recommended if you are working with large data sets or you have slower network speeds.

Node setting	Services that require changes to the setting
CRI-O container settings	<ul style="list-style-type: none"> ○ Cognos® Analytics ○ Data Virtualization ○ Db2 ○ Db2 Big SQL ○ Db2 Warehouse ○ Watson Discovery ○ Watson Knowledge Catalog ○ Watson Studio ○ Watson Machine Learning Accelerator
Kernel parameter settings	<ul style="list-style-type: none"> ○ Data Virtualization ○ Db2 ○ Db2 Big SQL ○ Db2 Warehouse ○ Watson Knowledge Catalog ○ Watson Studio
GPU settings	<ul style="list-style-type: none"> ○ Jupyter Notebooks with Python 3.9 for GPU

Options	What to do
You plan to install one or more of these services	<ol style="list-style-type: none"> 1. Update the node settings. For details, see Changing required node settings. 2. Go to f. How are you going to access the software images?
You don't plan to install any of these services	<ol style="list-style-type: none"> 1. Go to f. How are you going to access the software images?

f. How are you going to access the 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.

- >
Where should you pull images from?

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.

Options	What to do
You are pulling images from the IBM Entitled Registry	<ol style="list-style-type: none"> 1. Complete Updating the global image pull secret. 2. Go to 4. Installing the Cloud Pak for Data platform and services.
You are pulling images from a private container registry	<ol style="list-style-type: none"> 1. Complete Updating the global image pull secret. 2. Complete Mirroring images to a private container registry. 3. Go to 4. Installing the Cloud Pak for Data platform and services.

4. Installing the Cloud Pak for Data platform and services

After you prepare your cluster, you can install the Cloud Pak for Data platform and services.

What to do
1. Complete the appropriate tasks for your environment in Installing the IBM Cloud Pak for Data platform and services .
2. Go to 5. Completing post-installation tasks .

5. 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.

What to do
Complete the appropriate tasks for your environment in Post-installation setup (Day 1 operations) .

6. Installing services

Options	What to do
You installed the services when you installed the platform	Best practice Review Getting started with Cloud Pak for Data .
You didn't install the services when you installed the platform	Install the services that you want to use. See the instructions for installing each service individually in Services .

- [Setting up a client workstation](#)
To install IBM Cloud Pak for Data, you must have a client workstation that can connect to the Red Hat OpenShift Container Platform cluster.
- [Collecting required information](#)
To successfully install IBM Cloud Pak for Data, you must have specific information about your environment. Complete the following tasks to ensure that you have the information that you need.
- [Preparing your cluster](#)
Before you install IBM Cloud Pak for Data, complete the following tasks to prepare your cluster.
- [Installing the IBM Cloud Pak for Data platform and services](#)
To install the IBM Cloud Pak for Data platform and services, you must create the required Operator Lifecycle Manager (OLM) objects and custom resources (CRs) for the software that you want to use.
- [Uninstalling the platform and services](#)
If you need to uninstall the IBM Cloud Pak for Data platform and services, you can remove the custom resources and the Operator Lifecycle Manager (OLM) objects that are associated with the components.

Setting up a client workstation

To install IBM® Cloud Pak for Data, you must have a client workstation that can connect to the Red Hat® OpenShift® Container Platform cluster.

Before you begin

Before you install any software on the client workstation, ensure that the workstation meets the requirements in:

- [Internet connection requirements](#)
- [Operating system requirements](#)
- [Container runtime requirements](#)

After you confirm that the workstation meets these requirements, you can install the [required command-line interfaces](#).

Internet connection requirements

Some tasks require a connection to the internet. If your cluster is in a restricted network, you can either:

- Move the workstation behind the firewall after you complete the tasks that require an internet connection
- Prepare a client workstation that can connect to the internet and a client workstation that can connect to the cluster and transfer any files from the internet-connected workstation to the cluster-connected workstation.

When the workstation is connected to the internet, the workstation must be able to access the following sites:

- GitHub (<https://github.com/IBM>)
If your company does not permit access to GitHub, contact IBM Support.
- IBM Entitled Registry (<http://icr.io:43>)
To validate that you can connect, run the following command:

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

The command should return a message similar to:

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

Note: The IP address might be different.

Operating system requirements

The client workstation must be running a supported operating system:

- Linux®
 - Mac OS
 - Windows
- Important: To run on Windows, you must install Windows Subsystem for Linux.

Container runtime requirements

The workstation must have a supported container runtime.

Operating system	Docker	Podman	Notes
Linux	✓	✓	
Mac OS	✓		
Windows	✓	✓	Set up the container runtime inside Windows Subsystem for Linux.

Required command-line interfaces

To install or upgrade the Cloud Pak for Data platform, you must have the following command-line interfaces:

- OpenShift command-line interface (`oc`)
 - Cloud Pak for Data command-line interface (`cpd-cli`)
1. [Installing the IBM Cloud Pak for Data command-line interface](#)
To install IBM Cloud Pak for Data software on your Red Hat OpenShift Container Platform cluster, you must install the Cloud Pak for Data command-line interface (`cpd-cli`) on the workstation from which you are running the installation commands.
 2. [Installing the OpenShift command-line interface](#)
The IBM Cloud Pak for Data command-line interface (`cpd-cli`) interacts with the OpenShift command-line interface (`oc` CLI) to issue commands to your Red Hat OpenShift Container Platform cluster.

Related tasks

- [Collecting required information](#)
- [Installing the IBM Cloud Pak for Data platform and services](#)
- [Uninstalling the platform and services](#)


Related reference

- [Preparing your cluster](#)

Installing the IBM Cloud Pak for Data command-line interface

To install IBM Cloud Pak for Data software on your Red Hat® OpenShift® Container Platform cluster, you must install the Cloud Pak for Data command-line interface (`cpd-cli`) on the workstation from which you are running the installation commands.

Installation phase

-  Setting up a client workstation
 - Collecting required information
 - Preparing your cluster
 - Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

User	Purpose
Cluster administrators	<ul style="list-style-type: none">• Required to create catalog sources and operator subscriptions.• Required to change node settings
Project administrators	<ul style="list-style-type: none">• Required to install IBM Cloud Pak for Data• Required to integrate with the IAM Service (if applicable)
Private container registry administrators	Required to mirror images to the private container registry.

When do you need to complete this task?

You must complete this task on any workstation from which you plan to run installation commands.
You can also complete this task if you need to use the `cpd-cli` to complete other tasks, such as backing up and restoring your installation or managing users.

Before you begin

Ensure that the workstation meets the requirements in [Setting up a client workstation](#).

About this task

Install the `cpd-cli` on a client workstation that can connect to your cluster.

Procedure

1. Download Version 11.0.0 of the `cpd-cli` from the [IBM/cpd-cli](#) repository on GitHub.
Ensure that you download the correct package:

Workstation operating system	Enterprise Edition	Standard Edition
Linux®	<code>cpd-cli-linux-EE-11.0.0.tgz</code>	<code>cpd-cli-linux-SE-11.0.0.tgz</code>
Mac OS	<code>cpd-cli-darwin-EE-11.0.0.tgz</code>	<code>cpd-cli-darwin-SE-11.0.0.tgz</code>
Windows	You must download the Linux package and run it in Windows Subsystem for Linux: <code>cpd-cli-linux-EE-11.0.0.tgz</code>	You must download the Linux package and run it in Windows Subsystem for Linux: <code>cpd-cli-linux-SE-11.0.0.tgz</code>

Restriction: Do not download the Power® (`ppc64le`) or IBM Z® (`s390x`) packages if you plan to use the client to run an installation or upgrade. The `cpd-cli manage` commands cannot be run on these operating systems.

2. Extract the contents of the package to the directory where you want to run the `cpd-cli`.
3. On Mac OS, you must trust the following components of the `cpd-cli`:

- cpd-cli
- plugins/lib/darwin/cpdbr
- plugins/lib/darwin/manage
- plugins/lib/darwin/cpdtool
- plugins/lib/darwin/cpdbr-oadp
- plugins/lib/darwin/platform-mgmt
- plugins/lib/darwin/platform-diag
- plugins/lib/darwin/config

For each component:

- Right-click the component and select Open.
You will see a message with the following format:

```
macOS cannot verify the developer of "component-name". Are you sure you want to open it?
```

- Click Open.

4. Best practice Make the `cpd-cli` executable from any directory.

By default, you must either change to the directory where the `cpd-cli` is located or specify the fully qualified path of the `cpd-cli` to run the commands.

However, you can make the `cpd-cli` executable from any directory so that you only need to type `cpd-cli` *command-name* to run the commands.

Workstation operating system	Details
Linux	Add the following line to your <code>~/.bashrc</code> file: <code>export PATH=<fully-qualified-path-to-the-cpd-cli>:\$PATH</code>
Mac OS	Add the following line to your <code>~/.bash_profile</code> file: <code>export PATH=<fully-qualified-path-to-the-cpd-cli>:\$PATH</code>
Windows	From the Windows Subsystem for Linux, add the following line to your <code>~/.bashrc</code> file: <code>export PATH=<fully-qualified-path-to-the-cpd-cli>:\$PATH</code>

Next topic: [Installing the OpenShift command-line interface](#)

Installing the OpenShift command-line interface

The IBM® Cloud Pak for Data command-line interface (`cpd-cli`) interacts with the OpenShift command-line interface (`oc` CLI) to issue commands to your Red Hat® OpenShift Container Platform cluster.

Installation phase

- 📍 Setting up a client workstation
 - Collecting required information
 - Preparing your cluster
 - Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

Any users who are completing Cloud Pak for Data installation tasks.

When do you need to complete this task?

You must complete this task on any workstation from which you plan to run installation commands.

You must install a version of the `oc` CLI that is compatible with your Red Hat OpenShift Container Platform cluster.

Note: The links to the installation instructions are provided for your convenience. If the provider changes the location of the documentation, the links might not work.

- >
Self-managed clusters

Install the version of the `oc` CLI that corresponds to the version of Red Hat OpenShift Container Platform that you are running.

OpenShift Container Platform Version	Installation instructions
Version 4.6	See Getting started with the OpenShift CLI in the Red Hat OpenShift Container Platform documentation.
Version 4.8	See Getting started with the OpenShift CLI in the Red Hat OpenShift Container Platform documentation.
Version 4.10	See Getting started with the OpenShift CLI in the Red Hat OpenShift Container Platform documentation.

- >
 Managed OpenShift clusters

Follow the appropriate guidance for your managed OpenShift environment.

OpenShift environment	Installation instructions
Red Hat OpenShift on IBM Cloud®	See Installing the OpenShift CLI in the IBM Cloud documentation.
Microsoft Azure Red Hat OpenShift (ARO)	See Install the OpenShift CLI in the Microsoft Azure Red Hat OpenShift documentation
Red Hat OpenShift Service on AWS (ROSA)	See Getting started with the ROSA CLI in the Red Hat OpenShift Service on AWS documentation.

Previous topic: [Installing the IBM Cloud Pak for Data command-line interface](#)

Collecting required information

To successfully install IBM® Cloud Pak for Data, you must have specific information about your environment. Complete the following tasks to ensure that you have the information that you need.

- [Obtaining your IBM entitlement API key](#)
 All IBM Cloud Pak for Data images are accessible from the IBM Entitled Registry. The IBM entitlement API key enables you to pull software images from the IBM Entitled Registry, either for installation or for mirroring to a private container registry.
- [Determining which components to install](#)
 IBM Cloud Pak for Data is comprised of numerous components so that you can install the specific services that support your needs. Before you install Cloud Pak for Data, determine which components you need to install.
- [Setting up installation environment variables](#)
 The commands for installing and upgrading IBM Cloud Pak for Data use variables with the format `${VARIABLE_NAME}`. You can create a script to automatically export the appropriate values as environment variables before you run the installation commands. After you source the script, you will be able to copy most install and upgrade commands from the documentation and run them without making any changes.

Related tasks

- [Installing the IBM Cloud Pak for Data platform and services](#)
- [Uninstalling the platform and services](#)


Related reference

- [Setting up a client workstation](#)
- [Preparing your cluster](#)

Obtaining your IBM entitlement API key

All IBM® Cloud Pak for Data images are accessible from the IBM Entitled Registry. The IBM entitlement API key enables you to pull software images from the IBM Entitled Registry, either for installation or for mirroring to a private container registry.

Installation phase

- Setting up a client workstation
-  Collecting required information
- Preparing your cluster
- Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

Any users who are completing Cloud Pak for Data installation tasks that require the API key.

When do you need to complete this task?

You must complete this task if you don't have your IBM entitlement API key.

IBM entitlement API key

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

To obtain the IBM entitlement API key that is associated with your My IBM account:

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, complete [Determining which components to install](#).


Next topic: [Determining which components to install](#)

Determining which components to install

IBM® Cloud Pak for Data is comprised of numerous components so that you can install the specific services that support your needs. Before you install Cloud Pak for Data, determine which components you need to install.

You can use this information to complete [Setting up installation environment variables](#).

Installation phase

- Setting up a client workstation
-  Collecting required information
- Preparing your cluster
- Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

Everyone involved in installing Cloud Pak for Data should agree on the components that will be installed on the cluster.

When do you need to complete this task?

Complete this task before you complete either of the following tasks:

- Mirroring images to a private container registry
- Installing the Cloud Pak for Data software on your cluster

Options for installing components

You have two options for installing the components:

Option	Benefits	Drawbacks
--------	----------	-----------

Option	Benefits	Drawbacks
Install each component individually	If you feel more comfortable running installs one at a time, this option gives you more granular control over the install process. Note: All of the services that you install must be installed at the same release. You cannot install the services at different releases .	You must complete more steps to successfully install the software on your environment.
Install multiple components at the same time	You can complete the installation in fewer steps.	There are no known drawbacks associated with this option. If you encounter an issue when installing a specific component, the <code>cpd-cli</code> gives you the option to resume your install from the point of failure.

Required components

At a minimum, you must install the following components:

Software	Component ID	Notes
IBM Cloud Pak® foundational services	<code>cpfs</code>	Required. This component is a prerequisite for IBM Cloud Pak for Data. The component is installed once on the cluster and is shared by any instances of Cloud Pak for Data on the cluster. If the <code>cpd-cli</code> detects the minimum required version of the <code>cpfs</code> component on the cluster, it does not attempt to install it again.
Scheduling service	<code>scheduler</code>	Required if you plan to: <ul style="list-style-type: none"> • Install Watson™ Machine Learning Accelerator • Use the quota enforcement feature in Cloud Pak for Data The component is installed once on the cluster and is shared by any instances of Cloud Pak for Data on the cluster. If the <code>cpd-cli</code> detects the minimum required version of <code>scheduler</code> component on the cluster, it does not attempt to install it again.
IBM Cloud Pak for Data	<code>cpd_platform</code>	Required. This component is a prerequisite for installing any services. The component ensures that the Cloud Pak for Data control plane is installed and running. The component is installed once in each project (namespace) where you want to install the platform. If the <code>cpd-cli</code> detects the minimum required version of <code>cpd_platform</code> in the project, it does not attempt to install it again.

Important: The sequence in which you install the components is important.

Batch installs

If you plan to install all of the components at the same time, ensure that the components are specified in the following order:

```
--components=cpfs , scheduler , cpd_platform , <component-ID> . . .
```

If you don't plan to install the scheduling service, you can remove it from the list of components.

Individual component installs

If you plan to install each component individually, ensure that you install the components in the following order:

1. `cpfs`
2. `scheduler`

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

3. `cpd_platform`

The other components that you install depend on your use case. If you are installing the services required to support a particular solution, see the appropriate section for the solution:

- [Business Analytics](#)
- [Customer Care](#)
- [Data Fabric](#)
- [Data Management](#)

If you are designing your own solution, see [All services](#)

Components for the Business Analytics solution

The Business Analytics solution supports several use cases. The services that you install depend on the use cases that you want to implement:

- | >
Business Intelligence | |
|----------------------------|-------------------------------|
| Software | Component ID |
| Cognos® Analytics | <code>cognos_analytics</code> |
- | >
Planning, Budgeting, and Forecasting | |
|---|---------------------------------|
| Software | Component ID |
| Planning Analytics | <code>planning_analytics</code> |

Components for the Customer Care solution

The Customer Care solution supports several use cases. The services that you install depend on the use cases that you want to implement:

- | >
Content Intelligence | |
|---------------------------|-------------------------------|
| Software | Component ID |
| Watson Discovery | <code>watson_discovery</code> |
- | >
Conversational AI | |
|------------------------|-------------------------------|
| Software | Component ID |
| Watson Assistant | <code>watson_assistant</code> |
- | >
Speech | |
|------------------------|----------------------------|
| Software | Component ID |
| Watson Speech services | <code>watson_speech</code> |

Components for the Data Fabric solution

The Data Fabric supports several use cases. The services that you install depend on the use cases that you want to implement:

- | >
Customer 360 | | |
|---------------------------|-----------------------|-------|
| Software | Component ID | Notes |
| Data Virtualization | <code>dv</code> | |
| IBM Match 360 with Watson | <code>match360</code> | |

Software	Component ID	Notes
Watson Knowledge Catalog	wkc	When you install Watson Knowledge Catalog, the following services are automatically installed: <ul style="list-style-type: none"> Data Refinery (dr)

Optional components

If you want to use [dashboards](#) to share analytics results, you can optionally install the Cognos Dashboards component:

Software	Component ID
Cognos Dashboards	cde

- > Data Governance and Privacy

Software	Component ID	Notes
Data Privacy	dp	Before you install or upgrade this service, you must have the following services installed: <ul style="list-style-type: none"> Analytics Engine Powered by Apache Spark (analyticsengine), which is automatically installed by Watson Knowledge Catalog. Watson Knowledge Catalog (wkc) If you plan to run a batch installation or upgrade, specify the components in the following order: wkc , dp
Data Virtualization	dv	
Watson Knowledge Catalog	wkc	When you install Watson Knowledge Catalog, the following services are automatically installed: <ul style="list-style-type: none"> Data Refinery (dr)

- > MLOps and Trustworthy AI

Software	Component ID	Notes
Watson Knowledge Catalog	wkc	When you install Watson Knowledge Catalog, the following services are automatically installed: <ul style="list-style-type: none"> Data Refinery (dr)
Watson Machine Learning	wml	
Watson OpenScale	openscale	
Watson Studio	ws	When you install Watson Studio, the following services are automatically installed: <ul style="list-style-type: none"> Data Refinery (dr)

- > Multicloud Data Integration

Software	Component ID	Notes
Data Virtualization	dv	
DataStage® Enterprise	datastage_ent	
Watson Knowledge Catalog	wkc	When you install Watson Knowledge Catalog, the following services are automatically installed: <ul style="list-style-type: none"> Data Refinery (dr)

Optional components

If you want to use [QualityStages](#), such as the Address Verification stage, you can optionally upgrade from DataStage Enterprise to DataStage Enterprise Plus:

Software	Component ID
DataStage Enterprise Plus	datastage_ent_plus

Components for the Data Management solution

The Data Management solution includes a variety of data storage options. Choose the components that support your business needs:

- | > | |
|------------------------|--------------|
| Analytics data sources | |
| Software | Component ID |
| Data Virtualization | dv |
| Db2® Warehouse | db2wh |
- | > | |
|----------------------------|----------------------|
| Transactional data sources | |
| Software | Component ID |
| Db2 | db2oltp |
| Informix® | informix_cp4d |
- | > | |
|------------------|---------------------|
| OEM data sources | |
| Software | Component ID |
| EDB Postgres | edb_cp4d |
| MongoDB | mongodb_cp4d |

All services

You can install a custom set of services based on your business needs.

