

Implementing, Tuning, and Optimizing Workloads with Red Hat OpenShift on IBM Power

Dino Quintero

Tim Simon

Tushar Agrawal

Sambasiva Andaluri

Shahid Ali

Daniel Casali

Munshi Hafizul Haque

Diogo Horta

Shrirang Kulkarni

Nick Lawrence

Laszlo Niesz

Gabriel Padilla

Gustavo Santos

Shiv Tiwari

Sundaragopal Venkatraman



 **Cloud**

Power Systems

In partnership with
IBM Academy of Technology



IBM Redbooks

**Implementing, Tuning, and Optimizing Workloads with
Red Hat OpenShift on IBM Power**

June 2023

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

First Edition (June 2023)

This edition applies to:

- ▶ Red Hat OpenShift Local 2.10
- ▶ Red Hat Enterprise Linux 8.6 (Ootpa)
- ▶ Red Hat Enterprise Linux 8.5 ppc64le Linux OS
- ▶ Red Hat OpenShift 4.8.23,
- ▶ Red Hat OpenShift Data Foundation (Previously OCS) 4.8
- ▶ Red Hat OpenShift Container Platform 4.10
- ▶ IBM Cloud Pak for Data 4.6
- ▶ IBM Cloud Pak for Business Automation (IBM CP4BA) 22.0.2
- ▶ IBM Cloud Pak for Integration (IBM CP4I) 2021.4
- ▶ IBM Cloud Pak for Integration 2022.4
- ▶ IBM Cloud Pak for Watson AIOps 3.6.1
- ▶ IBM Cloud Pak for WebSphere Hybrid Edition 5.1.0
- ▶ IBM Storage Scale (previously IBM Spectrum Scale) 5.1.5
- ▶ IBM Instana Observability 1.0.232

© Copyright International Business Machines Corporation 2023. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Trademarks	viii
Preface	ix
Authors	ix
Now you can become a published author, too!	xii
Comments welcome	xiii
Stay connected to IBM Redbooks	xiii
Chapter 1. Introduction	1
1.1 Adapting to a new infrastructure paradigm	2
1.1.1 Cloud benefits	2
1.1.2 New performance paradigm	3
1.2 Red Hat OpenShift on IBM Power	3
1.2.1 Red Hat OpenShift	4
1.3 IBM Power servers	6
1.4 Summary	7
Chapter 2. Performance and tuning	9
2.1 Definitions	10
2.1.1 Performance components	10
2.1.2 Performance tuning terminology	11
2.1.3 Service-level agreement, service-level objective, and service-level indicator. . . .	12
2.2 Models	13
2.2.1 Queuing theory	13
2.2.2 Little's Law	13
2.2.3 The four Golden Signals	13
2.2.4 USE method	14
2.2.5 RED method	14
2.3 An example use case scenario	14
2.4 Red Hat OpenShift performance baseline	18
2.4.1 Red Hat OpenShift Container Platform: baseline best practices	18
2.4.2 Cluster performance considerations	18
2.4.3 Node host best practices	18
2.4.4 Control plane node sizing	19
2.4.5 etcd best practices	20
2.4.6 Red Hat OpenShift Container Platform infrastructure baseline considerations . .	20
2.4.7 Infrastructure node sizing	22
2.4.8 Optimizing network performance	22
2.4.9 Storage considerations	23
2.4.10 Other considerations	24
2.5 Red Hat OpenShift starting configuration	25
2.6 Tools	26
2.6.1 Observability	26
2.6.2 IBM Instana	26
2.6.3 IBM Turbonomic Application Resource Management	35
2.6.4 Sysdig	36
2.6.5 Prometheus	38
2.6.6 Grafana	41

Chapter 3. IBM Power processor performance capabilities	43
3.1 IBM Power hardware	44
3.1.1 IBM Power10 processor-based server capabilities	44
3.1.2 IBM Power10 processor-based server packaging	46
3.1.3 IBM Power10 processor	50
3.1.4 IBM Power10 processor core	53
3.1.5 Simultaneous multi-threading	56
3.1.6 Matrix Math Accelerator AI workload acceleration	56
3.1.7 On-chip L3 cache and intelligent caching	58
3.1.8 Open Memory Interface	58
3.1.9 Pervasive memory encryption	59
3.1.10 Nest accelerator	60
3.1.11 SMP interconnect and accelerator interface	61
3.1.12 IBM Power and performance management	63
3.2 IBM Power Systems Virtual Server	66
3.2.1 Architecture	67
3.2.2 Capabilities	70
3.2.3 Ecosystem	71
3.3 Components	71
3.3.1 Software-defined storage	72
3.3.2 Software-defined networking	82
3.3.3 I/O operations per second	86
3.3.4 Tier 1 and Tier 3 storage	86
3.3.5 Fibre Channel	87
3.3.6 Network File System	88
3.3.7 Network	88
3.3.8 Single root I/O virtualization	88
3.3.9 Partition mobility	92
Chapter 4. Red Hat OpenShift architecture and design	95
4.1 Design considerations for Red Hat OpenShift	96
4.2 Red Hat OpenShift capabilities on IBM Power	96
4.3 IBM Cloud Paks capabilities	97
4.4 Red Hat OpenShift architecture	100
4.4.1 Enterprise Kubernetes	102
4.4.2 Classic Red Hat OpenShift 4 components	105
4.4.3 Red Hat OpenShift Local (formerly Red Hat CodeReady Containers)	107
4.4.4 High availability for master nodes	114
4.4.5 Disaster recovery	116
4.5 Red Hat OpenShift ecosystem	125
4.5.1 Operator Lifecycle Manager	127
4.5.2 Service Mesh	134
4.5.3 DevOps and CI/CD pipelines	135
4.5.4 GitOps for Red Hat OpenShift node tuning and configuration	136
4.6 Running Red Hat OpenShift on IBM Power	140
4.6.1 Red Hat OpenShift on IBM Power	140
4.6.2 How to install Red Hat OpenShift Container Platform in IBM Cloud	143
Chapter 5. IBM Cloud Paks on Red Hat OpenShift running on IBM Power	149
5.1 Introduction	150
5.2 IBM Cloud Paks	150
5.3 IBM Cloud Paks offerings on IBM Power	152
5.3.1 IBM Cloud Pak for Data	152
5.3.2 IBM Cloud Pak for Business Automation	153

5.3.3 IBM Cloud Pak for Integration	154
5.3.4 IBM Cloud Pak for Watson AIOps.	155
5.3.5 IBM Cloud Pak for WebSphere Hybrid Edition	156
5.4 IBM Cloud Pak for Watson AIOps and IBM Cloud Pak for Data	157
5.4.1 IBM Cloud Pak for Watson AIOps.	158
5.4.2 IBM Cloud Pak for Data	158
5.5 IBM Db2 workloads on IBM Cloud Pak for Data on IBM Power	166
5.5.1 IBM Db2	167
5.5.2 IBM Db2 Warehouse.	168
5.5.3 IBM Db2 Data Management Console	170
5.5.4 Additional Db2 use cases	172
Chapter 6. Use cases	173
6.1 Artificial intelligence inferencing with Red Hat OpenShift and IBM Power10 processor-based servers	174
6.1.1 Matrix Math Accelerator	174
6.1.2 Optimized AI libraries	174
6.1.3 ONNX Runtime	174
6.1.4 Inferencing engine tutorial.	175
6.1.5 Model lifecycle	182
6.1.6 Summary.	182
6.2 Running Db2 workloads on IBM Cloud Pak for Data on IBM Power.	183
6.2.1 Lab environment	183
6.2.2 Installing IBM Cloud Pak for Data on Red Hat OpenShift.	184
6.3 GitOps for system configuration	190
Abbreviations and acronyms	203
Related publications	205
IBM Redbooks	205
Online resources	205
Help from IBM	206

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.

AIX®	IBM Elastic Storage®	PowerHA®
Cognos®	IBM Spectrum®	PowerVM®
DataStage®	IBM Spectrum Fusion™	Redbooks®
Db2®	IBM Watson®	Redbooks (logo)  ®
DS8000®	IBM Z®	Spectrum Fusion™
FileNet®	Instana®	SystemMirror®
IBM®	POWER®	Turbonomic®
IBM Automation®	Power Architecture®	WebSphere®
IBM Cloud®	POWER8®	z/OS®
IBM Cloud Pak®	POWER9™	z/VM®

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.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Preface

Enterprises everywhere are challenged to provide new services and environments based on hybrid cloud. The new paradigm of using hybrid cloud solutions to meet new business requirements can be challenging.

This IBM Redbooks® publication is designed to show you how to implement a hybrid cloud solution that uses the industry leading hybrid cloud platform (Red Hat OpenShift) on IBM Power based servers. By combining Red Hat OpenShift and IBM Power servers, you can create a highly reliable and scalable cloud environment. We provide hints and tips about how to install your Red Hat OpenShift cluster, and also provide guidance about how to size and tune your environment to meet your user's expectations.

This publication is suitable for a broad group of readers that are interested in understanding how IBM Power servers can be used in a Red Hat OpenShift environment and how they can take advantage of the benefits that are delivered by an IBM Power based cloud solution, that is, scalability, reliability, and security. This publication provides implementation details and case studies that make this publication especially helpful to system admins and cloud services implementors.

Authors

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

Dino Quintero is a Systems Technology Architect with IBM Redbooks. He has 28 years of experience with IBM Power technologies and solutions. Dino shares his technical computing passion and expertise by leading teams developing technical content in the areas of enterprise continuous availability, enterprise systems management, high-performance computing (HPC), cloud computing, artificial intelligence (AI) (including machine and deep learning), and cognitive solutions. He is a Certified Open Group Distinguished Technical Specialist. Dino is formerly from the province of Chiriqui in Panama. Dino holds a Master of Computing Information Systems degree and a Bachelor of Science degree in Computer Science from Marist College.

Tim Simon is an IBM Redbooks Project Leader in Tulsa, Oklahoma, US. He has over 40 years of experience with IBM®, primarily in a technical sales role working with customers to help them create IBM solutions to solve their business problems. He holds a BS degree in Math from Towson University in Maryland. He has worked with many IBM products, and has extensive experience creating customer solutions by using IBM Power, IBM Storage, and IBM zSystems throughout his career.

Tushar Agrawal is a Customer Success Leader in the United States. He has 20 years of experience in enterprise architecture and solution design in e-commerce and supply chain for large-scale data processing and high availability (HA) by using microservices and event-driven architecture. Tushar successfully launched 50+ complex solutions for customers across industry verticals and segments. Tushar leads a team of architects to drive the adoption of IBM Hybrid Cloud by using Red Hat OpenShift and develop reusable assets to reduce time to market. Tushar has filed 100+ patent applications with the US Patent Office, and continuously works on innovation with emerging technologies to co-create intellectual property for IBM. Tushar is originally from India. Tushar holds an MBA degree from North Dakota State University, Fargo, ND, and a BS degree in Computer Science & Engineering from MNNIT, Allahabad (India).

Sambasiva Andaluri (Sam) is an experienced developer turned Solution Architect Leader with over 30 years of experience. For the past decade, he has been a pre- and post-sales solution architect for trading systems at Fidessa, a pre-sales solution architect at AWS, a Site Reliability Engineer (SRE) that onboarded independent software vendors (ISVs) for the Google marketplace at a business partner. He brings multifaceted experience to the table and is a continuous learner.

Shahid Ali is a Cloud Solution Lead for the MEA Region. At the time of writing, he is based in Riyadh, Saudi Arabia, and leads hybrid multi-cloud solutions in the MEA region. Shahid is an experienced Enterprise Architect who joined IBM 5 years ago as an Enterprise Architect. He has 28 years of experience as an architect and consultant. Before joining IBM, he provided consultancy services in some of the largest projects in Saudi Arabia for the Ministries of Interior, Education, and Labor, and related organizations. These projects produced nationwide solutions for fingerprinting, country-wide secure networks, smart ID cards, e-services portals, enterprise resource planning systems, and massive, open online course platforms. Shahid has several IBM and industry certifications, and is a member of the IBM Academy of Technology.

Daniel Casali is a Thought Leader Information Technology Specialist that has worked for 15 years at IBM with IBM Power, HPC, big data, and storage. His role at IBM is producing solutions that address client's needs by exploring new technologies and for different workloads. He works with real multicloud implementations to abstract and simplify the new challenges of the heterogeneous architectures that are intrinsic to this new consumption model, whether it is on-premises or in the public cloud.

Munshi Hafizul Haque is a Senior Platform Consultant at Red Hat in Kuala Lumpur, Malaysia. Munshi is an experienced technologist in the engineering, design, and architecture of platform as a service (PaaS) and cloud infrastructures. At the time of writing, he is part of the Red Hat Consulting Services team, where he helps organizations adopt automation, container technology and DevOps practices. Before that, he worked for IBM as a senior consultant with IBM Systems Lab Services in Petaling Jaya, Malaysia, where he took part in various projects with different people in different ASEAN countries, and as a specialist in IBM Power and associated enterprise edition technology.

Diogo Horta is a Technical Specialist Thought Leader, Entrepreneur, Solution Architect, and Certified Data Engineering Expert. He has 20+ years of experience in the Data and AI field with multi-industry knowledge and vast experience in sales. At the time of writing, he works as a Senior Customer Success Manager Architect within the IBM Americas Customer Success Manager team in Brazil. His areas of expertise include computer engineering, computer science, data engineering, data science, AI, machine learning, data governance, data quality, and cloud. He has worked extensively on diverse and complex projects by developing and implementing solutions in the leading enterprises in the banking, telecommunications, insurance, and government industries.

Shrirang Kulkarni is a LinuxONE and Cloud Architect who has been with IBM over 17 years working with IBM System Labs as a LinuxONE and Cloud Architect supporting IBM Z® Global System Integrators. He has worked with various clients in over 25 countries worldwide from IBM Dubai as a Lab services consultant for IBM zSystems in the MEA region. He has achieved “IBM Expert Level IT specialist” and “The Open Group Certified Master IT Specialist” certifications. He coauthored *Security for Linux on System z*, SG24-7728, and also authored “Bringing Security to Container Environments, Performance Toolkit and Streamline Fintech Data Management With IBM Hyper Protect Services” which was published in IBM System Magazine. His areas of expertise include Linux on IBM zSystems, IBM z/VM®, cloud solutions, IBM z/OS® Container Extensions (zCX), Red Hat OpenShift, architecture design and solutions for z/VM and Linux on zSystems, performance tuning Linux on zSystems, IBM z/VM, Oracle, IBM Power, and IBM System x.

Nick Lawrence is an IT Management Consultant on the IBM Technology Lifecycle Services team (IBM Power). Nick specializes in emerging technologies, cloud, and AI applications. Before joining Technology Services in the spring of 2022, Nick was a software developer who was responsible for building healthcare solutions that are powered by IBM Watson® technologies.

Laszlo Niesz is a Software Specialist in Hungary. He has 25 years of experience in software support, systems management, and implementation fields at IBM. He holds a degree in Computer Science from University of Szeged, Hungary. His areas of expertise include machine learning with Python, IBM PowerVM®, IBM Spectrum® Scale, Red Hat OpenShift, infrastructure as code (IaC), and GitOps. He has written extensively on performance monitoring, software-defined infrastructure (SDN), and the Red Hat OpenShift ecosystem.

Gabriel Padilla is the Linux Test Architect for IBM Power who focuses on hardware assurance. He has a bachelor’s degree as an Electronic Engineer and a master’s degree in Information Technology. He has been at IBM 10+ years, and his experience ranges from Test Development (Design) to Supply Chain process. Gabriel is considered a technical leader for IBM Systems, including Linux, Red Hat OpenShift. and cloud.

Gustavo Santos is an IBM Brand Technical Specialist and IBM Power Consultant. He has been with IBM since 1997. He has 25 years of experience in IBM Power, cognitive solutions, and hybrid cloud architecture. He holds a degree in Systems Engineering from Universidad Abierta Interamericana. During the last 7 years, he worked as an IBM Power Consultant, and during the last year, he was working as a Brand Technical Specialist to create solutions for clients and add value to the IBM Solutions portfolio. This residency is his eighth IBM Redbooks residency. He writes extensively on the IBM Power infrastructure.

Shiv Tiwari is a seasoned Information Technology professional with 18 years of technical experience. As a Data and AI Technical Sales Specialist at IBM India South Asia, he leverages his expertise to help clients harness the power of data and AI to uncover valuable insights. Working closely with clients, he provides expert guidance on developing effective data architecture strategies and has a record of success working with leading enterprises across various industries, including banking, telecommunications, insurance, and retail. He co-authored *IBM AIX and Enterprise Cloud Solutions*, REDP-5660. His areas of expertise include data engineering, data science, AI and machine learning, data governance, data quality, and data observability.

Sundaragopal Venkatraman (Sundar) is a CTO at the Center of Excellence, IBM Expert Labs. He has diversified skills in IBM Cloud Pak®, migration, and modernization. Sundar has over 23 years of experience working closely with customers to overcome business challenges by leveraging technologies. A prolific author, he has been recognized as “Gold Author” for IBM Redbooks publications. He has various filed patents and is an Invention Plateau holder. He delivered key notes on WW conferences on Technology Transformation & Modernization. He is a co-chair for the IT specialist board in the Asia-Pacific region.

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

Sukumar Subburaj
Senior Consultant
CoE IBM Expert Labs, India

Cesar Araujo
STSM - Architect - IBM Automation® (IBM Cloud Pak for Multicloud Management, IBM Cloud Pak for Watson AIOps, IBM Monitoring, and Application and Performance Monitoring (APM) IBM Instana®)

Attila Grósz
Storage Systems Technical Sales
IBM Sales, Hungary

Ashwin Srinivas
Senior AI and Hybrid Cloud Technical Architect
IBM India

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 books to be as helpful as possible. Send us your comments about this book 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>



Introduction

This chapter introduces the concepts and considerations to implement your workloads in a cloud environment by using Red Hat OpenShift on IBM Power servers, either on-premises or in the cloud.

This chapter contains the following topics:

- ▶ Adapting to a new infrastructure paradigm
- ▶ Red Hat OpenShift on IBM Power
- ▶ IBM Power servers
- ▶ Summary

1.1 Adapting to a new infrastructure paradigm

It seems to be a business imperative for enterprises to embrace cloud resources and infrastructure. To compete in the modern world, an enterprise must adapt. The journey to cloud is undertaken because businesses see many benefits from cloud computing infrastructures and platforms, such as the ones that are provided by Red Hat OpenShift.

1.1.1 Cloud benefits

Moving to a cloud infrastructure, whether it is a private cloud, public cloud, or a mixture of multiple clouds in a hybrid cloud environment, is done because business users are demanding a more adaptive IT infrastructure that can support the new business requirements of agile programming and flexible infrastructure. Then, business quickly can leverage new business opportunities that better use the enterprise's assets. Enterprises look to cloud for the following benefits:

- ▶ Flexibility
- ▶ Efficiency
- ▶ Strategic value

Flexibility

Users can scale services to fit their needs, customize applications, and access cloud services from anywhere by using an internet connection. The benefits come from the following items:

- Scalability:** The cloud infrastructure scales on demand to support fluctuating workloads.
- Storage options:** Users can choose public, private, or hybrid storage offerings, depending on their security needs and other considerations.
- Control choices:** Organizations can determine their level of control with “as a service” options, which include software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS).
- Tool selection:** Users can select from a menu of prebuilt tools and features to build a solution that fits their specific needs.
- Security features:** Virtual private cloud (VPC), encryption, and application programming interface (API) keys help keep data secure.

Efficiency

Enterprise users can get applications to market quickly without worrying about underlying infrastructure costs or maintenance. These benefits come from the following items:

- Accessibility:** Cloud-based applications and data are accessible from many internet-connected devices.
- Speed to market:** By developing in the cloud, users get their applications to market quickly.
- Data security:** Hardware failures do not result in data loss because of networked backups.
- Equipment savings:** Cloud computing uses remote resources, which save organizations the cost of servers and other equipment.
- Pay structure:** A “utility” pay structure means users pay only for the resources that they use.

Strategic value

Cloud services give enterprises a competitive advantage by providing the most innovative technology that is available. These benefits come from the following items:

- Streamlined work:** Cloud service providers (CSPs) manage the underlying infrastructure so that organizations can focus on application development and other priorities.
- Regular updates:** Service providers regularly update offerings to give users the most up-to-date technology.
- Collaboration:** Worldwide access means that teams can collaborate from widespread locations.
- Competitive edge:** Organizations can move more nimbly than competitors who must devote IT resources to managing infrastructure.

The benefits of cloud can be significant, and when properly used they can provide a business advantage in today's competitive world. However, moving to a cloud environment requires a new way of thinking in terms of how to get the most out of your infrastructure in the new cloud world.

1.1.2 New performance paradigm

Enterprises have grown adept at managing and monitoring their application performance in their traditional IT environment. They have developed tools and techniques that help them meet the requirements of their users, and they understand how their systems interact.

When moving to a cloud infrastructure (private, public, or hybrid), the “tried and true” processes and techniques are not going to be as effective. In a traditional IT infrastructure, the enterprise controls all aspects of where their applications run, and the environment is relatively static. In comparison, components in the cloud are designed to start, scale, stop, and move quickly based on the current workloads. This approach creates challenges, so new tools and techniques must be developed to monitor and manage your cloud environment to provide the appropriate user experience.

It is our intention to help you adapt to this new performance paradigm by showing you some tools and techniques that help you plan for, implement, and manage a cloud environment running on your IBM Power servers.

1.2 Red Hat OpenShift on IBM Power

There are many options that are available for running your cloud workloads, both in the world of private clouds and in public clouds. This book describes the usage of Red Hat OpenShift on IBM Power servers either in your enterprise or in a public cloud environment, such as the IBM Power Systems Virtual Server (IBM PowerVS) offering, which is an IaaS offering running on IBM Power servers in IBM data centers.

This section provides an overview of Red Hat OpenShift running on IBM Power.

1.2.1 Red Hat OpenShift

Red Hat OpenShift is an open-source container application platform that runs on Red Hat Enterprise Linux CoreOS (RHCOS), and it is built on Kubernetes. It includes integrated scaling, monitoring, logging, and metering functions. Red Hat OpenShift includes everything that you need for hybrid cloud, like a container runtime, networking, monitoring, container registry, authentication, and authorization.

“Red Hat OpenShift architecture and components” provides a high-level overview of Red Hat OpenShift. This overview is an introduction. For more information about Red Hat OpenShift, see Chapter 4, “Red Hat OpenShift architecture and design” on page 95.

Red Hat OpenShift architecture and components

To best use Red Hat OpenShift, you must understand its architecture. Figure 1-1 provides an overview of Red Hat OpenShift.

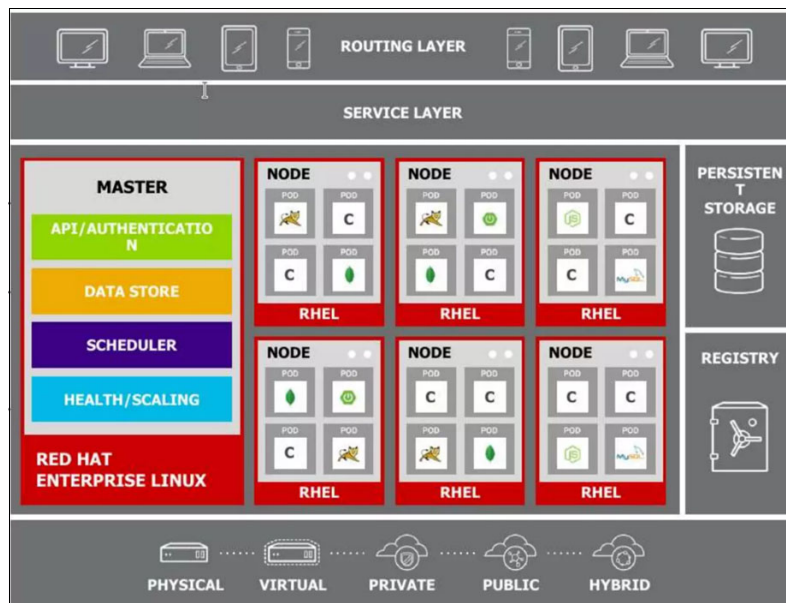


Figure 1-1 Red Hat OpenShift architecture

Red Hat OpenShift consists of the following layers and components, and each component has its own responsibilities:

- ▶ Infrastructure layer
- ▶ Service layer
- ▶ Control nodes
- ▶ Worker nodes
- ▶ Containers
- ▶ Registry
- ▶ Persistent storage
- ▶ Routing layer

Infrastructure layer

In the infrastructure layer, you can host your applications on physical servers, virtual servers, or even on the cloud (private or public). The physical server or virtual server is called a *node* in Red Hat OpenShift. The node is the smallest unit of computer hardware that can be defined. Nodes store and process data.

This book focuses on using IBM Power servers with built in virtualization functions that can run your traditional workloads and your new cloud workloads on the same physical servers in different logical partitions (LPARs).

You can use your existing IBM Power infrastructure to integrate efficiently your new cloud applications with your traditional applications, and continue to get business value from the current applications while quickly building new workflows to leverage new business opportunities.

Service layer

The service layer is responsible for defining pods and access policy. A *pod* is the smallest unit that can be defined, deployed, and managed, and it can contain one or more containers. Containers are objects that run applications or tasks. For more information, see “Containers” on page 5.

The service layer provides a permanent IP address and hostname to the pods; connects applications together; and allows simple internal load-balancing by distributing tasks across application components.

There are two types of nodes in an Red Hat OpenShift cluster: control nodes and worker nodes. Control nodes are involved in managing the cluster, and worker nodes run applications. As a best practice, use multiple control nodes for availability. You can have multiple worker nodes in the cluster. Adding worker nodes scales the environment or provides isolation for sensitive workloads. The worker nodes are where all your coding occurs.

Control nodes

The *control nodes* manage the cluster as part of the control plane, and they manage the worker nodes. They are responsible for four main tasks:

- ▶ API and authentication: Any administration request goes through the API. These requests are SSL-encrypted and authenticated to ensure the security of the cluster.
- ▶ Data store: Stores the state and information that are related to the environment and applications.
- ▶ Scheduler: Determines pod placements while considering current memory, CPU, and other environment utilization.
- ▶ Health and scaling: Monitors the health of pods and scales them based on CPU utilization. If a pod fails, the main node restarts it automatically. If it fails too often, it is marked as a bad pod, and temporarily is not restarted.

Worker nodes

A *worker node* is a node that runs the application in a cluster and reports to a control plane. The main responsibilities of a worker node are to process data that is stored in the cluster and handle the networking to ensure that traffic between the application parts, both across the cluster and outside of the cluster, is properly facilitated.

Containers

A *container* is a lightweight package of your application code together with dependencies such as specific versions of programming language run times and the libraries that are required to run your software services.

Containers are ephemeral, so saving data in a container risks the loss of data. To prevent loss, use persistent storage to save data for applications and databases.

All containers in one pod share an IP address and volume. In the same pod, you can have a sidecar container, which can be a service mesh or used for security analysis. The sidecar container must be defined in the same pod, and it shares the resources of all other containers. Applications can be scaled horizontally or vertically by adding more containers and pods, and they are wired together by services.

Registry

The *registry* saves your images locally in the cluster. When a new image is pushed to the registry, it notifies Red Hat OpenShift and passes image information.

Persistent storage

Persistent storage is where all your data is saved and connected to containers. You must have persistent storage because containers are ephemeral, which means that when they are restarted or deleted, any saved data is lost. Therefore, persistent storage prevents any loss of data and allows the usage of stateful applications.

Routing layer

The *routing layer* provides external access to the applications in the cluster from any device. It also provides load-balancing and auto-routing around unhealthy pods.

1.3 IBM Power servers

IBM Power servers have a reputation for reliability, security, and longevity. Some of the largest companies in the world run their business on IBM Power server, which includes 80 of the Fortune 100. They trust IBM Power servers to run their business in a secure environment with minimal unplanned downtime so that they can implement the best hybrid cloud strategy to manage effectively large amounts of data and better serve their customers.

IBM Power servers are designed for security and for performance. They have one of the smallest number of known security issues in the industry. They are built with high performance with industry-leading connectivity and scalability to handle many concurrent users and work with large data sources effectively.

IBM Power servers provide a flexible platform with cloud-like scalability and pricing. IBM Power servers are available as a hybrid cloud offering (IBM PowerVS). An advantage of choosing IBM Power servers for your cloud infrastructure is the ease of migration of your current workload and data into your new cloud environment without having to replatform them.

IBM Power servers can be the right solution for your cloud requirements. This book is designed to help you design, implement, and tune those applications that run in your IBM Power server cloud.

For more information about the advantages of using IBM Power servers in your hybrid cloud, see Chapter 3, “IBM Power processor performance capabilities” on page 43.

1.4 Summary

There are many choices that you can make as you move forward in your journey to cloud. As you make those choices, you must consider what workloads you are moving to the cloud and consider the best platform for each workload. The platform that you choose must meet the performance requirements of your users; provide flexibility; are manageable by your IT staff; and provide the security that is required to keep your data and your customer's data safe.

We recommend that you choose an IBM Power solution running Red Hat OpenShift.



Performance and tuning

This chapter describes some techniques to maximize the performance of your applications in a Red Hat OpenShift cluster. We explore how to define performance, and describe some of the tools that can be used to measure the different aspects of performance in your cluster. We provide some specific best practices to help you plan, set up, and tune your Red Hat OpenShift environment to provide the performance that your users are expecting.

This chapter contains the following topics:

- ▶ Definitions
- ▶ Models
- ▶ An example use case scenario
- ▶ Red Hat OpenShift performance baseline
- ▶ Red Hat OpenShift starting configuration
- ▶ Tools

2.1 Definitions

In this section, we define important terms and concepts about monitoring and managing Red Hat OpenShift performance. We describe the steps for doing performance tuning and the components that are involved.

2.1.1 Performance components

Red Hat OpenShift Container Platform is an automated Kubernetes container platform that you can use to deploy and manage cloud applications. Here are some of the components to consider when designing your cluster for performance:

- ▶ Optimized and lightweight images
- ▶ Dependency
- ▶ Memory utilization
- ▶ Disk utilization
- ▶ CPU utilization
- ▶ Performance monitoring plan
- ▶ Results

Optimized and lightweight images

In the Bundle Format, a *bundle image* is a container image that is built from Operator manifests that contains one bundle. Bundle images are stored and distributed by Open Container Initiative (OCI)-defined container registries, such as Quay.io or DockerHub. Because an optimized or lightweight image has a lower impact, using these containers means that you can fit more workloads within your infrastructure.

Dependency

An Operator might have a dependency on another Operator that is in the cluster. For example, the Vault Operator has a dependency on the etcd Operator for its data persistence layer.

Red Hat Operator Lifecycle Manager (OLM) resolves dependencies by ensuring that all specified versions of Operators and Custom Resource Definitions (CRDs) are installed on the cluster during the installation phase. This dependency is resolved by finding and installing an Operator in a catalog that satisfies the required CRD application programming interface (API) and is not related to packages or bundles.

Memory utilization

Memory is an important component that the system needs to do work. If there is not enough memory that is available, the system swaps out some sections of memory to load new content. This swap causes delays in starting new tasks and might create more latency for tasks whose memory was swapped out as they wait to be swapped back in. Memory utilization can be monitored at both the pod and the node level:

- ▶ Monitoring the pod level can help identify pods that exceed memory utilization limits and terminate them.
- ▶ Monitoring the node level can help identify nodes running low on available memory. In this case, the kubelet flags the node as under memory pressure and starts reclaiming resources.

Disk utilization

The amount of space that is available on the disk can have implications on system performance. Low available disk space on the root volume can lead to issues with scheduling pods. When the node's remaining disk capacity exceeds a certain threshold, it is flagged as under disk pressure.

CPU utilization

CPU utilization is an important metric of the performance of your Red Hat OpenShift cluster. High CPU utilization can cause extended latency to your user because the infrastructure cannot perform tasks in a timely manner. Monitoring CPU utilization by using Grafana or any other monitoring tool can help identify whether CPU utilization is related to health issues in the cluster.

Performance monitoring plan

The performance monitoring plan is a detailed document that describes your indicators, measures, and approach to data collection, acquisition, analysis, use, and reporting.

Results

The *results* are changes that happen because of what a project or program does. They include outcomes and outputs.

2.1.2 Performance tuning terminology

Table 2-1 provides a list of terms that are used when describing performance tuning. Understanding these terms can help you plan for and manage the performance of your Red Hat OpenShift cluster.

Table 2-1 Performance tuning terminology

Term	Definition
Concurrent users	The number of application users actively using and accessing the container application, or an element such as a process at a particular time.
Latency	Delay that is experienced in network transmissions as network packets traverse the network infrastructure.
Think time	The wait time between user operations. For example, a user brings up the Account screen and spends 10 seconds reviewing the data for an account. These 10 seconds are the think time for this operation. Think time is a critical element in performance and scalability tuning, particularly for a process. When think time values are correctly forecasted, then actual load levels are close to anticipated loads.
Multithreaded process (or MT server)	A process running on a multithreaded container component that supports multiple threads (tasks) per process. Tasks and components run multithreaded processes that support threads.
Task	A concept for container applications of a unit of work that can be done by a container component. Container tasks are typically implemented as threads.

Term	Definition
Response time	The amount of time that the container takes to complete an operation. The time is an aggregate of the time that is incurred by all server processing and transmission latency for an operation. The response time might be experienced by an application user or might be the amount of time that is needed for some other operation that is unrelated or indirectly related to user sessions.
Throughput	Typically expressed in transactions per second (TPS), it expresses how many operations or transactions can be processed in a set amount of time.
Thread	An operating system feature for performing a unit of work. Threads are used to implement tasks for most containers. A multithreaded process supports running multiple threads to perform work, such as to support user sessions.

2.1.3 Service-level agreement, service-level objective, and service-level indicator

When planning for performance and the availability of a system, you must have objectives that are agreed on before implementation that describe the expectations users have for the availability of the system. Service-level agreements (SLAs), service-level indicators (SLIs), and service-level objectives (SLOs) represent different concepts describing the promises that you make to the users about the availability of your system to ensure that the users' expectations are met. For example, these components address:

- ▶ How often will the system be available?
- ▶ How quickly will you respond when the system is down?
- ▶ What is the expected performance?

Maintaining these expectations and promises is an important part of maintaining your user's satisfaction and confidence in your applications.

Service-level agreement

An SLA is an agreement between a provider and their clients about measurable metrics like uptime, responsiveness, and responsibilities. Generally, SLAs are agreements between a vendor and paying customers. SLAs are legal documents and represent both the expectations and the consequences of failure in meeting those expectations. Consequences might include financial penalties or service credits, for example.

Service-level objective

An SLO is a statement about a specific metric within an SLA. This metric might be related to uptime, response time, or other measurable metrics that are important to the user. Where SLAs are relevant only to paying customers, SLOs can be useful to both paying and nonpaying users, and both internal and external users. They also provide your development and IT staff targets for the goals against which they set and measure themselves.

Service-level indicator

The SLI is the specific metric that shows compliance (or noncompliance) for the SLOs that are set up. The metrics must be specific and measurable indicators. To meet an SLO, the SLI for that component must be equal to or higher than the SLO target.

2.2 Models

Applications running in Red Hat OpenShift or any Kubernetes cluster are distributed systems. When analyzing performance and tuning the workloads in these clusters, you must understand a few theoretical models. This section describes some of these models and provides some examples. These model concepts were used by the authors of this book in the field to troubleshoot performance issues in large-scale, distributed systems.

This section describes some of the models and concepts that are involved in performance measurement and management.

2.2.1 Queuing theory

The Danish mathematician and engineer Agner Erlang is credited for discovering queuing theory in 1920. This theory has applications in several fields, such as designing call centers, optimizing restaurant operations, and understanding bottlenecks in a distributed system. Whether you are running applications on-premises or in a cloud, your systems, interconnects, and applications are increasingly distributed. With the advent of the microservices paradigm, where you might find an average of 100 microservices or more, this complexity increases.

In the context of performance testing and tuning, large-scale distributed systems are difficult to test for assessing capacity because the testing requires a production copy of the infrastructure and a simulated load. By understanding queuing theory, you can build a mathematical model of the system and use commonly available tools to find an optimal architecture without huge cost overlay. For more information, see [KubeCon 2017](#).

2.2.2 Little's Law

John Little, an operation research professor at MIT, discovered that the average number of tasks in a queue can be obtained by a product of the arrival rate of tasks and their holding or processing time. Little's Law is an intuitive way to derive the relationship between latency and throughput. For more information about how to use Little's law to calculate sizing for a web server, see [IBM WebSphere® Performance Cookbook](#).

2.2.3 The four Golden Signals

Google distilled their years of experience of running millions of commodity servers into the Site Reliability Engineering (SRE) practices. The Google methodology proposes monitoring the four Golden Signals: Latency, Traffic, Errors, and Saturation.

Latency	How long a request took to process.
Traffic	How much traffic a service is receiving, which might be described by the volume of data that is processed or the network or disk I/O rates. Traffic includes both the application and the subsystem level.
Errors	How many requests resulted in errors, which include both the errors a service is returning and errors that occur when a request results in an exception that causes application failures.
Saturation	How busy the servers are in terms of compute, storage, and networking. Are the applications starving for resources, meaning that some resource utilization is nearing 100%? For some resources, even a lower utilization can be an issue if they cause excessive queuing of requests.

Google wrote [several books and workbooks](#) on their SRE practices. These resources are available at no additional charge to help others implement their best practices.

