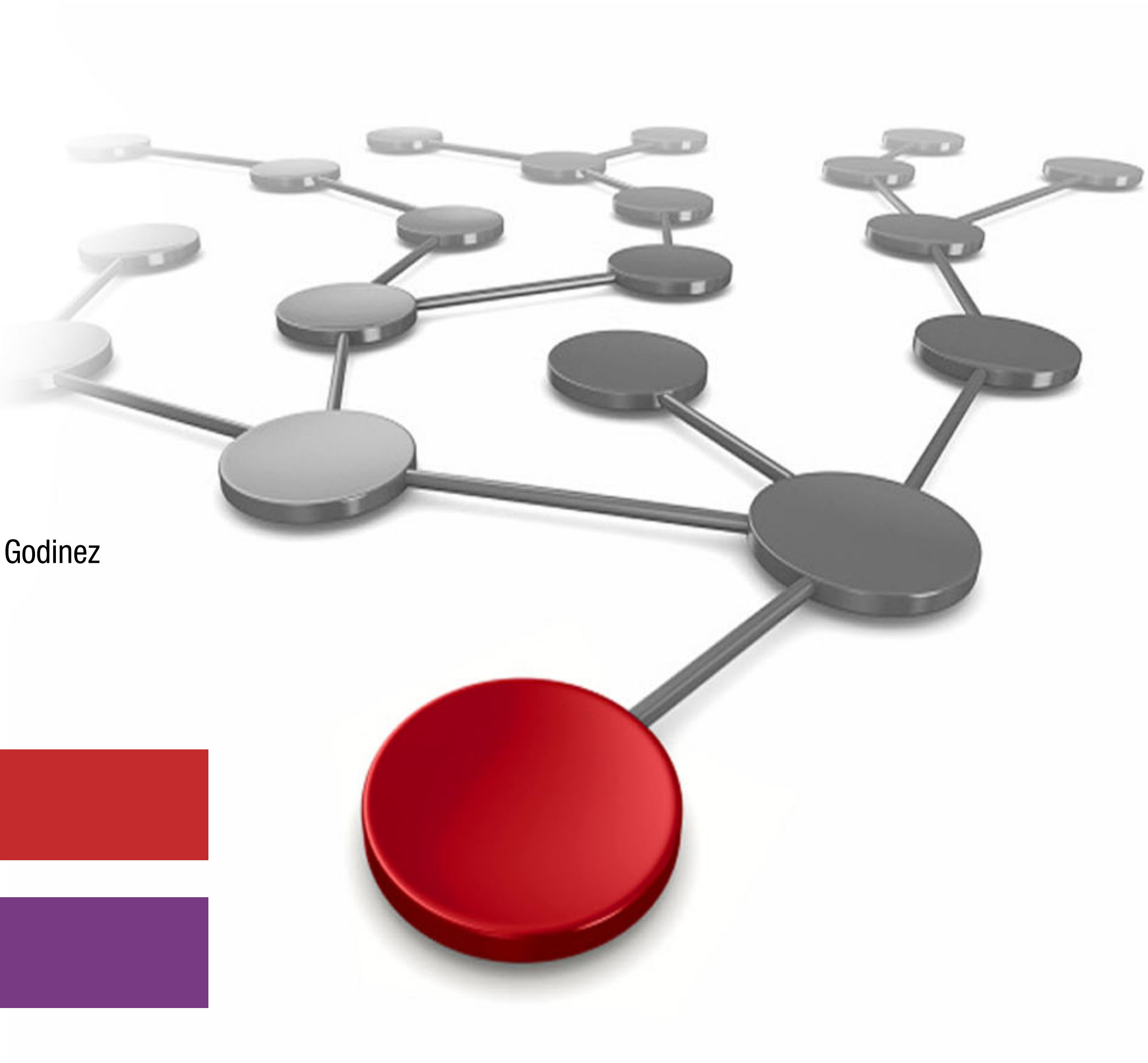
Redbooks
ibm.com/redbooks

IBM Cloud Pak for Data with IBM Spectrum Scale Container Native

Gero Schmidt
Tara Astigarraga
Paulina Acevedo
JJ Miller
Dessa Simpson
Austen Stewart
Todd Tosseth
Jayson Tsingine
Israel Andres Vizcarra Godinez



Cloud

Storage

IBM

Redpaper

IBM Redbooks

IBM Cloud Pak for Data with IBM Spectrum Scale Container Native

December 2021

REDP-5652-00

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (December 2021)

This edition applies to Version 5, Release 1, Modification 0.3 of IBM Spectrum Scale, Version 3, Release 5, Modification 2 of IBM Cloud Pak for Data, and Version 4, Release 6 of Red Hat OpenShift Container Platform

This document was created or updated on December 17, 2021.

Contents

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

Cognos®	IBM®	IBM Z®
DataStage®	IBM Cloud®	POWER®
Db2®	IBM Cloud Pak®	Redbooks®
DB2®	IBM Elastic Storage®	Redbooks (logo) ®
DS8000®	IBM Spectrum®	SPSS®
Guardium®	IBM Watson®	XIV®

The following terms are trademarks of other companies:

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

OpenShift, Red Hat, are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

VMware, and the VMware logo are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Other company, product, or service names may be trademarks or service marks of others.

Preface

This IBM® Redpaper® publication describes configuration guidelines and best practices when IBM Spectrum® Scale Container Native Storage Access is used as a storage provider for IBM Cloud® Pak for Data on Red Hat OpenShift Container Platform.

It also provides the steps to install IBM Db2® and several assemblies within IBM Cloud Pak® for Data, including Watson Knowledge Catalog, Watson Studio, IBM DataStage®, Db2 Warehouse, Watson Machine Learning, Watson OpenScale, Data Virtualization, Data Management Console, and Apache Spark.

This IBM Redpaper publication was written for IT architects, IT specialists, developers, and others who are interested in installing IBM Cloud Pak for Data with IBM Spectrum Scale Container Native.

Authors

This paper was produced by a team of specialists from around the world working at IBM Redbooks, Tucson Center.

Gero Schmidt is a Software Engineer at IBM Germany Research and Development GmbH in the IBM Spectrum Scale development group. He joined IBM in 2001 by working at the European Storage Competence Center in Mainz, Germany to provide technical presales support for a broad range of IBM storage products, with a primary focus on enterprise storage solutions, system performance, and IBM Power Systems servers. Gero participated in the product roll-out of the IBM System Storage DS6000 and DS8000® series, co-authored several IBM Redbooks publications, and was a frequent speaker at IBM international conferences. In 2015, he joined the storage research group at the IBM Almaden Research Center in California, US, where he worked on IBM Spectrum Scale, compression of genomic data in next generation sequencing pipelines, and the development of a cloud-native backup solution for containerized applications in Kubernetes and Red Hat OpenShift. In 2018, he joined the Big Data & Analytics team in the IBM Spectrum Scale development group. He holds a degree in Physics (Dipl.-Phys.) from the Braunschweig University of Technology in Germany.

Tara Astigarraga is a Senior Technical Staff Member (STSM) and Lead Architect for Cloud Pak Storage Test at IBM. She is an IBM Master Inventor with over 75 patents issued and a passionate mentor for STEM K-12 outreach. Tara joined IBM in 2001 and spent the first 10 years of her career in IBM Systems group focused on Enterprise Storage, Server, and Networking Solutions. Then, she expanded her expertise through a one-year assignment in the Corporate technology team and five years in Watson Research as a solution Architect and Developer, building Enterprise Hybrid Cloud, Blockchain, and AI Solutions for Technology Support Services. She returned to the Systems group in early 2020 and focused on IBM storage and server solutions to enable enterprise Hybrid Cloud and AI workloads.

Paulina Acevedo is one of two System Test architects for the Cloud Pak Storage Test team. Paulina has been with IBM for more than 15 years and has held several different positions within the Systems organization. She is a certified Project Manager and has been the System Test Project manager for several products including IBM XIV®, A9000, Spectrum NAS, and Spectrum Virtualize.

JJ Miller is a Storage Engineer for IBM Cloud Pak Storage Test team. Before this position, he worked on conducting testing and software development for IBM Storage DS8000 Field Support Test team. JJ has extensive background in Storage Backup and Recovery solutions with business partner AT&T, using various backup software products, such as TSM, Netbackup, Avamar, and IBM Spectrum Protect along with multiple storage systems.

Dessa Simpson is a storage test engineer at IBM on the Cloud Pak Storage Test team. Since joining IBM in 2018, she has worked in storage support for the XIV and A9000 storage systems diagnosing customer problems and writing patch scripts to work around issues found in the field. She later moved into storage interop test in the Open Systems Lab, before joining the Cloud Pak Storage Test team because of her interest in OpenShift and containerization.

Austen Stewart is a computer programmer and tester from Tucson, Arizona. He joined IBM as a co-op in 2020, working on DS8000 storage testing automation. After returning full-time in 2021, Austen began working on storage testing for Cloud Paks. He holds a BS degree in Computer Science from University of Arizona.

Todd Tosseth is a Software Engineer for IBM in Tucson, Arizona. Joining IBM in 2001, he has worked as a test and development engineer on several IBM storage products, such as IBM System Storage DS8000, IBM Spectrum Scale, and IBM Enterprise Storage Server. He is working on IBM Cloud Pak as a system test engineer, with an emphasis on Cloud Pak storage integration.

Jayson Tsingine is an Advisory Software Engineer in the IBM Systems Storage Group based in Tucson, Arizona. Jayson has worked in numerous test roles since he joined IBM in 2003, providing test support for storage products, including XIV, DS8000, FlashSystems, Hydra, Spectrum Scale, Spectrum NAS, and Cloud Pak Solutions. He holds a BS degree in Computer Science from the University of Arizona.

Israel Andres Vizcarra Godinez is a Mechatronics Engineer at Cloud Pak Storage Test team in IBM Systems organization. He joined IBM in 2015, working as a Test Specialist for the TS7700 System Storage Virtualization Engine product at Guadalajara, Mexico. He has participated in different phases of the test cycle from integration to system level test being closely related to customer requirements. Israel has developed and run manual and automation scenarios to test the stability of the code for IBM storage systems and SAN area. Currently, Israel is the automation test lead for Spectrum Scale integration with the different IBM Cloud Paks.

Thanks to the following people for their contributions to this project:

Larry Coyne
IBM Redbooks, Tucson Center

Claus Huempel
Matt LeVan
Aaron Palazzolo
Deepak Rangarao
Sundari Voruganti
Andre Tost
Deepak Ghuge
Tom Clark
Alasdair Paton
Sriram Srinivasan
Dany Drouin
Tony Sinclair
Sudarsha Wijenayake
IBM Systems

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an IBM Redbooks residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, IBM Redbooks
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

IBM Cloud Pak for Data introduction

This chapter provides an overview of IBM Cloud Pak for Data and how it works with IBM Spectrum Scale Container Native, along with the hardware and software requirements.

This chapter includes the following topics:

1.1 Overview

IBM Cloud Pak for Data is an open data and AI platform that integrates IBM Watson® AI technology with IBM Hybrid Data Management Platform, DataOps, and business analytics to provide data virtualization and container management. This offering is all about solving problems, helping customers become agile, and providing real competitive advantages for their businesses.

IBM Storage Suite for IBM Cloud Paks provides enterprise data services to container environments with flexible software-defined storage solutions for hybrid cloud. IBM Spectrum Scale Container Native, which is part of the IBM Storage Suite for IBM Cloud Paks, enables a fast, reliable way to modernize and move to the cloud. IBM Storage Suite for Cloud Paks is designed to simplify IBM Cloud Pak solutions setup with a storage layer for persistent data.

Similarly, IBM Spectrum Scale Container Native provides simplicity for applications and the global parallel file system that makes hybrid cloud easier to manage because localized disks are not limited and performance is optimized by external storage resources.

Today, most businesses construct their IT environments with a mix of on-premises private clouds and s public clouds and their existing platforms. As a result, businesses need a hybrid cloud strategy that includes data for current applications and workloads in addition to new modern applications and platforms.

IBM Spectrum Scale Native provides essential benefits to developers and administrators by enabling the deployment of storage inside containers and applications that are running within containers.

This publication describes configuration guidelines and best practices for using IBM Spectrum Scale Container Native Storage Access (CNSA) with IBM Spectrum Scale Container Storage Interface (CSI) driver as storage provider for IBM Cloud Pak for Data.

The document also describes the required steps to install IBM Cloud Pak for Data by using a storage class (SC) backed by IBM Spectrum Scale CNSA. Additionally, this document provides the steps to install multiple assemblies within IBM Cloud Pak for Data.

This IBM Redpaper publication was written for IT architects, IT specialists, developers, and others who are interested in deploying Cloud Pak for Data in the environment described.

IBM Cloud Pak for Data is a container-based software package that can be deployed on Red Hat OpenShift Container Platform. It relies on a storage provider in OpenShift to provide persistent storage to all its components and core modules. IBM Cloud Pak for Data v3.5.2 supports and is optimized for the following storage providers:

- Red Hat OpenShift Container Storage, Version 4.5 or later fixes
- Network File System (NFS), Version 4
- Portworx, Version 2.6.2 or later is required for Red Hat OpenShift Version 4.5 and 4.6
- IBM Cloud File Storage
- IBM Spectrum Scale Container Native, Version 5.1.0.3 or later

A basic setup is shown in Figure 1-1.

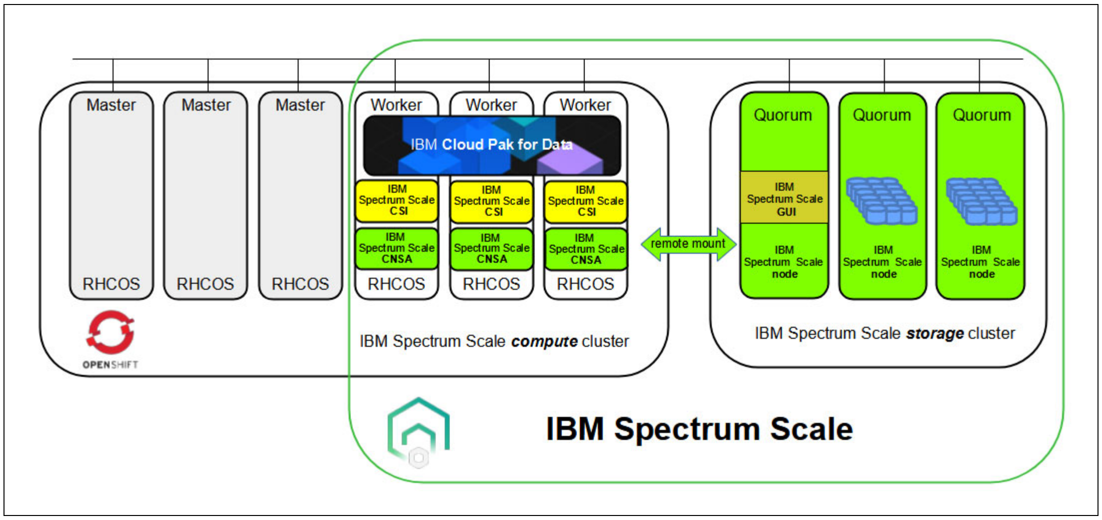


Figure 1-1 Architecture

Figure 1-1 shows an x86 based Red Hat OpenShift 4.6 cluster with a minimum configuration of three master and three worker nodes that are running IBM Spectrum Scale CNSA and IBM Spectrum Scale CSI driver, plus a remote IBM Spectrum Scale storage cluster that provides the shared persistent storage.

The hardware and software releases that are used in this document are listed.

- Software:
  - Container orchestration platform: Red Hat OpenShift 4.6.42
  - Container storage provider:
    - IBM Spectrum Scale Container Native Storage Access 5.1.0.3
    - IBM Spectrum Scale Container Storage Interface Driver 2.1.0
  - IBM Cloud Pak for Data: IBM Cloud Pak for Data v3.5.2
  - IBM Spectrum Scale (remote storage cluster): IBM Spectrum Scale 5.1.0.3
- Hardware:
  - OpenShift cluster with three master and five worker nodes (x86-64)

    Lenovo SR650, which consists of: 2x Intel Xeon Silver 4110 CPU @ 2.10GHz (32 cores), 64GB of memory, 600GB HDD, Red Hat Core OS 4.6
  - Load balancer/infrastructure node (HAproxy, DNSmasq, DHCP/TFT, NFS)

    Lenovo x3650 M5 server (5462-AC1), which consists of: 2x Intel Xeon CPU E5-2650 v3 @ 2.30GHz (40 cores), 128 GB of memory, 600GB HDD, RHEL 8.3
  - Network: 100 Gbps Ethernet (Note: a minimum of a 10 Gbps network is typically required)

– Remote IBM Spectrum Scale storage clusters in our environment:
  - IBM Elastic Storage® System (ESS) 3000 with an added I EMS/GUI node
  - IBM Spectrum Scale Storage Cluster 5.1.0.3 (Advanced Edition and Data Management Edition) with three x3650 nodes on RHEL 7.9 and the following back-end storage:

    - DS8910F 5331-994 (9.1)

    - V5100: 2077-424 (8.4.0.2)

Table 1-1 shows the IBM Cloud Pak for Data support and requirement details for IBM Spectrum Scale Container Native.

Table 1-1 IBM Cloud Pak for Data support and requirement details

Details[a]	**IBM Spectrum Scale Container Native**
Deployment environments	On-premises deployments on VMware or bare metal. For more information, see the IBM Spectrum Scale Container Native requirements.
Red Hat OpenShift 3.11	Not supported.
Red Hat OpenShift 4.5 and 4.6	Supported on 4.6 only. Requires 4.6.6 or later fixes. IBM Spectrum Scale Container Native adheres to the Red Hat OpenShift lifecycle.
x86-64	Supported.
IBM POWER®	Not supported.
IBM Z®	Not supported.
License requirements	A separate licenses is not required for IBM Spectrum Scale Container Native. You can use up to 12 TB of IBM Spectrum Scale Container Native, fully supported by IBM in production environments (Level 1 and Level 2), for up to 36 months. If the terms are exceeded, a separate license is required.
Storage classes	`ibm-spectrum-scale-sc`
Data replication for high availability	Replication is supported and can be enabled on the IBM Spectrum Scale Storage Cluster in various ways. For more information, see Data mirroring and replication.
Backup and restore	Use the IBM Spectrum Scale Container Storage Interface Volume snapshot as the primary backup and restore method. Combine volume snapshots with Container Backup Support that is provided by IBM Spectrum Protect Plus. Also, many methods can be use to back up the Spectrum Scale Storage Cluster. For more information, see Data protection and disaster recovery.
Encryption of data at rest	Supported For more information, see Encryption in the IBM Spectrum Scale documentation.
Network requirements	Sufficient network performance must exist to meet the storage I/O requirements.
I/O requirements	For more information, see Disk requirements in the system requirements.
Minimum amount of storage	1 TB or more of available space.
Minimum amount of vCPU	8 vCPU on each worker node. For more information, see the IBM Spectrum Scale Container Native requirements.
Minimum amount of memory	16 GB of RAM on each worker node. For more information, see the IBM Spectrum Scale Container Native requirements.
Installation documentation	For IBM Spectrum Scale and IBM Spectrum Scale Container Storage Interface, see the IBM Spectrum Scale Container Native installation documentation.

Details[a]	IBM Spectrum Scale Container Native
Troubleshooting documentation	Refer to the following documentation for your environment: ► IBM Spectrum Scale Container Native documentation ► IBM Spectrum Scale Container Storage Interface documentation

a. Source: IBM Cloud Pak for Data V3.5.0 IBM Documentation. For more information about other storage considerations, see this IBM Documentation web page.

1.2 IBM Cloud Pak for Data

IBM Cloud Pak for Data is a container-based software package that can be deployed on OpenShift Container Platform. It relies on a storage provider in OpenShift to provide persistent storage to all its components and core modules.

This document describes configuration guidelines and best practices for the use of IBM Spectrum Scale Container Native Storage Access (CNSA) with the IBM Spectrum Scale CSI driver as storage provider for IBM Cloud Pak for Data.

It is important to understand that on Kubernetes/OpenShift container orchestration platforms, the storage layer is fully abstracted from a user through persistent volumes (PVs) and persistent volume claims (PVCs). A container in a pod (the smallest schedulable compute unit on a Kubernetes/OpenShift cluster) can request storage with a PVC that is bound to a PV through a storage provider in the cluster. A storage class (SC), which is another Kubernetes/OpenShift object, allows to provide storage (that is, create and bind to a request for storage) on-demand.

1.2.1 Installing IBM Cloud Pak for Data

The IBM Cloud Pak for Data installation and run time relies on a storage class that is specified as flag (`-c/--storageclass [storageclass name]`) at the time of deployment to use dynamic provisioning and satisfy all storage requests of its components automatically. All required PVs automatically are created and bound to the related PVCs of these components on-demand by the CSI driver of the storage provider.

The IBM Spectrum Scale CSI driver supports three options to create a storage class: with volumes that are backed by simple directories (light-weight), independent filesets, or dependent filesets in IBM Spectrum Scale. Here, we use a storage class with independent filesets to back the PVs because this selection allows for most options and supports snapshot functions when OCP 4.7 or later releases are used.

IBM Cloud Pak for Data specifies minimum disk performance requirements for the configured storage provider as described in 3.7.4, “Disk requirements” on page 50 Disk requirements and provides simple **dd** commands to determine latency and throughput, which easily can be run in a Kubernetes pod. The minimum throughput is specified with 209 MBps. For remote storage, such as IBM Spectrum Scale CNSA with a remote cross-cluster mounted file system, this configuration mandates a minimum of a 10 Gbps network to meet the throughput requirements.

1.2.2 Deploying IBM Cloud Pak for Data

The deployment of IBM Cloud Pak for Data[1] includes the installation of the IBM Cloud Pak for Data control plane (`lite` assembly) as base and the individual installation of other services from the catalog, including the following assembly examples:

- IBM Db2 (`db2oltp`)
- Watson Studio (`wsl`)
- Watson Knowledge Catalog (`wkc`)
- DataStage (`ds`)

The installation for each of these services or components (also referred to as *assemblies*) is done manually by using the **`cpd-cli`** command line tool through a cluster preparation (`cpd-cli adm`) and installation (`cpd-cli install`) step. Each installation step allows to specify only a single storage class (`-c/--storageclass [storageclass name]`) to be used by this component (and its sub-components) throughout the entire deployment. However, a different storage class can be used for the installation of each component (or assembly).

By using an IBM Spectrum Scale storage class[2] that is based on independent filesets with default uid/gid settings (that is, `uid=0 [root] / gid=0 [root]`) proved to work well with the installation of the following IBM Cloud Pak for Data assemblies:

- Control plane (`lite`)
- Db2 service (`db2oltp`)
- Db2 Warehouse (`db2wh`)
- Data Management Console (`dmc`)
- Watson OpenScale (`aiopenscale`)
- Watson Studio service (`wsl`)
- Watson Machine Learning (`wml`)
- Data Virtualization (`dv`)

The installation of the following IBM Cloud Pak for Data assembly services required adjustments to the default settings of the storage class:

- Watson Knowledge Catalog (`wkc`)
- DataStage (`ds`)

By using an IBM Spectrum Scale storage class[3] with a dedicated uid (here, it is uid: 10032 to meet the requirements of the iis sub-component) allows to successfully install Watson Knowledge Center and DataStage services.

---
[1] See IBM Cloud Pak for Data, v3.5, Storage considerations
[2] ibm-spectrum-scale-sc, see “IBM Spectrum Scale Storage Class with default uid and gid settings” on page 37
[3] ibm-spectrum-scale3-sc, see “IBM Spectrum Scale Storage Class with dedicated uid setting” on page 39

1.3 Hardware and software requirements

A minimum Red Hat OpenShift Container Platform 4.6 cluster typically consists of at least five nodes, three control plane or master nodes, and at least two compute or worker nodes. For more information see this web page. For installing a new OpenShift cluster, another bootstrap node is required temporarily.

For more information about the hardware and software requirements (for example, physical worker node Memory and CPU requirements) for IBM Spectrum Scale CNSA, see this web page.

At minimum, three worker nodes with Red Hat OpenShift Container Platform 4.6.6 (or higher) are required. A minimum of 10 Gb network is required; however, 40 GbE - 100 GbE is recommended. RDMA for InfiniBand or RoCE for Ethernet is *not* supported.

Figure 1-2 shows an example of general IBM Spectrum Scale CNSA hardware and software requirements.

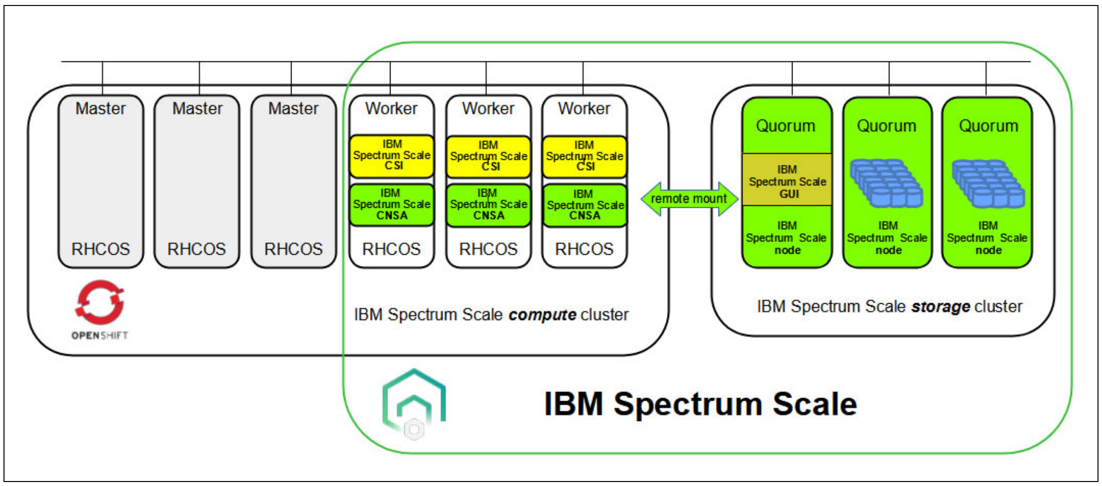


Figure 1-2 Example of general IBM Spectrum Scale CNSA hardware and software requirements

1.3.1 Remote storage cluster and IBM CNSA

IBM Spectrum Scale CNSA requires a remote mount from an additional IBM Spectrum Scale storage cluster (for example, an ESS), referred here to as the *remote storage cluster*. Also, all worker nodes in the OpenShift compute cluster must communicate with all nodes in the remote storage cluster.

Ideally, at least three nodes are available as contact nodes in the remote storage cluster for the remote mount. If no contact nodes are specified in the configuration of the CNSA Operator custom resource, the operator automatically chooses three nodes from the storage cluster (for more information, see this web page). Furthermore, a node that is running the IBM Spectrum Scale GUI is required for access to the REST interface in the remote storage cluster.

IBM Spectrum Scale CNSA relies on the IBM Spectrum Scale CSI driver to provide persistent storage in the form of PVs to Kubernetes or OpenShift applications through dynamic provisioning with SCs or static provisioning. A user requests and binds a PV to their namespace in OpenShift by using a PVC.

For more information about the hardware and software requirements for IBM Spectrum Scale CSI driver, see this web page. It requires IBM Spectrum Scale version 5.0.4.1 or later and Red Hat OpenShift 4.5 and 4.6 with Red Hat CoreOS based worker nodes when IBM Spectrum Scale CNSA is used. For more information, see this web page.

For IBM Cloud Pak for Data, we look primarily at dynamic provisioning with storage classes. The deployment of IBM Cloud Pak for Data requires only the name of a storage class to be used for providing persistent storage to all its components. In this publication, we provide this storage class through IBM Spectrum Scale CNSA with IBM Spectrum Scale CSI driver.

For more information about the supported storage providers for IBM Cloud Pak for Data 3.5, see this web page. For shared persistent storage, Cloud Pak for Data supports and is optimized for the following storage providers at the time of this writing:

- Red Hat OpenShift Container Storage, Version: 4.5 or later
- Network File System (NFS), Version: 4
- Portworx (Version 2.6.2 or later required for Red Hat OpenShift Version 4.5 and 4.6)
- IBM Cloud File Storage
- IBM Spectrum Scale Container Native, Version 5.1.0.3 or later

1.3.2 Minimum performance requirements

Also, IBM Cloud Pak for Data defines minimum performance requirements for the persistent storage provider in OpenShift, as described at this web page. IBM Cloud Pak for Data provides two simple tests that can be run from within a pod (Kubernetes Job YAML manifests and results provided later in this document) to ensure sufficient disk I/O performance:

- Disk latency: The value must be comparable to or better than 2.5 MBps:

  ```
  dd if=/dev/zero of=/PVC_mount_path/testfile bs=4096 count=1000 oflag=dsync
  ```

- Disk throughput: The value must be comparable to or better than 209 MBps:

  ```
  dd if=/dev/zero of=/PVC_mount_path/testfile bs=1G count=1 oflag=dsync
  ```

In our example with IBM Spectrum Scale CNSA as storage provider and a remotely mounted IBM Spectrum Scale file system, this storage hardware setup mandates that a minimum of a 10 Gbps network is required to meet the minimum throughput requirements of 209 MBps. For more information, see this web page.

In this deployment, we used a 100 Gbps network for our IBM Spectrum Scale remote mount network.

After the IBM Cloud Pak for Data control plane (`lite` assembly) is installed, you can install other services of interest from the service catalog that support your specific business needs.

> **Note:** Each service on IBM Cloud Pak for Data features specific storage requirements. Ensure that the required storage exists for the services that you plan to install.
>
> For more information about the storage that each service supports, see this web page.

For more information about other system requirements for IBM Cloud Pak for Data, see this web page. Here, the minimum configuration involves three master and a minimum of three worker nodes (see Figure 1-3).

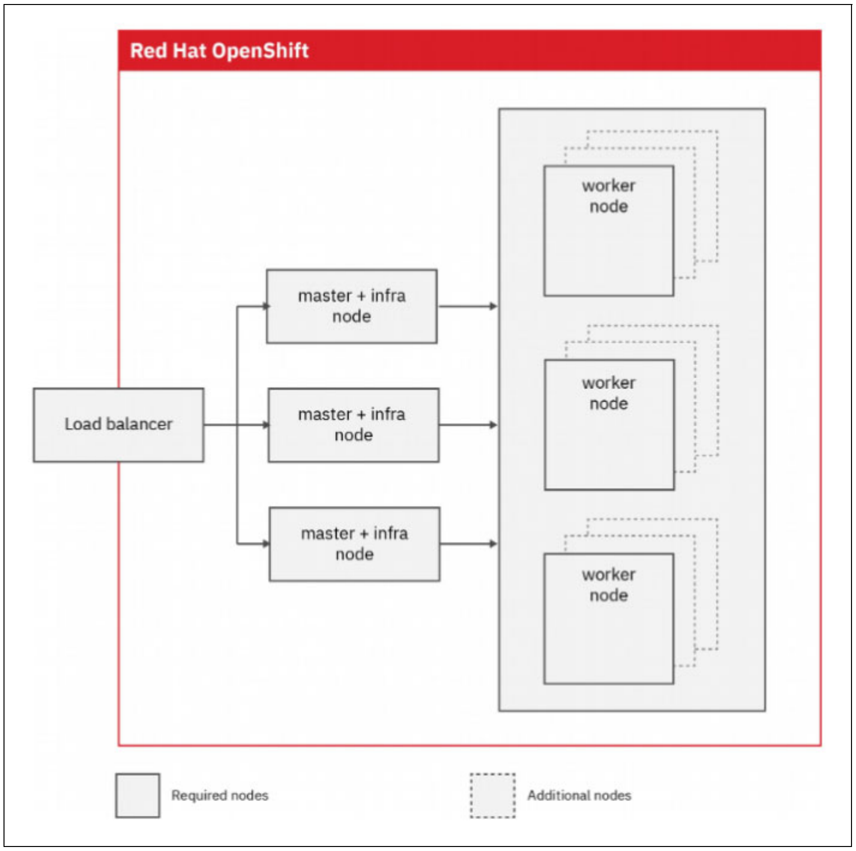


Figure 1-3 IBM Cloud Pak for Data with Red Hat OpenShift minimum configuration

The load balancer can be in the cluster or external to the cluster. However, in a production-level cluster, an enterprise-grade external load balancer is strongly recommended. The load balancer distributes requests between the three master + infra nodes. A production-level cluster must have at least three worker nodes, but you might need to deploy more worker nodes to support your workload.

2

Installing IBM Spectrum Scale Container Native Storage Access and Container Storage Interface

This chapter describes how to install IBM Spectrum Scale Container Native Storage Access (CNSA) and Container Storage Interface (CSI) and includes the following topics:

2.1 Installing IBM Spectrum Scale CNSA and CSI

For more information about installation steps for IBM Spectrum Scale CNSA and CSI, see the following resources:

- IBM Spectrum Scale Container Native Storage Access 5.1.0.3
- IBM Spectrum Scale Container Storage Interface Driver 2.1.0

This chapter provides an overview of the requirements, options, and steps to set up IBM Spectrum Scale CNSA and CSI. It also summarizes the basic configuration and preparation steps and offers a unified deployment with a central `config.yaml` file.

The installation of the IBM Spectrum Scale CNSA 5.1.0.3 release with CSI 2.1.0 requires two distinct installation steps because the CSI deployment is a separate step from the CNSA deployment. The IBM Spectrum Scale CSI deployment depends on a few manual steps that must be performed by an administration user *after* the IBM Spectrum Scale CNSA deployment and *before* the IBM Spectrum Scale CSI deployment.

These manual steps include:

- Creating a local CSI user or password on the running IBM Spectrum Scale CNSA cluster:

  ```
  oc exec -c liberty ibm-spectrum-scale-gui-0 -- /usr/lpp/mmfs/gui/cli/mkuser -p
  -g CsiAdmin
  ```

- Obtaining the local cluster ID of the created IBM Spectrum Scale CNSA cluster:

  ```
  oc exec [ibm-spectrum-scale-core-pod] -- mmlscluster | grep 'GPFS cluster id'
  ```

If you configured your environment and OpenShift cluster to meet all the IBM Spectrum Scale Container Native Storage Access (CNSA) and CSI requirements, skip the next section and see “Deployment steps” on page 16.

2.2 Requirements

To install IBM Spectrum Scale CNSA and CSI on OpenShift 4.6 or higher, the following requirements must be met in addition to the other prerequisites for IBM Spectrum Scale CNSA and CSI:

- The CNSA 5.1.0.3 tar archive are extracted on a local installation node with access to the OpenShift cluster (for example, by using **oc** commands).
- A regular OpenShift cluster admin user with the cluster-admin role is on the OpenShift cluster to deploy CNSA and push the CNSA images to the internal OpenShift image registry; for example, add an identity provider, such as `htpasswd` and add a cluster-admin user:

  ```
  $ oc adm policy add-cluster-role-to-user cluster-admin <user>
  ```

- Podman is on the local installation node to load, tag, and push the IBM Spectrum Scale CNSA images into the internal OpenShift registry or external registry.
- Internet access is available to pull all other dependent images for IBM Spectrum Scale CNSA and CSI from their respective external image registries; for example, quay, and us.gcr.io.