Software	Component IDs	Notes
Analytics Engine Powered by Apache Spark	analyticsengine	This service is automatically installed or upgraded if you install Watson Knowledge Catalog (wkc)
Cognos Analytics	cognos_analytics	
Cognos Dashboards	cde	
Data Privacy	dp	<p>Before you install or upgrade this service, you must have the following services installed:</p> <ul style="list-style-type: none"> Analytics Engine Powered by Apache Spark (analyticsengine), which is automatically installed by Watson Knowledge Catalog. Watson Knowledge Catalog (wkc) <p>If you plan to run a batch installation or upgrade, specify the components in the following order:</p> <p>wkc , dp</p>
Data Refinery	dr	<p>This service is automatically installed or upgraded if you install either of the following services:</p> <ul style="list-style-type: none"> Watson Knowledge Catalog Watson Studio <p>If you complete the following actions for either of these services, the Data Refinery objects are automatically included:</p> <ul style="list-style-type: none"> Mirror images Create catalog sources or operator subscriptions
Data Virtualization	dv	
DataStage Enterprise	datastage_ent	
DataStage Enterprise Plus	datastage_ent_plus	
Db2	db2oltp	

Software	Component IDs	Notes
Db2 Big SQL	<code>bigsql</code>	
Db2 Data Gate	<code>datagate</code>	
Db2 Data Management Console	<code>dmc</code>	
Db2 Warehouse	<code>db2wh</code>	
Decision Optimization	<code>dods</code>	
EDB Postgres	<ul style="list-style-type: none"> <code>edb_cp4d</code> <code>postgres</code> 	The <code>postgresql</code> component is automatically installed when you install the <code>edb_cp4d</code> component.
Execution Engine for Apache Hadoop	<code>hee</code>	
IBM Match 360 with Watson	<code>match360</code>	
Informix	<ul style="list-style-type: none"> <code>informix_cp4d</code> <code>informix</code> 	The <code>informix</code> component is automatically installed when you install the <code>informix_cp4d</code> component.
MongoDB	<ul style="list-style-type: none"> <code>mongodb</code> <code>mongodb_cp4d</code> 	You must specify both components to install MongoDB. Specify the components in the following order: <code>mongodb,mongodb_cp4d</code>
OpenPages®	<code>openpages</code>	
Planning Analytics	<code>planning_analytics</code>	
Product Master	<code>productmaster</code>	
RStudio® Server with R 3.6	<code>rstudio</code>	
SPSS® Modeler	<code>spss</code>	
Watson Assistant	<code>watson_assistant</code>	
Watson Discovery	<code>watson_discovery</code>	
Watson Knowledge Catalog	<code>wkc</code>	When you install Watson Knowledge Catalog, the following services are automatically installed: <ul style="list-style-type: none"> Data Refinery (<code>dr</code>)
Watson Machine Learning	<code>wml</code>	
Watson Machine Learning Accelerator	<code>wml_accelerator</code>	
Watson OpenScale	<code>openscale</code>	
Watson Speech services	<code>watson_speech</code>	
Watson Studio	<code>ws</code>	When you install Watson Studio, the following services are automatically installed: <ul style="list-style-type: none"> Data Refinery (<code>dr</code>)
Watson Studio Runtimes	<code>ws_runtimes</code>	


Previous topic: [Obtaining your IBM entitlement API key](#)

Next topic: [Setting up installation environment variables](#)

Setting up installation environment variables

The commands for installing and upgrading IBM® Cloud Pak for Data use variables with the format `${VARIABLE_NAME}`. You can create a script to automatically export the appropriate values as environment variables before you run the installation commands. After you source the script, you will be able to copy most install and upgrade commands from the documentation and run them without making any changes.

Installation phase

- Setting up a client workstation
-  Collecting required information
- Preparing your cluster
- Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

Any users who will run installation or upgrade commands that require information about your environment, such as information about your Red Hat® OpenShift® Container Platform cluster or your private container registry.

When do you need to complete this task?

You should complete this task before you start preparing your cluster.

Before you begin

Before you create the script, consider whether you need to support:

Repeatable deployments across clusters

In this scenario, you can re-use the same script to improve the consistency of deployments across your environments.

Multiple deployments on the same cluster

In this scenario, you can create multiple scripts to simplify the process of managing each instance. Clearly name each script to ensure that you run the correct script before you run the installation or upgrade commands.

Tip: If multiple people are working together to complete the installation, you should share a copy of the appropriate files with each user. Each user can edit the scripts to supply their own credentials and [source the script](#) on their own workstation.

Creating an environment variables file

1. Copy the following example to a text editor on your local file system:

```
#####
# Cloud Pak for Data installation variables
#####

# -----
# Cluster
# -----

export OCP_URL=<enter your Red Hat OpenShift Container Platform URL>
export OPENSIFT_TYPE=<enter your deployment type>
# export OCP_USERNAME=<enter your username>
# export OCP_PASSWORD=<enter your password>
# export OCP_TOKEN=<enter your token>

# -----
# Projects
# -----

export PROJECT_CPFS_OPS=ibm-common-services
export PROJECT_CPD_OPS=<enter your Cloud Pak for Data operator installation project>
export PROJECT_CATSRC=openshift-marketplace
export PROJECT_CPD_INSTANCE=<enter your Cloud Pak for Data installation project>
# export PROJECT_TETHERED=<enter the tethered project>
```

```

# -----
# Storage
# -----

export STG_CLASS_BLOCK=<RWO-storage-class-name>
export STG_CLASS_FILE=<RWX-storage-class-name>

# -----
# IBM Entitled Registry
# -----

export IBM_ENTITLEMENT_KEY=<enter your IBM entitlement API key>

# -----
# Private container registry
# -----
# Set the following variables if you mirror images to a private container registry.
#
# To export these variables, you must uncomment each command in this section.

# export PRIVATE_REGISTRY_LOCATION=<enter the location of your private container
registry>
# export PRIVATE_REGISTRY_PUSH_USER=<enter the username of a user that can push to the
registry>
# export PRIVATE_REGISTRY_PUSH_PASSWORD=<enter the password of the user that can push to
the registry>
# export PRIVATE_REGISTRY_PULL_USER=<enter the username of a user that can pull from the
registry>
# export PRIVATE_REGISTRY_PULL_PASSWORD=<enter the password of the user that can pull
from the registry>

# -----
# Cloud Pak for Data version
# -----

export VERSION=4.5.0

# -----
# Components
# -----
# Set the following variable if you want to install or upgrade multiple components at the
same time.
#
# To export the variable, you must uncomment the command.

# export COMPONENTS=cpfs,scheduler,cpd_platform,<component-ID>

```

2. Update each section in the script for your environment. See the following sections to learn about the variables and valid values in each section of the script:

- [Cluster](#)
- [Projects](#)
- [Storage](#)
- [IBM Entitled Registry](#)
- [Private container registry](#)
- [Cloud Pak for Data version](#)
- [Components](#)

3. Save the file as a shell script. For example, save the file as `cpd_vars.sh`.

4. Confirm that the script does not contain any errors. For example, if you named the script `cpd_vars.sh`, run:

```
bash ./cpd_vars.sh
```

5. If you stored passwords in the file, prevent others from reading the file. For example, if you named the script `cpd_vars.sh`, run:

```
chmod 700 cpd_vars.sh
```

Sourcing the environment variables

Save a copy of the script to your workstation and run it from a bash prompt before you the run installation and upgrade commands. The script exports the environment variables to your command-line session.

Important: You must re-run the script each time you open a new bash prompt.

1. Change to the directory where you saved the script.
2. Source the environment variables. For example, if you named the script `cpd_vars.sh`, run:

```
source ./cpd_vars.sh
```

Cluster

The variables in the **Cluster** section of the script specify information about your Red Hat OpenShift Container Platform cluster.

Need more information on installing Red Hat OpenShift Container Platform? See:

- [Installing Red Hat OpenShift Container Platform](#)

Variable	Description
<code>OCP_URL</code>	<p>The URL of the Red Hat OpenShift Container Platform server. For example, <code>https://openshift1.example.com:8443</code>.</p> <p>Default value There is no default value.</p> <p>Valid values Specify the URL of your Red Hat OpenShift Container Platform server.</p>
<code>OPENSSHIFT_TYPE</code>	<p>The type of Red Hat OpenShift Container Platform cluster that you are running.</p> <p>Default value self-managed</p> <p>Valid values</p> <ul style="list-style-type: none">aro Specify <code>aro</code> if you are running Microsoft Azure Red Hat OpenShift (ARO), the managed OpenShift offering on Microsoft Azure.roks Specify <code>roks</code> if you are running Red Hat OpenShift on IBM Cloud®, the managed OpenShift offering on IBM Cloud.rosa Specify <code>rosa</code> if you are running Red Hat OpenShift Service on AWS (ROSA), the managed OpenShift offering on Amazon Web Services.self-managed Specify <code>self-managed</code> if you are running self-managed OpenShift on:<ul style="list-style-type: none">• On-premises infrastructure• AWS infrastructure• IBM Cloud infrastructure• Microsoft Azure infrastructure

Variable	Description
<code>OCP_USERNAME</code>	<p>The username that you use to authenticate to your cluster. You must have sufficient privileges to complete each installation or upgrade task.</p> <p>This variable is used to run the <code>cpd-cli manage login-to-ocp</code> command. The</p> <p>Tip: It is recommended that you prevent other users from reading the contents of the environment variable script by running <code>chmod 700</code>. However, if you still have concerns about storing your OpenShift credentials in this file, you can:</p> <ul style="list-style-type: none"> • Enter the credentials directly instead of using the environment variable in the commands. • Manually export the credentials before you run the commands. <p>Remember: You must uncomment the <code>export OCP_USERNAME</code> command if you want to use this environment variable.</p>
<code>OCP_PASSWORD</code>	<p>The password that you use to authenticate to your cluster.</p> <p>Remember: You must uncomment the <code>export OCP_PASSWORD</code> command if you want to use this environment variable.</p>
<code>OCP_TOKEN</code>	<p>Remember: You must uncomment the <code>export OCP_TOKEN</code> command if you want to use this environment variable.</p>

Projects

The variables in the **Projects** section of the script specify where the components that comprise Cloud Pak for Data are installed.

Need more information about projects? See:

- [Supported project \(namespace\) configurations](#)
- [Setting up projects \(namespaces\) on Red Hat OpenShift Container Platform](#)

Variable	Description
<code>PROJECT_CPFS_OPS</code>	<p>The Red Hat OpenShift project where the IBM Cloud Pak® foundational services operators are installed.</p> <p>Default value <code>ibm-common-services</code></p> <p>Valid values Change the value of <code>PROJECT_CPFS_OPS</code> only if you installed the IBM Cloud Pak foundational services in a different project.</p>
<code>PROJECT_CPD_OPS</code>	<p>The Red Hat OpenShift project where the Cloud Pak for Data software operators are installed.</p> <p>Default value There is no default value.</p> <p>Valid values The location of the operators depends on the operator installation architecture that you want to use:</p> <p>Express installation In an express installation, the Cloud Pak for Data operators are installed in the same project as the IBM Cloud Pak foundational services operators: <code>ibm-common-services</code></p> <p>Specialized installation In a specialized installation, the Cloud Pak for Data operators are installed in a separate project from the IBM Cloud Pak foundational services operators. The recommended project name is <code>cpd-operators</code>.</p>

Variable	Description
<code>PROJECT_CATS_RC</code>	<p>The Red Hat OpenShift project where the catalog sources are installed.</p> <p>Default value <code>openshift-marketplace</code></p> <p>Valid values The only supported value is <code>openshift-marketplace</code>.</p>
<code>PROJECT_CPD_INSTANCE</code>	<p>The Red Hat OpenShift project where the Cloud Pak for Data control plane and services are installed. (The Cloud Pak for Data control plane is installed in a <i>separate</i> project from the operators.)</p> <p>Default value There is no default value.</p> <p>Valid values <code>cpd-instance</code> is used as an example, but you can choose a name that works for you.</p> <p>Tip: If you want to install multiple instances of Cloud Pak for Data on your cluster, create a separate script for each instance that you plan to install.</p>
<code>PROJECT_TETHERED</code>	<p>A Red Hat OpenShift project that is tethered to the project where the Cloud Pak for Data control plane is installed.</p> <p>This variable is required only if you plan to install a service that supports tethering into a tethered project.</p> <p>Remember: You must uncomment the <code>export PROJECT_TETHERED</code> command if you want to use this environment variable.</p> <p>Default value There is no default value.</p> <p>Valid values <code>cpd-instance-tether</code> is sometimes used as an example, but you can choose a name that works for you.</p>

Storage

The variables in the **Storage** section of the script specify the storage classes that the installation should use.

Need more information about storage? See:

- [Storage considerations](#)
- [Storage requirements](#)
- [Setting up persistent storage](#)

Variable	Description
<code>STG_CLASS_BLOCK</code>	<p>The name of a block storage class on a supported storage option.</p> <p>Default value There is no default value.</p> <p>Valid values Specify the name of a storage class that points to block storage (storage that supports ReadWriteOnce, also called RWO, access). The following list provides the recommended storage classes for the supported storage options. If you use different storage classes, identify an equivalent storage class on the cluster.</p> <ul style="list-style-type: none"> • OpenShift Data Foundation: <code>ocs-storagecluster-ceph-rbd</code> • IBM Spectrum® Fusion: <code>ibm-spectrum-scale-sc</code> • IBM Spectrum Scale Container Native: <code>ibm-spectrum-scale-sc</code> • Portworx: <code>portworx-metastoredb-sc</code> • NFS: <code>managed-nfs-storage</code> • Amazon Elastic Block Store: <code>gp2-csi</code> or <code>gp3-csi</code> • IBM Cloud Block Storage: <code>ibmc-block-gold</code>

Variable	Description
<code>STG_CLASS_FILE</code>	<p>The name of a file storage class on a supported storage option.</p> <p>Default value There is no default value.</p> <p>Valid values Specify the name of a storage class that points to file storage (storage that supports ReadWriteMany, also called RWX, access). The following list provides the recommended storage classes for the supported storage options. If you use different storage classes, identify an equivalent storage class on the cluster.</p> <ul style="list-style-type: none"> • OpenShift Data Foundation: <code>ocs-storagecluster-cephfs</code> • IBM Spectrum Fusion: <code>ibm-spectrum-scale-sc</code> • IBM Spectrum Scale Container Native: <code>ibm-spectrum-scale-sc</code> • Portworx: <code>portworx-rwx-gp3-sc</code> • NFS: <code>managed-nfs-storage</code> • Amazon Elastic File System: <code>efs-nfs-client</code> • IBM Cloud File Storage: <code>ibmc-file-gold-gid</code> or <code>ibm-file-custom-gold-gid</code>

IBM Entitled Registry

The variables in the **IBM Entitled Registry** section of the script enable you to connect to the IBM Entitled Registry and access the Cloud Pak for Data software images that you are entitled to.

Depending on whether you pull images from the IBM Entitled Registry or from a private container registry, the variables might also be used to configure the global image pull secret.

Need more information about the IBM Entitled Registry? See:

- [Obtaining your IBM entitlement API key](#)

Variable	Description
<code>IBM_ENTITLEMENT_KEY</code>	<p>The entitlement API key that is associated with your My IBM account.</p> <p>Default value There is no default value.</p> <p>Valid values Specify your IBM entitlement API key.</p>

Private container registry

It is strongly recommended that you use a private container registry. The variables in the **Private container registry** section are required only if you mirror images to a private container registry.

The variables in the **Private container registry** section of the script enable you to mirror images from the IBM Entitled Registry to the private container registry.

Important: If you want to use these variables, you must uncomment each command in the **Private container registry** section of the script.

Need more information about private container registries? See:

- [Private container registry requirements](#)

Variable	What to specify
----------	-----------------

Variable	What to specify
<code>PRIVATE_REGISTRY_LOCATION</code>	<p>The location of the private container registry.</p> <p>Default value There is no default value.</p> <p>Valid values Specify the fully qualified location of the private container registry.</p>
<code>PRIVATE_REGISTRY_PUSH_USER</code>	<p>The username of a user who has the required privileges to <i>push</i> images to the private container registry.</p> <p>Default value There is no default value.</p> <p>Valid values Specify a username.</p>
<code>PRIVATE_REGISTRY_PUSH_PASSWORD</code>	<p>The password of the user who has the required privileges to <i>push</i> images to the private container registry.</p> <p>Tip: It is recommended that you prevent other users from reading the contents of the environment variable script by running chmod 700. However, if you still have concerns about storing passwords in this file, you can:</p> <ul style="list-style-type: none"> • Enter the password directly instead of using the environment variable in the commands. • Manually export the password before you run the commands. <p>Default value There is no default value.</p> <p>Valid values Specify the password associated with the username.</p>
<code>PRIVATE_REGISTRY_PULL_USER</code>	<p>The username of a user who has the required privileges to <i>pull</i> images from the private container registry.</p> <p>Default value There is no default value.</p> <p>Valid values Specify a username.</p>
<code>PRIVATE_REGISTRY_PULL_PASSWORD</code>	<p>The password of the user who has the required privileges to <i>pull</i> images from the private container registry.</p> <p>Tip: It is recommended that you prevent other users from reading the contents of the environment variable script by running chmod 700. However, if you still have concerns about storing passwords in this file, you can:</p> <ul style="list-style-type: none"> • Enter the password directly instead of using the environment variable in the commands. • Manually export the password before you run the commands. <p>Default value There is no default value.</p> <p>Valid values Specify the password associated with the username.</p>

Cloud Pak for Data version

The variable in the **Cloud Pak for Data version** section specifies which version of Cloud Pak for Data to install or upgrade to.

Remember: All of the components must be installed at the same version.

Variable	Description
----------	-------------

Variable	Description
VERSION	The version of the Cloud Pak for Data software to install. Default value 4.5.0 Valid values <ul style="list-style-type: none"> 4.5.0

Components

If you want to install or upgrade multiple components at the same time, you can specify a comma separated list of components. The variable in the **Components** section is recommended to help you easily specify the list of components.

Important: If you want to use this variable, you must uncomment the command in the **Components** section of the script.

Variable	Description
COMPONENTS	The comma separated list of the components that you want to install or upgrade. Default value By default, the list includes the required and recommended components: cpfs,scheduler,cpd_platform You can optionally remove scheduler if you don't plan to install Watson™ Machine Learning Accelerator or if you don't plan to use the quota enforcement feature. Important: You must specify cpfs and cpd_platform . These components are required for all installations. Do not add any components to the comma separated list before cpfs or cpd_platform . Only add components after the cpd_platform entry. This ensures that the components are installed in the correct order. Valid values Review the guidance in Determining which components to install to determine which components to specify.

Previous topic: [Determining which components to install](#)

Preparing your cluster

Before you install IBM® Cloud Pak for Data, complete the following tasks to prepare your cluster.

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 persistent storage](#)
Before you can install IBM 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, an administrator must create and configure the OpenShift projects (Kubernetes namespaces) where you plan to deploy the Cloud Pak for Data software.
4. [Creating custom security context constraints for services](#)
Most Cloud Pak for Data services use the **restricted** security context constraint (SCC) that is provided by Red Hat OpenShift Container Platform. However, if you plan to install certain Cloud Pak for Data services, you might need to create one or more custom SCCs.
5. [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.

6. [Updating 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 add to the global image pull secret depend on where you want to pull images from.

7. [Mirroring images to a 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.

Related tasks

- [Collecting required information](#)
- [Installing the IBM Cloud Pak for Data platform and services](#)
- [Uninstalling the platform and services](#)

Related reference

- [Setting up a client workstation](#)

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.

Installation phase

Setting up a client workstation

Collecting required information



Preparing your cluster

Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

The user who will be the primary cluster administrator should install the Red Hat OpenShift Container Platform cluster.

When do you need to complete this task?