2.2.4 USE method

Brendan Gregg, a Solaris engineer, is known as a performance expert for his work in tools, such as Dtrace for Solaris and the newer eBPF in Linux. He pioneered two performance models, one of which is the resource-focused *Utilization, Saturation, and Errors* (USE) method. This model can be applied in the context of performance tuning. Resources such as CPU, memory, disk, and network are finite and limited, so it is critical to determine their utilization, saturation, and errors.

Utilization	Determines how busy a resource is. For example, a CPU might be 100% utilized or disk I/O might be 100%, so the system cannot service new requests.
Saturation	Denotes how many processes are queued to access the resources. Queuing theory and Little's Law both are applied here to understand how these queues affect the performance of a system.
Errors	Deals with network retransmissions or other resource-level issues where the presence of an issue might have a domino effect on applications that rely on a specific resource.

For more information, see the [USE method](#).

2.2.5 RED method

For the microservices architecture, Weave Works, the people behind the Flux GitOps, coined the term *Rate, Errors, and Duration* (RED) for measuring microservices performance:

Rate	The number of requests served per second (or a duration) or throughput.
Errors	Number of failed requests.
Duration	Denotes the duration or latency of each request.

The RED method is based on the 4 Golden Signals, but it provides a different abstraction to help measure and troubleshoot performance issues.

2.3 An example use case scenario

This section described a specific customer situation where an Red Hat OpenShift environment was deployed. It describes the environment as it was set up and then lists the tuning actions that were taken to improve the performance of the environment to meet the customer's expectations for the applications that are deployed.

Products deployed

- ▶ IBM Cloud Pak For Integration (IBM CP4I) 2021.4 - App Connect and API Connect
- ▶ Red Hat OpenShift 4.8.23
- ▶ Red Hat OpenShift Data Foundation (previously named Red Hat OpenShift Container Storage) 4.8

Compute resources

Table 2-2 shows the compute resources and storage that are assigned to the different components that are used in the use case.

Table 2-2 Server details with compute resources and storage

Nodes	Type	Server	Server IP address	Hostname	CPUs	Memory	Disk space
Boot	Virtual machine (VM) (temp)	Server B	10.198.34.16	boot.ocpuat. domainname.com	4	16	200
Master	VM	Server M1	10.198.34.17	master0.ocpuat. domainname.com	8	32	300
	VM	Server M2	10.198.34.18	master1.ocpuat. domainname.com	8	32	300
	VM	Server M3	10.198.34.19	master2.ocpuat. domainname.com	8	32	300
Infrastructure	VM	Server IF1	10.198.34.20	infra0.ocpuat. domainname.com	16	32	300 + additional disk 1 TiB (RAW - unformatted) (OCS)
	VM	Server IF2	10.198.34.21	infra1.ocpuat. domainname.com	16	32	300 + additional disk 1 TiB (RAW - unformatted) (OCS)
	VM	Server IF3	10.198.34.22	infra2.ocpuat. domainname.com	16	32	300 + additional disk 1 TiB (RAW - unformatted) (OCS)
HAProxy	VM	Server P1	10.198.34.27	haproxy.ocpuat. domainname.com	4	16	200
Worker	VM	Server W1	10.198.34.23	worker0.ocpuat. domainname.com	16	32	300
	VM	Server W2	10.198.34.24	worker1.ocpuat. domainname.com	16	32	300
	VM	Server W3	10.198.34.25	worker2.ocpuat. domainname.com	32	32	300
	VM	Server W4	10.198.34.29	worker3.ocpuat. domainname.com	32	32	300
	VM	Server W5	10.198.34.16	worker4.ocpuat. domainname.com	32	32	300
Bastion	VM	Server BT	10.198.34.26	bastion.ocpuat. domainname.com	16	32	200

Non-functional requirements

Conc. Users	3800
Peak utilization	48%
Total Users	1.25 million
Response time	1.5 s
Load generator used	JMeter

Tuning activities

We made the following tuning changes in the environment:

- ▶ We evaluated the compute resources and corrected the sizing requirements.
- ▶ We evaluated the application pod (integration server) and changed the max and min CPU values.
- ▶ We changed the failover by using replicas for the application pods.
- ▶ We used Grafana to monitor the complete metrics in and out of the Red Hat OpenShift system.
- ▶ We monitored the subsystem by using an external tool to identify the utilization.
- ▶ We made the pod crash scenario and changed the values of threshold.
- ▶ We resized the utilization cap to 50%.
- ▶ We observed the worker plane utilization was less than 40% according to customer requirements.
- ▶ We also changed a few application product parameters (primarily Liberty and nginx).

The test case was run on a user acceptance testing (UAT) environment, where we monitored the following parameters:

- ▶ Log I/O
- ▶ Log size
- ▶ CPU utilization on each node
- ▶ Pod utilization
- ▶ Container logs review
- ▶ Grafana to monitor the cluster health

An initial result of our tuning efforts was that the memory utilization of both the apiserver (master) and etcd processes were reduced.

Recommendations

After gathering additional data (“Tuning activities”), we made the following recommendations:

- ▶ In Figure 2-1 on page 17, you see the sizes of the I/O operations of a 3-node cluster of etcd 3.1 (using storage V3 mode and with quorum reads enforced). Because there are many small-sized writes, etcd should be run on a system with SSD storage to optimize performance.

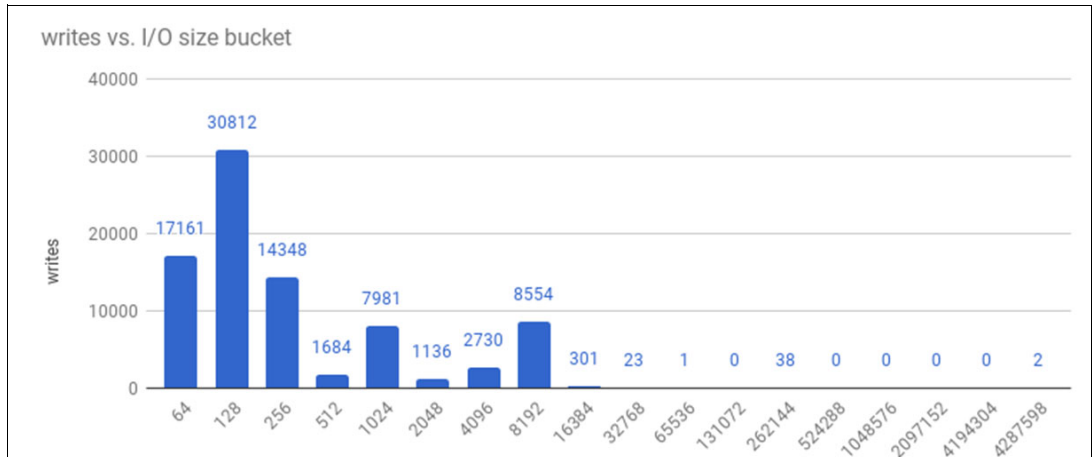


Figure 2-1 The etcd write sizes

- ▶ Because etcd processes are memory-intensive and master and apiserver processes are CPU-intensive, they are a reasonable fit for colocation on a single machine or VM.
- ▶ Optimize the communication between etcd and the master hosts either by collocating them on the same host or providing a dedicated network.

Tip: After profiling etcd under Red Hat OpenShift Container Platform, as shown in Figure 2-1, you see that etcd frequently performs small reads/writes. Using etcd with storage that handles small read/write operations quickly, such as SSD, is a best practice.

Additional tuning actions

To further optimize the environment, we also took the following actions:

- ▶ Changed the parameters of Liberty for heap.
- ▶ Changed the application-logging level that is required.
- ▶ Changed the access controls.
- ▶ Changed the CPU size for the pods.
- ▶ Changed the replica count.
- ▶ Started with 50 users, and then the test continued to grow with 500, 1000, 2000, 3800, and 4000 users.
- ▶ Changed the ingress replicas to 3.
- ▶ Moved the logging and Grafana pods to infra nodes.
- ▶ Moved ingress pods to infra nodes.
- ▶ Moved Prometheus to infra nodes.
- ▶ Moved registry pods to infra nodes.

Results

As a result of the tuning activities, we achieved the response time that was required by the customer. The response time includes the time that the user experienced starting from the user's computer and includes moving through the HAProxy and routers.

2.4 Red Hat OpenShift performance baseline

This section describes Red Hat OpenShift Container Platform performance. We provide information about baseline considerations and estimates about expected performance.

Red Hat OpenShift Container Platform has a microservices-based architecture of smaller, decoupled units that work together. It runs on a Kubernetes cluster, with data about the objects stored in etcd, which is a reliable clustered key-value store.

A node provides the runtime environments for containers. Each node in a Kubernetes cluster runs the required services, which are managed by the master. Nodes also provide the required services to run pods and services, a kubelet, and a service proxy.

2.4.1 Red Hat OpenShift Container Platform: baseline best practices

This section provides general best practices for successfully running an Red Hat OpenShift Container Platform.

General baseline best practices

As you consider running your Red Hat OpenShift Container Platform environment, consider the following general guidelines:

- ▶ Do the proper capacity planning. Identify and document your requirements.
- ▶ Install by using the installation recommendations in the product documentation.
- ▶ Identify the product version that fits your need and is compatible with your application deployment.
- ▶ Plan for Day 2 operations properly.
- ▶ Do not change any CoreOS values without recommended instructions from product service (either Red Hat or IBM).
- ▶ Consider externalizing logs by using external tools.
- ▶ Ensure that the proper user governance is set up.
- ▶ Plan for your continuous integration and continuous delivery (CI/CD) deployment pipeline and consider using fully automated environments.

2.4.2 Cluster performance considerations

Designing and sizing your cluster for the initial deployment and additional growth is important.

When installing large clusters or scaling the cluster to larger node counts, set the cluster network `cidr` in your `install-config.yaml` file before you install the cluster.

Important: The default cluster network-address setting of 10.128.0.0/14 cannot be used if the cluster size is more than 500 nodes. It must be set to 10.128.0.0/12 or 10.128.0.0/10 to get to node counts beyond 500 nodes.

2.4.3 Node host best practices

The Red Hat OpenShift Container Platform node configuration file contains important options. For example, two parameters control the maximum number of pods that can be scheduled to a node: `PodsPerCore` and `maxPods`.

When both options are specified, the *lower* of the two values limits the number of pods on a node. Exceeding these values can result in the following conditions:

- ▶ Increased CPU utilization by Red Hat OpenShift Container Platform
- ▶ Slow pod scheduling
- ▶ Potential out-of-memory scenarios, depending on the amount of memory in the node
- ▶ Exhausting the IP address pool
- ▶ Resource over committing, leading to poor user application performance

Creating a KubeletConfig CRD to edit kubelet parameters

The kubelet configuration is serialized as an Ignition configuration, so it can be directly edited. However, there is also a new `kubelet-config-controller` that was added to the Machine Config Controller (MCC). With this controller, you can use a `KubeletConfig` custom resource (CR) to edit the kubelet parameters.

Modifying the number of unavailable worker nodes

You can change the type and size of worker nodes. By default, only one machine may be unavailable when applying the kubelet-related configuration to the available worker nodes. For a large cluster, it can take a long time for the configuration change to be reflected. At any time, you can adjust the number of machines that are updating to speed up the process.

2.4.4 Control plane node sizing

The control plane node resource requirements depend on the number of nodes in the cluster. The control plane node size recommendations that are shown in Table 2-3 are based on the results of control plane density focused testing.

Important: The control plane size cannot be changed when the cluster is provisioned and running. Ensure that you use the suggested control plane size during installation.

Table 2-3 Control plane sizing recommendations¹

Number of worker nodes	Cluster-density (namespaces)	CPU cores	Memory (GB)
27	500	4	15
120	1000	8	32
252	4000	16	64
501	4000	16	96

On a large and dense cluster with three masters or control plane nodes, the CPU and memory utilization spikes when one of the nodes is stopped or restarted, or fails. Failures can be due to unexpected issues with power, networks, or underlying infrastructure in addition to intentional cases where the cluster is restarted after shutting it down to save costs. The remaining two control plane nodes must handle the load to ensure high availability (HA), which increases resource utilization. This situation is expected during upgrades because the masters are cordoned, drained, and restarted serially to apply the operating system updates and the control plane Operators update.

¹ https://docs.openshift.com/container-platform/4.11/scalability_and_performance/recommended-host-practices.html

Important: To avoid cascading failures, keep the overall CPU and memory resource utilization on the control plane nodes to at most 60% of all available capacity to handle the resource utilization spikes. Increase the CPU and memory on the control plane nodes to avoid potential downtime due to lack of resources.

2.4.5 etcd best practices

For large volume clusters, etcd can suffer from poor performance if the keyspace grows too large and exceeds the space quota. Periodically maintain and defragment etcd to free space in the data store. Monitor Prometheus for etcd metrics and defragment it when required; otherwise, etcd can raise a cluster-wide alarm that puts the cluster into a maintenance mode that accepts only key reads and deletes.

Because etcd writes data to disk and persists proposals on disk, its performance depends on disk performance. Although etcd is not i/O-intensive, it requires a low latency block device for optimal performance and stability. Because etcd's consensus protocol depends on persistently storing metadata to a log (WAL), etcd is sensitive to disk-write latency. Slow disks and disk activity from other processes can cause long fsync latencies.

Tip: For large volume clusters, consider using nodes with SSD- or NVMe-backed storage to contain etcd to improve performance in the cluster.

In terms of latency, run etcd on a block device that can write at least 50 I/O operations per second (IOPS) of 8000 bytes long sequentially. With a latency of 20 ms, use `fdatsync` to synchronize each write in the WAL. For heavy loaded clusters, sequential 500 IOPS of 8000 bytes (2 ms) are recommended. To measure those numbers, you can use a benchmarking tool, such as `fio`.

To achieve such performance, complete the following tasks:

- ▶ Run etcd on machines that are backed by SSD or NVMe disks with low latency and high throughput.
- ▶ Avoid NAS setups and hard disk drives.
- ▶ Always benchmark by using utilities such as `fio`.
- ▶ Continuously monitor the cluster performance as it increases.

2.4.6 Red Hat OpenShift Container Platform infrastructure baseline considerations

The two general types of nodes in your Red Hat OpenShift cluster are control (master) nodes and worker nodes. However, worker nodes can be deployed that run only infrastructure components, such as the default router, the registry, and components for monitoring. These components are defined as infrastructure nodes, and they are not counted toward the number of subscriptions that is required to run your environment.

Separating these infrastructure functions from your application worker nodes can provide a better environment for your applications because it reduces “overhead” on the application nodes. In a production deployment, it is a best practice that you deploy at least three machine sets to hold infrastructure components. Both Red Hat OpenShift Logging and Red Hat OpenShift Service Mesh deploy Elasticsearch, which requires three instances to be installed on different nodes.

The following infrastructure workloads do not incur Red Hat OpenShift Container Platform worker subscriptions:

- ▶ Kubernetes and Red Hat OpenShift Container Platform control plane services that run on masters.
- ▶ The default ingress router.
- ▶ The integrated container image registry.
- ▶ The HAProxy-based Ingress Controller.
- ▶ The cluster metrics collection, or monitoring service, including components for monitoring user-defined projects.
- ▶ Cluster aggregated logging.
- ▶ Service brokers.
- ▶ Red Hat Quay.
- ▶ Red Hat OpenShift Container Storage.
- ▶ Red Hat Advanced Cluster Manager.
- ▶ Red Hat Advanced Cluster Security for Kubernetes.
- ▶ Red Hat OpenShift GitOps.
- ▶ Red Hat OpenShift Pipelines.

Any node that runs any other container, pod, or component is a worker node that your subscription must cover.

Additional resources

For more information about infrastructure nodes and which components can run on infrastructure nodes, see the [Red Hat OpenShift control plane and infrastructure nodes](#) section in Red Hat OpenShift sizing and subscription guide for enterprise Kubernetes document.

Moving the monitoring solution

By default, the Prometheus Cluster Monitoring stack, which contains Prometheus, Grafana, and Alertmanager, is deployed to provide cluster monitoring. It is managed by the Cluster Monitoring Operator. To move its components to different machines, create and apply a custom configmap.

Moving the router

You can deploy the router pod to a different machine set. By default, the pod is deployed to a worker node.

2.4.7 Infrastructure node sizing

The infrastructure node resource requirements depend on the cluster age, nodes, and objects in the cluster because these factors can lead to an increase in the number of metrics or time series in Prometheus. The infrastructure node size recommendations that are shown in Table 2-4 are based on the results of cluster maximums and control plane density focused testing.

Table 2-4 Infrastructure node sizing recommendations

Number of worker nodes	CPU cores	Memory (GB)
25	4	15
100	8	32
250	16	128
500	32	128

2.4.8 Optimizing network performance

The Red Hat OpenShift SDN uses OpenvSwitch, virtual extensible local area network (VXLAN) tunnels, OpenFlow rules, and iptables. This network can be tuned by using jumbo frames, network interface cards (NIC) offloads, multi-queue, and ethtool settings.

VXLAN provides benefits over VLANs, such as an increase in networks from 4096 to over 16 million, and layer 2 connectivity across physical networks. With VXLAN, all pods behind a service can communicate with each other, even if they are running on different systems.

Cloud, VM, and bare metal CPU performance can handle more than 1 Gbps network throughput. When using higher bandwidth links such as 10 or 40 Gbps, reduced performance can occur. This reduced performance is a known issue in VXLAN-based environments, and it is not specific to containers or Red Hat OpenShift Container Platform. Any network that relies on VXLAN tunnels performs similarly because of the VXLAN implementation.

Routing optimization

Data traffic must move in and out of your Red Hat OpenShift cluster and between different pods and nodes running different services in the cluster. Traffic must be quickly and efficiently routed for the cluster services to be available to your users.

Scaling a Red Hat OpenShift Container Platform HAProxy router

The Red Hat OpenShift Container Platform router is the ingress point for all external traffic that is destined for Red Hat OpenShift Container Platform services.

When evaluating a single HAProxy router performance in terms of HTTP requests that are handled per second, the performance varies depending on many factors:

- ▶ HTTP keep-alive/close mode
- ▶ Route type
- ▶ TLS session resumption client support
- ▶ Number of concurrent connections per target route
- ▶ Number of target routes
- ▶ Back-end server page size
- ▶ Underlying infrastructure (network or SDN solution, CPU, and so on)

HAProxy can support routes for up to about 1000 applications, depending on the technology in use. Ingress Controller performance might be limited by the capabilities and performance of the applications behind it, such as the language that is used or if you use static or dynamic content.

Ingress, or router, sharding should be used to serve more routes toward applications and help horizontally scale the routing tier. With sharding, you can add additional Ingress Controllers to your cluster to optimize routing by creating shards, which are subsets of routes based on selected characteristics. Labels (either in the route or the namespace metadata field) are used to select which routers serve those routes. Ingress sharding is useful in cases where you want to load balance incoming traffic across multiple Ingress Controllers or when you want to isolate traffic that is routed to a specific Ingress Controller.

Using infrastructure nodes along with sharding, you can isolate and accelerate traffic in and out of your most important applications.

2.4.9 Storage considerations

Figure 2-2 summarizes the recommended and configurable storage technologies for a Red Hat OpenShift Container Platform cluster application. The figure provides general guidelines to help determine what storage types are available and what the recommended use cases for those storage types are.

Storage type	RWO [1]	ROX [2]	RWX [3]	Registry	Scaled registry	Monitoring	Logging	Apps
Block	Yes	Yes [4]	No	Configurable	Not configurable	Recommended	Recommended	Recommended
File	Yes	Yes [4]	Yes	Configurable	Configurable	Configurable [5]	Configurable [6]	Recommended
Object	Yes	Yes	Yes	Recommended	Recommended	Not configurable	Not configurable	Not configurable [7]

Figure 2-2 Storage types for Red Hat OpenShift Container Platform

1. Read Write Once (RWO).
2. Read Only Many.
3. Read Write Many (RWX).
4. Does not apply to physical disk, VM physical disk, VMDK, loopback over Network File System (NFS), AWS EBS, Azure Disk, and Cinder (the latter for block).
5. For monitoring components, using file storage with the RWX access mode is unreliable. If you use file storage, do not configure the RWX access mode on any persistent volume claims (PVCs) that are configured for use with monitoring.
6. For logging, using any shared storage would be an anti-pattern. One volume per logging-es is required.
7. Object storage is not consumed through Red Hat OpenShift Container Platform PVs or PVCs. Apps must integrate with the object storage REST API.

2.4.10 Other considerations

This section describes a few other things to consider as you create your Red Hat OpenShift environment on your IBM Power server.

Overcommit

- ▶ You can use overcommit procedures so that resources such as CPU and memory are more accessible to the parts of your cluster that need them.

Note: When you overcommit, there is a risk that another application might not have access to the resources that it requires when it needs them, which results in reduced performance.

However, this situation might be an acceptable tradeoff in favor of increased density and reduced costs. For example, development, quality assurance (QA), or test environments might be overcommitted, but production might not be.

- ▶ Red Hat OpenShift Container Platform implements resource management through the compute resource model and quota system. For more information about the Red Hat OpenShift resource model, see [Optimizing Compute Resources](#).

For more information and strategies for overcommitting, see [Placing pods onto overcommitted nodes](#).

Using a pre-deployed image to improve efficiency

You can create a base Red Hat OpenShift Container Platform image with various tasks that are built in to improve efficiency, maintain configuration consistency on all node hosts, and reduce repetitive tasks. This approach is known as a *pre-deployed image*.

Pre-pulling images

To produce images efficiently, you can pre-pull any necessary container images to all node hosts. The image does not have to be initially pulled, which saves time and performance over slow connections, especially for images (source-to-image (S2I)², metrics, and logging) that can be large.

Optimizing persistent storage

Optimizing storage helps to minimize storage use across all resources. By optimizing storage, administrators help ensure that existing storage resources are working efficiently.

² https://docs.openshift.com/container-platform/3.11/architecture/core_concepts/builds_and_image_streams.html#source-build

2.5 Red Hat OpenShift starting configuration

Table 2-5 shows example hardware requirements for a customer. The baseline estimate may be changed based on further requirements.

Table 2-5 Baseline configuration for Red Hat OpenShift

Machine	Operating system	Sizing according to the questionnaire		Local storage or node	Number of machines
		CPU	RAM		
Bootstrap (Temp)	Red Hat Enterprise Linux CoreOS (RHCOS)	8	16	120 GB	1
Bastion or installation (for triggering deployment)	RHEL 7 or 8	2	8	200 GB	1
Control	RHCOS	4	16	300 GB	3
Compute ^a	RHCOS	8	32	120 GB	3
Infra ^b	RHCOS	4	16	120 GB	3
Storage (OCS or Red Hat OpenShift Data Foundation) ^c	RHCOS	4	16	120 GB	3

a. The compute node resource requirement is based on sizing requirements.

b. Infra nodes are based on customer requirements. These nodes are optional.

c. Storage is based on customer requirements with additional external storage.

The resources that are specified in Table 2-5 are meant as a starting point, and the actual resources will likely change based on your specific requirements. Installation is based on the user-provisioned infrastructure (UPI) or installer-provisioned infrastructure (IPI) within the scenario.

2.6 Tools

This section shows what tools that you can use to monitor performance-related characteristics of a system and its applications, and IBM Power server and IBM Power Systems Virtual Server (IBM PowerVS) specific tools.

2.6.1 Observability

Observability is more than old school monitoring in the sense that you try to understand the internal state of a system by knowing most of the possible external outputs that the system produces. Observability can help with faster problem identification and resolution. You can view observability as an evolution of traditional application performance monitoring, which provides the necessary tools to manage distributed and highly dynamic application environments, which include rapid changes to the running services. Traditional monitoring and Application and Performance Monitoring (APM) tools with a once-a-minute sampling rate cannot keep pace anymore.

Observability can discover and collect telemetry data, which can be logs, metrics, traces, and dependencies. This data can help SREs, DevOps teams, and other IT personnel by providing complete, contextual information to help resolve performance issues, for example.

Automation is a key feature in this rapidly changing application environment. One of the main differentiators of observability tools is the ability to automatically discover new telemetry sources and integrate the collected information while filtering out noise (unrelated data). The main benefit of observability in contrast to traditional monitoring is its ability to discover and address the “unknowns”.

2.6.2 IBM Instana

IBM Instana provides an Enterprise Observability Platform (with automated application performance monitoring capabilities) to businesses that operate complex, modern, and cloud-native applications on-premises, or in public and private clouds, including mobile devices or IBM zSystems mainframe computers. You can control modern hybrid applications with the Instana artificial intelligence (AI)-powered discovery of deep contextual dependencies inside hybrid applications.

Instana also provides visibility into development pipelines to help enable closed-loop DevOps automation. These capabilities provide actionable feedback that is needed for customers as they optimize application performance, enable innovation, and mitigate risk, which helps DevOps increase efficiency and add value to software delivery pipelines while meeting their service-level and business-level objectives.

Features

Instana provides the following features:

- ▶ Automated discovery by using a lightweight agent and sensors that automatically collect data with a 1-second granularity. Every request that is made by microservices is traced, and the response time and context is captured. This data is enhanced with other related metrics to produce a complete picture of the applications and infrastructure.
- ▶ Builds a dependency map by using the gathered data.

- ▶ Helps root cause analysts by analyzing the incoming data in real time and creating issues and incidents that are raised if users are impacted. An incident includes metrics, traces, exceptions, logged data, and configuration data, which are correlated through the Dynamic Graph.
- ▶ Performance optimization through Unbounded Analytics, which uses all the collected trace information. The information can be filtered for performance outliers, patterns of known problem signs, and traces that are tagged uniquely.

Instana uses sensors to provide automated infrastructure and application monitoring with no plug-ins or application restarts. Each sensor supports an application component, middleware component, operating system, or other integration point so that you can manage and monitor your infrastructure. At the time of writing, Instana has the following integrations:

- ▶ AI Ops integrations (18):
 - CI/CD Automation (7)
 - DevOps Tools (9)
- ▶ Cloud Operations (24)
- ▶ Containers and Orchestration (23)
- ▶ User Monitoring (3)
- ▶ Infrastructure and Middleware Components (114):
 - Database (37)
 - Messaging (25)
 - OS (11)
 - Web / App Servers (39)
- ▶ Kubernetes Distributions (5)
- ▶ Legacy Middleware (3)
- ▶ Secrets and Identity Management (2)
- ▶ Serverless (4)
- ▶ Tracing, Supported Languages, and Frameworks (34):
 - Application Frameworks (7)
 - Application Monitoring (17)
 - Proxies and Service Meshes (4)
 - Tracing Technology (7)

Instana has a GUI that can be used through a web browser.

Instana architecture

Instana provides automatic, continuous discovery of your application stack. A single, lightweight agent per host continually discovers all the components and deploys sensors that are crafted to monitor each technology. With no human intervention, sensors automatically collect configuration, changes, metrics, and events. Metrics from all components are collected in high fidelity with a 1-second data granularity as every request is traced across each microservice, automatically capturing the response time and context.

To understand how a system of services works together and the impact of component failure, Instana enhances traces with information about the underlying service, application, and system infrastructure by using the Dynamic Graph. The Dynamic Graph provides a dependency map so that you can get to root cause of issues quickly.

Figure 2-3 shows the architecture of the Instana Enterprise Observability platform.³

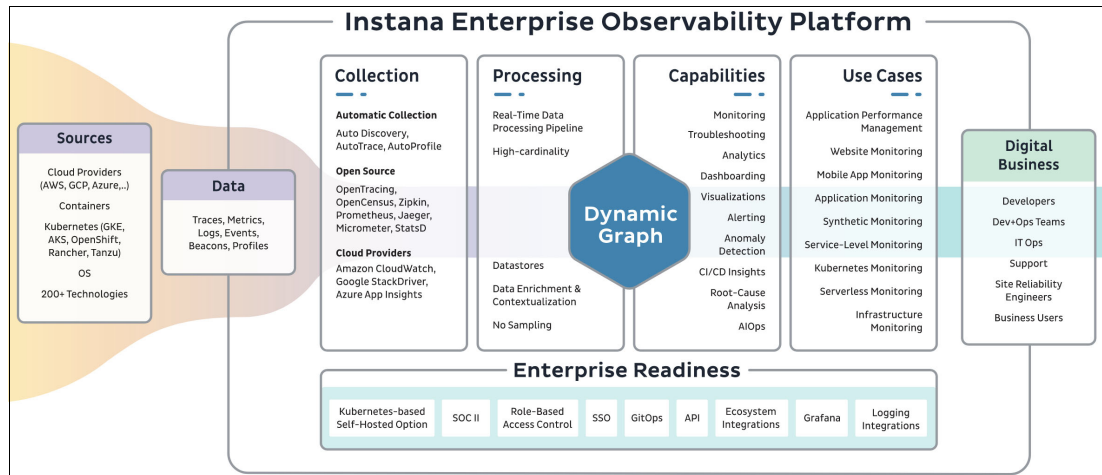


Figure 2-3 Instana platform architecture

Installation methods

The Instana agent can be installed in many ways depending on the target infrastructure. For Red Hat OpenShift clusters, there are the Operator-based, Helm-based, and YAML file-based manual installation methods. Compared to Kubernetes, there are extra prerequisites for Red Hat OpenShift, so read the documentation before starting the installation.

Note: At the time of writing, Operator-based installation is not supported on IBM Power servers because there is no image for `instana-agent-operator` that is available for the `ppc64le` architecture on OperatorHub.

On Red Hat OpenShift and Kubernetes clusters, Instana agents are defined and running as pods that are managed by a DaemonSet, which means that all worker nodes have an agent running with the same configuration by default. The configuration is managed as a configmap in Red Hat OpenShift. The Red Hat OpenShift cluster nodes can be seen in the Instana Infrastructure window, as shown in the Figure 2-4.

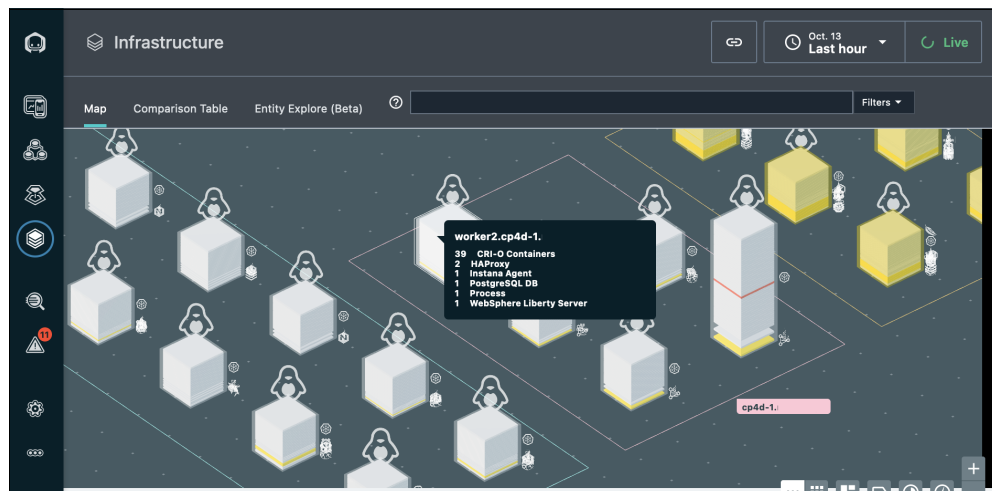


Figure 2-4 Instana: Infrastructure window

³ <https://instanaimg.imgix.net/media/ObservabilityGraph-01.svg>

Monitoring IBM Power servers CPU utilization in Instana

Figure 2-5 shows CPU utilization statistics for an IBM Power server that is on a Red Hat OpenShift node.

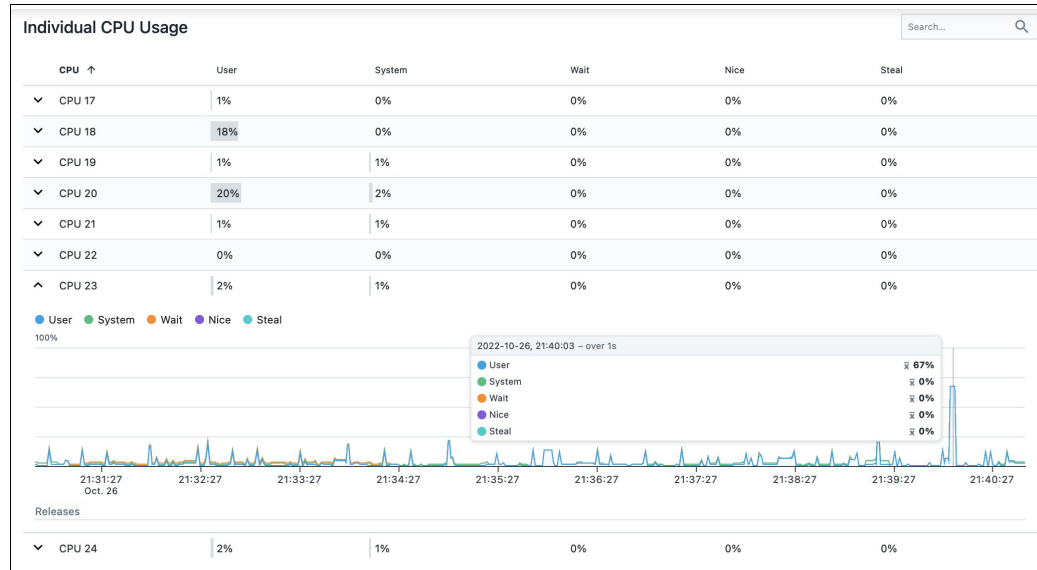


Figure 2-5 CPU utilization

The number of CPUs is based on actual virtual CPUs that are assigned to the logical partition (LPAR) and the simultaneous multi-threading (SMT) settings. In this example, the system has four vCPUs and SMT is set to 8 by default, so 32 CPUs are shown in Instana and by the commands that are shown in Example 2-1. The commands are started in terminal window of the Red Hat OpenShift GUI. They also can run after running the command-line interface (CLI) command `oc debug node/"nodename"`.

Example 2-1 shows the `lscpu` and `lparstat` Linux commands to check allocated CPUs and settings. The `lparstat` command is available only on LPARs (not on bare metal servers).

Example 2-1 The `lscpu` and `lparstat` commands

```
sh-4.4# chroot /host

sh-4.4# lscpu
Architecture:      ppc64le
Byte Order:        Little Endian
CPU(s):            32
On-line CPU(s) list: 0-31
Thread(s) per core: 8
Core(s) per socket: 4
Socket(s):         1
NUMA node(s):     1
Model:             2.0 (pvr 0080 0200)
Model name:        Power10 (architected), altivec supported
Hypervisor vendor: pHyp
Virtualization type: para
L1d cache:         32K
L1i cache:         48K
L2 cache:          1024K
L3 cache:          4096K
NUMA node10 CPU(s): 0-31
```

```
Physical sockets: 16
Physical chips: 1
Physical cores/chip: 15
```

```
sh-4.4# lparstat -i
Node Name : worker1.example.com
Partition Name : cp4d-1-worker-1
Partition Number : 188
Type : Dedicated
Mode : Capped
Entitled Capacity : 4.00
Partition Group-ID : 32956
Online Virtual CPUs : 4
Maximum Virtual CPUs : 4
Minimum Virtual CPUs : 1
Online Memory : 133980160 kB
Minimum Memory : 1024
Desired Memory : 131072
Maximum Memory : 136902082560
Minimum Capacity : 1.00
Maximum Capacity : 4.00
Capacity Increment : 1.00
Active Physical CPUs in system : 237
Active CPUs in Pool : 0
Shared Physical CPUS in system : 0
Maximum Capacity of Pool : 0.00
Entitled Capacity of Pool : 0
Unallocated Processor Capacity : 0
Physical CPU Percentage : 100
Unallocated Weight : 0
Memory Mode : Dedicated
Total I/O Memory Entitlement : 137438953472
Variable Memory Capacity Weight : 0
Memory Pool ID : 65535
Unallocated Variable Memory Capacity Weight : 0
Unallocated I/O Memory Entitlement : 0
Memory Group ID of LPAR : 32956
Desired Variable Capacity Weight : 0
```

IBM Power Hardware Management Console Instana sensor

IBM Power Hardware Management Console (HMC) provides interfaces to monitor the utilization of physical and virtual resources of IBM Power servers. The interfaces are REST API interfaces for Performance and Capacity Monitoring (PCM).

Instana has a sensor that uses this REST API to collect performance data and discover any performance-related anomalies. The sensor is supported in HMC Version 10 Release 1 Service Pack 1010 and later.

Figure 2-6 on page 31 shows the architecture of the IBM Power HMC Instana sensor.

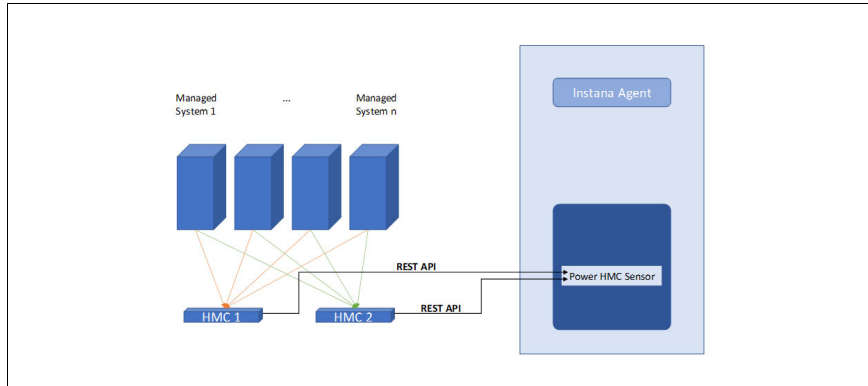


Figure 2-6 IBM Power HMC Instana sensor architecture

The sensor works as a remote sensor, so it must be configured on another monitored system, and it connects to the HMC from there.