2.3 Preinstallation tasks

Complete the following tasks for the IBM Spectrum Scale CNSA and CSI deployment:

- Obtain the IBM Spectrum Scale Container Native Storage Access tar archive file from Fix Central or Passport Advantage.
- Extract the CNSA .tar archive. For more information, see this web page.
- Configure Red Hat OpenShift Container Platform to increase the PIDS_LIMIT, add the `kernel-devel` extensions (required on OpenShift 4.6 and higher only), and increase the `vmalloc` kernel parameter (this parameter is required for Linux on System Z only).
- Configure IBM Spectrum Scale fileset quotas and configuration parameters for CSI; for example:
  - `--perfileset-quota`
  - `--filesetdf`
  - `enforceFilesetQuotaOnRoot`
  - `controlSetxattrImmutableSELinux`
- Continue on in the procedure to prepare the OpenShift cluster for the deployment of IBM Spectrum Scale CNSA and CSI.

2.3.1 Uploading IBM Spectrum Scale CNSA images to local image registry

After you extract the IBM Spectrum Scale CNSA tar archive, load, tag, and push the IBM Spectrum Scale CNSA images to a local container image registry.

If you have enabled and exposed the internal Red Hat OpenShift image registry in your OpenShift cluster, push all the IBM Spectrum Scale CNSA images into this registry by following the instructions at this web page.

> **Note:** A regular production Red Hat OpenShift cluster includes a correctly configured identity provider and a regular cluster admin user other than the default admin users, such as `kube:admin` or `system:admin`, which are meant primarily as temporary accounts for the initial deployment. They do not provide a token (`oc whoami -t`) to access the internal OpenShift image registry.
>
> For more information about creating a regular cluster admin user, see this web page.
>
> If you configured an identity provider, such as `htpasswd` on your OpenShift cluster, and added a regular OpenShift cluster admin user with cluster-admin role (for example, with `oc adm policy add-cluster-role-to-user cluster-admin <user-name>`), this admin user can push images to the internal OpenShift registry.

2.3.2 Preparing OpenShift cluster nodes to run IBM Spectrum Scale CNSA

To increase the PIDS_LIMIT limit to a minimum of `pidsLimit: 4096` by using the Machine Config Operator (MCO) on OpenShift, apply the provided YAML file in the IBM Spectrum Scale CNSA tar archive, as shown in the following example:

```
---
apiVersion: machineconfiguration.openshift.io/v1
kind: ContainerRuntimeConfig
metadata:
  name: increase-pid-limit
```

```
spec:
  machineConfigPoolSelector:
    matchLabels:
      pid-crio: config-pid
  containerRuntimeConfig:
    pidsLimit: 4096
```

Apply it by using the following commands:

```
# oc create -f <cnsa_extracted_dir>/machineconfig/increase_pid_mco.yaml
# oc label machineconfigpool worker pid-crio=config-pid
```

Note: Running this command drives a rolling update across your Red Hat OpenShift Container Platform worker nodes and can take more than 30 minutes to complete, depending on the size of the worker node pool because the worker is restarted. You can check the progress of the update by using the following command:

```
# oc get MachineConfigPool
```

Wait until the update finished successfully.

Note: IBM Cloud Pak for Data requires a higher PID setting of `12288`. You can apply this setting at this step by changing `pidsLimit: 4096` to `pidsLimit: 12288`.

Confirm the update, as shown in the following example:

```
# oc get nodes -lnode-role.kubernetes.io/worker= \
-ojsonpath="{range .items[*]}{.metadata.name}{'\n'}" |\
xargs -I{} oc debug node/{} -T -- chroot /host crio-status config | grep pids_limit
```

The output for every node should appear as shown in the following example:

```
# oc get nodes -lnode-role.kubernetes.io/worker= \-ojsonpath="{range
.items[*]}{.metadata.name}{'\n'}" |\xargs -I{} oc debug node/{} -T -- chroot /host
crio-status config | grep pids_limit
Starting pod/worker0cpst-ocp-cluster-bcpst-labno-usersibmcom-debug ...
To use host binaries, run `chroot /host`
    pids_limit = 4096
Removing debug pod ...
Starting pod/worker1cpst-ocp-cluster-bcpst-labno-usersibmcom-debug ...
To use host binaries, run `chroot /host`
    pids_limit = 4096
Removing debug pod ...
Starting pod/worker2cpst-ocp-cluster-bcpst-labno-usersibmcom-debug ...
To use host binaries, run `chroot /host`
Pod IP: 9.114.194.185
If you don't see a command prompt, try pressing enter.
    pids_limit = 4096
```

If you are running on OpenShift 4.6.6 (or a higher minor level), you must add the `kerneldevel` extensions by way of the Machine Config Operator by creating a YAML file (here, named `machineconfigoperator.yaml`) as shown in the following example:

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
```

```
  labels:
    machineconfiguration.openshift.io/role: "worker"
  name: 02-worker-kernel-devel
spec:
  config:
    ignition:
      version: 3.1.0
  extensions:
     - kernel-devel
```

Apply it by using following command:

```
# oc create -f <cnsa_extracted_dir>/machineconfig/machineconfigoperator.yaml
```

Check the status of the update by using the following command:

```
# oc get MachineConfigPool
```

Wait until the update finishes successfully.

Validate that the `kernel-devel` package is successfully applied by running the following command:

```
# oc get nodes -lnode-role.kubernetes.io/worker= \
-ojsonpath="{range .items[*]}{.metadata.name}{'\n'}" |\
xargs -I{} oc debug node/{} -T -- chroot /host sh -c "rpm -q kernel-devel"
```

The output for every node should resemble the following example:

```
# oc debug node/worker0.cpst-ocp-cluster-b.cpst-lab.no-users.ibm.com -T -- chroot
/host sh -c "rpm -q kernel-devel"
Starting pod/worker0cpst-ocp-cluster-bcpst-labno-usersibmcom-debug ...
To use host binaries, run `chroot /host`
kernel-devel-4.18.0-193.60.2.el8_2.x86_64
Removing debug pod ...
```

2.3.3 Labeling OpenShift worker nodes for IBM Spectrum Scale CSI

By using the default configuration for IBM Spectrum Scale CNSA, you must label the worker nodes that are eligible to run IBM Spectrum Scale CSI:

```
# oc label nodes -l node-role.kubernetes.io/worker scale=true --overwrite=true
```

2.4 Deployment steps

In this section, we describe the process to deploy IBM Spectrum Scale CNSA and CSI, which includes the following steps:

1. Prepare IBM Spectrum Scale remote storage cluster, OpenShift namespaces, and secrets.
2. Edit the `operator.yaml` and `ibm_v1_scalecluster_cr.yaml` files to reflect your local. environment.
3. Deploy IBM Spectrum Scale CNSA (`ibm-spectrum-scale-ns`).
4. Deploy IBM Spectrum Scale CSI (`ibm-spectrum-scale-csi`).

2.4.1 Step 1: Preparing IBM Spectrum Scale remote storage cluster, OpenShift namespaces, and secrets

This step prepares required settings and GUI user accounts on the *remote* IBM Spectrum Scale storage cluster for the IBM Spectrum Scale CNSA and CSI deployment. The following user accounts are needed:

- One user account for a CNSA user (here, we use `cnsa_admin` with password `CNSA_PASSWORD`)
- One user account for a CSI user (here, we use `csi_admin` with password `CSI_PASSWORD`)

This step also prepares the namespaces (that is, *projects*) and creates the Kubernetes secrets in OpenShift for the IBM Spectrum Scale CNSA and IBM Spectrum Scale CSI driver deployment. The secrets include the credentials for the required CNSA and CSI users for the local and remote IBM Spectrum Scale GUIs.

Preparing GUI users for CNSA on the remote IBM Spectrum Scale storage cluster

Complete the following steps:

1. Check the IBM Spectrum Scale remote storage cluster to determine whether the GUI user group `ContainerOperator` exists by running the following command:

   ```
   # /usr/lpp/mmfs/gui/cli/lsusergrp ContainerOperator
   ```

2. If the GUI user group `ContainerOperator` does not exist, create it by using the following command:

   ```
   # /usr/lpp/mmfs/gui/cli/mkusergrp ContainerOperator --role containeroperator
   ```

3. Check to see if no user for CNSA exists in the `ContainerOperator` group:

   ```
   # /usr/lpp/mmfs/gui/cli/lsuser | grep ContainerOperator
   #
   ```

   Create a user if none exists:

   ```
   # /usr/lpp/mmfs/gui/cli/mkuser cnsa_admin -p CNSA_PASSWORD -g ContainerOperator
   ```

   This user is used later by IBM Spectrum Scale CNSA through the `cnsa-remote-gui-secret` secret.

Preparing GUI user for CSI on the remote IBM Spectrum Scale storage cluster

Complete the following steps:

1. Check the IBM Spectrum Scale remote storage cluster to determine whether the GUI user group CsiAdmin exists by issuing the following command:

   ```
   # /usr/lpp/mmfs/gui/cli/lsusergrp CsiAdmin
   ```

   If the GUI user group CsiAdmin does not exist, create it by using the following command:

   ```
   # /usr/lpp/mmfs/gui/cli/mkusergrp CsiAdmin --role csiadmin
   ```

2. Check to see if no user for the CSI driver exists in the CsiAdmin group:

   ```
   # /usr/lpp/mmfs/gui/cli/lsuser | grep CsiAdmin
   #
   ```

   Create a user if none exists:

   ```
   # /usr/lpp/mmfs/gui/cli/mkuser csi_admin -p CSI_PASSWORD -g CsiAdmin
   ```

This user is used later by the IBM Spectrum Scale CSI driver through the `csi-remote-secret` secret.

Applying quota and configuration settings for CSI on the remote IBM Spectrum Scale storage cluster

Complete the following steps:

1. Ensure that per `--fileset-quota` on the file systems to be used by IBM Spectrum Scale CNSA and CSI is set to `no`. Here, we use `ess3000_1M` as the file system for IBM Spectrum Scale CNSA and CSI:

```
# mmlsfs ess3000_1M --perfileset-quota
flag                value                    description
------------------- ------------------------ -----------------------------------
 --perfileset-quota no                       Per-fileset quota enforcement
```

   If it is set to `yes`, change it to `no` by using the **`mmchfs`** command:

   ```
   # mmchfs ess3000_1M --noperfileset-quota
   ```

2. Enable quota for all the file systems that are used for fileset-based dynamic provisioning with IBM Spectrum Scale CSI by using the **`mmchfs`** command:

   ```
   # mmchfs ess3000_1M -Q yes
   ```

   Verify that quota is enabled for the file system (in our example, `ess3000_1M`) by using the **`mmlsfs`** command:

```
# mmlsfs ess3000_1M -Q
flag                value                    description
------------------- ------------------------ -----------------------------------
-Q                  user;group;fileset       Quotas accounting enabled
                    user;group;fileset       Quotas enforced
                    none                     Default quotas enabled
```

3. Enable quota for the root user by issuing the following command:

   ```
   # mmchconfig enforceFilesetQuotaOnRoot=yes -i
   ```

4. For Red Hat OpenShift, ensure that the `controlSetxattrImmutableSELinux` parameter is set to `yes` by issuing the following command:

   ```
   # mmchconfig controlSetxattrImmutableSELinux=yes -i
   ```

5. Display the correct volume size in a container by enabling `filesetdf` on the file system by using the following command:

```
# mmchfs ess3000_1M --filesetdf
```

Preparing namespaces in OpenShift

Log in to the OpenShift cluster as regular cluster admin user with a `cluster-admin` role to perform the next steps.

Create the following namespaces (that is, projects) in OpenShift:

- One for the IBM Spectrum Scale CNSA deployment; in our example, we use `ibm-spectrum-scale-ns` as name for the CNSA namespace.
- One for the IBM Spectrum Scale CSI driver deployment; in our example, we use `ibm-spectrum-scale-csi-driver` for the CSI namespace.

If not yet done, create a namespace/project for CNSA:

```
# oc new-project <ibm-spectrum-scale-ns>
```

At this time, we also prepare the namespace/project for the IBM Spectrum Scale CSI driver in advance:

```
# oc new-project <ibm-spectrum-scale-csi-driver>
```

The `oc new-project <my-namespace>` also switches immediately to the newly created namespace/project. Therefore, you must switch back with `oc project <ibm-spectrum-scale-ns>` to the CNSA namespace as first step of the deployment. Alternatively, you can also use `oc create namespace <my-namespace>`, which does not switch to the created namespace.

Creating a secret for CNSA

IBM Spectrum Scale CNSA requires a GUI user account on the remote IBM Spectrum Scale storage cluster. The credentials are provided as `username` and `password` through a Kubernetes secret in the CNSA namespace.

Create a Kubernetes secret in the CNSA namespace holding the user credentials from the CNSA GUI user on the remote IBM Spectrum Scale storage cluster:

```
# oc create secret generic cnsa-remote-gui-secret
--from-literal=username='cnsa_admin'--from-
literal=password='CNSA_PASSWORD' -n ibm-spectrum-scale-ns
```

Creating secrets for CSI

CSI requires a GUI user account on the remote IBM Spectrum Scale storage cluster and the local CNSA cluster. The credentials are provided as username and password through Kubernetes secrets in the IBM Spectrum Scale CSI namespace (in our example, `ibm-spectrum-scale-csi-driver`).

Create and label the Kubernetes secret in the CSI namespace holding the user credentials from the CSI GUI user on the remote IBM Spectrum Scale storage cluster that we created earlier:

```
# oc create secret generic csi-remote-secret --from-literal=username='csi_admin'
--from- literal=password='CSI_PASSWORD'-n ibm-spectrum-scale-csi-driver
# oc label secret csi-remote-secret product=ibm-spectrum-scale-csi -n
ibm-spectrum-scale-csi-driver
```

At this time, we plan ahead and also create the required Kubernetes secret for the CSI admin user in the local CNSA cluster in advance; that is, before we deploy CNSA or create the CSI admin user in the GUI of the local CNSA cluster:

```
# oc create secret generic csi-local-secret --from-literal=username='csi_admin'
--from-literal=password='CSI_PASSWORD' -n ibm-spectrum-scale-csi-driver
# oc label secret csi-local-secret product=ibm-spectrum-scale-csi -n
ibm-spectrum-scale-csi-driver
```

Note: The CSI driver user credentials on the local compute (CNSA) and remote storage cluster can be created and configured with different user names and passwords and do not need to be identical.

We use these credentials when creating the CSI admin user in the local CNSA cluster after the IBM Spectrum Scale CNSA deployment.

Verifying access to the remote IBM Spectrum Scale storage cluster GUI

Before moving on, it is a good idea to verify access to the GUI of the remote IBM Spectrum Scale storage cluster by running, for example, with the CNSA admin user and the CSI admin user credentials (from an admin node on the OpenShift cluster network):

```
# curl -k -u 'csi_admin:CSI_PASSWORD' https://<remote storage cluster GUI
host>:443/scalemgmt/v2/cluster
```

Successfully running this command ensures that the user credentials are correct and that the nodes on the OpenShift network can access the remote IBM Spectrum Scale storage cluster.

2.5 Editing the operator.yaml and ibm_v1_scalecluster_cr.yaml files to reflect your local environment

The `operator.yaml` file orchestrates some of the configuration activities for IBM Spectrum Scale CNSA deployment. The `operator.yaml` file is included in the IBM Spectrum Scale CNSA tar file.

Make sure to provide the local or external registry where the IBM Spectrum Scale images resides (see 2.3.1, "Uploading IBM Spectrum Scale CNSA images to local image registry" on page 13) in the `operator.yaml` file:

```
...
# Replace the value to point at the operator image
          # Example using internal image repository:
image-registry.openshift-image-registry.svc:5000/ibm-spectrum-scale-ns/ibm-spectru
m-scale-core-operator:vX.X.X.X
         image: REPLACE_SCALE_CORE_OPERATOR_IMAGE
...
```

The `ibm_v1_scalecluster_cr.yaml` holds the configurable parameters for your local environment.

Edit the `ibm_v1_scalecluster_cr.yaml` to match the configuration of your local environment for the IBM Spectrum Scale CNSA and the IBM Spectrum Scale CSI deployment.

To configure the custom resource YAMLs, see the following CNSA and CSI IBM Documentation:

- CNSA - Custom Resource
- CSI - Custom Resource

2.5.1 Minimum required configuration

At a minimum, you must configure the following parameters for IBM Spectrum Scale Container Native Storage Access (CNSA).

Here, we configure the `primaryFilesystem` that is to be mounted on the local CNSA cluster from the remote IBM Spectrum Scale storage cluster and also host the primary fileset of IBM Spectrum Scale CSI to store its configuration data:

```
# ------------------------------------------------------------------------------
  # filesystems block is required for Remote Mount
  # ------------------------------------------------------------------------------
  # filesystems[name].remoteMount.storageCluster refers to the name of a remoteCluster defined in the 
proceeding block
  # note: adding, removing, or updating a filesystem name or mountPoint after first deployment will 
require manual pod deletions.
  filesystems:
    - name: "fs1"
      remoteMount:
        storageCluster: "storageCluster1"
        storageFs: "fs1"
      # mountPoint must start with `/mnt`
      mountPoint: "/mnt/fs1"
```

The following parameters are used to configure `primaryFilesystem`:

- `name`: Local name of the file system on the IBM Spectrum Scale CNSA cluster

  > **Note:** This local `name` must comply with Kubernetes DNS label rules (see DNS Label Names).

- `mountPoint`: Local mount point of the remote file system on OpenShift (must be under `/mnt`).
- `storageCluster`: Internal object name to reference the remote cluster definition object in the next section.
- `storageFs`: Original name of the file system on the remote IBM Spectrum Scale storage cluster (for example, from `mmlsconfig or curl -k -u 'cnsa_admin:CNSA_PASSWORD' https://<remote storage cluster GUI host>:443/scalemgmt/v2/filesystems`).

Here, we configure the `remoteClusters` that provides the file system for the remote mount:

```
# ------------------------------------------------------------------------------
  # The remoteCluster field is required for remote mount
  # ------------------------------------------------------------------------------
  # A remoteCluster definition provides the name, hostname, its GUI secret, and contact node.
  # The remoteCluster name is referenced in the filesystems[name].remoteMount.storageCluster
  # used for Remote Mount
  remoteClusters:
    - name: storageCluster1
      gui:
        cacert: "cacert-storage-cluster-1"
        host: ""
        secretName: "cnsa-remote-gui-secret"
        insecureSkipVerify: false
```

```
# contactNodes:
#   - storagecluster1node1
#   - storagecluster1node2
```

The following parameters are used to configure `remoteClusters`:

- `name`: This name is used to identify the remote Storage Cluster.
- `gui`: This information is used to access the remote Storage Cluster's GUI.
- `cacert`: This name is the name of the Kubernetes `configmap` that contains the CA certificate for the storage cluster GUI.
- `host`: Hostname for the GUI endpoint on the storage cluster.

> **Note:** If `insecureSkipVerify` is set to `false`, the hostname that is encoded in the `cacert ConfigMap` must match the value that is provided for host.

- `secretName`: This name of the Kubernetes secret is created during the storage cluster configuration.

> **Note:** Specify the secret name that you noted in Create Secret.

- `insecureSkipVerify`: Controls whether a client verifies the storage cluster's GUI certificate chain and hostname. If set true, TLS is susceptible to machine-in-the-middle attacks. The default setting is `false`.
- `contactNodes` (optional): Provide a list of storage nodes to be used as the contact nodes list. If not specified, the operator uses three nodes from the storage cluster.

Based on the registry option, replace the registry in the following section of the `ibm_v1_scalecluster_cr.yaml`. For example, use `image-registry.openshift-image-registry.svc:5000/ibm-spectrum-scale-ns` for the internal OpenShift image registry and CNSA namespace `ibm-spectrum-scale-ns`:

```
# ------------------------------------------------------------------------------
  # images is the list of Docker container images required to deploy and run IBM Spectrum Scale
  # ------------------------------------------------------------------------------
  # note: changing the following fields after first deployment will require manual pod deletions.
  images:
    core: REPLACE_CONTAINER_REGISTRY/ibm-spectrum-scale-core:v5.1.0.3
    coreInit: REPLACE_CONTAINER_REGISTRY/ibm-spectrum-scale-core:v5.1.0.3
    gui: REPLACE_CONTAINER_REGISTRY/ibm-spectrum-scale-gui:v5.1.0.3
    postgres:
"docker.io/library/postgres@sha256:a2da8071b8eba341c08577b13b41527eab3968bf1c8d28123b5b07a493a26862"
    pmcollector: REPLACE_CONTAINER_REGISTRY/ibm-spectrum-scale-pmcollector:v5.1.0.3
    sysmon: REPLACE_CONTAINER_REGISTRY/ibm-spectrum-scale-monitor:v5.1.0.3
    logs: "registry.access.redhat.com/ubi8/ubi-minimal:8.3"
```

For more information about the IBM Spectrum Scale CNSA configuration parameters, see CNSA Operator - Custom Resource.

For more information about the IBM Spectrum Scale CSI driver configuration parameters, see Configuring Custom Resource for CSI driver.

2.5.2 Optional configuration parameters

In this section, we describe the available configuration parameters.

Call Home

You can enable and configure Call Home for IBM Spectrum Scale CNSA in the following section of the `ibm_v1_scalecluster_cr.yaml` file:

```
callHome:
# call home functionality is optional #
 # # TO ENABLE: Remove the first # character on each line of this section to
configure and enable call home
# callhome:
# # By accepting this request, you agree to allow IBM and its subsidiaries to
store and use your contact information and your support information anywhere they
do business worldwide. For more information, please refer to the Program license
agreement and documentation.
# # If you agree, please respond with "true" for acceptance, else with "false" to
decline.
# acceptLicense: true | false
# # companyName of the company to which the contact person belongs.
# # This name can consist of any alphanumeric characters and these
non-alphanumeric characters: '-', '_', '.', ' ', ','.
# companyName:
# # customerID of the system administrator who can be contacted by the IBM
Support.
# # This can consist of any alphanumeric characters and these non-alphanumeric
characters: '-', '_', '.'.
# customerID: ""
# # companyEmail address of the system administrator who can be contacted by the
IBM Support.
# # Usually this e-mail address is directed towards a group or task e-mail
address. For example, itsupport@mycompanyname.com.
# companyEmail:
# # countryCode two-letter upper-case country codes as defined in ISO 3166-1
alpha-2.
# countryCode:
# # Marks the cluster as a "test" or a "production" system. In case this parameter
is not explicitly set, the value is set to "production" by default. # type:
production | test
# # Remove or leave the proxy block commented if a proxy should not be used for
uploads
# proxy:
# # host of proxy server as hostname or IP address
# host:
# # port of proxy server
# port:
# # secretName of a basic-auth secret, which contains username and password for
proxy server # # Remove the secretName if no authentication to the proxy server is
needed.
# secretName:
```

Host name aliases

The host names of the remote IBM Spectrum Scale storage cluster contact nodes must be resolvable by way of DNS by the OpenShift nodes.

If the IP addresses of these contact nodes cannot be resolved by way of DNS (including a reverse lookup), the hostname and their IP addresses can be specified in the `hostAliases` section of `ibm_v1_scalecluster_cr.yaml` file that is shown in Example 2-1.

Example 2-1 Specifying hostname and their IP addresses

```
# hostAliases is used in an environment where DNS cannot resolve the remote (storage) cluster
# note: changing this field after first deployment will require manual pod deletions.
# hostAliases:
#  - hostname: example.com
#  ip: 10.0.0.1
```

2.6 Deploying IBM Spectrum Scale CNSA

Log in to the OpenShift cluster as regular admin user with a `cluster-admin` role, switch to the CNSA namespace (here, `ibm-spectrum-scale-ns`):

```
# oc project ibm-spectrum-scale-ns
```

Deploy the Operator by creating the provided yaml files, as shown in Example 2-2.

Example 2-2 Creating the provided yaml files

```
oc create -f spectrumscale/deploy/crds/ibm_v1_scalecluster_crd.yaml -n ibm-spectrum-scale-ns
oc create -f spectrumscale/deploy/crds/ibm_v1_scalecluster_cr.yaml -n ibm-spectrum-scale-ns
oc create -f spectrumscale/deploy -n ibm-spectrum-scale-ns
```

Verify that the Operator creates the ScaleCluster Custom Resource by checking pods and Operator logs:

- Get the pods:

  ```
  # oc get pods -n ibm-spectrum-scale-ns
  ```

- Tail the operator log:

  ```
  # oc logs $(oc get pods -lname=ibm-spectrum-scale-core-operator -n
  ibm-spectrum-scale-ns -ojsonpath="{range .items[0]}{.metadata.name}") -n
  ibm-spectrum-scale-ns -f
  ```

Sample output:

```
[root@arcx3650fxxnh ~]# oc get pods -o wide
NAME                                       READY   STATUS    RESTARTS   AGE     IP        NODE
NOMINATED NODE                                       READINESS       GATES
ibm-spectrum-scale-core-5btzt              1/1     Running   0          3h59m   9.11.110.126
worker5.cpst-ocp-cluster-a.cpst-lab.no-users.ibm.com   <none>          <none>
ibm-spectrum-scale-core-k4gbd              1/1     Running   0          3h59m   9.11.110.157
worker3.cpst-ocp-cluster-a.cpst-lab.no-users.ibm.com   <none>          <none>
ibm-spectrum-scale-core-q5svl              1/1     Running   0          3h59m   9.11.110.150
worker4.cpst-ocp-cluster-a.cpst-lab.no-users.ibm.com   <none>          <none>
ibm-spectrum-scale-gui-0                   9/9     Running   0          3h59m   10.128.4.9
worker5.cpst-ocp-cluster-a.cpst-lab.no-users.ibm.com   <none>          <none>
```

```
ibm-spectrum-scale-operator-7b7dc6cb5-fjlw2   1/1     Running   0          3h59m   10.131.2.25
worker4.cpst-ocp-cluster-a.cpst-lab.no-users.ibm.com   <none>          <none>
ibm-spectrum-scale-pmcollector-0              2/2     Running   0          3h59m   10.128.4.8
worker5.cpst-ocp-cluster-a.cpst-lab.no-users.ibm.com   <none>          <none>
ibm-spectrum-scale-pmcollector-1              2/2     Running   0          3h58m   10.130.2.10
worker3.cpst-ocp-cluster-a.cpst-lab.no-users.ibm.com   <none>          <none>
```

- Verity that the IBM Spectrum Scale cluster has been created:

```
oc exec $(oc get pods -lapp=ibm-spectrum-scale-core \
-ojsonpath="{.items[0].metadata.name}" -n ibm-spectrum-scale-ns)  \
-n ibm-spectrum-scale-ns -- mmlscluster
oc exec $(oc get pods -lapp=ibm-spectrum-scale-core \
-ojsonpath="{.items[0].metadata.name}" -n ibm-spectrum-scale-ns) \
-n ibm-spectrum-scale-ns -- mmgetstate -a
Verify that the storage cluster has been configured:
oc exec $(oc get pods -lapp=ibm-spectrum-scale-core \
-ojsonpath="{.items[0].metadata.name}" -n ibm-spectrum-scale-ns) \
-n ibm-spectrum-scale-ns -- mmremotecluster show all
```

- Verify the storage cluster file system has been configured:

```
oc exec $(oc get pods -lapp=ibm-spectrum-scale-core \
-ojsonpath="{.items[0].metadata.name}" -n ibm-spectrum-scale-ns) \
-n ibm-spectrum-scale-ns -- mmremotefs show
```

- Verify the storage cluster file system has been remotely mounted:

```
oc exec $(oc get pods -lapp=ibm-spectrum-scale-core \
-ojsonpath="{.items[0].metadata.name}" -n ibm-spectrum-scale-ns) \
-n ibm-spectrum-scale-ns -- mmlsmount fs1 -L
```

> **Note:** This `fs1` file system is the name of the CNSA cluster's file system and not the remote cluster's file system. Therefore, the name of the file system can vary based on the name that you used for the file system.

- Verify status and events of the IBM Spectrum Scale Operator:

```
oc describe gpfs
```

During the CNSA deployment, several Docker images are pulled. You might experience a case where you exceed the Docker pull requests. To prevent this issue, add the Docker secret and link:

```
oc create secret docker-registry dockerio-secret \
    --docker-server=docker.io \
    --docker-username=<docker-username> \
    --docker-password=<docker-password> \
    --docker-email=<docker-user>
```

To link to a pod that indicated Docker pull failure, you can run the following command. In this example, we are linking it to a GUI pod (see Example 2-3).

Example 2-3   Linking to a GUI pod

```
oc secrets link ibm-spectrum-scale-gui dockerio-secret --for=pull -n ibm-spectrum-scale-ns
```

Later, you can delete the failing pod. It is automatically recreated, and deployed successfully.

You can check the IBM Spectrum Scale CNSA operator log by using the following command:

```
# oc logs <ibm-spectrum-scale-operator-pod> -f
```

Or, you can quickly check for errors by using the following command:

```
# oc logs <ibm-spectrum-scale-operator-pod> | grep -i error
```

2.7 Deploying IBM Spectrum Scale CSI

Remain in the `ibm-spectrum-scale-ns` namespace of the IBM Spectrum Scale CNSA deployment and perform the steps that are described in this section.

Before we can deploy the IBM Spectrum Scale CSI driver, we must create a GUI user for IBM Spectrum Scale CSI on the GUI pod of the local IBM Spectrum Scale CNSA cluster that we just deployed (see Example 2-4). Then, we use the same credentials that we used when creating the `csi-local-secret` earlier (see “Creating a secret for CNSA” on page 18):

Example 2-4 Creating a GUI user

```
# oc -n ibm-spectrum-scale-ns exec -c liberty ibm-spectrum-scale-gui-0 --
/usr/lpp/mmfs/gui/cli/mkuser csi_admin -p CSI_PASSWORD
-g CsiAdminDeploy Operator
# oc create -f
https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi/v2.1.0/generated/installer/ibm-spec
trum-scale-csi-operator.yaml -n ibm-spectrum-scale-csi-driver
```

For CSI driver, the following custom resource must be downloaded:

```
# curl -O
https://raw.githubusercontent.com/IBM/ibm-spectrum-scale-csi/v2.1.0/operator/deplo
y/crds/csiscaleoperators.csi.ibm.com_cr.yaml
```

After downloading the file, the following parameters must be modified according to the environment:

- The path to the file system mounted at IBM Spectrum Scale CNSA cluster (for example, `/mnt/fs1`):

  ```
  # ===============================================================================
    scaleHostpath: "< GPFS FileSystem Path >"
  ```

- Fulfill the information for the IBM Spectrum Scale CNSA cluster in the area that is shown in Example 2-5.

Example 2-5 Fulfilling the information for the IBM Spectrum Scale CNSA cluster

```
================================================================================
  clusters:
    - id: "< Primary Cluster ID - WARNING - THIS IS A STRING NEEDS YAML QUOTES! >"
      secrets: "secret1"
      secureSslMode: false
      primary:
        primaryFs: "< Primary Filesystem >"
```

```
#       primaryFset: "< Fileset in Primary Filesystem >"   # Optional - 
default:spectrum-scale-csi-volume-store
#       inodeLimit: "< inode limit for Primary Fileset >" # Optional
#       remoteCluster: "< Remote ClusterID >"           # Optional - This is only required if 
primaryFs is remote cluster's filesystem and this ID should have separate entry in Clusters map 
too.
#      cacert:  "< Name of CA cert configmap for GUI >"   # Optional
      restApi:
        - guiHost: "< Primary cluster GUI IP/Hostname >"
#
# In the case we have multiple clusters, specify their configuration below.
# ==================================================================================
```

- Fulfill the information for the remote IBM Spectrum Scale cluster in this area:

```
#    - id: "< Cluster ID >"
#      secrets: "< Secret for Cluster >"
#      secureSslMode: false
#      restApi:
#      - guiHost: "< Cluster GUI IP/Hostname >"
#      cacert:  "< Name of CA cert configmap for GUI >"   # Optional

# Attacher image name, in case we do not want to use default image.
#
==================================================================================
```

- To find the mandatory cluster ID for IBM Spectrum Scale CNSA cluster, run the following command:

```
# oc -n ibm-spectrum-scale-ns exec <ibm-spectrum-scale-core-pod> -- curl -s -k 
https://ibm-spectrum-scale-gui.ibm-spectrum-scale-ns/scalemgmt/v2/cluster -u 
"cnsa_admin:CNSA_PASSWORD" | grep clusterId
```

- To find the mandatory cluster ID for Spectrum Scale Remote cluster, run the following command:

```
# curl -s -k https://example-gui.com/scalemgmt/v2/cluster -u 
"csi_admin:CSI_PASSWORD" | grep clusterId
```

The following parameters are available to remote-mount a file system:

id (Mandatory) Cluster ID of the primary IBM Spectrum Scale cluster. For more information, see **mmlscluster** command in the IBM Spectrum Scale: Concepts, Planning, and Installation Guide.

primaryFs (Mandatory) Primary file system name (local CNSA file system name).

primaryFset (Optional) Primary fileset name: This name is created if the fileset does not exist.

Default value: `spectrum-scale-csi-volume-store`

inodeLimit (Optional) Inode limit for the primary fileset. If not specified, fileset is created with 1 M inodes, which is the IBM Spectrum Scale default.

cacert Mandatory if `secureSslMode` is `true`. Name of the pre-created CA certificate `configmap` that is used to connect to the GUI server (running on the “guiHost”). For more information, see IBM Documentation.

secrets (Mandatory) Name of the pre-created Secret that contains the username and password that are used to connect to the GUI server for the cluster that is specified against the ID parameter. For more information, see IBM Documentation.

guiHost (Mandatory) FQDN or IP address of the GUI node of IBM Spectrum Scale cluster that is specified against the ID parameter.

scaleHostpath (Mandatory) Mount path of the primary file system (primaryFs).

imagePullSecrets (Optional) An array of `imagePullSecrets` to be used for pulling images from a private registry. This pass-through option distributes the `imagePullSecrets` array to the containers that are generated by the Operator. For more information about creating `imagePullSecrets`, see this web page.

Create the custom resource from the `csiscaleoperators.csi.ibm.com_cr.yaml` file that was downloaded and modified in the previous section:

```
# oc create -f csiscaleoperators.csi.ibm.com_cr.yaml -n
ibm-spectrum-scale-csi-driver
```

Verify that the IBM Spectrum Scale CSI driver is installed, Operator and driver resources are ready, and pods are in running state. It might take some time for the CSI driver pods to get scheduled and running.:

```
# oc get pod,daemonset,statefulset -n ibm-spectrum-scale-csi-driver
NAME                                                    READY   STATUS    RESTARTS   AGE
pod/ibm-spectrum-scale-csi-8pk49                        2/2     Running   0          3m3s
pod/ibm-spectrum-scale-csi-attacher-0                   1/1     Running   0          3m12s
pod/ibm-spectrum-scale-csi-b2f7x                        2/2     Running   0          3m3s
pod/ibm-spectrum-scale-csi-operator-67448f6956-2xlsv    1/1     Running   0          27m
pod/ibm-spectrum-scale-csi-provisioner-0                1/1     Running   0          3m7s
pod/ibm-spectrum-scale-csi-vjsvc                        2/2     Running   0          3m3s