If you don't have an existing Red Hat OpenShift Container Platform cluster, you must complete this task before you can install IBM Cloud Pak for Data.

If you already have an existing Red Hat OpenShift Container Platform cluster, you can skip this task.

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:

Deployment environment	Managed OpenShift	Self-managed OpenShift
On-premises	✓	✓
IBM Cloud	✓	✓
Amazon Web Services (AWS)	✓	✓
Microsoft Azure	✓	✓
Google Cloud		✓

The following sections include additional guidance about how to install managed and self-managed OpenShift on each deployment environment.

- [On-premises infrastructure](#)

OpenShift type	Deployment options
----------------	--------------------

OpenShift type	Deployment options
Managed	You can deploy a managed OpenShift cluster on your on-premises infrastructure with IBM Cloud Satellite .
Self-managed	You can deploy a self-managed OpenShift cluster on your on-premises infrastructure by following the Red Hat OpenShift Container Platform documentation: <ul style="list-style-type: none"> ◦ Version 4.6 documentation ◦ Version 4.8 documentation ◦ Version 4.10 documentation Additional guidance on setting up OpenShift is available in the IBM Cloud Paks documentation .

- >
IBM Cloud infrastructure

OpenShift type	Deployment options
Managed	You can deploy a managed OpenShift cluster on IBM Cloud infrastructure from the IBM Cloud catalog. For details, see Red Hat OpenShift on IBM Cloud® in the IBM Cloud catalog. Ensure that you select a supported version of Red Hat OpenShift. You can deploy Cloud Pak for Data on the following infrastructure: <ul style="list-style-type: none"> ◦ Classic infrastructure single zone or multizone ◦ Virtual Private Cloud (VPC) Gen2 single or multi zone Remember: The infrastructure that you choose determines the storage options that are available, which determines the services that you can install.
Self-managed	You can deploy a self-managed OpenShift cluster on IBM Cloud infrastructure by following the Red Hat OpenShift Container Platform documentation: <ul style="list-style-type: none"> ◦ Version 4.6 documentation ◦ Version 4.8 documentation ◦ Version 4.10 documentation Additional guidance on setting up OpenShift is available in the IBM Cloud Paks documentation .

- >
AWS infrastructure

OpenShift type	Deployment options
Managed	You can deploy a managed OpenShift cluster on AWS infrastructure using one of the following deployment methods: <p>rosa CLI</p> For details, see Red Hat OpenShift Service on AWS (ROSA) in the Red Hat OpenShift Container Platform documentation. <p>IBM Cloud Satellite</p> For details, see IBM Cloud Satellite in the IBM Cloud marketplace.
Self-managed	You can deploy a self-managed OpenShift cluster on IBM Cloud infrastructure by following the Red Hat OpenShift Container Platform documentation: <ul style="list-style-type: none"> ◦ Version 4.6 documentation ◦ Version 4.8 documentation ◦ Version 4.10 documentation Additional guidance on setting up OpenShift is available in the IBM Cloud Paks documentation .

- >
Azure infrastructure

OpenShift type	Deployment options
Managed	You can deploy a managed OpenShift cluster on Azure infrastructure using the <code>az aro</code> CLI. For details, see Microsoft Azure Red Hat OpenShift 4 (ARO) in the Red Hat OpenShift Container Platform documentation.

OpenShift type	Deployment options
Self-managed	<p>You can deploy a self-managed OpenShift cluster on IBM Cloud infrastructure by following the Red Hat OpenShift Container Platform documentation:</p> <ul style="list-style-type: none"> ◦ Version 4.6 documentation ◦ Version 4.8 documentation ◦ Version 4.10 documentation <p>Additional guidance on setting up OpenShift is available in the IBM Cloud Paks documentation.</p>

- >
Google Cloud infrastructure

OpenShift type	Deployment options
Managed	Managed OpenShift on Google Cloud infrastructure is not supported.
Self-managed	<p>You can deploy a self-managed OpenShift cluster on Google Cloud infrastructure by following the Red Hat OpenShift Container Platform documentation:</p> <ul style="list-style-type: none"> ◦ Version 4.6 documentation ◦ Version 4.8 documentation ◦ Version 4.10 documentation <p>Additional guidance on setting up OpenShift is available in the IBM Cloud Paks documentation.</p>

What's next

Now that you've installed Red Hat OpenShift Container Platform, you are ready to complete [Setting up persistent storage](#).

Next topic: [Setting up persistent storage](#)


Setting up persistent storage

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

Installation phase

Setting up a client workstation

Collecting required information

 Preparing your cluster

Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

A cluster administrator or storage administrator must set up set up persistent storage on the Red Hat OpenShift Container Platform cluster.

When do you need to complete this task?

If you don't have supported storage configured for your cluster, you must complete this task before you can install IBM Cloud Pak for Data.

If you already running supported storage confirm that it is properly configured.

Important: Before you set up 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](#), which identifies the type of storage each service supports.
- [Hardware requirements](#), which specifies the minimum amount of storage that each service requires.

The type of storage that you want to use and complete the appropriate steps to install and configure the storage.

Storage option	Documentation links
----------------	---------------------

Storage option	Documentation links
OpenShift Data Foundation	<p>Installation</p> <p>To install OpenShift Data Foundation, see the Red Hat OpenShift Data Foundation documentation.</p> <p>Post-installation set up</p> <p>No additional set up is required.</p> <p>You are ready to complete Setting up projects (namespaces) on Red Hat OpenShift Container Platform.</p>
OpenShift Data Foundation as a Service	<p>Installation</p> <p>Contact IBM Support.</p> <p>Post-installation setup</p> <p>Contact IBM Support.</p>
IBM Spectrum® Fusion	<p>Installation</p> <p>To install IBM Spectrum Fusion, see the IBM Spectrum Fusion installation overview in the IBM Spectrum Fusion documentation.</p> <p>Post-installation set up</p> <p>Create the specified storage class with the required parameters. For details, see Setting up IBM Spectrum Scale Container Native storage.</p>
IBM Spectrum Scale Container Native (with IBM Spectrum Scale Container Storage Interface)	<p>Installation</p> <p>To install IBM Spectrum Scale and IBM Spectrum Scale Container Storage Interface, see Installing the IBM Spectrum Scale container native operator and cluster (Version 5.1.1.3) in the IBM Spectrum Scale Container Native documentation.</p> <p>Post-installation set up</p> <p>Create the specified storage class with the required parameters. For details, see Setting up IBM Spectrum Scale Container Native storage.</p>
Portworx	<p>Installation</p> <p>To install Portworx Enterprise, see Install Portworx on OpenShift in the Portworx documentation.</p> <p>Post-installation set up</p> <p>Create the specified storage classes with the required parameters. For details, see Creating Portworx storage classes.</p>
NFS	<p>Installation</p> <p>Refer to the installation documentation for your NFS storage provider.</p> <p>Post-installation set up</p> <p>At a minimum, you must set up dynamic storage provisioning. Depending on the services you plan to install, you might need to configure your NFS server. For details, see Configuring your NFS server.</p>
Amazon Elastic Block Store (EBS)	<p>Installation</p> <p>EBS is provisioned by default when you install a Red Hat OpenShift Container Platform cluster on AWS.</p> <p>Post-installation setup</p> <p>No additional set up is required.</p> <p>You are ready to complete Setting up projects (namespaces) on Red Hat OpenShift Container Platform.</p>
Amazon Elastic File System (EFS)	<p>Installation</p> <p>Install EFS from the AWS Console. It is recommended that you create a regional file system.</p> <p>Post-installation setup</p> <p>You must set up dynamic storage provisioning. For details, see Setting up Amazon Elastic File System.</p>

Storage option	Documentation links
IBM Cloud Block Storage	<p>Installation</p> <p>IBM Cloud Block Storage is provisioned by default when you install a Red Hat OpenShift Container Platform cluster on IBM Cloud.</p> <p>Post-installation setup</p> <p>No additional set up is required.</p> <p>You are ready to complete Setting up projects (namespaces) on Red Hat OpenShift Container Platform.</p>
IBM Cloud File Storage	<p>Installation</p> <p>When you configure your Red Hat OpenShift cluster on IBM Cloud, ensure that you select one of the following IBM Cloud File Storage storage classes:</p> <ul style="list-style-type: none"> • <code>ibmc-file-gold-gid</code> • <code>ibm-file-custom-gold-gid</code> <p>Post-installation set up</p> <p>Depending on the services you plan to install, you might need to configure your IBM Cloud File Storage. For details, see Setting up IBM Cloud File Storage.</p> <p>If you are running a production workload on your cluster, it is recommended that you adjust your I/O and performance and storage size:</p> <ul style="list-style-type: none"> • The default I/O settings are typically lower than the minimums specified in the Disk requirements section. <p>To improve the I/O performance for production environments, you must adjust the I/O settings. Contact IBM Software Support for guidance on how to adjust the settings according to Changing the size and IOPS of your existing storage device.</p> <ul style="list-style-type: none"> • Storage is not automatically expanded and is created in smaller chunks. Increasing the size of the volumes improves I/O performance for production environments. Contact IBM Software Support for assistance.

- [Setting up IBM Spectrum storage](#)
If you decide to use IBM Spectrum Fusion or IBM Spectrum Scale Container Native for persistent storage, you must create a storage class with the appropriate settings for use with IBM Cloud Pak for Data.
- [Creating Portworx storage classes](#)
If you decide to use Portworx as your storage option, Cloud Pak for Data recommends several storage classes for optimized performance and availability.
- [Setting up NFS storage](#)
If you plan to use NFS for persistent storage, you must set up your NFS storage before you install Cloud Pak for Data.
- [Setting up Amazon Elastic File System](#)
Amazon Elastic File System (EFS) does not support dynamic storage provisioning by default, and Red Hat OpenShift does not include a provisioner plug-in to create an NFS-based storage class. Therefore, you must set up dynamic storage provisioning on your Amazon Elastic File System.
- [Setting up IBM Cloud File Storage](#)
If you are installing services that depend on NFS and you are planning to use IBM Cloud File Storage on NFS 4 for persistent storage, you must configure ID mapping, which enables `no_root_squash`. Configuring `no_root_squash` allows root clients to retain root permissions on the remote NFS share.

Previous topic: [Installing Red Hat OpenShift Container Platform](#)

Next topic: [Setting up projects \(namespaces\) on Red Hat OpenShift Container Platform](#)


Setting up IBM Spectrum storage

If you decide to use IBM Spectrum® Fusion or IBM Spectrum Scale Container Native for persistent storage, you must create a storage class with the appropriate settings for use with IBM® Cloud Pak for Data.

Installation phase

Setting up a client workstation

Collecting required information

 Preparing your cluster

Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

A cluster administrator or a storage administrator must complete this task.

When do you need to complete this task?

If you plan to use IBM Spectrum storage and you do not have a storage class that meets the required criteria, you must complete this task before you install IBM Cloud Pak for Data.

About this task

The storage class must have read, write, and execute permissions on the local file system.

Remember: Storage classes are cluster-wide resources.

Procedure

1. Log in to your cluster as a user with sufficient permissions.
2. Set the following environment variables:

Variable	Notes
<code>FILE_SYSTEM_NAME</code>	<p>Specify the name of the file system that is mounted by the IBM Spectrum Fusion or IBM Spectrum Scale Container Native cluster.</p> <p>IBM Spectrum Fusion</p> <p>Run the following command to get the name of the file system on your Red Hat® OpenShift® Container Platform cluster:</p> <pre>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 -- mmlsfs all -T</pre> <p>IBM Spectrum Scale Container Native</p> <p>Run the following command to get the name of the file system on your Red Hat OpenShift Container Platform cluster:</p> <pre>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</pre> <p>To set this environment variable, run the following command.</p> <pre>export FILE_SYSTEM_NAME=<file-system-name></pre> <p>Replace <file-system-name> with the appropriate value for your environment.</p>

Variable	Notes
<code>REMOTE_STORAGE_CLUSTER_ID</code>	<p>Specify the ID of the remote storage cluster.</p> <p>IBM Spectrum Fusion Run the following command to get the cluster ID:</p> <pre>oc get daemons ibm-spectrum-scale -n ibm-spectrum-scale \ -o jsonpath='{.status.clusterID}'</pre> <p>IBM Spectrum Scale Container Native Run the following command on the remote IBM Spectrum Scale cluster to get the cluster ID:</p> <pre>mmlscluster</pre> <p>To set this environment variable, run the following command.</p> <pre>export REMOTE_STORAGE_CLUSTER_ID=<cluster-id></pre> <p>Replace <code><cluster-id></code> with the appropriate value for your environment.</p>

3. Create the storage class.

The following sample includes the minimum required information for the storage class. Review [Storage Class](#) in the IBM Spectrum Scale Container Native documentation for additional options.

```
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: "${FILE_SYSTEM_NAME}"
  clusterId: "${REMOTE_STORAGE_CLUSTER_ID}"
  permissions: "777" # The permissions must be set to 777. Do not change this setting.
reclaimPolicy: Delete
EOF
```

What to do next

Now that you've created the storage class, you are ready to complete [Setting up projects \(namespaces\) on Red Hat OpenShift Container Platform](#).


Creating Portworx storage classes

If you decide to use Portworx as your storage option, Cloud Pak for Data recommends several storage classes for optimized performance and availability.

Installation phase

- Setting up a client workstation

- Collecting required information

-  Preparing your cluster

- Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

A cluster administrator or a storage administrator must complete this task.

When do you need to complete this task?

If you plan to use Portworx storage, you should create the recommended storage classes. You must complete this task before you install IBM® Cloud Pak for Data.

You can optionally use your own block storage and file storage classes instead of creating the recommended storage classes. However, this might result in decreased performance.

Before you begin

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.

Block storage

The following storage classes use the ReadWriteOnce (RWO) access mode.

portworx-couchdb-sc

Used for: Block storage with 3 replicas for components, such as common core services, that store data in CouchDB.

Storage class definition:

```
# 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
```

portworx-elastic-sc

Used for: Block storage with 2 replicas for components, such as common core services, that store data in Elasticsearch.

Storage class definition:

```
# 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-metastoredb-sc

Used for: Block storage with 3 replicas for components, such as common core services and the Cloud Pak for Data control plane, that need to store metadata.

Storage class definition:

```
# 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
```

portworx-gp3-sc

Used for: Block storage with 3 replicas, used by the common core services for message brokers, such as RabbitMQ and Redis HA.

Storage class definition:

```
# 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-db-gp

Used for: Block storage with 1 replica for databases.

Storage class definition:

```
# 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

Used for: Block storage with 2 replicas for databases.

Storage class definition:

```
# 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

Used for: Block storage with 3 replicas for databases.

Storage class definition:

```
# 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-rwo-sc

Used for: Block storage with 3 replicas for user data storage for Db2® databases and the Watson™ Knowledge Catalog Db2 metastore.

Storage class definition:

```
# 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-watson-assistant-sc

Used for: Block storage with 3 replicas. Optimized for Watson Assistant.

Storage class definition:

```
# 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

Used for: Block storage with 3 replicas. Optimized for Informix®.

Storage class definition:

```

# 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

```

File storage

The following storage classes use the ReadWriteMany (RWX) access mode.

portworx-rwx-gp3-sc

Used for: General purpose file storage class with 3 replicas, used by components that don't require more specific storage.

Storage class definition:

```

# 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-db2-rwx-sc

Used for: File storage class with 3 replicas for system and backup data for Db2 databases.

Storage class definition:

```

# 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

```

Deprecated storage classes - block storage

Important: Create the storage classes in this section only if you have an existing IBM Cloud Pak for Data and you need to migrate the installation to a new cluster. These storage classes should not be used for new installations of IBM Cloud Pak for Data Version 4.5.

The following storage classes use the ReadWriteOnce (RWO) access mode.

portworx-nonshared-gp2

Used for: Block storage with 3 replicas, used by the common core services for message brokers, such as RabbitMQ and Redis HA. (Applicable only for upgrades where this storage class is already in use.)

Access mode: ReadWriteOnce (RWO)

Storage class definition:

```
# 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
```

portworx-solr-sc

Used for: Block storage for components that store data in Solr.

Access mode: ReadWriteOnce (RWO)

Storage class definition:

```
# 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

Used for: Block storage for components that store data in Cassandra.

Access mode: ReadWriteOnce (RWO)

Storage class definition:

```
# 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

Used for: Block storage for components that store data in Kafka.

Access mode: ReadWriteOnce (RWO)

Storage class definition:

```
# 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-db2-sc

Used for: Block storage with 3 replicas for the Watson Knowledge Catalog Db2 metastore (Applicable only for upgrades where this storage class is already in use.)

Access mode: ReadWriteOnce (RWO)

Storage class definition:

```
# 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
```

Deprecated storage classes - file storage

Important: Create the storage classes in this section only if you have an existing IBM Cloud Pak for Data and you need to migrate the installation to a new cluster. These storage classes should not be used for new installations of IBM Cloud Pak for Data Version 4.5.

The following storage classes use the ReadWriteMany (RWX) access mode.

portworx-shared-gp3

Used for: General purpose file storage with 3 replicas, used by components that don't require more specific storage. Previous name for `portworx-rwx-gp3-sc`. (Applicable only for upgrades where this storage class is already in use.)

Storage class definition:

```
# 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

Used for: General purpose file storage with 2 replicas, used by components that don't require more specific storage. (Applicable only for upgrades where this storage class is already in use.)

Storage class definition:

```

# 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

Used for: File storage class with a single replica, used by the Data Virtualization service. (Applicable only for upgrades where this storage class is already in use.)

Storage class definition:

```

# 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

Used for: File storage class with 3 replicas, used by the Data Virtualization service. (Applicable only for upgrades where this storage class is already in use.)

Storage class definition:

```

# 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

```

portworx-shared-gp

Used for: File storage with 3 replicas and high IOPS. (Applicable only for upgrades where this storage class is already in use.)

Storage class definition:

```

# 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-shared-gp1

Used for: File storage with 1 replica and high IOPS. (Applicable only for upgrades where this storage class is already in use.)

Access mode: ReadWriteMany (RWX)

Storage class definition:

```

#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

```

Setting up NFS storage

If you plan to use NFS for persistent storage, you must set up your NFS storage before you install Cloud Pak for Data.

It is highly recommended that your NFS storage is served by an enterprise class storage system that ensures availability with a sufficiently high throughput and reduced latency. It is recommended that the storage is in close proximity to the cluster. For example, the storage should be in the same network as the cluster. For details, see [Hardware requirements](#).

Ensure that the following statements are true on all the nodes you plan to run Cloud Pak for Data on:

- All of the nodes in the cluster have access to mount the NFS server.
- All of the nodes in the cluster have read/write access to the NFS server.
- Containerized processes have read/write access to the NFS server.

Important: Containerized processes create files that are owned by various UIDs. (In Cloud Pak for Data, most services use long UIDs between 1000320900 and 1000361000, as an example. UID range is decided by OpenShift® and project namespaces.) Db2® reserves a specific UID range because it uses a custom security context constraint (SCC). For more information, see [Basic security features](#). If you restrict access to the NFS served to specific UIDs, you might encounter errors when installing or running Cloud Pak for Data.

Before you begin

Determine which services you will install on IBM® Cloud Pak for Data. At a minimum, you must configure dynamic provisioning of NFS storage. However, if you plan to install any of the following services, you must also configure the NFS export options on your NFS server:

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

Use the following information to ensure that you complete the appropriate tasks for setting up your NFS storage:

Options	What to do
I am installing one of the services on Cloud Pak for Data that requires specific NFS export options.	<ol style="list-style-type: none">1. Configuring your NFS server2. Setting up dynamic provisioning
I am not installing any of the services on Cloud Pak for Data that require specific NFS export options.	<ol style="list-style-type: none">1. Setting up dynamic provisioning

- [Configuring your NFS server](#)
If you are installing services that depend on Db2 on Cloud Pak for Data, you must configure the NFS export options and enable `no_root_squash` before you set up dynamic provisioning.
- [Setting up dynamic provisioning](#)
NFS does not support dynamic storage provisioning by default, and Red Hat OpenShift does not include a provisioner plug-in to create an NFS storage class. Therefore, you must set up dynamic storage provisioning on your NFS server.

Configuring your NFS server

If you are installing services that depend on Db2® on Cloud Pak for Data, you must configure the NFS export options and enable `no_root_squash` before you set up dynamic provisioning.

About this task

The following services have specific configuration requirements when setting up your NFS server:

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

Enabling `no_root_squash` allows root clients to retain root permissions on the remote NFS share. If you do not configure `no_root_squash` before installing your services, the Db2u instance will fail.

A Network Attached Storage (NAS) or Network File System (NFS) share must be exported to all NFS clients by using the following export options: `rw`, `sync`, `no_root_squash`, `no_all_squash`.

You must export the NFS mount to each of the Red Hat® OpenShift® worker nodes that host your service.

Procedure

1. Run a command similar to the following example:

```
echo "/nfsmount db2_openshiftworker1_IP_address(rw,sync,no_root_squash,no_all_squash)
db2_openshiftworker2_IP_address(rw,sync,no_root_squash,no_all_squash)" > /etc/exports
```

Where `db2_openshiftworker_IP_address` is repeated for each worker node.

2. Run the `exportfs -a` or `exportfs -r` command if a NAS controller is used to ensure that the newly added NFS shares are broadcast to the defined client list.
3. Add `fstab` content into a new text file.

Note: In following example, replace `hostname`. If there is already some content in `/etc/fstab` on the worker nodes, then copy the content into a text file since the content in `/etc/fstab` will overwrite the existing file.