Prerequisites

Here are the prerequisites for IBM Power HMC Sensor:

- ▶ The user that is set in the sensor configuration must have the `hmcviewer` role on the HMC.
- ▶ PCM must be enabled for the managed system to monitor.
- ▶ Enable the following flags:
 - `LongTermMonitoringEnabled`
 - `AggregationEnabled`
 - `EnergyMonitorEnabled`

Setup and configuration

Because the sensor is a remote sensor, the Instana Agent must be installed first on a system that can access the HMC.

The Instana Agent has a configuration file that is stored in the following directory:

```
<agent_install_dir>/etc/instana
```

On a Linux server, the directory is `/opt/instana/agent/etc/instana`.

The configuration file name is `configuration.yaml`, which has the settings that are listed in Example 2-2.

Example 2-2 IBM Power HMC Sensor configuration settings in the `configuration.yaml` file

```
#PowerHMC
#com.instana.plugin.powerhmc:
# remote: # multiple hosts supported
#   - host: ''# hostname or IP of PowerHMC server
#     port: ''# default port is '12443' of PowerHMC API Server
#     user: '' # username to access the PowerHMC server
#     password: '' # password to access the PowerHMC server
#     availabilityZone: 'PowerHMC Remote Monitoring'
#     poll_rate: 300 # Poll rate in seconds. Poll rate cannot be lesser than 300 seconds. If it
# is configured below 300 seconds, then default value (300 seconds) will be set.
#     eventsPollRate: 900 # Poll rate in seconds. Poll rate cannot be lesser than 900 seconds.
# If it is configured below 900 seconds, then default value (900 seconds) will be set.
```

```
#      connectionTimeout: 50 # It is the timeout until a connection with the server is
established. Default is 50 seconds.
#      connectionRequestTimeout: 50 # It is the time to fetch a connection from the connection
pool. Default is 50 seconds.
#      socketTimeout: 50 # It is socket read time out. Default is 50 seconds.
```

The sensor can collect metrics about the following items:

- ▶ Processor, memory, and network metrics for IBM Power managed servers.
- ▶ Processor and memory metrics for hypervisor.
- ▶ Processor, memory, network, and storage metrics for LPARs and Virtual I/O Server (VIOS).
- ▶ CPU and memory utilization, power consumption, LPAR data, and more.

The default collection granularity is 300 seconds. For more information, see the full list of collected metrics [Monitoring IBM Power HMC](#).

Troubleshooting

If the monitored IBM Power HMC uses self-signed certificates, then these certificates must be imported into the Instana Agent trusted certificate store. Example 2-3 shows the error messages that appear if the certificates are not in the store.

Example 2-3 Self-signed certificate errors

```
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid
certification path to requested target. PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid
certification path to requested target.
```

Download and import the HMC certificate file, but check where the certificate store is on the installation before using a command to import it.

The certificate file can be downloaded through a web browser or by using the commands that are shown in Example 2-4. Before running the commands, set the *HMC*, *PORT*, and *SERVERNAME* variables according to the actual setup.

Example 2-4 Getting and importing the HMC certificate into the Instana Agent certificate store

```
# echo -n | openssl s_client -connect $HOST:$PORT -servername $SERVERNAME |
openssl x509 > $SERVERNAME.cer
# export CACERTS=$(find /opt/instana -name cacerts)
# keytool -import -alias ibm.com -keystore $CACERTS -file $SERVERNAME.cer
-storepass changeit
```

Note: In our example, we store the actual location of the cacerts file because the documentation points to a different location and the command would fail.

Performance-tuning-related considerations

Instana provides real-time monitoring with 1-second collection granularity for metrics, logs, and traces. Instana also provides Unbound Analytics⁴ to speed up root cause analysis. Collecting and combining this data with IBM Power server-based metrics from HMC sensor extends the base capabilities to uncover performance-related problems.

⁴ <https://www.ibm.com/docs/en/instana-observability/current?topic=capabilities-unbounded-analytics>

Figure 2-7 shows a view of an Instana instance with a managed IBM Power HMC.

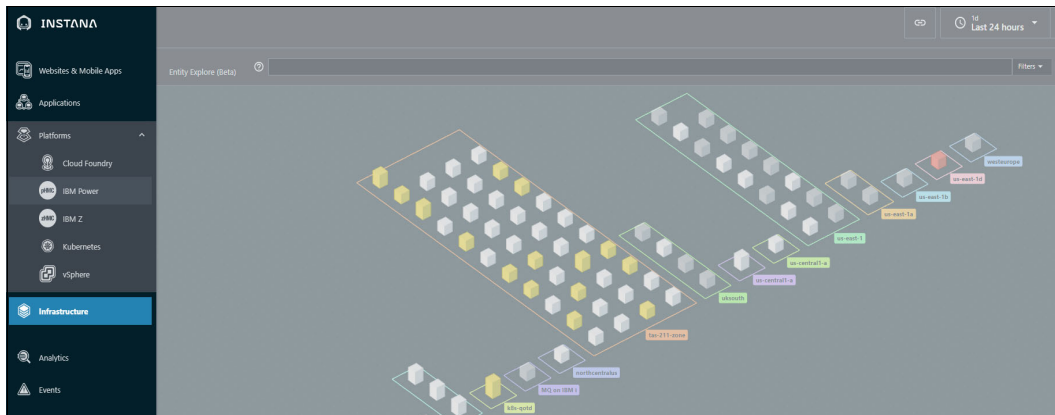


Figure 2-7 Instana Platform menu with Power HMC entry

Figure 2-8 - Figure 2-13 on page 35 are examples of Instana metrics that are gathered from the IBM Power server through the HMC.

Figure 2-8 shows the IBM Power servers that are managed by the HMC.

The screenshot shows the 'IBM Power HMC' dashboard with a 'Systems' tab. It displays a table of server metrics:

Name	Partitions	Virtual I/O Servers	Utilized Processing Units	Utilized Processing Units (%)	Memory Available (MB)	Memory Available (%)
Server-8247-22L-SN	13	2	3	16%	43,008	15%
Server-8286-42A-SN	11	2	3	13%	626,176	60%
Server-9119-MHE-SN	22	2	23	24%	11,785,000	94%

Figure 2-8 IBM Power servers that are shown in the IBM Power HMC dashboard

Figure 2-9 shows the summary of one of the servers.

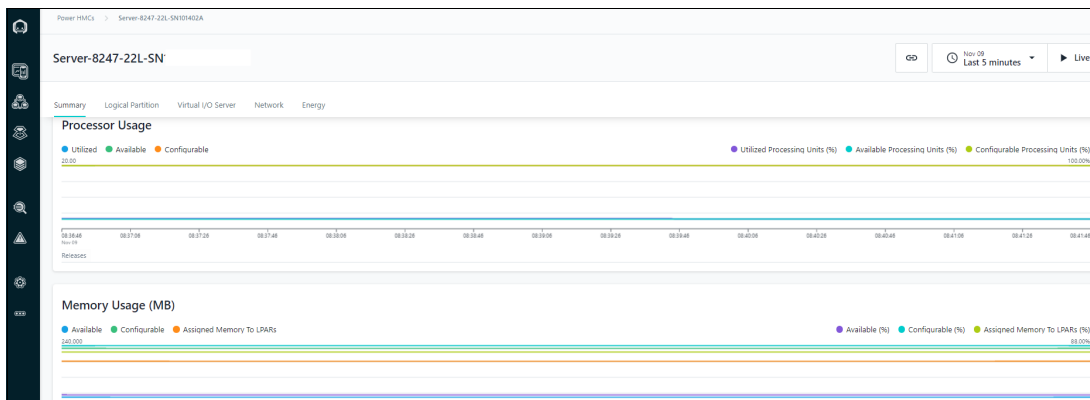


Figure 2-9 IBM Power server summary window

Figure 2-10 shows the LPARs running on one of the servers.

The screenshot displays the Instana interface for a server named 'Server-8247-22L-SN'. It shows a list of Logical Partitions (LPARs) with the following data:

Name	Entitled Processing Units Used	Memory (MB)	Maximum Virtual Processors	Mode	State
ADXT1-NXST-180234bc-000003a8	1%	4,096	1	Uncapped	Running
RHEL8_image_ef13de3e-000001ed	0%	4,096	1	Uncapped	Running
ansctf1108-243cdca-0000038e	0%	16,384	4	Uncapped	Running
ansble-demo-c9123090-00000398	2%	4,096	1	Uncapped	Running
ansble2-1819c0ba-00000391	0%	32,768	8	Uncapped	Running
dcloudadmin_c-255c5376-000003a4	1%	8,192	2	Uncapped	Running
mg_SAP_HANA_P-9e0823ba-000003a0	0%	16,384	4	Uncapped	Running
ocp4p8-m3-de36ea33-000003c6	7%	16,384	1	Uncapped	Running

Figure 2-10 LPARs of a specific IBM Power server

Figure 2-11 shows the processor utilization of the VIOS on the managed server.

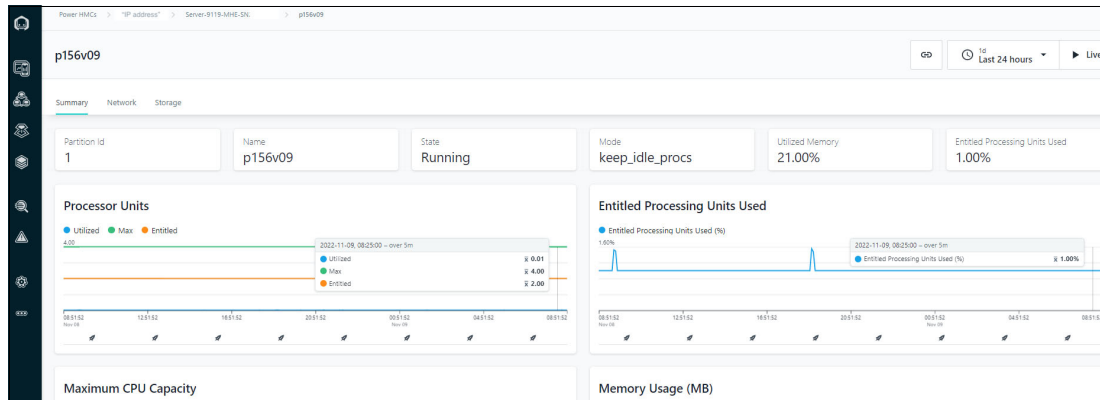


Figure 2-11 VIOS processor utilization shown in Instana

Figure 2-12 shows the processor utilization for one of the LPARs.

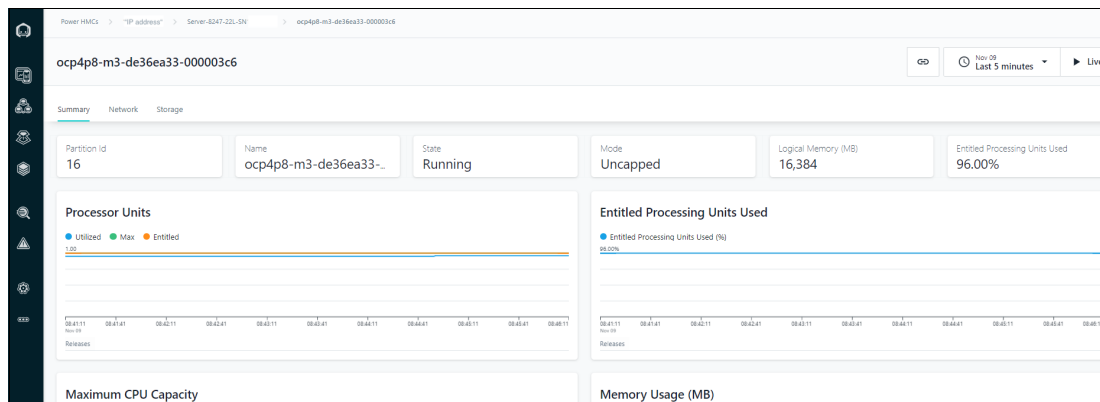


Figure 2-12 LPAR processor utilization shown in Instana

Figure 2-13 on page 35 shows the network utilization of one of those LPARs.

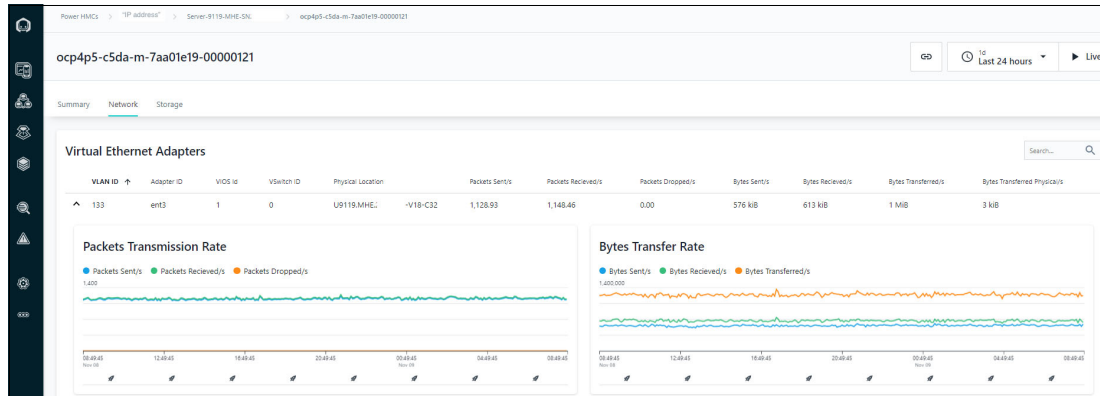


Figure 2-13 Network utilization

2.6.3 IBM Turbonomic Application Resource Management

In hybrid cloud, there are two main problems when you provision infrastructure for applications and services: You can overprovision, or you can request and provision fewer resources than what is needed for your application at peak time. Overprovisioning is expensive because you are not using the resources for which you paid, but you cannot risk not supporting the peak load of your systems. There will be times where you have requests that cannot be served due to resource contention, unforeseen load, or planned outages during a peak load period.

IBM Turbonomic® Application Resource Management (Turbonomic) is designed to solve these issues while still keeping the cost of your solution at an optimal level.

With Turbonomic, you can automate critical actions that proactively deliver the most efficient utilization of compute, storage, and network resources to your apps at every layer of the stack. This task is done continuously in real time and without human intervention. The following features help with this task:

- ▶ Full-stack visualization by using an application-centric, top-down approach to discover how each entity in the system from application and services to infrastructure layer impacts the behavior of the business application.
- ▶ AI-powered insights drives actions that are preventive, preemptive, and precise, and can be automated.
- ▶ With the help of intelligent automation, you gain speed, elasticity, and cost savings.
- ▶ Integrations with most of the players in the market, from application management to hypervisors, from databases to storage.

The four main use cases of Turbonomic are the following ones:

- ▶ Cloud optimization
- ▶ Data center optimization
- ▶ Kubernetes optimization
- ▶ Sustainable IT

Turbonomic architecture

Figure 2-14 shows the architecture of Turbonomic.

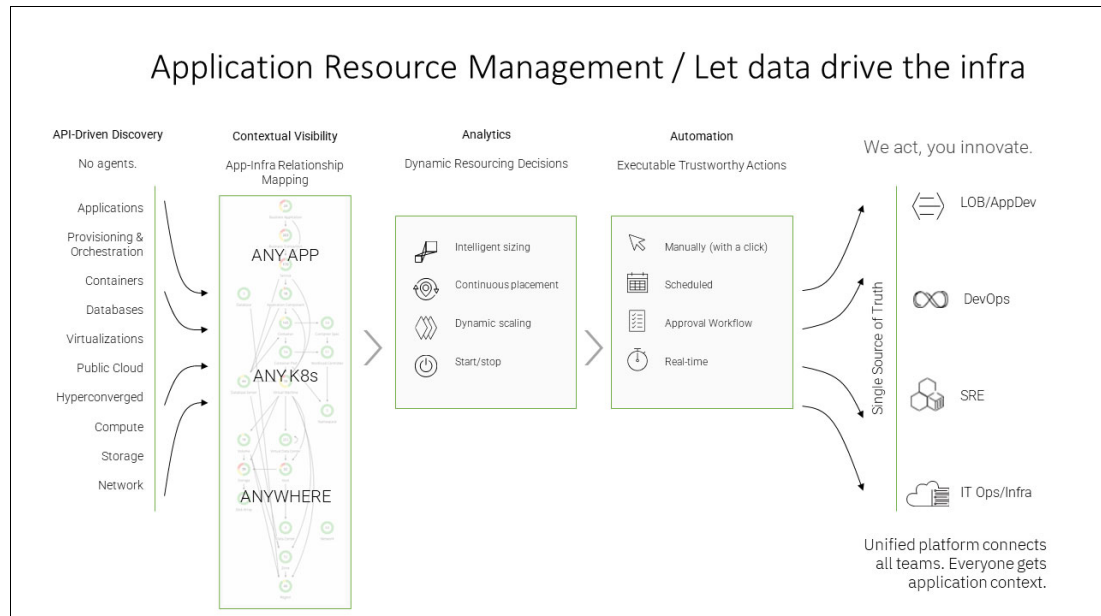


Figure 2-14 Turbonomic architecture

At the time of writing, Turbonomic does not provide integration with IBM PowerVM and HMC. However, it does support Instana, which has a sensor for IBM Power HMC for IBM Power server monitoring, which provides a full picture of application resource monitoring for IBM Power servers.

2.6.4 Sysdig

Sysdig is a software as a service (SaaS) provider for security and monitoring tools. It can help embed security and compliance management into DevOps workflows.

Here are the main features of Sysdig:

- ▶ Infrastructure as code (IaC) security: Shift security further left.
- ▶ Cloud Security Posture Management: Continuous security of cloud-based services by flagging misconfiguration, suspicious activities, and excessive and unnecessary permissions.
- ▶ Vulnerability management: Consolidate container and host security scanning, and build security scanning in DevOps workflows.
- ▶ Threat detection and response: Consolidate and unify threat detection with incident response for containers, Kubernetes, and cloud.
- ▶ Network segmentation: Help configure micro-segmentation by using and automating a Kubernetes-native network policy.
- ▶ Monitoring and troubleshooting: Provide monitoring with deep insight by using Kubernetes-native Prometheus.
- ▶ Compliance: Validate compliance with major security standards like PCI, SOC2, and NIST for containers and hosts.

Architecture

The Sysdig platform is split into two major parts: Sysdig Secure and Sysdig Monitor.

Figure 2-15 shows the architecture of the platform.

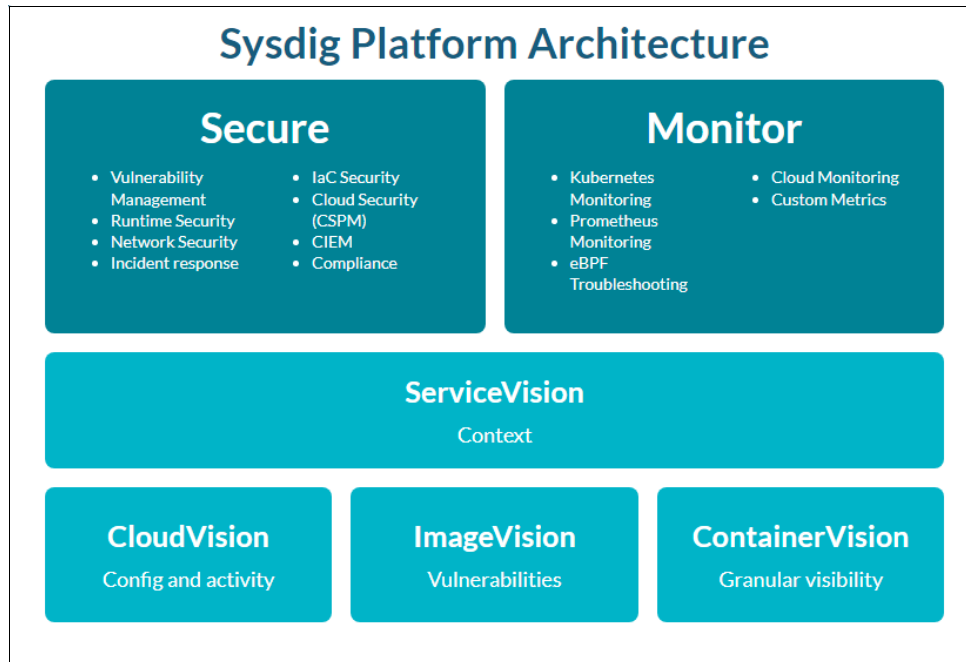


Figure 2-15 Sysdig architecture

Sysdig is built on an open-source security stack that provides accelerated innovation and standardization. Its monitoring capabilities are built on Prometheus, so Red Hat OpenShift workloads on IBM Power Architecture® based servers can be monitored in a standard way. Sysdig is an agent-based service. Monitored sources can be Kubernetes and Red Hat OpenShift clusters, Linux machines, and Docker containers.

IBM Cloud Monitoring

Sysdig also provides integration with all major cloud providers. Therefore, it is integrated into IBM Cloud® Monitoring, and it is the base of the capabilities of that service.

IBM Cloud Monitoring can be found and activated from the IBM Cloud catalog, as shown in Figure 2-16.

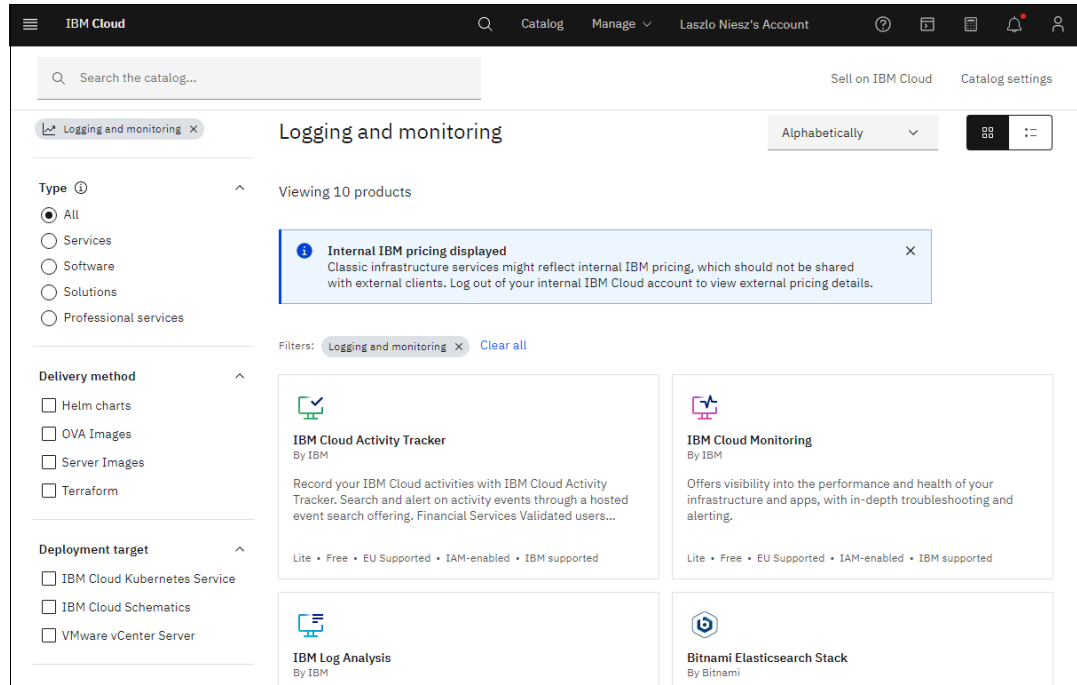


Figure 2-16 IBM Cloud Monitoring in the IBM Cloud catalog

This service can monitor Red Hat OpenShift and other Kubernetes clusters that are running in IBM Cloud. There is a no-charge version of this service for 30 days with some restrictions, but it is an option to learn about IBM Cloud Monitoring and Sysdig.

2.6.5 Prometheus

Prometheus⁵ is an open-source systems monitoring and alerting toolkit that originally was built by SoundCloud. Since its inception in 2012, many companies and organizations have adopted Prometheus, and the project has an active developer and user community. Prometheus is now a stand-alone, open-source project that is maintained independently of any company. To emphasize this situation and clarify the project's governance structure, Prometheus joined the Cloud Native Computing Foundation (CNCF) in 2016 as the second hosted project after Kubernetes.

Prometheus collects and stores its metrics as time series data, that is, metrics information is stored with the timestamp at which it was recorded, alongside optional key-value pairs that are called labels.

Red Hat OpenShift Monitoring

Red Hat OpenShift contains a full-featured monitoring stack that you can use to monitor the cluster health by default, but you can use it for user projects too. A key part of this stack is Prometheus, which is shown in Figure 2-17 on page 39.

⁵ <https://prometheus.io/docs/introduction/overview/#what-is-prometheus>

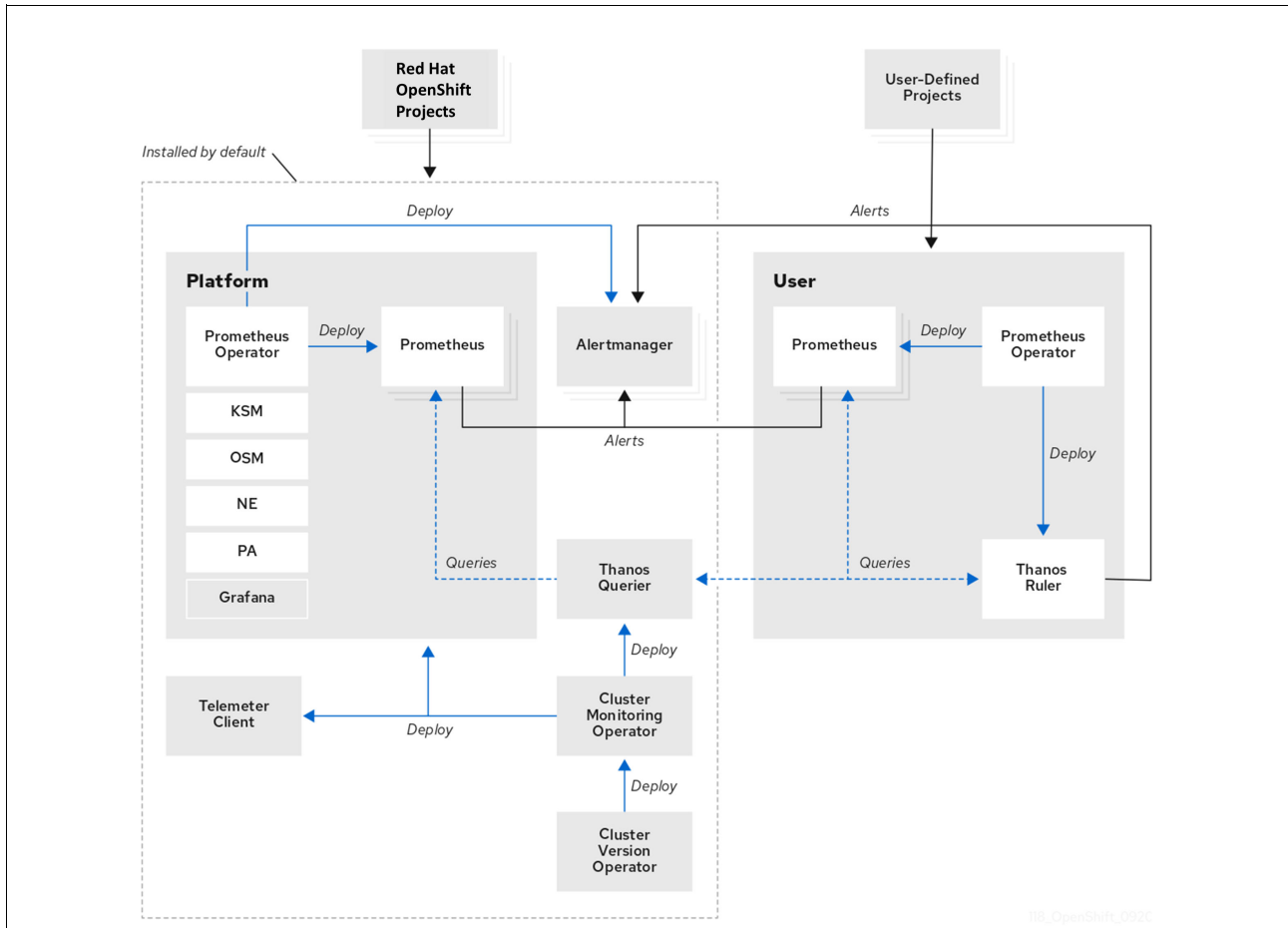


Figure 2-17 Red Hat OpenShift Monitoring stack

Figure 2-17 shows the parts of the monitoring stack that are installed by default. They monitor the Red Hat OpenShift platform base and controlling components. These components are deployed into the `openshift-monitoring` namespace.

Also, user-defined projects and applications can be monitored if enabled. In this case, there will be a new namespace that is created and named `openshift-user-workload-monitoring`.

Prometheus as a part of Red Hat OpenShift Monitoring provides a time-series database and a rule engine for metrics. It sends alerts to Alertmanager. In Red Hat OpenShift, it is controlled by the Prometheus Operator, which manages Prometheus instances and automatically generates monitoring target configurations based on Kubernetes labels. The monitoring of operator system metrics and nodes is done by a node-exporter agent.

After you enable user-defined project monitoring, a separate instance of Prometheus is created in namespace `openshift-user-workload-monitoring`.

Because node-related monitoring is implemented in the base Prometheus instance, we concentrate on that instance in this section.

Figure 2-18 shows a node as the target of the node-exporter agent.

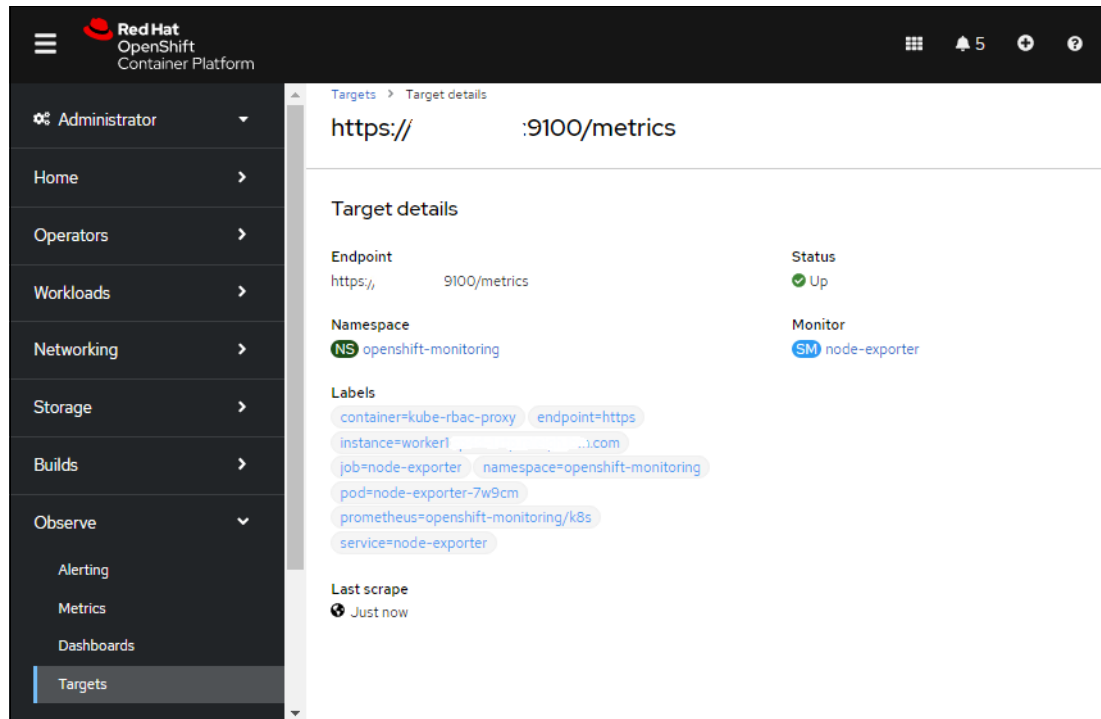


Figure 2-18 Red Hat OpenShift node as a node-exporter target

Red Hat OpenShift provides an observability framework, and the GUI has the following features for this framework, which you can find by clicking the **Observe** menu:

- ▶ **Alerting:** You see alerting rules that are based on metrics that are collected by Prometheus, silencers (to silence a predefined alerting rule), and the alerts that are fired by alerting rules.
- ▶ **Metrics:** This window provides a GUI to build custom queries that are based on the metrics that are collected by Red Hat OpenShift Monitoring.
- ▶ **Dashboards:** Preconfigured dashboards for monitoring, with the possibility to dig deeper and show or edit the query of each dashboard element by clicking **Inspect**.
- ▶ **Targets:** Shows all the monitoring targets that are supported by the platform.

If you go to the Dashboard window by selecting **Observe** → **Dashboard**, you see two dashboards that are based on node-export by default:

- ▶ **Node Exporter / USE Method / Cluster:** This dashboard has cluster-wide data by nodes.
- ▶ **Node Exporter / USE Method / Node:** This dashboard has the same data, but only for the selected node.

By default, the dashboard does not have any IBM Power Architecture specific metrics.

Figure 2-19 shows the cluster dashboard.

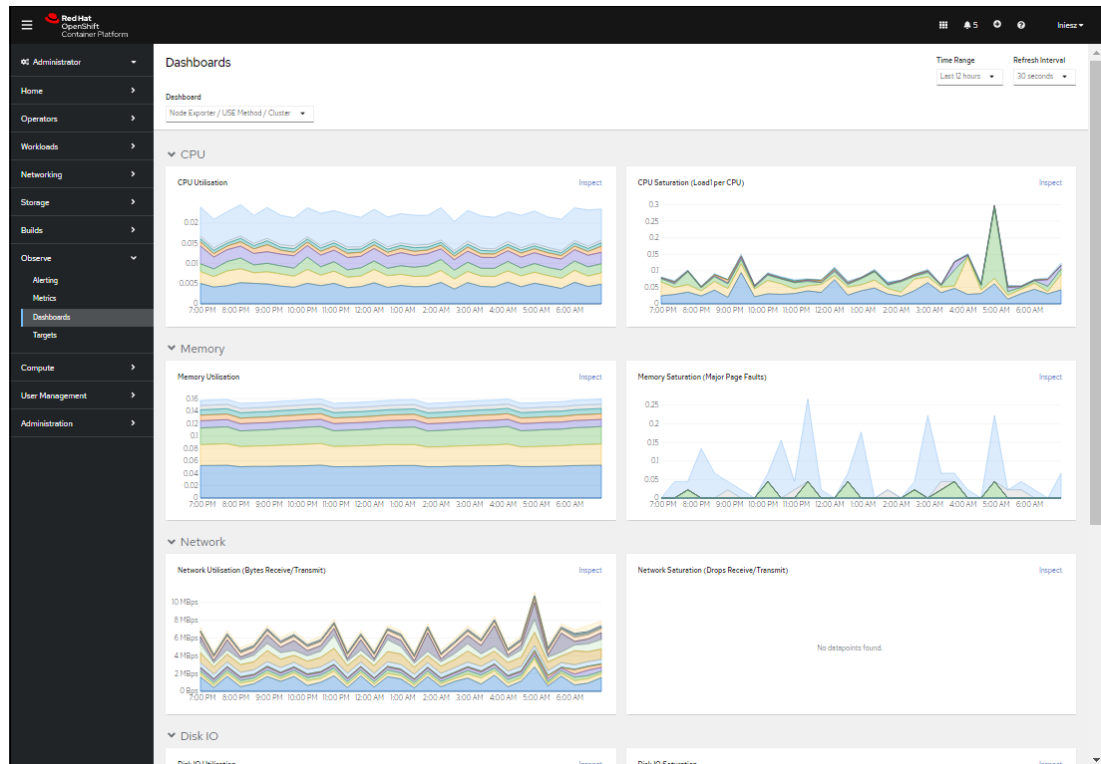


Figure 2-19 Node Exporter / USE Method / Cluster

Prometheus has a user interface (UI), but it is deprecated in Red Hat OpenShift 4.10 and will be removed from Version 4.11 because Red Hat OpenShift has its own Metrics UI that you can access by selecting **Observe** → **Metrics**.

To see the detailed configuration of predefined alerts and recored metrics, see the ConfigMaps `prometheus-k8s-rules-0` in the `openshift-monitoring` namespace. This namespace is mounted into the Prometheus pod as `/etc/prometheus/rules/prometheus-k8s-rules-0`, which is used when the pod starts.

2.6.6 Grafana

Grafana is a widely used operational dashboard that can be connected to many metric collection systems, such as Prometheus.

Grafana provided the dashboard service in earlier versions of Red Hat OpenShift, but the service is deprecated in Version 4.10 and will be removed from Version 4.11. A similar function is available by using Red Hat OpenShift dashboards, which you can view by selecting **Observe** → **Dashboards**.

To configure Grafana, first define the data sources, which in Red Hat OpenShift is normally the integrated Prometheus. This configuration in Red Hat OpenShift is done by using a secret that is named grafana-datasources-v2. The default data source in Red Hat OpenShift looks like Example 2-5.