NAME                                   DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/ibm-spectrum-scale-csi   3         3         3       3            3           scale=true      3m3s

NAME                                                   READY   AGE
statefulset.apps/ibm-spectrum-scale-csi-attacher       1/1     3m12s
statefulset.apps/ibm-spectrum-scale-csi-provisioner    1/1     3m7s
```

The deployment is now completed and IBM Spectrum Scale CNSA and CSI are successfully running on your OpenShift cluster.

Now, you can start creating Kubernetes StorageClasses (SCs), persistent volumes (PVs) and persistent volume claims (PVCs) to provide persistent storage to your containerized applications. For more information, see IBM Documentation.

2.8 Removing IBM Spectrum Scale CNSA and CSI deployment

For more information about removing IBM Spectrum Scale Container Native Storage Access, see IBM Documentation.

2.9 Example of use of IBM Spectrum Scale provisioned storage

A set of YAML manifest files are available in the `examples` directory of this this GitHub repository.

These example YAML manifest files are helpful to quickly test dynamic provisioning of persistent volumes with IBM Spectrum Scale CNSA.

These examples feature the following components:

- ibm-spectrum-scale-sc.yaml: An SC to allow dynamic provisioning of PVs (created by an admin)
- ibm-spectrum-scale-pvc.yaml: A PVC to request a PV from the storage class (issued by a user)
- ibm-spectrum-scale-test-pod.yaml: A test pod that is writing a time stamp every 5 seconds into the volume backed by IBM Spectrum Scale (started by *user*)

An OpenShift admin user must create an SC for dynamic provisioning. In this example, we use an SC that provides dynamic provisioning of persistent volumes that are backed by independent filesets in IBM Spectrum Scale.

IBM Spectrum Scale CSI driver allows the use of the following types of SCs for dynamic provisioning:

- Light-weight volumes that use simple directories in IBM Spectrum Scale
- File-set based volumes that use:
  - Independent filesets in IBM Spectrum Scale
  - Dependent filesets in IBM Spectrum Scale

For more information, see IBM Documentation.

Edit the provided storage class `ibm-spectrum-scale-sc.yaml` and set the values of `volBackendFs` and `clusterId` to match your configured environment:

```
volBackendFs: "<filesystem name of the local CNSA cluster>"
clusterId: "<cluster ID of the remote storage cluster>"
```

Apply the SC, as shown in Example 2-6.

Example 2-6 Applying the storage class

```
# oc apply -f ./examples/ibm-spectrum-scale-sc.yaml
storageclass.storage.k8s.io/ibm-spectrum-scale-sc created
# oc get sc
NAME                  PROVISIONER              RECLAIMPOLICY VOLUMEBINDINGMODE ALLOWVOLUMEEXPANSION AGE
ibm-spectrum-scale-sc spectrumscale.csi.ibm.com  Delete      Immediate     false                  2s
```

Now, we can switch to a regular user profile in OpenShift, and create a namespace:

```
# oc new-project test-namespace
Now using project "test-namespace" on server "https://api.ocp4.scale.com:6443".
```

Then, we issue a request for a PVC by applying `ibm-spectrum-scale-pvc.yaml`:

```
# oc apply -f ./examples/ibm-spectrum-scale-pvc.yaml
persistentvolumeclaim/ibm-spectrum-scale-pvc created
# oc get pvc
NAME                   STATUS  VOLUME                                     CAPACITY ACCESS MODES STORAGECLASS          AGE
ibm-spectrum-scale-pvc Bound   pvc-87f18620-9fac-44ce-ad19-0def5f4304a1   1Gi      RWX          ibm-spectrum-scale-sc 75s
```

Wait until the PVC is bound to a PV. A PVC (like a pod) is bound to a namespace in OpenShift (unlike a PV which is not a namespaced object).

After we see that the PVC is bound to a PV, we can run the test pod by applying `ibm-spectrum-scaletest-pod.yaml`:

```
# oc apply -f ./examples/ibm-spectrum-scale-test-pod.yaml
pod/ibm-spectrum-scale-test-pod created
```

When the pod is running, you can see that a time stamp is written in 5 second intervals to a log `stream1.out` in the local `/data` directory of the pod:

```
# oc get pods
NAME                         READY   STATUS   RESTARTS AGE
ibm-spectrum-scale-test-pod  1/1     Running  0        23s

# oc rsh ibm-spectrum-scale-test-pod
/ # cat /data/stream1.out
ibm-spectrum-scale-test-pod 20210215-12:00:29
ibm-spectrum-scale-test-pod 20210215-12:00:34
ibm-spectrum-scale-test-pod 20210215-12:00:39
ibm-spectrum-scale-test-pod 20210215-12:00:44
ibm-spectrum-scale-test-pod 20210215-12:00:49
ibm-spectrum-scale-test-pod 20210215-12:00:54
ibm-spectrum-scale-test-pod 20210215-12:00:59
ibm-spectrum-scale-test-pod 20210215-12:01:04
ibm-spectrum-scale-test-pod 20210215-12:01:09
ibm-spectrum-scale-test-pod 20210215-12:01:14
ibm-spectrum-scale-test-pod 20210215-12:01:19
```

The pod's `/data` directory is backed by the `pvc-87f18620-9fac-44ce-ad19-0def5f4304a1/pvc-87f18620-9fac-44ce-ad19-0def5f4304a1-data/` directory in the file system on the remote IBM Spectrum Scale storage cluster:

```
# cat /<mount point of filesystem on remote storage
cluster>/pvc-87f18620-9fac-44ce-ad19-0def5f4304a1/pvc-87f18620-9fac-44ce-ad19-
0def5f4304a1-data/stream1.out
ibm-spectrum-scale-test-pod 20210215-12:00:29
ibm-spectrum-scale-test-pod 20210215-12:00:34
ibm-spectrum-scale-test-pod 20210215-12:00:39
ibm-spectrum-scale-test-pod 20210215-12:00:44
ibm-spectrum-scale-test-pod 20210215-12:00:49
ibm-spectrum-scale-test-pod 20210215-12:00:54
ibm-spectrum-scale-test-pod 20210215-12:00:59
ibm-spectrum-scale-test-pod 20210215-12:01:04
```

2.9.1 Other configuration options

This section describes other available configuration options.

Specify node labels for IBM Spectrum Scale CSI (optional)

IBM Spectrum Scale CSI also makes use of node labels to determine on which OpenShift nodes the attacher, provisioner, and plug-in resources are to run. The default node label that is used is `scale:true`, which designates the nodes on which IBM Spectrum Scale CSI resources are running. These nodes must be part of a local IBM Spectrum Scale cluster (here, IBM Spectrum Scale CNSA).

For Cloud Pak for Data to function correctly with IBM Spectrum Scale CSI driver provisioner, all worker nodes must be labeled as `scale=true`.

Label the nodes that are selected to run IBM Spectrum Scale CNSA and IBM Spectrum Scale CSI as show in the following example:

```
# oc label node <worker-node> scale=true --overwrite=true
```

You can define this label in the IBM Spectrum Scale CSI CR file `csiscaleoperators.csi.ibm.com_cr.yaml`, as shown in the following example:

```
# pluginNodeSelector specifies nodes on which we want to run plugin daemoset
 # In below example plugin daemonset will run on nodes which have label as
# "scale=true". Can have multiple entries.
# ===========================================================================
pluginNodeSelector:
key: "scale"
value: "true"
```

Here, we used the default configuration for IBM Spectrum Scale CNSA and CSI and labeled all OpenShift worker nodes with `scale=true`:

```
# oc label nodes -l node-role.kubernetes.io/worker scale=true --overwrite=true
# oc get nodes -l scale=true
NAME                    STATUS   ROLES    AGE      VERSION
worker01.ocp4.scale.com Ready    worker   2d22h    v1.18.3+65bd32d
worker02.ocp4.scale.com Ready    worker   2d22h    v1.18.3+65bd32d
worker03.ocp4.scale.comReady     worker   2d1h     v1.18.3+65bd32d
```

Optional: IBM Spectrum Scale CSI also allows the use of more node labels for the attacher and provisioner `StatefulSet`. These node labels should be used only if running these `StatefulSets` on specific nodes (for example, highly available infrastructure nodes) is required. Otherwise, the use of a single label, such as `scale=true` for running `StatefulSets` and IBM Spectrum Scale CSI driver `DaemonSet`, is strongly recommended. Nodes that are specifically marked for running `StatefulSet` must be a subset of the nodes that are marked with the `scale=true` label.

Managing node annotations for IBM Spectrum Scale CNSA (optional)

The IBM Spectrum Scale CNSA operator automatically (recommended) adds Kubernetes annotations to the nodes in the OpenShift cluster to designate their specific role with respect to IBM Spectrum Scale; for example, quorum, manager and collector nodes:

- `scale.ibm.com/nodedesc=quorum::`
- `scale.ibm.com/nodedesc=manager::`
- `scale.ibm.com/nodedesc=collector::`

Supported IBM Spectrum Scale node designations are manager, quorum, and collector. To designate a node with more than one value, add a dash in between the designations, as shown in the following example:

```
scale.ibm.com/nodedesc=quorum-manager-collector::
```

Node annotations can be viewed by issuing the `oc describe <node>` command.

Automatic node designations that are performed by the IBM Spectrum Scale operator are recommended. For manual node designations with annotations, see IBM Documentation.

You can manually add or remove node annotations. To add node annotations, run the following command:

```
# oc annotate node <node name> scale.ibm.com/nodedesc=quorum-manager::
```

To remove node annotations, run the following command:

```
# oc annotate node <node name> scale.ibm.com/nodedesc-
```

Specifying pod tolerations for IBM Spectrum Scale CSI (optional)

In the `csiscaleoperators.csi.ibm.com_cr.yaml` for IBM Spectrum Scale CSI, you also can specify Kubernetes tolerations that are applied to IBM Spectrum Scale CSI pods (see Example 2-7).

Example 2-7 Specifying Kubernetes tolerations

```
# Array of tolerations that will be distributed to CSI pods. Please refer to official
k8s documentation for your environment for more details. #
https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/
# ===============================================================================
# tolerations:
# - key: "key1"
# operator: "Equal"
# value: "value1"
# effect: "NoExecute"
# tolerationSeconds: 3600
```

3

Installing Cloud Pak for Data with IBM Spectrum Scale Container Native Storage Access

In this chapter, we describe creating the IBM Spectrum Scale storage classes. Then, we discuss the processes for using the IBM Spectrum Scale storage classes when installing IBM Cloud Pak for Data and the various service assemblies that are within IBM Cloud Pak for Data.

This chapter includes the following topics:

- 3.17, “Installing Apache Spark” on page 100
- 3.18, “Installing Db2 Data Management Console” on page 101

3.1 Overview

The following lists the official IBM resources for installing IBM Cloud Pak for Data v3.5.2.

- Install Cloud Pak for Data 3.5 on Red Hat OpenShift
- System requirements for IBM Cloud Pak for Data

The installation process includes the following major tasks:

- Providing a storage class (SC) for dynamic provisioning to be used by IBM Cloud Pak for Data.
- Configuring to prepare the Red Hat OpenShift cluster and nodes for IBM Cloud Pak for Data.
- Installing IBM Cloud Pak for Data control plane (`lite` assembly).
- Installing other services from the IBM Cloud Pak for Data catalog, including the following examples:
  - Db2 (`db2oltp` assembly)
  - Watson Studio (`wsl` assembly)
  - Watson Knowledge Catalog (`wkc` assembly)
  - DataStage (`ds` assembly)
  - Db2 Warehouse (`db2wh` assembly)
  - Watson Machine Learning (`wml` assembly)
  - Watson OpenScale (`aiopenscale` assembly)
  - Data Virtualization (`dv` assembly)
  - Data Management Console (`dmc` assembly)
  - Apache Spark (`spark` assembly)

IBM Cloud Pak for Data v3.5.2 supports and is optimized for the following storage providers in OpenShift v4.6:

- Red Hat OpenShift Container Storage, Version 4.5 or later fixes
- Network File System (NFS), Version 4
- Portworx, Version 2.6.2 or later is required for Red Hat OpenShift Version 4.5 and 4.6
- IBM Cloud File Storage
- IBM Spectrum Scale Container Native, Version 5.1.0.3 or later

In this deployment, we use IBM Spectrum Scale CNSA v5.1.0.3 with IBM Spectrum Scale CSI driver v2.1.0 as the storage provider for the IBM Cloud Pak for Data installation.

3.2 Providing an IBM Spectrum Scale Storage Class for Cloud Pak for Data

These steps assume that Red Hat OpenShift Container Platform v4.6 and IBM Spectrum Scale Container Native Storage Access (CNSA) and the IBM Spectrum Scale CSI driver were installed on the cluster.

In this deployment, we provide an SC that is backed by IBM Spectrum Scale that is used as input parameter and storage provider for the installation of IBM Cloud Pak for Data. Based on this SC, IBM Cloud Pak for Data uses dynamic provisioning to satisfy all storage requests of its components automatically. All required persistent volumes (PVs) automatically are created and bound to the related persistent volume claims (PVCs) of these components on-demand.

In this section, we describe how to create the SC with the IBM Spectrum Scale CSI driver. With IBM Spectrum Scale CNSA, we also must consider the specific configuration where the IBM Spectrum Scale file system on the local CNSA compute cluster is remotely mounted (or cross mounted) from a remote IBM Spectrum Scale storage cluster. For more information, see IBM Documentation.

3.2.1 Supported storage classes with IBM Spectrum Scale CSI

The IBM Spectrum Scale CSI driver supports the following three options to create an SC:

- Directories in IBM Spectrum Scale (lightweight volumes)
- Fileset-based volumes with:
  - independent filesets
  - dependent filesets

The YAML manifests for these three types look as shown in the following examples:

- Lightweight volumes:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ibm-spectrum-scale-csi-lt
provisioner: spectrumscale.csi.ibm.com
parameters:
  volBackendFs: "<filesystem name on local CNSA cluster>"
  volDirBasePath: "pvfileset/lwdir"
reclaimPolicy: Delete
```

- Fileset-based volumes (independent fileset):

```
apiVersion: storage.k8s.io/v1
kind: StorageClass metadata:
  name: ibm-spectrum-scale-csi-fileset
provisioner: spectrumscale.csi.ibm.com
parameters:
  volBackendFs: "<filesystem name on local CNSA cluster>"
  clusterId: "<cluster ID of remote storage cluster>"
  uid: "1000"
  gid: "1000"
reclaimPolicy: Delete
```

- Fileset-based volumes (dependent fileset):

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ibm-spectrum-scale-csi-fileset-dependent
provisioner: spectrumscale.csi.ibm.com
parameters:
  volBackendFs: "<filesystem name on local CNSA cluster>"
  clusterId: "<cluster ID of remote storage cluster>"
  uid: "1000"
  gid: "1000"
  filesetType: "dependent"
  parentFileset: "independent-fileset-fset1"
reclaimPolicy: Delete
```

The uid/gid stanzas are optional. The YAML manifests include the following fields:

- `volBackendFs`

  The name of the file system under which the fileset is created. The file system name is the name of the remotely mounted file system on the primary cluster.

- `clusterId`

  Cluster ID of the owning cluster (in our example, the remote storage cluster).

- `uid`

  The uid or username that is assigned to the fileset. This field is optional. The uid or gid must exist on the IBM Spectrum Scale GUI node of the accessing and owning clusters. The default value is `root`.

- `parentFileset`

  The parent fileset name. It is valid with `filesetType=dependent`. The default value is `root`.

The following fields are optional:

- `gid`

  The gid or group name that is assigned to the fileset. The gid or group name must exist on the IBM Spectrum Scale GUI node of the accessing and owning clusters. The default value is `root`.

- `filesetType`

  The default is independent.

- `inodeLimit`

  The inode limit for the fileset. This field is valid with `filesetType=independent`. If not specified, inodeLimit is calculated by using formula Volume Size / Block size of file system.

For more information about SCs and their options, see IBM Documentation.

3.3 Selecting an IBM Spectrum Scale storage class

This section describes how to select an IBM Spectrum Scale SC for IBM Cloud Pak for Data.

3.3.1 IBM Spectrum Scale Storage Class with default uid and gid settings

In this deployment, we use the following default SC ibm-spectrum-scale-sc for dynamic provisioning of PVs that are based on independent filesets in IBM Spectrum Scale, as shown in the following example:

```
# cat ibm-spectrum-scale-sc.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ibm-spectrum-scale-sc
provisioner: spectrumscale.csi.ibm.com
parameters:
  volBackendFs: "<filesystem name on local CNSA cluster>"
  clusterId: "<cluster ID of remote storage cluster>"
reclaimPolicy: Delete
```

The IBM Spectrum Scale file system that is backing this SC (`volBackendFs`) on the remote storage cluster is created with a block size of 1 MiB, which is generally recommended for best performance with ESS 3000. This block size is less than the default block size of 4 MiB, which is used if the block size parameter is not specified during file system creation (`mmcrfs -B [block size]`) process.

For this deployment, a small block size of 1 MiB also proves to be a recommended choice because IBM Spectrum Scale CSI v2.1.0 has some other dependencies on this parameter.

If the **`inodeLimit`** (another optional parameter in the SC) is not specified, the maximum number of inodes per fileset that is backing the PV is calculated based on the following formula:

```
PV size / IBM Spectrum Scale block size
```

With large IBM Spectrum Scale block sizes (up to 16 MiB) and many small files on the PV, this setup can easily lead to “no space left on device” errors because of a lack of enough available inodes on the fileset that is backing the PV.

The **`inodeLimit`** parameter should be verified and set to at least 10 K (4.4 K is required for Cloud Pak for Data base installation) after the `user-home-pv` is created and bound during the installation process. For more information about the workaround, see “Installing the control plane with cpd-cli” on page 65.

Cloud Pak for Data also created PVs as small as 1 MiB, which can lead to attempts to create filesets and quotas that are smaller to the IBM Spectrum Scale block size.

Here, we intentionally do not specify a specific uid or gid, which default to `root.` Typically, pods and containers even run with various different uids, and selecting a specific uid (or gid) typically is not required.

Also, uids for the pods often are arbitrarily assigned by OpenShift based on a range as specified in the `restricted` SCC that applies to a regular user on OpenShift (for example, in the UID range of 1xxxxxxxxx).

A pod with an arbitrarily assigned uid can read and write to the PV, even if the directory in IBM Spectrum Scale is owned by `drwxrwx--x. root root` because the pod is assigned to the gid root that is granting read/write access to the PV.

This SC proved to work well with the installation of the following IBM Cloud Pak for Data assemblies:

- Control plane (`lite`)
- Db2 (`db2oltp`)
- Watson Studio (`wsl`)
- Data Management Console (`dmc`)
- Watson OpenScale (`aiopenscale`)
- Watson Studio (`wsl`)
- Watson Machine Learning (`wml`)
- Data Virtualization (`dv`)

However, this SC with default uid and gid settings failed with the installation of the following IBM Cloud Pak for Data services:

- Watson Knowledge Catalog (`wkc` assembly)
- DataStage (`ds` assembly)

In both cases, the same `is-en-conductor-0` pod of the sub-component `iis` of this installation failed. This sub-component uses an `initContainer` in its pod with a security context that specifically sets the uid to `10032`. By doing so, this pod loses access to the directory that is backing the PV because it also is assigned an arbitrary gid of `1000` by OpenShift in the process (depending on the SCC that is applied). With a uid of `10032` and gid other than `root`, the pod then loses access permission to read/write in the directory in IBM Spectrum Scale backing the PV.

Pods not enforcing a uid also do not automatically receive a non-root gid that is assigned in the process; therefore, they do not lose access to the directory backing the PV. Although the uid is assigned by OpenShift to an arbitrary uid from a predefined range in the 1xxxxxxxxx range as specified by the SCC (for example, MustRunAsRange), the gid is `root`.

This assignment of root grid is the standard case and enables the pod to have read/write access to the directory that is backing the PV (which is set to **`drwxrwx--x root root`**).

3.3.2 IBM Spectrum Scale Storage Class with dedicated uid setting

We successfully installed the IBM Cloud Pak for Data *Watson* Knowledge Catalog and DataStage services with IBM Spectrum Scale as storage provider by using an SC (`ibm-spectrum-scale3-sc`) with a dedicated uid of `10032`, as shown in the following example:

```
# cat ibm-spectrum-scale3-sc.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ibm-spectrum-scale3-sc
provisioner: spectrumscale.csi.ibm.com
parameters:
  volBackendFs: "<filesystem name on local CNSA cluster>"
clusterId: "<cluster ID of remote storage cluster>"
  uid: "10032"
reclaimPolicy: Delete
```

By using the dedicated uid of 10032, it meets the intrinsic requirements of the `is-en-conductor-0` pod in the `iis` sub-component of the DataStage service, which uses the following security context in the `initContainer` section:

```
securityContext:
  allowPrivilegeEscalation: true
  capabilities:
    drop:
    - MKNOD
privileged: false
readOnlyRootFilesystem: false
runAsNonRoot: true
runAsUser: 10032
```

3.4 Testing default IBM Spectrum Scale storage class with regular user

In this section, we briefly demonstrate that a regular user (that is, a non-admin user) in OpenShift can create and access a PV from the default `ibm-spectrum-scale-sc` SC with no specific uid or gid settings defined (defaulting to uid `root` and gid `root`) when no specific uid or gid settings are enforced on the pod:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ibm-spectrum-scale-sc
provisioner: spectrumscale.csi.ibm.com
parameters:
  volBackendFs: "fs0"
  clusterId: "215057217487177715"
reclaimPolicy: Delete
```

By using a PVC.

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: ibm-spectrum-scale-pvc
spec:
  storageClassName: ibm-spectrum-scale-sc
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
```

The PVC created the following PV:

```
# oc get pvc
NAME                   STATUS    VOLUME                                     CAPACITY   ACCESS MODES
ibm-spectrum-scale-pvc Bound     pvc-66cc6cfa-86de-4180-8a8a-b42de185cfc0   1Gi        RWX
```

Running the test pod as shown in the follow:

```
apiVersion: v1
kind: Pod
metadata:
  name: ibm-spectrum-scale-test-pod
spec:
  containers:
    - name: ibm-spectrum-scale-test-pod
      image: alpine:latest
      command: [ "/bin/sh", "-c", "--" ]
      args: [ "while true; do echo $(hostname) $(date +%Y%m%d-%H:%M:%S) | tee -a /data/stream1.out ;
sleep 5 ; done;" ]
      volumeMounts:
        - name: vol1
          mountPath: "/data"
  volumes:
    - name: vol1
      persistentVolumeClaim:
        claimName: ibm-spectrum-scale-pvc
```

The data is created in the PV of the pod by a user (in this example, with an arbitrarily assigned uid of 1000600000 by OpenShift and a gid of root):

```
# oc get pod
NAME READY STATUS RESTARTS AGE
ibm-spectrum-scale-test-pod 1/1 Running 0 14s
# oc rsh ibm-spectrum-scale-test-pod
~ $ whoami
1000600000
~ $ df -h
Filesystem              Size        Used Available Use% Mounted on
fs0                     1.0G        0        1.0G   0% /data
~ $ ls -al /data/
total 1
drwxrwx--x    2 root     root          4096 Feb 15 12:07 .
drwxr-xr-x    1 root     root            40 Feb 15 12:07 ..
-rw-r--r--    1 10006000 root           230 Feb 15 12:07 stream1.out
~ $ cat /data/stream1.out
ibm-spectrum-scale-test-pod 20210215-12:07:16
ibm-spectrum-scale-test-pod 20210215-12:07:21
ibm-spectrum-scale-test-pod 20210215-12:07:26
ibm-spectrum-scale-test-pod 20210215-12:07:31
ibm-spectrum-scale-test-pod 20210215-12:07:36
ibm-spectrum-scale-test-pod 20210215-12:07:41
ibm-spectrum-scale-test-pod 20210215-12:07:47
ibm-spectrum-scale-test-pod 20210215-12:07:52
```

The data is in IBM Spectrum Scale on the remote storage cluster, as shown in the following example with uid 1000600000 and gid `root`:

```
# ls -al
/gpfs/ess3000_1M/pvc-66cc6cfa-86de-4180-8a8a-b42de185cfc0/pvc-66cc6cfa-86de-4180-8
a8ab42de185cfc0-
data/
total 2
drwxrwx--x. 2 root root 4096 Feb 15 13:07 .
drwxrwx--x. 3 root root 4096 Feb 15 13:06 ..
-rw-r--r--. 1 1000600000 root 1426 Feb 15 13:09 stream1.out
# cat
/gpfs/ess3000_1M/pvc-66cc6cfa-86de-4180-8a8a-b42de185cfc0/pvc-66cc6cfa-86de-4180-8
a8ab42de185cfc0-
data/stream1.out
ibm-spectrum-scale-test-pod 20210215-12:07:16
ibm-spectrum-scale-test-pod 20210215-12:07:21
ibm-spectrum-scale-test-pod 20210215-12:07:26
ibm-spectrum-scale-test-pod 20210215-12:07:31
ibm-spectrum-scale-test-pod 20210215-12:07:36
ibm-spectrum-scale-test-pod 20210215-12:07:41
ibm-spectrum-scale-test-pod 20210215-12:07:47
ibm-spectrum-scale-test-pod 20210215-12:07:52
```

A regular user on OpenShift who is running a pod under the “restricted” security context constraints (SCC) can read and write data to the assigned PV from within the pod, even if the uid is arbitrarily set to 1000600000 because the gid is `root`, which grants read/write access to the directory in IBM Spectrum Scale that backs the PV.

3.5 Defining IBM Spectrum Scale as default storage class in OpenShift

We set the created IBM Spectrum Scale SC `ibm-spectrum-scale-sc` as default SC in the OpenShift cluster. Therefore, every PVC request for storage without a specific referenced SC uses the created `ibm-spectrum-scale-sc` SC as default. This feature can be convenient for the deployment of future applications in which no specific SC is provided.

The following example shows how to set and verify the specific storage class to the default:

```
# oc patch storageclass ibm-spectrum-scale-sc -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
storageclass.storage.k8s.io/ibm-spectrum-scale-sc patched
# oc get sc
NAME                            PROVISIONER                    RECLAIMPOLICY VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
ibm-spectrum-scale-internal     kubernetes.io/no-provisioner   Delete        WaitForFirstConsumer   false                  7d16h
ibm-spectrum-scale-sc (default) spectrumscale.csi.ibm.com      Delete        Immediate              false                  2d3h
```

If required, you can remove the configured default SC again by using the following command:

```
# oc patch storageclass ibm-spectrum-scale-sc -p '{"metadata": {"annotations":
{"storageclass.kubernetes.io/is-default-class":"false"}}}'
storageclass.storage.k8s.io/ibm-spectrum-scale-sc patched
```

3.6 Identifying when inodes are exhausted with storage class for Cloud Pak for Data

In this section, an example of how running out of inodes on a fileset-backed SC can be debugged and identified.

When installing Cloud Pak for Data 3.5.2 on OpenShift 4.6.42, the installation of the Cloud Pak for Data control plane (`lite` assembly) failed with the SC `ibm-spectrum-scale-sc`, which uses filesets in IBM Spectrum Scale. The deployment of the `lite` assembly of Cloud Pak for Data with this SC failed, as shown in Example 3-1.

Example 3-1 Failed deployment

```
[ERROR] [2021-04-09 06:18:20-0551] Exiting due to error (Module 0020-core x86_64 has failed).
Please
check /root/cpd-cli-workspace/logs/CPD-2021-04-09T06-04-15.log for details
[ERROR] 2021-04-09T06:18:20.554133Z Execution error: exit status 1
```

The control plane installation is left in the state that is shown in Example 3-2.

Example 3-2 Control plane installation state

```
# ./cpd-cli status -a ${ASSEMBLY} -n ${PROJECT}
Status for assembly lite and relevant modules in project zen-manual:
Assembly Name                   Status          Version         Arch
lite                            Not Ready/Failed 3.5.2          x86_64
   Module Name                  Version         Arch            Storage Class
   0010-infra                   3.5.2           x86_64          ibm-spectrum-scale-sc
   0015-setup                   3.5.2           x86_64          ibm-spectrum-scale-sc
   0020-core                    3.5.2           x86_64          ibm-spectrum-scale-sc
```

Only the following pods were deployed:

```
oc get pods
NAME                                 READY     STATUS    RESTARTS AGE
cpd-install-operator-fcd585775-b4zrv 1/1      Running    0        43m
dsx-influxdb-75d46-mdwcp             1/1      Running    0        41m
dsx-influxdb-set-auth-w57qz          0/1      Completed  0        41m
ibm-nginx-5b6c878d4b-9qnmr           1/1      Running    0        37m
ibm-nginx-5b6c878d4b-rbmfr           1/1      Running    0        37m
icpd-till-5ff49f5c99-sfkfg           1/1      Running    0        43m
pv-prep-job-fncv5                    0/1      Completed  0        41m
usermgmt-bd688dfbc-25vh6             1/1      Running    2        41m
usermgmt-bd688dfbc-hrgjg             1/1      Running    2        41m
zen-metastoredb-0                    1/1      Running    0        41m
zen-metastoredb-1                    1/1      Running    0        41m
zen-metastoredb-2                    1/1      Running    0        41m
zen-metastoredb-init-th5m2           0/1      Completed  0        41m
```

While the deployment is running, we see that the following job `zen-pre-requisite-job-sxssv` fails:

```
[root@arcx3650fxxnh ~]# oc get pods -w
NAME                          READY  STATUS          RESTARTS AGE
cpd-install-operator-fcd585775-b4zrv 1/1 Running    0        11m
dsx-influxdb-75d46-mdwcp      1/1    Running         0        9m2s
dsx-influxdb-set-auth-w57qz   0/1    Completed       0        9m2s
ibm-nginx-5b6c878d4b-9qnmr    1/1    Running         0        5m24s
ibm-nginx-5b6c878d4b-rbmfr    1/1    Running         0        5m24s
icpd-till-5ff49f5c99-sfkfg    1/1    Running         0        11m
pv-prep-job-fncv5             0/1    Completed       0        9m2s
usermgmt-bd688dfbc-25vh6      1/1    Running         2        9m2s
usermgmt-bd688dfbc-hrgjg      1/1    Running         2        9m2s
zen-metastoredb-0             1/1    Running         0        9m2s
zen-metastoredb-1             1/1    Running         0        9m2s
zen-metastoredb-2             1/1    Running         0        9m2s
zen-metastoredb-init-th5m2    0/1    Completed       0        9m2s
zen-pre-requisite-job-sxssv   0/1    CrashLoopBackOff 5       4m13s
```

The last state, reason, and exit code are shown in Example 3-3.

Example 3-3 Last state, reason, and exit code

```
Containers:
  zen-pre-requisite-job:
    Container ID:    cri-o://679a1a73f85d5cc47a4dc5e78237153916ab83656ed61b52532c8aaa51e5e0ce
    Image:      image-registry.openshift-image-registry.svc:5000/zen-manual/icpd-requisite:3.5.2-
x86_64-97
    Image ID:     image-registry.openshift-image-registry.svc:5000/zen-manual/icpd-
requisite@sha256:7ee5abde4677ce7139d01f74b993cf5cc35f891b4311a66f77cb7c490544c0d8
   Port:        <none>
   Host Port:    <none>
   State:       Waiting
     Reason:    CrashLoopBackOff
   Last State: Terminated
     Reason: Error
     Exit Code: 1
     Started: Fri, 09 Apr 2021 06:14:42 -0700
```

```
  Finished: Fri, 09 Apr 2021 06:14:42 -0700
Ready: False
Restart Count: 5
```

The following error message is the only such message that we can retrieve:

```
[root@arcx3650fxxnh ~]# oc logs zen-pre-requisite-job-sxssv
cp: cannot create regular file '/user-home/_global_/tmp/./cacerts': No space left on device
```

The volume backing the data for the pod is `user-home-pvc`:

```
[root@arcx3650fxxnh ~]# oc get pvc
NAME                         STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS            AGE
cpd-install-operator-pvc     Bound    pvc-3334785d-0e4a-4551-ab36-ac5c418a32b2   1Gi        RWX            ibm-spectrum-scale-sc   10m
cpd-install-shared-pvc       Bound    pvc-293cca5a-8f60-4c91-8eab-36df481dfd85   1Gi        RWX            ibm-spectrum-scale-sc   10m
datadir-zen-metastoredb-0    Bound    pvc-918b8d40-d4be-495f-b0a0-e9eb8f8c3667   10Gi       RWO            ibm-spectrum-scale-sc   8m18s
datadir-zen-metastoredb-1    Bound    pvc-086aa51d-6074-40f1-89f6-f0db21706a99   10Gi       RWO            ibm-spectrum-scale-sc   8m18s
datadir-zen-metastoredb-2    Bound    pvc-799f0c4f-dda2-446a-b35b-2b99577a205b   10Gi       RWO            ibm-spectrum-scale-sc   8m18s
influxdb-pvc                 Bound    pvc-82dc93e7-bf1e-40fa-a2ea-1f1fa6b13179   10Gi       RWX            ibm-spectrum-scale-sc   8m18s
user-home-pvc                Bound    pvc-dd886cbd-cc00-4df8-af3b-217bdbf7846e   10Gi       RWX            ibm-spectrum-scale-sc   8m18s
```

The data in it looks as shown in Example 3-4 in IBM Spectrum Scale.

Example 3-4 Data format

```
# ls -al
/ibm/fs1/pvc-dd886cbd-cc00-4df8-af3b-217bdbf7846e/pvc-dd886cbd-cc00-4df8-af3b-217bdbf7846e-data/
total 6
drwxr-xr-x. 11 1000321000 root     4096 Jan 19 15:57 .
drwxrwx--x.  3 root       root     4096 Apr  9 07:34 ..
drwxr-xr-x.  3 1000321000 stgadmin 4096 Apr  9 07:35 1000330999
drwxr-xr-x.  2 1000321000 stgadmin 4096 Jan 19 15:57 _assets_approvals_
drwxr-xr-x. 20 1000321000 stgadmin 4096 Jan 19 15:57 _global_
drwxr-xr-x.  3 1000321000 stgadmin 4096 Jan 19 15:57 masterRepos
drwxr-xr-x.  2 1000321000 stgadmin 4096 Jan 19 15:57 .tmp
drwxr-xr-x.  4 1000321000 stgadmin 4096 Jan 19 15:57 zen
drwxr-xr-x.  3 1000321000 stgadmin 4096 Jan 19 15:57 _zen-addons
drwxr-xr-x.  3 1000321000 stgadmin 4096 Jan 19 15:57 _zen-admin
drwxr-xr-x.  4 1000321000 stgadmin 4096 Jan 19 15:57 _zen-content
```

Solution A: Using light-weight storage class

A `No space left on device` error is a typical sign for running out of inodes (rather than storage capacity) on the specific fileset that is backing the PV.

Here, we use a light-weight SC (`ibm-spectrum-scale-lt`) as an alternative example to demonstrate how to provide a higher number of inodes to the PVs without the need to specify a dedicated inodeLimit in the SC. Generally, we recommend reviewing Solution B and defining a suitable inodeLimit when needed:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ibm-spectrum-scale-lt
provisioner: spectrumscale.csi.ibm.com parameters:
  volBackendFs: "fs1"
  volDirBasePath: "cpd-volumes"
reclaimPolicy: Delete
```

Then, you create the directory `cpd-volumes` in IBM Spectrum Scale by using the following commands:

```
# mkdir /ibm/fs1/cpd-volumes
# chmod a+rwx /ibm/fs1/cpd-volumes
# ls -al /ibm/fs1/
drwxr-xr-x. 4 root root 262144 Apr 9 08:20 .
drwxr-xr-x. 6 root root 69     Apr 8 15:32 ..
drwxrwxrwx. 2 root root 4096   Apr 9 08:20 cpd-volumes
```

This directory holds all the backing directories for the PVCs. Now, the installation of the Cloud Pak for Data control plane succeeds, as shown in Example 3-5.