```
cat << EOF > text_file
hostname_OR_IP_address_of_NAS/NFS_server:NAS/NFS_share_exported /nfsmount/ nfs
rw,relatime,vers=3,rsize=1048576,wsiz=1048576,namlen=255,hard,intr,proto=tcp,port=0,time
o=600,retrans=2,sec=sys,nolock 0 0
EOF
```

4. Create the `machineconfig` object YAML file.

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 create -f -
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 99-configure-fstab
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
      - contents:
          source: data:text/plain;charset=utf-8;base64,$(cat text_file | base64 -w0)
          path: /etc/fstab
          overwrite: true
EOF
```

Note: On Mac OS systems, remove `-w0` at the end of the `source` value so that you do not receive an error when you create the `machineconfig` object YAML file.

```
cat << EOF | oc create -f -
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 99-configure-fstab
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
      - contents:
          source: data:text/plain;charset=utf-8;base64,$(cat text_file | base64)
          path: /etc/fstab
```

```
    overwrite: true
EOF
```


Setting up dynamic provisioning

NFS does not support dynamic storage provisioning by default, and Red Hat® OpenShift® does not include a provisioner plugin to create an NFS storage class. Therefore, you must set up dynamic storage provisioning on your NFS server.

Installation phase

Setting up a client workstation

Collecting required information

 Preparing your cluster

Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

A cluster administrator must complete this task.

When do you need to complete this task?

If you plan to use NFS storage, you must set up dynamic provisioning before you install Cloud Pak for Data.

Before you begin

Best practice: You can run many of the commands in this task exactly as written if you set up environment variables for your installation. For instructions, see [Setting up installation environment variables](#).

Ensure that you source the environment variables before you run the commands in this task.

If you are installing any of the following services on Cloud Pak for Data, ensure [your NFS server is configured](#) before you set up dynamic provisioning:

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

About this task

The steps in this procedure use the [Kubernetes NFS-Client Provisioner](#) (from the [Kubernetes SIGs](#) organization) to set up dynamic provisioning with NFS storage.


Important: You must have an existing NFS server to complete this task. Ensure that you know how to connect to your NFS server. At a minimum, you must have the hostname of the server.

Your NFS server must be accessible from your Red Hat OpenShift Container Platform cluster.

Mirroring the provisioner images to a private container registry

If you pull images from a private container registry, mirror the images for the Kubernetes NFS-Client Provisioner to your private container registry. Complete the appropriate task for your environment:

- If your client workstation can connect to the internet and to the private container registry, you can [mirror the images directly to your private container registry](#).
- If your client workstation cannot connect to the internet and to the private container registry, you must [mirror images to an intermediary container registry before you can mirror the images to your private container registry](#).

-  Mirroring the provisioner images directly to a private container registry

To mirror the images for the Kubernetes NFS-Client Provisioner to your private container registry:

1. Log in to your private container registry:

```
cpd-cli manage login-private-registry \  
${PRIVATE_REGISTRY_LOCATION} \  
${PRIVATE_REGISTRY_PUSH_USER} \  
${PRIVATE_REGISTRY_PUSH_PASSWORD}
```

If your private registry is not secured, see [cpd-cli manage login-private-registry](#) for additional options.

2. Mirror the images to the private container registry:

```
cpd-cli manage mirror-nfs-provisioner \  
--target_registry=${PRIVATE_REGISTRY_LOCATION} \  
--source_registry=k8s.gcr.io/sig-storage
```

- > Mirroring the provisioner images using an intermediary container registry

To mirror the images for the Kubernetes NFS-Client Provisioner to your private container registry:

1. Mirror the images to the intermediary container registry:

```
cpd-cli manage mirror-nfs-provisioner \  
--target_registry=127.0.0.1:12443 \  
--source_registry=k8s.gcr.io/sig-storage
```

2. Move the intermediary container registry behind the firewall.
3. Log in to your private container registry:

```
cpd-cli manage login-private-registry \  
${PRIVATE_REGISTRY_LOCATION} \  
${PRIVATE_REGISTRY_PUSH_USER} \  
${PRIVATE_REGISTRY_PUSH_PASSWORD}
```

If your private registry is not secured, see [cpd-cli manage login-private-registry](#) for additional options.

4. Mirror the images to the private container registry:

```
cpd-cli manage mirror-nfs-provisioner \  
--target_registry=${PRIVATE_REGISTRY_LOCATION} \  
--source_registry=127.0.0.1:12443
```

Configuring dynamic storage

To configure dynamic storage:

1. Run the `cpd-cli manage`

`login-to-ocp` command to log in to the cluster as a user with sufficient permissions to complete this task. For example:

```
cpd-cli manage login-to-ocp \  
--username=${OCP_USERNAME} \  
--password=${OCP_PASSWORD} \  
--server=${OCP_URL}
```

Tip: The `login-to-ocp` command takes the same input as the `oc login` command. Run `oc login --help` for details.

2. If you mirrored the images to a private container registry, update the global image pull secret so that the cluster can access the Kubernetes NFS-Client Provisioner images.
The global image pull secret must contain the credentials of an account that can *pull* images from the private container registry:

```
cpd-cli manage add-cred-to-global-pull-secret \  
${PRIVATE_REGISTRY_LOCATION} \  

```

```
 ${PRIVATE_REGISTRY_PULL_USER} \
 ${PRIVATE_REGISTRY_PULL_PASSWORD}
```

3. Set the following environment variables:

a. Set `NFS_SERVER_LOCATION` to the IP address or fully qualified domain name (FQDN) of the NFS server:

```
export NFS_SERVER_LOCATION=<server-address>
```

b. Set `NFS_PATH` to the exported path where you want the provisioner to create sub-directories. (The default path is `/`.)

```
export NFS_PATH=<path>
```

c. Set `PROJECT_NFS_PROVISIONER` to the project (namespace) where you want to deploy the Kubernetes NFS-Client Provisioner provisioner. The recommended project is `nfs-provisioner`; however you can specify any project.

Important: You must specify an existing project (namespace).

```
export PROJECT_NFS_PROVISIONER=<project-name>
```

d. Set `NFS_STORAGE_CLASS` to the name that you want to use for the NFS storage class:

```
export NFS_STORAGE_CLASS=<storage-class-name>
```

By default, the documentation uses `managed-nfs-storage`, but you can pick a different storage class name.

e. Set the `NFS_IMAGE` to the correct value for your Red Hat OpenShift Container Platform architecture:

Architecture	Command
x86-64	<code>export NFS_IMAGE=k8s.gcr.io/sig-storage/nfs-subdir-external-provisioner:v4.0.2</code>
ppc64le	<code>export NFS_IMAGE=gcr.io/k8s-staging-sig-storage/nfs-subdir-external-provisioner:v4.0.2</code>
s390x	<code>export NFS_IMAGE=gcr.io/k8s-staging-sig-storage/nfs-subdir-external-provisioner:v4.0.2</code>

4. Run the following command to set up dynamic provisioning:

```
cpd-cli manage setup-nfs-provisioner \
--nfs_server=${NFS_SERVER_LOCATION} \
--nfs_path=${NFS_PATH} \
--nfs_provisioner_ns=${PROJECT_NFS_PROVISIONER} \
--nfs_storageclass_name=${NFS_STORAGE_CLASS} \
--nfs_provisioner_image=${NFS_IMAGE}
```

If the command succeeds, the storage class is ready to use.

5. If you are using any of the following services, add the `mountOptions` entry to the storage class:

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

Run the following command to update the storage class:

```
cat <<EOF |oc apply -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ${NFS_STORAGE_CLASS}
provisioner: nfs-client-provisioner
parameters:
  archiveOnDelete: "false"
mountOptions:
- v4.2
- context="system_u:object_r:container_file_t:s0"
EOF
```


Setting up Amazon Elastic File System

Amazon Elastic File System (EFS) does not support dynamic storage provisioning by default, and Red Hat® OpenShift® does not include a provisioner plug-in to create an NFS-based storage class. Therefore, you must set up dynamic storage provisioning on your Amazon Elastic File System.

Installation phase

Setting up a client workstation

Collecting required information

 Preparing your cluster

Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

A cluster administrator or a storage administrator must complete this task.

When do you need to complete this task?

If you plan to use EFS storage, you must set up dynamic provisioning before you install Cloud Pak for Data.

If you used a Terraform® script from <https://github.com/IBM/cp4d-deployment> to set up your Red Hat OpenShift Container Platform cluster on Amazon Web Services, the Terraform automatically set up EFS for you, and you can skip this task.

Before you begin

Best practice: You can run many of the commands in this task exactly as written if you set up environment variables for your installation. For instructions, see [Setting up installation environment variables](#).

Ensure that you source the environment variables before you run the commands in this task.

About this task

The steps in this procedure use the [Kubernetes NFS-Client Provisioner](#) (from the [Kubernetes SIGs](#) organization) to set up dynamic provisioning with EFS storage.

Important: You must have an existing Amazon Elastic File System to complete this task.


Your Amazon Elastic File System must be accessible from your Red Hat OpenShift Container Platform cluster. To ensure that EFS is accessible from the cluster:

- Ensure that you create the file system in the same VPC as the Red Hat OpenShift Container Platform cluster.
- Select the security group attached to the worker nodes and private subnet.
- Add an NFS inbound rule to the security group to allow ingress from the worker nodes to the file system.

Mirroring the provisioner images to a private container registry

If you pull images from a private container registry, mirror the images for the Kubernetes NFS-Client Provisioner to your private container registry. Complete the appropriate task for your environment:

- If your client workstation can connect to the internet and to the private container registry, you can [mirror the images directly to your private container registry](#).
- If your client workstation cannot connect to the internet and to the private container registry, you must [mirror images to an intermediary container registry before you can mirror the images to your private container registry](#).

-  [Mirroring the provisioner images directly to a private container registry](#)

To mirror the images for the Kubernetes NFS-Client Provisioner to your private container registry:

1. Log in to your private container registry:

```
cpd-cli manage login-private-registry \  
${PRIVATE_REGISTRY_LOCATION} \  

```



```
 ${PRIVATE_REGISTRY_PUSH_USER} \
 ${PRIVATE_REGISTRY_PUSH_PASSWORD}
```

If your private registry is not secured, see [cpd-cli manage login-private-registry](#) for additional options.

2. Mirror the images to the private container registry:

```
cpd-cli manage mirror-nfs-provisioner \
--target_registry=${PRIVATE_REGISTRY_LOCATION} \
--source_registry=k8s.gcr.io/sig-storage
```

- [Mirroring the provisioner images using an intermediary container registry](#)

To mirror the images for the Kubernetes NFS-Client Provisioner to your private container registry:

1. Mirror the images to the intermediary container registry:

```
cpd-cli manage mirror-nfs-provisioner \
--target_registry=127.0.0.1:12443 \
--source_registry=k8s.gcr.io/sig-storage
```

2. Move the intermediary container registry behind the firewall.
3. Log in to your private container registry:

```
cpd-cli manage login-private-registry \
${PRIVATE_REGISTRY_LOCATION} \
${PRIVATE_REGISTRY_PUSH_USER} \
${PRIVATE_REGISTRY_PUSH_PASSWORD}
```

If your private registry is not secured, see [cpd-cli manage login-private-registry](#) for additional options.

4. Mirror the images to the private container registry:

```
cpd-cli manage mirror-nfs-provisioner \
--target_registry=${PRIVATE_REGISTRY_LOCATION} \
--source_registry=127.0.0.1:12443
```

Getting the connection details for your Amazon Elastic File System

Before you can set up dynamic provisioning, you must obtain the DNS name or IP address of your Amazon Elastic File System:

DNS name (recommended)

You can obtain the DNS name from the AWS Console on the Amazon EFS File systems. Select the file system that you want to use. The DNS name is in the General section.

The DNS name has the following format: `<file-storage-id>.efs.<region>.amazonaws.com`.

IP address

You can obtain the IP address from the AWS Console on the Amazon EFS File systems. Select the file system that you want to use. The IP address is on the Network tab.

Configuring dynamic storage

To configure dynamic storage:

1. Run the `cpd-cli manage login-to-ocp` command to log in to the cluster as a user with sufficient permissions to complete this task. For example:

```
cpd-cli manage login-to-ocp \
--username=${OCP_USERNAME} \
```

```
--password=${OCP_PASSWORD} \  
--server=${OCP_URL}
```

Tip: The `login-to-ocp` command takes the same input as the `oc login` command. Run `oc login --help` for details.

2. If you mirrored the images to a private container registry, update the global image pull secret so that the cluster can access the Kubernetes NFS-Client Provisioner images.

The global image pull secret must contain the credentials of an account that can *pull* images from the private container registry:

```
cpd-cli manage add-cred-to-global-pull-secret \  
${PRIVATE_REGISTRY_LOCATION} \  
${PRIVATE_REGISTRY_PULL_USER} \  
${PRIVATE_REGISTRY_PULL_PASSWORD}
```

3. Set the following environment variables:
 - a. Set `EFS_LOCATION` to the DNS name or IP address EFS server:

```
export EFS_LOCATION=<location>
```

- b. Set `EFS_PATH` to the EFS exported path. (The default path is `/`.)

```
export EFS_PATH=/  

```

- c. Set `PROJECT_NFS_PROVISIONER` to the project (namespace) where you want to deploy the Kubernetes NFS-Client Provisioner provisioner. The recommended project is `nfs-provisioner`; however you can specify any project.

Important: You must specify an existing project (namespace).

```
export PROJECT_NFS_PROVISIONER=<project-name>
```

- d. Set `EFS_STORAGE_CLASS` to the name that you want to use for the EFS storage class. The recommended name is `efs-nfs-client`.

```
export EFS_STORAGE_CLASS=efs-nfs-client
```

- e. Set the `NFS_IMAGE` to the correct value for your Red Hat OpenShift Container Platform architecture:

Architecture	Command
x86-64	<code>export NFS_IMAGE=k8s.gcr.io/sig-storage/nfs-subdir-external-provisioner:v4.0.2</code>
ppc64le	<code>export NFS_IMAGE=gcr.io/k8s-staging-sig-storage/nfs-subdir-external-provisioner:v4.0.2</code>
s390x	<code>export NFS_IMAGE=gcr.io/k8s-staging-sig-storage/nfs-subdir-external-provisioner:v4.0.2</code>

4. Run the following command to set up dynamic provisioning:

```
cpd-cli manage setup-nfs-provisioner \  
--nfs_server=${EFS_LOCATION} \  
--nfs_path=${EFS_PATH} \  
--nfs_provisioner_ns=${PROJECT_NFS_PROVISIONER} \  
--nfs_storageclass_name=${EFS_STORAGE_CLASS} \  
--nfs_provisioner_image=${NFS_IMAGE}
```

Setting up IBM Cloud File Storage

If you are installing services that depend on NFS and you are planning to use IBM Cloud File Storage on NFS 4 for persistent storage, you must configure ID mapping, which enables `no_root_squash`. Configuring `no_root_squash` allows root clients to retain root permissions on the remote NFS share.

About this task

If you are installing any of the following services on Cloud Pak for Data, you must configure ID mapping:

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

You can configure ID mapping through a daemon set or by running manual commands on worker nodes. These steps also enable `no_root_squash` in the IBM Cloud environment. For more details, see [Implementing no_root_squash for NFS](#) in the IBM Cloud documentation.

- **Configuring ID mapping through a daemon set**

1. Create a service account called `norootsquash` by running the following command:

```
oc create -f - << EOF
apiVersion: v1
kind: ServiceAccount
metadata:
  name: norootsquash
  namespace: kube-system
EOF
```

2. Give the service account privileged security context constraints (SCC) by running the following command:

```
oc adm policy add-scc-to-user privileged system:serviceaccount:kube-
system:norootsquash
```

3. Create the daemon set by running the following command:

```
export DOMAIN_NAME=<>

oc create -f - << EOF
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: norootsquash
  namespace: kube-system
  labels:
    tier: management
    app: norootsquash
spec:
  selector:
    matchLabels:
      name: norootsquash
  template:
    metadata:
      labels:
        name: norootsquash
    spec:
      serviceAccountName: norootsquash
      initContainers:
        - resources:
            requests:
              cpu: 0.1
          securityContext:
            privileged: true
          image: alpine:3.6
          name: unrootsquash
          command: ["chroot", "/host", "sh", "-c"]
          args:
            - >
              grep "^Domain = ${DOMAIN_NAME}" /etc/idmapd.conf;
              if [ "\${?}" -ne "0" ] ; then
                sed -i 's/.*/Domain = ${DOMAIN_NAME}/g' /etc/idmapd.conf;
                nfsidmap -c;
                rpc.idmapd
              fi;
      volumeMounts:
```

```

- name: host
  mountPath: /host
  containers:
  - resources:
    requests:
      cpu: 0.1
    image: alpine:3.6
    name: sleep
    command: ["/bin/sh", "-c"]
    args:
      - >
        while true; do
          sleep 100000;
        done
  volumes:
  - hostPath:
    path: /
    type: Directory
    name: host
EOF

```

- > Configuring ID mapping by running manual commands on worker nodes
 - Run the following command to perform the same task as the daemonset. The command takes about 30 seconds per node. Note that these settings do not apply to new worker nodes, so you must add them.

```

oc get no -l node-role.kubernetes.io/worker --no-headers -o name | xargs -I {} --
oc debug {} -- chroot /host sh -c 'grep "^Domain = ${DOMAIN_NAME}" /etc/idmapd.conf
|| ( sed -i "s/.*Domain = ./Domain = slnfsv4.com/g" /etc/idmapd.conf; nfsidmap -c;
rpc.idmapd )'


```

Note: The `DOMAIN_NAME` for `ibm-cloud-file-storage` would be `slnfsv4.com`

Setting up projects (namespaces) on Red Hat OpenShift Container Platform

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

Installation phase

- Setting up a client workstation
- Collecting required information
-  Preparing your cluster
- Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

A cluster administrator, or a user with permissions to create projects, must complete this task.

When do you need to complete this task?