Example 2-5 The prometheus.yaml file

```
{
  "apiVersion": 1,
  "datasources": [
    {
      "access": "proxy",
      "basicAuth": true,
      "secureJsonData": {
        "basicAuthPassword":
"8BCcK+MXnQUwzgYGcKEq5k+InSNWXZkueMRbZQvn8T/DwdFQ4XpfcKv5g76gjKfZgxQZBAJajyiQkRKJw
kRJk4hecJUHw1SCu0wGTm16/NCxVomCWvnMNB7pZYWSDv0QGDpK6VC066RdLhX7vF+SUZEe33/x4A+5iw
EN78wWtoMx+Ehb7WboET7jESxqpr1Yanj5Nr+bLeLslcwj2qYH0gxkrwKZBm06YqKrYSeNqbJfm0kvwuFa
/QPQJBt8hEI4xBFWchTS0BrHaSjKhC/8zd+Z7mtK/i9VGioKEqEX0zYRAxGBh4GgdjX1Tn233tFBgypScD
tZ9FgJawhMIoy"
      },
      "basicAuthUser": "internal",
      "editable": false,
      "jsonData": {
        "tlsSkipVerify": true
      },
      "name": "prometheus",
      "orgId": 1,
      "type": "prometheus",
      "url": "https://prometheus-k8s.openshift-monitoring.svc:9091",
      "version": 1
    }
  ]
}
```

This secret and all others are mounted in `/etc/grafana/provisioning/datasources` of the Grafana pod directory. You can choose these secrets to create a dashboard from the collected data.

The data source can be local, in which case use the internal service URL, or it can be remote. A single Grafana server can show dashboards for many remote data sources.

Dashboards are configured as JSON files. In Red Hat OpenShift, the configuration of the dashboards is stored in ConfigMaps, which are mounted in a directory that is specified by another configmap, which is called `grafana-dashboards`. Many specialized dashboard configurations can be downloaded from [Grafana Dashboards](#).

Chapter 5, “Red Hat OpenShift V4.x on IBM Power Systems cluster administration and monitoring”, in *Red Hat OpenShift V4.X and IBM Cloud Pak on IBM Power Systems Volume 2*, SG24-8486 contains a good example about how to use Grafana to monitor IBM Power HMC collected performance data.



IBM Power processor performance capabilities

This chapter describes how you can leverage the performance features that are integrated into the IBM Power processor when you are running your containerized workloads.

The IBM Power Architecture has many performance enhancements over competing products that can reduce your investment in both hardware and software in a Red Hat OpenShift environment.

This chapter contains the following topics:

- ▶ IBM Power hardware
- ▶ IBM Power Systems Virtual Server
- ▶ Components

3.1 IBM Power hardware

IBM Power servers have been in the market for the last 30 years, and they have proven their reliability, availability, and serviceability over those years. Its robust technology is used by clients in every industry worldwide who are looking for an enterprise-grade solution to meet their business requirements. IBM Power technology provides powerful features to protect data by using built-in data encryption. IBM Power servers are designed to reduce energy consumption while still providing robust performance that increases with each new family generation.

IBM Power servers lead the industry in infrastructure reliability with 25% lower downtime versus comparable high-end servers.¹ With IBM Power E1080 (Power E1080), we make the most reliable server platform in its class even better with advanced recovery, diagnostic capabilities, and Open Memory Interface (OMI)-attached advance memory DIMMs. The continuous operation of in-memory systems depends on memory reliability because of their large memory footprint. IBM Power10 processor-based server memory DIMMs deliver 2X better memory reliability and availability than industry-standard DIMMs.²

With high-performance features like up to 8 way multi-threading (SMT8) capability, high-speed memory interconnections, and high-speed data and networking connections, the IBM POWER® processor family can run more processes per core than competing architectures, which results in a decreased investment in hardware and software investment. The IBM Power10 family provides a range of options, from small entry servers like IBM Power S1014 (Power S1014) to enterprise architectures like the Power E1080.

3.1.1 IBM Power10 processor-based server capabilities

The IBM Power10 processor was introduced to the public on 17 August 2020 at the 32nd HOT CHIPS³ semiconductor conference. At that meeting, the new capabilities and features of the latest IBM POWER processor microarchitecture and the IBM Power Instruction Set Architecture (ISA) 3.1B were revealed and categorized according to the following IBM Power10 processor design priority focus areas:

- ▶ Data plane bandwidth focus area
 - Terabyte per second signaling bandwidth on processor functional interfaces, petabyte system memory capacities, 16-socket symmetric multiprocessing (SMP) scalability, and memory clustering and memory inception capability.
- ▶ Powerful enterprise core focus area
 - New core micro-architecture, flexibility, larger caches, and reduced latencies.
- ▶ End-to-end security focus area
 - Hardware enabled security features that are co-optimized with IBM PowerVM hypervisor support.

¹ <https://www.ibm.com/downloads/cas/VQ5B65YZ>

² Based on IBM internal analysis of the IBM product failure rate of differential DIMM (DDIMMs) versus industry-standard DIMMs.

³ <https://hotchips.org/>

- ▶ Energy-efficiency focus area
Up to threefold energy-efficiency improvement in comparison to IBM POWER9™ processor technology.
- ▶ Artificial intelligence (AI)-infused core focus area
A 10 - 20x matrix-math performance improvement per socket compared to the IBM POWER9 processor technology capability.

As a result of these design objectives, IBM Power10 processor cores have a new set of improvements that bring the most powerful IBM processor to the market. The core was designed around five market trends based on client's needs:

- ▶ AI to provide rapid adoption of AI to realize growth and operational benefits.
Faster adoption of AI technologies compared to IBM POWER9 from enhanced in-core AI inferencing capability in every server without requiring GPUs or any additional specialized hardware, which reduces the solution cost.
- ▶ Sustainability focus grows because of positive and negative drivers.
The IBM Power10 processor delivers new levels of performance compared to IBM POWER9 with 33% lower energy consumption for the same workload in Power E1080 versus IBM Power E980 (Power E980).
- ▶ Resiliency that is driven by accelerated digitization and expectation of continuous access.
The IBM Power family of servers historically offers at least 25% less downtime compared to other high-end servers, which reduces the cost of downtime and improves business results.
- ▶ Security to protect against the growth in scale and frequency of cyberattacks and data breaches.
IBM Power10 cores provide transparent memory encryption, which means all data in memory remains encrypted when in transit between the memory and processor to reduce the risk and cost of potential data breaches and security incidents.
- ▶ Hybrid cloud to mix on-premises with cloud flexibility.
IBM Power10 cores deliver a frictionless experience in extending mission-critical workloads across hybrid cloud because they can offer up to 2.5x more value than a public cloud-only approach.

IBM Power10 processor design

The IBM Power10 processor is built for a cloud environment, whether it is a private cloud, public cloud, or hybrid cloud. One of the improvements in the architecture is an advanced data plane for data-centric workloads, which improves the data bandwidth and capacity. IBM Power10 processors support PCIe generation 5 devices, and IBM Power10 processors add an OMI for better flexibility in supporting memory technologies.

The processor's core architecture is built on 7-nm technology, which improves performance by adding larger caches and reducing latency. For security, it has an enhanced end-to-end encryption capability that provides hardware memory encryption without any performance degradation by adding new crypto engines at the core level. In addition, there are new AI Matrix Math Acceleration engines that are added to each core to improve AI inferencing.

All these features and capabilities leverage energy efficiency for enterprise hybrid cloud with substantial scaling improvements for the largest partitions running mission-critical workloads, such as Oracle DB, SAP HANA, and EPIC healthcare software, by using the same power with less energy consumption.

IBM Power10 processor chip

- ▶ Technology and packaging:
 - 602-mm², 7-nm Samsung (18B devices)
 - Eighteen-layer metal stack, with an enhanced device
 - Single-chip or dual-chip sockets with computational capabilities
 - Up to fifteen SMT8 cores (2 MB L2 cache / core) (Up to 120 simultaneous hardware threads)
 - Up to 120 MB L3 cache (low-latency non-uniform cache access (NUCA) management)
 - Improved energy efficiency relative to IBM POWER9
 - Enterprise thread strength optimizations
 - AI and security focused ISA additions
 - Two times general, 4x matrix single instruction multiple data (SIMD) relative to IBM POWER9
 - EA-tagged L1 cache, with 4x MMU relative to IBM POWER9
- ▶ OMI:
 - Sixteen x8 at up to 32 GTps (1 TBps)
 - Technology-neutral support: near, main, and storage tiers
 - Minimal (< 10 ns latency) add versus DDR direct-attach
- ▶ PowerAXON interface:
 - Sixteen x8 at up to 32 GTps (1 TBps)
 - SMP interconnect for up to 16 sockets
 - OpenCAPI attach for memory, accelerators, and I/O
 - Integrated clustering (memory semantics)
- ▶ PCIe Gen 5 Interface: x64 / DCM at up to 32 GTps

For more information about the IBM Power10 processor and its features, see 3.1.4, “IBM Power10 processor core” on page 53.

3.1.2 IBM Power10 processor-based server packaging

IBM Power10 processor-based servers are designed to create business agility with a flexible and secure hybrid cloud infrastructure, which modernizes applications to maximize value from data. These servers integrate new cloud-native microservices that innovate with existing applications, with a “build once, deploy anywhere” approach for optimized workload placement.

IBM Power10 processor-based servers have an improved secure infrastructure to defend against attacks and protect data by using workload isolation and platform integrity from processor to the cloud. Simplify protection without impacting performance by using transparent memory encryption, and prepare for cryptography advancements, such as Quantum-safe Cryptography and Fully Homomorphic Encryption (FHE).

IBM Power10 processor-based servers offer dynamic agility to adjust to changing business needs seamlessly, and the servers have flexible consumption options with built-in cost optimization.

IBM Power10 processor-based server scale-out servers

IBM Power10 processor-based server scale-out servers provide enhanced performance and scale. The IBM Power10 processor-based server scale-out server family includes the following features:

- ▶ Six new 1- and 2-socket 2U and 4U height server models
- ▶ Up to 48 cores and 8 TB memory footprints
- ▶ Up to 50% performance per price increase and 1.4X more system performance versus IBM POWER9 processor-based servers
- ▶ Expanded Dynamic Capacity consumption features with Capacity Upgrade on Demand (CUoD) and IBM Power Enterprise Pools (PEP) 2.0
- ▶ Value-driven solutions and higher technical standards

Details about each of the models in the following sections:

- ▶ Power S1014 highlights
- ▶ IBM Power S1022s highlights
- ▶ IBM Power S1022 and IBM Power L1022 highlights
- ▶ IBM Power S1024 and IBM Power L1024 highlights

Power S1014 highlights

- ▶ Rack and tower form factors.
- ▶ IBM Power10 processors with 4 or 8 cores per server.
- ▶ Eight DDIMM slots that provide up to 1 TB maximum memory capacity (GA: 512 GB).
- ▶ Main memory encryption for added security.
- ▶ Five PCIe FHHL slots (four are Gen 5-capable). All slots are concurrently maintainable.
- ▶ Up to 16 NVMe U.2 Flash Bays provide up to 102.4 TB of storage.
- ▶ Secure and Trusted Boot with the Trust Platform Module (TPM).
- ▶ Supports external PCIe I/O Expansion Drawers.
- ▶ Supports external SAS Storage Expansion Drawers.
- ▶ Titanium power supplies to meet EU Efficiency Directives: 2x 220 VAC (rack only) or 4x 1200 W 110 VAC with C14 inlet.
- ▶ Enterprise BMC managed (eBMC).

IBM Power S1022s highlights

- ▶ IBM Power10 processors with 4, 8, or 16 total cores per server.
- ▶ One-hop flat CPU interconnect for maximum scalability.
- ▶ Six DDIMM slots that provide up to 2 TB maximum memory capacity (GA: 1 TB).
- ▶ Main memory encryption for added security.
- ▶ Active memory mirroring support to reduce unplanned outages.
- ▶ Ten PCIe HHHL slots (eight are Gen 5-capable). All slots are concurrently maintainable.
- ▶ Up to eight NVMe U.2 Flash Bays provide up to 51.2 TB of storage.
- ▶ Secure and Trusted Boot with TPM.
- ▶ Supports external PCIe I/O Expansion Drawers.
- ▶ Supports external SAS Storage Expansion Drawers.

- ▶ Titanium power supplies to meet EU Efficiency Directives: 2x 220 VAC with C14 inlet.
- ▶ eBMC.

IBM Power S1022 and IBM Power L1022 highlights

- ▶ IBM Power10 processors with 12, 24, 32, or 40 total cores per server.
- ▶ One-hop flat CPU interconnect for maximum scalability.
- ▶ Thirty-two DDIMM slots that provide up to 4 TB maximum memory capacity (GA: 2 TB).
- ▶ Main memory encryption for added security.
- ▶ Active memory mirroring support to reduce unplanned outages.
- ▶ Shared Capacity Utility support.
- ▶ Ten PCIe HHL slots (eight are Gen 5-capable). All slots are concurrently maintainable.
- ▶ Up to eight NVMe U.2 Flash Bays provide up to 51.2 TB of storage.
- ▶ Secure and Trusted Boot with TPM.
- ▶ Supports external PCIe I/O Expansion Drawers.
- ▶ Supports external SAS Storage Expansion Drawers.
- ▶ Titanium power supplies to meet EU Efficiency Directives: 2x 220 VAC with C14 inlet.
- ▶ eBMC.

IBM Power S1024 and IBM Power L1024 highlights

- ▶ IBM Power10 processors with 12, 24, 32, or 48 total cores per server.
- ▶ One-hop flat CPU interconnect for maximum scalability.
- ▶ Thirty-two DDIMM slots that provide up to 8 TB maximum memory capacity (GA: 2 TB).
- ▶ Main memory encryption for added security.
- ▶ Active memory mirroring support to reduce unplanned outages.
- ▶ Shared Capacity Utility support.
- ▶ Ten PCIe FHL slots (eight are Gen 5-capable). All slots are concurrently maintainable.
- ▶ Up to 16 NVMe U.2 Flash Bays provide up to 102.4 TB of storage.
- ▶ Secure and Trusted Boot with TPM.
- ▶ Supports external PCIe I/O Expansion Drawers.
- ▶ Supports external SAS Storage Expansion Drawers.
- ▶ Titanium power supplies to meet EU Efficiency Directives: 4x 220 VAC with C14 inlet.
- ▶ eBMC.

IBM Power10 processor-based enterprise servers

The IBM Power10 processor-based enterprise servers add more scalability and virtualization capabilities to address the most challenging workloads. In addition, they add levels of hardware redundancy so that the system dynamically can recover from hardware errors without an outage. Along with enterprise scalability, the enterprise servers come with an enhanced level of support that is called IBM Power Expert Care. The IBM Power E1050 (Power E1050) provides up to 96 cores and 16 TB of RAM in a rack-mounted format. The Power E1050 can start with one 4U enclosure and dynamically scale to a maximum of four enclosures. The Power E1080 can scale up to 240 SMT8 cores and 64 TB of RAM. The Power E1080 can start with one 2U control unit plus one 5U drawer and expand dynamically to up to a maximum of four 5U drawers.

IBM Power E1050 highlights

- ▶ 4U server with a 19" rack enclosure.
- ▶ An IBM Power10 DCM processor with 12, 18, or 24 cores or sockets delivers up to 96 cores (doubling the Power E950).
- ▶ One-hop flat CPU interconnect for maximum scalability and efficiency.
- ▶ Sixty-four DDIMM slots provide up to 16 TB maximum memory capacity (GA: 8 TB).
- ▶ Main Memory Encryption for added security:
 - Active Memory Mirroring support to reduce unplanned outages.
 - Eleven PCIe slots (eight are Gen 5-capable). All slots are concurrently maintainable.
- ▶ Up to 10 NVMe U.2 Flash Bays provide up to 64 TB of internal storage:
 - Secure and Trusted Boot with TPM.
 - Supports external PCIe I/O Expansion Drawers.
 - Supports external SAS Storage Expansion Drawers.
- ▶ Titanium power supplies to meet EU Efficiency Directives.
- ▶ eBMC:
 - Flexible Consumption with Capacity on Demand (CoD) and PEP 2.0.
 - Built-in IBM PowerVM virtualization.
 - Cloud Management Console.
 - IBM Power Cloud Rewards.
- ▶ Standard 3-year warranty with IBM Power Expert Care.

Power E1080 highlights

- ▶ Follow-on to the Power E980 enterprise server.
- ▶ Modular rack-mounted design scales up to four 5U node drawers + one 2U control unit.
- ▶ Maximum of 240 IBM Power10 SMT8 cores (10, 12, or 15 core offerings in a single-chip module (SCM) package).
- ▶ New 32 Gb SMP Cables (low latency) with Concurrent Maintenance capability.
- ▶ Secure and Trusted Boot with redundant TPM.
- ▶ 2U System Control Unit Drawer.
- ▶ Up to 64 TB total memory (16 TB per drawer).
- ▶ New OMI DDIMMs provide increased memory bandwidth of 409 GBps per socket.
- ▶ Main Memory Encryption for added security.
- ▶ Ports are available for support of Memory Inception/Clustering.
- ▶ Eight PCIe slots per drawer that are Blindswap with Gen 5 support for I/O.
- ▶ Internal storage: Four NVMe Flash 7 mm U.2 Bays per drawer.
- ▶ Up to 16 I/O Expansion Drawers (four Drawers per CEC Drawer).

- ▶ eBMC:
 - Flexible Consumption with CoD and PEP 2.0.
 - Built-in IBM PowerVM virtualization.
 - Cloud Management Console.
 - IBM Power Cloud Rewards.
- ▶ Standard 3-year warranty with IBM Power Expert Care.

3.1.3 IBM Power10 processor

This section provides more specific information about the IBM Power10 processor technology that is used in the Power E1080 scale-up enterprise class server.

The IBM Power10 processor session material that was presented at the 32nd HOT CHIPS conference is available through the HC32 conference proceedings archive at [this web page](#).

IBM Power10 processor overview

The IBM Power10 processor is a superscalar symmetric multiprocessor that is manufactured in complementary metal-oxide-semiconductor (CMOS) 7 nm lithography with 18 layers of metal. The processor contains up to 15 cores that support SMT8 independent execution contexts.

Each core has private access to 2 MB of L2 cache and local access to 8 MB of L3 cache capacity. The local L3 cache region of a specific core is accessible from all other cores on the processor chip. The cores of one IBM Power10 processor share up to 120 MB of latency-optimized NUCA L3 cache.

The processor supports the following three distinct functional interfaces. All of them can run with a signaling rate of up to 32 GTps.⁴

▶ OMI

The IBM Power10 processor has eight memory controller unit (MCU) channels that support one OMI port with two OMI links each.⁵ One OMI link aggregates eight lanes running at 32 GTps speed and connects to one memory buffer-based DDIMM slot to access main memory. Physically, the OMI interface is implemented in two separate die areas of 8 OMI links each. The maximum theoretical full-duplex bandwidth aggregated over all 128 OMI lanes is 1 TBps.

▶ SMP fabric interconnect (PowerAXON)

A total of 144 lanes are available in the IBM Power10 processor to facilitate the connectivity to other processors in an SMP architecture configuration. Each SMP connection requires 18 lanes for eight data lanes plus one spare lane per direction (2 x(8+1)). The processor can support a maximum of eight SMP connections with at total of 128 data lanes per processor. This configuration yields a maximum theoretical full-duplex bandwidth aggregated over all SMP connections of 1 TBps.

The generic nature of the interface implementation enables 128 data lanes to potentially connect accelerator or memory devices through the OpenCAPI protocols. Also, this implementation can support memory cluster and memory interception architectures.

⁴ Giga transfers per second (GTps).

⁵ The OMI links are also referred to as OMI subchannels.

Because of the versatile characteristic of the technology, it is also referred to as the *PowerAXON* interface (Power A-bus/X-bus/OpenCAPI/Networking).⁶ At the time of writing, the OpenCAPI and the memory clustering and memory interception use cases are not used by available technology products.

► PCIe 5.0 interface

To support external I/O connectivity and access to internal storage devices, the IBM Power10 processor provides differential Peripheral Component Interconnect Express 5.0 interface buses (PCIe Gen 5) with a total of 32 lanes. The lanes are grouped in two sets of 16 lanes that can be used in one of the following configurations:

- One x16 PCIe Gen 4
- Two x8 PCIe Gen 4
- One x8, 2 x4 PCIe Gen 4
- One x8 PCIe Gen 5, 1 x8 PCIe Gen 4
- One x8 PCIe Gen 5, 2 x4 PCIe Gen 4

Figure 3-1 shows the IBM Power10 processor die with several functional units labeled. Sixteen SMT8 processor cores are shown, but only 10-, 12-, or 15-core processor options are available for Power E1080 server configurations.

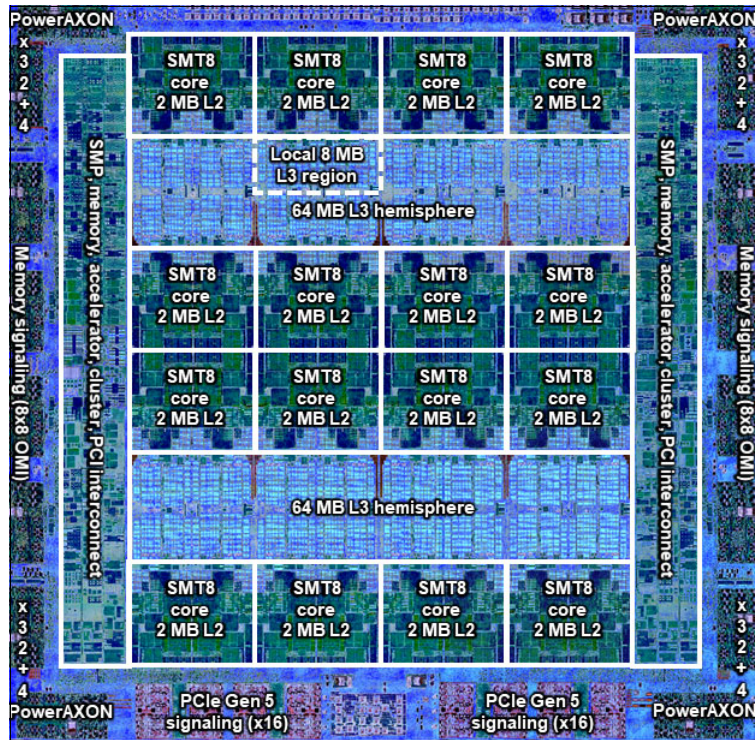


Figure 3-1 The IBM Power10 processor chip (die photo courtesy of Samsung Foundry)

⁶ A-buses and X-buses provide SMP fabric ports that are used between CEC drawers or within CEC drawers respectively.

Important IBM Power10 processor characteristics are listed in Table 3-1.

Table 3-1 Summary of the IBM Power10 processor chip and processor core technology

Technology	IBM Power10 processor
Processor die size	602 mm ²
Fabrication technology	<ul style="list-style-type: none"> ▶ CMOS 7-nm lithography ▶ Eighteen layers of metal
Maximum processor cores per chip	15
Maximum execution threads per core / chip	8 / 120
Maximum L2 cache core	2 MB
Maximum On-chip L3 cache per core / chip	8 MB / 120 MB
Number of transistors	18 billion
Processor compatibility modes	Support for IBM Power ISA of IBM POWER8® and POWER9 processors

The IBM Power10 processor is packaged as an SCM for exclusive use in the Power E1080 servers. The SCM contains the IBM Power10 processor plus more logic that is needed to facilitate power supply and external connectivity to the chip. It also holds the connectors to plug SMP cables directly onto the socket to build 2-, 3-, and 4-node Power E1080 servers.

Figure 3-2 shows the logical diagram of the IBM Power10 processor-based server SCM.

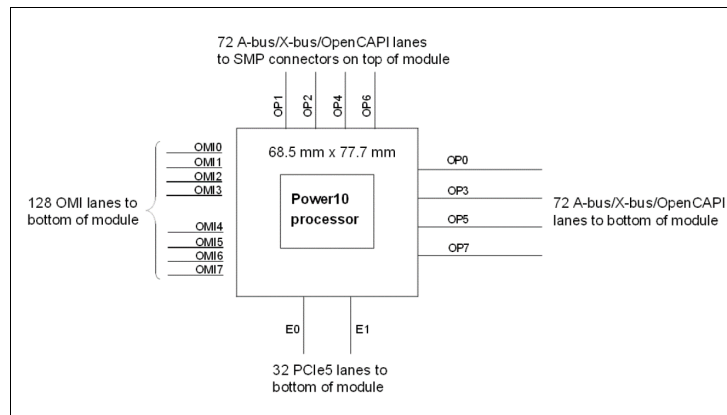


Figure 3-2 IBM Power10 processor-based server SCM logical diagram

As indicated in Figure 3-2, the PowerAXON interface lanes are grouped in two sets of 72 lanes each. One set provides four interface ports (OP1, OP2, OP4, and OP6), which are accessible to SMP connectors that are physically placed on the top of the SCM module.

The second set of ports (OP0, OP3, OP5, and OP7) are used to implement the fully connected SMP fabric between the four sockets within a system node. Eight OMI ports (OMI0 - OMI7) with two OMI links each provide access to the buffered main memory DDIMMs. The 32 PCIe Gen 5 lanes are grouped into two PCIe host bridges (E0 and E1).

Figure 3-3 shows a physical diagram of the IBM Power10 processor-based server SCM. The eight SMP connectors (OP1A, OP1B, OP2A, OP2B, OP4A, OP4B, OP6A, and OP6B) externalize four SMP buses, which are used to connect system node drawers in 2-, 3-, and 4-node Power E1080 configurations. The OpenCAPI connectivity options are indicated, although at the time of writing they are not used by any commercially available product.

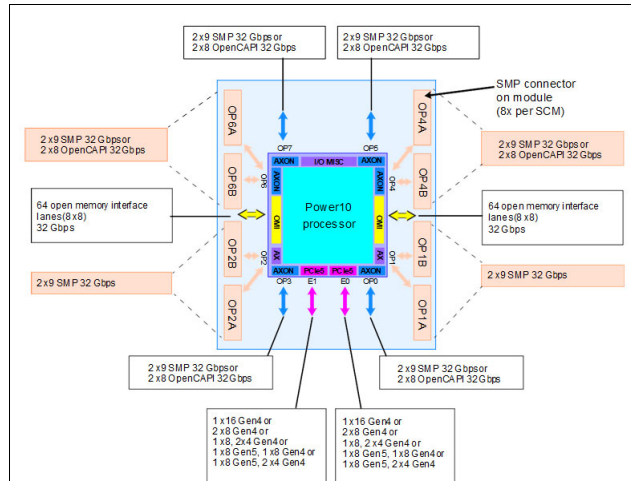


Figure 3-3 IBM Power10 processor-based server single-chip module

3.1.4 IBM Power10 processor core

The IBM Power10 processor core inherits the modular architecture of the IBM POWER9 processor core, but the redesigned and enhanced micro-architecture increases the processor core performance and processing efficiency. The peak computational throughput is markedly improved by new execution capabilities and optimized cache bandwidth characteristics. Extra matrix math acceleration engines can deliver performance gains for machine learning, particularly for AI-inferencing workloads.

The Power E1080 server uses the IBM Power10 enterprise-class processor variant in which each core can run with up to eight independent hardware threads. If all threads are active, the mode of operation is referred to as SMT8 mode. An IBM Power10 core with SMT8 capability is a Power10 SMT8 core or SMT8 core for short. The IBM Power10 core also supports modes with four active threads (SMT4), two active threads (SMT2), and one single active thread (single-threaded (ST)).

The SMT8 core includes two execution resource domains. Each domain provides the functional units to service up to four hardware threads. Figure 3-4 shows the functional units of an SMT8 core where all eight threads are active. The two execution resource domains are highlighted with colored backgrounds in two different shades of blue.

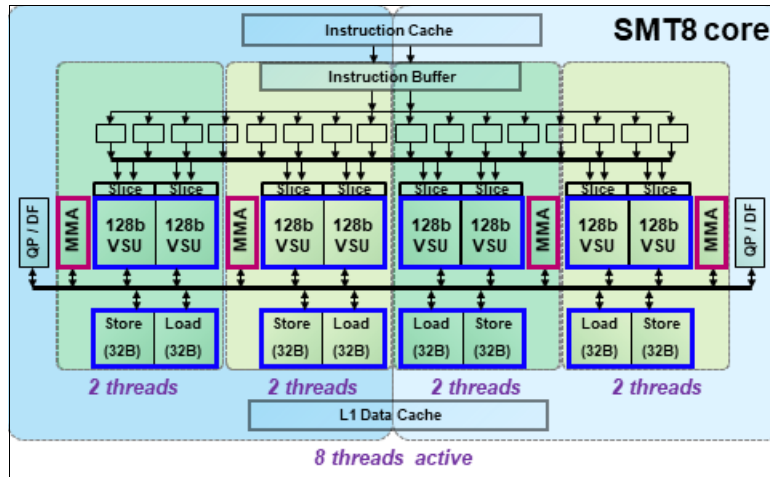


Figure 3-4 IBM Power10 SMT8 core

Each of the two execution resource domains supports 1 - 4 threads and include four vector scalar units (VSUs) of 128-bit width, two Matrix Math Accelerator (MMA) accelerators, and one quad-precision floating-point (QP) and decimal floating-point (DFP) unit.

One VSU and the directly associated logic are called an execution *slice*. Two neighboring slices can be used as a combined execution resource that is named a *super-slice*. When operating in SMT8 mode, eight SMT threads are subdivided in pairs that collectively run on two adjacent slices as indicated through colored backgrounds in different shades of green.

In SMT4 or lower thread modes, 1 - 2 threads each share a four-slice resource domain. Figure 3-4 indicates other essential resources that are shared among the SMT threads, such as an instruction cache, an instruction buffer, and an L1 data cache.

The SMT8 core supports automatic workload balancing to change the operational SMT thread level. Depending on the workload characteristics, the number of threads that is running on one chiplet can be reduced from four to two and even further to only one active thread. An individual thread can benefit in terms of performance if fewer threads run against the core's execution resources.

Micro-architecture performance and efficiency optimization lead to an improvement of the performance per watt signature compared with the previous IBM POWER9 core implementation. The overall energy efficiency is better by a factor of approximately 2.6, which demonstrates the advancement in processor design that is manifested by the IBM Power10 core.

The IBM Power10 processor core includes the following key features and improvements that affect performance:

- ▶ Enhanced load and store bandwidth
- ▶ Deeper and wider instruction windows
- ▶ Enhanced data prefetch
- ▶ Branch execution and prediction enhancements
- ▶ Instruction fusion

Enhancements in the area of computation resources, working set size, and data access latency are described in the following sections. The change in relation to the IBM POWER9 processor core implementation is provided in parentheses.

Enhanced computation resources

Here are the major computational resource enhancements:

- ▶ Eight VSU execution slices, each supporting 64-bit scalar or 128-bit SIMD +100% for permute, fixed-point, floating-point, and crypto (Advanced Encryption Standard (AES) or Secure Hash Algorithm (SHA)) +400% operations.
- ▶ Four units for MMA acceleration, each capable of producing a 512-bit result per cycle (new) +400% single and double precision FLOPS, and support for reduced precision AI acceleration).
- ▶ Two units for QP and DFP operations (more instruction types).

Larger working sets

The following major changes were implemented in working set sizes:

- ▶ L1 instruction cache: Two 48 KB 6-way (96 KB total) (+50%)
- ▶ L2 cache: 2 MB 8-way (+400%)
- ▶ L2 translation lookaside buffer (TLB): Two 4 K entries (8 K total) (+400%)

Data access with reduced latencies

The following major changes reduce latency for load data:

- ▶ L1 data cache access at four cycles nominal with zero penalty for store-forwarding (- 2 cycles)
- ▶ L2 data access at 13.5 cycles nominal (-2 cycles)
- ▶ L3 data access at 27.5 cycles nominal (-8 cycles)
- ▶ TLB access at 8.5 cycles nominal for effective-to-real address translation (ERAT) miss, including for nested translation (-7 cycles)

Micro-architectural innovations that complement physical and logic design techniques and specifically address energy efficiency include the following examples:

- ▶ Improved clock-gating.
- ▶ Reduced flush rates with improved branch prediction accuracy.
- ▶ Fusion and gather operating merging.
- ▶ Reduced number of ports and reduced access to selected structures.
- ▶ Effective address (EA)-tagged L1 data and instruction cache yields ERAT access only on a cache miss.

In addition to improvements in performance and energy efficiency, security represents a major architectural focus area. The IBM Power10 processor core supports the following security features:

- ▶ Enhanced hardware support that provides improved performance while mitigating speculation-based attacks.
- ▶ Dynamic Execution Control Register (DEXCR) support.
- ▶ Return-oriented programming (ROP) protection.

3.1.5 Simultaneous multi-threading

Each core of the IBM Power10 processor supports multiple hardware threads that represent independent execution contexts. If only one hardware thread is used, the processor core runs in ST mode.

If more than one hardware thread is active, the processor runs in SMT mode. In addition to the ST mode, the IBM Power10 processor supports the following different SMT modes:

- ▶ SMT2: Two hardware threads active
- ▶ SMT4: Four hardware threads active
- ▶ SMT8: Eight hardware threads active

SMT enables a single physical processor core to simultaneously dispatch instructions from more than one hardware thread context. Computational workloads can use the processor core's execution units with a higher degree of parallelism, which enhances the throughput and scalability of multi-threaded applications and optimizes the compute density for ST workloads.

SMT is primarily beneficial in commercial environments where the speed of an individual transaction is not as critical as the total number of transactions that are performed. SMT typically increases the throughput of most workloads, especially those workloads with large or frequently changing working sets, such as database servers and web servers.

Table 3-2 lists a historic account of the SMT capabilities that are supported by each implementation of the IBM Power Architecture since IBM POWER4.

Table 3-2 SMT levels that are supported by IBM POWER processors

Technology	Cores per system	Supported hardware threading modes	Maximum hardware threads per partition
IBM POWER4	32	ST	32
IBM POWER5	64	ST and SMT2	128
IBM POWER6	64	ST and SMT2	128
IBM POWER7	256	ST, SMT2, and SMT4	1024
IBM POWER8	192	ST, SMT2, SMT4, and SMT8	1536
IBM POWER9	192	ST, SMT2, SMT4, and SMT8	1536
IBM Power10	240	ST, SMT2, SMT4, and SMT8	1920

3.1.6 Matrix Math Accelerator AI workload acceleration

The MMA facility was introduced by the IBM Power Instruction Set Architecture (ISA) 3.1. The related instructions implement numerical linear algebra operations on small matrices that are meant to accelerate computation-intensive kernels, such as matrix multiplication, convolution, and discrete Fourier transform.

To efficiently accelerate MMA operations, the IBM Power10 processor core implements a *dense math engine* (DME) microarchitecture that effectively provides an accelerator for cognitive computing, machine learning, and AI inferencing workloads.

The DME encapsulates compute efficient pipelines, a physical register file, and associated data flow that keeps the resulting accumulator data local to the compute units. Each MMA pipeline performs outer-product matrix operations, reading from and writing back a 512-bit accumulator register.

Figure 3-5 shows an MMA diagram.

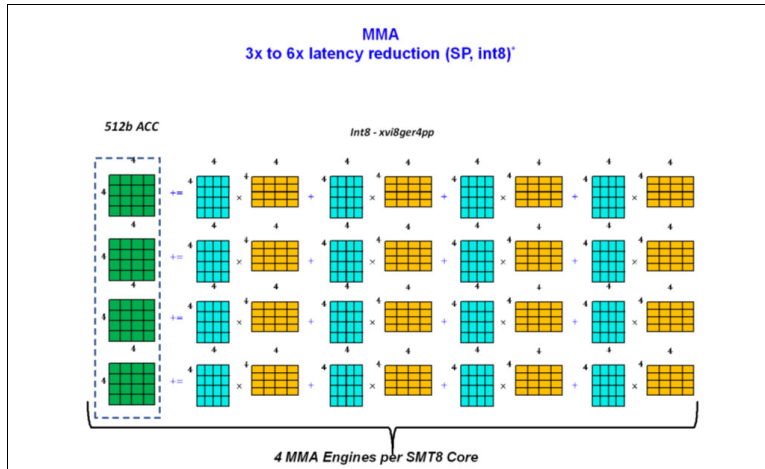


Figure 3-5 MMA engines per SMT8 core

An IBM Power10 processor-based server implements the MMA accumulator architecture without adding an architected state. Each architected 512-bit accumulator register is backed by four 128-bit Vector Scalar eXtension (VSX) registers. Figure 3-6 shows a complete distribution from the OMI to the MMA accumulators passing through the cache levels.