Example 3-5 Successful installation of Cloud Pak for Data control plane

```
ASSEMBLY="lite"
ARCH="x86_64"
PROJECT="zen-manual"
STORAGE_CLASS="ibm-spectrum-scale-lt"
PUSHREGISTRY="$(oc registry info)"
PULLREGISTRY="image-registry.openshift-image-registry.svc:5000"

# ./cpd-cli install -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} -c ${STORAGE_CLASS} --arch ${ARCH} \
  --transfer-image-to=${PUSHREGISTRY}/${PROJECT} --target-registry-username=$(oc whoami) \
  --target-registry-password=$(oc whoami -t) --insecure-skip-tls-verify \
  --cluster-pull-prefix=image-registry.openshift-image-registry.svc:5000/${PROJECT} \
  --latest-dependency --accept-all-licenses
```

A status of Ready is shown for all modules of the `lite` assembly (see Example 3-6).

Example 3-6 Ready status for all modules

```
[root@arcx3650fxxnh ~]# ./cpd-cli status -a ${ASSEMBLY} -n ${PROJECT} Displaying CR status for
all assemblies and relevant modules
Status for assembly lite and relevant modules in project zen-manual:

Assembly Name                           Status           Version          Arch
lite                                    Ready            3.5.2            x86_64

  Module Name                           Version          Arch             Storage Class
  0010-infra                            3.5.2            x86_64           ibm-spectrum-scale-lt
  0015-setup                            3.5.2            x86_64           ibm-spectrum-scale-lt
  0020-core                             3.5.2            x86_64           ibm-spectrum-scale-lt
```

The data for the volume `user-home-pvc` (`pvc-1a091254-cc80-4989-92cd-653195cc53f5`) is shown in the last line of the following example:

```
# oc get pvc
NAME                         STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS            AGE
cpd-install-operator-pvc     Bound    pvc-d9584544-73aa-44ea-93fd-e1f5cda9355a   1Gi        RWX            ibm-spectrum-scale-lt   22m
cpd-install-shared-pvc       Bound    pvc-6751d353-9896-4f87-a819-6e8ad1c62ce8   1Gi        RWX            ibm-spectrum-scale-lt   22m
datadir-zen-metastoredb-0    Bound    pvc-e1f3ec97-c528-48f8-99fc-654e2d7efea4   10Gi       RWO            ibm-spectrum-scale-lt   19m
datadir-zen-metastoredb-1    Bound    pvc-4b36fe37-0caf-4f3b-982d-872090920a7d   10Gi       RWO            ibm-spectrum-scale-lt   19m
datadir-zen-metastoredb-2    Bound    pvc-88575f9e-5a31-4848-be87-301868035c1f   10Gi       RWO            ibm-spectrum-scale-lt   19m
influxdb-pvc                 Bound    pvc-6fb24ede-d992-43ab-81ef-25bf981921b5   10Gi       RWX            ibm-spectrum-scale-lt   19m
user-home-pvc                Bound    pvc-1a091254-cc80-4989-92cd-653195cc53f5   10Gi       RWX            ibm-spectrum-scale-lt   19m
```

The volume looks in IBM Spectrum Scale as shown in Example 3-7.

Example 3-7 Volume format in IBM Spectrum Scale

```
# ls -al /ibm/fs1/cpd-volumes/
total 260
drwxrwxrwx.  9 root       root       4096 Apr  9 08:27 .
drwxr-xr-x.  4 root       root     262144 Apr  9 08:20 ..
drwxr-xr-x. 11 1000321000 root       4096 Jan 19 15:57 pvc-1a091254-cc80-4989-92cd-653195cc53f5
drwxr-xr-x.  4 1000321000 root       4096 Apr  9 08:42 pvc-4b36fe37-0caf-4f3b-982d-872090920a7d
drwxrwsr-x.  3 1000321000 stgadmin   4096 Apr  9 08:27 pvc-6751d353-9896-4f87-a819-6e8ad1c62ce8
drwxr-xr-x.  5 1000321000 root       4096 Apr  9 08:28 pvc-6fb24ede-d992-43ab-81ef-25bf981921b5
drwxr-xr-x.  4 1000321000 root       4096 Apr  9 08:42 pvc-88575f9e-5a31-4848-be87-301868035c1f
drwxrwsr-x.  5 1000321000 stgadmin   4096 Apr  9 08:27 pvc-d9584544-73aa-44ea-93fd-e1f5cda9355a
drwxr-xr-x.  4 1000321000 root       4096 Apr  9 08:42 pvc-e1f3ec97-c528-48f8-99fc-654e2d7efea4

[root@stg-node0 ~]# ls -al /ibm/fs1/cpd-volumes/pvc-1a091254-cc80-4989-92cd-653195cc53f5/
total 6
drwxr-xr-x. 11 1000321000 root     4096 Jan 19 15:57 .
drwxrwxrwx.  9 root       root     4096 Apr  9 08:27 ..
drwxr-xr-x.  3 1000321000 stgadmin 4096 Apr  9 08:28 1000330999
drwxr-xr-x.  2 1000321000 stgadmin 4096 Jan 19 15:57 _assets_approvals_
drwxr-xr-x. 20 1000321000 stgadmin 4096 Jan 19 15:57 _global_
drwxr-xr-x.  3 1000321000 stgadmin 4096 Jan 19 15:57 masterRepos
drwxr-xr-x.  2 1000321000 stgadmin 4096 Jan 19 15:57 .tmp
drwxr-xr-x.  4 1000321000 stgadmin 4096 Jan 19 15:57 zen
drwxr-xr-x.  3 1000321000 stgadmin 4096 Jan 19 15:57 _zen-addons
drwxr-xr-x.  3 1000321000 stgadmin 4096 Jan 19 15:57 _zen-admin
drwxr-xr-x.  5 1000321000 stgadmin 4096 Jan 19 15:57 _zen-content
```

With light-weight provisioned volumes, we share the inode space of the underlying IBM Spectrum Scale file system (`ess3000_1M`; here, `15490304` maxInodes) without the need to specify a dedicated inodeLimit in the SC.

Solution B: Increase number of inodes in fileset based storage class

A `No space left on device` error is a typical sign for running out of inodes (rather than storage capacity) on the specific fileset backing the PV.

The default `ibm-spectrum-scale-sc` SC also can be extended to specify a fixed number of inodes for the filesets that are backing the PVs by using the **`inodeLimit`** parameter, as shown in the following example:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass metadata:
  name: ibm-spectrum-scale-sc
provisioner: spectrumscale.csi.ibm.com
parameters:
  volBackendFs: "fs0"
  clusterId: "215057217487177715"
  inodeLimit: "1000000"
reclaimPolicy: Delete
```

The parameter is valid only with `filesetType=independent`, which is the default setting if it is not specified otherwise.

If the **inodeLimit** is omitted, a default value is applied, which is calculated by using the following formula as described in SC v2.1.0 (for more information, see IBM Documentation):

```
(volume size of the PV / block size of the file system)
```

This setting changed with the release of CSI 2.1.0. In CSI 2.0.0 the default was an **inodeLimit** of one million.

In this deployment of Cloud Pak for Data, we used an IBM Spectrum Scale file system with a block size of 1MiB:

```
# mmlsfs ess3000_1M -B
flag                value                    description
------------------- ------------------------ -----------------------------------
 -B                 1048576                  Block size
```

The installation of the Cloud Pak for Data control plane (`lite` assembly) created the following PVs of 1 GiB and 10 GiB PV sizes:

```
# oc get pv
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM
pvc-048b209e-851f-49f7-9482-a5086daab6b0   10Gi       RWO            Delete           Bound    zen/datadir-zen-metastoredb-0
pvc-4fcff6ae-94bc-46e3-b660-e43432dfb941   10Gi       RWO            Delete           Bound    zen/datadir-zen-metastoredb-2
pvc-6748d517-e0de-4543-b2c5-c9c45d00a271   1Gi        RWX            Delete           Bound    zen/cpd-install-shared-pvc
pvc-78ab6afa-8f29-463d-965b-79967c9ec2cf   10Gi       RWX            Delete           Bound    zen/user-home-pvc
pvc-a6851d97-88ac-4eb6-a889-a92a57ce53af   10Gi       RWO            Delete           Bound    zen/datadir-zen-metastoredb-1
pvc-cea71faf-0586-4744-b27d-d67d1a10cd35   1Gi        RWX            Delete           Bound    zen/cpd-install-operator-pvc
pvc-f59ed123-3ced-4864-9f3d-8587e73b467b   10Gi       RWX            Delete           Bound    zen/influxdb-pvc
```

These PVs were backed by the following filesets:

```
# mmlsfileset ess3000_1M -L
Filesets in file system 'ess3000_1M':
Name                            Id      RootInode  ParentId Created                      InodeSpace MaxInodes    AllocInodes
root                             0            3        -- Mon May 11 20:19:22 2020        0  15490304         500736
spectrum-scale-csi-volume-store  1         524291        0 Thu Apr 15 00:15:25 2021        1   1048576          52224
pvc-cea71faf-0586-4744-b27d-d67d1a10cd35 2 1048579        0 Mon Apr 19 10:23:02 2021        2  1024            1024
pvc-6748d517-e0de-4543-b2c5-c9c45d00a271 3 1572867        0 Mon Apr 19 10:23:05 2021        3  1024            1024
pvc-78ab6afa-8f29-463d-965b-79967c9ec2cf 4 2097155        0 Mon Apr 19 10:26:03 2021        4      10240          10240
pvc-f59ed123-3ced-4864-9f3d-8587e73b467b 5 2621443        0 Mon Apr 19 10:26:06 2021        5      10240          10240
pvc-048b209e-851f-49f7-9482-a5086daab6b0 6 3145731        0 Mon Apr 19 10:26:09 2021        6      10240          10240
pvc-a6851d97-88ac-4eb6-a889-a92a57ce53af 7 3670019        0 Mon Apr 19 10:26:12 2021        7      10240          10240
pvc-4fcff6ae-94bc-46e3-b660-e43432dfb941 8 4194307        0 Mon Apr 19 10:26:14 2021        8      10240          10240
```

By using the formula (`[volume size of the PV / block size of the file system]`) with a file system block size of 1 MiB, we can confirm the number of inodes that are associated with each fileset that is backing a PV:

```
file system Blocksize = 1 MiB

Cloud Pak for Data Volumes sizes (control plane/"lite"):
1GiB  =  1024 MiB / 1 MiB = 1024 Inodes
10GiB = 10240 MiB / 1 MiB = 10240 Inodes
```

Using independent filesets to back the PVs allows to pick an independent inode space.

The default (`PV-size/blocksize`) assumes that we have an average file size that is similar to the IBM Spectrum Scale file system block size. It also does not consider that IBM Spectrum Scale still can efficiently save more smaller files by using subblocks. Also, the OpenShift user (or even the OpenShift admin) does not know the block size of the underlying IBM Spectrum Scale file system.

The IBM Spectrum Scale file system block size also is important regarding the smallest PVC volume sizes that are used in a deployment. If the requested PVC volume size is smaller than the file system block size, the provisioning might fail with quota errors. With average file sizes or PVCs as small as 1 MiB (or less), we recommend small file system block sizes; for example, 1 MiB (or less).

In the setup for this publication, we intentionally used 1 MiB as file system block size, which provided a sufficient number of inodes in this case. With the default block size of 4 MiB, we saved only 2560 files in the 10 GiB volumes, which caused the `No space left on device` error.

For more information about changing the default inode calculation for fileset-based SCs with IBM Spectrum Scale CSI, see this web page.

3.7 Installing IBM Cloud Pak for Data

The IBM Cloud Pak for Data requirements are described in System requirements for IBM Cloud Pak for Data. Several pre-installation tasks must be completed. For more information, see this web page (specifically, Changing required node settings in the Cloud Pak for Data documentation.

3.7.1 Local container image registry

The container image registry stores the container images for the Cloud Pak for Data control plane and services. A minimum of 300 GB of storage space is required.

In the following example, we adjust the created NFS volume for the internal OpenShift image registry to 500 GiB:

```
# oc get pv registry-pv -o yaml
apiVersion: v1
kind: PersistentVolume
[...]
spec:
  accessModes:
  - ReadWriteMany
  capacity:
    storage: 500Gi
  claimRef:
    apiVersion: v1
    kind: PersistentVolumeClaim
    name: image-registry-storage
    namespace: openshift-image-registry
    resourceVersion: "48976"
    uid: 1372f514-646d-4864-84de-7b7928ab43d6
  nfs:
    path: /data2/registry
    server: 192.168.1.1
  persistentVolumeReclaimPolicy: Retain
  volumeMode: Filesystem
status:
  phase: Bound
```

3.7.2 Local storage for container images

Each node on your cluster must include local storage for the container images that are running on that node. In Version 4.6, local copies of the images are stored in the `/var/lib/containers` directory. A minimum of 300 GB of storage space is required.

Because CoreOS uses the entire boot disk as the root file system, we must ensure that a large enough boot disk is used when installing OpenShift:

```
[core@worker02 ~]$ df -h /var/lib/containers
Filesystem                            Size  Used Avail Use% Mounted on
/dev/mapper/coreos-luks-root-nocrypt  446G   28G  418G   7% /var
```

In the following example, we use a 446 GiB boot disk on all nodes:

```
[core@worker02 ~]$ lsblk
NAME                         MAJ:MIN RM    SIZE  RO TYPE MOUNTPOINT
sda                            8:0    0  446.1G  0 disk
|--sda1                        8:1    0    384M  0 part /boot
|--sda2                        8:2    0    127M  0 part /boot/efi
|--sda3                        8:3    0      1M  0 part
|--sda4                        8:4    0  445.6G  0 part
  |--coreos-luks-root-nocrypt   253:0    0  445.6G  0 dm   /sysroot
```

3.7.3 Shared persistent storage for services

The Cloud Pak for Data control plane and services store data in shared persistent storage. The platform supports the following types of shared storage:

- Red Hat OpenShift Container Storage: Version: 4.5 or later fixes
- Network File System (NFS): Version 4
- Portworx: Version 2.6.2 or later is required for Red Hat OpenShift Version 4.5 and 4.6
- IBM Cloud File Storage
- IBM Spectrum Scale Container Native, Version 5.1.0.3 or later

The minimum amount of storage depends on the type of storage that you plan to use. For more information, see this web page.

As a general rule, Cloud Pak for Data with all services installed can use up to 700 GB of storage space. Review the this web page to ensure that you have sufficient storage space available for user data that is based on the type of storage that you select. You can add capacity depending on your user data volume requirements.

The Cloud Pak for Data control plane supports all of the shared persistent storage types that are supported by the platform. When you install the control plane, you must specify the suitable SC:

- OpenShift Container Storage: Required SC `ocs-storagecluster-cephfs`
- NFS: Specify an SC with ReadWriteMany (RWX) access.
- Portworx: Required SC `portworx-shared-gp3`
- IBM Cloud File Storage: Supported SCs:
  - `ibmc-file-gold-gid`
  - `ibm-file-custom-gold- gid`
- IBM Spectrum Scale Container Native: ibm-spectrum-scale-sc

3.7.4 Disk requirements

To prepare your storage disks, ensure that you have good I/O performance, and prepare the disks for encryption. IBM Cloud Pak for Data defines minimum performance requirements for the persistent storage provider in OpenShift as described in Disk requirements and Checking I/O performance for IBM Cloud Pak for Data. It provides the following simple tests that can be run from within a pod to ensure sufficient disk I/O performance:

- Disk latency test

  The value must be comparable to or better than: 2.5 MBps, as shown in the following example:

  ```
  dd if=/dev/zero of=/PVC_mount_path/testfile bs=4096 count=1000 oflag=dsync
  ```

- Disk throughput test

  The value must be comparable to or better than: 209 MBps, as shown in the following example:

  ```
  dd if=/dev/zero of=/PVC_mount_path/testfile bs=1G count=1 oflag=dsync
  ```

These tests can be run without any other effort as a Kubernetes Job in OpenShift on the `ibm-spectrum-scale-sc` SC, as shown in the following examples:

- Disk latency Job:

```
# cat ibm-spectrum-scale-cp4d-latency-test-job.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: ibm-spectrum-scale-cp4d-latency-test-job
spec:
  template:
    spec:
      containers:
        - name: latency-test
          image: registry.access.redhat.com/ubi8/ubi-minimal
          command: [ "/bin/sh", "-c", "--" ]
          args: [ "echo Running Disk Latency Test on $HOSTNAME at $(date
+%Y%m%d-%H:%M:%S) && for i in {1..5}; do dd if=/dev/zero of=/data/cp4d-testfile
bs=4096 count=1000 oflag=dsync 2>&1|tail -1 ; sleep 5; done;" ]
          volumeMounts:
            - name: vol1
              mountPath: "/data"
      restartPolicy: Never
      volumes:
        - name: vol1
          persistentVolumeClaim:
            claimName: ibm-spectrum-scale-pvc
  backoffLimit: 1
```

- Disk throughput Job:

```
# cat ibm-spectrum-scale-cp4d-throughput-test-job.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: ibm-spectrum-scale-cp4d-throughput-test-job
spec:
  template:
```

```
      spec:
        containers:
          - name: throughput-test
            image: registry.access.redhat.com/ubi8/ubi-minimal
            command: [ "/bin/sh", "-c", "--" ]
            args: [ "echo Running Disk Throughput Test on $HOSTNAME at $(date
  +%Y%m%d-%H:%M:%S) && for i in {1..5}; do dd if=/dev/zero of=/data/cp4d-testfile
  bs=1G count=1 oflag=dsync 2>&1|tail -1 ; sleep 5; done;" ]
            volumeMounts:
              - name: vol1
                mountPath: "/data"
        restartPolicy: Never
        volumes:
          - name: vol1
            persistentVolumeClaim:
              claimName: ibm-spectrum-scale-pvc
    backoffLimit: 1
```

Here, we make use of the PVC:

```
# cat ibm-spectrum-scale-pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: ibm-spectrum-scale-pvc
spec:
  storageClassName: ibm-spectrum-scale-sc
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
```

The Kubernetes Jobs are run with `oc apply` and the results are displayed by using:

```
# oc logs jobs/ibm-spectrum-scale-cp4d-latency-test-job
```

or

```
# oc logs jobs/ibm-spectrum-scale-cp4d-throughput-test-job
```

3.7.5 I/O performance

When I/O performance is insufficient, services can experience poor performance or cluster instability when the services are handling a heavy load, such as functional failures with timeouts.

The following I/O performance requirements are based on repeated workloads that test performance on the platform and validated in various cloud environments. The current requirements are based on the performance of writing data to representative storage locations by using two chosen block sizes (4 KB and 1 GB). These tests use the `dd` command-line utility.

Use the MBps metric from the tests and ensure that your test result is comparable to or better than the targets.

To ensure that the storage partition has good disk I/O performance, you can run the advanced tests that are described next.

Advanced Disk performance tests

Note: If your storage volumes are remote, network speed can be a key factor in your I/O performance. For good I/O performance, ensure that you have sufficient network speed and a storage backend, such as IBM ESS, that can deliver the high performance storage.

To run a throughput test, we created a job file (see Example 3-8), which depends on the registry containing an `fio` container with a suitable Dockerfile.

Example 3-8 Job file

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: fio-seq-w-1-scale-default-sc-pvc
spec:
  storageClassName: scale-default-sc
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
---
apiVersion: v1
kind: Pod
metadata:
  name: fio-seq-w-1-scale-default-sc
spec:
  volumes:
    - name: target
      persistentVolumeClaim:
        claimName: fio-seq-w-1-scale-default-sc-pvc
  containers:
    - name: fio
      image: registry.cpst-lab.no-users.ibm.com/fio
      args:
        - '--name=fio-seq-w-1'
        - '--filename=/target/fio-seq-w-1'
        - '--rw=write'
        - '--bs=1M'
        - '--numjobs=1'
        - '--time_based'
        - '--runtime=300'
        - '--direct=1'
        - '--size=10G'
        - '--ioengine=libaio'
        - '--iodepth=16'
      volumeMounts:
        - mountPath: "/target"
          name: target
  restartPolicy: Never
```

An example of a Dockerfile is shown in Example 3-9.

Example 3-9 Dockerfile example

```
FROM alpine
VOLUME ["/target"]
RUN chmod 0775 /target && apk add fio
ENTRYPOINT ["/usr/bin/fio"]
```

The `fio` output for sequential write with one thread is shown in Example 3-10.

Example 3-10 Sequential write fio output

```
file1: (g=0): rw=write, bs=(R) 1024KiB-1024KiB, (W) 1024KiB-1024KiB, (T) 1024KiB-1024KiB,
ioengine=libaio, iodepth=16
fio-3.27
Starting 1 process
file1: Laying out IO file (1 file / 10240MiB)

file1: (groupid=0, jobs=1): err= 0: pid=56: Thu Jul  1 20:24:52 2021
  write: IOPS=2742, BW=2742MiB/s (2876MB/s)(803GiB/300008msec); 0 zone resets
    slat (usec): min=31, max=30736, avg=104.30, stdev=98.91
    clat (usec): min=1528, max=283454, avg=5727.79, stdev=3938.40
     lat (usec): min=1689, max=283897, avg=5832.39, stdev=3944.61
    clat percentiles (usec):
     |  1.00th=[  1909],  5.00th=[  2040], 10.00th=[  2147], 20.00th=[  2311],
     | 30.00th=[  2769], 40.00th=[  5604], 50.00th=[  6521], 60.00th=[  6980],
     | 70.00th=[  7373], 80.00th=[  7767], 90.00th=[  8586], 95.00th=[  9241],
     | 99.00th=[ 11469], 99.50th=[ 13566], 99.90th=[ 27657], 99.95th=[ 43254],
     | 99.99th=[173016]
   bw (  MiB/s): min=  616, max= 3146, per=100.00%, avg=2743.73, stdev=273.74, samples=599
   iops        : min=  616, max= 3146, avg=2743.64, stdev=273.73, samples=599
  lat (msec)   : 2=3.36%, 4=29.47%, 10=64.68%, 20=2.31%, 50=0.14%
  lat (msec)   : 100=0.02%, 250=0.02%, 500=0.01%
  cpu          : usr=16.09%, sys=13.58%, ctx=611788, majf=0, minf=11640
  IO depths    : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=100.0%, 32=0.0%, >=64=0.0%
     submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
     complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.1%, 32=0.0%, 64=0.0%, >=64=0.0%
     issued rwts: total=0,822729,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=16

Run status group 0 (all jobs):
  WRITE: bw=2742MiB/s (2876MB/s), 2742MiB/s-2742MiB/s (2876MB/s-2876MB/s), io=803GiB (863GB),
run=300008-300008msec
```

3.7.6 Software requirements

IBM Cloud Pak for Data requires the following software:

- Red Hat OpenShift Container Platform Version 4.5 or later fixes for on-premises.
- Red Hat OpenShift Container Platform Version 4.6.1 or later fixes for on-premises.

The OpenShift cluster should include the following components:

- Container runtime: CRI-O version 1.13 or later
- OpenShift version: 4.6, 4.5
- Storage type: NFS, IBM Cloud File Storage, Portworx
- Storage requirement: 200 GB

In the following example, we use OpenShift v4.6.42 with container runtime CRI-O 1.19.3-8:

```
# oc get clusterversion
NAME      VERSION   AVAILABLE   PROGRESSING   SINCE   STATUS
version   4.6.42    True        False         3m42s   Cluster version is 4.6.42

oc get nodes -o wide
NAME                                         STATUS   ROLES   AGE  VERSION        INTERNAL-IP    EXTERNAL-IP
OS-IMAGE                                                     KERNEL-VERSION                 CONTAINER-RUNTIME
master0.cpst-ocp-cluster-b.cpst-lab.no-users.ibm.com  Ready  master  93m  v1.19.0+4c3480d  9.114.195.134  <none>
Red Hat Enterprise Linux CoreOS 46.82.202108022057-0 (Ootpa)   4.18.0-193.60.2.el8_2.x86_64   cri-o://1.19.3-8.rhaos4.6.git 0fa2911.el8
master1.cpst-ocp-cluster-b.cpst-lab.no-users.ibm.com  Ready  master  92m  v1.19.0+4c3480d  9.114.194.59  <none>
Red Hat Enterprise Linux CoreOS 46.82.202108022057-0 (Ootpa)   4.18.0-193.60.2.el8_2.x86_64   cri-o://1.19.3-8.rhaos4.6.git 0fa2911.el8
master2.cpst-ocp-cluster-b.cpst-lab.no-users.ibm.com  Ready  master  92m  v1.19.0+4c3480d  9.114.193.104  <none>
Red Hat Enterprise Linux CoreOS 46.82.202108022057-0 (Ootpa)   4.18.0-193.60.2.el8_2.x86_64   cri-o://1.19.3-8.rhaos4.6.git 0fa2911.el8
worker0.cpst-ocp-cluster-b.cpst-lab.no-users.ibm.com  Ready  worker  41m  v1.19.0+4c3480d  9.114.194.124  <none>
Red Hat Enterprise Linux CoreOS 46.82.202108022057-0 (Ootpa)   4.18.0-193.60.2.el8_2.x86_64   cri-o://1.19.3-8.rhaos4.6.git 0fa2911.el8
worker1.cpst-ocp-cluster-b.cpst-lab.no-users.ibm.com  Ready  worker  41m  v1.19.0+4c3480d  9.114.195.27  <none>
Red Hat Enterprise Linux CoreOS 46.82.202108022057-0 (Ootpa)   4.18.0-193.60.2.el8_2.x86_64 cri-o://1.19.3-8.rhaos4.6.git
```

3.7.7 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 features the required settings for these services, an operating system administrator with root privileges must review and adjust the settings on the suitable nodes in the cluster.

3.7.8 Load balancer timeout settings (HAPROXY)

To prevent connections from being closed before processes complete, you might need to adjust the timeout settings on your load balancer node. The recommended timeout is at least 5 minutes. In some situations, you might need to set the timeout even higher.

On the load balancer node, check the HAProxy timeout settings in the `/etc/haproxy/haproxy.cfg` file.

> **Note:** We recommend that values for the `timeout client` and `timeout server` are set to at least `300s`.

The following default settings for the HAPROXY were used in this environment:

```
# grep timeout /etc/haproxy/haproxy.cfg
    timeout connect         10s
    timeout client          30s
    timeout server          30s
```

For Cloud Pak for Data, we change these setting to the following values:

```
# grep timeout /etc/haproxy/haproxy.cfg
    timeout connect      100s
    timeout client       300s
    timeout server       300s
```

We also adjust the `timeout connect` setting to `100s` to maintain the default relation of one-third of the other two timeout settings.

Restart the service after the change:

```
# systemctl restart haproxy
```

3.7.9 Setting up your local image registry

To install Cloud Pak for Data, you must have a registry server where you can host the images for the Cloud Pak for Data control plane and the services that you want to install. For more information, see this web page.

You can use the internal registry in your Red Hat OpenShift cluster. This option is recommended because it does not require you to manage pull secrets. Ensure that you can access the registry externally (that is, you must set the registry to `managed` state and expose the registry).

Here, we use the internal OpenShift image registry that we prepared and exposed for the IBM Spectrum Scale CNSA v5.1.0.3 installation.

For more information about the internal registry, see this web page.

3.7.10 Adjusting OpenShift CRI-O container settings

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

These settings are required if you are using the CRI-O container runtime:

- `ulimit -n`: The recommended value is at least `66560`.
- `ulimit -u`: The recommended value is at least `12288`.

The `/etc/crio/crio.conf` on the OpenShift 4.6 worker nodes resembles the following example:

```
# The crio.runtime table contains settings pertaining to the OCI runtime used
# and options for how to set up and manage the OCI runtime.
[crio.runtime]

# A list of ulimits to be set in containers by default, specified as
# "<ulimit name>=<soft limit>:<hard limit>", for example:
# "nofile=1024:2048"
# If nothing is set here, settings will be inherited from the CRI-O daemon
#default_ulimits = [
#]
...
# Maximum number of processes allowed in a container.
pids_limit = 1024
```

The `ulimit` on the worker nodes seems to be set to `1048576`, which likely is sufficient, as shown in Example 3-11.

Example 3-11 ulimit setting on worker nodes

```
# oc get nodes -l node-role.kubernetes.io/worker|grep -v NAME|while read a b; do echo "## Node: $a -
ulimit -n = $(oc debug node/$a -- chroot /host ulimit -n 2>/dev/null)" ; done
## Node: worker01.ocp4.scale.com - ulimit -n = 1048576
## Node: worker02.ocp4.scale.com - ulimit -n = 1048576
## Node: worker03.ocp4.scale.com - ulimit -n = 1048576

# oc get nodes -l node-role.kubernetes.io/worker|grep -v NAME|while read a b; do echo "## Node: $a -
ulimit -u = $(oc debug node/$a -- chroot /host ulimit -u 2>/dev/null)" ; done
## Node: worker01.ocp4.scale.com - ulimit -u = 1048576
## Node: worker02.ocp4.scale.com - ulimit -u = 1048576
## Node: worker03.ocp4.scale.com - ulimit -u = 1048576
```

The Cloud Pak for Data documentation demands that if the `ulimit` value is less than `12288`, apply a `cri-o` runtime change by way of `machineConfig` to set it to the following limit:

```
pids_limit = 12288
```

Also, according to Override CRI-O settings, a requirement for Cloud Pak for Data appears to be to set the `pids_limit = 12288`, as shown in the following example:

```
apiVersion: machineconfiguration.openshift.io/v1
kind: ContainerRuntimeConfig
metadata:
  name: cp4d-crio-limits
spec:
  machineConfigPoolSelector:
    matchLabels:
      limits-crio: cp4d-crio-limits
  containerRuntimeConfig:
    pidsLimit: 12288
```

Because we changed and increased the `pids_limit` for CNSA (see "Preparing OpenShift cluster nodes to run IBM Spectrum Scale CNSA" on page 13), we are ready to update the `pids_limit`, as shown in Example 3-12.

Example 3-12 Updating pids_limit

```
# oc get nodes -l node-role.kubernetes.io/worker | grep -v NAME| while read a b; do echo "## Node: $a -
$(oc debug node/$a -- chroot /host crio-status config 2>/dev/null | grep pids_limit)" ; done
## Node: worker01.ocp4.scale.com - pids_limit = 4096
## Node: worker02.ocp4.scale.com - pids_limit = 4096
## Node: worker03.ocp4.scale.com - pids_limit = 4096
```

Now, we modify the existing setting and adjust it to the requested value of `12288`:

```
# oc get ContainerRuntimeConfig
NAME                 AGE
increase-pid-limit    11d

# oc edit ContainerRuntimeConfig increase-pid-limit -o yaml
apiVersion: machineconfiguration.openshift.io/v1
```

```
kind: ContainerRuntimeConfig
[...]
spec:
  containerRuntimeConfig:
    pidsLimit: 4096                               <- change to 12288
  machineConfigPoolSelector:
   matchLabels:
     pid-crio: config-pid
```

Watch the MachineConfigPool until all nodes are updated:

```
# oc get MachineConfigPool
NAME     CONFIG                                             UPDATED   UPDATING   DEGRADED   MACHINECOUNT   READYMACHINECOUNT
UPDATEDMACHINECOUNT   DEGRADEDMACHINECOUNT   AGE
master   rendered-master-0c380523c7edc064c672ce4c7a533831   True      False      False      3              3
3                     0                      14d
worker   rendered-worker-4b63f598fc8da2c0cce5960f45318947   False     True       False      3              0
0                     0                      14d
```

Note: This process takes some time because the nodes are disabled from scheduling and rebooted individually.

After all nodes are updated, we see the following result:

```
# oc get nodes -l node-role.kubernetes.io/worker | grep -v NAME| while read a b; do echo "## Node: $a - 
$(oc debug node/$a -- chroot /host crio-status config 2>/dev/null | grep pids_limit)" ; done
## Node: worker01.ocp4.scale.com -      pids_limit = 12288
## Node: worker02.ocp4.scale.com -      pids_limit = 12288
## Node: worker03.ocp4.scale.com -      pids_limit = 12288
```

3.7.11 Kernel parameter settings

To ensure that specific microservices can run correctly, you must verify the kernel parameters. These settings are required for all deployments and depend on the machine RAM size and the operating system page size.

In this section, we assume that you have worker nodes with 64 GB of RAM on an x86 platform with a 4 K OS page size. If the worker nodes have 128 GB of RAM each, you must double the values for the kernel.shm* values:

- Virtual memory limit (`vm.max_map_count`)
- Message limits (`kernel.msgmax`, `kernel.msgmnb`, and `kernel.msgmni`)
- Shared memory limits (`kernel.shmmax`, `kernel.shmall`, and `kernel.shmmni`)

The following settings are recommended:

- `kernel.shmmni`: 256 * <*size of RAM in GB*>
- `kernel.shmmax`: <*size of RAM in bytes*>
- `kernel.shmall`: 2 * <*size of RAM in the default operating system page size*>

The default operating system page size on Power Systems is 64 KB. Take this operating system page size into account when you set the value for `kernel.shmall`.

Semaphore limits (kernel.sem)

As of Red Hat Enterprise Linux version 7.8 and Red Hat Enterprise Linux version 8.1, the `kernel.shmmni`, `kernel.msgmni`, and `kernel.semmni` settings in the `kernel.sem` settings are capped at 32768.

Values larger than 32768 are not applied and default values are used. The following default values are available:

- `kernel.shmmni`: 4096
- `kernel.msgmni`: 32000
- `kernel.semmni`: 128

Although you can apply values larger than 32768 by using the boot parameter `ipcmni_extend`, the values are still capped to 32768 internally by Red Hat Enterprise Linux On Red Hat OpenShift. You can use the Node Tuning Operator to manage node-level profiles.

For more information, see this web page.

The following example YAML manifest file (`42-cp4d.yaml`) is for worker nodes with 64 GB of RAM.


```
apiVersion: tuned.openshift.io/v1
kind: Tuned
metadata:
  name: cp4d-wkc-ipc
  namespace: openshift-cluster-node-tuning-operator
spec:
  profile:
  - name: cp4d-wkc-ipc
    data: |
     [main]
     summary=Tune IPC Kernel parameters on OpenShift Worker Nodes running WKC Pods
     [sysctl]
     kernel.shmall = 33554432
     kernel.shmmax = 68719476736
     kernel.shmmni = 16384
     kernel.sem = 250 1024000 100 16384
     kernel.msgmax = 65536
     kernel.msgmnb = 65536
     kernel.msgmni = 32768
     vm.max_map_count = 262144
  recommend:
  - match:
    - label: node-role.kubernetes.io/worker
    priority: 10
    profile: cp4d-wkc-ipc
```


If your current settings are less than these recommendations, adjust the settings in your YAML file.