You must complete this task before you install Cloud Pak for Data for the first time.

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.

Before you begin

Review the guidance in [Supported project \(namespace\) configurations](#) to understand the relationship between the projects (namespaces) and the security considerations that you need to take into account.

Best practice: You can run the commands in this task exactly as written if you set up environment variables. For instructions, see [Setting up installation environment variables](#).

Ensure that you source the environment variables before you run the commands in this task.

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 ${OCP_URL}
```

2. Create the appropriate projects for your environment.

- | >
Express installations | |
|----------------------------|--|
| Project to create | Command |
| ibm-common-services | <p>You can skip this step if IBM Cloud Pak® foundational services is already installed on the cluster.</p> <p>ibm-common-services is the default and recommended name. If you sourced the installation environment variables, the value from the PROJECT_CPD_INSTANCE variable is used.</p> <p>If you choose to use a different project for the IBM Cloud Pak foundational services operators, you must create configmap. For details, see Installing IBM Cloud Pak foundational services in a custom namespace.</p> <pre>oc new-project \${PROJECT_CPFS_OPS}</pre> |
| cpd-instance | <p>Source the installation environment variables to ensure that the correct value for the PROJECT_CPD_INSTANCE variable is used.</p> <pre>oc new-project \${PROJECT_CPD_INSTANCE}</pre> <p>Remember: If you plan to install multiple instances of Cloud Pak for Data on the cluster, you must create a project for each instance.</p> |
| cpd-instance-tether | <p>You can skip this step if you don't plan to deploy any service instances or workloads in a tethered project.</p> <p>Source the installation environment variables to ensure that the correct value for the PROJECT_TETHERED variable is used.</p> <pre>oc new-project \${PROJECT_TETHERED}</pre> <p>Important:
Many services support only one service instance in a given project. So if you want to create multiple instances of a service, you must deploy each instance of the service in a different project. You can achieve this by creating multiple tethered projects and creating one instance of the service in each tethered project.</p> <p>You can co-locate service instances and workloads for different services in the same tethered project, or you can create different tethered projects if one service requires or workload requires more privileges. You can use different tethered projects to give each service instance or workload the exist privileges it needs to align with the <i>Principle of Least Privileges</i>.</p> |
- | >
Specialized installations | |
|--------------------------------|---------|
| Project to create | Command |
| | |

Project to create	Command
<code>ibm-common-services</code>	<p>You can skip this step if IBM Cloud Pak foundational services is already installed on the cluster.</p> <p><code>ibm-common-services</code> is the default and recommended name. If you sourced the installation environment variables, the value from the <code>PROJECT_CPD_INSTANCE</code> variable is used.</p> <p>If you choose to use a different project for the IBM Cloud Pak foundational services operators, you must create <code>configmap</code>. For details, see Installing IBM Cloud Pak foundational services in a custom namespace.</p> <pre>oc new-project \${PROJECT_CPFS_OPS}</pre>
<code>cpd-operators</code>	<p><code>cpd-operators</code> is the recommended name. If you sourced the installation environment variables, the value from the <code>PROJECT_CPD_INSTANCE</code> variable is used.</p> <pre>oc new-project \${PROJECT_CPD_OPS}</pre>
<code>cpd-instance</code>	<p>Source the installation environment variables to ensure that the correct value for the <code>PROJECT_CPD_INSTANCE</code> variable is used.</p> <pre>oc new-project \${PROJECT_CPD_INSTANCE}</pre> <p>Remember: If you plan to install multiple instances of Cloud Pak for Data on the cluster, you must create a project for each instance.</p>
<code>cpd-instance-tether</code>	<p>You can skip this step if you don't plan to deploy any service instances or workloads in a tethered project.</p> <p>Source the installation environment variables to ensure that the correct value for the <code>PROJECT_TETHERED</code> variable is used.</p> <pre>oc new-project \${PROJECT_TETHERED}</pre> <p>Important: Many services support only one service instance in a given project. So if you want to create multiple instances of a service, you must deploy each instance of the service in a different project. You can achieve this by creating multiple tethered projects and creating one instance of the service in each tethered project.</p> <p>You can co-locate service instances and workloads for different services in the same tethered project, or you can create different tethered projects if one service requires or workload requires more privileges. You can use different tethered projects to give each service instance or workload the exist privileges it needs to align with the <i>Principle of Least Privileges</i>.</p>

3. Red Hat OpenShift Container Platform Version 4.6 only. If you plan to install one of the following services, you must label the project where their operators are installed:

- IBM Match 360 with Watson™
- Watson Knowledge Catalog

To label the project, run the following command:

```
oc label namespace ${PROJECT_CPD_OPS} kubernetes.io/metadata.name=${PROJECT_CPD_OPS}
```

4. If you created a tethered project, you must tether `${PROJECT_TETHERED}` to the project where the Cloud Pak for Data control plane is installed (`${PROJECT_CPD_INSTANCE}`):

a. Run the `cpd-cli manage`

`login-to-ocp` command to log in to the cluster as a user with sufficient permissions to complete this task. For example:

```
cpd-cli manage login-to-ocp \
--username=${OCP_USERNAME} \
--password=${OCP_PASSWORD} \
--server=${OCP_URL}
```

Tip: The `login-to-ocp` command takes the same input as the `oc login` command. Run `oc login --help` for details.

- b. Tether the `${PROJECT_TETHERED}` to the project where the Cloud Pak for Data control plane is installed:

```
cpd-cli manage setup-tethered-ns \  
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \  
--tethered_instance_ns=${PROJECT_TETHERED}
```

If you created multiple tethered projects, export the `${PROJECT_TETHERED}` environment variable with the correct project name and repeat this step to tether each project to the `${PROJECT_CPD_INSTANCE}` project.

After you tether a project to the project where the Cloud Pak for Data control plane is installed, you can deploy service instances to the tethered project or run workloads in the tethered project. For information about which services support this, see [Multitenancy support](#).

What to do next

Now that you've set up the required projects, you are ready to complete [Creating custom security context constraints for services](#).

Previous topic: [Setting up persistent storage](#)

Next topic: [Creating custom security context constraints for services](#)


Creating custom security context constraints for services

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

Installation phase

Setting up a client workstation

Collecting required information

 Preparing your cluster

Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

A cluster administrator must complete this task.

When do you need to complete this task?

You must complete this task before you install a service that uses a custom SCC.

The **restricted** SCC

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 an administrator. By default, containers are granted access to the **restricted** SCC and have only the capabilities that are defined by the **restricted** SCC. For more information, see *Managing security context constraints* in the Red Hat OpenShift Container Platform documentation:

- [Version 4.6](#)
- [Version 4.8](#)
- [Version 4.10](#)

When you install Cloud Pak for Data, the default service account is associated with the **restricted** SCC. Cloud Pak for Data does not support the use of privileged SCCs in OpenShift.

Most Cloud Pak for Data services use the **restricted** SCC.

SCCs for IBM Cloud Pak foundational services

For information about the SCCs that are required by the IBM Cloud Pak® foundational services, see [Security context constraints](#) in the IBM Cloud Pak foundational services documentation.

Custom SCCs

If you plan to install any of the following Cloud Pak for Data services, you must create the appropriate custom SCCs:

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

Service	Required SCCs
Data Virtualization	Data Virtualization embeds an instance of Db2, which requires a custom SCC. This SCC is used only by the instance of Data Virtualization that embeds the Db2 database. For details, see Creating the custom security context constraint for embedded Db2 databases .
Db2	Db2 requires a custom SCC. By default, the SCC is created automatically; however, you can choose to create the SCC manually. For details, see Creating the custom security context constraint for Db2 .
Db2 Big SQL	Db2 Big SQL embeds an instance of Db2, which requires a custom SCC. This SCC is used only by the instance of Db2 Big SQL that embeds the Db2 database. For details, see Creating the custom security context constraint for embedded Db2 databases .
Db2 Warehouse	Db2 Warehouse requires a custom SCC. By default, the SCC is created automatically; however, you can choose to create the SCC manually. For details, see Creating the custom security context constraint for Db2 Warehouse .
OpenPages	The OpenPages service can optionally embed an instance of Db2. If you chose to use an embedded instance of Db2, OpenPages requires a custom SCC for the Db2 database. This SCC is used only by the instance of OpenPages that embeds the Db2 database. For details, see Creating the custom security context constraint for embedded Db2 databases . If you choose to use an external database, the custom SCC is not required.
Watson Knowledge Catalog	Watson Knowledge Catalog requires two custom SCCs: <ul style="list-style-type: none"> • An SCC for Watson Knowledge Catalog. You must create this SCC manually. For details, see Creating the custom security context constraint for Watson Knowledge Catalog. • An SCC for the instance of Db2 that is embedded in Watson Knowledge Catalog. This SCC is used only by the instance of Watson Knowledge Catalog that embeds the Db2 database. For details, see Creating the custom security context constraint for embedded Db2 databases. <p>If you install Data Privacy, the service uses the Watson Knowledge Catalog SCC.</p>

- [Creating the custom security context constraint for Watson Knowledge Catalog](#)
The Watson Knowledge Catalog service requires a custom security context constraint (SCC).
- [Creating the custom security context constraint for Db2](#)
The Db2 service requires a custom security context constraint (SCC).
- [Creating the custom security context constraint for Db2 Warehouse](#)
The Db2 Warehouse service requires a custom security context constraint (SCC).
- [Creating the custom security context constraint for embedded Db2 databases](#)
Embedded Db2 databases require a custom security context constraint (SCC).

Previous topic: [Setting up projects \(namespaces\) on Red Hat OpenShift Container Platform](#)

Next topic: [Changing required node settings](#)

Creating the custom security context constraint for Watson Knowledge Catalog

The Watson Knowledge Catalog service requires a custom security context constraint (SCC).

If you plan to install the Watson Knowledge Catalog service, you must create the `wkc-iis-scc` security context constraint.

About this task

The Watson Knowledge Catalog SCC is created once and used by each instance of Watson Knowledge Catalog that you install.

Run the `cpd-cli manage`

`apply-scc` command to bind the SCC to the `wkc-iis-sa` service account in the projects where you plan to install Watson Knowledge Catalog. For example, if you plan to install Watson Knowledge Catalog in two projects, you must run the command twice to bind the SCC to the service account in each project.

- [Watson Knowledge Catalog SCC definition](#)

```
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
```

Procedure

To create the `wkc-iis-scc` SCC:

1. Run the `cpd-cli manage login-to-ocp` command to log in to the cluster as a user with sufficient permissions to complete this task. For example:

```
cpd-cli manage login-to-ocp \  
--username=${OCP_USERNAME} \  
--password=${OCP_PASSWORD} \  
--server=${OCP_URL}
```

Tip: The `login-to-ocp` command takes the same input as the `oc login` command. Run `oc login --help` for details.

2. Run the following command to create the SCC:

```
cpd-cli manage apply-scc \  
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \  
--components=wkc
```

Tip: If you want to preview the `oc` commands that the `cpd-cli` will issue on your behalf, you can run the command with `--preview=true`.

The `oc` commands are saved to the `preview.sh` file in the `cpd-cli-workspace/olm-utils-workspace/work` directory.

3. (Optional) If you want to isolate the containers that use the `wkc-iis-scc` SCC on specific worker nodes:
 - a. Identify the nodes where you want to run the containers.
A single node should be sufficient, but you can optionally allocate additional nodes for high availability in case of a node failure.
 - b. For each node that you identify, run the following command:

```
oc label node <NODE_NAME> wkc-privileged=wkc-privileged-node
```

Replace `<NODE_NAME>` with the appropriate value for your cluster

Results

The `wkc-iis-scc` SCC is created if it doesn't exist and is bound to the `wkc-iis-sa` service account in the `${PROJECT_CPD_INSTANCE}` project.

If you want to confirm that the `wkc-iis-sa` service account can use the `wkc-iis-scc` SCC, run:

```
oc adm policy who-can use scc wkc-iis-scc \  
--namespace ${PROJECT_CPD_INSTANCE} | grep "wkc-iis-sa"
```

Creating the custom security context constraint for Db2

The Db2 service requires a custom security context constraint (SCC).

Default SCC

When you create a Db2 instance, an SCC named `<NAMESPACE>-c-db2oltp-<INSTANCE_ID>-scc` is created automatically.

The contents of the SCC depend on whether you change the [node settings to allow Db2U to make unsafe sysctl changes](#).

- >
You do not change the node settings

```
allowHostDirVolumePlugin: true  
allowHostIPC: false  
allowHostNetwork: false  
allowHostPID: false  
allowHostPorts: false
```

```

allowPrivilegeEscalation: true
allowPrivilegedContainer: true
allowedCapabilities:
- FOWNER
- SETGID
- SETUID
- CHOWN
- DAC_OVERRIDE
- SYS_RESOURCE
- IPC_OWNER
- SYS_NICE
- FSETID
- SETFCAP
- SETPCAP
- SYS_CHROOT
- KILL
- AUDIT_WRITE
apiVersion: security.openshift.io/v1
defaultAddCapabilities: null
fsGroup:
  type: RunAsAny
groups: []
kind: SecurityContextConstraints
metadata:
  name: ${SCC_NAME}
priority: 10
readOnlyRootFilesystem: false
requiredDropCapabilities:
- ALL
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: MustRunAs
supplementalGroups:
  type: RunAsAny
users:
- system:serviceaccount:${PROJECT_CPD_INSTANCE}:${SERVICE_ACCOUNT}
volumes:
- '*'

```

- >
You change the node settings to allow Db2U to make unsafe sysctl changes

```

allowHostDirVolumePlugin: false
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: false
allowPrivilegedContainer: false
allowedCapabilities: []
allowedUnsafeSysctls:
- kernel.shmni
- 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: ${SCC_NAME}
priority: null
readOnlyRootFilesystem: false

```

```

requiredDropCapabilities:
- KILL
- SETUID
- SETGID
- MKNOD
- ALL
runAsUser:
  type: MustRunAs
  uid: 500
seLinuxContext:
  type: MustRunAs
supplementalGroups:
  type: RunAsAny
users:
- system:serviceaccount:${PROJECT_CPD_INSTANCE}:${SERVICE_ACCOUNT}
volumes:
- configMap
- downwardAPI
- emptyDir
- persistentVolumeClaim
- projected
- secret

```

Extended description of the allowed capabilities

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.

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.

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.

FSETID

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

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.

Creating the custom security context constraint for Db2 Warehouse

The Db2 Warehouse service requires a custom security context constraint (SCC).

Default SCC for SMP databases

When you create a Db2 Warehouse SMP instance, an SCC named `<NAMESPACE>-c-db2wh-<INSTANCE_ID>-scc` is created automatically.

The contents of the SCC depend on whether you change the [node settings to allow Db2U to make unsafe `sysctl` changes](#).

- >

You do not change the node settings

```
allowHostDirVolumePlugin: true
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: true
allowPrivilegedContainer: true
allowedCapabilities:
- FOWNER
- SETGID
- SETUID
- CHOWN
- DAC_OVERRIDE
- SYS_RESOURCE
- IPC_OWNER
- SYS_NICE
- FSETID
- SETFCAP
- SETPCAP
- SYS_CHROOT
- KILL
- AUDIT_WRITE
apiVersion: security.openshift.io/v1
defaultAddCapabilities: null
fsGroup:
  type: RunAsAny
groups: []
kind: SecurityContextConstraints
metadata:
  name: ${SCC_NAME}
priority: 10
readOnlyRootFilesystem: false
requiredDropCapabilities:
- ALL
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: MustRunAs
supplementalGroups:
  type: RunAsAny
users:
- system:serviceaccount:${PROJECT_CPD_INSTANCE}:${SERVICE_ACCOUNT}
volumes:
- '*'
```
- >

You change the node settings to allow Db2U to make unsafe `sysctl` changes

```
allowHostDirVolumePlugin: false
allowHostIPC: false
```

```

allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: true
allowPrivilegedContainer: false
allowedCapabilities:
- FOWNER
- SETGID
- SETUID
- CHOWN
- DAC_OVERRIDE
- SYS_RESOURCE
- IPC_OWNER
- SYS_NICE
- FSETID
- SETFCAP
- SETPCAP
- SYS_CHROOT
- KILL
- AUDIT_WRITE
allowedUnsafeSysctls:
- kernel.shmni
- kernel.shmmax
- kernel.shmall
- kernel.sem
- kernel.msgmni
- kernel.msgmax
- kernel.msgmnb
apiVersion: security.openshift.io/v1
defaultAddCapabilities: null
fsGroup:
  type: RunAsAny
groups: []
kind: SecurityContextConstraints
metadata:
  name: ${SCC_NAME}
priority: 10
readOnlyRootFilesystem: false
requiredDropCapabilities:
- ALL
runAsUser:
  type: MustRunAsNonRoot
seLinuxContext:
  type: MustRunAs
supplementalGroups:
  type: RunAsAny
users:
- system:serviceaccount:${PROJECT_CPD_INSTANCE}:${SERVICE_ACCOUNT}
volumes:
- '*'

```

Default SCC for MPP databases

When you create a Db2 Warehouse MPP instance, an SCC named `<NAMESPACE>-c-db2wh-<INSTANCE_ID>-scc` is created automatically.

The contents of the SCC depend on whether you change the [node settings to allow Db2U to make unsafe sysctl changes](#).

- >
You do not change the node settings

```

allowHostDirVolumePlugin: true
allowHostIPC: true
allowHostNetwork: false
allowHostPID: false
allowHostPorts: true
allowPrivilegeEscalation: true
allowPrivilegedContainer: true
allowedCapabilities:
- FOWNER

```

```

- SETGID
- SETUID
- CHOWN
- DAC_OVERRIDE
- SYS_RESOURCE
- IPC_OWNER
- SYS_NICE
- FSETID
- SETFCAP
- SETPCAP
- SYS_CHROOT
- KILL
- AUDIT_WRITE
apiVersion: security.openshift.io/v1
defaultAddCapabilities: null
fsGroup:
  type: RunAsAny
groups: []
kind: SecurityContextConstraints
metadata:
  name: ${SCC_NAME}
priority: 10
readOnlyRootFilesystem: false
requiredDropCapabilities:
- ALL
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: MustRunAs
supplementalGroups:
  type: RunAsAny
users:
- system:serviceaccount:${PROJECT_CPD_INSTANCE}:${SERVICE_ACCOUNT}
volumes:
- '*'

```

- >
You change the node settings to allow Db2U to make unsafe sysctl changes

```

allowHostDirVolumePlugin: false
allowHostIPC: true
allowHostNetwork: false
allowHostPID: false
allowHostPorts: true
allowPrivilegeEscalation: true
allowPrivilegedContainer: false
allowedCapabilities:
- FOWNER
- SETGID
- SETUID
- CHOWN
- DAC_OVERRIDE
- SYS_RESOURCE
- IPC_OWNER
- SYS_NICE
- FSETID
- SETFCAP
- SETPCAP
- SYS_CHROOT
- KILL
- AUDIT_WRITE
apiVersion: security.openshift.io/v1
defaultAddCapabilities: null
fsGroup:
  type: RunAsAny
groups: []
kind: SecurityContextConstraints
metadata:
  name: ${SCC_NAME}
priority: 10
readOnlyRootFilesystem: false
requiredDropCapabilities:

```