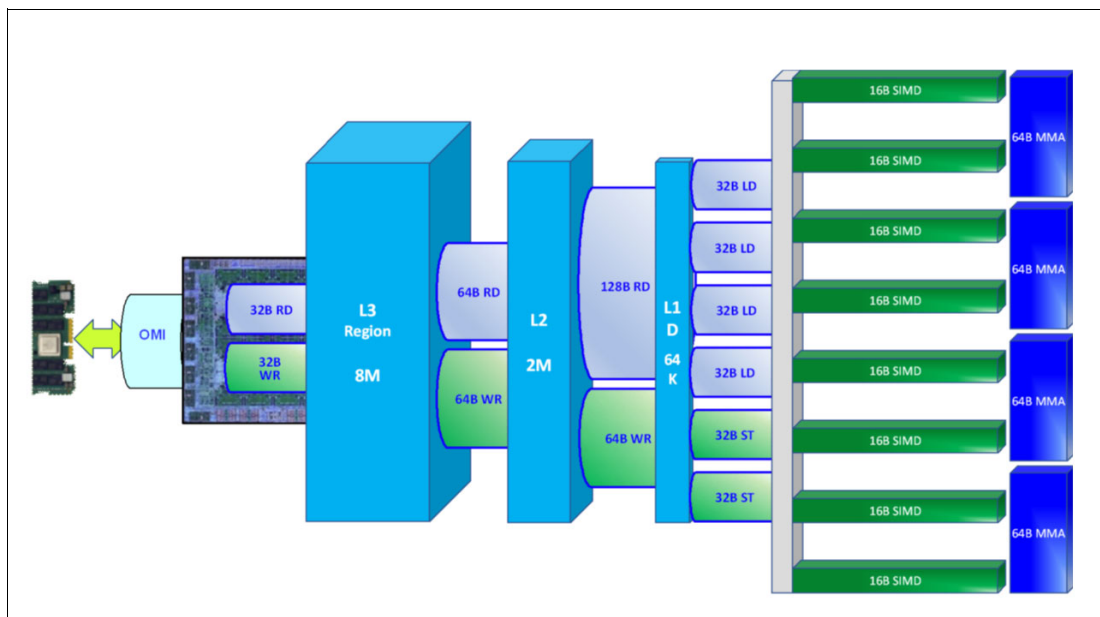


Figure 3-6 Open Memory Interface to MMA

To leverage the performance advantages of the MMA capabilities in the IBM Power10 processor, use libraries that are built for IBM Power10 processor-based server MMA. For AI inferencing, look at the OpenBLAS library, which is used by frameworks such as PyTorch, TensorFlow, and ONNX Runtime. Section 6.1, “Artificial intelligence inferencing with Red Hat OpenShift and IBM Power10 processor-based servers” on page 174 describes how to build an application that includes versions of these libraries that are optimized for MMA.

For more information about the implementation of the IBM Power10 processor’s high-throughput math engine, see [A matrix math facility for Power ISA processors](#).

For more information about fundamental MMA architecture principles with detailed instruction set usage, register file management concepts, and various supporting facilities, see *Matrix-Multiply Assist Best Practices Guide*, REDP-5612.

3.1.7 On-chip L3 cache and intelligent caching

The IBM Power10 processor includes a large on-chip L3 cache of up to 120 MB with a NUCA architecture that provides mechanisms to distribute and share cache footprints across a set of L3 cache regions. Each processor core can access an associated local 8 MB of L3 cache and the data in the other L3 cache regions on the chip and throughout the system.

Each L3 region serves as a victim cache for its associated L2 cache. The L3 region also can provide aggregate storage for the on-chip cache footprint.

With intelligent L3 cache management, the IBM Power10 processor can optimize the access to L3 cache lines and minimize cache latencies. The L3 cache includes a replacement algorithm with data type and reuse awareness. The cache also supports an array of prefetch requests from the core, including instruction and data, and works cooperatively with the core, memory controller, and SMP interconnection fabric to manage prefetch traffic, which optimizes system throughput and data latency.

The L3 cache supports the following key features:

- ▶ Enhanced bandwidth that supports up to 64 bytes per core processor cycle to each SMT8 core.
- ▶ Enhanced data prefetch that is enabled by 96 L3 prefetch request machines that service prefetch requests to memory for each SMT8 core.
- ▶ Plus-one prefetching at the memory controller for enhanced effective prefetch depth and rate.
- ▶ IBM Power10 processor-based server software prefetch modes that support fetching blocks of data into the L3 cache.
- ▶ Data access with reduced latencies.

3.1.8 Open Memory Interface

The IBM Power10 processor introduces a new and innovative OMI. The OMI is driven by eight on-chip MCUs, and it is implemented in two separate physical building blocks that lay in opposite areas at the outer edge of the IBM Power10 die. Each area supports 64 OMI lanes that are grouped in four ports. Each port consists of two links with eight lanes each, which operate in a latency-optimized manner with unprecedented bandwidth and scale at 32 Gbps speed.

The aggregated maximum theoretical full-duplex bandwidth of the OMI interface culminates at $2 \times 512 \text{ GBps} = 1 \text{ TBps}$ per IBM Power10 processor-based server SCM.

The OMI physical interface enables low-latency, high-bandwidth, and technology-neutral host memory semantics to the processor and allows attaching established and emerging memory elements. With the Power E1080 server, OMI initially supports one main-tier, low-latency, and enterprise-grade Double Data Rate 4 (DDR4) DDIMM per OMI link. This configuration yields a total memory capacity of 16 DDIMMs per SCM and 64 DDIMMs per Power E1080 server node. The memory bandwidth depends on the DDIMM density that is configured for a Power E1080 server.

The maximum theoretical duplex memory bandwidth is 409 GBps per SCM if 32 GB or 64 GB DDIMMs running at 3200 MHz are used. The maximum memory bandwidth is reduced to 375 GBps per SCM if 128 GB or 256 GB DDIMMs running at 2933 MHz are used.

In summary, the IBM Power10 processor-based server SCM supports 128 OMI lanes with the following characteristics:

- ▶ 32 Gbps signaling rate
- ▶ Eight lanes per OMI link
- ▶ Two OMI links per OMI port (2 x 8 lanes)
- ▶ Eight OMI ports per SCM (16 x 8 lanes)

3.1.9 Pervasive memory encryption

The IBM Power10 processor-based server MCU provides the system memory interface between the on-chip SMP interconnect fabric and the OMI links. This design qualifies the MCU as ideal functional unit to implement memory encryption logic. The IBM Power10 processor-based server on-chip MCU encrypts and decrypts all traffic to and from system memory that is based on the AES technology.

The IBM Power10 processor supports the following modes of operation:

- ▶ AES XTS mode

XTS is the xor-encrypt-xor based tweaked-codebook mode with ciphertext stealing. AES XTS provides a block cipher with strong encryption, which is useful for encrypting persistent memory.

Persistent DIMM technology retains the data that is stored inside the memory DIMMs, even if the power is turned off. A malicious attacker who gains physical access to the DIMMs can steal memory cards. The data that is stored in the DIMMs can leave the data center in the clear if not encrypted.

Also, memory cards that leave the data center for repair or replacement can be a potential security breach. Because the attacker might have arbitrary access to the persistent DIMM data, the stronger encryption of the AES XTS mode is required for persistent memory. The AES XTS mode of the IBM Power10 processor is supported if persistent memory solutions become available for IBM Power servers.

- ▶ AES CTR mode

CTR is the *counter* mode of operation that designates a low-latency AES block cipher. Although the level of encrypting is not as strong as with the XTS mode, the low-latency characteristics make it the preferred mode for memory encryption of volatile memory. AES CTR makes it more difficult to physically gain access to data through the memory card interfaces. The goal is to protect against physical attacks, which becomes increasingly important in the context of cloud deployments.

The Power E1080 servers support the AES CTR mode for pervasive memory encryption. Each IBM Power10 processor holds a 128-bit encryption key that is used by the processor's MCU to encrypt the data of the DDIMMs that are attached to the OMI links.

The MCU crypto engine is integrated into the data path, which ensures that the data fetch and store bandwidth are not compromised by the AES CTR encryption mode. Because the encryption has no noticeable performance effect and because of the obvious security benefit, the pervasive memory encryption is enabled by default and cannot be turned off through any administrative interface.

Note: The pervasive memory encryption of the IBM Power10 processor does not affect the encryption status of a system dump content. All data that is coming from the DDIMMs is decrypted by the MCU before it is passed onto the dump devices under the control of the dump program code. This statement applies to the traditional system dump under the operating system control and the firmware assist dump utility.

3.1.10 Nest accelerator

The IBM Power10 processor has an on-chip accelerator that is called the nest accelerator (NX) unit. The coprocessor features that are available on the IBM Power10 processor are similar to the features on the IBM POWER9 processor. These coprocessors provide specialized functions, such as the following examples:

- ▶ IBM proprietary data compression and decompression
- ▶ Industry-standard gzip compression and decompression
- ▶ AES and SHA cryptography
- ▶ Random number generation

Figure 3-7 shows a block diagram of the NX unit.

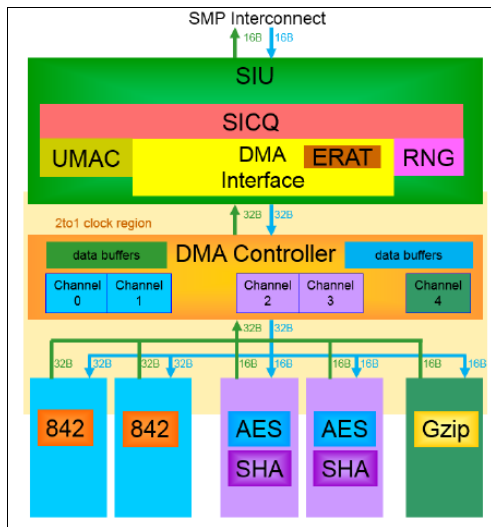


Figure 3-7 Block diagram of the NX unit

Each one of the AES or SHA engines, data compression, and gzip units consist of a coprocessor type. The NX unit features three coprocessor types. The NX unit also includes more support hardware to support coprocessor invocation by user code, usage of EAs, high-bandwidth storage accesses, and interrupt notification of job completion.

The direct memory access (DMA) controller of the NX unit helps to start the coprocessors and move data on behalf of coprocessors. SMP interconnect unit (SIU) provides the interface between the IBM Power10 processor-based server SMP interconnect and the DMA controller.

The NX coprocessors can be started transparently through library or operating system kernel calls to speed up operations that are related to data compression, Live Partition Mobility (LPM) migration, IPsec, JFS2 encrypted file systems, PKCS11 encryption, random number generation, and the most recently announced logical volume encryption.

In effect, this on-chip NX unit on IBM Power10 processor-based servers implements a high-throughput engine that can perform the equivalent work of multiple cores. The system performance can benefit by offloading these expensive operations to on-chip accelerators, which can reduce the CPU usage and improve the performance of applications.

The accelerators are shared among the logical partitions (LPARs) under the control of the IBM PowerVM hypervisor and accessed by using a hypervisor call. The operating system, along with the IBM PowerVM hypervisor, provides a send address space that is unique per process requesting the coprocessor access. This configuration allows the user process to directly post entries to the first in - first out (FIFO) queues that are associated with the NX accelerators. Each NX coprocessor type has a unique receive address space corresponding to a unique FIFO for each of the accelerators.

For more information about the use of the xgzip tool that uses the gzip accelerator engine, see the following resources:

- ▶ [Using the POWER9 NX \(gzip\) accelerator in AIX](#)
- ▶ [POWER9 GZIP Data Acceleration with IBM AIX](#)
- ▶ [Performance improvement in OpenSSH with on-chip data compression accelerator in POWER9](#)
- ▶ [nxstat Command](#)

3.1.11 SMP interconnect and accelerator interface

The IBM Power10 processor provides a highly optimized, 32 Gbps differential signaling technology interface that is structured in 16 entities. Each entity consists of eight data lanes and one spare lane. This interface can facilitate the following functional purposes:

- ▶ First- or second-tier SMP link interface enabling up to 16 IBM Power10 processors to be combined into a large, robustly scalable, and single-system image.
- ▶ Open Coherent Accelerator Processor Interface (OpenCAPI) to attach cache coherent and I/O-coherent computational accelerators, load/store addressable host memory devices, low latency network controllers, and intelligent storage controllers.
- ▶ Host-to-host integrated memory clustering interconnect enabling multiple IBM Power10 processor-based servers to directly use memory throughout the cluster.

Note: The OpenCAPI interface and the memory clustering interconnect are IBM Power10 technology options for future use.

Because of the versatile nature of signaling technology, the 32-Gbps interface also is referred to as the PowerAXON interface. The IBM proprietary X-bus links connect two processors on a board with a common reference clock. The IBM proprietary A-bus links connect two processors in different drawers on different reference clocks by using a cable.

OpenCAPI is an open interface architecture that allows any microprocessor to attach to the following items:

- ▶ Coherent user-level accelerators and I/O devices
- ▶ Advanced memories accessible through read/write or user-level DMA semantics

The OpenCAPI technology is developed, enabled, and standardized by the OpenCAPI Consortium. For more information about the consortium's mission and the OpenCAPI protocol specification, see [OpenCAPI Consortium](#).

The PowerAXON interface is implemented on dedicated areas that are at each corner of the IBM Power10 processor die. The Power E1080 server uses this interface to implement single-drawer chip-to-chip and drawer-to-drawer chip interconnects.

The Power E1080 single-drawer chip-to-chip SMP interconnect features the following properties:

- ▶ Three (2 x 9)-bit on system board buses per IBM Power10 processor-based server SCM.
- ▶ Eight data lanes, plus one spare lane in each direction per chip-to-chip connection.
- ▶ 32 Gbps signaling rate providing 128 GBps per chip-to-chip SMP connection bandwidth, which is an increase of 33% compared to the Power E980 single-drawer implementation.
- ▶ 4-way SMP architecture implementations build-out of four IBM Power10 processor-based server SCMs per drawer in a 1-hop topology.

The Power E1080 drawer-to-drawer SMP interconnect features the following properties:

- ▶ Three (2 x 9)-bit buses per IBM Power10 processor-based server SCM.
- ▶ Eight data lanes plus one spare lane in each direction per chip-to-chip connection.
- ▶ Each of the four SCMs in a drawer is connected directly to an SCM at the same position in every other drawer in a multi-node system.
- ▶ 32 Gbps signaling rate, which provides 128 GBps per chip-to-chip inter-node SMP connection bandwidth.
- ▶ 8-socket, 12-socket, and 16-socket SMP configuration options in a 2-hop topology.

Figure 3-8 shows the SMP connections for a fully configured 4-node, 16-socket Power E1080 server. The blue lines represent the chip-to-chip connections within one system node. The green lines represent the drawer-to-drawer SMP connections.

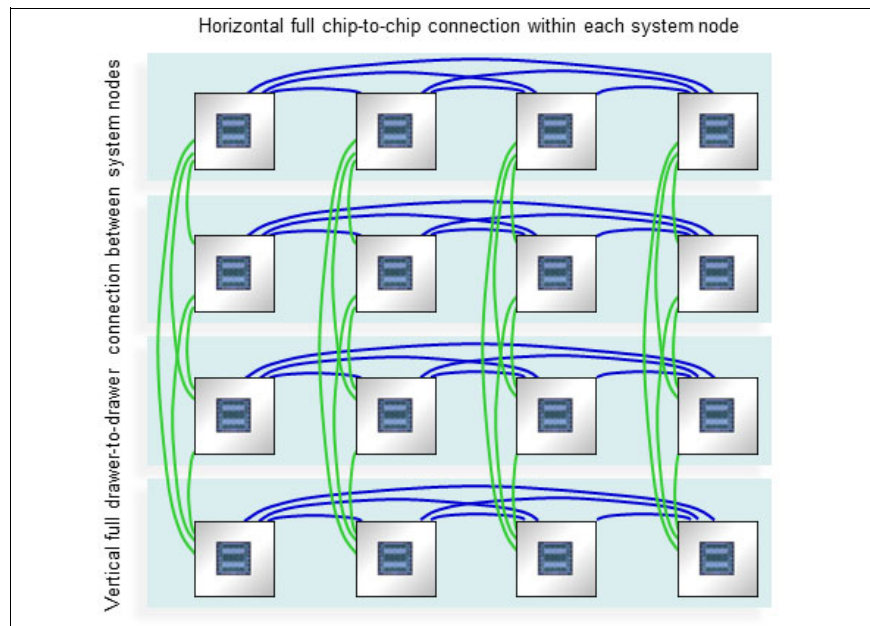


Figure 3-8 SMP interconnect in a 4-node, 16-socket Power E1080 server

From the drawing that is shown in Figure 3-8 on page 62, you can deduce that each socket is directly connected to any other socket within one system node and only one intermediary socket is required to get from a chip to any other chip in another CEC drawer.

3.1.12 IBM Power and performance management

IBM Power10 processor-based servers implement an enhanced version of the power management EnergyScale technology.

As in the previous IBM POWER9 EnergyScale implementation, the IBM Power10 processor-based server EnergyScale technology supports dynamic processor frequency changes that depend on several factors, such as workload characteristics, the number of active cores, and environmental conditions.

Based on the extensive experience that was gained over the past few years, the IBM Power10 processor-based server EnergyScale technology evolved to use the following effective and simplified set of operational modes:

- ▶ Power-saving mode
- ▶ Static mode (nominal frequency)
- ▶ Maximum performance mode (MPM)

The IBM POWER9 dynamic performance mode (DPM) has many features in common with the IBM POWER9 MPM. Because of this redundant nature of characteristics, the DPM for IBM Power10 processor-based systems was removed in favor of an enhanced MPM. For example, the maximum frequency is now achievable in the IBM Power10 processor-based server enhanced MPM (regardless of the number of active cores), which was not always the case with IBM POWER9 processor-based servers.

In the Power E1080, MPM is enabled by default. This mode dynamically adjusts the processor frequency to maximize performance and enable a higher processor frequency range. Each of the power saver modes delivers consistent system performance without any variation if the nominal operating environment limits are met.

For IBM Power10 processor-based servers that are under control of the IBM PowerVM hypervisor, the MPM is a system-wide configuration setting, but each processor module frequency is optimized separately.

The following factors determine the maximum frequency at which a processor module can run:

- ▶ Processor utilization: Lighter workloads run at higher frequencies.
- ▶ Number of active cores: Fewer active cores run at higher frequencies.
- ▶ Environmental conditions: At lower ambient temperatures, cores can run at higher frequencies.

The following IBM Power10 processor-based server EnergyScale modes are available:

▶ Power-saving mode

The frequency is set to the minimum frequency to reduce energy consumption. Enabling this feature reduces power consumption by lowering the processor clock frequency and voltage to fixed values. This configuration reduces power consumption of the system while delivering predictable performance.

▶ Static mode

The frequency is set to a fixed point that can be maintained with all normal workloads and in all normal environmental conditions. This frequency is also referred to as *nominal frequency*.

▶ MPM

Workloads run at the highest frequency possible, depending on workload, active core count, and environmental conditions. The frequency does not go below the static frequency for all normal workloads and in all normal environmental conditions.

In MPM, the workload is run at the highest frequency possible. The higher power draw enables the processor modules to run in an MPM typical frequency range (MTFR), where the lower limit is above the nominal frequency and the upper limit is provided by the system's maximum frequency.

The MTFR is published as part of the system specifications of a specific IBM Power10 processor-based server if it is running by default in MPM. The higher power draw potentially increases the fan speed of the respective system node to meet the higher cooling requirements, which causes a higher noise emission level of up to 15 decibels.

The processor frequency typically stays within the limits that are set by the MTFR, but can be lowered to frequencies between the MTFR lower limit and the nominal frequency at high ambient temperatures above 27 °C (80.6 °F). If the data center ambient environment is less than 27 °C, the frequency in MPM consistently is in the upper range of the MTFR (roughly 10% - 20% better than nominal). At lower ambient temperatures (below 27 °C, or 80.6 °F), MPM mode also provides deterministic performance. As the ambient temperature increases above 27 °C, determinism can no longer be ensured.

This mode is the default mode in Power E1080.

▶ Idle power saver (IPS) mode

IPS mode lowers the frequency to the minimum if the entire server (all cores of all sockets) is idle. It can be enabled or disabled separately from all other modes. Power E1080 does not support this mode.

Figure 3-9 shows the comparative frequency ranges for the IBM Power10 processor-based server power-saving mode, static or nominal mode, and the MPM. The frequency adjustments for different workload characteristics, ambient conditions, and idle states are also indicated.

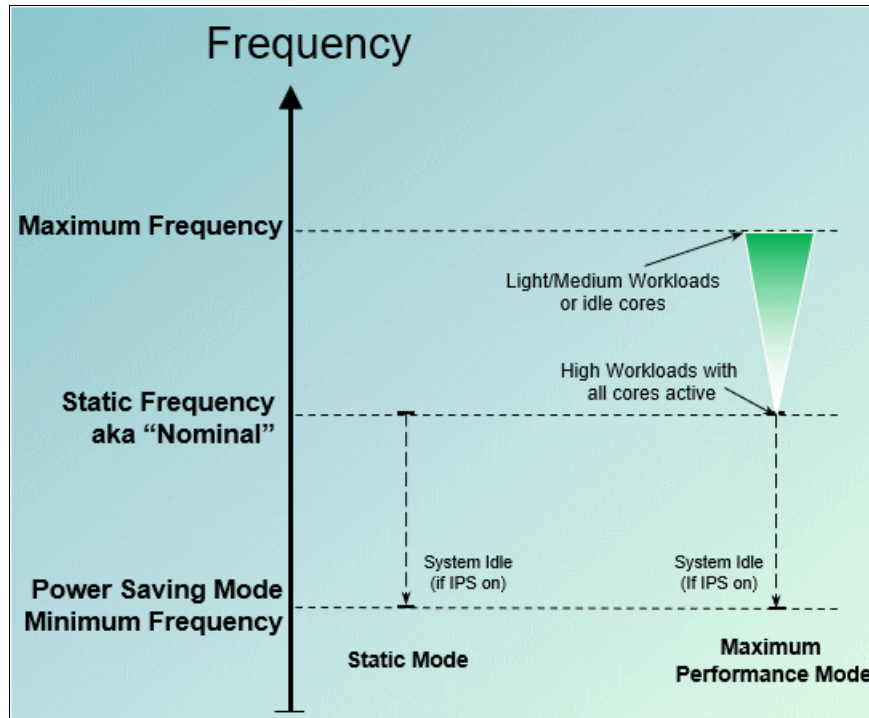


Figure 3-9 IBM Power10 processor-based server power management modes and related frequency ranges

Table 3-3 shows the power-saving mode and the static mode frequencies, and the frequency ranges of the MPM for all three processor module types that are available for the Power E1080 server.

Table 3-3 Characteristic frequencies and frequency ranges for the Power E1080 server

Feature Code	Cores per single-chip module	Power-saving mode frequency (GHz)	Static mode frequency (GHz)	Maximum performance mode frequency range (GHz)
EDP2	10	3.25	3.65	3.65 - 3.90 GHz (max)
EDP3	12	3.40	3.60	3.60 - 4.15 GHz (max)
EDP4	15	3.25	3.55	3.55 - 4.00 GHz (max)

For Power E1080 servers, the MPM is enabled by default.

The controls for all power-saver modes are available on the Advanced System Management Interface (ASMI) and can be dynamically modified. A system administrator can use the Hardware Management Console (HMC) to set power-saver mode or to enable static mode or MPM.

Figure 3-10 shows the **ASM interface** menu for Power and Performance Mode Setup on a Power E1080 server.

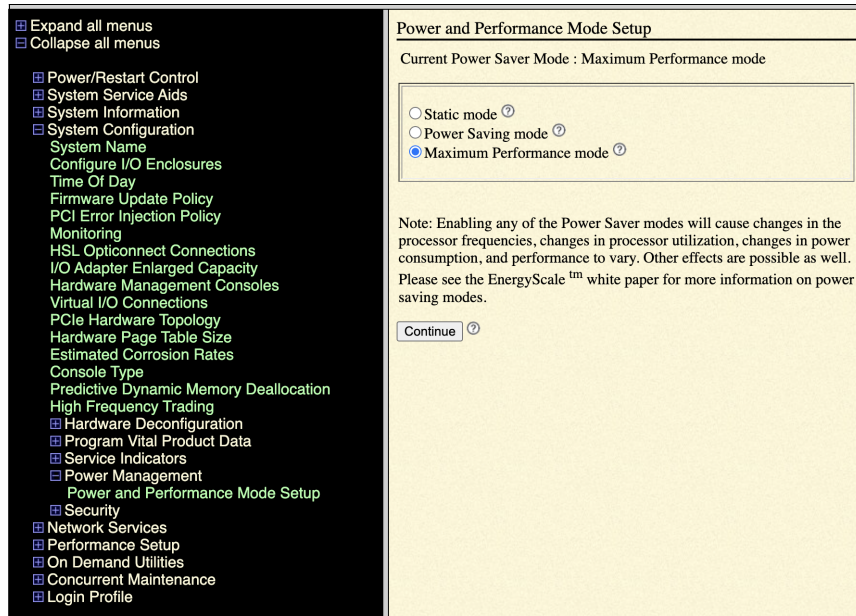


Figure 3-10 Power E1080 ASMI menu for Power and Performance Mode setup

Figure 3-11 shows the **HMC** menu for Power and Performance Mode Setup.

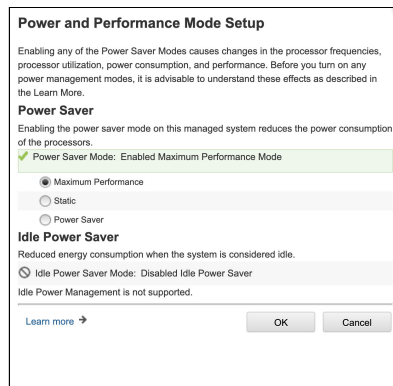


Figure 3-11 Power E1080 HMC menu for Power and Performance Mode Setup

3.2 IBM Power Systems Virtual Server

IBM PowerVS is a hybrid cloud offering running on IBM Power servers in IBM data centers around the world. This offering allows you to run IBM AIX®, Linux, and IBM i workloads in a shared infrastructure with direct-connected storage that is managed by IBM. There are high-speed interconnects between IBM PowerVS data centers and other IBM Cloud data centers that allow you to host all of cloud workloads with minimal latency between applications running on IBM Power servers and applications running on x86 servers.

Because IBM PowerVS runs on IBM POWER processors, it integrates seamlessly with workloads that you are running on-premises, so you an entry point to moving your existing workloads to the cloud if needed. With IBM PowerVS, you get fast, self-service provisioning, flexible management both on-premises and off-premises, and like on-premises, IBM PowerVS can be connected to access a stack of enterprise services from IBM. All these features use pay-as-you-use billing that lets you scale up and out. You can quickly deploy an IBM PowerVS infrastructure to meet your specific business needs and control workload demands.

3.2.1 Architecture

IBM PowerVS machines are in IBM data centers, and they can be provisioned by using an IBM Cloud account. The IBM PowerVS servers are distinct from the IBM Cloud servers because they use separate networks and direct-attached storage, and they can run either the AIX, IBM i, or Linux operating systems. The IBM PowerVS internal networks are fenced but offer connectivity options to IBM Cloud infrastructure or on-premises environments. The virtual servers run on IBM Power hardware with the IBM PowerVM hypervisor. By using an IBM Cloud account, IBM PowerVS, also known as an LPAR, can be deployed easily and quickly. On IBM Cloud, the IBM PowerVS workspace acts as a container for all IBM PowerVS instances at a specific geographic region.

On IBM Cloud, the Identity and Access Management (IAM) service can authenticate users securely, control access to IBM PowerVS resources with resource groups, and allow access to specific resources for a set of users with access groups. IBM PowerVS requires more access for IBM Cloud features, such as Direct Link, Transit Gateway service, and Virtual Private Cloud (VPC).

Compute resources and operating systems

IBM PowerVS provides access to infrastructure and physical computing resources without the need to manage or operate them. Networking, storage, servers, and virtualization are managed by the IBM Cloud team, but the operating system, middleware, run time, and software applications and data are managed by the client, as shown in Figure 3-12.

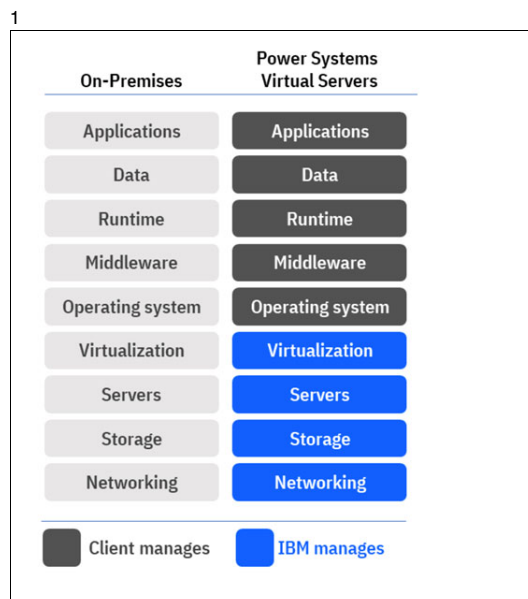


Figure 3-12 IBM PowerVS RACI matrix⁷

⁷ <https://cloud.ibm.com/docs/power-iaas?topic=power-iaas-about-virtual-server>

IBM provides the stock AIX and IBM i images during IBM PowerVS provisioning. The clients can bring their own custom AIX, IBM i, or Linux image. IBM PowerVS does not provide Linux stock images, and the only option is to bring your own Linux image and subscription (SUSE Linux Enterprise Server and Red Hat Enterprise Linux OVA images are supported).

To use a custom AIX or IBM i image, the image must be loaded into an IBM Cloud Object Storage location. The IBM Cloud Object Storage bucket should be created, and the custom image should be uploaded there. The IBM PowerVS offering allows provisioning a new virtual server based on an OVA image.

Storage and networking

IBM PowerVS instances support two storage tiers (Tier 1 or Tier 3), which are based on I/O operations per second (IOPS), so the performance of storage volumes is limited to the maximum number of IOPS based on volume size and storage tier. Tier 3 storage is not suitable for production workloads. Your storage tier choice should consider the average I/O load and the peak IOPS of the storage workload. Tier 3 storage is set to 3 IOPS/GB, and the Tier 1 storage is set to 10 IOPS/GB, but these numbers can change over time for IBM PowerVS. After the IOPS limit is reached for the storage volume, the I/O latency increases.

IBM PowerVS instances support both private and public network interfaces:

- ▶ A public network interface provides a quick method to connect to an IBM PowerVS instance. In this case, IBM configures the network environment to enable a secure public network connection from the internet to the IBM PowerVS instance. The connectivity is implemented on IBM Cloud by using an IBM Cloud Virtual Router Appliance (VRA) and a Direct Link Connect connection. The connection is protected by a firewall and supports various secure network protocols.
- ▶ The private network allows the IBM PowerVS instance to access existing IBM Cloud resources. The private network uses a Direct Link Connect connection to connect to the IBM Cloud account network and resources.

IBM Cloud Transit Gateway service on IBM Cloud supports IBM PowerVS connections. Connecting an IBM PowerVS instance to IBM Cloud Transit Gateway network grants access to all networks that are connected on the transit gateway, such as providing connectivity between IBM PowerVS environments at two different data centers, and interconnecting IBM PowerVS to the IBM Cloud classic and VPC infrastructures to keep traffic within the IBM Cloud network.

For example, the network architecture that is shown in Figure 3-13 on page 69 allows connectivity between multiple IBM PowerVS locations for high availability and disaster recovery (HADR) solutions, and connectivity to the IBM Cloud classic infrastructure environment and the IBM Cloud VPC environment.

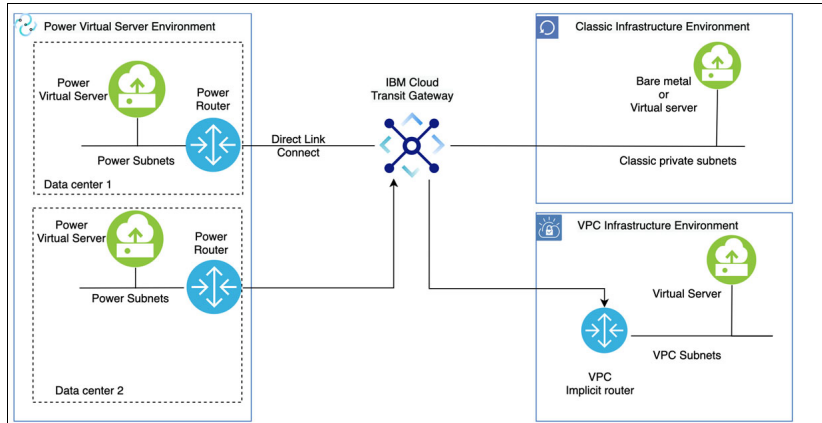


Figure 3-13 Transit Gateway connectivity across different environments

IBM PowerVS instances support VM pinning to hosts, and affinity or anti-affinity rules. The affinity or anti-affinity rules are supported for storage volumes.

High availability and disaster recovery

To support basic HA capabilities, the IBM PowerVS instance restarts the virtual servers on a different host system if a hardware failure occurs. For more advanced HA, you can use IBM PowerHA® SystemMirror® for IBM AIX running in the IBM PowerVS environment. For IBM PowerVS instances that are part of the IBM PowerHA SystemMirror cluster, IBM PowerVS allows you to select a different server by using Colocation Rules. IBM PowerVS does not provide access to the HMC, Virtual I/O Server (VIOS), and the host system, and the same is true for IBM PowerHA SystemMirror functions that require access to these capabilities.

Starting with IBM PowerHA SystemMirror 7.2.6 SP1, IBM PowerHA supports Resource Optimized High Availability (ROHA) functions to allow you to move workloads to hosts that are not configured with the same hardware.

A disaster recovery (DR) mechanism between two AIX virtual server instances in separate IBM Cloud data centers can be implemented by using Geographic Logical Volume Manager (GLVM) replication. The DR mechanisms between two IBM i virtual server instances can be implemented by using IBM PowerHA geographic mirroring. DR solutions for Linux workloads can be done by using various application and database replication technologies.

Figure 3-14 illustrates the major HADR options for IBM PowerVS servers.

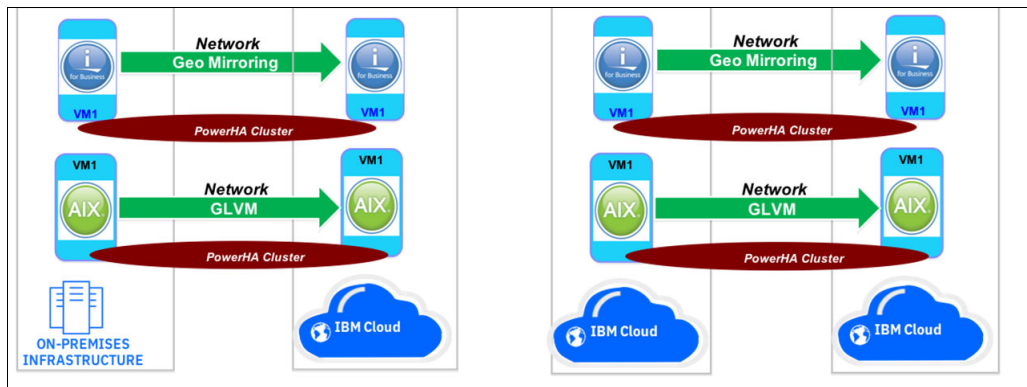


Figure 3-14 HADR options for IBM PowerVS

IBM PowerVS also offers Global Replication Service (GRS), which provides asynchronous data replication between two regions. GRS is a valuable feature for DR because a copy of your data is kept consistent at a second IBM PowerVS site that is hundreds or thousands of kilometers apart. GRS uses Global Mirror Change Volume Replication and consistency groups technologies to protect your data volumes. The change volumes are used in Global Mirror relationships to optimize the bandwidth requirements. Point-in-time-copies of the source volumes are periodically created at regular intervals, and then replicated to the secondary site rather than continuously. This approach requires less network bandwidth, is less costly, and has less impact on the active volumes.

Backup and recovery

IBM PowerVS configuration and data are not backed up automatically. However, your IBM PowerVS instance can be backed up and restored from an IBM Cloud Object Storage bucket. Any compatible agent-based backup software can be used. IBM Spectrum Protect and Veeam for AIX are two commonly used backup mechanisms for AIX. A common IBM i backup strategy is to use IBM Backup, Recovery, and Media Services (BRMS) and IBM Cloud Storage Solutions for automatically backing up the LPARs to IBM Cloud Object Storage.

You can back up and restore applications running on your Red Hat OpenShift Container Platform by using the Red Hat OpenShift application programming interface (API) for Data Protection. Red Hat OpenShift API for Data Protection backs up and restores Kubernetes resources and internal images, at the granularity of a namespace, by using Velero 1.7. Red Hat OpenShift API for Data Protection backs up and restores persistent volumes (PVs) by using snapshots or Restic.

3.2.2 Capabilities

IBM Power clients who rely on an on-premises-only infrastructure can quickly and economically extend their IBM Power IT resources off-premises by using IBM PowerVS on IBM Cloud, and avoid the large capital expense or added risk when migrating the essential workloads.

In the data centers, the IBM PowerVS separation from the rest of the IBM Cloud servers with separate networks and direct-attached storage enables IBM PowerVS to maintain key enterprise software certification and support because the IBM PowerVS architecture is identical to the certified on-premises infrastructure for IBM Power servers.

IBM PowerVS provides AIX, IBM i, or Linux capabilities in an off-premises environment that is distinct from IBM Cloud. IBM PowerVS provides fast, self-service provisioning; flexible management both on-premises and off-premises; and pay-as-you-use billing that allows scale-up and scale-out. Thus, IBM PowerVS can meet specific business needs in terms of server specifications and control workload demands by scaling up and out. Also, IBM PowerVS instances can be connected to access a stack of enterprise services either on-premises or on IBM Cloud.

While provisioning an IBM PowerVS instance, the user can specify the number of cores, amount of memory, network interfaces, and data volume size and type. IBM PowerVS processors can be either dedicated or shared (capped or uncapped). IBM PowerVS uses a monthly billing rate that is pro-rated by the hour based on the resources that are deployed for the month and includes the licenses for the AIX and IBM i operating systems. In addition to the stock AIX and IBM i images, the clients can bring their own custom AIX, IBM i, or Linux image that was tested and deployed.

An IBM PowerVS instance can support SAP NetWeaver applications on versions of IBM provided AIX or Linux stock operating system images. For SAP HANA applications, the IBM provided Linux stock image is supported. IBM i operating system and custom AIX and Linux images are not supported for SAP workloads. Red Hat OpenShift Cluster on IBM PowerVS is supported. To support installation, IBM provides automation to create the entire cluster of servers and install Red Hat OpenShift.