The nodes in our proof of concept environment feature 128 GB of memory and show the default values before any changes (see Example 3-13).

Example 3-13 Default values before changes

```
[core@worker0 ~]$ free && sysctl kernel.shmall kernel.shmmax kernel.shmmni kernel.sem kernel.msgmax
kernel.msgmnb kernel.msgmni vm.max_map_count
              total      used       free     shared  buff/cache   available
Mem:      131692644  9485372  116861348     39400     5345924   121913772
Swap: 0 0 0
kernel.shmall = 18446744073692774399
kernel.shmmax = 18446744073692774399
kernel.shmmni = 4096
kernel.sem = 32000   1024000000    500       32000
```

```
kernel.msgmax = 8192
kernel.msgmnb = 16384
kernel.msgmni = 32000

vm.max_map_count = 262144
```

For nodes with 128 GB of RAM, we change the YAML file as recommended:

- `kernel.shmmni`: 256 * <*size of RAM in GB*>; 256 x 128 = 32768
- `kernel.shmmax`: <*size of RAM in bytes*>; 128GB = 137438953472
- `kernel.shmall`: 2 * <*size of RAM in page size*>; 2* 128GB/4k = 2*33554432 = 67108864

Setting the kernel parameters for microservices

To set the wanted kernel parameters for all worker nodes running WKC service, apply the tuning YAML to cause matching worker nodes to be tuned as specified:

```
[root@vm-1022 ~]# cat /root/42-cp4d.yaml
apiVersion: tuned.openshift.io/v1
kind: Tuned
metadata:
  name: cp4d-wkc-ipc
  namespace: openshift-cluster-node-tuning-operator
spec:
  profile:
  - name: cp4d-wkc-ipc
    data: |
      [main]
      summary=Tune IPC Kernel parameters on OpenShift Worker Nodes running WKC
Pods
      [sysctl]
      kernel.shmall = 33554432
      kernel.shmmax = 68719476736
      kernel.shmmni = 32768
      kernel.sem = 250 1024000 100 32768
      kernel.msgmax = 65536
      kernel.msgmnb = 65536
      kernel.msgmni = 32768
      vm.max_map_count = 262144
  recommend:
  - match:
    - label: node-role.kubernetes.io/worker
    priority: 10
    profile: cp4d-wkc-ipc
```

And apply the tuning YAML; for example:

```
# oc create -f 42-cp4d.yaml
tuned.tuned.openshift.io/cp4d-wkc-ipc created

# oc get tuned -n openshift-cluster-node-tuning-operator
NAME         AGE
cp4d-wkc-ipc  4m17s
default       14d
rendered      14d
```

However, by applying the recommended settings, we also decreased the following default settings:

```
kernel.shmall = 18446744073692774399
kernel.shmmax = 18446744073692774399
kernel.sem = 32000   1024000000 500   32000
```

We can confirm that the new settings were applied on the worker nodes (see Example 3-14).

Example 3-14 New settings applied

```
[core@worker0 ~]$ free && sysctl kernel.shmall kernel.shmmax kernel.shmmni kernel.sem
kernel.msgmax kernel.msgmnb kernel.msgmni vm.max_map_count
              total        used        free      shared  buff/cache   available
Mem:       65936876    23402672    18938172     9142608    23596032    34978028
Swap:             0           0           0
kernel.shmall = 33554432
kernel.shmmax = 68719476736
kernel.shmmni = 32768
kernel.sem = 250102400010032768
kernel.msgmax = 65536
kernel.msgmnb = 65536
kernel.msgmni = 32768
vm.max_map_count = 262144
```

3.7.12 Downloading the Cloud Pak for Data installation files

For more information about obtaining the installation file, see this web page. Then, begin the installation process.

An entitlement key for the software is associated with your My IBM account. To get the entitlement key, log in to Container software library on My IBM with the IBM ID and password that are associated with the entitled software at this web page.

Copy the entitlement key to the clipboard and save the API key in a text file (see Figure 3-1 on page 61):

```
# cat key.txt
eyJhb...[snip]...LOvHrI
```

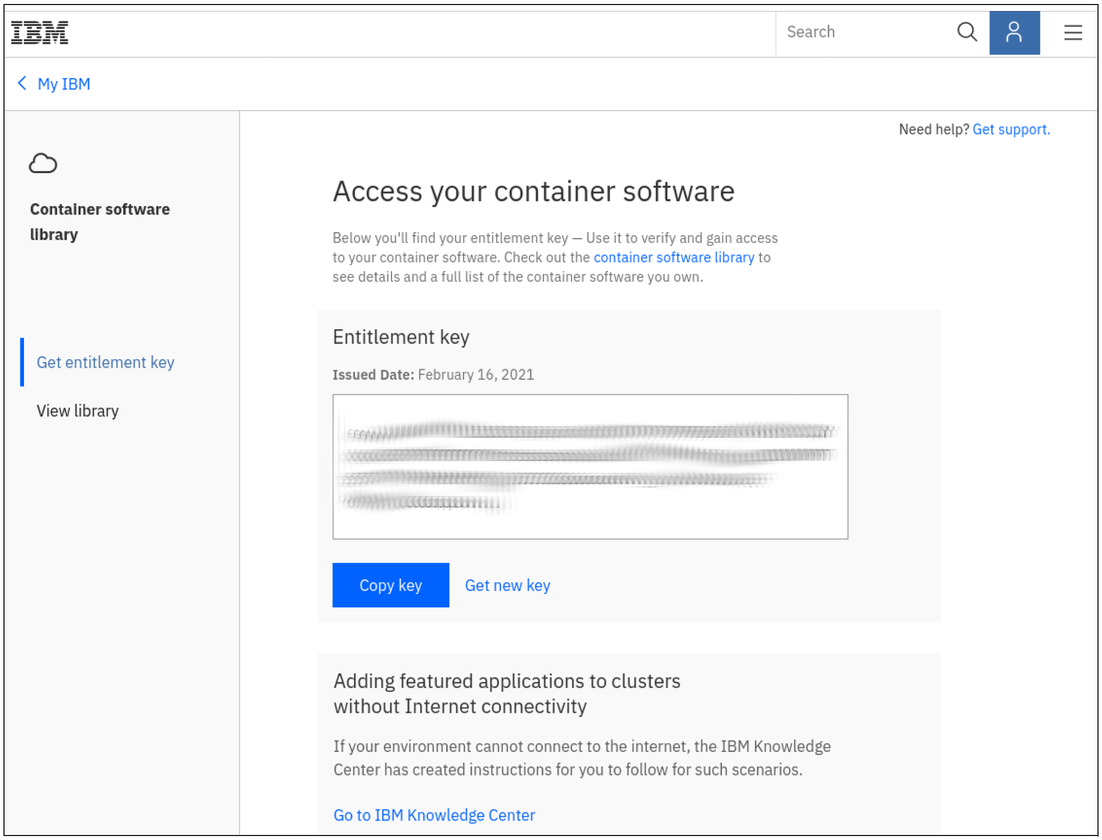


Figure 3-1 Get entitlement key

A Linux or Mac OS client workstation is required to run the installation. The workstation must have internet access and connect to the Red Hat OpenShift cluster.

Cloud Pak for Data can be installed by using an internet-connected or air-gapped environment. Here, we have an internet-connected environment and perform the internet-connected deployment.

Download and extract the installer `.tar` archive from Cloud Pak for Data 3.5.2 Installer (see Example 3-15).

Example 3-15 Downloading and extracting the .tar archive

```
# wget https://github.com/IBM/cpd-cli/releases/download/v3.5.2/cpd-cli-linux-EE-3.5.2.tgz
# mkdir cp4d-installer
# cd cp4d-installer/
# tar -xzf ../cpd-cli-linux-EE-3.5.2.tgz
# ls -al
-rwxr-xr-x. 1 gero 1006 6938228 Jan 19 18:43 cpd-cli
drwxr-xr-x. 2 gero 1006   4096 Jan 19 18:50 LICENSES
drwxr-xr-x. 5 gero 1006 48 Jan 19 18:50 plugins
-rw-r--r--. 1 root root    220 Jan 29 22:22 repo.yaml
```

Then, add your entitlement key to the `repo.yaml` file:

```
# cat repo.yaml
---
fileservers:
```

Some services are hosted in separate repositories. If you plan to install any of the following services, you must add the suitable entries to the `repo.yaml` file (for more information, see this web page):

- IBM Guardium® External S-TAP
- Watson Assistant
- Watson Assistant for Voice Interaction
- Watson Discovery
- Watson Knowledge Catalog
- Watson Language Translator
- Watson Speech to Text or Watson Text to Speech
- Edge Analytics

For example, if we wanted to use the Watson Knowledge Studio service, we must append the following YAML code to the `repo.yaml` file:

```
- url: cp.icr.io
  username: cp
  apikey: entitlement-key
  namespace: "cp/knowledge-studio"
  name: wks-registry
- url: cp.icr.io
  username: "cp"
  apikey: entitlement-key
  namespace: "cp"
  name: prod-entitled-registry
- url: cp.icr.io
  username: cp
  apikey: entitlement-key
  namespace: "cp"
  name: entitled-registry
- url: cp.icr.io
  username: cp
  apikey: entitlement-key
  namespace: "cp/cpd"
  name: databases-registry
```

Replace *entitlement-key* with the same entitlement license API key that you specified for the default registry in the `repo.yaml`.

3.7.13 Installing prerequisite components

Some required components must be installed before you install IBM Cloud Pak for Data; for example:

- Cloud Pak for Data Operator
- Scheduling service

For more information, see this web page.

Cloud Pak for Data Operator

The Cloud Pak for Data Operator is installed only once on your Red Hat OpenShift cluster.

> **Important:** Install the Cloud Pak for Data Operator if any of the following situations applies to you:
>
> - You are installing IBM Cloud Pak for Data from the Red Hat Marketplace.
> - You plan to use the Volumes API to create volumes on an external NFS server.

In this setup, we skip the Cloud Pak for Data Operator installation because we do not need it.

Scheduling service

The scheduling service is a cluster-wide pod service that you can install on your IBM Cloud Pak for Data cluster. The scheduling service offers enhancements over the default Kubernetes scheduler, including the following examples:

- Quota enforcement (various Cloud Pak for Data services)
- Co-scheduling of pods (Watson Machine Learning Accelerator service)
- GPU sharing (Watson Machine Learning Accelerator service)

> **Important:** If you plan to install the Watson Machine Learning Accelerator service on your cluster, you must install the scheduling service.

In this setup, we also skip the scheduling service installation because we do not need it.

3.8 Installing the Cloud Pak for Data control plane (lite)

First, we install the control plane for IBM Cloud Pak for Data. The name for associated assembly for the IBM Cloud Pak for Data *control plane* is `lite`.

3.8.1 Setting up the cluster for the control plane with cpd-cli

Set up the cluster for the control plane by completing the following steps for your environment:

1. From your installation node, change to the directory where you placed the Cloud Pak for Data command-line interface and the `repo.yaml` file.
2. Log in to your Red Hat OpenShift cluster as an administrator. Create the project (namespace) where you plan to install Cloud Pak for data Control plane.
3. Run the following command to see a preview of the list of resources that must be created on the cluster:

   ```
   ./cpd-cli adm \
   --repo ./repo.yaml \
   --assembly lite \
   --arch Cluster_architecture \
   --namespace Project
   ```

4. Specify the architecture of your cluster hardware:
   - Cluster_architecture:
     - For x86-64 hardware, remove this flag or specify x86_64
     - For POWER hardware, specify ppc64le
     - For IBM Z hardware, specify s390x

– Project

The project where the Cloud Pak for Data control plane is installed.

In our example, we use:

```
ASSEMBLY="lite"
ARCH="x86_64"
PROJECT="zen"
# oc new-project zen
Now using project "zen" on server "https://api.ocp4.scale.com:6443".

# ./cpd-cli adm -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} --arch ${ARCH} --apply
--latest-dependency
--accept-all-licenses
[INFO] [2021-02-17 18:37:49-0265] Parsing custom YAML file repo.yaml
[INFO] [2021-02-17 18:37:49-0266] Overwritten default download settings using
repo.yaml
[INFO] [2021-02-17 18:37:49-0267] 1 file servers and 1 registries detected from
current configuration
[INFO] [2021-02-17 18:37:49-0267] Server configure files validated
[...]
```

When you run the **cpd-cli adm** command with the **--apply** flag, the OpenShift service accounts that are listed in Table 3-1 are created (including the necessary SCC bindings and the configuration of all of the necessary resources).

Table 3-1 Created OpenShift service accounts

Service account	GET permissions	PUT/POST/DELETE permissions	Elevated security context
cpd-viewer-sa	Y	N	N
cpd-editor-sa	Y	Y	N
cpd-admin-sa	Y	Y	Y
cpd-norbac-sa	N	N	N

Important: Run the following command to grant the cpd-admin-role to the project administration user:

```
oc adm policy add-role-to-user cpd-admin-role Project_admin --role-namespace=
```

For more information, see this web page.

3.8.2 Installing the control plane with cpd-cli

A project administrator can install the control plane service on IBM Cloud Pak for Data.

To complete this task, you must be an administrator of the project (namespace) where you deploy control plane (see Example 3-16).

Example 3-16 Deploying the control plane

```
ASSEMBLY="lite"
ARCH="x86_64"
PROJECT="zen"
STORAGE_CLASS="ibm-spectrum-scale-sc"
PUSHREGISTRY="$(oc registry info)"
PULLREGISTRY="image-registry.openshift-image-registry.svc:5000"
# ./cpd-cli install -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} -c ${STORAGE_CLASS} --arch ${ARCH} \
--transfer-image-to=${PUSHREGISTRY}/${PROJECT} \
--target-registry-username=$(oc whoami) --target-registry-password=$(oc whoami -t) \
--insecure-skip-tls-verify \
--cluster-pull-prefix=image-registry.openshift-image-registry.svc:5000/${PROJECT} \
--latest-dependency --accept-all-licenses
```

If you run the previous command with the **`--dry-run`** flag, it downloads only the charts and stops.

To verify that the installation completed successfully, we run the **`cpd-cli status`** command (see Example 3-17).

Example 3-17 Verifying successful installation

```
# ./cpd-cli status -a ${ASSEMBLY} -n ${PROJECT}
[INFO] [2021-02-18 16:54:22-0068] Arch override not found. Assuming default architecture x86_64
[INFO] [2021-02-18 16:54:22-0347]
Displaying CR status for all assemblies and relevant modules
Status for assembly lite and relevant modules in project zen:
Assembly Name                    Status         Version         Arch
lite                             Ready          3.5.2           x86_64
  Module Name                    Version        Arch            Storage Class
  0010-infra                     3.5.2          x86_64          ibm-spectrum-scale-sc
  0015-setup                     3.5.2          x86_64          ibm-spectrum-scale-sc
  0020-core                      3.5.2          x86_64          ibm-spectrum-scale-sc
===============================================================================================
```

If the installation completed successfully, the status of the assembly and the modules in the assembly is `Ready`. We can check whether patches are available by running the **`cpd-cli status`** --patches command (see Example 3-18).

Example 3-18 Checking patches availability

```
# ./cpd-cli status -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} --arch ${ARCH} --patches --available- updates
*** Checking upgrade availability for assembly lite ***

[INFO] [2021-02-18 17:00:01-0743] Parsing custom YAML file repo.yaml
[INFO] [2021-02-18 17:00:01-0744] Overwritten default download settings using repo.yaml
[INFO] [2021-02-18 17:00:01-0744] 1 file servers and 1 registries detected from current configuration
[INFO] [2021-02-18 17:00:01-0745] Server configure files validated

[INFO] [2021-02-18 17:00:03-0514] Upgrade is not available for assembly lite
*** Checking available patches for assembly lite ***
```

```
[INFO] [2021-02-18 17:00:03-0514] Parsing custom YAML file repo.yaml
[INFO] [2021-02-18 17:00:03-0516] Overwritten default download settings using repo.yaml
[INFO] [2021-02-18 17:00:03-0516] 1 file servers and 1 registries detected from current configuration
[INFO] [2021-02-18 17:00:03-0517] Server configure files validated

[INFO] [2021-02-18 17:00:03-0658] Downloading for patch files
[INFO] [2021-02-18 17:00:04-0143] Parsing patch definition files for list of available patches
[INFO] [2021-02-18 17:00:04-0144] List of available patches for assembly lite: [cpd-3.5.2-lite-patch-1]
[INFO] [2021-02-18 17:00:04-0150] The assembly lite has no service types defined in their manifest file
[WARNING] [2021-02-18 17:00:04-0150] Assembly lite does not contain any service instance type definitions
[INFO] [2021-02-18 17:00:04-0151]
Displaying CR status for all assemblies and relevant modules
Status for assembly lite and relevant modules in project zen:
Assembly Name                Status          Version         Arch
lite                         Ready           3.5.2           x86_64

  Module Name                        Version          Arch            Storage Class
  0010-infra                         3.5.2            x86_64          ibm-spectrum-scale-sc
  0015-setup                         3.5.2            x86_64          ibm-spectrum-scale-sc
  0020-core                          3.5.2            x86_64          ibm-spectrum-scale-sc
  Upgrade availability check:

    Upgrade is not available for assembly lite: Target version is the same as deployed version

  Patch availability check:

    List of available patches for assembly lite:

      Patch cpd-3.5.2-lite-patch-1

=========================================================================================
```

We apply the patches in the sequence that is shown in Example 3-19.

Example 3-19 Sequence for applying patches

```
# PATCH_NAME="cpd-3.5.2-lite-patch-1"
# ./cpd-cli patch --patch-name ${PATCH_NAME} --action transfer -r repo.yaml -a ${ASSEMBLY} -n $
{PROJECT} --arch ${ARCH} --transfer-image-to=${PUSHREGISTRY}/${PROJECT}
--target-registry-username=$(oc whoami) --target-registry-password=$(oc whoami -t)
--insecure-skip-tls-verify --cluster-pull-
prefix=image-registry.openshift-image-registry.svc:5000/${PROJECT}
[…]
---------------------------
[INFO] [2021-02-18 17:16:04-0259] Patch description per module

------------------------------------------------------------------------------------------
---------------------------
Assembly     Module         Patchfile            Description
lite         0010-infra     patchCommands.txt    Fixing issue #22638, #22708
lite         0020-core      patchCommands.txt    need TLS renegotiation, issue with RootCA
failing to validate against a signed https server that does not need RootCA in the first place
------------------------------------------------------------------------------------------
[…]
[INFO] [2021-02-18 17:18:49-0417] No service instances detected for further patching
[INFO] [2021-02-18 17:18:49-0417] Patch for assembly lite completed successfully
```

After the deployment, we can see the following objects were created in the namespace of the control plane:

```
# oc get all
NAME                                                      READY   STATUS      RESTARTS   AGE
pod/cpd-install-operator-6767c9b88b-j4kml                 1/1     Running     0          93m
pod/diagnostics-cronjob-1613665200-n9pcs                  0/1     Completed   0          9m
pod/dsx-influxdb-6f986d999d-qr8hp                         1/1     Running     0          89m
pod/dsx-influxdb-set-auth-7b55s                           0/1     Completed   0          89m
pod/ibm-nginx-784b987fff-7cjjr                            1/1     Running     0          86m
pod/ibm-nginx-784b987fff-dlhgf                            1/1     Running     0          86m
pod/icpd-till-7bf9cfb465-lkxp8                            1/1     Running     0          92m
pod/pv-prep-job-vsbmt                                     0/1     Completed   0          89m
pod/usermgmt-67c5b95db6-cf2k6                             1/1     Running     0          12m
pod/usermgmt-67c5b95db6-g5gj9                             1/1     Running     0          12m
pod/usermgmt-ldap-sync-cron-job-1613661600-ppdnh          0/1     Completed   0          69m
pod/usermgmt-ldap-sync-cron-job-1613662800-jrrvd          0/1     Completed   0          49m
pod/usermgmt-ldap-sync-cron-job-1613664000-xg98t          0/1     Completed   0          28m
pod/watchdog-alert-monitoring-cronjob-1613665200-bqkbk    0/1     Completed   0          9m
pod/zen-audit-5fc5f8f679-dcgv9                            1/1     Running     0          84m
pod/zen-core-6775f9bbff-k7q7g                             1/1     Running     1          84m
pod/zen-core-6775f9bbff-xbl5k                             1/1     Running     0          84m
pod/zen-core-api-99dcd5f84-66k9x                          1/1     Running     0          12m
pod/zen-core-api-99dcd5f84-lcq5d                          1/1     Running     0          11m
pod/zen-data-sorcerer-7c8bdc95b4-vfcb7                    1/1     Running     0          84m
pod/zen-metastoredb-0                                     1/1     Running     0          89m
pod/zen-metastoredb-1                                     1/1     Running     0          89m
pod/zen-metastoredb-2                                     1/1     Running     0          89m
pod/zen-metastoredb-init-rjfjm                            0/1     Completed   0          89m
pod/zen-migrate-user-roles-job-rrr7z                      0/1     Completed   0          84m
pod/zen-watchdog-5785c8fc44-hbcjt                         1/1     Running     0          84m
pod/zen-watchdog-cronjob-1613665200-th9w2                 0/1     Completed   0          9m
pod/zen-watcher-7cbb787ccc-q79r8                          1/1     Running     0          12m

NAME                                 TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                      AGE
service/dsx-influxdb                 ClusterIP   172.30.124.22    <none>        8086/TCP                     89m
service/ibm-nginx-svc                ClusterIP   172.30.246.63    <none>        443/TCP                      86m
service/internal-nginx-svc           ClusterIP   172.30.241.217   <none>        12443/TCP,12080/TCP          86m
service/tiller-svc                   ClusterIP   172.30.49.89     <none>        44134/TCP                    92m
service/usermgmt-svc                 ClusterIP   172.30.121.63    <none>        8080/TCP,3443/TCP            89m
service/zen-audit-svc                ClusterIP   172.30.255.217   <none>        9880/TCP,9890/TCP,5140/TCP   84m
service/zen-core-api-svc             ClusterIP   172.30.58.77     <none>        3333/TCP,4444/TCP            84m
service/zen-core-svc                 ClusterIP   172.30.252.218   <none>        3003/TCP,3443/TCP            84m
service/zen-data-sorcerer-svc        ClusterIP   172.30.15.8      <none>        2222/TCP                     84m
service/zen-metastoredb              ClusterIP   None             <none>        26257/TCP,8080/TCP           89m
service/zen-metastoredb-public       ClusterIP   172.30.224.183   <none>        26257/TCP,8080/TCP           89m
service/zen-watchdog-svc             ClusterIP   172.30.229.157   <none>        3333/TCP,4444/TCP            84m

NAME                                   READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/cpd-install-operator   1/1     1            1           93m
deployment.apps/dsx-influxdb           1/1     1            1           89m
deployment.apps/ibm-nginx              2/2     2            2           86m
deployment.apps/icpd-till              1/1     1            1           92m
deployment.apps/usermgmt               2/2     2            2           89m
deployment.apps/zen-audit              1/1     1            1           84m
deployment.apps/zen-core               2/2     2            2           84m
deployment.apps/zen-core-api           2/2     2            2           84m
deployment.apps/zen-data-sorcerer      1/1     1            1           84m
deployment.apps/zen-watchdog           1/1     1            1           84m
deployment.apps/zen-watcher            1/1     1            1           84m

NAME                                              DESIRED   CURRENT   READY   AGE
replicaset.apps/cpd-install-operator-6767c9b88b   1         1         1       93m
replicaset.apps/dsx-influxdb-6f986d999d           1         1         1       89m
replicaset.apps/ibm-nginx-784b987fff              2         2         2       86m
replicaset.apps/icpd-till-7bf9cfb465              1         1         1       92m
replicaset.apps/usermgmt-67c5b95db6               2         2         2       12m
replicaset.apps/usermgmt-68dbd44fcf               0         0         0       89m
replicaset.apps/zen-audit-5fc5f8f679              1         1         1       84m
replicaset.apps/zen-core-6775f9bbff               2         2         2       84m
replicaset.apps/zen-core-api-6f7ffb5496           0         0         0       84m
replicaset.apps/zen-core-api-99dcd5f84            2         2         2       12m
replicaset.apps/zen-data-sorcerer-7c8bdc95b4      1         1         1       84m
replicaset.apps/zen-watchdog-5785c8fc44           1         1         1       84m
replicaset.apps/zen-watcher-7cbb787ccc            1         1         1       12m
replicaset.apps/zen-watcher-85cd47cf87            0         0         0       84m

NAME                               READY   AGE
statefulset.apps/zen-metastoredb   3/3     89m

NAME                                                      COMPLETIONS   DURATION   AGE
```

```
job.batch/diagnostics-cronjob-1613665200                1/1          9s         9m
job.batch/dsx-influxdb-set-auth                         1/1          81s        89m
job.batch/pv-prep-job                                   1/1          60s        89m
job.batch/usermgmt-ldap-sync-cron-job-1613661600        1/1          15s        69m
job.batch/usermgmt-ldap-sync-cron-job-1613662800        1/1          15s        49m
job.batch/usermgmt-ldap-sync-cron-job-1613664000        1/1          15s        28m
job.batch/watchdog-alert-monitoring-cronjob-1613665200  1/1          9s         9m
job.batch/zen-metastoredb-init                          1/1          65s        89m
job.batch/zen-migrate-user-roles-job                    1/1          2m9s       84m
job.batch/zen-watchdog-cronjob-1613665200               1/1          9s         9m

NAME                                                   SCHEDULE      SUSPEND   ACTIVE   LAST SCHEDULE   AGE
cronjob.batch/diagnostics-cronjob                      */10 * * * *  False     0        9m3s            82m
cronjob.batch/usermgmt-ldap-sync-cron-job              */20 * * * *  True      0        29m             84m
cronjob.batch/watchdog-alert-monitoring-cronjob        */10 * * * *  False     0        9m3s            83m
cronjob.batch/watchdog-alert-monitoring-purge-cronjob  @daily        False     0        <none>          83m
cronjob.batch/zen-watchdog-cronjob                     */10 * * * *  False     0        9m3s            83m

NAME                                              IMAGE REPOSITORY
TAGS                  UPDATED
imagestream.image.openshift.io/cp4d-tiller-ubi8       default-route-openshift-image-registry.apps.ocp4.scale.com/zen/cp4d-tiller-ubi8
v2.16.6.2-x86_64        About an hour ago
imagestream.image.openshift.io/cpd-operator-init      default-route-openshift-image-registry.apps.ocp4.scale.com/zen/cpd-operator-init
v1.0.2-x86_64           About an hour ago
imagestream.image.openshift.io/cpdoperator            default-route-openshift-image-registry.apps.ocp4.scale.com/zen/cpdoperator
v3.5.2-43-x86_64        About an hour ago
imagestream.image.openshift.io/privatecloud-usermgmt  default-route-openshift-image-registry.apps.ocp4.scale.com/zen/privatecloud-
usermgmt
3.5.2-patch-x86_64-2   12 minutes ago
imagestream.image.openshift.io/zen-core-api           default-route-openshift-image-registry.apps.ocp4.scale.com/zen/zen-core-api
3.5.2-patch-x86_64-5   12 minutes ago

NAME                               HOST/PORT                        PATH   SERVICES        PORT                  TERMINATION
WILDCARD
route.route.openshift.io/zen-cpd   zen-cpd-zen.apps.ocp4.scale.com         ibm-nginx-svc   ibm-nginx-https-port passthrough/Redirect
None
```

The following PVCs were successfully created from the IBM Spectrum Scale SC:

```
# oc get pvc
NAME                         STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   AGE
cpd-install-operator-pvc     Bound    pvc-e6a6faea-3025-4412-8aed-ee08a687686f   1Gi        RWX  ibm-spectrum-scale-sc   125m
cpd-install-shared-pvc       Bound    pvc-72f67a18-ec77-49ba-9e40-b6fd902faa61   1Gi        RWX  ibm-spectrum-scale-sc   125m
datadir-zen-metastoredb-0    Bound    pvc-beaf0d8d-881e-42af-b856-86dfda8f0358   10Gi       RWO  ibm-spectrum-scale-sc   121m
datadir-zen-metastoredb-1    Bound    pvc-47a7b534-84a7-4fab-b1ea-a64fc613f709   10Gi       RWO  ibm-spectrum-scale-sc   121m
datadir-zen-metastoredb-2    Bound    pvc-817d3d86-c7e0-4662-9207-f4e9ff4b527b   10Gi       RWO  ibm-spectrum-scale-sc   121m
influxdb-pvc                 Bound    pvc-a9d3467f-56d5-47b9-88ac-89c28a38d697   10Gi       RWX  ibm-spectrum-scale-sc   121m
user-home-pvc                Bound    pvc-bf336703-eace-444f-b00f-b2690fd83fed   10Gi       RWX  ibm-spectrum-scale-sc   121m
```

The following container images were uploaded into the internal OpenShift registry:

```
# oc get is
NAME                    IMAGE REPOSITORY                                                            TAGS
UPDATED
cp4d-tiller-ubi8         default-route-openshift-image-registry.apps.ocp4.scale.com/zen/cp4d-tiller-ubi8       v2.16.6.2-x86_64       2
hours ago
cpd-operator-init        default-route-openshift-image-registry.apps.ocp4.scale.com/zen/cpd-operator-init      v1.0.2-x86_64          2
hours ago
cpdoperator              default-route-openshift-image-registry.apps.ocp4.scale.com/zen/cpdoperator            v3.5.2-43-x86_64       2
hours ago
privatecloud-usermgmt default-route-openshift-image-registry.apps.ocp4.scale.com/zen/privatecloud-usermgmt    3.5.2-patch-x86_64-2  49
minutes ago
zen-core-api default-route-openshift-image-registry.apps.ocp4.scale.com/zen/zen-core-api                       3.5.2-patch-x86_64-5  49
minutes ago
```

Watch for the creation of the `user-home-pvc` by using the following command:

```
oc get pvc -n zen -w
```

Wait for the creation of the PVC and for STATUS of Bound:

```
[root@arcx3650fxxnh ~]# oc get pvc user-home-pvc -n zen
NAME            STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS            AGE
user-home-pvc   Bound     pvc-90346e43-cbf5-467c-b83e-c93a4832eb5e   10Gi       RWX            ibm-spectrum-scale-sc   16h
```

Note the PVC volume for `user-home-pvc` (in this example, the volume is:

```
pvc-90346e43-cbf5-467c-b83e-c93a4832eb5e
```

On the remote storage cluster, adjust the inode-limit for the PVC:

```
[root@stg-node0 ~]# mmchfileset fs1 pvc-90346e43-cbf5-467c-b83e-c93a4832eb5e --inode-limit '6144:6144'
Set maxInodes for inode space 8 to 6144
Fileset pvc-90346e43-cbf5-467c-b83e-c93a4832eb5e changed.
[root@stg-node0 ~]# mmlsfileset fs1 pvc-90346e43-cbf5-467c-b83e-c93a4832eb5e -i
Collecting fileset usage information ...
Filesets in file system 'fs1':
Name                     Status    Path                                           InodeSpace      MaxInodes    AllocInodes     UsedInodes
pvc-90346e43-cbf5-467c-b83e-c93a4832eb5e Linked /ibm/fs1/pvc-90346e43-cbf5-467c-b83e-c93a4832eb5e 8       6144      6144              20
```

3.8.3 Accessing the Cloud Pak for Data web client

By default, the route to the Cloud Pak for Data web client uses the following format:

`https://namespace-cpd-namespace.apps.cluster-subdomain/zen/`

You also can get the routes by using the following command against the Cloud Pak for Data Control Plane namespace:

```
# oc -n <cp4d_project_name> get routes
```

For more information about creating a custom route to override the default URL, see this web page.

Our cluster features the `ocp4.scale.com` domain; therefore, we can reach the Cloud Pak for Data console by using the following URL `https://zen-cpd-zen.apps.ocp4.scale.com`.

This URL is redirected to `https://zen-cpdzen.apps.ocp4.scale.com/zen/`.

After you install Cloud Pak for Data, you can configure the web client to add users and set up email notifications. The default user name is `admin`, and the default password is `password` (see Figure 3-2).

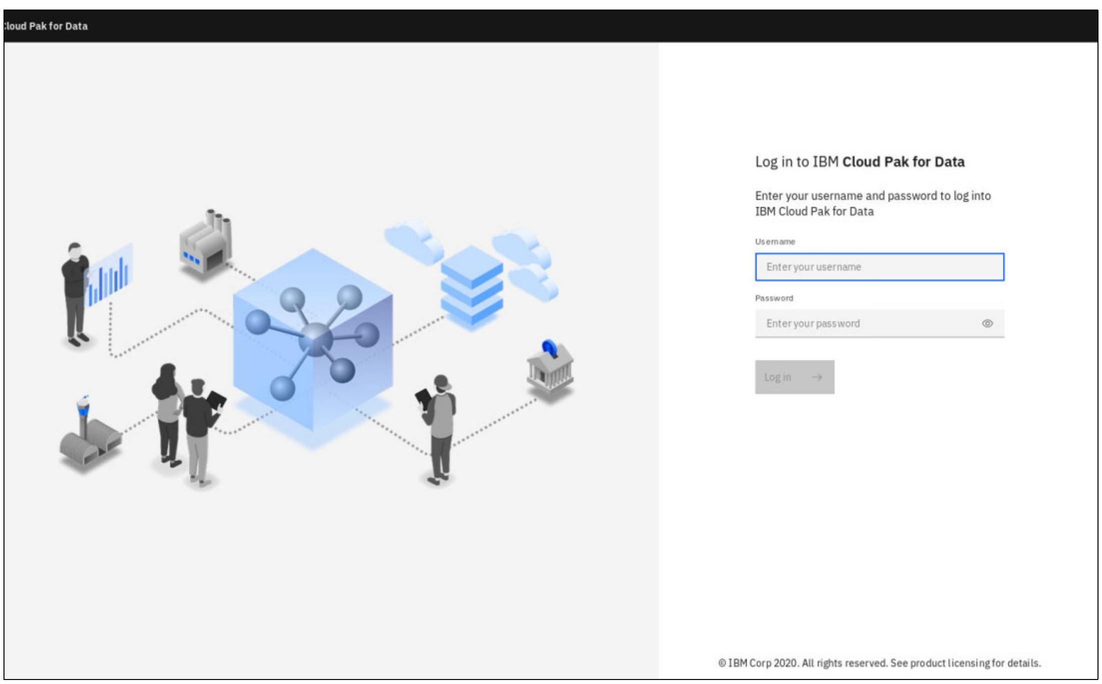


Figure 3-2   Log in for IBM Cloud Pak for Data

3.8.4 Installing more IBM Cloud Pak for Data services

After the IBM Cloud Pak for Data control plane is installed, a Red Hat OpenShift project administrator can install the services that support your business needs on the platform. Use the guidance that is available at this web page to ensure that you have the required tools and information before you install or upgrade any services on your cluster.

Available services are also listed in the services catalog in the Cloud Pak for data web GUI.

When selecting a service of interest, information about this service is shown (see Figure 3-3).