```

- ALL
runAsUser:
  type: MustRunAsNonRoot
seLinuxContext:
  type: MustRunAs
supplementalGroups:
  type: RunAsAny
users:
- system:serviceaccount:${PROJECT_CPD_INSTANCE}:${SERVICE_ACCOUNT}
volumes:
- '*'

```

Extended description of the allowed capabilities

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.

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.

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.

FSETID

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

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.

Creating the custom security context constraint for embedded Db2 databases

Embedded Db2 databases require a custom security context constraint (SCC).

Each embedded Db2 database has its own SCC.

The following services use an embedded Db2 database:

- Data Virtualization
- Db2 Big SQL
- OpenPages® (created only if you use an embedded instance of Db2)
- Watson™ Knowledge Catalog

The embedded database is accessible only by the service or service instance that creates the database:

Default SCC

When you install a service that uses an embedded Db2 database, an SCC is created automatically.

The name of the SCC depends on the service that embeds the database. For example, if you install Watson Knowledge Catalog, the SCC is named `<NAMESPACE>-c-db2oltp-wkc-scc`.

The contents of the SCC depend on whether you change the [node settings to allow Db2U to make unsafe `sysctl` changes](#).

- >

You do not change the node settings

```
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_CPD_INSTANCE}:${SERVICE_ACCOUNT}
volumes:
- configMap
- downwardAPI
- emptyDir
- hostPath
- persistentVolumeClaim
- projected
- secret
```
- >

You change the node settings to allow Db2U to make unsafe `sysctl` changes

```

allowHostDirVolumePlugin: false
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: false
allowPrivilegedContainer: false
allowedCapabilities: []
allowedUnsafeSysctls:
- kernel.shmni
- 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: ${SCC_NAME}
priority: null
readOnlyRootFilesystem: false
requiredDropCapabilities:
- KILL
- SETUID
- SETGID
- MKNOD
- ALL
runAsUser:
  type: MustRunAs
  uid: 500
seLinuxContext:
  type: MustRunAs
supplementalGroups:
  type: RunAsAny
users:
- system:serviceaccount:${PROJECT_CPD_INSTANCE}:${SERVICE_ACCOUNT}
volumes:
- configMap
- downwardAPI
- emptyDir
- persistentVolumeClaim
- projected
- secret

```

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.

Node settings for services

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

Node settings	Services that require changes to the setting	Environments	Instructions
---------------	--	--------------	--------------

Node settings	Services that require changes to the setting	Environments	Instructions
Load balancer timeout settings	<ul style="list-style-type: none"> • Db2® Data Gate • OpenPages® • Watson™ Discovery • Watson Knowledge Catalog • Also recommended if you are working with large data sets or you have slower network speeds. 	All environments	<ul style="list-style-type: none"> • Load balancer timeout settings
CRI-O container settings	<ul style="list-style-type: none"> • Cognos® Analytics • Data Virtualization • Db2 • Db2 Big SQL • Db2 Warehouse • Watson Discovery • Watson Knowledge Catalog • Watson Studio • Watson Machine Learning Accelerator 	All environments except IBM Cloud	<ul style="list-style-type: none"> • CRI-O container settings
Kernel parameter settings	<ul style="list-style-type: none"> • Data Virtualization • Db2 • Db2 Big SQL • Db2 Warehouse • Watson Knowledge Catalog • Watson Studio 	All environments	<ul style="list-style-type: none"> • 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 • Provisioning Db2 Big SQL on ROKS • Preparing to install the service (Data Virtualization)
Power settings			<ul style="list-style-type: none"> • Power settings
GPU settings	<ul style="list-style-type: none"> • Jupyter Notebooks with Python 3.9 for GPU 	All environments	<ul style="list-style-type: none"> • GPU settings

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.

This setting is required if you plan to install the following services:

- Db2 Data Gate
- OpenPages
- Watson Discovery
- Watson Knowledge Catalog

This setting is also recommended if you are working with large data sets or you have slower network speeds. For example, you might need to increase this value if you receive a timeout or failure when you upload a large file.

The following procedures show how to change the timeout settings if 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 settings.

If you are using HAProxy, the load balancer node is the OpenShift cluster public node.

- >
Changing HAProxy 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 minimum values are as follows:

Db2 Data Gate

```
timeout client 7500s
timeout server 7500s
```

OpenPages

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

Watson Discovery

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

Watson Knowledge Catalog

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

2. If necessary, change the timeout values by running the following commands:

- 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 HAProxy 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. You can use the `annotate` command to set the timeout value to a maximum of 50s. If you need to set the timeout value higher than 50s, open a support ticket with the Load Balance Service team. The server might time out during long running transactions. For more information, see [Connection timeouts](#).

CRI-O container settings

To ensure that some services can run correctly, you must run the `cpd-cli manage apply-crio` command to change settings that are required for the CRI-O container runtime on the OpenShift Container Platform.

Run the command if you plan to install one or more of the following services:

- Cognos Analytics
- Data Virtualization
- Db2
- Db2 Big SQL
- Db2 Warehouse

- Watson Discovery
- Watson Knowledge Catalog
- Watson Studio
- Watson Machine Learning Accelerator

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 run this command.

Apply the required Container Runtime Interface (CRI-O) settings to your cluster nodes. When you run this command, the `pids_limit` is set to **12288**.

Best practice: You can run the commands in this task exactly as written if you set up environment variables. For instructions, see [Setting up installation environment variables](#).

Ensure that you source the environment variables before you run the commands in this task.

1. Run the `cpd-cli manage`

`login-to-ocp` command to log in to the cluster as a user with sufficient permissions to complete this task. For example:

```
cpd-cli manage login-to-ocp \
--username=${OCP_USERNAME} \
--password=${OCP_PASSWORD} \
--server=${OCP_URL}
```

2. Run the following command to apply the CRI-O settings:

```
cpd-cli manage apply-crio \
--openshift-type=${OPENSIFT_TYPE}
```

Kernel parameter settings

- [Configuring kubelet to allow Db2U to make unsafe sysctl calls in on-premises and private cloud deployments](#)

Run the `cpd-cli manage apply-db2-kubelet` command to configure `kubelet` to allow Db2U to make unsafe sysctl calls for Db2 to manage required memory settings.

Run the command if you plan to install one or more of the following services:

- Data Virtualization
- Db2
- Db2 Big SQL
- Db2 Warehouse
- Watson Knowledge Catalog
- Watson Studio

Apply the required kubelet configuration to all cluster nodes to allow Db2U to make unsafe sysctl changes for kernel parameters.

Best practice: You can run the commands in this task exactly as written if you set up environment variables. For instructions, see [Setting up installation environment variables](#).

Ensure that you source the environment variables before you run the commands in this task.

1. Run the `cpd-cli manage`

`login-to-ocp` command to log in to the cluster as a user with sufficient permissions to complete this task. For example:

```
cpd-cli manage login-to-ocp \
--username=${OCP_USERNAME} \
--password=${OCP_PASSWORD} \
--server=${OCP_URL}
```

2. Run the following command to apply the kubelet configuration:

```
cpd-cli manage apply-db2-kubelet \
--openshift-type=${OPENSIFT_TYPE}
```



```
FYVEhSRUFEUy8kve9UQUxDt1JFwPdvVJSRU5UU01UPSQoL2Jpbi9sc2NwdSB8IC9iaW4vYXdrIC1GOiAnICQxIH4
gL15UaHJLYWRcKHncKSBwZXIgy29yZSQvIHtwcmludCakMn0nfc9iaW4veGFyZ3MpcGpTTVRMQUJFTD0kKC9iaW4v
b2MgZ2V0IG5vZGUgJEhPU1ROQU1FIC1MIFNNVCAAtLW5vLWhlYWRLcnMgfc9iaW4vYXdrICd7chJpbnQgJDZ9JykKa
WYgW1sgLW4gJFNNVExBQkVMIF1dCiAgdGh1bgogICAgY2FzZSAkU01UTEFRCRUwgaW4KICAgICAgMSkgVEFSR0VUU0
1UPTeKICAgIDs7CiAgICAgIDIpIFRBUkdFVFNND0yCiAgICA7OwogICAgICA0KSBUQVJHRVRTTVQ9NAogICAgOzs
KICAgICAgOCkgVEFSR0VUU01UPTgKICAgIDs7CiAgICAgICopIFRBUkdFVFNND0kQ1VSUkVOVFNNVCA7IGVjaG8g
I1NNVCB2YWxlZSBtdXN0IGJlIDEsIDIsIDQsIG9yIDggYW5kIHNTYXxsZXIgdGhhbiBNYXhpbXVtIFNNVC4iCiAgI
CA7OwogICAgZXNhYwogIGVsc2UKICAgIFRBUkdFVFNND0kTUFYU01UCmZpCgpDVVJSRU5UU01UPSQoL2Jpbi9sc2
NwdSB8IC9iaW4vYXdrIC1GOiAnICQxIH4gL15UaHJLYWRcKHncKSBwZXIgy29yZSQvIHtwcmludCakMn0nfc9iaW4
veGFyZ3MpcGppZiBbWyAkQ1VSUkVOVFNNVCAtbmUgJFRBUkdFVFNND0kQ1VSUkVOVFNNVCA7IGVjaG8gRD0wCiAgI
CBJTk1UT0ZGVEhSRUFEPSRUQVJHRVRTTVQKICAgIGlmIFtbICRNQVhTTTVQgLWdlICRUQVJHRVRTTVQgX
V0KICAgICAgdGh1bgogICAgICAgIHdoaWxlIFtbICRJTtk1UT05USFJFQUQgLWx0ICRNQVhUSFJFQURTIF1dCiAgIC
AgICAgZG8KICAgICAgICAgIE90VEhSRUFEPSRJTtk1UT05USFJFQUQKICAgICAgICAgIE9GRlRIUkVBRD0kSU5JVE9
GRlRIUkVBRaoKICAgICAgICAgIHdoaWxlIFtbICRPTlRIUkVBRCAtbHQgJE9GRlRIUkVBRCBdXQogICAgICAgICAg
ZG8KICAgICAgICAgICAgL2Jpbi91Y2hvIDEgPiAvc3lzL2Rldm1jZXMvc3lzZGVtL2NwdS9jcHUkT05USFJFQUQvb
25saW5lCiAgICAgICAgICAgICAgldCBPTlRIUkVBRD0kT05USFJFQUQrMQogICAgICAgICAgZG9uZQogICAgICAgIC
AgbGV0IElOSVRPTlRIUkVBRD0kSU5JVE90VEhSRUFEKyRNQVhTTTVQKICAgICAgICAgIHdoaWxlIFtbICRPRkZUSFJ
FQUQgLWx0ICRJTtk1UT05USFJFQUQgXV0KICAgICAgICAgICRvCiAgICAgICAgICAgIC9iaW4vZWNobyAwID4gL3N5
cy9kZXZpY2VzL3N5c3Rlbn9jcHUvY3B1JE9GRlRIUkVBRD0kT05USFJFQUQrMQogICAgICAgICAgICAgbGV0IE9GRlRIUkVBR
D0kT0ZGVEhSRUFEKzEKICAgICAgICAgICRvbmUKICAgICAgICAgICAgldCBJTk1UT0ZGVEhSRUFEPSRJTtk1UT0ZGVE
hSRUFEKyRNQVhTTTVQKICAgICAgICAgICBkb25lCiAgICAgICAgICAgICVsc2UKICAgICAgICAgICBlY2hvICJUYXJnZXQgU01UIG1lc3Q
gYmUgc21hbGxlciBvcilB1cXVhbCB0aGFuIE1heGltdW0gU01UIH1cHBvcnRlZCIKICAgICAgICAgZpCmZp
```

```
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.

GPU settings

To install the NVIDIA GPU Operator on a cluster connected to the internet:

1. [Install the Node Feature Discovery Operator.](#)
2. [Install the GPU Operator.](#)

To install the NVIDIA GPU Operator on an air-gapped environment, see [Deploy GPU Operators in a disconnected or airgapped environment.](#)

- [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: [Creating custom security context constraints for services](#)

Next topic: [Updating the global image pull secret](#)

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 nodes, 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 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 [Kernel parameter requirements \(Linux\)](#).
- **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/d
oc/c0057140.html
# and generate the Node Tuning Operator CRD yaml.

tuned_crud_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_cr_d_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)))
((IPC_MNI_LIMIT=32 * 1024))
tr ' ' '\n' < /proc/cmdline | grep -q ipc_mni_extend && ((IPC_MNI_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
'ipc_mni_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} ${SEMMS} ${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

```

Updating 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 add to the global image pull secret depend on where you want to pull images from.

Installation phase

- Setting up a client workstation
- Collecting required information
-  Preparing your cluster
- Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

A cluster administrator must complete this task.

When do you need to complete this task?

You must complete this task before you install Cloud Pak for Data for the first time.

Before you begin

Best practice: You can run the commands in this task exactly as written if you set up environment variables. For instructions, see [Setting up installation environment variables](#).

Ensure that you source the environment variables before you run the commands in this task.

About this task

Use the appropriate `cpd-cli`

`manage` command to create or update the global image pull secret with the appropriate credentials:

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](#).

Use the [cpd-cli](#)

[manage](#)

[add-icr-cred-to-global-pull-secret](#) command to create or update the global image pull secret.

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.

Use the [cpd-cli](#)

[manage](#)

[add-cred-to-global-pull-secret](#) command to create or update the global image pull secret.

Procedure

1. Run the `cpd-cli`

`manage`

`login-to-ocp` command to log in to the cluster as a user with sufficient permissions to complete this task. For example:

```
cpd-cli manage login-to-ocp \  
--username=${OCP_USERNAME} \  
--password=${OCP_PASSWORD} \  
--server=${OCP_URL}
```


Tip: The `login-to-ocp` command takes the same input as the `oc login` command. Run `oc login --help` for details.

2. Run the appropriate command to update the global image pull secret:

-  IBM Entitled Registry

Run the following command to provide your IBM entitlement API key to the global image pull secret:

```
cpd-cli manage add-icr-cred-to-global-pull-secret \  
${IBM_ENTITLEMENT_KEY}
```

-  Private container registry

Run the following command to provide the credentials of a user that can *pull* images from the registry:

```
cpd-cli manage add-cred-to-global-pull-secret \  
${PRIVATE_REGISTRY_LOCATION} \  
${PRIVATE_REGISTRY_PULL_USER} \  
${PRIVATE_REGISTRY_PULL_PASSWORD}
```

3. Get the status of the nodes.

```
cpd-cli manage 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 to do next

- If you are pulling images from the IBM Entitled Registry, you are ready to complete [Installing the IBM Cloud Pak for Data platform and services](#).
- If you are pulling images from a private container registry, you are ready to complete [Mirroring images to a private container registry](#).

Previous topic: [Changing required node settings](#)

Next topic: [Mirroring images to a private container registry](#)


Mirroring images to a 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.

Installation phase

Setting up a client workstation

Collecting required information

 Preparing your cluster

Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

A cluster administrator and private container registry administrator must work together to mirror images to the private container registry.

When do you need to complete this task?

If you want to mirror images to a private container registry, you must complete this task in the following situations:

- Before you install Cloud Pak for Data for the first time.
- Before you upgrade to a newer release of Cloud Pak for Data

If you do not want to mirror images to a private container registry, go to [Installing the IBM Cloud Pak for Data platform and services](#).

Before you begin

Ensure that your private container registry meets the [Private container registry requirements](#).

About this task

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.

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 by answering the following question:

Can you set up a client workstation that can connect to the internet and the private container registry?

Yes

You can [mirror the images directly from the IBM Entitled Registry to the private container registry](#).

No, the private container registry is in a restricted network

You must [mirror the images to an intermediary container registry before you can mirror the images to the private container registry](#).

The `cpd-cli manage`

`mirror-images` command automatically sets up an intermediary container registry on the client workstation. You must be able to move the intermediary container registry behind your firewall. For example, you can use:

Options	Details
Use a <i>portable compute device</i> , such as a laptop, that you can move behind your firewall.	You can use the same device to: <ul style="list-style-type: none"> • Mirror images from the IBM Entitled Registry to the intermediary container registry. • Mirror images from the intermediary container registry to the private container registry.
Use a <i>portable storage device</i> , such as a USB drive, that you can move behind your firewall.	You must set up two client workstations: <ul style="list-style-type: none"> • A workstation that can connect to the internet. From this workstation, you can mirror the images from the IBM Entitled Registry to the intermediary container registry on the portable storage device. • A workstation that can connect to the private container registry. After you move the portable storage device to this workstation, you can mirror the images from the intermediary container registry to the private container registry.
Use a <i>file transfer protocol</i> , such as <code>scp</code> or <code>sftp</code> , to move images behind your firewall.	You must set up two client workstations: <ul style="list-style-type: none"> • A workstation that can connect to the internet. From this workstation, you can mirror the images from the IBM Entitled Registry to the intermediary container registry. • A workstation that can connect to the private container registry. After you transfer the intermediary container registry to this workstation, you can mirror the images from the intermediary container registry to the private container registry.

Procedure

1. Complete the appropriate task to mirror images to your private container registry:
 - [Mirroring images directly to the private container registry](#)
 - [Mirroring images using an intermediary container registry](#)
2. Complete [Configuring an image content source policy](#).
 - [Mirroring images directly to the private container registry](#)
If your client workstation can connect to the internet and to the private container registry, you can mirror the images directly to your private container registry.
 - [Mirroring images using an intermediary container registry](#)
If your client workstation cannot connect to the internet and to the private container registry, you must mirror images to

- an intermediary container registry before you can mirror the images to your private container registry.
- **Configuring an image content source policy**
If you mirror images to a private container registry, you must tell your cluster where to find the software images by creating an image content source policy.

Previous topic: [Updating the global image pull secret](#)

Mirroring images directly to the private container registry


If your client workstation can connect to the internet and to the private container registry, you can mirror the images directly to your private container registry.

If your client workstation *cannot* connect to the internet and to the private container registry, see [Mirroring images using an intermediary container registry](#).

Installation phase

Setting up a client workstation

Collecting required information

 Preparing your cluster

Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

A user with permissions to *push* images to the private container registry must complete this task.

When do you need to complete this task?

If you want to mirror images to a private container registry, you must complete this task in the following situations:

- Before you install Cloud Pak for Data for the first time.
- Before you upgrade to a newer release of Cloud Pak for Data

Before you begin

Best practice: You can run the commands in this task exactly as written if you set up environment variables. For instructions, see [Setting up installation environment variables](#).

Ensure that you source the environment variables before you run the commands in this task.

About this task

Use the [cpd-cli manage](#) commands to mirror the images from the IBM® Entitled Registry to the private container registry.

The following steps assume that you will mirror all of the components in a single step. The components that are mirrored are determined by the `COMPONENTS` variable, from the installation environment variables script. If you want to mirror a specific component instead of multiple components, you can export `COMPONENTS` with the appropriate [component ID](#).

Procedure

1. Log in to the IBM Entitled Registry registry:

```
cpd-cli manage login-entitled-registry \  
${IBM_ENTITLEMENT_KEY}
```

2. Log in to the private container registry.

The following command assumes that you are using private container registry that is secured with credentials:

```
cpd-cli manage login-private-registry \  
${PRIVATE_REGISTRY_LOCATION} \  
${PRIVATE_REGISTRY_PUSH_USER} \  
${PRIVATE_REGISTRY_PUSH_PASSWORD}
```


If your private registry is not secured, see [cpd-cli manage login-private-registry](#) for additional options.

3. Confirm that you have access to the images that you want to mirror from the IBM Entitled Registry:
 - a. Inspect the IBM Entitled Registry:

Tip: If you want to validate that you have access to the images for a specific component, you can run the following command before you run the `list-images` command:

```
export COMPONENTS=<component-ID>

cpd-cli manage list-images \
--components=${COMPONENTS} \
--release=${VERSION} \
--inspect_source_registry=true
```

The output is saved to the `list_images.csv` file in the `cpd-cli-workspace/olm-utils-workspace/work/offline/${VERSION}` directory.

- b. Check the output for errors:

```
grep "level=fatal" list_images.csv
```

The command returns images that failed because of authorization errors or network errors.