IBM PowerVS instances run in a multi-tenant environment. Dedicated processors provide the best overall performance. Shared, uncapped processors are slightly more flexible in addressing licensing restrictions than the capped processors. The processors are all charged on an hourly prorated basis according to the machine type (which differs across IBM Cloud regions), processor type, and the number of cores that is used in a month.

The IBM PowerVS service can be deployed for several use cases, such as AIX and IBM i production application hosting; AIX and IBM i development and test environments; DR destination for on-premises IBM Power environment; Oracle database in IBM PowerVS; and cloud-native development and application modernization by using Red Hat OpenShift on IBM PowerVS.

3.2.3 Ecosystem

For enterprise customers, an IBM Power server is an important tool, and enterprise customers consistently look for options to run an IBM Power server in the cloud because an IBM Power server can support high-performance and mission-critical workloads, such as SAP applications and Oracle databases.

IBM PowerVM is available on other clouds than IBM Cloud. Skytap is an infrastructure as a service (IaaS) platform that combines infrastructure, networking, OS, software, storage, and memory state into a single environment. This environment can be saved, cloned, copied, and shared. Skytap supports IBM PowerVM for hosting Linux, AIX, and IBM i workloads. Also, Skytap on Azure offers consumption-based pricing, and on-demand access to compute and storage resources on Azure cloud. Skytap on Azure is delivered on the Microsoft Azure global cloud infrastructure. Google Cloud also offers IBM Power as a service on Google Cloud for running AIX, IBM i, or Linux on IBM Power.

Several services and products are available to accelerate adoption and migration of workloads to IBM PowerVM on the cloud. For example, Comarch PowerCloud is a solution to migrate traditional IT infrastructures to the cloud. Comarch also provides full technical support and delivers an extensive portfolio of managed services. Comarch PowerCloud facilitates cloud deployment for IBM AIX, IBM i, and Linux virtual machines (VMs) running on IBM Power environments on-premises.

3.3 Components

The following sections explain technologies that are either built into IBM Power servers or can be implemented as part of a Red Hat OpenShift ecosystem on the servers.

In modern hybrid cloud-based environments, most of the traditional physical hardware-based infrastructure elements are accessed and used through a software-defined manner.

Software-defined resource types help with automation, scaling, and flexibility, but they change the way development, application, and operation teams work together. This change of work responsibility started with virtualization, especially as many of IBM PowerVM capabilities and features required a change in the way platform, network, and storage teams worked together.

For example, when using virtual SCSI and Shared Storage Pools, most of the traditional storage volume-mapping work moved to the IBM Power platform team. When working with virtual networking in IBM PowerVM and configuring Shared Ethernet Adapters (SEAs) and virtual switches, the platform operation team must know about VLANs, switches, and software-defined storage. Software-defined networking (SDN) features continue this evolution.

3.3.1 Software-defined storage

IBM provides the following software-defined storage solutions for Red Hat OpenShift on IBM Power. We grouped them based on the access type, like file, block, and object.

- ▶ File:
 - IBM Storage Scale (previously IBM Storage Scale)
 - Red Hat OpenShift Data Foundation - CephFS
 - Network File System (NFS) through IBM Storage Scale - Cluster Export Services without dynamic provisioning
- ▶ Block:
 - IBM Spectrum Virtualize and IBM DS8000® (Container Storage Interface (CSI))
 - Red Hat OpenShift Data Foundation - CephRBD
- ▶ Object:
 - IBM Cloud Object Storage
 - Red Hat OpenShift Data Foundation - NooBaa

The file access type provides Read Write Many (RWX) mode, and the block type provides Read Write Once (RWO) mode to access the storage in Red Hat OpenShift pods. For more information about the uses of block, file, and object solutions in your Red Hat OpenShift environment, see Figure 2-2 on page 23.

Other providers provide software-defined storage solutions, and there is the widely used NFS protocol to mount exported file systems from remote servers. NFS can be a good starting point for test and sandbox systems, but we do not recommend it for applications with high storage performance requirements. For automatic storage provisioning of NFS exported directories, use the solution at this [GitHub repository](#).

There are two IBM and Red Hat software-defined storage solutions that are recommended for use in your IBM Power based Red Hat OpenShift clusters:

- ▶ Red Hat OpenShift Data Foundation, which is described in “Red Hat OpenShift Data Foundation” on page 73.
- ▶ IBM Storage Scale, which is described in “IBM Storage Scale (previously IBM Spectrum Scale)” on page 74.

Container Storage Interface

Managing container storage in different container orchestration systems like Kubernetes and Red Hat OpenShift is done by using a standard API specification that is called CSI, which you can use to manage storage volumes without needing to know how exactly the underlying storage infrastructure works.

Red Hat OpenShift Data Foundation

Red Hat OpenShift Data Foundation integrates Ceph with multiple storage presentations, including object storage (compatible with S3), block storage, and POSIX-compliant shared file systems.

Red Hat OpenShift Data Foundation as a storage cluster can be implemented in several ways. The basic building block of an Red Hat OpenShift Data Foundation cluster is a storage node. For availability, you must have three storage nodes, which can be placed on Red Hat OpenShift master nodes, on separated infra nodes, on dedicated storage nodes, or on normal worker nodes with regular applications.

Red Hat OpenShift Data Foundation can be installed on IBM Power server-based Red Hat OpenShift clusters in internal mode. The storage nodes use SSD storage devices that are assigned to the VMs (LPARs). These SSDs are accessed through a storage class that is provided by the Local Storage Operator. At the time of writing, SSDs with capacities of 4 TB or less are supported, and you can have maximum of nine devices per storage node.

The architecture of Red Hat OpenShift Data Foundation is shown in Figure 3-15.

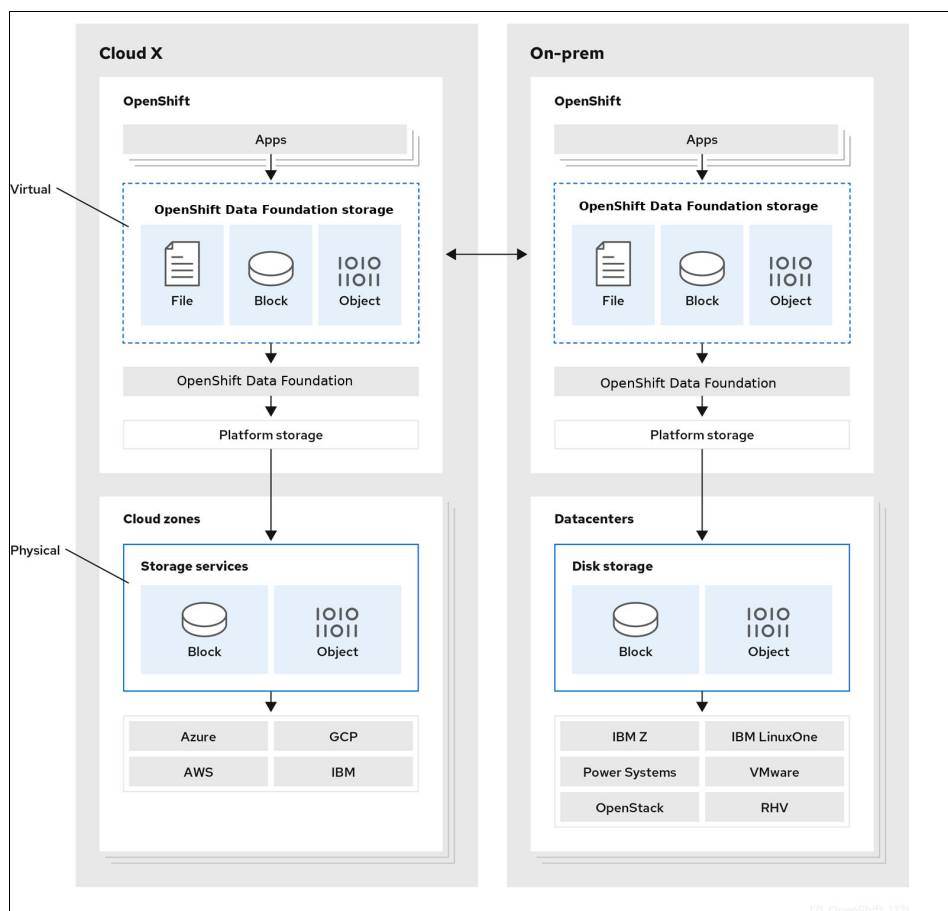


Figure 3-15 Red Hat OpenShift Data Foundation architecture

On IBM Power10 processor-based servers, you can assign NVMe SSD devices to LPARs in pairs or one by one based on the server type, as shown in Table 3-4. These NVMe devices can be used as the storage backing for Red Hat OpenShift Data Foundation.

Table 3-4 NVMe drives per IBM Power10 processor-based server types

Server type (machine type)	Maximum NVMe drives	Assign to LPARs
IBM Power S1024 (Power S1024) and IBM Power L1024 (Power L1024) (9105-42A and 9786-42H)	16	One by one
Power S1014 (9105-41B)	16	In pairs
IBM Power S1022 (Power S1022) and IBM Power L1022 (Power L1022) (9105-22A and 9786-22H)	8	In pairs
Power E1050 (9043-MRX)	10	One by one
Power E1080 (9080-HEX)	4 per CEC drawer (4 x 4 max.)	One by one

After a successful deployment of Red Hat OpenShift Data Foundation, there are the following storage classes:

- ▶ `ocs-storagecluster-ceph-rbd`: Block-based storage class for RWO and RWX modes, and RWO mode for file system volume mode access of CephRBDs.
- ▶ `ocs-storagecluster-cephfs`: File-based storage class for RWO and RWX modes.
- ▶ `openshift-storage.noobaa.io`: Storage class for Object Buckets, which is based on NoobBaa (MultiCloud Object Gateway), which provides an AWS S3 based object API for cloud-native object storage.
- ▶ `ocs-storagecluster-ceph-rgw`: Storage class for object storage, which is based on the Ceph Object Gateway native object storage interface.

When using the object gateway, the endpoint determines which storage class and interface handles the storage requests.

IBM Storage Scale (previously IBM Spectrum Scale)

IBM Storage Scale provides a global data platform for high-performance, next-generation data services. It has been used nearly 20 years in demanding enterprise environments.

Here are the key features of IBM Storage Scale:

- ▶ Connects applications by providing a unified data fabric and a single namespace.
- ▶ Accesses data independently of underlying storage technology.
- ▶ Scalability and optimization.
- ▶ Policy engine and active file management (AFM).
- ▶ Increases security through encryption immutability.
- ▶ Eliminates data loss and data corruption.
- ▶ Integrates with backup solutions.
- ▶ Single point of management with an intuitive GUI.
- ▶ High-performance S3 interface.

IBM Storage Scale can be implemented as a storage cluster in which local or storage area network (SAN)-attached storage devices are used to store the user data, and it can be implemented as a compute cluster in which remote IBM Storage Scale file systems are mounted to allow access of remote data.

The nodes can be bare metal and virtual servers, and it is possible to install IBM Storage Scale in containerized mode too.

In an IBM Power processor-based environment, NVMe drives can be used as locally attached disks for the best performance. The NVMe drives can be attached to a VM (LPAR in IBM Power servers) in pairs or one by one, as shown in Table 3-4 on page 74.

Container-native access of data that is stored in IBM Storage Scale

IBM Storage Scale Container Native Storage Access is a containerized version of IBM Storage Scale. To provide container-native access for data on an IBM Storage Scale cluster, a containerized IBM Storage Scale cluster must be installed in the Red Hat OpenShift cluster. This cluster and its nodes do not have local storage that is attached, but you can access remote data by using the IBM Storage Scale remote mount option.

Note: For more information about architecture-specific prerequisites before installing IBM Storage Scale Container Native Storage Access, see [Red Hat OpenShift Container Platform configuration](#).

Applications can access remote data as PVs through the IBM Storage Scale CSI driver, as shown in Figure 3-16.

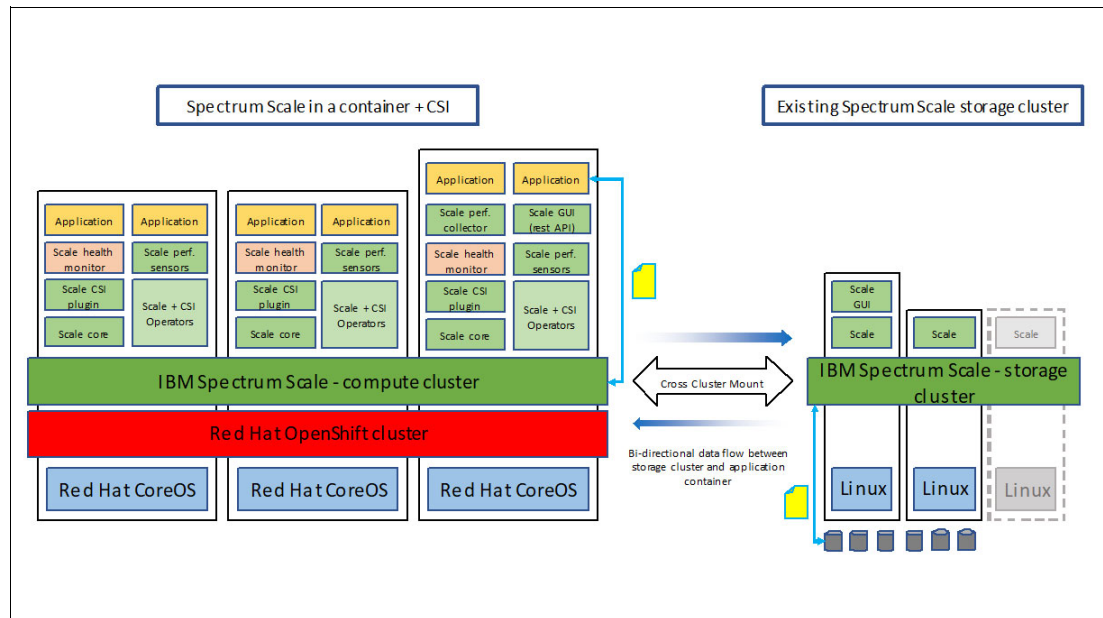


Figure 3-16 Container Native Storage Access architecture

After installing IBM Storage Scale Container Native Storage Access on an Red Hat OpenShift cluster, you can see the default Storage Classes. Example 3-1 shows the default Storage Classes when IBM Storage Scale Container Native Storage Access and NFS client storage provisioner is installed.

Example 3-1 Listing the Storage Classes in Red Hat OpenShift

```
(py39) [root@build-cp4d-1 ~]# oc get sc
```

NAME	PROVISIONER	RECLAIMPOLICY
ibm-spectrum-scale-csi-fileset	spectrumscale.csi.ibm.com	Delete
Immediate	false	30d
ibm-spectrum-scale-internal	kubernetes.io/no-provisioner	Delete
WaitForFirstConsumer	false	30d
ibm-spectrum-scale-sample	spectrumscale.csi.ibm.com	Delete
Immediate	false	30d
nfs-storage-provisioner (default)	nfs-storage	Delete
Immediate	false	30d

Red Hat OpenShift users can use the default Storage Class that is created when IBM Storage Scale Container Native Storage Access is configured to allocate PVs and persistent volume claims (PVCs) or create IBM Storage Scale file set-based or lightweight (directory-based) Storage Classes:

- ▶ For file set-based PVCs, IBM Storage Scale creates a separate file set for each PVC and the underlying PV.
- ▶ For lightweight volumes, you must create a directory in the remotely mounted IBM Storage Scale file system. The CSI driver creates directories for the PVCs under that directory. The lightweight volumes are not mounted to the containers the same way as file set-based volumes, as shown in Example 3-2.

Example 3-2 Attached file set-based and lightweight volumes

```
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
overlay         257G   42G  215G  17% /
tmpfs           64M    0   64M   0% /dev
tmpfs           64G    0   64G   0% /sys/fs/cgroup
shm            64M    0   64M   0% /dev/shm
tmpfs           64G  102M   64G   1% /etc/passwd
remote-sample   10G   7.7G   2.4G  77% /testpvc
/dev/sda4       257G   42G  215G  17% /etc/hosts
tmpfs          127G  256K  127G   1% /run/secrets/kubernetes.io/serviceaccount
tmpfs           64G    0   64G   0% /proc/scsi
tmpfs           64G    0   64G   0% /sys/firmware
$ ls -ld /test*
drwxrwsrwx. 2 root 1000760000 4096 Nov  9 09:05 /testpvc
drwxrws--x. 2 root 1000760000 4096 Nov  9 13:13 /testpvc1w1
drwxrws--x. 2 root 1000760000 4096 Nov  9 13:34 /testpvc1w2
```

Note: Because a lightweight volume does not enforce quota, it can grow beyond its defined size and might use a whole file system. To avoid this situation, manually create or use an existing file set to host the lightweight PVC volumes by specifying the directory inside the file set by using the `volDirBasePath` option.

Moving pods between nodes with persistent volume claims attached

PVs and PVCs can be used in RWX and RWO modes. RWX mode enables mounting volumes from multiple nodes concurrently, but RWO does not. Account for this situation if you use a PV or PVC from one pod but want to move the pod to a new node, which means changing the deployment configuration.

Kubernetes provides a configuration option for deployments that defines what happens when you upgrade the configuration. This setting is the *strategy* setting, and it is shown in Example 3-3.

Example 3-3 Deployment strategy

```
strategy:
  type: RollingUpdate
  rollingUpdate:
    maxUnavailable: 25%
    maxSurge: 25%
```

With the RollingUpdate strategy, new pods are started before the old ones are stopped, which might cause a problem when you use PVCs in RWO mode.

Example 3-4 shows moving a pod from one node to another one by using an RWX volume.

Example 3-4 Moving a pod by using an RWX PVC

```
(py39) [root@build-cp4d-1 ~]# oc get pod -o wide
NAME          READY  STATUS   RESTARTS  AGE  IP            NODE                                     NOMINATED NODE
READINESS GATES
ubuntu1-5d6c67bd95-c6bhb 1/1    Running  0         54m  10.131.0.145  worker1.cp4d-1.rtp.raleigh.ibm.com  <none>         <none>
ubuntu2-54db65648b-69x69 1/1    Running  0         81s  10.128.2.22   worker2.cp4d-1.rtp.raleigh.ibm.com  <none>         <none>
ubuntu3-9b76df65c-4szdn  1/1    Running  0         49m  10.129.2.98   worker3.cp4d-1.rtp.raleigh.ibm.com  <none>         <none>
(py39) [root@build-cp4d-1 ~]# oc patch deployment/ubuntu2 -p '{"op": "replace", "path": "/spec/template/spec/nodeName", "value": "worker4.cp4d-1.rtp.raleigh.ibm.com"}' --type=json
deployment.apps/ubuntu2 patched
(py39) [root@build-cp4d-1 ~]# oc get pod -o wide
NAME          READY  STATUS             RESTARTS  AGE  IP            NODE                                     NOMINATED
NODE READINESS GATES
ubuntu1-5d6c67bd95-c6bhb 1/1    Running           0         55m  10.131.0.145  worker1.cp4d-1.rtp.raleigh.ibm.com  <none>
<none>
ubuntu2-54db65648b-69x69 1/1    Running           0         110s  10.128.2.22   worker2.cp4d-1.rtp.raleigh.ibm.com  <none>
<none>
ubuntu2-5dc4b447cb-fpd6d 0/1    ContainerCreating 0         4s    <none>        worker4.cp4d-1.rtp.raleigh.ibm.com  <none>
<none>
ubuntu3-9b76df65c-4szdn  1/1    Running           0         50m  10.129.2.98   worker3.cp4d-1.rtp.raleigh.ibm.com  <none>
<none>
(py39) [root@build-cp4d-1 ~]# oc get pod -o wide
NAME          READY  STATUS             RESTARTS  AGE  IP            NODE                                     NOMINATED NODE
READINESS GATES
ubuntu1-5d6c67bd95-c6bhb 1/1    Running           0         55m  10.131.0.145  worker1.cp4d-1.rtp.raleigh.ibm.com  <none>
<none>
ubuntu2-54db65648b-69x69 1/1    Terminating     0         2m   10.128.2.22   worker2.cp4d-1.rtp.raleigh.ibm.com  <none>
<none>
ubuntu2-5dc4b447cb-fpd6d 1/1    Running           0         14s  10.130.2.253  worker4.cp4d-1.rtp.raleigh.ibm.com  <none>
<none>
ubuntu3-9b76df65c-4szdn  1/1    Running           0         50m  10.129.2.98   worker3.cp4d-1.rtp.raleigh.ibm.com  <none>
<none>
(py39) [root@build-cp4d-1 ~]# oc get pod -o wide
NAME          READY  STATUS   RESTARTS  AGE  IP            NODE                                     NOMINATED NODE
READINESS GATES
ubuntu1-5d6c67bd95-c6bhb 1/1    Running  0         56m  10.131.0.145  worker1.cp4d-1.rtp.raleigh.ibm.com  <none>         <none>
ubuntu2-5dc4b447cb-fpd6d 1/1    Running  0         65s  10.130.2.253  worker4.cp4d-1.rtp.raleigh.ibm.com  <none>         <none>
ubuntu3-9b76df65c-4szdn  1/1    Running  0         51m  10.129.2.98   worker3.cp4d-1.rtp.raleigh.ibm.com  <none>         <none>
```

Example 3-5 shows an attempt to move a pod from one node to another by using an RWO volume. This attempt fails, and the new pod remains in the Pending state because Red Hat OpenShift cannot attach the volume to the pod. Scale down the deployment and scale up again to move it to another node and get the same RWO volume.

Example 3-5 Moving a pod by using an RWO PVC

```
(py39) [root@build-cp4d-1 ~]# oc patch deployment/ubuntu1 -p '[{"op": "replace", "path": "/spec/template/spec/nodeName", "value": "worker2.cp4d-1.rtp.raleigh.ibm.com"}]' --type=json
deployment.apps/ubuntu1 patched
(py39) [root@build-cp4d-1 ~]# oc get pod -o wide
NAME                                READY  STATUS             RESTARTS  AGE   IP              NODE                                NOMINATED
NODE  READINESS GATES
ubuntu1-5d6c67bd95-c6bhb           1/1    Running            0          62m   10.131.0.145   worker1.cp4d-1.rtp.raleigh.ibm.com <none>
<none>
ubuntu1-7899dcb98-v958d            0/1    ContainerCreating  0          82s   <none>         worker2.cp4d-1.rtp.raleigh.ibm.com <none>
<none>
ubuntu2-5dc4b447cb-fpd6d           1/1    Running            0          6m47s 10.130.2.253   worker4.cp4d-1.rtp.raleigh.ibm.com <none>
<none>
ubuntu3-9b76df65c-4szdn           1/1    Running            0          57m   10.129.2.98    worker3.cp4d-1.rtp.raleigh.ibm.com <none>
<none>
(py39) [root@build-cp4d-1 ~]# oc describe pod ubuntu1-7899dcb98-v958d
Name:                ubuntu1-7899dcb98-v958d
Namespace:           lniesz
Priority:              0
Node:                worker2.cp4d-1.rtp.raleigh.ibm.com/9.42.76.22
Start Time:          Wed, 09 Nov 2022 05:03:29 -0500
Labels:              app=ubuntu1
                    pod-template-hash=7899dcb98
Annotations:         openshift.io/scc: restricted
Status:              Pending
IP:
IPs:                 <none>
Controlled By:       ReplicaSet/ubuntu1-7899dcb98
Containers:
...
#Details removed from here!###
...
Conditions:
  Type           Status
  Initialized     True
  Ready          False
  ContainersReady False
  PodScheduled   True
Volumes:
  testpvc:
    Type:          PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:     testpvc1
    ReadOnly:     false
  kube-api-access-678cv:
    Type:          Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:  kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI:    true
    ConfigMapName:  openshift-service-ca.crt
    ConfigMapOptional: <nil>
QoS Class:       BestEffort
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                 node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type           Reason          Age   From          Message
  ----           -
  Warning        FailedAttachVolume 3m7s  attachdetach-controller Multi-Attach error for volume "pvc-c1e354a5-1849-455e-9d2a-149902ff9d45"
  Volume is already used by pod(s) ubuntu1-5d6c67bd95-c6bhb
  Warning        FailedMount       64s   kubelet        Unable to attach or mount volumes: unmounted volumes=[testpvc], unattached
  volumes=[testpvc kube-api-access-678cv]: timed out waiting for the condition
(py39) [root@build-cp4d-1 ~]# oc scale deployment/ubuntu1 --replicas=0
deployment.apps/ubuntu1 scaled
(py39) [root@build-cp4d-1 ~]# oc get pod -o wide
NAME                                READY  STATUS             RESTARTS  AGE   IP              NODE                                NOMINATED
NODE  READINESS GATES
ubuntu1-5d6c67bd95-c6bhb           1/1    Terminating      0          68m   10.131.0.145   worker1.cp4d-1.rtp.raleigh.ibm.com <none>
<none>
ubuntu1-7899dcb98-v958d            0/1    Terminating      0          8m4s   <none>         worker2.cp4d-1.rtp.raleigh.ibm.com <none>
<none>
ubuntu2-5dc4b447cb-fpd6d           1/1    Running            0          13m   10.130.2.253   worker4.cp4d-1.rtp.raleigh.ibm.com <none>
<none>
```

```

ubuntu3-9b76df65c-4szdn 1/1 Running 0 63m 10.129.2.98 worker3.cp4d-1.rtp.raleigh.ibm.com <none>
<none>
(py39) [root@build-cp4d-1 ~]# oc get pod -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE
READINESS GATES
ubuntu2-5dc4b447cb-fpd6d 1/1 Running 0 14m 10.130.2.253 worker4.cp4d-1.rtp.raleigh.ibm.com <none> <none>
ubuntu3-9b76df65c-4szdn 1/1 Running 0 64m 10.129.2.98 worker3.cp4d-1.rtp.raleigh.ibm.com <none> <none>
(py39) [root@build-cp4d-1 ~]# oc scale deployment/ubuntu1 --replicas=1
deployment.apps/ubuntu1 scaled
(py39) [root@build-cp4d-1 ~]# oc get pod -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE
READINESS GATES
ubuntu1-7899dcb98-hm664 1/1 Running 0 19s 10.128.2.23 worker2.cp4d-1.rtp.raleigh.ibm.com <none> <none>
ubuntu2-5dc4b447cb-fpd6d 1/1 Running 0 14m 10.130.2.253 worker4.cp4d-1.rtp.raleigh.ibm.com <none> <none>
ubuntu3-9b76df65c-4szdn 1/1 Running 0 65m 10.129.2.98 worker3.cp4d-1.rtp.raleigh.ibm.com <none> <none>

```

The movement of pods among nodes might be necessary for performance tuning, maintenance, and spreading the load between nodes.

Monitoring the throughput of PVCs on IBM Storage Scale

There are many ways to monitor IBM Storage Scale throughput, but IBM Storage Scale Container Native Storage Access based IBM Storage Scale volumes are network-attached volumes. Therefore, you do not see the local disk load on the nodes where the application pods run, but the network traffic increases.

The following examples show Red Hat OpenShift and IBM Storage Scale based monitoring screen captures to check the load that is generated by `dd` commands.

Figure 3-17 shows the network-related section of the Red Hat OpenShift Observability dashboard: Kubernetes / Compute Resources / Namespace (Workloads) for the namespace: `ibm-spectrum-scale-csi` and type: `daemonset`.

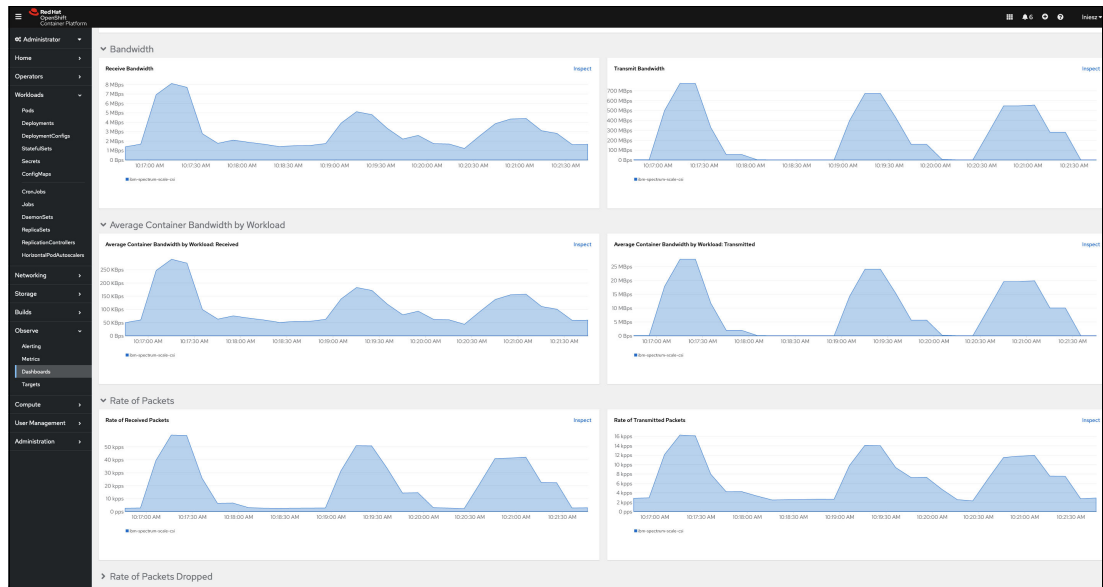


Figure 3-17 Network traffic increased on nodes with IBM Storage Scale Container Native Storage Access

You can check the metric `node_network_transmit_byte_excluding_lo` for the IBM Storage Scale nodes, as shown in Figure 3-18.

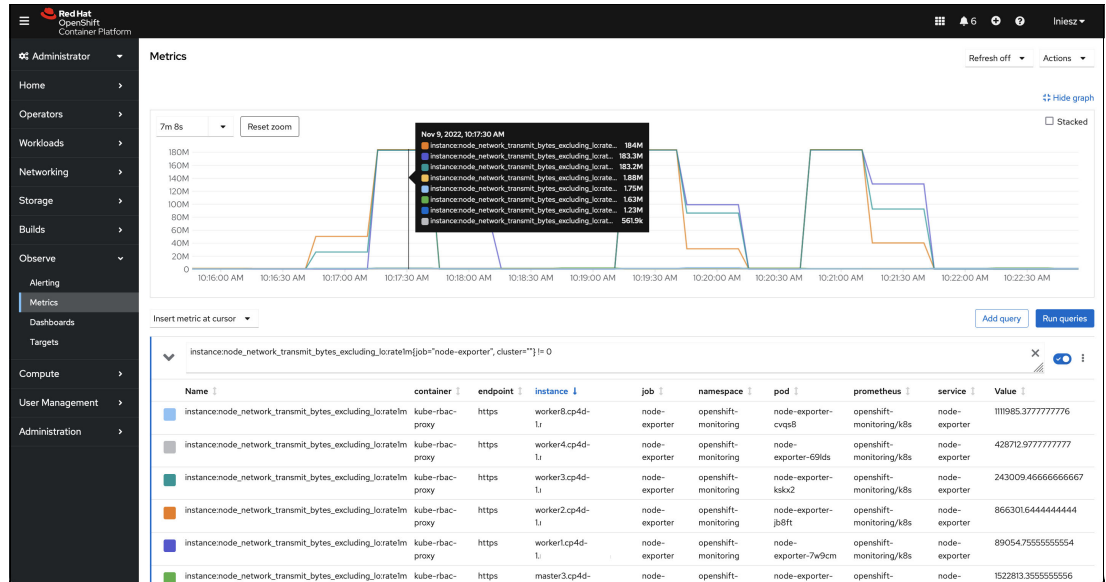


Figure 3-18 IBM Storage Scale node transmit metrics in the Red Hat OpenShift Observability dashboard

IBM Storage Scale on Red Hat OpenShift also has a GUI that is accessible through the auto-created Red Hat OpenShift route `ibm-spectrum-scale-gui`. This user interface (UI) has a monitoring menu where you can see dashboards, statistics, events, thresholds, and audit logs.

Figure 3-19 shows the client throughput to disk statistics.

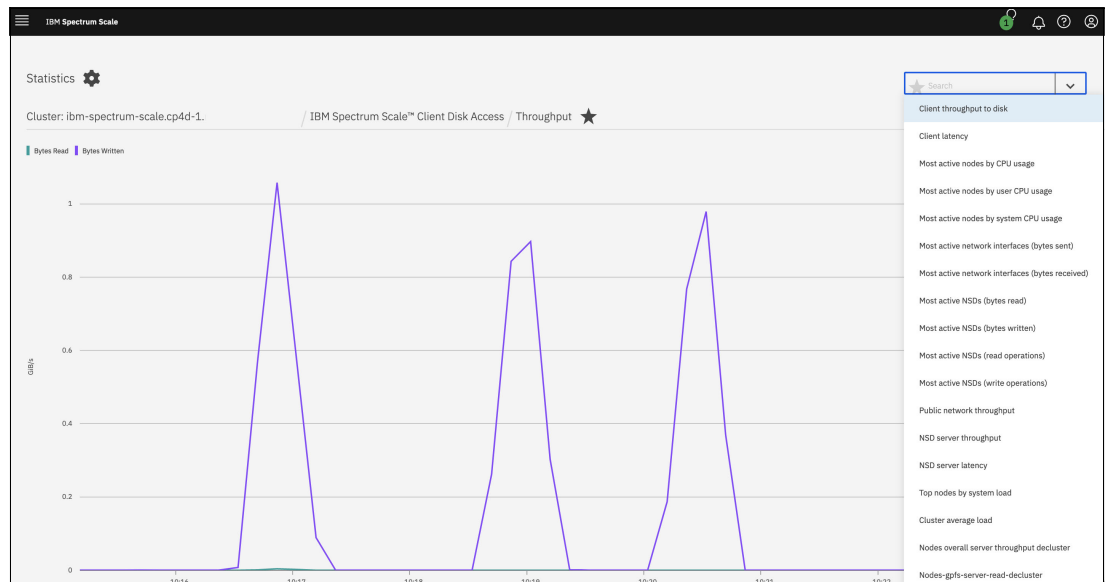


Figure 3-19 Client throughput to disk while running the `dd` command in a Red Hat OpenShift pod

IBM Storage Scale Data Access Service

IBM Storage Scale provides a high-performance S3 interface to access data as objects. The implementation architecture for IBM Storage Scale is shown in Figure 3-20.

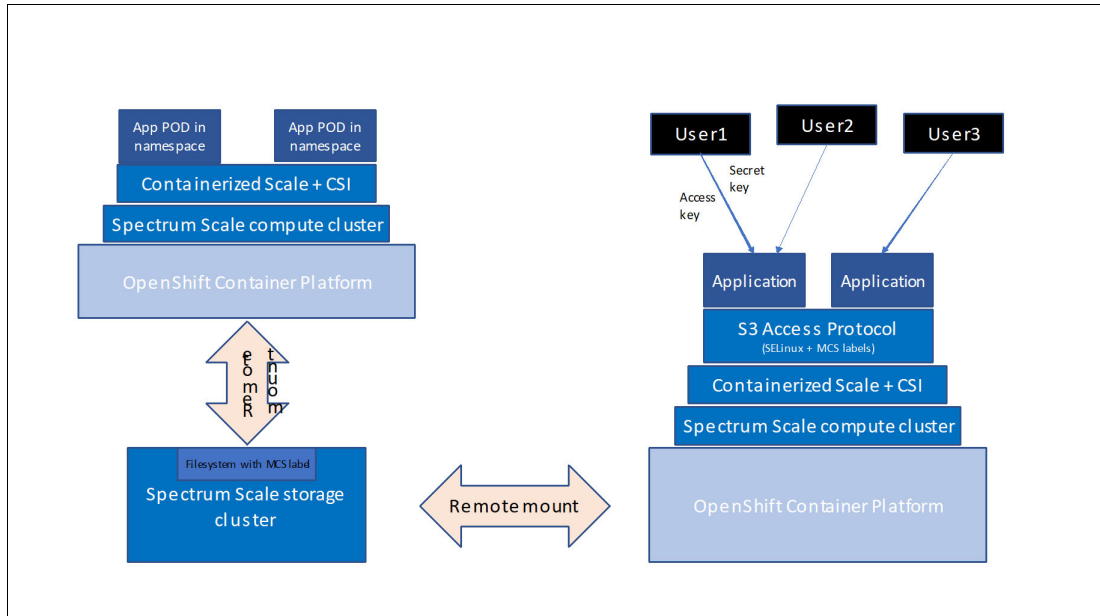


Figure 3-20 CSI with IBM Storage Scale Data Access Service cluster

IBM Storage Scale enables the access of the same data both as objects (by using the S3 protocol) and as normal file system files (by using containerized applications) concurrently.

Note: The IBM Storage Scale Data Access Service cluster must be a dedicated IBM Storage Scale Container Native Storage Access cluster on x86_64 based bare metal servers, and the remotely mounted IBM Storage Scale cluster must be based on IBM Elastic Storage® System (IBM ESS).

IBM Storage Scale Data Access Service includes an embedded license for Red Hat OpenShift Data Foundation. Therefore, the IBM Storage Scale Data Access Service operator implicitly deploys Red Hat OpenShift Data Foundation. The usage of Red Hat OpenShift Data Foundation is limited to the features that can be configured with the IBM Storage Scale Data Access Service management interfaces.