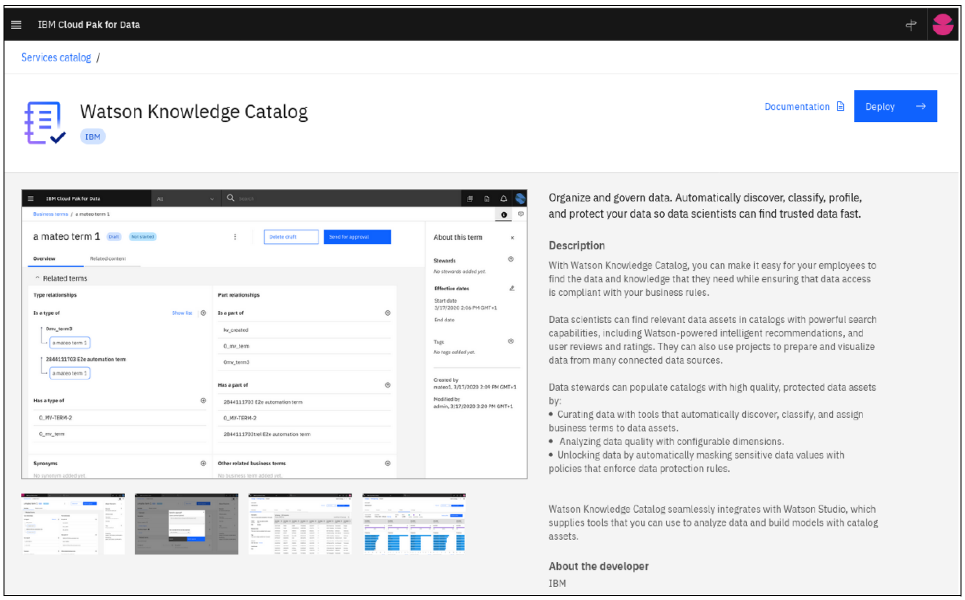


Figure 3-3 Watson Knowledge Catalog service selection

When the Deploy button is clicked for the selected service, links to the installation instructions are displayed in a small pop-up window, as shown in Figure 3-4.

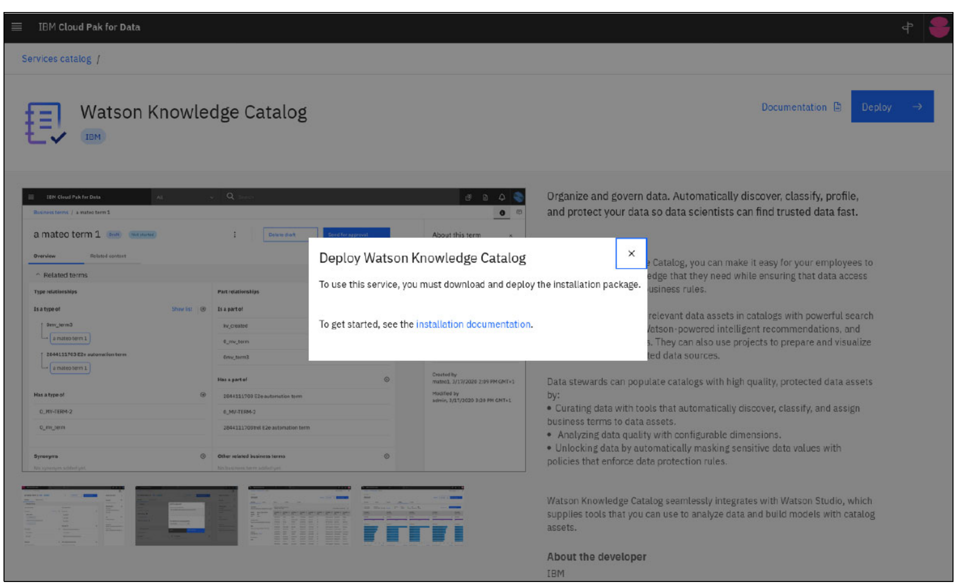


Figure 3-4 Installation instructions displayed after selecting DEPLOY button

You can install other services, including the following examples:

- Watson Studio: `wsl`
- Watson Machine Learning: `wml`
- Watson Knowledge Catalog: `wkc`
- Data Virtualization: `dv`
- Db2: `db2oltp`
- Db2 Warehouse: `db2wh`
- Db2 Event Store: `db2eventstore`
- IBM SPSS® Modeler: `spss-modeler`
- Decision Optimization: `dods`
- IBM Cognos® Analytics: `ca`
- DataStage: `ds`

You can install many other services by selecting them as a new assembly and installing them by following similar steps as for the control plane with the `cpd-cli` command line tool. Generally, the installation of these assemblies is similar, but can differ in some prerequisites for preparing the cluster (for example, dedicating nodes for a specific service by using labels and taints, or selecting a different SC).

Some assemblies, such as Watson Discovery, require a different SC or other overrides that can be specified by using an override YAML manifest. For more information, see the installation instructions of the specific service.

If you want to install an assembly that is not listed here, installation instructions for all assemblies are available at this web page.

In the following sections, we describe installing the following assemblies:

- Successful installation with `ibm-spectrum-scale-sc`[1] SC and created a database:
  - AI: Watson Studio (`wsl` assembly)
  - Data sources: Db2 (`db2oltp` assembly)
- Failed installation (because of `iis` sub-component) with `ibm-spectrum-scale-sc` SC but successful installation with `ibm-spectrum-scale3-sc`[2] SC:
  - Data governance: Watson Knowledge Catalog (`wkc` assembly)
  - Data governance: DataStage (`ds` assembly)
- Successful installation with `ibm-spectrum-scale-sc` SC:
  - Db2 Warehouse (`db2wh` assembly)
  - Watson Machine Learning (`wml` assembly)
  - Watson OpenScale (`aiopenscale` assembly)
  - Data Virtualization (`dv` assembly)
  - Apache Spark (`spark` assembly)
  - Data Management Console (`dmc` assembly)

[1] ibm-spectrum-scale-sc, see “IBM Spectrum Scale Storage Class with default uid and gid settings” on page 37
[2] ibm-spectrum-scale3-sc, see “IBM Spectrum Scale Storage Class with dedicated uid setting” on page 39

3.9 Installing Db2 (db2oltp)

To add a Db2 database to IBM Cloud Pak for Data, you first prepare your Red Hat OpenShift cluster and then install and deploy the database. For more information, see this web page.

Change to the directory where you placed the Cloud Pak for Data command-line interface and the `repo.yaml` file.

Run the command that is shown in Example 3-20 to prepare your Red Hat OpenShift cluster.

Example 3-20 Command to prepare Red Hat OpenShift cluster

```
# ASSEMBLY="db2oltp"
# ARCH="x86_64"
# PROJECT="zen"
# STORAGE_CLASS="ibm-spectrum-scale-sc"
# PUSHREGISTRY="$(oc registry info)"
# PULLREGISTRY="image-registry.openshift-image-registry.svc:5000"

# ./cpd-cli adm -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} --arch ${ARCH} --apply --accept-all-licenses
```

If you want to dedicate specific nodes to Db2, see this web page.

Install the Db2 database service:

```
# ./cpd-cli install -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} -c ${STORAGE_CLASS} \
 --arch ${ARCH} --transfer-image-to=${PUSHREGISTRY}/${PROJECT} \
 --target-registry-username=$(oc whoami) \
 --target-registry-password=$(oc whoami -t) --insecure-skip-tls-verify \
 --cluster-pull-prefix=image-registry.openshift-image-registry.svc:5000/${PROJECT}\
 --latest-dependency --accept-all-licenses
```

Verify the status of the assembly (see Example 3-21).

Example 3-21 Verifying the assembly status

```
# ./cpd-cli status -a ${ASSEMBLY} -n ${PROJECT}
[INFO] [2021-02-19 22:06:08-0468] Arch override not found. Assuming default architecture x86_64
[INFO] [2021-02-19 22:06:09-0680]
Displaying CR status for all assemblies and relevant modules Status for assembly db2oltp and
relevant modules in project zen:
Assembly Name                   Status          Version         Arch
db2oltp                         Ready           3.5.2           x86_64

  SubAssembly Name              Status          Version         Arch
  databases                      Ready           3.5.2           x86_64
  db2u-operator                  Ready           3.5.2           x86_64
  lite                           Ready           3.5.2           x86_64

  Module Name                   Version         Arch            Storage Class
  db2oltp                       3.5.2           x86_64          ibm-spectrum-scale-sc
=========================================================================================
```

Search for patches (see Example 3-22).

Example 3-22 Searching for patches

```
# ./cpd-cli status -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} --arch ${ARCH} --patches --availableupdates
*** Checking upgrade availability for assembly db2oltp ***
[...]
  Upgrade availability check:
    Upgrade is not available for assembly db2oltp: Target version is the same as deployed version
  Patch availability check:
    No info on available patches has been found
```

The `db2oltp` service is now successfully installed by using IBM Spectrum Scale as storage provider.

3.9.1 Creating a Db2 database

You can create a database deployment on your cluster from the IBM Cloud Pak for Data web client. For more information, see this web page.

From **Instances**, select **Create Instance** and then, select **DB2**. From the left navigation menu, select **Data** → **Databases**, select the installed Db2 database type and then, click **Next** button (see Figure 3-5).

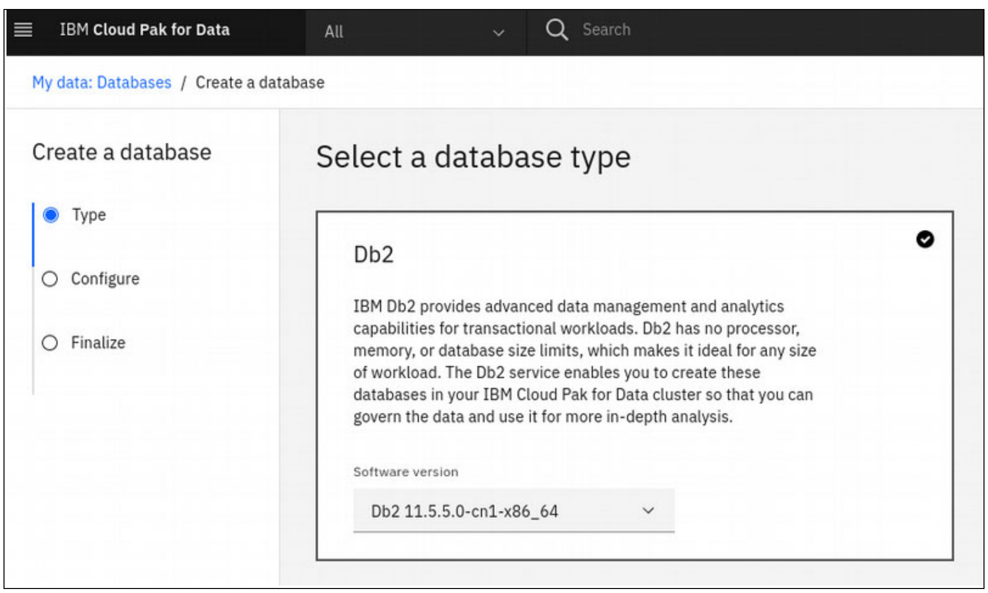


Figure 3-5 Selecting a database type

In the Configure window, set the parameters to meet your specifications (see Figure 3-6).

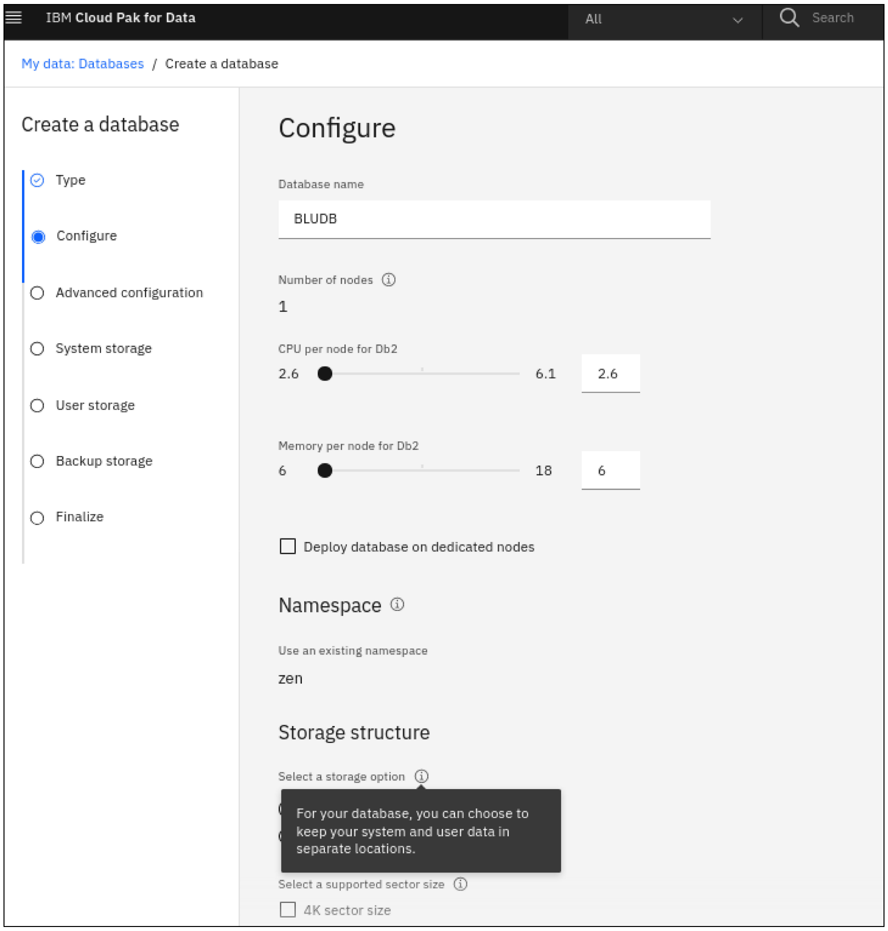


Figure 3-6 Configure the Db2 database

In this deployment, we configured the Db2 database with the settings that are shown in (see Figure 3-7). Make sure the correct SC (`spectrum-scale-sc`) is selected.

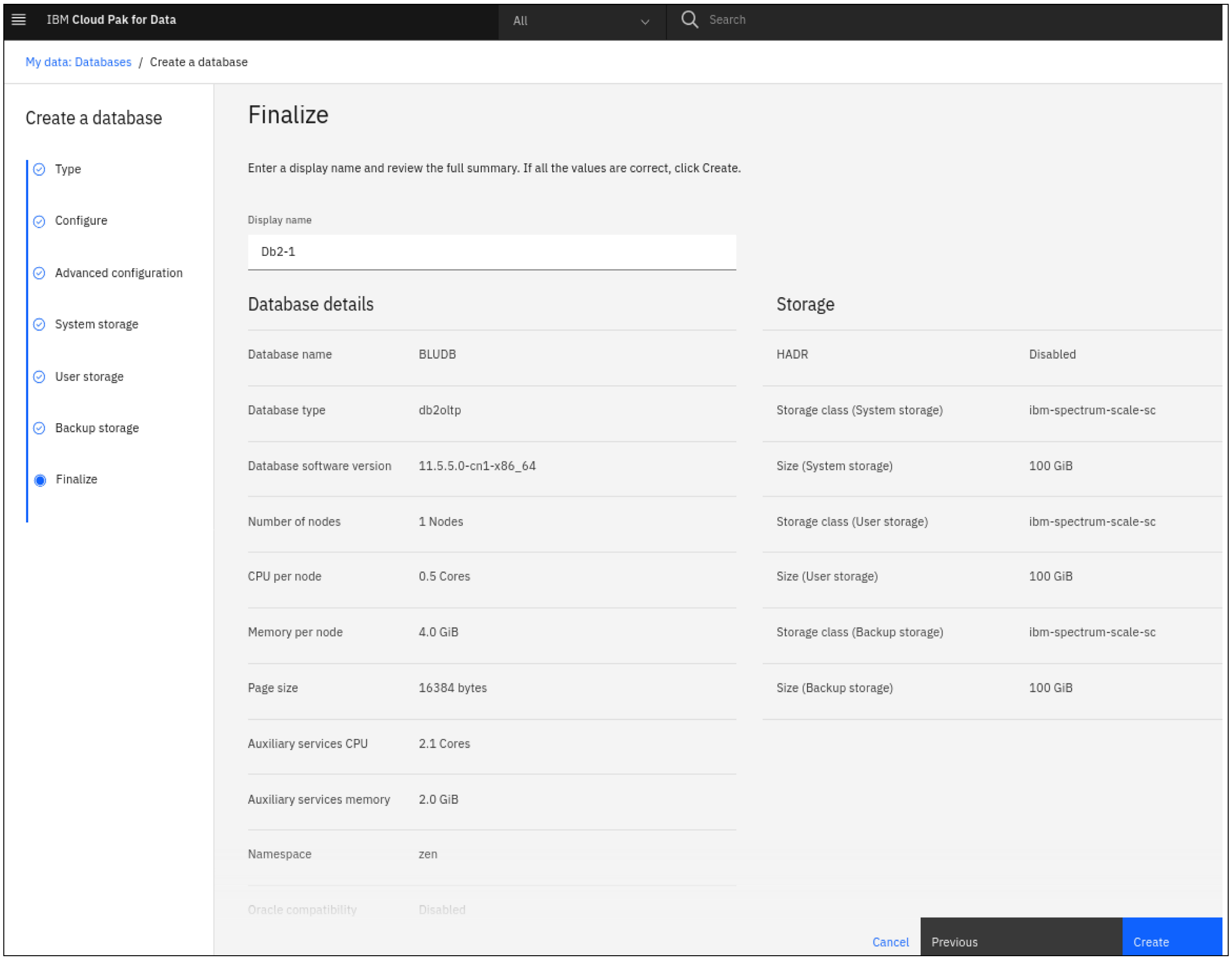


Figure 3-7 Db2 database configuration settings

Click Create to create the database (see Figure 3-8).

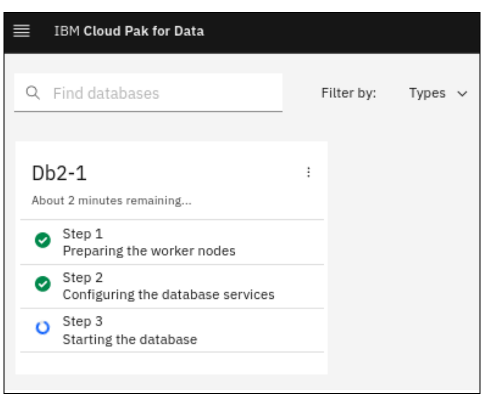


Figure 3-8 Creating the Db2 database

After the installation of the `db2oltp` service and a first Db2 database is created in the Cloud Pak for Data web client, the volumes that are shown in the following example are created by Cloud Pak for Data on IBM Spectrum Scale:

```
# oc get pvc
NAME                                     STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS             AGE
c-db2oltp-1614095761315872-backup        Bound    pvc-13e4e8bd-42b9-4e3e-a6ca-385c899cd005   100Gi      RWX            ibm-spectrum-scale- sc   17h
c-db2oltp-1614095761315872-meta          Bound    pvc-f2d37c36-0ed6-4a39-82ce-adcab07c9b5c   100Gi      RWX            ibm-spectrum-scale- sc   17h
cpd-install-operator-pvc                 Bound    pvc-e6a6faea-3025-4412-8aed-ee08a687686f   1Gi        RWX            ibm-spectrum-scale- sc   5d18h
cpd-install-shared-pvc                   Bound    pvc-72f67a18-ec77-49ba-9e40-b6fd902faa61   1Gi        RWX            ibm-spectrum-scale- sc   5d18h
data-c-db2oltp-1614095761315872-db2u-0   Bound    pvc-5802d3c1-07fe-4d63-b274-27abb80b999a   100Gi      RWO            ibm-spectrum-scale- sc   17h
datadir-zen-metastoredb-0                Bound    pvc-beaf0d8d-881e-42af-b856-86dfda8f0358   10Gi       RWO            ibm-spectrum-scale- sc   5d18h
datadir-zen-metastoredb-1                Bound    pvc-47a7b534-84a7-4fab-b1ea-a64fc613f709   10Gi       RWO            ibm-spectrum-scale- sc   5d18h
datadir-zen-metastoredb-2                Bound    pvc-817d3d86-c7e0-4662-9207-f4e9ff4b527b   10Gi       RWO            ibm-spectrum-scale- sc   5d18h
influxdb-pvc                             Bound    pvc-a9d3467f-56d5-47b9-88ac-89c28a38d697   10Gi       RWX            ibm-spectrum-scale- sc   5d18h
user-home-pvc                            Bound    pvc-bf336703-eace-444f-b00f-b2690fd83fed   10Gi       RWX            ibm-spectrum-scale- sc   5d18h
```

3.10 Installing Watson Knowledge Catalog (wkc)

For more information about how to install the Watson Knowledge Catalog service, see IBM Documentation.

3.10.1 Time zones

If the service is to be installed on a remote machine that runs in a different time zone than the master node, the time zone for the master node is overwritten by the time zone for the installer node.

This time zone discrepancy results in scheduled jobs that do not run at the correct time. For more information, see IBM Documentation.

If a specific time zone is required to be set, you can use a YAML manifest file; for example, `override.yaml`, as shown in the following example:

```
global:
masterTimezone: 'America/Los_Angeles'
```

Include it with `--override override.yaml` in the installation command.

The cluster that is used for this deployment runs in the UTC time zone, which is the expected default for Red Hat OpenShift clusters and not necessarily meant to be changed (see Example 3-23). For more information, see Red Hat #4994241.

Example 3-23 Cluster runs in UTC time zone

```
#  for i in $(oc get nodes -ojsonpath='{.items[*].metadata.name}'); do oc debug node/$i -- chroot /host
date 2>/dev/null; done
Mon Feb 22 10:32:49 UTC 2021
Mon Feb 22 10:32:51 UTC 2021
Mon Feb 22 10:32:52 UTC 2021
Mon Feb 22 10:32:54 UTC 2021
Mon Feb 22 10:32:55 UTC 2021
Mon Feb 22 10:32:57 UTC 2021
```

The OpenShift nodes (for example, by way of SSH) are also set to UTC time zone:

```
# ssh core@master01.ocp4.scale.com “timedatectl”
               Local time:  Mon 2021-02-22 09:55:02 UTC
           Universal time:  Mon 2021-02-22 09:55:02 UTC
                 RTC time:  Mon 2021-02-22 09:55:02
```

```
      Time zone:  UTC (UTC, +0000) System clock synchronized: yes
      NTP service:  active
   RTC in local TZ:  no
```

The time zone code for UTC is shown in the following example:

```
# timedatectl list-timezones | grep UTC
UTC
```

Here, we chose not to apply any timezone changes during the installation.

3.10.2 Preparing the OpenShift cluster

We prepare the OpenShift cluster for the installation by running the **cpd-cli adm apply** command (see Example 3-24).

Example 3-24 Preparing the OpenShift cluster

```
ASSEMBLY="wkc"
ARCH="x86_64"
PROJECT="zen"
STORAGE_CLASS="ibm-spectrum-scale-sc"
PUSHREGISTRY="$(oc registry info)"
PULLREGISTRY="image-registry.openshift-image-registry.svc:5000"

./cpd-cli adm -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} --arch ${ARCH} --apply --accept-all-licenses
```

Here, we show an example if what it looks like if an assembly is installed with an incorrectly defined SC and how it takes to debug it. This example might be helpful to identify, debug, and solve similar issues when new services are deployed that are not described in this publication.

Therefore, we continue to install Watson Knowledge Catalog by using the same default SC (ibm-spectrum-scale-sc) as we used with the previous Cloud Pak for Data services that we know fail in this specific case (see Example 3-25).

Example 3-25 Installing Watson Knowledge catalog

```
./cpd-cli install -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} -c ${STORAGE_CLASS} --arch ${ARCH} \
--transfer-image-to=${PUSHREGISTRY}/${PROJECT} \
--target-registry-username=$(oc whoami) --target-registry-password=$(oc whoami -t) \
--insecure-skip-tls-verify \
--cluster-pull-prefix=image-registry.openshift-image-registry.svc:5000/${PROJECT} \
--latest-dependency --accept-all-licenses
```

Watson Knowledge Catalog requires the Cloud Pak for Data common core services. The common core services are installed once in a specific Red Hat OpenShift project. If the common core services are not installed in the project where you plan to install Watson Knowledge Catalog, the common core services are automatically installed when you install Watson Knowledge Catalog. If the common core services need to be installed, it might take longer to install Watson Knowledge Catalog.

In our environment, Watson Knowledge Catalog required downloading more 95 container images, which took a considerable amount of time.

With the `ibm-spectrum-scale-sc` SC, the installation step of Watson Knowledge Catalog hangs at module `0072-iis x86_64`, as shown in Example 3-26.

Example 3-26 Installation hang at module 0072-iis x86_64

```
2021-02-22 14:47:05.454823723 +0100 CET m=+9223.694075602
Module                      |Instance Name |Status
0010-infra x86_64           |            |Ready
0015-setup x86_64           |            |Ready
0020-core x86_64            |            |Ready
0022-wkc-base-prereqs x86_64      |      |Ready
volume-setup x86_64         |            |Ready
0027-wkc-base x86_64        |            |Ready
dap-base x86_64             |            |Ready
environments x86_64         |            |Ready
runtime-base x86_64         |            |Ready
ibm-job-scheduler-prod x86_64 |          |Ready
spaces x86_64               |            |Ready
wml-ui x86_64               |            |Ready
ibm-dataview-prod x86_64    |            |Ready
ccs-post-install x86_64     |            |Ready
0071-wkc-prereqs x86_64     |            |Ready
0075-wkc-lite x86_64        |            |Ready
data-refinery x86_64        |            |Ready
0072-iis x86_64             |            |In Progress
   ============================
    |Resource |Ready    |Total
    |Deployment |1     |2
    |PVC      |5       |5
    |StateFulSet  |5    |6
    |Job      |3       |3
    |ReplicaSet   |1    |2
============================
0073-ug x86_64 | |To Be Installed
```

Finally, it fails (see Example 3-27).

Example 3-27 Installation failure

```
[INFO] [2021-02-22 16:10:16-0640] Assembly version history update complete
[INFO] [2021-02-22 16:10:16-0790] Collecting cpd install information
[INFO] [2021-02-22 16:10:16-0807] Collecting operator pod log information
[INFO] [2021-02-22 16:10:17-0863] Collecting tiller pod log information
[INFO] [2021-02-22 16:10:18-0202] Collecting configmap information
[INFO] [2021-02-22 16:10:18-0249] Creating the diag tarball
[INFO] [2021-02-22 16:10:18-0566] Install diag file /root/CP4D/install/cp4d-installer/cpd-cli
workspace/CPD-3.5.2-43-2021-02-22T16-10-18.tgz created successfully. Please provide this file
for support
[ERROR] [2021-02-22 16:10:18-0588] Exiting due to error (Module 0072-iis x86_64 has failed).
Please check
/root/CP4D/install/cp4d-installer/cpd-cli-workspace/logs/CPD-2021-02-22T12-13-21.log
for details
[ERROR] 2021-02-22T16:10:18.595557Z Execution error: exit status 1
```

The wkc assembly state after the failed install is shown in bold in Example 3-28.

Example 3-28 wkc assembly state

```
# ./cpd-cli status -a ${ASSEMBLY} -n ${PROJECT}
[INFO] [2021-02-23 15:13:56-0235] Arch override not found. Assuming default architecture x86_64
[INFO] [2021-02-23 15:14:01-0046]
Displaying CR status for all assemblies and relevant modules

Status for assembly wkc and relevant modules in project zen:

Assembly Name                       Status            Version         Arch
wkc                                 Ready             3.5.2           x86_64

  SubAssembly Name                  Status            Version         Arch
  iis                               Not Ready/Failed  3.5.2           x86_64
  lite                              Ready             3.5.2           x86_64
  spaces                            Ready             3.5.2           x86_64
  common-core-services              Ready             3.5.2           x86_64
  dataview                          Ready             3.5.2           x86_64
  spaces-ui                         Ready             3.5.2           x86_64
  wkc-core                          Ready             3.5.2           x86_64
  data-refinery                     Ready             3.5.2           x86_64

  Module Name                       Version           Arch            Storage Class
  0073-ug                           13.5.739          x86_64          ibm-spectrum-scale-sc
=========================================================================================
Status for assembly wkc-core and relevant modules in project zen:

Assembly Name                       Status            Version         Arch
wkc-core                            Ready             3.5.2           x86_64

  subAssembly Name                  Status            Version         Arch
  data-refinery                     Ready             3.5.2           x86_64
  lite                              Ready             3.5.2           x86_64
  spaces                            Ready             3.5.2           x86_64
  common-core-services              Ready             3.5.2           x86_64
  dataview                          Ready             3.5.2           x86_64
  spaces-ui                         Ready             3.5.2           x86_64

  Module Name                       Version           Arch            Storage Class
  0071-wkc-prereqs                  3.5.708           x86_64          ibm-spectrum-scale-sc
  0075-wkc-lite                     3.5.738           x86_64          ibm-spectrum-scale-sc
=========================================================================================
```

As shown in Example 3-28, iis is the sole failed component of this deployment.

3.10.3 Debugging the failed Watson Knowledge Catalog installation

Debugging the issue reveals that the `is-en-conductor-0` pod does not start successfully (see Example 3-29).

Example 3-29 Debugging results

```
# oc get pods | grep -v "Running\|Completed"
NAME                                                READY STATUS     RESTARTS    AGE
is-en-conductor-0                                   0/1   Init:1/2   0          161m
```

The following events show the failing pod.

Example 3-30

Example 3-30 Failing pod

```
Events:
  Type     Reason                  Age                  From                     Message
  ----     ------                  ----                 ----                     -------
  Warning  FailedScheduling        161m (x4 over 161m)  default-scheduler        running "VolumeBinding" filter plugin
for pod "is-en- conductor-0": pod has unbound immediate PersistentVolumeClaims
  Normal   Scheduled               161m                 default-scheduler        Successfully assigned
zen/is-en-conductor-0 to worker03.ocp4.scale.com
  Normal   SuccessfulAttachVolume  161m                 attachdetach-controller  AttachVolume.Attach succeeded for
volume "pvc-309a60b0- ee5d-4d43-8946-b1bf8be9e145"
  Normal   AddedInterface          161m                 multus                   Add eth0 [10.129.3.17/23]
  Normal   Pulling                 161m                 kubelet                  Pulling image
"image-registry.openshift-image- registry.svc:5000/zen/is-engine-image:b231-CP4D-3_5_0-b111"
  Normal   Pulled                  158m                 kubelet                  Successfully pulled image
"image-registry.openshift- image-registry.svc:5000/zen/is-engine-image:b231-CP4D-3_5_0-b111"
  Normal   Created                 158m                 kubelet                  Created container load-data
  Normal   Started                 158m                 kubelet                  Started container load-data
  Normal   Pulled                  158m                 kubelet                  Container image
"image-registry.openshift-image- registry.svc:5000/zen/wkc-init-container:1.0.122" already present on machine
  Normal   Created                 158m                 kubelet                  Created container wait-services
  Normal   Started                 158m                 kubelet                  Started container wait-services
```

Review the container logs of the pod (see Example 3-31).

Example 3-31 Container log

```
# oc logs is-en-conductor-0 --all-containers=true > wkc/wkc_pod_logs.out
Error from server (BadRequest): container "iis-en-conductor" in pod "is-en-conductor-0" is
waiting to start: PodInitializing
```

The logs reveals many storage-related error messages, such as `permission denied` (see Example 3-32).

Example 3-32 Storage-related error messages

```
# cat wkc/wkc_pod_logs.out | grep Permission | sort -u
mkdir: cannot create directory ‘/mnt/dedicated_vol/Engine/is-en-conductor-0’: Permission denied
mkdir: cannot create directory ‘/mnt/dedicated_vol/Engine/temp’: Permission denied
: Permission denied
Projects: Cannot mkdir: Permission denied
StagingArea: Cannot mkdir: Permission denied
tar: ASBNode: Cannot mkdir: Permission denied
tar: Configurations: Cannot mkdir: Permission denied
tar: Datasets: Cannot mkdir: Permission denied
tar: DSEngine: Cannot mkdir: Permission denied
tar: DSODB: Cannot mkdir: Permission denied
tar: DSOMD: Cannot mkdir: Permission denied
tar: DSWLM: Cannot mkdir: Permission denied
tar: Projects: Cannot mkdir: Permission denied
tar: Projects: Cannot mkdir: Permission deniedProjects/ANALYZERPROJECT/DSG_BP.O/DSLogWarn.B
tar: Projects: Cannot mkdir: Permission deniedProjects/ANALYZERPROJECT/DSG_BP.O/DSSetProjectProperty.B
tar: Projects: Cannot mkdir: Permission deniedProjects/ANALYZERPROJECT/Quality/BEMNAD.RCR
tar: Projects: Cannot mkdir: Permission deniedProjects/ANALYZERPROJECT/Quality/JPKOAZA09.TBL
tar: Projects: Cannot mkdir: Permission deniedProjects/ANALYZERPROJECT/Quality/USNAME.RCR
tar: Projects: Cannot mkdir: Permission deniedProjects/dstage1/Quality/BRMNAD.IPO
tar: Projects: Cannot mkdir: Permission deniedProjects/dstage1/Quality/HKCADDR.UCL
tar: Projects: Cannot mkdir: Permission deniedProjects/dstage1/Quality/HKNAME.DCT
tar: Projects: Cannot mkdir: Permission deniedProjects/dstage1/Quality/ITADDR.DCT
tar: Projects: Cannot mkdir: Permission deniedProjects/dstage1/Quality/USNAME.RCR
tar: Scratch: Cannot mkdir: Permission deniedDatasets/
tar: StagingArea: Cannot mkdir: Permission denied
tar: StagingArea: Cannot mkdir: Permission
deniedStagingArea/Installed/AzureFileStorageConnector/META-INF/zh_TW/StageDsc.xml
tar: StagingArea: Cannot mkdir: Permission deniedStagingArea/Installed/DTStage/Server/Shared/
tar: StagingArea: Cannot mkdir: Permission deniedStagingArea/Installed/HDFSConnector/CC_HDFS.jar
tar: StagingArea: Cannot mkdir: Permission
deniedStagingArea/Installed/HDFSFileConnector/META-INF/it/StageDsc.xml
tar: StagingArea: Cannot mkdir: Permission deniedStagingArea/Installed/IADataRule/Server/DSComponents/bin/
tar: StagingArea: Cannot mkdir: Permission deniedStagingArea/Installed/ODBCConnector/META-INF/de/
tar: StagingArea: Cannot mkdir: Permission deniedStagingArea/Installed/ORAOCIBL/Bitmaps/ORAOCIBl.bmp
tar: StagingArea: Cannot mkdir: Permission deniedStagingArea/Installed/ORAOCIBL/META-INF/fr/StageDsc.xml
tar: StagingArea: Cannot mkdir: Permission deniedStagingArea/Installed/ORAOCIBL/META-INF/ja/Deploy.xml
tar: StagingArea: Cannot mkdir: Permission deniedStagingArea/Installed/PS_HRY/META-INF/es/Deploy.xml
tar: StagingArea: Cannot mkdir: Permission deniedStagingArea/Installed/Siebel_DA/META-INF/
tar: StagingArea: Cannot mkdir: Permission deniedStagingArea/Installed/SYBASEOC/META-INF/
tar: StagingArea: Cannot mkdir: Permission
deniedStagingArea/Installed/SYBASEOC/Server/linux64/sybaseocchs.so
tar: StagingArea: Cannot mkdir: Permission deniedStagingArea/Installed/XMLOutput/META-INF/
```

The logs also show `cannot access` messages (see Example 3-33).