4. Mirror the images to the private container registry.

Tip: If you want to mirror images for a specific component, you can run `export COMPONENTS=<component-ID>` before you run the command.

```
cpd-cli manage mirror-images \
--components=${COMPONENTS} \
--release=${VERSION} \
--target_registry=${PRIVATE_REGISTRY_LOCATION}
```

For each component, the command generates a log file in the `cpd-cli-workspace/olm-utils-workspace/work` directory.

5. Confirm that the images were mirrored to the private container registry:
 - a. Inspect the contents of the private container registry:

```
cpd-cli manage list-images \
--components=${COMPONENTS} \
--release=${VERSION} \
--target_registry=${PRIVATE_REGISTRY_LOCATION} \
--case_download=false
```

The output is saved to the `list_images.csv` file in the `cpd-cli-workspace/olm-utils-workspace/work/offline/${VERSION}` directory.

- b. Check the output for errors:

```
grep "level=fatal" list_images.csv
```

The command returns images that are missing or that cannot be inspected.

What to do next

Now that you've mirrored the images to the private container registry, you are ready to [Configuring an image content source policy](#).

Mirroring images using an intermediary container registry


If your client workstation cannot connect to the internet and to the private container registry, you must mirror images to an intermediary container registry before you can mirror the images to your private container registry.

If your client workstation *can* connect to the internet and to the private container registry, see [Mirroring images directly to the private container registry](#).

Installation phase

Setting up a client workstation

Collecting required information

 Preparing your cluster

Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

A user with permissions to *push* images to the private container registry must complete this task.

When do you need to complete this task?

If you want to mirror images to a private container registry, you must complete this task in the following situations:

- Before you install Cloud Pak for Data for the first time.
- Before you upgrade to a newer release of Cloud Pak for Data

Before you begin

Best practice: You can run the commands in this task exactly as written if you set up environment variables. For instructions, see [Setting up installation environment variables](#).

Ensure that you source the environment variables before you run the commands in this task.

About this task

Use the `cpd-cli` `manage` commands to:

- Mirror the images from the IBM® Entitled Registry to an intermediary container registry on the client workstation.
- Mirror the images from the intermediary container registry to the private container registry.

The `cpd-cli` `manage`

`mirror-images` command automatically sets up an intermediary container registry on the client workstation. The address of the intermediary container registry is `127.0.0.1:12443`.

You must be able to move the intermediary container registry behind your firewall.

- | > | |
|---|---|
| Options for moving the intermediary container registry | |
| Options | Details |
| Use a <i>portable compute device</i> , such as a laptop, that you can move behind your firewall. | You can use the same device to: <ul style="list-style-type: none">◦ Mirror images from the IBM Entitled Registry to the intermediary container registry.◦ Mirror images from the intermediary container registry to the private container registry. |
| Use a <i>portable storage device</i> , such as a USB drive, that you can move behind your firewall. | You must set up two client workstations: <ul style="list-style-type: none">◦ A workstation that can connect to the internet. From this workstation, you can mirror the images from the IBM Entitled Registry to the intermediary container registry on the portable storage device.◦ A workstation that can connect to the private container registry. After you move the portable storage device to this workstation, you can mirror the images from the intermediary container registry to the private container registry. |

Options	Details
Use a <i>file transfer protocol</i> , such as scp or sftp , to move images behind your firewall.	You must set up two client workstations: <ul style="list-style-type: none"> ○ A workstation that can connect to the internet. From this workstation, you can mirror the images from the IBM Entitled Registry to the intermediary container registry. ○ A workstation that can connect to the private container registry. After you transfer the intermediary container registry to this workstation, you can mirror the images from the intermediary container registry to the private container registry.

Procedure

1. From a client workstation that can connect to the internet:
 - a. Log in to the IBM Entitled Registry registry:

```
cpd-cli manage login-entitled-registry \
${IBM_ENTITLEMENT_KEY}
```

- b. Confirm that you have access to the images that you want to mirror from the IBM Entitled Registry:

- i. Inspect the IBM Entitled Registry:

Tip: If you want to validate that you have access to the images for a specific component, you can run the following command before you run the **list-images** command:

```
export COMPONENTS=<component-ID>
```

```
cpd-cli manage list-images \
--components=${COMPONENTS} \
--release=${VERSION} \
--inspect_source_registry=true
```

The output is saved to the **list_images.csv** file in the **cpd-cli-workspace/olm-utils-workspace/work/offline/\${VERSION}** directory.

- ii. Check the output for errors:

```
grep "level=fatal" list_images.csv
```

The command returns images that failed because of authorization errors or network errors.

- c. Mirror the images to the intermediary container registry.

The command automatically sets up an intermediary container registry on the client workstation. The address of the intermediary container registry is **127.0.0.1:12443**.

Tip: If you want to mirror images for a specific component, you can run **export COMPONENTS=<component-ID>** before you run the command.

```
cpd-cli manage mirror-images \
--components=${COMPONENTS} \
--release=${VERSION} \
--target_registry=127.0.0.1:12443
```

For each component, the command generates a log file in the **cpd-cli-workspace/olm-utils-workspace/work** directory.

- d. Confirm that the images were mirrored to the intermediary container registry:

- i. Inspect the contents of the intermediary container registry:

```
cpd-cli manage list-images \
--components=${COMPONENTS} \
--release=${VERSION} \
--target_registry=127.0.0.1:12443 \
--case_download=false
```

The output is saved to the **list_images.csv** file in the **cpd-cli-workspace/olm-utils-workspace/work/offline/\${VERSION}** directory.

ii. Check the output for errors:

```
grep "level=fatal" list_images.csv
```

The command returns images that are missing or that cannot be inspected.

2. Move the intermediary container registry behind the firewall.
3. From a client workstation that connect to private container registry:
 - a. Log in to the private container registry.

The following command assumes that you are using private container registry that is secured with credentials:

```
cpd-cli manage login-private-registry \  
{PRIVATE_REGISTRY_LOCATION} \  
{PRIVATE_REGISTRY_PUSH_USER} \  
{PRIVATE_REGISTRY_PUSH_PASSWORD}
```

If your private registry is not secured, see [cpd-cli manage login-private-registry](#) for additional options.

- b. Mirror the images from the intermediary container registry to the private container registry.

Tip: If you want to mirror images for a specific component, you can run `export COMPONENTS=<component-ID>` before you run the command.

```
cpd-cli manage mirror-images \  
--components={COMPONENTS} \  
--release={VERSION} \  
--source_registry=127.0.0.1:12443 \  
--target_registry={PRIVATE_REGISTRY_LOCATION} \  
--case_download=false
```

For each component, the command generates a log file in the `cpd-cli-workspace/olm-utils-workspace/work` directory.

- c. Confirm that the images were mirrored to the private container registry:
 - i. Inspect the contents of the intermediary container registry:

```
cpd-cli manage list-images \  
--components={COMPONENTS} \  
--release={VERSION} \  
--target_registry={PRIVATE_REGISTRY_LOCATION} \  
--case_download=false
```

The output is saved to the `list_images.csv` file in the `cpd-cli-workspace/olm-utils-workspace/work/offline/{VERSION}` directory.

ii. Check the output for errors:

```
grep "level=fatal" list_images.csv
```

The command returns images that are missing or that cannot be inspected.


What to do next

Now that you've mirrored the images to the private container registry, you are ready to [Configure an image content source policy](#).

Configuring an image content source policy

If you mirror images to a private container registry, you must tell your cluster where to find the software images by creating an image content source policy.

Installation phase

- Setting up a client workstation
- Collecting required information
-  Preparing your cluster
- Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

A cluster administrator must complete this task.

When do you need to complete this task?

If you mirrored images to a private container registry, you must complete this task before you install Cloud Pak for Data for the first time.

Before you begin

Best practice: You can run the commands in this task exactly as written if you set up environment variables. For instructions, see [Setting up installation environment variables](#).

Ensure that you source the environment variables before you run the commands in this task.

About this task

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.

Use the [cpd-cli manage](#)

[apply-icsp](#) command to create or update the image content source policy for the Cloud Pak for Data images.

Procedure

1. Run the `cpd-cli`

`manage`

`login-to-ocp` command to log in to the cluster as a user with sufficient permissions to complete this task. For example:

```
cpd-cli manage login-to-ocp \
--username=${OCP_USERNAME} \
--password=${OCP_PASSWORD} \
--server=${OCP_URL}
```

Tip: The `login-to-ocp` command takes the same input as the `oc login` command. Run `oc login --help` for details.

2. Create or update the required image content source policy:

```
cpd-cli manage apply-icsp \
${PRIVATE_REGISTRY_LOCATION}
```

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 to do next

Now that you've created the image content source policy, you are ready to complete [Installing the IBM Cloud Pak for Data platform and services](#).

Installing the IBM Cloud Pak for Data platform and services

To install the IBM® Cloud Pak for Data platform and services, you must create the required Operator Lifecycle Manager (OLM) objects and custom resources (CRs) for the software that you want to use.

Before you begin

Ensure that the following tasks are complete before you install the platform and services:

- You set up a [client workstation](#) from which you will run the installation.
- You collected the [required information](#) that you will need to complete the installation.
- A cluster administrator [prepared the cluster](#) for the installation.
Important: If you are using a private container registry, the images must be in the private container registry before you run the installation.

Ensure that you have the required permissions:

- You must have sufficient privileges to install operators on the cluster.
If you don't have sufficient privileges to install operators, you will need to work with a cluster administrator to complete specific tasks.
- You must be an administrator of the projects where you need to install the operators and the software.

About this task

The `cpd-cli`

`manage` commands enable you to install all of the components that you need to install at the same time. For example, if you plan to implement the [MLOps and Trustworthy AI](#) use case, you can install all of the components that are required to support that use case at the same time.

The installation instructions assume that you are installing all of the components at the same time, which enables you to complete the installation in fewer steps. (You can always install additional services as needed on your environment by following the appropriate instructions in [Services](#).)

The `cpd-cli`

`manage` commands are designed to simplify the installation process:

- If it detects that you already have a component installed at the specified release, the `cpd-cli` does not attempt to install the component again.
For example, if the `cpd-cli` detects that IBM Cloud Pak® foundational services is already installed at Version 3.19.0 or later, the `cpd-cli` will not make any changes to the existing IBM Cloud Pak foundational services installation.

However, if the `cpd-cli` detects that the IBM Cloud Pak foundational services is installed at a version below 3.19.0, the `cpd-cli` will automatically upgrade the IBM Cloud Pak foundational services installation.
- If you encounter an issue when you are installing a specific component, the `cpd-cli` gives you the option to resume the installation from the point of failure.

Important: All of the Cloud Pak for Data components must be installed at the same release.
The installation method that you choose depends on:

- How strictly you want to enforce the division between projects (namespaces)
 - If you want the Cloud Pak for Data operators to watch the same projects that the IBM Cloud Pak foundational services operators watch, follow the [express installation instructions](#).
 - If you want the Cloud Pak for Data operators to watch only the projects where the Cloud Pak for Data platform and services are installed, follow the [specialized installation instructions](#).
- If you want to install IBM Cloud Pak foundational services in a project other than `ibm-common-services`, follow the [specialized installation instructions](#).
- **Express installations**
In an express installation, the IBM Cloud Pak for Data operators and the IBM Cloud Pak foundational services operators are installed in the same project (namespace). The operators are granted permission to watch the project or projects where the Cloud Pak for Data platform and services are installed.
- **Specialized installations**
In a specialized installation, the IBM Cloud Pak foundational services operators are installed in one project (namespace) and the IBM Cloud Pak for Data operators are installed in another project.

Related tasks

- [Collecting required information](#)
- [Uninstalling the platform and services](#)

Related reference

- [Setting up a client workstation](#)
- [Preparing your cluster](#)

Express installations


In an express installation, the IBM® Cloud Pak for Data operators and the IBM Cloud Pak® foundational services operators are installed in the same project (namespace). The operators are granted permission to watch the project or projects where the Cloud Pak for Data platform and services are installed.

Installation phase

Setting up a client workstation

Collecting required information

Preparing your cluster

 Installing the Cloud Pak for Data platform and services

Before you begin

Ensure that the following tasks are complete before you install the platform and services:

- You set up a [client workstation](#) from which you will run the installation.
- You collected the [required information](#) that you will need to complete the installation.
- A cluster administrator [prepared the cluster](#) for the installation.
Important: If you are using a private container registry, the images must be in the private container registry before you run the installation.

Ensure that you have the required permissions:

- You must have sufficient privileges to install operators on the cluster.
If you don't have sufficient privileges to install operators, you will need to work with a cluster administrator to complete specific tasks.
- You must be an administrator of the projects where you need to install the operators and the software.

About this task

To perform an express installation, complete the following tasks:

1. [Creating OLM objects for an express installation](#)

A cluster administrator, or a user with the appropriate permissions to install operators, must create the Operator Lifecycle Manager (OLM) objects, such as operators and operator subscriptions, that are required to install the IBM Cloud Pak for Data platform and services. The OLM objects that you create depend on the services that you plan to install.

2. [Installing components in an express installation](#)

A project administrator can create custom resources to install the IBM Cloud Pak for Data platform and services in a project (namespace). The custom resources that you create depend on the services that you plan to install.

Creating OLM objects for an express installation

A cluster administrator, or a user with the appropriate permissions to install operators, must create the Operator Lifecycle Manager (OLM) objects, such as operators and operator subscriptions, that are required to install the IBM® Cloud Pak for Data platform and services. The OLM objects that you create depend on the services that you plan to install.

Installation phase

Setting up a client workstation

Collecting required information

Preparing your cluster



Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

A user with the appropriate permissions to install operators must complete this task.

When do you need to complete this task?

You must complete this task before you install Cloud Pak for Data for the first time.

Before you begin

Best practice: You can run the commands in this task exactly as written if you set up environment variables. For instructions, see [Setting up installation environment variables](#).

Ensure that you source the environment variables before you run the commands in this task.

About this task

Use the `cpd-cli` [manage](#)

[apply-olm](#) command to create the IBM Cloud Pak® foundational services operators and the Cloud Pak for Data operators in the `ibm-common-services` project.

The instructions assume that you are installing the operators for all of the components at the same time, which enables you to complete the task in fewer steps. You can always install additional operators if you decide to install additional services on your environment.

Remember: If the `cpd-cli` detects that you already have the OLM objects for the components at the specified release, the `cpd-cli` does not attempt to create the OLM objects again.

Procedure

1. Run the `cpd-cli` `manage`

`login-to-ocp` command to log in to the cluster as a user with sufficient permissions to complete this task. For example:


```
cpd-cli manage login-to-ocp \  
--username=${OCP_USERNAME} \  
--password=${OCP_PASSWORD} \  
--server=${OCP_URL}
```

Tip: The `login-to-ocp` command takes the same input as the `oc login` command. Run `oc login --help` for details.

2. Create the OLM objects for the specified components:

```
cpd-cli manage apply-olm \  
--release=${VERSION} \  
--components=${COMPONENTS}
```

Tip: If you want to preview the `oc` commands that the `cpd-cli manage`

`apply-olm` will issue on your behalf, you can run the command with `--preview=true`.

The `oc` commands are saved to the `preview.sh` file in the `cpd-cli-workspace/olm-utils-workspace/work` directory.

3. Run the following command to ensure that any services that have a dependency on third-party operators can automatically set the namespace scope on the third-party operators:

```
oc patch NamespaceScope common-service \  
-n ${PROJECT_CPFS_OPS} \  
--type=merge \  
--patch='{"spec": {"csvInjector": {"enable": true} } }'
```

Results

The operators for the specified components are created in the `ibm-common-services` project.

You can optionally run the [cpd-cli manage](#)

[get-olm-artifacts](#) command to get the list of catalog sources and operator subscriptions that are on the cluster.

```
cpd-cli manage get-olm-artifacts \  
--subscription_ns=${PROJECT_CPFS_OPS}
```

What to do next

Now that you've created the OLM objects, you are ready to complete [Installing components in an express installation](#).

Next topic: [Installing components in an express installation](#)

Installing components in an express installation

A project administrator can create custom resources to install the IBM® Cloud Pak for Data platform and services in a project (namespace). The custom resources that you create depend on the services that you plan to install.

Installation phase

- Setting up a client workstation

- Collecting required information

- Preparing your cluster



- Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

A project administrator must complete this task.

When do you need to complete this task?

You must complete this task at least once to install Cloud Pak for Data.

You can complete this task multiple times if you want to install Cloud Pak for Data in multiple projects (namespaces).

Before you begin

Best practice: You can run the commands in this task exactly as written if you set up environment variables. For instructions, see [Setting up installation environment variables](#).

Ensure that you source the environment variables before you run the commands in this task.

Ensure that you have storage that supports the components that you plan to install. For details, see [Storage requirements](#). The commands assume that you set up the environment variables with the appropriate storage classes names.

About this task

Use the `cpd-cli manage apply-cr` command to create the custom resources for the specified components.

The instructions assume that you are installing all of the components at the same time, which enables you to complete the task in fewer steps. You can always install additional services separately.

If you want to install a service in a tethered project, install the service separately.

Remember: If the `cpd-cli` detects that you already have a component installed at the specified release, the `cpd-cli` does not attempt to install the component again.

Procedure

1. Run the `cpd-cli manage login-to-ocp` command to log in to the cluster as a user with sufficient permissions to complete this task. For example:

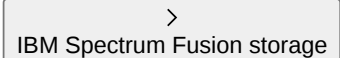
```
cpd-cli manage login-to-ocp \
--username=${OCP_USERNAME} \
--password=${OCP_PASSWORD} \
--server=${OCP_URL}
```

Tip: The `login-to-ocp` command takes the same input as the `oc login` command. Run `oc login --help` for details.

2. Create the custom resources for the specified components.
The command that you run depends on the storage on your cluster:

-  Red Hat OpenShift Data Foundation
Create the custom resources for the specified components:

```
cpd-cli manage apply-cr \
--components=${COMPONENTS} \
--release=${VERSION} \
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \
--block_storage_class=${STG_CLASS_BLOCK} \
--file_storage_class=${STG_CLASS_FILE} \
--license_acceptance=true
```

-  IBM Spectrum Fusion storage
Create the custom resources for the specified components.

Remember: When you use IBM Spectrum Fusion storage, both `${STG_CLASS_BLOCK}` and `${STG_CLASS_FILE}` point to the same storage class, typically `ibm-spectrum-scale-sc`.

```
cpd-cli manage apply-cr \
--components=${COMPONENTS} \
--release=${VERSION} \
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \
--block_storage_class=${STG_CLASS_BLOCK} \
```

```
--file_storage_class=${STG_CLASS_FILE} \  
--license_acceptance=true
```

- >
IBM Spectrum Scale Container Native storage

Create the custom resources for the specified components.

Remember: When you use IBM Spectrum Scale Container Native storage, both `${STG_CLASS_BLOCK}` and `${STG_CLASS_FILE}` point to the same storage class, typically `ibm-spectrum-scale-sc`.

```
cpd-cli manage apply-cr \  
--components=${COMPONENTS} \  
--release=${VERSION} \  
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \  
--block_storage_class=${STG_CLASS_BLOCK} \  
--file_storage_class=${STG_CLASS_FILE} \  
--license_acceptance=true
```

- >
Portworx storage

Create the custom resources for the specified components.

```
cpd-cli manage apply-cr \  
--components=${COMPONENTS} \  
--release=${VERSION} \  
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \  
--storage_vendor=portworx \  
--license_acceptance=true
```

- >
NFS storage

Create the custom resources for the specified components.

Remember: When you use NFS storage, both `${STG_CLASS_BLOCK}` and `${STG_CLASS_FILE}` point to the same storage class, typically `managed-nfs-storage`.

```
cpd-cli manage apply-cr \  
--components=${COMPONENTS} \  
--release=${VERSION} \  
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \  
--block_storage_class=${STG_CLASS_BLOCK} \  
--file_storage_class=${STG_CLASS_FILE} \  
--license_acceptance=true
```

- >
AWS with EFS storage only

Create the custom resources for the specified components.