For more information, see the IBM published benchmark results for IBM Storage Scale Data Access Service at [IBM Spectrum Scale DAS 5.1.3.1 performance evaluation using COSBench](#), where COSBench uses objects with a size of 1 GB running against a 3-node IBM Storage Scale Data Access Service cluster, and uses IBM Elastic Storage System 3200 (IBM ESS 3200) as the back-end storage produced the following results:

- ▶ More than 60 GBps aggregated throughput for read workloads
- ▶ More than 20 GBps aggregated throughput for write workloads

3.3.2 Software-defined networking

SDN is the decoupling of the network control logic from the devices that perform the function, such as routers, which control the movement of information in the underlying network. This approach simplifies the management of infrastructure, which might be specific to one organization or partitioned to be shared among several organizations.⁸

SDN features controllers that lay above the network hardware in the cloud or on-premises, which offer policy-based management. The network control plane and forwarding plane are separated from the data plane (or underlying infrastructure) so that the organization can program network control directly. This approach differs from traditional data center environments, where a router or switch (whether in the cloud or physically in the data center) is aware of the status of only network devices that are adjacent to it. With SDN, the intelligence is centralized and prolific, so it can view and control everything.

SDN has three main components:

- ▶ Applications, which need information about the network capabilities and request resources from the network.
- ▶ SDN controllers, which communicate with the applications and determine the destination of data packets, and play a load-balancing role.
- ▶ Networking devices, which are controlled by the controllers to route the traffic.

SDN works in virtualized environments, and it enables policy-based network management.

SDN types include the following ones:

- ▶ An *open SDN* uses an open protocol to manage virtual and physical devices.
- ▶ An *API SDN* is an API-based solution.
- ▶ An *overlay model SDN* creates a virtual network over existing physical or virtualized network devices to provide tunnels for traffic channels.
- ▶ A *hybrid model SDN* combines traditional networking with SDN features to enable the optimal protocol for each type of traffic.

The controller function is critical in the SDN implementation both from security and availability points of view, so it is necessary to create a highly available (HA) and secure solution.

IBM PowerVM virtual Ethernet in CoreOS

IBM PowerVM provides hypervisor-based network virtualization. The client VMs (LPARs) see virtual Ethernet adapters the same as in bare metal configurations.

Example 3-6 shows network device-related information from a CoreOS based Red Hat OpenShift node that is on an IBM Power10 processor-based server-based LPAR.

Example 3-6 Virtual Ethernet adapter in CoreOS

```
sh-4.4# lsdevinfo
device:
  name="env2"
  uniquetype="adapter/vdevice/IBM,1-lan"
  class="adapter"
  subclass="vdevice"
  type="IBM,1-lan"
  prefix="eth"
```

⁸ <https://www.ibm.com/cloud/blog/software-defined-networking>

```

driver="ibmveth"
status="1"

path:
  parent="vio"
  physloc="U9080.HEX.785EDA8-V188-C2-T0"
  connection="3000002"
...
sh-4.4# ls-veth
env2 U9080.HEX.785EDA8-V188-C2-T0
sh-4.4# ethtool env2
Settings for env2:
  Supported ports: [ FIBRE ]
  Supported link modes:   1000baseT/Full
  Supported pause frame use: No
  Supports auto-negotiation: Yes
  Supported FEC modes: Not reported
  Advertised link modes:  1000baseT/Full
  Advertised pause frame use: No
  Advertised auto-negotiation: Yes
  Advertised FEC modes: Not reported
  Speed: 1000Mb/s
  Duplex: Full
  Auto-negotiation: on
  Port: FIBRE
  PHYAD: 0
  Transceiver: internal
  Link detected: yes

```

You can use the Linux `nmcli` command on the CoreOS node to check the network configuration, as shown in Example 3-7.

Example 3-7 The nmcli command

```

sh-4.4# nmcli connection show
NAME                                UUID                                TYPE    DEVICE
Wired Connection d36fa633-27fb-46d5-a905-9bb8298eab0d ethernet env2

```

The `nmcli device show` command shows the detailed configuration of the physical or virtual Ethernet device and the dynamic SDN configuration that is based on Open vSwitch (OVS), which is the default SDN solution on Red Hat OpenShift.

The network interface setup is done at the ignition phase of Red Hat OpenShift installation, in which operating system- and device-specific configuration is done on the CoreOS nodes.

The configuration is managed by the Machine Config Operator using MachineConfig custom resources (CRs). The network and SDN-related configuration is placed in the `00-master` and `00-work` MachineConfigs for master and worker nodes. At installation time and when a new MachineConfig resource is created, the operator combines all configurations into a rendered configuration and the nodes are restarted with this new configuration, so changing an existing or creating a configuration can result in application outages because the pods are available while the nodes restart.

Open vSwitch

OVS is a multilayer software switch that is licensed under the open-source Apache 2 license.⁹

OVS functions as a virtual switch in VM environments. In addition to exposing standard control and visibility interfaces to the virtual networking layer, it supports distribution across multiple physical servers. OVS uses virtual extensible local area network (VXLAN) technology as an overlay network to transport an L2 network over an existing L3 network. For more information, see [RFC7348](#).

Example 3-8 shows how OVS service is configured by default on a Red Hat OpenShift worker node.

Example 3-8 OVS configuration on a Red Hat OpenShift node

```
(py39) [root@build-cp4d-1 ~]# oc debug node/worker1.cp4d-1.rtp.raleigh.ibm.com
Starting pod/worker1cp4d-1rtpraleighibmcom-debug ...
To use host binary files, run `chroot /host`
Pod IP: 9.42.76.21
If you don't see a command prompt, try pressing enter.

sh-4.4# chroot /host

sh-4.4# systemctl status openvswitch
? openvswitch.service - Open vSwitch
   Loaded: loaded (/usr/lib/systemd/system/openvswitch.service; enabled; vendor
   preset: disabled)
   Active: active (exited) since Mon 2022-10-24 11:17:34 UTC; 2 weeks 4 days ago
   Main PID: 1526 (code=exited, status=0/SUCCESS)
     Tasks: 0 (limit: 836372)
    Memory: 0B
       CPU: 0
    CGroup: /system.slice/openvswitch.service

Oct 24 11:17:34 localhost systemd[1]: Starting Open vSwitch...
Oct 24 11:17:34 localhost systemd[1]: Started Open vSwitch.

sh-4.4# cat /usr/lib/systemd/system/openvswitch.service
[Unit]
Description=Open vSwitch
Before=network.target network.service
After=network-pre.target ovssdb-server.service ovs-vswitchd.service
PartOf=network.target
Requires=ovssdb-server.service
Requires=ovs-vswitchd.service

[Service]
Type=oneshot
ExecStart=/bin/true
ExecReload=/usr/share/openvswitch/scripts/ovs-systemd-reload
ExecStop=/bin/true
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

⁹ <https://github.com/openvswitch/ovs>

To check the OVS configuration on Red Hat OpenShift, run `ovs-vsctl`.

Red Hat OpenShift and Open vSwitch

Red Hat OpenShift Container Platform uses an SDN approach to provide a unified cluster network that enables communication between pods across the Red Hat OpenShift Container Platform cluster. This pod network is established and maintained by the Red Hat OpenShift SDN, which configures an overlay network by using OVS.

The [Red Hat OpenShift SDN](#) uses OVS, VXLAN tunnels, OpenFlow rules, and iptables. This network can be tuned by using jumbo frames, network interface controllers (NIC) offloads, multi-queue, and ethtool settings.

OVN-Kubernetes uses Generic Network Virtualization Encapsulation (Geneve) instead of VXLAN as the tunnel protocol.

VXLAN provides benefits over VLANs, such as an increase in networks from 4096 to over 16 million, and layer 2 connectivity across physical networks. With these benefits, pods behind a service may communicate with each other, even if they are running on different systems.

VXLAN encapsulates all tunneled traffic in user datagram protocol (UDP) packets. However, this approach leads to increased CPU utilization. Both the outer and inner packets are subject to normal checksumming rules to ensure that data is not corrupted during transit. Depending on CPU performance, this extra processing can reduce throughput and increase latency compared to traditional, non-overlay networks.

Optimizing networking

Here are some items to consider to optimize networking:

- ▶ You can use VXLAN-offload capable network adapters to move the packet checksum calculation and associated CPU impact off from the system CPU and onto dedicated hardware on the network adapter, which can increase network throughput over Gbps.
- ▶ Another approach to optimization is to increase the maximum transmission units (MTUs), which must be done by using the whole traffic route of network packets:
 - Physical Ethernet adapter ports in an IBM Power server
 - Etherchannel or a Link Aggregation adapter on a VIOS (if used)
 - SEA on a VIOS
 - VIOS virtual Ethernet adapter
 - LPAR or Red Hat OpenShift node virtual Ethernet adapter
 - Red Hat OpenShift network

You can find the Red Hat OpenShift network- and node-related steps for MTU migration at [Changing the MTU for the cluster network](#).

- ▶ Using IPsec can further decrease performance because it can prevent you from using some network interface acceleration features like NIC offloading.
- ▶ View the OVS-related logs to find areas for optimization, as shown in Example 3-9.

Example 3-9 Listing OVS logs on a Red Hat OpenShift node

```
[root@build-cp4d-1 ~]# oc adm node-logs worker8.cp4d-1.rtp.raleigh.ibm.com -u ovs-vswitchd|tail -5
Nov 11 17:10:08.736229 worker8.cp4d-1.rtp.raleigh.ibm.com ovs-vswitchd[2779]: ovs|123908|bridge|INFO|bridge br0: deleted interface
veth14ff84f5 on port 10501
Nov 11 17:10:09.508285 worker8.cp4d-1.rtp.raleigh.ibm.com ovs-vswitchd[2779]: ovs|123909|connmgr|INFO|br0<->unix#320069: 2 flow_mods in
the last 0 s (2 deletes)
Nov 11 17:10:09.539912 worker8.cp4d-1.rtp.raleigh.ibm.com ovs-vswitchd[2779]: ovs|123910|connmgr|INFO|br0<->unix#320072: 4 flow_mods in
the last 0 s (4 deletes)
Nov 11 17:10:09.649754 worker8.cp4d-1.rtp.raleigh.ibm.com ovs-vswitchd[2779]: ovs|123911|bridge|INFO|bridge br0: deleted interface
vethff04024a on port 10502
```

```

Nov 11 17:10:10.386408 worker8.cp4d-1.rtp.raleigh.ibm.com ovs-vswitchd[2779]: ovs|123912|connmgr|INFO|br0<->unix#320075: 92 flow_mods in
the last 0 s (84 adds, 8 deletes)
(py39) [root@build-cp4d-1 ~]# oc debug node/master1.cp4d-1.rtp.raleigh.ibm.com
Starting pod/master1cp4d-1rtpraleighibmcom-debug ...
To use host binary files, run `chroot /host`
Pod IP: 9.42.76.43
If you don't see a command prompt, try pressing enter.

sh-4.4# chroot /host

sh-4.4# journalctl -b -u ovs-vswitchd.service|tail -8
Nov 11 16:04:10 master1.cp4d-1.rtp.raleigh.ibm.com ovs-vswitchd[1202]: ovs|02593|bridge|INFO|bridge br0: deleted interface vethbfd1b7e on
port 270
Nov 11 16:43:01 master1.cp4d-1.rtp.raleigh.ibm.com ovs-vswitchd[1202]: ovs|02594|bridge|INFO|bridge br0: added interface vethf9f30158 on
port 271
Nov 11 16:43:02 master1.cp4d-1.rtp.raleigh.ibm.com ovs-vswitchd[1202]: ovs|02595|connmgr|INFO|br0<->unix#30466: 5 flow_mods in the last 0
s (5 adds)
Nov 11 16:43:02 master1.cp4d-1.rtp.raleigh.ibm.com ovs-vswitchd[1202]: ovs|02596|connmgr|INFO|br0<->unix#30469: 2 flow_mods in the last 0
s (2 deletes)
Nov 11 16:43:14 master1.cp4d-1.rtp.raleigh.ibm.com ovs-vswitchd[1202]: ovs|02597|connmgr|INFO|br0<->unix#30472: 2 flow_mods in the last 0
s (2 deletes)
Nov 11 16:43:14 master1.cp4d-1.rtp.raleigh.ibm.com ovs-vswitchd[1202]: ovs|02598|connmgr|INFO|br0<->unix#30475: 4 flow_mods in the last 0
s (4 deletes)
Nov 11 16:43:14 master1.cp4d-1.rtp.raleigh.ibm.com ovs-vswitchd[1202]: ovs|02599|bridge|INFO|bridge br0: deleted interface vethf9f30158 on
port 271
Nov 11 16:44:50 master1.cp4d-1.rtp.raleigh.ibm.com ovs-vswitchd[1202]: ovs|02600|connmgr|INFO|br0<->unix#30479: 8 flow_mods in the last 0
s (8 deletes)

```

3.3.3 I/O operations per second

IOPS is one measurement of your storage requirements as you plan for a Red Hat OpenShift cluster and during day to day operation. IOPS is not always directly related to a user transaction because each transaction can result in one to many interactions with the storage device. Applications that use databases often require more IOPS than simple web applications, so understanding your application is important as you plan your cluster.

IOPS on an operational cluster can be monitored through different Red Hat OpenShift utilities and should be monitored as you continue to scale your cluster.

The IOPS requirement for your cluster is an important factor as you decide what type of storage to provision for the applications and services that you are running. Storage can generally be characterized by *tiers* (for more information about storage tiers, see 3.3.4, “Tier 1 and Tier 3 storage” on page 86). If you have higher IOPS requirements, you should consider using Tier 1 storage to provide a good user experience. Lower IOPS requirements often can be satisfied by using lower storage tiers.

3.3.4 Tier 1 and Tier 3 storage

The type of storage that is used to back up the PVCs in your cluster heavily influences the performance of the cluster and the experience that your users have with the applications running there. The different performance characteristics of the available storage technologies often are used to provide generic tiers of storage.

Tiering is a broad methodology of defining the capabilities of your storage. It is not standardized across the industry, but in general Tier 1 represents the highest performance devices and Tier 3 represents lower-level performance devices. Tier 1 technology tends to be more expensive to procure and operate and usually has less capacity, which leads to a higher cost per unit of storage compared to lower-tier devices. In a cloud or managed service environment, the specification of the expected performance for the different tiers of storage is an important consideration.

Not all applications and services require the highest tier storage, either due to the way that the service uses external storage or your willingness to accept a lower level of service for that application.

Both internal and external storage can be provided by many technologies, all of which have specific performance characteristics. These different device classes do not necessarily relate directly to the tiers, but you can be assured that devices with lower performance capabilities are used to provide storage in the lower storage tiers.

Storage can be provided by hard disk drives, which tend to be relatively slow. Newer technology for storage includes SSDs, which provide lower latency and better performance than hard disk drives. They also tend to be more reliable because they do not rely on moving parts to operate. Even SSDs have a range of capabilities and connection types that differentiate the performance of those devices.

As you design your cluster, account for the user requirements of each of the applications and services, and choose the appropriate level of storage to provide the solution to meet those requirements. Having a mix of storage tiers is common and provides the ability to balance cost versus performance.

With Fibre Channel (FC) technology, you efficiently can provide storage in a shared environment to reduce the cost of storage by sharing it across multiple clusters. FC is described in 3.3.5, “Fibre Channel” on page 87.

3.3.5 Fibre Channel

FC is a high-speed SAN connection that is used for connecting storage, disk, and tape to the processors. The FC specification provides for line speeds of 1 - 64 Gb. At the time of writing, a line speed of 128 Gb is planned.

FC protocol provides an in-order and lossless delivery for raw block data. This protocol provides a high-performance channel for connecting your storage.

The FC protocol provides a switched environment (SAN), enabling the sharing of ports with multiple devices and flexible distances between connected devices. This protocol also enables sharing of devices between different processors for scalability and migration capability. An FC network can provide low latency connections to shared storage (the latencies often are as low as or lower than direct-connected disk in your servers). By connecting to external storage, FC provides an extra layer of flexibility and availability and can move some of the processing power that is required for data replication out of the processor into the SAN storage, which releases that processing power to your applications.

FC block devices can connect to your Red Hat OpenShift cluster through the CSI driver in Red Hat OpenShift, which is supported by the storage devices. By using the CSI driver, you leverage the flexibility and availability features of SAN storage seamlessly in your Red Hat OpenShift cluster.

3.3.6 Network File System

NFS is a mechanism for sharing files across multiple servers or clusters over a network connection. NFS is a low-cost solution for sharing files among different applications because it does not require special connections to storage.

NFS is a good solution for applications with low IOPS and latency requirements, but can be a challenge when those requirements grow. The performance of NFS is less than any class of storage (for more information, see 3.3.3, “I/O operations per second” on page 86 and 3.3.4, “Tier 1 and Tier 3 storage” on page 86) and should be used with caution for high-volume applications.

3.3.7 Network

Your network connections contribute to higher network speeds that provide lower latency and support more users. However, higher network speeds are expensive.

Choose network connections that can fulfill your user’s expectations. Monitor network utilization as you scale users and applications, and add more network capacity as needed.

With technologies like single root I/O virtualization (SR-IOV), your design for network connectivity can be more flexible. For more information about SR-IOV, see 3.3.8, “Single root I/O virtualization” on page 88.

3.3.8 Single root I/O virtualization

With SR-IOV, you can have multiple virtual servers whose operating systems simultaneously share a PCIe adapter with little or no runtime involvement from a hypervisor or other virtualization intermediary.

SR-IOV enables virtualization without hypervisor interaction. It is a successor of Integrated Virtual Ethernet adapter (IVE), which was available in some IBM POWER7 processor-based servers. SR-IOV is a newer technology that is based on physical adapters, but in many IBM Power processor-based servers, SEA is the standard for virtualizing network adapters.

You can leverage SR-IOV technology through VIOSs, where the case client partitions use virtual Network Interface Controllers (vNICs) that are based on SR-IOV-capable adapters that are configured in the VIOSs.

Table 3-5 shows the key differences of the Ethernet network virtualization technologies.

Table 3-5 Ethernet network virtualization technologies

Technology	Live Partition Mobility	Quality of service	Direct-access performance	Redundancy options	Server-side Failover	Requires VIOS
SR-IOV	No ^a	Yes	Yes	Yes ^b	No	No
vNIC	Yes	Yes	No ^c	Yes ^b	vNIC Failover	Yes
SEA / vEth	Yes	No	No	Yes	SEA Failover	Yes
Hybrid Network Virtualization	Yes	Yes	Yes	Yes	No	No

a. SR-IOV can be combined with VIOS and virtual Ethernet to use higher-level virtualization functions like LPM, but the client partition does not receive performance or quality of service (QoS) benefits.

b. Some limitations apply. For more information, see this [FAQ document](#).

c. Better performance and requires fewer system resources compared to SEA or virtual Ethernet.

For more information about SR-IOV in IBM Power servers, see *IBM Power Systems SR-IOV: Technical Overview and Introduction*, REDP-5065 and this [FAQ document](#).

Benefits of SR-IOV

SR-IOV can provide direct access to the adapter hardware without control or data flow going through the hypervisor. Depending on the adapter type, there is a maximum of logical ports, but the technology provides an improved partition per PCI slot ratio.

On IBM Power servers, SR-IOV provides QoS controls to set the capacity values for each logical port, which enables prioritization of partition traffic.

Sharing physical adapters can reduce the cost due to consolidation, and there is no extra CPU and memory usage compared to SEAs.

Red Hat OpenShift and SR-IOV

SR-IOV is supported on IBM Power processor-based Red Hat OpenShift clusters on specific physical network interfaces.¹⁰ In IBM Power servers with Red Hat OpenShift running on them, there are multiple ways to use SR-IOV.

- ▶ Using a bare metal server as an Red Hat OpenShift node and sharing the SR-IOV-capable PCI adapter between pods.
- ▶ Using a VIOS and configuring virtual servers (LPARs) as Red Hat OpenShift nodes by using vNICs, which are configured on shared SR-IOV-capable physical adapters that are set up in VIOSs.

Bonding network interfaces is supported on Red Hat OpenShift pods to combine multiple SR-IOV virtual function (VF) interfaces, which can increase the available network bandwidth or availability for the pods.

It is also possible to configure OVS hardware offloading, which can increase data processing performance. This feature is available on compatible bare metal Red Hat OpenShift nodes. Offloading removes data processing tasks from the CPU and transfers the data to dedicated units of network interface controllers, which increase transfer rate and reduce the load on the CPU.

Red Hat OpenShift supports Ethernet and InfiniBand device attachment by using SR-IOV.

¹⁰ https://docs.openshift.com/container-platform/4.11/networking/hardware_networks/about-sriov.html#supported-devices_about-sriov

In Red Hat OpenShift, SR-IOV is configured and managed by the SR-IOV Network Operator. The operator creates and manages the components by completing the following steps.

1. Discovers the SR-IOV network devices that are available in nodes.
2. Generates NetworkAttachmentDefinition CRs for the SR-IOV Container Network Interface (CNI).
3. Creates and updates the configuration of the SR-IOV network device plug-in.
4. Creates a node-specific SrioNetworkNodeState CR.
5. Updates the spec.interfaces field in each SrioNetworkNodeState CR.

There is a daemonset that is deployed to every worker node by the operator to discover and initialize the SR-IOV network devices. Other plug-ins discover, advertise, and allocate VF resources into pods.

For more information about installation and the initial configuration, see the [Red Hat OpenShift documentation](#).

After installing the operator, we can configure whether it drains the nodes and injects the configuration automatically.

The SR-IOV Network Operator discovers devices and creates the SrioNetworkNodeState CR for the worker nodes where there is a compatible adapter, as shown in Example 3-10.

Example 3-10 SrioNetworkNodeState without and with an SR-IOV capable physical adapter

```
(py39) [root@build-cp4d-1 ~]# oc get SrioNetworkNodeState -A
NAMESPACE                                NAME                                     AGE
openshift-sriov-network-operator         worker1.cp4d-1.rtp.raleigh.ibm.com     23h
openshift-sriov-network-operator         worker2.cp4d-1.rtp.raleigh.ibm.com     23h
openshift-sriov-network-operator         worker3.cp4d-1.rtp.raleigh.ibm.com     23h
openshift-sriov-network-operator         worker4.cp4d-1.rtp.raleigh.ibm.com     23h
openshift-sriov-network-operator         worker8.cp4d-1.rtp.raleigh.ibm.com     23h

(py39) [root@build-cp4d-1 ~]# oc get SrioNetworkNodeState worker1.cp4d-1.rtp.raleigh.ibm.com -n
openshift-sriov-network-operator -o yaml
apiVersion: srioNetworkOperator.openshift.io/v1
kind: SrioNetworkNodeState
metadata:
  creationTimestamp: "2022-11-15T08:58:39Z"
  generation: 1
  name: worker1.cp4d-1.rtp.raleigh.ibm.com
  namespace: openshift-sriov-network-operator
  ownerReferences:
  - apiVersion: srioNetworkOperator.openshift.io/v1
    blockOwnerDeletion: true
    controller: true
    kind: SrioNetworkNodePolicy
    name: default
    uid: bb671997-fee9-4010-82b5-16280dbfb592
  resourceVersion: "33330348"
  uid: 2a6d0ed4-9818-42ef-8049-eea2bbfa08f1
spec:
  dpConfigVersion: "33329739"
status: {}

(py39) [root@build-cp4d-1 ~]# oc get SrioNetworkNodeState worker8.cp4d-1.rtp.raleigh.ibm.com -n
openshift-sriov-network-operator -o yaml
apiVersion: srioNetworkOperator.openshift.io/v1
kind: SrioNetworkNodeState
metadata:
  creationTimestamp: "2022-11-15T08:58:39Z"
  generation: 1
```

```

name: worker8.cp4d-1.rtp.raleigh.ibm.com
namespace: openshift-sriov-network-operator
ownerReferences:
- apiVersion: sriovnetwork.openshift.io/v1
  blockOwnerDeletion: true
  controller: true
  kind: SriovNetworkNodePolicy
  name: default
  uid: bb671997-fee9-4010-82b5-16280dbfb592
resourceVersion: "33330574"
uid: 625aaf01-3818-477f-b6a4-15e84d9c86c9
spec:
  dpConfigVersion: "33329739"
status:
  interfaces:
  - deviceID: "1657"
    driver: tg3
    linkSpeed: 1000 Mb/s
    linkType: ETH
    mac: 08:94:ef:80:98:7e
    mtu: 1500
    name: enP5p1s0f0
    pciAddress: "0005:01:00.0"
    vendor: "14e4"
  - deviceID: "1657"
    driver: tg3
    linkSpeed: -1 Mb/s
    linkType: ETH
    mac: 08:94:ef:80:98:7f
    mtu: 1500
    name: enP5p1s0f1
    pciAddress: "0005:01:00.1"
    vendor: "14e4"
  - deviceID: "1015"
    driver: mlx5_core
    linkSpeed: 10000 Mb/s
    linkType: ETH
    mac: b8:ce:f6:df:ba:14
    mtu: 1500
    name: enP48p1s0f0
    pciAddress: "0030:01:00.0"
    totalvfs: 8
    vendor: 15b3
  - deviceID: "1015"
    driver: mlx5_core
    linkSpeed: -1 Mb/s
    linkType: ETH
    mac: b8:ce:f6:df:ba:15
    mtu: 1500
    name: enP48p1s0f1
    pciAddress: "0030:01:00.1"
    totalvfs: 8
    vendor: 15b3
  syncStatus: Succeeded

```

As Example 3-10 on page 90 shows, if the node is not supported for SR-IOV, then there are no interfaces that are listed in the Status section. The operator sets the following label for supported nodes:

```
“feature.node.kubernetes.io/network-sriov.capable: true”
```

The actual configurations of VFs to pods are controlled by the SriovNetworkNodePolicy CR.

Note: When a configuration is applied to a `SriovNetworkNodePolicy` CR, the nodes can be drained and restarted, depending on the `SriovOperatorConfig` CR.

With this CR, you can select on what nodes what type of adapters, or specifically which adapters, are configured for SR-IOV. Here, you can set whether remote direct memory access (RDMA) is enabled or not. The `resourceName` parameter also is set, which helps attach the `SriovNetwork` (created later) and pod to the VFs of SR-IOV.

After the `SriovNetworkNodePolicy` CR is processed by the operator, the `SriovNetworkNodeStates` resource is updated for each supported node with the same name as the node. This resource contains all supported interfaces with the vendor code, device ID, and physical location codes.

After you create the `SriovNetwork` CR, build the base of a new additional network, which can be assigned to pods in annotations. It is possible to set transmission rate limits and specify valid IP address ranges, DNS, and gateway settings.

The SR-IOV operator creates a `NetworkAttachmentDefinition` with the same name as the `SriovNetwork`, which can be used in the additional network configuration for pods, as shown in Example 3-11.

Example 3-11 Adding a pod to additional networks