Example 3-33   Messages: Cannot access

```
# cat wkc/wkc_pod_logs.out | grep chown | sort -u
chown: cannot access '/mnt/dedicated_vol/Engine/is-en-conductor-0/ASBNode': No such file or directory
chown: cannot access '/mnt/dedicated_vol/Engine/is-en-conductor-0/Datasets': No such file or directory
chown: cannot access '/mnt/dedicated_vol/Engine/is-en-conductor-0/DSEngine/.Fileconnector.ini': No such
file or directory
chown: cannot access '/mnt/dedicated_vol/Engine/is-en-conductor-0/DSWLM': No such file or directory
chown: cannot access '/mnt/dedicated_vol/Engine/is-en-conductor-0/MsgHandlers': No such file or directory
chown: cannot access '/mnt/dedicated_vol/Engine/is-en-conductor-0': No such file or directory
chown: cannot access '/mnt/dedicated_vol/Engine/temp': No such file or directory
+ chown dsadm:dstage /mnt/dedicated_vol/Engine/is-en-conductor-0/DSEngine/.Fileconnector.ini
+ chown dsadm:dstage /mnt/dedicated_vol/Engine/is-en-conductor-0/MsgHandlers
+ chown -R dsadm:dstage /home/dsadm
+ chown -R dsadm:dstage /mnt/dedicated_vol/Engine/is-en-conductor-0
+ chown -R dsadm:dstage /mnt/dedicated_vol/Engine/is-en-conductor-0/ASBNode
+ chown -R dsadm:dstage /mnt/dedicated_vol/Engine/is-en-conductor-0/Datasets
+ chown -R dsadm:dstage /mnt/dedicated_vol/Engine/is-en-conductor-0/DSWLM
+ chown -R dsadm:dstage /mnt/dedicated_vol/Engine/temp
+ chown -R dsadm:dstage /opt/IBM/InformationServer/Server/../ASBNode/bin
+ chown -R dsadm:dstage /opt/IBM/InformationServer/Server/Scratch
```

The PVC behind the `mount /mnt/dedicated_vol` is `0072-iis-en-dedicated-pvc`, as shown in the following example:

```
   volumeMounts:
   - mountPath: /mnt/dedicated_vol/Engine
     name: engine-dedicated-volume
  volumes:
  - name: engine-dedicated-volume
    persistentVolumeClaim:
     claimName: 0072-iis-en-dedicated-pvc
```

It is successfully bound:

```
# oc get pvc 0072-iis-en-dedicated-pvc
NAME                        STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS            AGE
0072-iis-en-dedicated-pvc   Bound    pvc-e00c1089-32e9-4469-88b3-8f8c55dadc53   40Gi       RWX            ibm-spectrum-scale-sc   25h
```

However, it contains no data; that is, it is empty (see Example 3-34).

Example 3-34   Contains no data

```
# ls -al
/gpfs/ess3000_1M/pvc-e00c1089-32e9-4469-88b3-8f8c55dadc53/pvc-e00c1089-32e9-4469-88b3-8f8c55dadc53-data/
total 1
drwxrwx--x. 2 root root 4096 Feb 22 13:33 .
drwxrwx--x. 3 root root 4096 Feb 22 13:33 ..
```

Up to this step, we successfully created 32 PVs with IBM Spectrum Scale as storage provider for the IBM Cloud Pak for Data control plane and the `db2oltp` service:

```
# oc get pvc
NAME                                            STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
           AGE
0071-wkc-p-data-stor-wdp-db2-0                  Bound    pvc-8840b06f-c535-47f7-808e-a905da695b4e 60Gi       RWO            ibm-spectrum-
scale-sc   26h
0072-iis-dedicatedservices-pvc                  Bound    pvc-b4e75cad-f068-4314-a606-b93b0c711245 20Gi       RWO            ibm-spectrum-
scale-sc   26h
0072-iis-dedicatedservices-pvc                  Bound    pvc-b4e75cad-f068-4314-a606-b93b0c711245 20Gi       RWO            ibm-spectrum-
scale-sc   26h
0072-iis-en-dedicated-pvc                       Bound    pvc-e00c1089-32e9-4469-88b3-8f8c55dadc53 40Gi       RWX            ibm-spectrum-
scale-sc   26h
0072-iis-sampledata-pvc                         Bound    pvc-309a60b0-ee5d-4d43-8946-b1bf8be9e145 100Mi      RWX            ibm-spectrum-
scale-sc   26h
0072-iis-xmeta-pvc                              Bound    pvc-c58787be-285e-43a1-a90a-81a007de3d70 100Gi      RWO            ibm-spectrum-
scale-sc   26h
cassandra-data-cassandra-0                      Bound    pvc-897ba7c8-13df-44cf-890a-f360f8eaa3c0 90Gi       RWO            ibm-spectrum-
scale-sc   26h
cc-home-pvc                                     Bound    pvc-b2405d21-a409-48b3-a894-638ee7b10f89 50Gi       RWX            ibm-spectrum-
scale-sc   27h
cpd-install-operator-pvc                        Bound    pvc-e6a6faea-3025-4412-8aed-ee08a687686f 1Gi        RWX            ibm-spectrum-
scale-sc   4d23h
cpd-install-shared-pvc                          Bound    pvc-72f67a18-ec77-49ba-9e40-b6fd902faa61 1Gi        RWX            ibm-spectrum-
scale-sc   4d23h
data-rabbitmq-ha-0                              Bound    pvc-60894705-8dad-4bd9-91d1-126e053c0cbf 10Gi       RWO            ibm-spectrum-
scale-sc   27h
data-rabbitmq-ha-1                              Bound    pvc-c7d7163e-e9f8-4e81-b298-269d717e5059 10Gi       RWO            ibm-spectrum-
scale-sc   27h
data-rabbitmq-ha-2                              Bound    pvc-6e72aaf7-4139-435b-8ba8-2d3e336ea564 10Gi       RWO            ibm-spectrum-
scale-sc   27h
data-redis-ha-server-0                          Bound    pvc-b563a8d7-c0ff-4727-89b5-079cc30508df 10Gi       RWO            ibm-spectrum-
scale-sc   27h
data-redis-ha-server-1                          Bound    pvc-32c63aa7-7c72-43ac-96d0-a7356f6e4f8b 10Gi       RWO            ibm-spectrum-
scale-sc   27h
data-redis-ha-server-2                          Bound    pvc-a9a5cd44-1647-45dc-916e-23438466fcf1 10Gi       RWO            ibm-spectrum-
scale-sc   27h
database-storage-wdp-couchdb-0                  Bound    pvc-760a9b8c-327d-4572-95e3-99b85834f39f 30Gi       RWO            ibm-spectrum-
scale-sc   27h
database-storage-wdp-couchdb-1                  Bound    pvc-5ce6136f-f302-434d-83ce-90194b3e8be9 30Gi       RWO            ibm-spectrum-
scale-sc   27h
database-storage-wdp-couchdb-2                  Bound    pvc-c1222dff-7c6e-4faf-96b5-bb0663109a21 30Gi       RWO            ibm-spectrum-
scale-sc   27h
datadir-zen-metastoredb-0                       Bound    pvc-beaf0d8d-881e-42af-b856-86dfda8f0358 10Gi       RWO            ibm-spectrum-
scale-sc   4d23h
datadir-zen-metastoredb-1                       Bound    pvc-47a7b534-84a7-4fab-b1ea-a64fc613f709 10Gi       RWO            ibm-spectrum-
scale-sc   4d23h
datadir-zen-metastoredb-2                       Bound    pvc-817d3d86-c7e0-4662-9207-f4e9ff4b527b 10Gi       RWO            ibm-spectrum-
scale-sc   4d23h
elasticsearch-master-backups                    Bound    pvc-dc5c66f0-306a-4ced-aa7c-7d4799f8e6d0 30Gi       RWX            ibm-spectrum-
scale-sc   27h
elasticsearch-master-elasticsearch-master-0     Bound    pvc-c44a45a9-bb55-4904-ad5a-5b008d49a17d 30Gi       RWO            ibm-spectrum-
scale-sc   27h
elasticsearch-master-elasticsearch-master-1     Bound    pvc-11fd9d30-efdf-400b-b5df-547392156cd1 30Gi       RWO            ibm-spectrum-
scale-sc   27h
elasticsearch-master-elasticsearch-master-2     Bound    pvc-7ca04221-14bd-4570-9720-fb073e707d29 30Gi       RWO            ibm-spectrum-
scale-sc   27h
file-api-claim                                  Bound    pvc-68f32db8-8f0f-416b-9a55-3657c215f508 30Gi       RWX            ibm-spectrum-
scale-sc   26h
iis-secrets-pv                                  Bound    pvc-0beb4af3-9a80-445d-81f8-a1b656b7e755 1Mi        RWX            ibm-spectrum-
scale-sc   26h
influxdb-pvc                                    Bound    pvc-a9d3467f-56d5-47b9-88ac-89c28a38d697 10Gi       RWX            ibm-spectrum-
scale-sc   4d23h
kafka-data-kafka-0                              Bound    pvc-add06372-6ab1-47a3-acf1-24c4f9f493b6 100Gi      RWO            ibm-spectrum-
scale-sc   26h
solr-data-solr-0                                Bound    pvc-d1f297e4-57dc-4a95-a3a4-3b9eace5b2a4 30Gi       RWO            ibm-spectrum-
scale-sc   26h
user-home-pvc                                   Bound    pvc-bf336703-eace-444f-b00f-b2690fd83fed 10Gi       RWX            ibm-spectrum-
scale-sc   4d23h
zookeeper-data-zookeeper-0                      Bound    pvc-0fa9c5ab-1cd5-4660-831b-f0d6ab55d29f 5Gi        RWO            ibm-spectrum-
scale-sc   26h
```

All of these other Cloud Pak for Data components, such as the control plane and the `db2oltp` service, did not raise any installation errors with IBM Spectrum Scale as storage provider and the same `ibm-spectrum-scale-sc` SC being used.

Further investigation shows that the `is-en-conductor-0` pod, as part of the `iis` sub-assembly, proceeds after a pod restart. This restart is possible if the permissions of the `0072-iis-en-dedicated-pvc` PVC in the IBM Spectrum Scale `pvc-...-data` directory in the fileset that is backing the PV are temporarily adjusted from `drwxrwx--x` to `drwxrwxrwx` by manually running `chmod o+rwx`.

The original permissions are shown in Example 3-35.

Example 3-35 Original permissions

```
# ls -al /gpfs/ess3000_1M/pvc-5c40fb5e-4d70-47b7-b030-31ed763c1c5e/pvc-5c40fb5e-4d70-47b7-b030-
31ed763c1c5e-data/
total 2
drwxrwx--x. 2 root root 4096 Feb 24 17:40 .
drwxrwx--x. 3 root root 4096 Feb 24 13:29 ..
```

The permissions after the temporary change are shown in Example 3-36.

Example 3-36 Permissions after temporary change

```
# ls -al /gpfs/ess3000_1M/pvc-5c40fb5e-4d70-47b7-b030-31ed763c1c5e/pvc-5c40fb5e-4d70-47b7-b030-
31ed763c1c5e-data/
total 2
drwxrwxrwx.  4 root  root 4096 Feb 24 17:40 .
drwxrwx--x.  3 root  root 4096 Feb 24 13:29 ..
drwxr-xr-x. 15 10032 1000 4096 Feb 24 17:41 is-en-conductor-0
drwxr-xr-x.  2 10032 1000 4096 Feb 24 17:40 temp
```

After the pod is restarted, the data access problems disappear in the logs and we can see that data is successfully written into the PV under the `uid 10032` and the `gid 1000`. However, this quick fix does not fully resolve the issue and the pod still does not successfully enter the `Running` state; therefore, the entire installation fails.

Drilling down further shows that the issue is related to the `initContainer` of this pod which uses the following security context:

```
securityContext:
  allowPrivilegeEscalation: true
  capabilities:
    drop:
    - MKNOD
  privileged: false
  readOnlyRootFilesystem: false
  runAsNonRoot: true
runAsUser: 10032
```

The `initContainer` enforces the use of a specific uid (`runAsUser: 10032`), which is outside the regular range of what OpenShift typically assigns to a regular user based on the restricted SCC (`MustRunAsRange`) with a uid in the range of `1xxxxxxxxx`.

PVCs of other components in the same IBM Cloud Pak for Data deployment (for example, the `solr-data-solr-0` PVC, which was created from the same IBM Spectrum Scale `ibm-spectrumscale-sc` SC) show a typical uid of `1000580000` that was assigned by OpenShift with an unchanged gid of `root` (see Example 3-37 on page 85).

Example 3-37 Listing PVCs and uid/gid

```
# ls -al
/gpfs/ess3000_1M/pvc-09527f8c-ee09-439b-aa5c-ddb9d92ff9b8/pvc-09527f8c-ee09-439b-aa5cddb9d92ff9b8-data/
total 9
drwxrwx--x.  17 root  root 4096 Feb 24 13:33 .
drwxrwx--x.  3 root  root 4096 Feb 24 13:29 ..
drwxr-xr-x.  3 1000580000 root 4096 Feb 24 13:32 analysis_shard1_replica_n1
drwxr-xr-x.  3 1000580000 root 4096 Feb 24 13:32 analysis_shard2_replica_n2
drwxrwxrwx. 10 1000580000 root 4096 Dec 21 12:10 configsets
drwxr-xr-x.  3 1000580000 root 4096 Feb 24 13:32 da-datasets_shard1_replica_n1
drwxr-xr-x.  3 1000580000 root 4096 Feb 24 13:32 da-datasets_shard2_replica_n2
drwxr-xr-x.  3 1000580000 root 4096 Feb 24 13:32 discovery_shard1_replica_n1
drwxr-xr-x.  3 1000580000 root 4096 Feb 24 13:32 discovery_shard2_replica_n2
drwxr-xr-x.  3 1000580000 root 4096 Feb 24 13:32 featureScoreUpdates_shard1_replica_n1
drwxr-xr-x.  3 1000580000 root 4096 Feb 24 13:32 featureScoreUpdates_shard2_replica_n2
drwxr-xr-x.  2 1000580000 root 4096 Feb 24 13:30 filestore
drwxr-xr-x.  3 1000580000 root 4096 Feb 24 13:33 ia-analysis_shard1_replica_n1
drwxr-xr-x.  3 1000580000 root 4096 Feb 24 13:33 ia-analysis_shard2_replica_n2
drwxr-xr-x.  3 1000580000 root 4096 Feb 24 13:33 odf_shard1_replica_n1
drwxr-xr-x.  3 1000580000 root 4096 Feb 24 13:33 odf_shard2_replica_n2
-rw-r--r--.  1 1000580000 root 977 Feb 24 13:30 solr.xml
drwxr-xr-x.  2 1000580000 root 4096 Feb 24 13:30 userfiles
```

Here, no specific uid was enforced by the pod. Also, the uid/gid assignment in OpenShift and IBM Spectrum Scale works well, even for a regular user that uses the `restricted` SCC.

By specifically setting the uid to `10032`, the initContainer of the `is-en-conductor-0` pod loses access to the directory that is backing the PV because it also is assigned an arbitrary gid of `1000` by OpenShift in the process (depending on the SCC that is applied). With a uid of `10032` and gid other than root, the pod then loses access permission to read/write in the directory in IBM Spectrum Scale that is backing the PV.

Although most of the other services and sub-components worked well with the IBM Spectrum Scale SC `ibm-spectrum-scale-sc` with default uid/gid settings and the default OpenShift security settings, we must define a new IBM Spectrum Scale SC `ibm-spectrum-scale3-sc`[3] with the uid that matches the uid that was used by the is-en-conductor-0 pod, which leaves the gid unspecified; that is, gid defaults to `root`, as shown in the following example:

```
# cat ibm-spectrum-scale3-sc.yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: ibm-spectrum-scale3-sc
provisioner: spectrumscale.csi.ibm.com
parameters:
  volBackendFs: "fs0"
  clusterId: "215057217487177715"
  uid: "10032"
reclaimPolicy: Delete
```

This SC allows to successfully deploy the Watson Knowledge Catalog and the DataStage service that are described in 3.3.2, "IBM Spectrum Scale Storage Class with dedicated uid setting" on page 39 that also failed at the same sub-component for the same access issues.

For more information, see "Installing DataStage" on page 92.

[3] ibm-spectrum-scale3-sc, see "IBM Spectrum Scale Storage Class with dedicated uid setting" on page 39

3.10.4 Watson Knowledge Catalog installation with a dedicated uid in storage class

Using the `ibm-spectrum-scale3-sc`[4] SC as defined in 3.3.2, "IBM Spectrum Scale Storage Class with dedicated uid setting" on page 39, we can successfully install the Watson Knowledge Catalog service (see Example 3-38).

Example 3-38 Successfully installing Watson Knowledge Catalog service

```
ASSEMBLY="wkc"
ARCH="x86_64"
PROJECT="zen"
STORAGE_CLASS="ibm-spectrum-scale3-sc"
PUSHREGISTRY="$(oc registry info)"
PULLREGISTRY="image-registry.openshift-image-registry.svc:5000"

# ./cpd-cli install -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} -c ${STORAGE_CLASS} --arch ${ARCH} \
--transfer-image-to=${PUSHREGISTRY}/${PROJECT} \
--target-registry-username=$(oc whoami) --target-registry-password=$(oc whoami -t) \
--insecure-skip-tls-verify \
--cluster-pull-prefix=image-registry.openshift-image-registry.svc:5000/${PROJECT} \
--latest-dependency --accept-all-licenses
```

The installation succeeds with the status (see Example 3-39).

Example 3-39 Installation succeeds

```
# ./cpd-cli status -a ${ASSEMBLY} -n ${PROJECT}
Displaying CR status for all assemblies and relevant modules Status for assembly wkc-core and
relevant modules in project zen:
Assembly Name                           Status            Version          Arch
wkc-core                                Ready             3.5.2            x86_64

  SubAssembly Name                      Status            Version          Arch
  dataview                              Ready             3.5.2            x86_64
  lite                                  Ready             3.5.2            x86_64
  spaces-ui                             Ready             3.5.2            x86_64
  spaces                                Ready             3.5.2            x86_64
  common-core-services                  Ready             3.5.2            x86_64
  data-refinery                         Ready             3.5.2            x86_64

  Module Name                           Version           Arch             Storage Class
  0071-wkc-prereqs                      3.5.708           x86_64
ibm-spectrum-scale3-sc
  0075-wkc-lite                         3.5.738           x86_64
ibm-spectrum-scale3-sc

=========================================================================================

Status for assembly wkc and relevant modules in project zen:

Assembly Name                           Status            Version          Arch
wkc                                     Ready             3.5.2            x86_64
```

[4] ibm-spectrum-scale3-sc, see "IBM Spectrum Scale Storage Class with dedicated uid setting" on page 39

```
SubAssembly Name                      Status            Version          Arch
data-refinery                         Ready             3.5.2            x86_64
iis                                   Ready             3.5.2            x86_64
wkc-core                              Ready             3.5.2            x86_64
dataview                              Ready             3.5.2            x86_64
lite                                  Ready             3.5.2            x86_64
spaces-ui                             Ready             3.5.2            x86_64
common-core-services                  Ready             3.5.2            x86_64
spaces                                Ready             3.5.2            x86_64

Module Name                           Version           Arch             Storage Class
0073-ug                               13.5.739          x86_64           ibm-spectrum-scale3-sc
```

Checking for patches does not show any available patches:

```
# ./cpd-cli status -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} --arch ${ARCH}
--patches --available updates
```

To now, IBM Cloud Pak for Data and the following services are installed:

- IBM DB2® (including a created Db2 database)
- Watson Studio
- DataStage
- Watson Knowledge Catalog

Also, the following 36 PVCs were created by Cloud Pak for Data from IBM Spectrum Scale SCs:

```
[root@fscc-sr650-12 cp4d-installer]# oc get pvc
NAME                                          STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
             AGE
0071-wkc-p-data-stor-wdp-db2-0                Bound    pvc-c880832f-055c-407b-b730-346abd8161f4   60Gi       RWO            ibm-spectrum-
scale3-sc   44m
0072-iis-dedicatedservices-pvc                Bound    pvc-b9da378a-010e-4d9b-8368-14ea5e3cb8cd   20Gi       RWO            ibm-spectrum-
scale3-sc   154m
0072-iis-en-dedicated-pvc                     Bound    pvc-3dcace20-9c4f-4890-9f71-bd6b977c2201   40Gi       RWX            ibm-spectrum-
scale3-sc   154m
0072-iis-sampledata-pvc                       Bound    pvc-9c81f45e-e3bc-42b2-a274-9317324cbfba   100Mi      RWX            ibm-spectrum-
scale3-sc   154m
0072-iis-xmeta-pvc                            Bound    pvc-e6c4535e-0f3b-452a-9769-6a8c16fd9885   100Gi      RWO            ibm-spectrum-
scale3-sc   154m
0073-ug-omag-pvc                              Bound    pvc-37461ed8-480f-42ef-aad1-63dcf1132181   5Gi        RWX            ibm-spectrum-
scale3-sc   23m
c-db2oltp-1614095761315872-backup             Bound    pvc-13e4e8bd-42b9-4e3e-a6ca-385c899cd005   100Gi      RWX            ibm-spectrum-
scale-sc    2d19h
c-db2oltp-1614095761315872-meta               Bound    pvc-f2d37c36-0ed6-4a39-82ce-adcab07c9b5c   100Gi      RWX            ibm-spectrum-
scale-sc    2d19h
cassandra-data-cassandra-0                    Bound    pvc-b480adb1-ed26-4542-bdbd-d27f396be3a6   90Gi       RWO            ibm-spectrum-
scale3-sc   154m
cc-home-pvc                                   Bound    pvc-9badfbfd-4a91-403b-8163-7f2a46534caf   50Gi       RWX            ibm-spectrum-
scale-sc    2d1h
cpd-install-operator-pvc                      Bound    pvc-e6a6faea-3025-4412-8aed-ee08a687686f   1Gi        RWX            ibm-spectrum-
scale-sc    7d20h
cpd-install-shared-pvc                        Bound    pvc-72f67a18-ec77-49ba-9e40-b6fd902faa61   1Gi        RWX            ibm-spectrum-
scale-sc    7d20h
data-c-db2oltp-1614095761315872-db2u-0        Bound    pvc-5802d3c1-07fe-4d63-b274-27abb80b999a   100Gi      RWO            ibm-spectrum-
scale-sc    2d19h
data-rabbitmq-ha-0                            Bound    pvc-970c83c5-959e-4a63-b54c-bbc4a3094769   10Gi       RWO            ibm-spectrum-
scale-sc    2d1h
data-rabbitmq-ha-1                            Bound    pvc-67c5ea06-1a65-4ddb-90c3-17bebd7d505d   10Gi       RWO            ibm-spectrum-
scale-sc    2d1h
data-rabbitmq-ha-2                            Bound    pvc-4b87bd74-08b7-45b9-9b63-2154452eef71   10Gi       RWO            ibm-spectrum-
scale-sc    2d1h
data-redis-ha-server-0                        Bound    pvc-c917d855-5c27-4677-8812-d5aeec9269b6   10Gi       RWO            ibm-spectrum-
scale-sc    2d1h
data-redis-ha-server-1                        Bound    pvc-f112a0c4-6997-4cb2-bbb3-03a77e53513d   10Gi       RWO            ibm-spectrum-
scale-sc    2d1h
data-redis-ha-server-2                        Bound    pvc-eae4b46b-af3f-44bb-bc92-a48354614670   10Gi       RWO            ibm-spectrum-
scale-sc    2d1h
database-storage-wdp-couchdb-0                Bound    pvc-f53b88dc-b6c0-42f6-9b22-0b48042ab151   30Gi       RWO            ibm-spectrum-
scale-sc    2d1h
database-storage-wdp-couchdb-1                Bound    pvc-fab7140f-a4bc-475c-8894-85618dd9424a   30Gi       RWO            ibm-spectrum-
scale-sc    2d1h
```

```
database-storage-wdp-couchdb-2               Bound    pvc-d9f7d306-7ad8-4890-98be-9cc116b2e889  30Gi       RWO            ibm-spectrum-
scale-sc    2d1h
datadir-zen-metastoredb-0                    Bound    pvc-beaf0d8d-881e-42af-b856-86dfda8f0358  10Gi       RWO            ibm-spectrum-
scale-sc    7d20h
datadir-zen-metastoredb-1                    Bound    pvc-47a7b534-84a7-4fab-b1ea-a64fc613f709  10Gi       RWO            ibm-spectrum-
scale-sc    7d20h
datadir-zen-metastoredb-2                    Bound    pvc-817d3d86-c7e0-4662-9207-f4e9ff4b527b  10Gi       RWO            ibm-spectrum-
scale-sc    7d20h
elasticsearch-master-backups                 Bound    pvc-b7f0f29f-8f3f-4083-a0c7-e93453447baf  30Gi       RWX            ibm-spectrum-
scale-sc    2d1h
elasticsearch-master-elasticsearch-master-0  Bound    pvc-2669207e-4bc0-41d6-9ad4-4f825412b361  30Gi       RWO            ibm-spectrum-
scale-sc    2d1h
elasticsearch-master-elasticsearch-master-1  Bound    pvc-5256ad3d-6ef0-401d-986c-d0cace66032d  30Gi       RWO            ibm-spectrum-
scale-sc    2d1h
elasticsearch-master-elasticsearch-master-2  Bound    pvc-000d96fb-ed10-4888-9e87-9684d532935b  30Gi       RWO            ibm-spectrum-
scale-sc    2d1h
file-api-claim                               Bound    pvc-cde8eb65-2adc-4091-96be-bfed90f2f11f  30Gi       RWX            ibm-spectrum-
scale-sc    2d1h
iis-secrets-pv                               Bound    pvc-7d0cd95c-9991-42a6-9427-b5cfee606565  1Mi        RWX            ibm-spectrum-
scale3-sc   154m
influxdb-pvc                                 Bound    pvc-a9d3467f-56d5-47b9-88ac-89c28a38d697  10Gi       RWX            ibm-spectrum-
scale-sc    7d20h
kafka-data-kafka-0                           Bound    pvc-e3354940-0a47-48e2-9d3c-e085e92bfcd9  100Gi      RWO            ibm-spectrum-
scale3-sc   154m
solr-data-solr-0                             Bound    pvc-be291254-d896-4d21-aa2f-222cd19ba370  30Gi       RWO            ibm-spectrum-
scale3-sc   154m
user-home-pvc                                Bound    pvc-bf336703-eace-444f-b00f-b2690fd83fed  10Gi       RWX            ibm-spectrum-
scale-sc    7d20h
zookeeper-data-zookeeper-0                   Bound    pvc-8fa9e54f-20fd-4eb8-94a1-24c6aea298cf  5Gi        RWO            ibm-spectrum-
scale3-sc   154m
```

With IBM Cloud Pak for Data and these four services installed, the OpenShift worker node usage is shown in the following example:

```
# for i in 1 2 3; do oc describe node worker0$i.ocp4.scale.com | tail -10 ; done
Allocated resources:
  (Total limits may be over 100 percent, i.e., overcommitted.)
  Resource          Requests      Limits
  --------           --------      ------
  cpu               8403m (21%)    37600m (95%)
  memory            19611Mi (15%)  91856Mi (72%)
  ephemeral-storage  0 (0%)        0 (0%)
  hugepages-1Gi     0 (0%)         0 (0%)
  hugepages-2Mi     0 (0%)         0 (0%)
Events:             <none>
Allocated resources:
  (Total limits may be over 100 percent, i.e., overcommitted.)
  Resource          Requests      Limits
  --------           --------      ------
  cpu               10139m (25%)   40100m (101%)
  memory            27375Mi (21%)  104568Mi (82%)
  ephemeral-storage  0 (0%)        0 (0%)
  hugepages-1Gi     0 (0%)         0 (0%)
  hugepages-2Mi     0 (0%)         0 (0%)
Events:             <none>
Allocated resources:
(Total limits may be over 100 percent, i.e., overcommitted.)
  Resource          Requests      Limits
  --------           --------      ------
  cpu               13849m (35%)      69200m (175%)
  memory            39431204Ki (30%)  156816420Ki (120%)
  ephemeral-storage  0 (0%)           0 (0%)
  hugepages-1Gi     0 (0%)            0 (0%)
  hugepages-2Mi     0 (0%)            0 (0%)
Events:             <none>
```

Watson Knowledge Catalog is now successfully installed on top of IBM Spectrum Scale as a storage provider that uses an SC `ibm-spectrum-scale3-sc` with a dedicated `uid: 10032` to meet the intrinsic requirements of the `is-en-conductor-0` pod in the common `iis` sub-component of the DataStage and Watson Knowledge Catalog service.

Troubleshooting Watson Knowledge Catalog

If you encounter an issue in which catalogs are failing to load after the IBM Spectrum Scale back end storage CNSA pods are restarted, you can complete the steps as described in this section as a workaround.

If you see the following error in the pod logs for the `catalog-api-xxx`, it might be that the `couchdb` instances failed and were vulnerable to release all of the locks it was holding:

```
Error: unknown_error. Reason: Lock obtain timed out:
NativeFSLock@/opt/couchdb/data/search_indexes/shards/00000000-7fffffff/v2_admin.16
23186691/1c51117f82fcda388c9cad50b8bf73ff/write.lock.
```

To resolve this issue, complete the following steps:

1. Manually remove the `write.lock` files from the shard directories of the `couchdb` pods (`wdp-couch-x`):

   ```
   oc exec -i wdp-couchdb-0 - find /opt/couchdb/data/search_indexes/shards/ -name write.lock -type f -delete
   oc exec -i wdp-couchdb-1 - find /opt/couchdb/data/search_indexes/shards/ -name write.lock -type f -delete
   oc exec -i wdp-couchdb-2 - find /opt/couchdb/data/search_indexes/shards/ -name write.lock -type f -delete
   ```

2. Restart the `couchdb` (`wdp-couch-x`) pods.
3. Restart the `catalog-api` pods.

Note: For removing the shard `write.lock` files, the commands provided resembled the following expands:

```
oc exec -i wdp-couchdb-2 -- find /opt/couchdb/data/search_indexes/shards/* -name write.lock -type f -delete
```

However, the `*` caused the search to fail. By removing the `*`, the search/delete of the `write.lock` files was successful.

This form of `write.lock` delete is used when `couchdb` pod instances are starting. This issue might be the reason that this failure occurred; that is, when `couchdb` is starting, it might not be removing the `write.lock` files correctly (after an abnormal termination of the pod):

```
oc exec -i wdp-couchdb-0 - find /opt/couchdb/data/search_indexes/shards/ -name write.lock -type f -delete
oc exec -i wdp-couchdb-1 - find /opt/couchdb/data/search_indexes/shards/ -name write.lock -type f -delete
oc exec -i wdp-couchdb-2 - find /opt/couchdb/data/search_indexes/shards/ -name write.lock -type f -delete
```

3.10.5 Uninstalling Watson Knowledge Catalog

Watson Knowledge Catalog can be uninstalled by successfully running the following **`cpd-cli uninstall`** command:

```
#./cpd-cli uninstall -a ${ASSEMBLY} -n ${PROJECT} --arch ${ARCH}
--include-dependent-assemblies
```

3.11 Installing Watson Studio

For more information about installing Watson Studio, see Ithis web page. If you installed the Watson Knowledge Catalog service and set the time zone for the master node, skip this task. You can also set the time zone after you finish installing your service.

If the service is installed on a remote machine that runs in a different time zone than the master node, the time zone for the master node is overwritten by the time zone for the installer node. This time zone discrepancy results in scheduled jobs that do not run at the correct time (see Chapter 1, "IBM Cloud Pak for Data introduction" on page 1).

The time zone also can be set after installing your service by editing the timezone configmap and changing the time zone string `data.masterTimezone` to the cluster time zone (see this web page). Here, we skip this step because the UTC is the regular time zone setting on OpenShift and our environment (see Example 3-40).

Example 3-40 Preparing the cluster for Watson Studio installation

```
ASSEMBLY="wsl"
ARCH="x86_64"
PROJECT="zen"
STORAGE_CLASS="ibm-spectrum-scale-sc"
PUSHREGISTRY="$(oc registry info)"
PULLREGISTRY="image-registry.openshift-image-registry.svc:5000"

./cpd-cli adm -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} --arch ${ARCH} --apply --accept-all-licenses
```

Next, we install the Watson Studio service (see Example 3-41).

Example 3-41 Installing Watson Studio service

```
./cpd-cli install -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} -c ${STORAGE_CLASS} --arch ${ARCH} \
--transfer-image-to=${PUSHREGISTRY}/${PROJECT} \
--target-registry-username=$(oc whoami) --target-registry-password=$(oc whoami -t) \
--insecure-skip-tls-verify \
--cluster-pull-prefix=image-registry.openshift-image-registry.svc:5000/${PROJECT} \
--latest-dependency --accept-all-licenses
```

We verify the status of the installation by running the following **cpd-cli status** command:

```
# ./cpd-cli status -a ${ASSEMBLY} -n ${PROJECT}
[INFO] [2021-02-24 11:56:46-0694] Arch override not found. Assuming default architecture x86_64
[INFO] [2021-02-24 11:56:49-0502]
Displaying CR status for all assemblies and relevant modules

Status for assembly wsl and relevant modules in project zen:
Assembly Name                   Status       Version          Arch
wsl                             Ready        3.5.1            x86_64

SubAssembly Name                Status       Version          Arch
 spaces                         Ready        3.5.2            x86_64
 lite                           Ready        3.5.2            x86_64
 runtime-addon-py37             Ready        3.5.1            x86_64
 spaces-ui                      Ready        3.5.2            x86_64
 common-core-services           Ready        3.5.2            x86_64
 data-refinery                  Ready        3.5.2            x86_64
 dataview                       Ready        3.5.2            x86_64
```

```
 Module Name                        Version      Arch              Storage Class
 notebooks                          3.5.106      x86_64            ibm-spectrum-scale-sc
 runtime-addon-py36                 3.5.106      x86_64            ibm-spectrum-scale-sc
 ws-ml                              3.5.5469     x86_64            ibm-spectrum-scale-sc
 ibm-0100-model-viewer-prod         3.5.26       x86_64            ibm-spectrum-scale-sc
=========================================================================================
```

We check for patches by running the following command:

```
./cpd-cli status -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} --arch ${ARCH} --patches --available-updates
```

However, no patches are listed. Otherwise, apply the patch by using the following **cpd-cli patch** command, as shown in Example 3-42.

Example 3-42 Applying patches

```
PATCH_NAME="patch_name"
./cpd-cli patch --patch-name ${PATCH_NAME} --action transfer -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT}
--arch ${ARCH} --transfer-image-to=${PUSHREGISTRY}/${PROJECT} --target-registry-username=$(oc whoami)
--target-registry-password=$(oc whoami -t) --insecure-skip-tls-verify --cluster-pull-prefix=imageregistry.
openshift-image-registry.svc:5000/${PROJECT}
```

Watson Studio is now successfully installed and uses IBM Spectrum Scale as the storage provider and the `ibm-spectrum-scale-sc` SC.