Remember: When you use only EFS storage, both `${STG_CLASS_BLOCK}` and `${STG_CLASS_FILE}` point to the same RWX storage class.

```
cpd-cli manage apply-cr \  
--components=${COMPONENTS} \  
--release=${VERSION} \  
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \  
--block_storage_class=${STG_CLASS_BLOCK} \  
--file_storage_class=${STG_CLASS_FILE} \  
--license_acceptance=true
```

- >
AWS with EFS and EBS storage

Create the custom resources for the specified components:

```
cpd-cli manage apply-cr \  
--components=${COMPONENTS} \  
--release=${VERSION} \  
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \  
--block_storage_class=${STG_CLASS_BLOCK} \  
--file_storage_class=${STG_CLASS_FILE} \  
--license_acceptance=true
```

```
--file_storage_class=${STG_CLASS_FILE} \  
--license_acceptance=true
```

- [>](#)
IBM Cloud with IBM Cloud File Storage only

Create the custom resources for the specified components.

Remember: When you use only IBM Cloud File Storage storage, both `${STG_CLASS_BLOCK}` and `${STG_CLASS_FILE}` point to the same storage class, typically `ibmc-file-gold-gid` or `ibm-file-custom-gold-gid`.

```
cpd-cli manage apply-cr \  
--components=${COMPONENTS} \  
--release=${VERSION} \  
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \  
--block_storage_class=${STG_CLASS_BLOCK} \  
--file_storage_class=${STG_CLASS_FILE} \  
--license_acceptance=true
```

- [>](#)
IBM Cloud with IBM Cloud File Storage and IBM Cloud Block Storage

Create the custom resources for the specified components:

```
cpd-cli manage apply-cr \  
--components=${COMPONENTS} \  
--release=${VERSION} \  
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \  
--block_storage_class=${STG_CLASS_BLOCK} \  
--file_storage_class=${STG_CLASS_FILE} \  
--license_acceptance=true
```

Tip: If you want to preview the `oc` commands that the `cpd-cli manage`

`apply-cr` will issue on your behalf, you can run the command with `--preview=true`.

The `oc` commands are saved to the `preview.sh` file in the `cpd-cli-workspace/olm-utils-workspace/work` directory.

Results

The specified components are installed in the project specified by `${PROJECT_CPD_INSTANCE}`.

The software is installed when the `apply-cr` command returns `[SUCCESS]...` The `apply-cr` command ran successfully.

However, you can optionally run the [cpd-cli manage](#)

[get-cr-status](#) command to get the status of the components that are installed in the specified project (namespace):

```
cpd-cli manage get-cr-status \  
--cpd_instance_ns=${PROJECT_CPD_INSTANCE}
```

For details about the values that are returned by the command, see [Getting the status of installed components](#).

What to do next

Now that you've installed the Cloud Pak for Data software, you are ready to complete [Post-installation setup \(Day 1 operations\)](#).

Previous topic: [Creating OLM objects for an express installation](#)

Specialized installations

In a specialized installation, the IBM Cloud Pak® foundational services operators are installed in one project (namespace) and the IBM® Cloud Pak for Data operators are installed in another project.


The operators in both projects are granted permission to watch the project or projects where the Cloud Pak for Data platform and services are installed. However, IBM Cloud Pak foundational services operators project can be granted permission to watch additional projects without granting additional permissions to the Cloud Pak for Data operators.

Installation phase

Setting up a client workstation

Collecting required information

Preparing your cluster

 Installing the Cloud Pak for Data platform and services

Before you begin

Ensure that the following tasks are complete before you install the platform and services:

- You set up a [client workstation](#) from which you will run the installation.
- You collected the [required information](#) that you will need to complete the installation.
- A cluster administrator [prepared the cluster](#) for the installation.
Important: If you are using a private container registry, the images must be in the private container registry before you run the installation.

Ensure that you have the required permissions:

- You must have sufficient privileges to install operators on the cluster.
If you don't have sufficient privileges to install operators, you will need to work with a cluster administrator to complete specific tasks.
- You must be an administrator of the projects where you need to install the operators and the software.

About this task

To perform a specialized installation, complete the following tasks:

1. [Creating OLM objects for a specialized installation](#)

A cluster administrator, or a user with the appropriate permissions to install operators, must create the Operator Lifecycle Manager (OLM) objects, such as operators and operator subscriptions, that are required to install the IBM Cloud Pak for Data platform and services. The OLM objects that you create depend on the services that you plan to install.

2. [Installing components in a specialized installation](#)

A project administrator can create custom resources to install the IBM Cloud Pak for Data platform and services in a project (namespace). The custom resources that you create depend on the services that you plan to install.

Creating OLM objects for a specialized installation


A cluster administrator, or a user with the appropriate permissions to install operators, must create the Operator Lifecycle Manager (OLM) objects, such as operators and operator subscriptions, that are required to install the IBM® Cloud Pak for Data platform and services. The OLM objects that you create depend on the services that you plan to install.

Installation phase

Setting up a client workstation

Collecting required information

Preparing your cluster

 Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

A user with the appropriate permissions to install operators must complete this task.

When do you need to complete this task?

You must complete this task before you install Cloud Pak for Data for the first time.

Before you begin

Best practice: You can run the commands in this task exactly as written if you set up environment variables. For instructions, see [Setting up installation environment variables](#).

Ensure that you source the environment variables before you run the commands in this task.

About this task

Use the `cpd-cli manage`

`apply-olm` command to create the IBM Cloud Pak® foundational services operators and the Cloud Pak for Data operators.

The OLM objects are created in two projects:

- The IBM Cloud Pak foundational services operators are installed in the project specified by the `${PROJECT_CPFS_OPS}` environment variable (typically the `ibm-common-services` project). If you install the scheduling service, the operator for this component is also installed in the `ibm-common-services` project.
- The Cloud Pak for Data operators are installed in the project specified by the `${PROJECT_CPD_OPS}` environment variable.

The instructions assume that you are installing the operators for all of the components at the same time, which enables you to complete the task in fewer steps. You can always install additional operators if you decide to install additional services on your environment.

Remember: If the `cpd-cli` detects that you already have the OLM objects for the components at the specified release, the `cpd-cli` does not attempt to create the OLM objects again.

Procedure

1. Run the `cpd-cli manage`

`login-to-ocp` command to log in to the cluster as a user with sufficient permissions to complete this task. For example:

```
cpd-cli manage login-to-ocp \  
--username=${OCP_USERNAME} \  
--password=${OCP_PASSWORD} \  
--server=${OCP_URL}
```

Tip: The `login-to-ocp` command takes the same input as the `oc login` command. Run `oc login --help` for details.

2. Create the OLM objects for the specified components:

- `>`
IBM Cloud Pak foundational services are in the default project (`ibm-common-services`)

Remember: If IBM Cloud Pak foundational services is not installed, this command will automatically create the required OLM objects in the `ibm-common-services` project.

```
cpd-cli manage apply-olm \  
--release=${VERSION} \  
--components=${COMPONENTS} \  
--cpd_operator_ns=${PROJECT_CPD_OPS}
```

- `>`
IBM Cloud Pak foundational services are in a different project

Remember: If IBM Cloud Pak foundational services is not installed, this command will automatically create the required OLM objects in the project that you specify for the `--cs_ns` option.

```
cpd-cli manage apply-olm \  
--release=${VERSION} \  
--cs_ns=${PROJECT_CPD_OPS}
```

```
--components=${COMPONENTS} \  
--cs_ns=${PROJECT_CPFS_OPS} \  
--cpd_operator_ns=${PROJECT_CPD_OPS}
```

Tip: If you want to preview the `oc` commands that the `cpd-cli manage`

`apply-olm` will issue on your behalf, you can run the command with `--preview=true`.

The `oc` commands are saved to the `preview.sh` file in the `cpd-cli-workspace/olm-utils-workspace/work` directory.

3. Run the following command to ensure that any services that have a dependency on third-party operators can automatically set the namespace scope on the third-party operators:

```
oc patch NamespaceScope cpd-operators \  
-n ${PROJECT_CPD_OPS} \  
--type=merge \  
--patch='{"spec": {"csvInjector": {"enable": true} } }'
```

Results

The operators for the specified components are created in the appropriate projects:

- IBM Cloud Pak foundational services operators are created in `ibm-common-services` or the project specified by the `${PROJECT_CPFS_OPS}` environment variable.
- The Cloud Pak for Data operators are created in the project specified by the `${PROJECT_CPD_OPS}` environment variable.

The operators are created when the `apply-olm` command returns `[SUCCESS]...` The `apply-olm` command ran successfully.

You can optionally run the `cpd-cli manage`

`get-olm-artifacts` command to get the list of catalog sources and operator subscriptions that are on the cluster.

- To see the catalog sources and operator subscriptions for the operators in the `${PROJECT_CPFS_OPS}` project, run:

```
cpd-cli manage get-olm-artifacts \  
--subscription_ns=${PROJECT_CPFS_OPS}
```

- To see the catalog sources and operator subscriptions for the operators in the `${PROJECT_CPD_OPS}` project, run:

```
cpd-cli manage get-olm-artifacts \  
--subscription_ns=${PROJECT_CPD_OPS}
```

What to do next

Now that you've created the OLM objects, you are ready to complete [Installing components in a specialized installation](#).

Next topic: [Installing components in a specialized installation](#)

Installing components in a specialized installation

A project administrator can create custom resources to install the IBM® Cloud Pak for Data platform and services in a project (namespace). The custom resources that you create depend on the services that you plan to install.

Installation phase

Setting up a client workstation

Collecting required information

Preparing your cluster



Installing the Cloud Pak for Data platform and services

Who needs to complete this task?

A project administrator must complete this task.
When do you need to complete this task?

You must complete this task at least once to install Cloud Pak for Data.

You can complete this task multiple times if you want to install Cloud Pak for Data in multiple projects (namespaces).

Before you begin

Best practice: You can run the commands in this task exactly as written if you set up environment variables. For instructions, see [Setting up installation environment variables](#).

Ensure that you source the environment variables before you run the commands in this task.

About this task

Use the `cpd-cli manage apply-cr` command to create the custom resources for the specified components.

The instructions assume that you are installing all of the components at the same time, which enables you to complete the task in fewer steps. You can always install additional services separately.

If you want to install a service in a tethered project, install the service separately.

Remember: If the `cpd-cli` detects that you already have a component installed at the specified release, the `cpd-cli` does not attempt to install the component again.

Procedure

1. Run the `cpd-cli manage`

`login-to-ocp` command to log in to the cluster as a user with sufficient permissions to complete this task. For example:

```
cpd-cli manage login-to-ocp \
--username=${OCP_USERNAME} \
--password=${OCP_PASSWORD} \
--server=${OCP_URL}
```

Tip: The `login-to-ocp` command takes the same input as the `oc login` command. Run `oc login --help` for details.

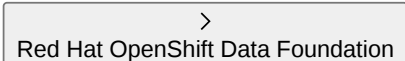
2. Enable the project (namespace) where the Cloud Pak for Data operators are installed to watch the project where you plan to install the control plane and services.

The following command creates or updates the `cpd-operators NamespaceScope` object in the project where the Cloud Pak for Data operators are installed. The command adds the name of the project where you plan to install the control plane and services to the `namespaceMembers` list.

```
cpd-cli manage setup-instance-ns \
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \
--cpd_operator_ns=${PROJECT_CPD_OPS}
```

3. Create the custom resources for the specified components.

The command that you run depends on the storage on your cluster:

- 

Create the custom resources for the specified components:

```
cpd-cli manage apply-cr \
--components=${COMPONENTS} \
--release=${VERSION} \
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \
--block_storage_class=${STG_CLASS_BLOCK} \
--file_storage_class=${STG_CLASS_FILE} \
--license_acceptance=true
```


- >
IBM Spectrum Fusion storage

Create the custom resources for the specified components.

Remember: When you use IBM Spectrum Fusion storage, both `${STG_CLASS_BLOCK}` and `${STG_CLASS_FILE}` point to the same storage class, typically `ibm-spectrum-scale-sc`.

```
cpd-cli manage apply-cr \  
--components=${COMPONENTS} \  
--release=${VERSION} \  
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \  
--block_storage_class=${STG_CLASS_BLOCK} \  
--file_storage_class=${STG_CLASS_FILE} \  
--license_acceptance=true
```

- >
IBM Spectrum Scale Container Native storage

Create the custom resources for the specified components.

Remember: When you use IBM Spectrum Scale Container Native storage, both `${STG_CLASS_BLOCK}` and `${STG_CLASS_FILE}` point to the same storage class, typically `ibm-spectrum-scale-sc`.

```
cpd-cli manage apply-cr \  
--components=${COMPONENTS} \  
--release=${VERSION} \  
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \  
--block_storage_class=${STG_CLASS_BLOCK} \  
--file_storage_class=${STG_CLASS_FILE} \  
--license_acceptance=true
```

- >
Portworx storage

Create the custom resources for the specified components.

```
cpd-cli manage apply-cr \  
--components=${COMPONENTS} \  
--release=${VERSION} \  
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \  
--storage_vendor=portworx \  
--license_acceptance=true
```

- >
NFS storage

Create the custom resources for the specified components.

Remember: When you use NFS storage, both `${STG_CLASS_BLOCK}` and `${STG_CLASS_FILE}` point to the same storage class, typically `managed-nfs-storage`.

```
cpd-cli manage apply-cr \  
--components=${COMPONENTS} \  
--release=${VERSION} \  
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \  
--block_storage_class=${STG_CLASS_BLOCK} \  
--file_storage_class=${STG_CLASS_FILE} \  
--license_acceptance=true
```

- >
AWS with EFS storage only

Create the custom resources for the specified components.

Remember: When you use only EFS storage, both `${STG_CLASS_BLOCK}` and `${STG_CLASS_FILE}` point to the same RWX storage class.

```
cpd-cli manage apply-cr \  
--components=${COMPONENTS} \  
--release=${VERSION} \  
--cpd_instance_ns=${PROJECT_CPD_INSTANCE} \  
--block_storage_class=${STG_CLASS_BLOCK} \  
--file_storage_class=${STG_CLASS_FILE}
```


For details about the values that are returned by the command, see [Getting the status of installed components](#).

What to do next

If you choose to

Previous topic: [Creating OLM objects for a specialized installation](#)

Uninstalling the platform and services

If you need to uninstall the IBM® Cloud Pak for Data platform and services, you can remove the custom resources and the Operator Lifecycle Manager (OLM) objects that are associated with the components.

Who needs to complete this task?

A user with the appropriate permissions to manage projects must complete this task.

When do you need to complete this task?

Complete this task only if you want to uninstall IBM Cloud Pak for Data.

Before you begin

Important: Ensure that you delete any service instances that are associated with Cloud Pak for Data services before you uninstall the platform.

About this task

To uninstall the platform and services, complete the following tasks:

1. [Uninstalling the components](#)

If you want to uninstall IBM Cloud Pak for Data, you must uninstall the running instance of the control plane and services.

2. [Uninstalling the OLM objects](#)

If you want to completely remove the IBM Cloud Pak for Data software from your cluster, you must remove the Operator Lifecycle Manager (OLM) objects, such as operators, operator subscriptions, catalog sources, and cluster service versions.

Related tasks

- [Collecting required information](#)
- [Installing the IBM Cloud Pak for Data platform and services](#)

Related reference

- [Setting up a client workstation](#)
- [Preparing your cluster](#)

Uninstalling the components

If you want to uninstall IBM® Cloud Pak for Data, you must uninstall the running instance of the control plane and services.

Who needs to complete this task?

A user with the appropriate permissions to manage projects must complete this task.

When do you need to complete this task?

Complete this task only if you want to uninstall IBM 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.

About this task

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.

Use the [cpd-cli manage delete-cr](#) command to remove the custom resources.

The instructions assume that you are removing all of the components at the same time, which enables you to complete the task in fewer steps.

Procedure

1. Run the `cpd-cli manage login-to-ocp` command to log in to the cluster as a user with sufficient permissions to complete this task. For example:

```
cpd-cli manage login-to-ocp \  
--username=${OCP_USERNAME} \  
--password=${OCP_PASSWORD} \  
--server=${OCP_URL}
```

Tip: The `login-to-ocp` command takes the same input as the `oc login` command. Run `oc login --help` for details.

2. Delete the custom resources for all of the components in the project.

```
cpd-cli manage delete-cr \  
--cpd_instance_ns=${PROJECT_CPD_INSTANCE}
```

Tip: If you want to preview the `oc` commands that the `cpd-cli manage delete-cr` will issue on your behalf, you can run the command with `--preview=true`. The `oc` commands are saved to the `preview.sh` file in the `cpd-cli-workspace/olm-utils-workspace/work` directory.

Results

The components are uninstalled from the `${PROJECT_CPD_INSTANCE}` project.

What to do next

If you want to completely remove the Cloud Pak for Data software from your cluster, you must complete [Uninstalling the OLM objects](#).

Next topic: [Uninstalling the OLM objects](#)

Uninstalling the OLM objects

If you want to completely remove the IBM® Cloud Pak for Data software from your cluster, you must remove the Operator Lifecycle Manager (OLM) objects, such as operators, operator subscriptions, catalog sources, and cluster service versions.

Who needs to complete this task?

A user with the appropriate permissions to manage OLM objects must complete this task.

When do you need to complete this task?

Complete this task only if you want to completely remove the IBM Cloud Pak for Data software from your cluster.

About this task

Use the [cpd-cli manage](#)

[delete-olm-artifacts](#) command to remove the following OLM objects for the specified components:

- Catalog sources
- Cluster service versions
- Operator subscriptions

The instructions assume that you are removing the OLM objects for all of the components at the same time, which enables you to complete the task in fewer steps.

Procedure

1. Run the `cpd-cli manage`


`login-to-ocp` command to log in to the cluster as a user with sufficient permissions to complete this task. For example:

```
cpd-cli manage login-to-ocp \  
--username=${OCP_USERNAME} \  
--password=${OCP_PASSWORD} \  
--server=${OCP_URL}
```

Tip: The `login-to-ocp` command takes the same input as the `oc login` command. Run `oc login --help` for details.

2. Delete the OLM objects for the specified components.

The command that you run depends on where the operators are installed:

-  Express installations (all of the operators are in the same project)

```
cpd-cli manage delete-olm-artifacts
```

Tip: If you want to preview the `oc` commands that the `cpd-cli manage delete-olm-artifacts` will issue on your behalf, you can run the command with `--preview=true`. The `oc` commands are saved to the `preview.sh` file in the `cpd-cli-workspace/olm-utils-workspace/work` directory.


-  Specialized installations (the operators are installed in two different projects)

```
cpd-cli manage delete-olm-artifacts \  
--cpd_operator_ns=${PROJECT_CPD_OPS}
```

Tip: If you want to preview the `oc` commands that the `cpd-cli manage delete-olm-artifacts` will issue on your behalf, you can run the command with `--preview=true`. The `oc` commands are saved to the `preview.sh` file in the `cpd-cli-workspace/olm-utils-workspace/work` directory.

Results

The OLM objects are uninstalled from the appropriate projects:

-  Express installations

The operators, catalog sources, and operator subscriptions are removed from the `ibm-common-services` project.

- >
Specialized installations

- The Cloud Pak for Data operators, catalog sources, and operator subscriptions are removed from the `PROJECT_CPDP_OPS` project.
- The IBM Cloud Pak® foundational services operators, catalog sources, and operator subscriptions are removed from the `ibm-common-services` project.
If you installed the scheduling service, the operators, catalog sources, and operator subscriptions for this component is also removed from the `ibm-common-services` project.

Previous topic: [Uninstalling the components](#)