After the installation of the Watson Studio and Db2 service (including a created Db2 database), the following volumes are created by Cloud Pak for Data on IBM Spectrum Scale:

```
# oc get pvc
NAME                                             STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
           AGE
c-db2oltp-1614095761315872-backup                Bound    pvc-13e4e8bd-42b9-4e3e-a6ca-385c899cd005   100Gi      RWX            ibm-spectrum-
scale-sc   19h
c-db2oltp-1614095761315872-meta                  Bound    pvc-f2d37c36-0ed6-4a39-82ce-adcab07c9b5c   100Gi      RWX            ibm-spectrum-
scale-sc   19h
cc-home-pvc                                      Bound    pvc-9badfbfd-4a91-403b-8163-7f2a46534caf   50Gi       RWX            ibm-spectrum-
scale-sc   84m
cpd-install-operator-pvc                         Bound    pvc-e6a6faea-3025-4412-8aed-ee08a687686f   1Gi        RWX            ibm-spectrum-
scale-sc   5d20h
cpd-install-shared-pvc                           Bound    pvc-72f67a18-ec77-49ba-9e40-b6fd902faa61   1Gi        RWX            ibm-spectrum-
scale-sc   5d20h
data-c-db2oltp-1614095761315872-db2u-0           Bound    pvc-5802d3c1-07fe-4d63-b274-27abb80b999a   100Gi      RWO            ibm-spectrum-
scale-sc   19h
data-rabbitmq-ha-0                               Bound    pvc-970c83c5-959e-4a63-b54c-bbc4a3094769   10Gi       RWO            ibm-spectrum-
scale-sc   84m
data-rabbitmq-ha-1                               Bound    pvc-67c5ea06-1a65-4ddb-90c3-17bebd7d505d   10Gi       RWO            ibm-spectrum-
scale-sc   83m
data-rabbitmq-ha-2                               Bound    pvc-4b87bd74-08b7-45b9-9b63-2154452eef71   10Gi       RWO            ibm-spectrum-
scale-sc   81m
data-redis-ha-server-0                           Bound    pvc-c917d855-5c27-4677-8812-d5aeec9269b6   10Gi       RWO            ibm-spectrum-
scale-sc   84m
data-redis-ha-server-1                           Bound    pvc-f112a0c4-6997-4cb2-bbb3-03a77e53513d   10Gi       RWO            ibm-spectrum-
scale-sc   83m
data-redis-ha-server-2                           Bound    pvc-eae4b46b-af3f-44bb-bc92-a48354614670   10Gi       RWO            ibm-spectrum-
scale-sc   83m
database-storage-wdp-couchdb-0                   Bound    pvc-f53b88dc-b6c0-42f6-9b22-0b48042ab151   30Gi       RWO            ibm-spectrum-
scale-sc   84m
database-storage-wdp-couchdb-1                   Bound    pvc-fab7140f-a4bc-475c-8894-85618dd9424a   30Gi       RWO            ibm-spectrum-
scale-sc   84m
database-storage-wdp-couchdb-2                   Bound    pvc-d9f7d306-7ad8-4890-98be-9cc116b2e889   30Gi       RWO            ibm-spectrum-
scale-sc   84m
datadir-zen-metastoredb-0                        Bound    pvc-beaf0d8d-881e-42af-b856-86dfda8f0358   10Gi       RWO            ibm-spectrum-
scale-sc   5d20h
datadir-zen-metastoredb-1                        Bound    pvc-47a7b534-84a7-4fab-b1ea-a64fc613f709   10Gi       RWO            ibm-spectrum-
scale-sc   5d20h
datadir-zen-metastoredb-2                        Bound    pvc-817d3d86-c7e0-4662-9207-f4e9ff4b527b   10Gi       RWO            ibm-spectrum-
scale-sc   5d20h
elasticsearch-master-backups                     Bound    pvc-b7f0f29f-8f3f-4083-a0c7-e93453447baf   30Gi       RWX            ibm-spectrum-
scale-sc   84m
elasticsearch-master-elasticsearch-master-0      Bound    pvc-2669207e-4bc0-41d6-9ad4-4f825412b361   30Gi       RWO            ibm-spectrum-
scale-sc   84m
elasticsearch-master-elasticsearch-master-1      Bound    pvc-5256ad3d-6ef0-401d-986c-d0cace66032d   30Gi       RWO            ibm-spectrum-
scale-sc   84m
```

```
elasticsearch-master-elasticsearch-master-2   Bound    pvc-000d96fb-ed10-4888-9e87-9684d532935b   30Gi       RWO            ibm-spectrum-
scale-sc   84m
file-api-claim                                Bound    pvc-cde8eb65-2adc-4091-96be-bfed90f2f11f   30Gi       RWX            ibm-spectrum-
scale-sc   75m
influxdb-pvc                                  Bound    pvc-a9d3467f-56d5-47b9-88ac-89c28a38d697   10Gi       RWX            ibm-spectrum-
scale-sc   5d20h
user-home-pvc                                 Bound    pvc-bf336703-eace-444f-b00f-b2690fd83fed   10Gi       RWX            ibm-spectrum-
scale-sc   5d20h
```

3.12 Installing DataStage

We install DataStage Enterprise following the instructions that are available at this IBM Documentation web page.

First, we prepare the OpenShift cluster and apply all required changes by running the following **cpd-cli** command, as shown in Example 3-43.

Example 3-43 Preparing OpenShift cluster and applying required changes

```
ASSEMBLY="ds"
ARCH="x86_64"
PROJECT="zen"
STORAGE_CLASS="ibm-spectrum-scale3-sc"
PUSHREGISTRY="$(oc registry info)"
PULLREGISTRY="image-registry.openshift-image-registry.svc:5000"

./cpd-cli adm -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} --arch ${ARCH} --apply --accept-all-licenses
```

Note: We use the same `ibm-spectrum-scale3-sc`[a] SC that we used for Watson Knowledge Catalog. Installing the DataStage service fails at the same `iis` sub-component as observed and described in “Installing Watson Knowledge Catalog (wkc)” on page 76 if the default `ibm-spectrum-scale-sc`[b] SC is used.

a. ibm-spectrum-scale3-sc, see “IBM Spectrum Scale Storage Class with dedicated uid setting” on page 39
b. ibm-spectrum-scale-sc, see “IBM Spectrum Scale Storage Class with default uid and gid settings” on page 37

Then, we install the DataStage service by running the following **cpd-cli install** command, as shown in Example 3-44.

Example 3-44 Installing DataStage service

```
./cpd-cli install -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} -c ${STORAGE_CLASS} --arch ${ARCH} \
--transfer-image-to=${PUSHREGISTRY}/${PROJECT} \
--target-registry-username=$(oc whoami) \
--target-registry-password=$(oc whoami -t) --insecure-skip-tls-verify \
--cluster-pull-prefix=image-registry.openshift-image-registry.svc:5000/${PROJECT} \
--latest-dependency --accept-all-licenses
```

With the IBM Spectrum Scale SC `ibm-spectrum-scale3-sc` (see Example 3-45), the installation succeeds.

Example 3-45 Successful installation

```
# cat ibm-spectrum-scale3-sc.yaml
apiVersion: storage.k8s.io/v1 kind: StorageClass
metadata:
  name: ibm-spectrum-scale3-sc
provisioner: spectrumscale.csi.ibm.com
parameters:
```

```
  volBackendFs: "fs0"
  clusterId: "215057217487177715"
  uid: "10032"
reclaimPolicy: Delete
```

The installation succeeds with the status of `Ready` (see Example 3-46).

Example 3-46 Ready status shown

```
# ./cpd-cli status -a ${ASSEMBLY} -n ${PROJECT}
[INFO] [2021-02-26 10:22:37-0741] Arch override not found. Assuming default architecture x86_64
[INFO] [2021-02-26 10:22:38-0552]
Displaying CR status for all assemblies and relevant modules

Status for assembly ds and relevant modules in project zen:

Assembly Name                               Status            Version          Arch
ds                                           Ready             3.5.4            x86_64

  SubAssembly Name                          Status            Version          Arch
  lite                                      Ready             3.5.2            x86_64
  iis                                       Ready             3.5.4            x86_64

  Module Name                               Version           Arch             Storage Class
  0074-ds                           1        1.7.1-fp1.1822    x86_64           ibm-spectrum-scale3-sc
```

We check for patches, but none are available (see Example 3-47).

Example 3-47 No patches available

```
# ./cpd-cli status -r repo.yaml -a ${ASSEMBLY} -n ${PROJECT} --arch ${ARCH} --patches --available-
updates
```

The following PVCs are created (without the Watson Knowledge Catalog service installed):

```
# oc get pvc
NAME                                         STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS
AGE
0072-iis-dedicatedservices-pvc               Bound     pvc-b9da378a-010e-4d9b-8368-14ea5e3cb8cd   20Gi       RWO            ibm-spectrum-
scale3-sc   94m
0072-iis-en-dedicated-pvc                    Bound     pvc-3dcace20-9c4f-4890-9f71-bd6b977c2201   40Gi       RWX            ibm-spectrum-
scale3-sc   94m
0072-iis-sampledata-pvc                      Bound     pvc-9c81f45e-e3bc-42b2-a274-9317324cbfba   100Mi      RWX            ibm-spectrum-
scale3-sc   94m
0072-iis-xmeta-pvc                           Bound     pvc-e6c4535e-0f3b-452a-9769-6a8c16fd9885   100Gi      RWO            ibm-spectrum-
scale3-sc   94m
c-db2oltp-1614095761315872-backup            Bound     pvc-13e4e8bd-42b9-4e3e-a6ca-385c899cd005   100Gi      RWX            ibm-spectrum-
scale-sc    2d18h
c-db2oltp-1614095761315872-meta              Bound     pvc-f2d37c36-0ed6-4a39-82ce-adcab07c9b5c   100Gi      RWX            ibm-spectrum-
scale-sc    2d18h
cassandra-data-cassandra-0                   Bound     pvc-b480adb1-ed26-4542-bdbd-d27f396be3a6   90Gi       RWO            ibm-spectrum-
scale3-sc   94m
cc-home-pvc                                  Bound     pvc-9badfbfd-4a91-403b-8163-7f2a46534caf   50Gi       RWX            ibm-spectrum-
scale-sc    2d
cpd-install-operator-pvc                     Bound     pvc-e6a6faea-3025-4412-8aed-ee08a687686f   1Gi        RWX            ibm-spectrum-
scale-sc    7d19h
cpd-install-shared-pvc                       Bound     pvc-72f67a18-ec77-49ba-9e40-b6fd902faa61   1Gi        RWX            ibm-spectrum-
scale-sc    7d19h
data-c-db2oltp-1614095761315872-db2u-0       Bound     pvc-5802d3c1-07fe-4d63-b274-27abb80b999a   100Gi      RWO            ibm-spectrum-
scale-sc    2d18h
data-rabbitmq-ha-0                           Bound     pvc-970c83c5-959e-4a63-b54c-bbc4a3094769   10Gi       RWO            ibm-spectrum-
scale-sc    2d
data-rabbitmq-ha-1                           Bound     pvc-67c5ea06-1a65-4ddb-90c3-17bebd7d505d   10Gi       RWO            ibm-spectrum-
scale-sc    2d
data-rabbitmq-ha-2                           Bound     pvc-4b87bd74-08b7-45b9-9b63-2154452eef71   10Gi       RWO            ibm-spectrum-
scale-sc    2d
```

```
data-redis-ha-server-0                         Bound    pvc-c917d855-5c27-4677-8812-d5aeec9269b6  10Gi       RWO           ibm-spectrum-
scale-sc    2d
data-redis-ha-server-1                         Bound    pvc-f112a0c4-6997-4cb2-bbb3-03a77e53513d  10Gi       RWO           ibm-spectrum-
scale-sc    2d
data-redis-ha-server-2                         Bound    pvc-eae4b46b-af3f-44bb-bc92-a48354614670  10Gi       RWO           ibm-spectrum-
scale-sc    2d
database-storage-wdp-couchdb-0                 Bound    pvc-f53b88dc-b6c0-42f6-9b22-0b48042ab151  30Gi       RWO           ibm-spectrum-
scale-sc    2d
database-storage-wdp-couchdb-1                 Bound    pvc-fab7140f-a4bc-475c-8894-85618dd9424a  30Gi       RWO           ibm-spectrum-
scale-sc    2d
database-storage-wdp-couchdb-2                 Bound    pvc-d9f7d306-7ad8-4890-98be-9cc116b2e889  30Gi       RWO           ibm-spectrum-
scale-sc    2d
datadir-zen-metastoredb-0                      Bound    pvc-beaf0d8d-881e-42af-b856-86dfda8f0358  10Gi       RWO           ibm-spectrum-
scale-sc    7d19h
datadir-zen-metastoredb-1                      Bound    pvc-47a7b534-84a7-4fab-b1ea-a64fc613f709  10Gi       RWO           ibm-spectrum-
scale-sc    7d19h
datadir-zen-metastoredb-2                      Bound    pvc-817d3d86-c7e0-4662-9207-f4e9ff4b527b  10Gi       RWO           ibm-spectrum-
scale-sc    7d19h
elasticsearch-master-backups                   Bound    pvc-b7f0f29f-8f3f-4083-a0c7-e93453447baf  30Gi       RWX           ibm-spectrum-
scale-sc    2d
elasticsearch-master-elasticsearch-master-0    Bound    pvc-2669207e-4bc0-41d6-9ad4-4f825412b361  30Gi       RWO           ibm-spectrum-
scale-sc    2d
elasticsearch-master-elasticsearch-master-1    Bound    pvc-5256ad3d-6ef0-401d-986c-d0cace66032d  30Gi       RWO           ibm-spectrum-
scale-sc    2d
elasticsearch-master-elasticsearch-master-2    Bound    pvc-000d96fb-ed10-4888-9e87-9684d532935b  30Gi       RWO           ibm-spectrum-
scale-sc    2d
file-api-claim                                 Bound    pvc-cde8eb65-2adc-4091-96be-bfed90f2f11f  30Gi       RWX           ibm-spectrum-
scale-sc    2d
iis-secrets-pv                                 Bound    pvc-7d0cd95c-9991-42a6-9427-b5cfee606565  1Mi        RWX           ibm-spectrum-
scale3-sc   94m
influxdb-pvc                                   Bound    pvc-a9d3467f-56d5-47b9-88ac-89c28a38d697  10Gi       RWX           ibm-spectrum-
scale-sc    7d19h
kafka-data-kafka-0                             Bound    pvc-e3354940-0a47-48e2-9d3c-e085e92bfcd9  100Gi      RWO           ibm-spectrum-
scale3-sc   94m
solr-data-solr-0                               Bound    pvc-be291254-d896-4d21-aa2f-222cd19ba370
30Gi       RWO           ibm-spectrum- scale3-sc   94m
user-home-pvc                                  Bound    pvc-bf336703-eace-444f-b00f-b2690fd83fed
10Gi       RWX           ibm-spectrum- scale-sc    7d19h
zookeeper-data-zookeeper-0                     Bound    pvc-8fa9e54f-20fd-4eb8-94a1-24c6aea298cf
5Gi        RWO           ibm-spectrum- scale3-sc   94m
```

DataStage is now successfully installed on top of IBM Spectrum Scale as the storage provider that uses an SC `ibm-spectrum-scale3-sc` with a dedicated `uid: 10032`. This configuration meets the intrinsic requirements of the `is-en-conductor-0 pod` in common `iis` sub-components of the DataStage and Watson Knowledge Catalog service.

After installing DataStage, complete the following post-installation steps to enable persistent storage for DataStage:

1. Edit the `statefulset`:

   ```
   oc edit sts/is-en-conductor
   ```

2. Insert the following entry under the `VolumeMounts` file (upon saving, the pod restarts):

   ```
   VolumeMounts:
   mountPath: /home/dsadm
   name: engine-dedicated-volume
   subPath: is-en-conductor-0/EngineClients/db2_client/dsadm
   ```

3. Copy the file back into the pod:

   ```
   oc cp colleges.csv -n zen-automated is-en-conductor-0:/home/dsadm oc exec -n
   zen-automated is-en-conductor-0 -- ls /home/dsadm
   colleges.csv
   ```

   The file appears on the storage node:

   ```
   ls -al
   pvc-55074487-34ca-4d71-868f-e0aee66ddf26/pvc-55074487-34ca-4d71-868f-e0aee66ddf
   26-data/is-en-conductor-0/EngineClients/db2_client/dsadm/ total 162 drwxrwx--x.
   3 root  root       4096 Jun  2 17:02 . drwxr-x--x. 3 root  root       4096 Jun
   2 15:08 .. -rw-r--r--. 1 10032 stgadmin 160691 Jun  2 17:02 colleges.csv
   ```

```
-rw-r--r--. 1 10032 stgadmin      0 Jun  2 15:08 .extractComplete drwxrw----. 3
10032 stgadmin   4096 Jun  2 17:01 .pki
```

4. Restart the pod and check if files persisted:

```
oc delete pod is-en-conductor-0
pod "is-en-conductor-0" deleted
oc exec -n zen-automated is-en-conductor-0 -- ls /home/dsadm
colleges.csv
imam_logs
```

Note that the directory is still owned by `root:root`:

```
ls -al
pvc-55074487-34ca-4d71-868f-e0aee66ddf26/pvc-55074487-34ca-4d71-868f-e0aee66ddf
26-data/is-en-conductor-0/EngineClients total 2 drwxr-xr-x.  3 10032 stgadmin
4096 Jun  2 15:08 . drwxr-xr-x. 19 10032 stgadmin 4096 Jun  2 14:45 ..
drwxr-x--x.  3 root  root     4096 Jun  2 15:08 db2_client
```

If the pod was restarted since installation (or if the files that are in `/home/dsadm` by default were deleted), ensure that the following files exist within `/home/dsadm`:

```
oc exec -n zen-automated is-en-conductor-0 -- ls -al /home/dsadm total 204 drwxrwx--x. 5
root  root     4096 Jun  3 17:19 . drwxr-xr-x. 1 root  root       51 May  5 04:06 ..
-rw-------. 1 dsadm dstage    55 Jun  3 16:21 .bash_history -rwxr-xr-x. 1 dsadm dstage
18 Aug 21  2019 .bash_logout -rwxr-xr-x. 1 dsadm dstage    193 Aug 21  2019
.bash_profile
-rwxr-xr-x. 1 dsadm dstage    344 May  5 04:06 .bashrc
drwxr-xr-x. 2 dsadm dstage   4096 May  5 03:47 ds_logs
-rw-r--r--. 1 dsadm dstage      0 Jun  3 17:19 .extractComplete
drwxr-xr-x. 2 dsadm dstage   4096 Jun  3 17:22 imam_logs
drwxrw----. 3 dsadm dstage   4096 Jun  3 00:01 .pki
```

If any files are missing, delete `.extractComplete` and restart the pod:

```
oc exec -n zen-automated is-en-conductor-0 -- rm -f /home/dsadm/.extractComplete
```

3.12.1 Uninstalling DataStage

Uninstall DataStage by running the **`cpd-cli uninstall`** command (see Example 3-48):

Example 3-48 Uninstalling DataStage

```
#./cpd-cli uninstall -a ${ASSEMBLY} -n ${PROJECT} --arch ${ARCH} --include-dependent-assemblies
```

3.13 Installing Db2 Warehouse

To add a Db2 Warehouse database to IBM Cloud Pak for Data, prepare your Red Hat OpenShift cluster and then, install and deploy the database.

Complete the following steps:

1. From your installation node, change to the directory where you placed the Cloud Pak for Data `cli` and run the following command to prepare your Red Hat OpenShift cluster:

```
./cpd-cli adm -repo ./repo.yaml --assembly db2wh --arch x86_64 --namespace zen
--accept-all-licenses --apply
```

2. Run the following command to grant cpd-admin-role to the project administration user:

   oc adm policy add-role-to-user cpd-admin-role ocpadmin --role-namespace=zen -n zen

3. On the machine from which you run the commands, log in to the cluster as an administrator and set the following environment variables:

```
ASSEMBLY="db2wh"
ARCH="x86_64"
PROJECT="zen"
STORAGE_CLASS="ibm-spectrum-scale-sc"
PUSHREGISTRY="$(oc registry info)"
PULLREGISTRY="image-registry.openshift-image-registry.svc:5000"
```

4. Run the following command to see a preview of what will be installed when you install the service:

```
./cpd-cli install --repo ./repo.yaml --assembly db2wh --arch x86_64 --namespace
zen --storageclass ibm-spectrum-scale-sc --transfer-image-to
registry.cpst-lab.no-users.ibm.com --cluster-pull-prefix
registry.cpst.lab.no-users.ibm.com --ask-push-registry-credentials
--latest-dependency --dry-run
```

> **Important:** If you use the internal Red Hat OpenShift registry and the default self-signed certificate, specify the `--insecure-skip-tls-verify` flag to prevent x509 errors.

   Consider the following points:

   – By default, this command receives the latest version of the assembly. If you want to install a specific version of Db2 Warehouse, add the following line to your command after the `--assembly flag`:

   ```
   --version Assembly_version
   ```

   – The `--latest-dependency` flag receives the latest version of the dependent assemblies. If you remove the `--latest-dependency` flag, the installer receives the minimum version of the dependent assemblies.

   – Ensure that you use the same flags that you used during cluster set up. If you used the `--version` flag, ensure that you specify the same version of the assembly.

5. To install `db2wh`, run the following command (the same command as used in Step 4, but without the `--dry-run` flag):

```
./cpd-cli install --repo ./repo.yaml --assembly db2wh --arch x86_64 --namespace
zen --storageclass ibm-spectrum-scale-sc --transfer-image-to
registry.cpst-lab.no-users.ibm.com --cluster-pull-prefix
```

```
registry.cpst.lab.no-users.ibm.com --ask-push-registry-credentials
--latest-dependency
```

6. Verify that the installation complete successfully by running the following command:

```
./cpd-cli status --assembly db2wh --namespace zen
```

7. Check for available patches by running the following command:

```
./cpd-cli status --repo ./repo.yaml --namespace zen --assembly db2wh --patches
--available-updates
```

Fro more information about creating a database deployment, see this web page.

3.13.1 Uninstalling Db2 Warehouse

Complete the following steps to uninstall Db2 Warehouse:

1. Log in to the Cloud Pak for Data web client as an administrator.
2. From the menu, select **Services** → **Instances**.
3. Filter the list to show only `db2wh` service instances.
4. Delete all of the instances of the service.
5. Run the following command:

```
./cpd-cli uninstall --assembly db2wh --namespace zen --profile cpst-test-profile
```

3.14 Installing Watson Machine Learning

For more information about setting up Watson Machine Learning on Cloud Pak for Data, see IBM Documentation.

Complete the following steps:

1. Run the following command to prepare your Red Hat OpenShift cluster:

```
./cpd-cli adm -repo ./repo.yaml --assembly wml --arch x86_64 --namespace zen
--accept-all-licenses --apply
```

2. Run the following command to grant `cpd-admin-role` to the project administration user:

```
oc adm policy add-role-to-user cpd-admin-role ocpadmin --role-namespace=zen -n zen
```

3. On the machine from which you run the commands, log in to the cluster as an administrator and set the environment variables:

```
ASSEMBLY="wml"
ARCH="x86_64"
PROJECT="zen"
STORAGE_CLASS="ibm-spectrum-scale-sc"
PUSHREGISTRY="$(oc registry info)"
PULLREGISTRY="image-registry.openshift-image-registry.svc:5000"
```

4. Run the following command to see a preview of what is installed when you install the service:

```
./cpd-cli install --repo ./repo.yaml --assembly wml --arch x86_64 --namespace
zen --storageclass ibm-spectrum-scale-sc --transfer-image-to
registry.cpst-lab.no-users.ibm.com --cluster-pull-prefix
registry.cpst.lab.no-users.ibm.com --ask-push-registry-credentials
--latest-dependency --dry-run
```

5. Verify that the installation completed successfully by running the following command:

```
./cpd-cli status --assembly wml --namespace zen
```

6. Check for available patches by running the following command:

```
./cpd-cli status --repo ./repo.yaml --namespace zen --assembly wml --patches
--available-updates
```

3.14.1 Uninstalling Watson Machine Learning

Run the following command to uninstall Watson Machine Learning:

```
./cpd-cli uninstall --assembly wml --namespace zen --profile cpst-test-profile
```

3.15 Installing Watson OpenScale

Within IBM Cloud Pak for Data, you can deploy Watson Machine Learning to serve as the machine learning provider, Db2 to serve as the database, and Watson Studio to provide notebook authoring and model creation. These services are dependent on each other and should be installed to run Watson OpenScale and tutorials.

> **Note:** To function correctly, Watson Machine Learning must be installed from the same instance of IBM Cloud Pak for Data before the Watson OpenScale service is installed.

For Db2 options that are part of your cluster, see Data Sources, which is where you find options to include in your service, such as IBM Db2 Warehouse and IBM Db2.

Complete the following steps to install Watson OpenScale (for more information about setting up Watson OpenScale on Cloud Pak for Data, see IBM Documentation):

1. Run the following command to prepare your Red Hat OpenShift cluster:

```
./cpd-cli adm -repo ./repo.yaml --assembly aiopenscale --arch x86_64
--namespace zen --accept-all-licenses --apply
```

2. Run the following command to grant `cpd-admin-role` to the project administration user:

```
oc adm policy add-role-to-user cpd-admin-role ocpadmin --role-namespace=zen -n zen
```

3. On the machine from which you run the commands, log in to the cluster as an administrator and set the following environment variables:

```
ASSEMBLY="aiopenscale"
ARCH="x86_64"
PROJECT="zen"
STORAGE_CLASS="ibm-spectrum-scale-sc"
PUSHREGISTRY="$(oc registry info)"
PULLREGISTRY="image-registry.openshift-image-registry.svc:5000"
```

4. Run the following command to see a preview of what is installed when you install the service:

```
./cpd-cli install --repo ./repo.yaml --assembly aiopenscale --arch x86_64
--namespace zen --storageclass ibm-spectrum-scale-sc --transfer-image-to
registry.cpst-lab.no-users.ibm.com --cluster-pull-prefix
registry.cpst.lab.no-users.ibm.com --ask-push-registry-credentials
--latest-dependency --dry-run
```

5. Verify that the installation completed successfully by running the following command:

```
./cpd-cli status --assembly aiopenscale --namespace zen
```

6. Check for available patches by running the following command:

```
./cpd-cli status --repo ./repo.yaml --namespace zen --assembly aiopenscale
--patches --available-updates
```

3.15.1 Uninstalling Watson OpenScale (aiopenscale)

Run the following command to uninstall Watson OpenScale:

```
./cpd-cli uninstall --assembly aiopenscale --namespace zen --profile
cpst-test-profile
```

3.16 Installing Data Virtualization

Complete the following steps to install Data Virtualization (for more information about setting up Data Virtualization on Cloud Pak for Data, see IBM Documentation):

1. Run the following command to prepare your Red Hat OpenShift cluster:

```
./cpd-cli adm -repo ./repo.yaml --assembly dv --arch x86_64 --namespace zen
--accept-all-licenses --apply
```

2. Run the following command to grant `cpd-admin-role` to the project administration user:

```
oc adm policy add-role-to-user cpd-admin-role ocpadmin --role-namespace=zen -n
zen
```

3. On the machine from which you run the commands, log in to the cluster as an administrator and set the following environment variables:

```
ASSEMBLY="dv"
ARCH="x86_64"
PROJECT="zen"
STORAGE_CLASS="ibm-spectrum-scale-sc"
PUSHREGISTRY="$(oc registry info)"
PULLREGISTRY="image-registry.openshift-image-registry.svc:5000"
```

4. Run the following command to see a preview of what is installed when you install the service:

```
./cpd-cli install --repo ./repo.yaml --assembly dv --arch x86_64 --namespace
zen --storageclass ibm-spectrum-scale-sc --transfer-image-to
registry.cpst-lab.no-users.ibm.com --cluster-pull-prefix
registry.cpst.lab.no-users.ibm.com --ask-push-registry-credentials
--latest-dependency --dry-run
```

5. Verify that the installation completed successfully by running the following command:

```
./cpd-cli status --assembly dv --namespace zen
```

6. Check for available patches by running the following command:

```
./cpd-cli status --repo ./repo.yaml --namespace zen --assembly dv --patches
--available-updates
```

3.16.1 Uninstalling Data Virtualization

Run the following command to uninstall Data Virtualization:

```
./cpd-cli uninstall --assembly dv --namespace zen --profile cpst-test-profile
```

3.17 Installing Apache Spark

Complete the following steps to install Apache Spark (for more information about setting up Apache Spark on Cloud Pak for Data, see IBM Documentation):

1. Run the following command to prepare your Red Hat OpenShift cluster:

   ```
   ./cpd-cli adm -repo ./repo.yaml --assembly spark --arch x86_64 --namespace zen
   --accept-all-licenses --apply
   ```

2. Run the following command to grant `cpd-admin-role` to the project administration user:

   ```
   oc adm policy add-role-to-user cpd-admin-role ocpadmin --role-namespace=zen -n zen
   ```

3. On the machine from which you run the commands, log in to the cluster as an administrator and set the following environment variables:

   ```
   ASSEMBLY="spark"
   ARCH="x86_64"
   PROJECT="zen"
   STORAGE_CLASS="ibm-spectrum-scale-sc"
   PUSHREGISTRY="$(oc registry info)"
   PULLREGISTRY="image-registry.openshift-image-registry.svc:5000"
   ```

4. Run the following command to see a preview of what is installed when you install the service:

   ```
   ./cpd-cli install --repo ./repo.yaml --assembly spark --arch x86_64 --namespace
   zen --storageclass ibm-spectrum-scale-sc --transfer-image-to
   registry.cpst-lab.no-users.ibm.com --cluster-pull-prefix
   registry.cpst.lab.no-users.ibm.com --ask-push-registry-credentials
   --latest-dependency --dry-run
   ```

5. Verify that the installation completed successfully by running the following command:

   ```
   ./cpd-cli status -assembly spark --namespace zen
   ```

6. Check for available patches by running the following command:

   ```
   ./cpd-cli status --repo ./repo.yaml --namespace zen --assembly spark --patches
   --available-updates
   ```

3.17.1 Uninstalling Apache Spark

Run the following command uninstall Apache Spark:

```
./cpd-cli uninstall --assembly spark --namespace zen --profile cpst-test-profile
```

3.18 Installing Db2 Data Management Console

In this section, we describe how to install the Db2 Data Management Console.

3.18.1 Setting up the cluster for Db2 Data Management Console

Complete the following steps to install Db2 Data Management Console (for more information about setting it up on Cloud Pak for Data, see IBM Documentation):

1. Log in to the cluster as an administrator:

   ```
   oc login -u ocpadmin
   ```

2. Run the following command to see a preview of the list of resources that must be created on the cluster:

   ```
   ./cpd-cli adm \
   --repo ./repo.yaml \
   --assembly dmc \
   --arch x86_64 \
   --namespace zen \
   --accept-all-licenses \
   --latest-dependency
   ```

3. To automatically apply the changes to your cluster, re-run the **cpd adm** command with the `--apply` flag:

   ```
   ./cpd-cli adm \
   --repo ./repo.yaml \
   --assembly dmc \
   --arch x86_64 \
   --namespace zen \
   --accept-all-licenses \
   --apply
   ```

4. Run the following command to grant `cpd-admin-role` to the project administration user:

   ```
   oc adm policy add-role-to-user cpd-admin-role ocpadmin --role-namespace=zen -n zen
   ```

3.18.2 Installing Db2 Data Management Console

Complete the following steps to install Data Management Console:

1. On the machine from which you run the commands, log in to the cluster as an administrator:

   ```
   oc login -u ocpadmin
   ```

2. Set up the environment variables:

   ```
   ASSEMBLY="dmc"
   ARCH="x86_64"
   PROJECT="zen"
   STORAGE_CLASS="ibm-spectrum-scale-sc"
   PUSHREGISTRY="$(oc registry info)"
   PULLREGISTRY="image-registry.openshift-image-registry.svc:5000"
   ```

3. Run the following command to see a preview of what is installed when you install the service:

   ```
   ./cpd-cli install \
   ```

```
--repo ./repo.yaml \
--assembly dmc \
--arch x86_64 \
--namespace zen \
--storageclass ibm-spectrum-scale-sc \
--transfer-image-to registry.cpst-lab.no-users.ibm.com \
--cluster-pull-prefix registry.cpst-lab.no-users.ibm.com \
--ask-push-registry-credentials \
--latest-dependency \
--dry-run
Username: oc whoami
Password: oc whoami -t
```

4. Rerun the command that was used in Step 3 without the `--dry-run` flag to install the service.
5. Verifying that the installation completed successfully:

```
./cpd-cli status \
--assembly dmc \
--namespace zen
```

Sample output is shown in Example 3-49.

Example 3-49 Sample output

```
./cpd-cli status -a dmc -n zen
[INFO] [2021-05-03 10:39:20-0776] Arch override not found. Assuming default architecture x86_64
[INFO] [2021-05-03 10:39:21-0380]
Displaying CR status for all assemblies and relevant modules

Status for assembly dmc and relevant modules in project zen:

Assembly Name                Status          Version          Arch
dmc                          Ready           3.5.3            x86_64

  SubAssembly Name           Status          Version          Arch
  lite                       Ready           3.5.2            x86_64

  Module Name                Version         Arch             Storage Class
  dmc                        3.5.3           x86_64          ibm-spectrum-scale-sc

=========================================================================================
```

6. Check for available patches:

```
./cpd-cli status \
--repo ./repo.yaml \
--namespace zen \
--assembly dmc \
--patches \
--available-updates
```

3.18.3 Uninstalling Data Management Console

Complete the following steps to uninstall Data Management Console:

1. On the machine from which you run the commands, log in to the cluster as an administrator:

   ```
   oc login -u ocpadmin
   ```

2. Run the following command to preview what is removed:

   ```
   ./cpd-cli uninstall \
   --assembly dmc \
   --namespace zen \
   --profile cpst-test-profile
   --dry-run
   ```

3. Rerun the command that was used in Step 2 without the `--dry-run` flag to remove Data Management Console.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper.

Online resources

These websites are also relevant as further information sources:

- IBM Spectrum Scale

  https://www.ibm.com/products/spectrum-scale
- IBM Spectrum Scale Container Native 5.1.0.3

  https://www.ibm.com/docs/en/scalecontainernative?topic=5103-overview
- IBM Spectrum Scale Container Storage Interface Driver 2.1.0

  https://www.ibm.com/docs/en/spectrum-scale-csi?topic=spectrum-scale-container-storage-interface-driver-210
- IBM Cloud Pak for Data v 3.5.0

  https://www.ibm.com/docs/en/cloud-paks/cp-data/3.5.0?topic=planning-system-requirements
- IBM Documentation Overview of IBM Cloud Pak for Data

  https://www.ibm.com/support/producthub/icpdata/docs/content/SSQNUZ_latest/cpd/overview/overview.html
- IBM Cloud catalog

  https://cloud.ibm.com/catalog?category=ai#services

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

IBM®

REDP-5652-00

ISBN 0738460095

Printed in U.S.A.