```
metadata:
  annotations:
    k8s.v1.cni.cncf.io/networks: |-
      [
        {
          "name": "<network>",
          "namespace": "<namespace>",
          "default-route": ["<default-route>"]
        }
      ]
```

Red Hat OpenShift creates an annotation that is named `k8s.v1.cni.cncf.io/network-status` that is based on the additional network configuration that you set.

3.3.9 Partition mobility

Partition mobility, a component of the PowerVM Enterprise Edition hardware feature, migrates AIX, IBM i, and Linux LPARs from one system to another one. The mobility process transfers the system environment, which includes the processor state, memory, attached virtual devices, and connected users.

The following types of migrations are available.

- ▶ *Active partition migration*, or LPM migrating AIX, IBM i, and Linux LPARs that are running, including the operating system and applications, from one system to another one. The LPAR and the applications that are running on that migrated LPAR do not need to be shut down.
- ▶ *Inactive partition migration*, or cold partition mobility to migrate a powered off AIX, IBM i, or Linux LPAR from one system to another one.

Use cases for partition mobility

Partition mobility provides systems management flexibility, and it can improve system availability. Here are some examples where partition mobility can help you:

- ▶ Server consolidation: Migrate and consolidate partitions from many servers to fewer ones with higher capacity.
- ▶ Workload balancing: Distribute partitions between servers to share and balance the load, or migrate partitions to servers with specific hardware resources that are needed by the applications.
- ▶ Evacuating servers for planned maintenance: When you prepare for hardware, firmware, or VIOS upgrades or changes, which might require an outage, the partitions can be migrated uninterrupted to other servers, and then migrated back after successful upgrade, which avoids a planned outage.
- ▶ Migrating from older technology (IBM POWER8 and later) to newer technology: Certain new hardware features might require an operating system restart.

Note: Partition mobility does not provide automatic workload balancing, and it is not a replacement of HA or DR solutions.

Using VM remote restart in case of failed partitions or systems can provide higher availability for your systems. This feature is available in IBM PowerVC. For more information about this product, see [IBM PowerVC Documentation](#).

Processor compatibility between an LPM source and target

Processor compatibility modes enable you to migrate LPARs between servers that have different processor types without upgrading the operating environments that are installed in the LPARs.

For more information about supported scenarios, see [IBM Power Documentation](#).

Prerequisites

Both source and target systems must have an IBM PowerVM Enterprise Edition license that is activated, and both source and target partitions must be fully virtualized (no physical adapter or interface is attached) at the time of the migration.

The target system must provide the same virtual resources as the source system, and it must be connected to the same network with synchronized Time of Day clocks of the VIOSs. It is possible to migrate a partition to another system that is managed by a different HMC. The target VIOSs must be able to receive the same configuration for the migrated partition as on the source system, so VLAN IDs and subnets should match.

The SAN and external storage systems must be prepared to allocate the same storage volumes to the migrated partition on the target system. Virtual adapters cannot be marked as required and should not be marked for “any client”.

The processor mode must be compatible between the source and target systems. The processor compatibility mode is a value that is assigned to an LPAR by the hypervisor that specifies the processor environment in which the LPAR can successfully operate.

The migration can be started from a CLI, or the GUI of the HMC. The process uses SSH connections, so SSH must be configured and defined with SSH key authentications to the remote HMC and all involved LPARs (VIOS and actual LPAR).

For more information about active partition migration compatibility mode combinations, see the following resources:

- ▶ [For active partitions](#)
- ▶ [For inactive partitions](#)

For more information, see [configuration validation](#).

Note: If a partition has physical resources that are attached, then they must be deallocated before the migration, but they can be added back on the target system. So, you can create a workaround manually.

Migration phases

The migration phases can be different for active partition migration and for inactive partition migration.

Here are the active partition migration phases:

1. Validate the configuration.
2. Create an LPAR on the target server.
3. Create virtual resources on the target server.
4. Migrate the state of the LPAR in memory.
5. Remove the old LPAR configuration from the source server.
6. Free the old resources on the source server.

Here are the inactive partition migration phases:

1. Validate the configuration.
2. Create an LPAR on the target server.
3. Create virtual resources on the target server.
4. Remove the old LPAR configuration from the source server.
5. Free the old resources on the source server.

Red Hat OpenShift considerations

A production-ready Red Hat OpenShift cluster has its own HA features, including an etcd cluster on three master nodes, which is sensitive to the connectivity between the master nodes. This configuration keeps the etcd cluster in sync. For more information about Red Hat OpenShift HA, see 4.4.5, “Disaster recovery” on page 116.

It is possible to migrate partitions with Red Hat OpenShift nodes running on them because the switchover usually takes less than a couple of seconds. However, as a best practice, migrate master nodes one by one to ensure that no quorum loss can happen in the etcd cluster as the result of the migration.



Red Hat OpenShift architecture and design

Red Hat OpenShift is a leading enterprise Kubernetes platform that offers a consistent hybrid cloud foundation for building, deploying, and scaling containerized applications. This integrated platform can be used to run, orchestrate, monitor, and scale containerized workloads while helping maximize developer productivity with specially configured tool sets. These tools provide functions such as continuous integration and continuous delivery (CI/CD) pipelines, and source-to-image (S2I) build capability. This chapter explores the architecture and layout of a Red Hat OpenShift environment.

This chapter contains the following topics:

- ▶ Design considerations for Red Hat OpenShift
- ▶ Red Hat OpenShift capabilities on IBM Power
- ▶ IBM Cloud Paks capabilities
- ▶ Red Hat OpenShift architecture
- ▶ Red Hat OpenShift ecosystem
- ▶ Running Red Hat OpenShift on IBM Power

4.1 Design considerations for Red Hat OpenShift

Container-based computing is an important trend in today's environment, and there are many options that are available when you choose the infrastructure to run your cloud-native applications. Although container-based infrastructures can run on many different infrastructures, there are differences in how those different architectures perform. Choosing the right infrastructure can increase your client experience and reduce the overall cost of the resulting infrastructure.

Thus, Red Hat OpenShift Container Platform architecture planning and design are important to fulfilling your business requirements. The rest of this chapter presents information that shows how Red Hat OpenShift and IBM Power servers can be used to implement a cloud solution to meet your business requirements.

4.2 Red Hat OpenShift capabilities on IBM Power

Clients are demanding exceptional customer experiences, which is driving organizations to develop applications to meet customer expectations and modernize existing applications to accelerate their cloud-native journey. DevOps teams require a flexible and agile development approach, and they are faced with challenges to deploy the applications across multiple infrastructures, ranging from on-premises to the public cloud.

Red Hat OpenShift and IBM Cloud Paks on IBM Power provide developers a consistent and secure platform to innovate continuously with a skill set that is common across various platforms, including IBM Power, with more reliability, adaptability, and performance. Red Hat OpenShift on IBM Power offers flexibility and choice for various cloud consumption models across physical, virtual, private, and public clouds. It also provides scalability and leverages the added security that is built in to IBM Power servers to provide highly secure, cloud-based environments in a hybrid cloud for cloud-native development.

Red Hat OpenShift on IBM Power provides several advantages, including scalability (of both Red Hat OpenShift and IBM Power), a pay-per-use consumption model, and low latency. Some of the advantages of combining Red Hat OpenShift and IBM POWER processors are shown in the following list:

- ▶ Red Hat OpenShift enables scalability to thousands of instances across hundreds of nodes in seconds. This scalability is enhanced because IBM Power can scale the underlying infrastructure up and down based on demand.
- ▶ With built-in virtualization, IBM Power dynamically adds or removes memory and CPUs that are allocated to worker node virtual machines (VMs).
- ▶ IBM Power provides a pay-per-use consumption model in both on-premises and off-premise environments.
- ▶ By collocating cloud-native apps with existing VM-based apps running on AIX, IBM i, or Linux environments, IBM Power servers enable low-latency connections between apps and data.
- ▶ IBM Power Systems Virtual Server (IBM PowerVS) runs leading business applications like SAP HANA in an IBM Power based cloud.

Red Hat OpenShift on IBM Power provides a strong foundation that is built for security and reliability. For example, Live Partition Mobility (LPM) can provide uninterrupted access to critical data and applications. The IBM Power compute infrastructure reduces unplanned downtime with less than 2 minutes per year, which results in several advantages, including improved productivity for IT teams and reducing the impact on critical business processes and users.

Red Hat OpenShift on IBM Power helps optimize infrastructure usage and costs by reducing the number of servers that is needed without impacting performance; dynamically allocating cores to busy worker nodes in shared processor pools; and collocating containerized applications on the IBM Power server with AIX, IBM i data, which reduces the number of servers and the latency that is experienced by applications connecting to those legacy environments.

The built-in agility of Red Hat OpenShift and IBM Power is extended to a truly hybrid cloud model through IBM PowerVS. IBM PowerVS is an enterprise infrastructure as a service (IaaS) offering that is built on IBM Power servers that are colocated in IBM Cloud data centers, and it offers access to over hundreds of IBM Cloud services. Red Hat OpenShift is available on IBM PowerVS through a platform-neutral installer.

For workloads on Red Hat OpenShift on IBM Power, the solution offers a lower overall total cost of ownership (TCO) and greater throughput for service-level agreements (SLAs).¹

4.3 IBM Cloud Paks capabilities

Building containerized applications from scratch requires investment in cloud resources, talent, and management tools. Because there is a shortage of cloud-native skills, and short project timelines (businesses want solutions delivered immediately), IBM customers are seeking enterprise-grade and preintegrated software to accelerate digital transformation and innovation. IBM Cloud Paks are artificial intelligence (AI)-powered software that is designed for the hybrid cloud landscape.

To support the most complex projects and initiatives, IBM Cloud Paks have built-in collaboration and intelligent workflows across multiple stakeholders to streamline communications and project management. IBM Cloud Paks help enterprises overcome obstacles that are introduced with the new application and operational complexity of multicloud environments.

IBM Cloud Paks have the following features:

- ▶ **Portable and can run anywhere:** The portability of hybrid cloud solutions that are built with IBM Cloud Paks means that they are built to run on any hybrid cloud environment. They can run on an on-premises infrastructure, on a public hybrid cloud infrastructure, or in an integrated system by leveraging a common set of Kubernetes skills.
- ▶ **Certified and secure:** IBM Cloud Paks are certified by IBM, with high standards and up-to-date vulnerability scanning software to provide cloud security protection of sensitive data and full-stack support from hardware to applications.
- ▶ **Consumable:** IBM Cloud Paks are preintegrated to deliver use cases like application deployment and process automation for your DevOps teams, and they are priced and packaged for cost savings so that companies pay for what they use.

¹ <https://www.ibm.com/downloads/cas/26A60NY>

IBM Cloud Paks on IBM Power leverage the optimized hardware and improvements of the most powerful IBM processor in the market and several advantages of Red Hat OpenShift, including scalability, low latency, security, and reliability to provide AI-powered software that is designed to accelerate application modernization with preintegrated data, automation, and security capabilities.

To enable business and IT teams to build and modernize applications, IBM Cloud Paks have several features and are grouped into different Paks:

- ▶ IBM Cloud Pak for WebSphere Hybrid Edition
- ▶ IBM Cloud Pak for Integration (IBM CP4I)
- ▶ IBM Cloud Pak for Watson AIOps
- ▶ IBM Cloud Pak for Business Automation (IBM CP4BA)
- ▶ IBM Cloud Pak for Data

Figure 4-1 shows the capabilities of each of the IBM Cloud Paks that are available on IBM Power.

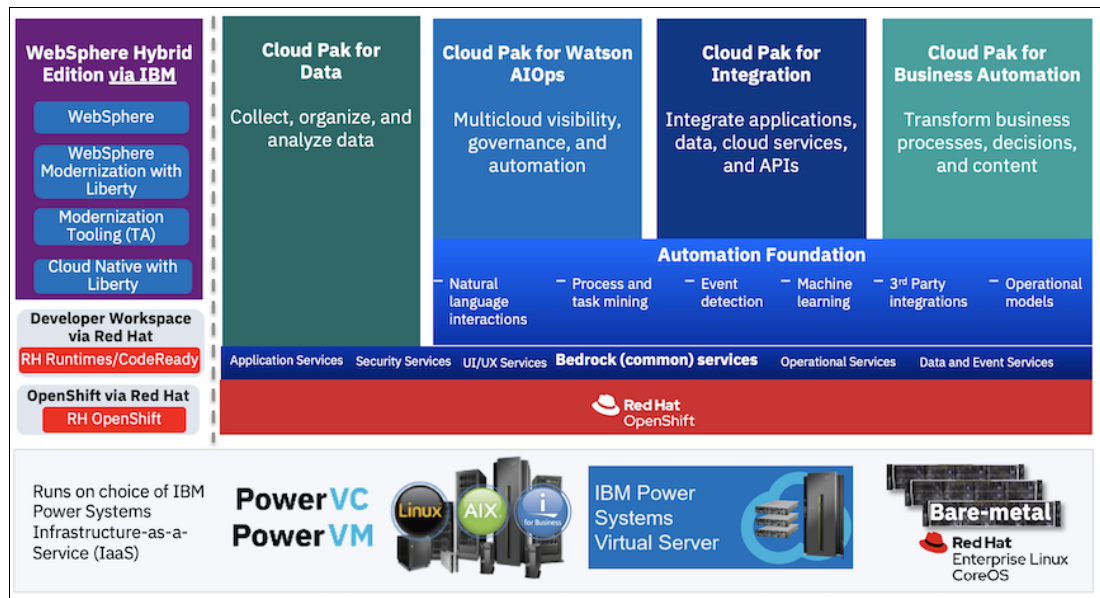


Figure 4-1 IBM Cloud Paks on IBM Power

Table 4-1 lists the IBM Cloud Paks capabilities that are available on IBM Power servers.

Table 4-1 IBM Cloud Pak capabilities that are available on IBM Power servers

IBM Cloud Pak	Capabilities that are available on IBM Power servers
IBM Cloud Pak for Data	IBM Db2® Warehouse
	IBM Db2 Advanced
	Data Management Console
	IBM Watson Machine Learning Accelerator (GPU support)
IBM Cloud Pak for WebSphere Hybrid Edition	Transformation Advisor (tool)
	Mobile Foundation (Traditional)
	IBM WebSphere Application Server
	IBM WebSphere Application Server Liberty
	IBM WebSphere Application Server ND
IBM CP4I	IBM MQ and IBM MQ Advanced
	App Connect Enterprise
	Platform Navigator
	Event Streams (Kafka)
	App Connect Designer
IBM Cloud Pak for Watson AIOps	Red Hat Advanced Cluster Management for Kubernetes: Manage-to IBM Power
	Red Hat Advanced Cluster Management for Kubernetes: Manage-from IBM Power
	IBM Cloud Pak for Watson AIOps: Infra Automation: Manage-to IBM Power
	IBM Cloud Pak for Watson AIOps: Infra Automation: Manage-from IBM Power (IBM Cloud Pak for Multicloud Management (IBM Cloud Pak for Multicloud Management) 2.3)
	IBM Instana: VM observability for AIX, IBM i, and Linux
	IBM Instana: Container observability through Red Hat OpenShift Container Platform on IBM Power
	IBM Turbonomic: Manage to IBM Power (containers)

IBM Cloud Pak	Capabilities that are available on IBM Power servers
IBM CP4BA	Operational Decision Manager (ODM)
	Business Automation Workflow (BAW)
	IBM FileNet® Content Manager (FNCM)
	Enterprise Records (ER)
	Business Automation Insights (BAI)
	Business Automation Studio (BAS)
	Application Designer (AD)
	Automation Decision Services (ADS)

For more information about IBM Cloud Paks, see Chapter 5, “IBM Cloud Paks on Red Hat OpenShift running on IBM Power” on page 149, or the following resources:

- ▶ [IBM Cloud Paks](#)
- ▶ [IBM Cloud Pak for Integration](#)
- ▶ [Infuse your AIOps platform with intelligent IT operations](#)
- ▶ [IBM Cloud Pak for Business Automation](#)
- ▶ [IBM Cloud Pak for Data](#)

4.4 Red Hat OpenShift architecture

This section provides an overview of Red Hat OpenShift and its underlying architecture and components. Running Red Hat OpenShift on IBM Cloud or on-premises provides your developers with a fast and secure way to containerize and deploy cloud-ready workloads in Kubernetes clusters.

Red Hat OpenShift Container Platform is a cloud-based Kubernetes container platform that you use to develop and run containerized applications. It is designed so that applications and the hosting providers that support them can scale from, for example, a small cluster with a few machines and applications to a cluster with thousands of machines that serve millions of users.

By using Kubernetes, Red Hat OpenShift Container Platform incorporates the same technology that serves as the engine for massive telecommunications, streaming video, gaming, banking, and other applications. The platform provides a common base for hosting applications to meet any industry need. Its implementation of open Red Hat technologies extends your containerized applications beyond a single cloud to on-premises and multi-cloud environments.

Red Hat OpenShift Container Platform provides enterprise-ready enhancements to Kubernetes:

- ▶ Hybrid model cloud deployments: You can deploy Red Hat OpenShift Container Platform clusters to various public cloud platforms or to your private cloud.
- ▶ Integrated Red Hat technology: Major components in Red Hat OpenShift Container Platform come from Red Hat Enterprise Linux and related Red Hat technologies. Red Hat OpenShift Container Platform benefits from the testing and certification initiatives for Red Hat enterprise quality software.
- ▶ Open-source development model: Development is completed in the open, and the source code is available from public software repositories. This open collaboration fosters rapid innovation and development.

Although Kubernetes excels at deploying your applications, it does not provide a full management suite to manage your infrastructure. Powerful and flexible platform management tools and processes are important benefits that Red Hat OpenShift Container Platform 4.10 offers. The following sections describe some unique features and benefits of Red Hat OpenShift Container Platform.

Architecture diagram

Figure 4-2 shows the various components of Red Hat OpenShift Container Platform and how developers and administrators interact with the cluster.

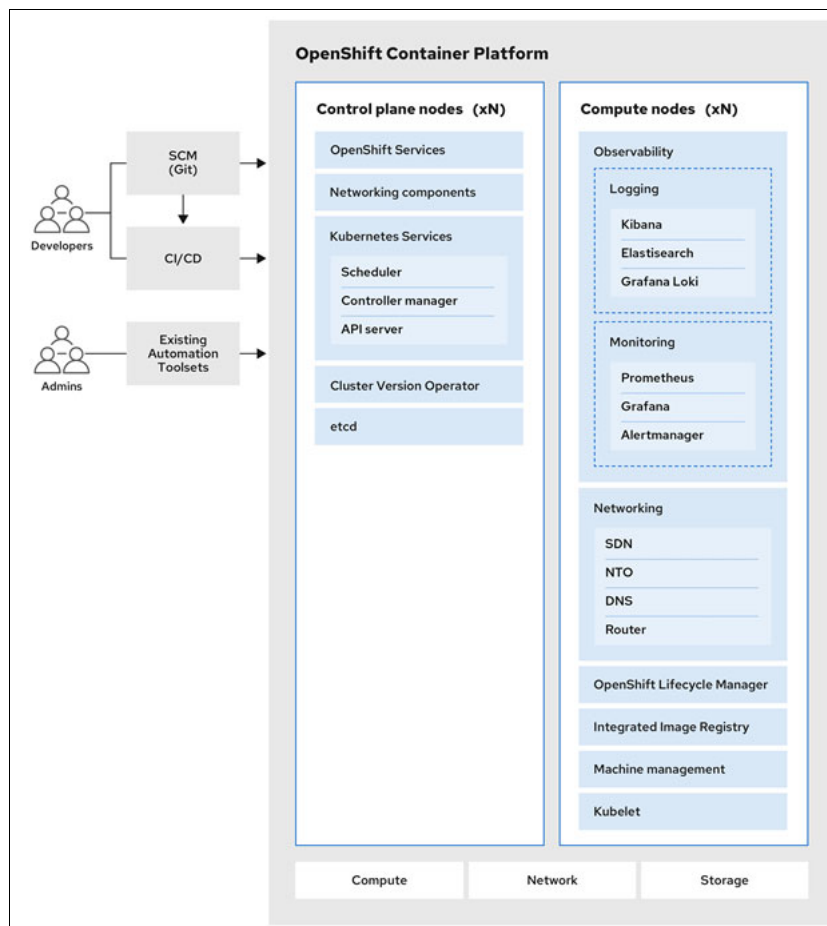


Figure 4-2 Red Hat OpenShift Container Platform architecture

There are various components in Red Hat OpenShift Container Platform that run on the underlying physical or cloud infrastructure that are divided into the compute plane, which provides application support, monitoring, and networking, for example, and the control plane, which provides management, security, and other services.

4.4.1 Enterprise Kubernetes

Red Hat OpenShift is based on Enterprise Kubernetes, which is a platform that is built by experts in Kubernetes and container technology who are driving key capabilities upstream.

Although container images and the containers that run from them are the primary building blocks for modern application development, running them at scale requires a reliable and flexible distribution system. Kubernetes is the de-facto standard for orchestrating containers.

Kubernetes is an open source container orchestration engine for automating deployment, scaling, and management of containerized applications. The general concept of Kubernetes is simple:

- ▶ Start with few or more worker nodes to run the container workloads.
- ▶ Manage the deployment of those workloads from one or more control plane nodes.
- ▶ Seal containers in a deployment unit called a *pod*. Pods provide extra metadata with the container and can group several containers in a single deployment entity.
- ▶ Create special kinds of assets. For example, services are represented by a set of pods and a policy that defines how they are accessed. This policy enables containers to connect to the services that they need even if they do not have the specific IP addresses for the services. Replication controllers are another special asset that indicates how many pod replicas are required to run concurrently. You can use this capability to automatically scale your application to adapt to its current demand.

Within a few years, Kubernetes has seen massive cloud and on-premises adoption. The open-source development model enables many people to extend Kubernetes by implementing different technologies for components, such as networking, storage, and authentication.

The benefits of containerized applications

Using containerized applications offers many advantages over using traditional deployment methods. Traditionally, applications were installed on operating systems that included all their dependencies, but containers contain an application and its dependencies. Creating containerized applications offers many benefits.

Operating system benefits

Containers use small, dedicated Linux operating systems without a kernel. Their file system, networking, cgroups, process tables, and namespaces are separate from the host Linux system, but the containers can integrate with the hosts seamlessly when necessary. Being based on Linux enables containers to use all the advantages that come with the open-source development model of rapid innovation.

Because each container uses a dedicated operating system, you can deploy applications that require conflicting software dependencies on the same host. Each container contains its own dependent software and manages its own interfaces, such as networking and file systems, so applications never compete for those assets.

Deployment and scaling benefits

If you employ rolling upgrades between major releases of your application, you can continuously improve your applications without downtime, and still maintain compatibility with the current release.

You can deploy and test a new version of an application alongside the existing version. If the container passes your tests, deploy more new containers and remove the old ones.

Similarly, scaling containerized applications is simple. Red Hat OpenShift Container Platform offers a simple, standard way of scaling any containerized service. For example, if you build applications as a set of microservices rather than large, monolithic applications, you can scale the individual microservices individually to meet demand. Therefore, you can scale only the required services instead of the entire application and meet application demands while using minimal resources.

Red Hat CoreOS optimized operating system

Red Hat OpenShift Container Platform uses Red Hat Enterprise Linux CoreOS (RHCOS), a container-oriented operating system that is designed for running containerized applications from Red Hat OpenShift Container Platform and works with new tools to provide fast installation, Operator-based management, and simplified upgrades.

Red Hat CoreOS includes the following features:

- ▶ Ignition, which Red Hat OpenShift Container Platform uses as a firstboot system configuration for initially starting and configuring machines.
- ▶ CRI-O, a Kubernetes native container runtime implementation that integrates closely with the operating system to deliver an efficient and optimized Kubernetes experience. CRI-O provides facilities for running, stopping, and restarting containers. It replaces the Docker Container Engine, which was used in Red Hat OpenShift Container Platform 3.
- ▶ Kubelet, the primary node agent for Kubernetes that is responsible for launching and monitoring containers.

In Red Hat OpenShift Container Platform 4.10, you must use Red Hat CoreOS for all control plane machines, but you can use Red Hat Enterprise Linux as the operating system for compute machines, which are also known as worker machines. If you choose to use Red Hat Enterprise Linux workers, you must perform more system maintenance than if you use Red Hat CoreOS for all the cluster machines.

Simple installation and update process

With Red Hat OpenShift Container Platform 4.10, if you have an account with the right permissions, you can deploy a production cluster in supported clouds by running a single command and providing a few values. You can customize your cloud installation or install your cluster in your data center if you use a supported platform.

For clusters that use Red Hat CoreOS for all machines, updating or upgrading Red Hat OpenShift Container Platform is a simple, highly automated process. Because Red Hat OpenShift Container Platform controls the systems and services that run on each machine, including the operating system itself, from a central control plane, upgrades are designed to become automatic events. If your cluster contains Red Hat Enterprise Linux worker machines, the control plane benefits from the streamlined update process, but you must perform more tasks to upgrade the worker machines running Red Hat Enterprise Linux.

Other key features

Operators are both the fundamental unit of the Red Hat OpenShift Container Platform 4.10 code base and a convenient way to deploy applications and software components for your applications to use. In Red Hat OpenShift Container Platform, Operators serve as the platform foundation and remove the need for manual upgrades of operating systems and control plane applications. Red Hat OpenShift Container Platform Operators, such as the Cluster Version Operator and Machine Config Operator, enable simplified, cluster-wide management of those critical components.

Red Hat Operator Lifecycle Manager (OLM) and the OperatorHub provide facilities for storing and distributing Operators to people developing and deploying applications.

The Red Hat Quay Container Registry is a Quay.io container registry that serves most of the container images and Operators for Red Hat OpenShift Container Platform clusters. Quay.io is a public registry version of Red Hat Quay that stores millions of images and tags.

Other enhancements to Kubernetes in Red Hat OpenShift Container Platform include improvements in software-defined networking (SDN), authentication, log aggregation, monitoring, and routing. Red Hat OpenShift Container Platform also offers a comprehensive web console and the custom Red Hat OpenShift command-line interface (CLI) (**oc**) interface.

Red Hat OpenShift Container Platform lifecycle

Figure 4-3 illustrates the basic Red Hat OpenShift Container Platform lifecycle:

- ▶ Creating an Red Hat OpenShift Container Platform cluster
- ▶ Managing the cluster
- ▶ Developing and deploying applications
- ▶ Scaling up applications

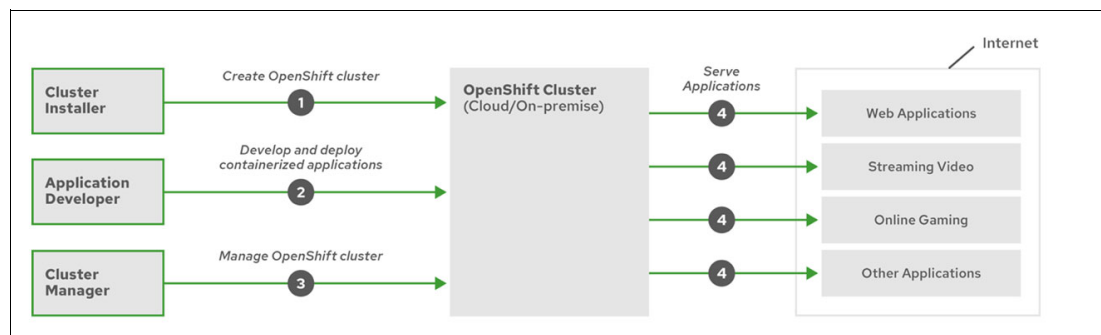


Figure 4-3 Red Hat OpenShift Container Platform lifecycle

In Red Hat OpenShift on IBM Cloud, your clusters are composed of an IBM managed master that secures components, such as the application programming interface (API) server and etcd, and customer-managed worker nodes that you configure to run your app workloads and Red Hat OpenShift provided default components. The default components within the cluster, such as the Red Hat OpenShift web console or OperatorHub, vary with the Red Hat OpenShift version of your cluster.

4.4.2 Classic Red Hat OpenShift 4 components

Figure 4-4 provides an architectural overview of a classic Red Hat OpenShift environment in IBM Cloud. It is composed of master nodes, which are managed by IBM, and worker nodes, which are managed by customers. Master nodes are used to run critical management functions in the cluster, and worker nodes run your applications.

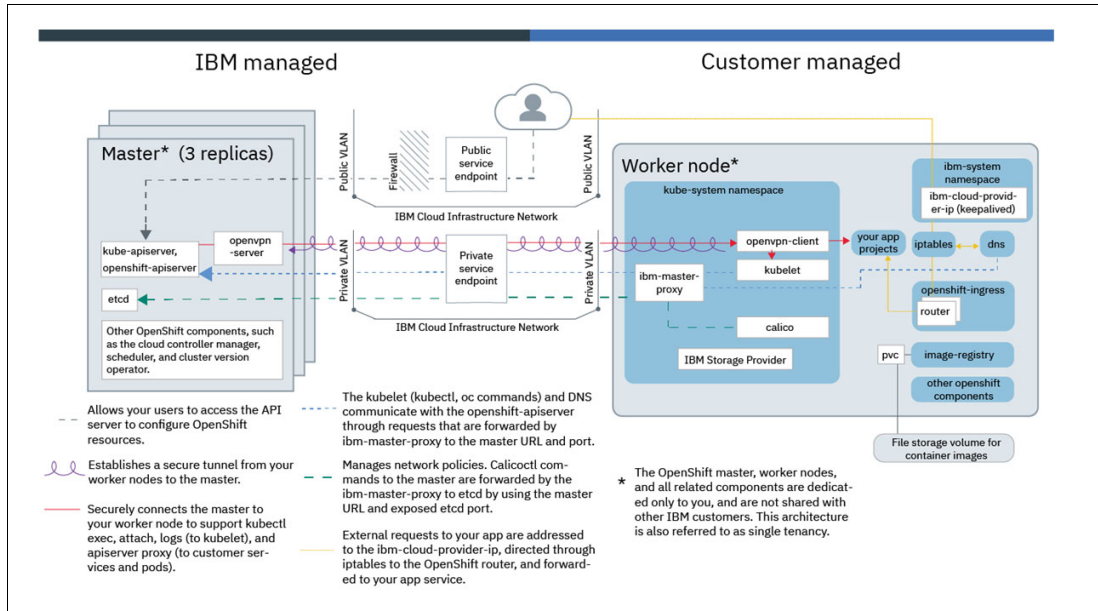


Figure 4-4 Red Hat OpenShift 4 master components

When you run `oc get nodes`, you might notice that the ROLES of your worker nodes are marked as both `master,worker`. These nodes are worker nodes in IBM Cloud, and they do not include the master components that are managed by IBM. Instead, these nodes are marked as `master` because they run Red Hat OpenShift Container Platform components that are required to set up and manage default resources within the cluster, such as the OperatorHub and internal registry.

Red Hat OpenShift 4 master components

The master nodes run the management components like the API server and etcd, among other critical management functions. You cannot modify these components. IBM manages the components and automatically updates them during master patch updates.

Master components, including the apiserver and etcd, have three replicas and are spread across zones for even high availability (HA). The master and all the master components are dedicated only to you, and are not shared with other IBM customers.

Red Hat OpenShift 4 worker node components

Figure 4-4 also shows the components that run in the worker nodes in your cluster in IBM Cloud.

These components run on your worker nodes because you can use them with the workloads that you deploy to your cluster. For example, your apps might use an Operator from the OperatorHub that runs a container from an image in the internal registry. You are responsible for your usage of these components, but IBM provides updates for them in the worker node patch updates that you choose to apply.

In Red Hat OpenShift Container Platform 4, many components are configured by a corresponding operator for ease of management.

With Red Hat OpenShift on IBM Cloud, the VMs that your cluster manages are instances that are called *worker nodes*. However, the underlying hardware is shared with other IBM customers. You manage the worker nodes through the automation tools that are provided by Red Hat OpenShift on IBM Cloud, such as the API, CLI, or a console. Unlike classic clusters, you do not see Virtual Private Cloud (VPC) compute worker nodes in your infrastructure portal or a separate infrastructure bill, but manage all maintenance and billing activity for the worker nodes from Red Hat OpenShift on IBM Cloud.

Single tenancy

The worker nodes and all worker node components are dedicated only to you, and are not shared with other IBM customers. However, if you use worker node VMs, the underlying hardware might be shared with other IBM customers depending on the level of hardware isolation that you choose.

Operating system

Worker nodes run on the Red Hat Enterprise Linux 7 or Red Hat Enterprise Linux 8 operating system.

- ▶ For cluster version 4.10 and later, only Red Hat Enterprise Linux 8 is supported.
- ▶ For cluster version 4.9, you can choose Red Hat Enterprise Linux 7 or Red Hat Enterprise Linux 8, but the default operating system is Red Hat Enterprise Linux 7.
- ▶ For cluster versions 4.8 and earlier, only Red Hat Enterprise Linux 7 is supported.

Cluster networking

Your worker nodes are created in a VPC subnet in the zone that you specify. Communication between the master and worker nodes is over the private network. If you create a cluster with the public and private cloud service endpoints enabled, authenticated external users can communicate with the master over the public network, such as to run `oc` commands. If you create a cluster with only the private cloud service endpoints enabled, authenticated external users can communicate with the master over the private network only. You can set up your cluster to communicate with resources in on-premises networks, other VPCs, or a classic infrastructure by setting up a VPC VPN, IBM Cloud Direct Link, or IBM Cloud Transit Gateway on the private network.

App networking

VPC load balancers automatically are created in your VPC outside the cluster for any networking services that you create in your cluster. For example, a VPC load balancer exposes the router services in your cluster by default, or you can create a Kubernetes LoadBalancer service for your apps, and a VPC load balancer is automatically generated. VPC load balancers are multi-zone and route requests for your app through the private node ports that automatically are opened on your worker nodes. If the public and private cloud service endpoints are enabled, the routers and VPC load balancers are created as public by default. If only the private cloud service endpoint is enabled, the routers and VPC load balancers are created as private by default. For more information, see [About IBM Cloud Network Load Balancer for VPC](#). Calico is used as the cluster networking policy fabric.

4.4.3 Red Hat OpenShift Local (formerly Red Hat CodeReady Containers)

Red Hat OpenShift Local (formerly known as Red Hat CodeReady Containers) might be a good choice for developers to build Red Hat OpenShift clusters in a sandbox environment. Red Hat OpenShift Local is designed to run on a local computer to simplify setup and testing, and to emulate the cloud development environment locally with all the tools that are needed to develop container-based applications.

Red Hat OpenShift Local components

The latest version of Red Hat OpenShift Local 2.5 ships with the Red Hat OpenShift versions of the main components, as shown in Table 4-2.

Table 4-2 Red Hat OpenShift Local components

Component	Version
Red Hat OpenShift Container Platform	4.10.18 or later
Red Hat OpenShift client binary (oc)	4.10.18 or later
Podman binary	4.1.0 or later

Minimum system requirements

Table 4-3 shows the minimum hardware and operating system requirements for Red Hat OpenShift Local.

Table 4-3 Minimum system requirements for Red Hat OpenShift Local

Operating system	CPU	Memory	Disk storage	Hardware architectures
Microsoft Windows 10 Fall Creators Update (Version 1709) or later	4	9 GB	35 GB	AMD64 and Intel 64 (x86_64)
MacOS 11 Big Sur or later.	4	9 GB	35 GB	Intel 64 (x86_64) or ARM-based M1 (aarch64)
Red Hat Enterprise Linux or CentOS 7, 8, and 9 minor releases	4	9 GB	35 GB	AMD64 and Intel 64 (x86_64)

Table 4-4 shows the Podman container run times.

Table 4-4 Minimum system requirements for Podman

Operating system	CPU	Memory	Disk storage	Hardware architectures
Microsoft Windows 10 Fall Creators Update (Version 1709) or later	2	2 GB	35 GB	AMD64 and Intel 64 (x86_64)
MacOS 11 Big Sur or later	2	2 GB	35 GB	Intel 64 (x86_64) or ARM-based M1 (aarch64)
Red Hat Enterprise Linux or CentOS 7, 8, and 9 minor releases	2	2 GB	35 GB	AMD64 and Intel 64 (x86_64)

For more information, see [Getting Started Guide for Red Hat OpenShift Local](#).

Installing Red Hat OpenShift Local

This section shows you how to use Red Hat OpenShift Local in your local system, such as a laptop or a VM running with Red Hat Enterprise Linux 8. Complete the following steps:

1. Download the latest release of Red Hat OpenShift Local from the [Red Hat download site](#), as shown in Figure 4-5.

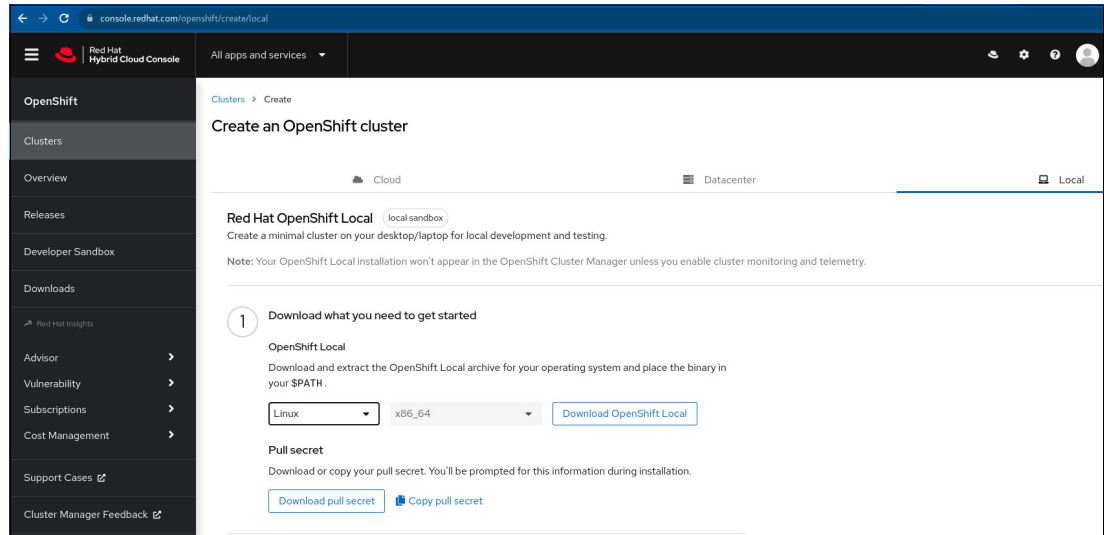


Figure 4-5 Creating an Red Hat OpenShift cluster

2. Copy the downloaded file to the `~/` directory and extract the contents of the file, as shown in Example 4-1.

Example 4-1 Downloading and extracting the Red Hat Local installation file

```
$ cd ~/Downloads
$ mkdir -p ~/bin
$ tar xvf crc-linux-amd64.tar.xz -C ~/bin
```

3. Add the `~/bin` directory to your `$PATH` environment variable, as shown in Example 4-2.

Example 4-2 Setting the `$PATH` variable

```
$ export PATH=$PATH:$HOME/bin
$ echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc
```

Initial setup for Red Hat OpenShift Local

Before you start the initial setup, make sure that your system is configured with a local rpm repository so that it can install the required rpms. Complete the following steps:

1. To set up the environment of your system for the Red Hat OpenShift Local instance, run the `crc setup` command, as shown in Example 4-3.

Example 4-3 The `crc setup` command and sample output

```
$ crc setup
CRC is constantly improving and we would like to know more about usage (more
details at https://developers.redhat.com/article/tool-data-collection)
Your preference can be changed manually if wanted using 'crc config set
consent-telemetry <yes/no>'
Would you like to contribute anonymous usage statistics? [y/N]: y
Thanks for helping us! You can disable telemetry with the command 'crc config set
consent-telemetry no'.
INFO Using bundle path /home/user/.crc/cache/crc_libvirt_4.11.7_amd64.crcbundle
INFO Checking if running as non-root
INFO Checking if running inside WSL2
INFO Checking if crc-admin-helper executable is cached
INFO Caching crc-admin-helper executable
INFO Using root access: Changing ownership of
/home/user/.crc/bin/crc-admin-helper-linux
```

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

```
[sudo] password for user:
INFO Using root access: Setting suid for
/home/user/.crc/bin/crc-admin-helper-linux
INFO Getting bundle for the CRC executable
3.15 GiB / 3.15 GiB
[----->_____] 69.60% 1.07
MiB p/s
```

Note: When you run the `crc setup` command for the first time, it prompts you to enable the telemetry for usage data collection. You can disable or enable the telemetry by running `crc config set consent-telemetry no` (disable) or `crc config set consent-telemetry yes` (enable) commands. Enabling or disabling telemetry does not modify a running instance. The change takes effect the next time that you run the `crc start` command.

Tip: Make sure that the DNS client is configured in the file `/etc/resolv.conf` and that it can resolve to the required Red Hat websites, or you see the following error messages:

```
INFO Getting bundle for the CRC executable
Get
"https://mirror.openshift.com/pub/openshift-v4/clients/crc/bundles/openshift/4.
11.7/crc_libvirt_4.11.7_amd64.crcbundle": dial tcp: lookup
mirror.openshift.com: no such host
```

2. To start the Red Hat OpenShift Local instance after the setup completes successfully, run the command that is shown in Example 4-4.

Example 4-4 The crc setup command and sample output

```
3.15 GiB / 3.15 GiB
[-----]
100.00% 1.07 MiB p/s
INFO Decompressing /home/ansible/.crc/cache/crc_libvirt_4.11.7_amd64.crcbundle
crc.qcow2: 12.01 GiB / 12.01 GiB
[-----] 100.00%
oc: 118.14 MiB / 118.14 MiB
[-----] 100.00%
Your system is correctly setup for using CRC. Use 'crc start' to start the
instance
```

Starting Red Hat OpenShift Local

To start Red Hat OpenShift Local, complete the following steps:

1. Start the new Red Hat OpenShift Local instance as shown in Example 4-5.

Example 4-5 The crc start command and sample output

```
$ crc start
INFO Checking if running as non-root
INFO Checking if running inside WSL2
INFO Checking if crc-admin-helper executable is cached
INFO Checking for obsolete admin-helper executable
INFO Checking if running on a supported CPU architecture
INFO Checking minimum RAM requirements
INFO Checking if crc executable symlink exists
INFO Checking if Virtualization is enabled
INFO Checking if KVM is enabled
INFO Checking if libvirt is installed
INFO Checking if user is part of libvirt group
INFO Checking if active user/process is currently part of the libvirt group
INFO Checking if libvirt daemon is running
INFO Checking if a supported libvirt version is installed
INFO Checking if crc-driver-libvirt is installed
INFO Checking crc daemon systemd socket units
INFO Checking if systemd-networkd is running
INFO Checking if NetworkManager is installed
INFO Checking if NetworkManager service is running
INFO Checking if /etc/NetworkManager/conf.d/crc-nm-dnsmasq.conf exists
INFO Checking if /etc/NetworkManager/dnsmasq.d/crc.conf exists
INFO Checking if libvirt 'crc' network is available
INFO Checking if libvirt 'crc' network is active
INFO Loading bundle: crc_libvirt_4.11.7_amd64...
CRC requires a pull secret to download content from Red Hat.
You can copy it from the Pull Secret section of
https://console.redhat.com/openshift/create/local.
? Please enter the pull secret
*****
WARN Cannot add pull secret to keyring: The name org.freedesktop.secrets was not
provided by any .service files
```

Note: The first time that you run the `crc start` command, you are prompted to provide the pull secret, which you can copy from the [Red Hat download site](#).

2. When you start the Red Hat OpenShift Local instance, you are prompted for the login credentials and the URLs for both CLI or web-based GUI access at the end of the output. A sample output is shown in Example 4-6.

Example 4-6 The `crc start` command and sample output

```
INFO Creating CRC VM for openshift 4.11.7...
INFO Generating new SSH key pair...
INFO Generating new password for the kubeadmin user
INFO Starting CRC VM for openshift 4.11.7...
INFO CRC instance is running with IP 192.168.130.11
INFO CRC VM is running
INFO Updating authorized keys...
INFO Configuring shared directories
INFO Check internal and public DNS query...
INFO Check DNS query from host...
INFO Verifying validity of the kubelet certificates...
INFO Starting kubelet service
INFO Waiting for kube-apiserver availability... [takes around 2min]
INFO Adding user's pull secret to the cluster...
INFO Updating SSH key to machine config resource...
INFO Waiting for user's pull secret part of instance disk...
INFO Changing the password for the kubeadmin user
INFO Updating cluster ID...
INFO Updating root CA cert to admin-kubeconfig-client-ca configmap...
INFO Starting openshift instance... [waiting for the cluster to stabilize]
INFO 2 operators are progressing: image-registry, openshift-controller-manager
INFO 2 operators are progressing: image-registry, openshift-controller-manager
INFO 2 operators are progressing: image-registry, openshift-controller-manager
INFO Operator openshift-controller-manager is progressing
INFO Operator openshift-controller-manager is progressing
INFO Operator openshift-controller-manager is progressing
INFO Operator openshift-controller-manager is progressing
INFO All operators are available. Ensuring stability...
INFO Operators are stable (2/3)...
INFO Operators are stable (3/3)...
INFO Adding crc-admin and crc-developer contexts to kubeconfig...
Started the Red Hat OpenShift cluster.
```

The server is accessible via web console at:
`https://console-openshift-console.apps-crc.testing`

Log in as administrator:
Username: kubeadmin
Password: 4vf9r-KevUw-7m8QD-qE3ry

Log in as user:
Username: developer
Password: developer

Use the 'oc' CLI:
\$ eval \$(crc oc-env)
\$ oc login -u developer https://api.crc.testing:6443

If the cluster is not ready, you receive the error messages that are shown in Example 4-7.

Example 4-7 Cluster not ready error sample output

```
ERRO Cluster is not ready: cluster operators are still not stable after
10m1.285389239s
INFO Adding crc-admin and crc-developer contexts to kubeconfig...
```

Tip: The first time that you start a Red Hat OpenShift Local instance, it might fail because of the predefined waiting time for the Red Hat OpenShift Local instance readiness. The time that is taken by the system to get your Red Hat OpenShift Local instance ready might take longer than the predefined waiting time, depending on your system. You still can log in to the Red Hat OpenShift Local instance and verify which cluster operators are not ready yet.

Logging in to Red Hat OpenShift Local

You can log in to the Red Hat OpenShift Local instance either from a CLI or web browser.

Examples of using a CLI to work on your cluster

Here are examples of the commands that you can use to work on your Red Hat OpenShift Local cluster.

Logging in to the Red Hat OpenShift Local instance by using the oc command

Logging in by using a CLI is shown in Example 4-8.

Example 4-8 Logging in to Red Hat OpenShift cluster by using the oc command and sample output

```
$ eval $(crc oc-env)
$ oc login -u kubeadmin https://api.crc.testing:6443
Logged in to "https://api.crc.testing:6443" as "kubeadmin" using existing
credentials.
```

You have access to 66 projects. The list has been suppressed. You can list all projects with 'oc projects'
Using project "default".

Verifying the cluster operators status

Example 4-9 shows how to verify the status of your cluster operators.

Example 4-9 Cluster operator status and sample output

```
$ oc get co
```

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE	MESSAGE
authentication	4.11.3	True	False	False	23m	
config-operator	4.11.3	True	False	False	35d	
console	4.11.3	True	False	False	46h	
dns	4.11.3	True	False	False	46h	
etcd	4.11.3	True	False	False	35d	
image-registry	4.11.3	True	False	False	46h	
ingress	4.11.3	True	False	False	35d	
kube-apiserver	4.11.3	True	False	False	35d	
kube-controller-manager	4.11.3	True	False	False	35d	
kube-scheduler	4.11.3	True	False	False	35d	
machine-api	4.11.3	True	False	False	35d	
machine-approver	4.11.3	True	False	False	35d	

machine-config	4.11.3	True	False	False	35d
marketplace	4.11.3	True	False	False	35d
network	4.11.3	True	False	False	35d
node-tuning	4.11.3	True	False	False	35d
openshift-apiserver	4.11.3	True	False	False	46h
openshift-controller-manager	4.11.3	True	False	False	34d
openshift-samples	4.11.3	True	False	False	35d
operator-lifecycle-manager	4.11.3	True	False	False	35d
operator-lifecycle-manager-catalog	4.11.3	True	False	False	35d
operator-lifecycle-manager-packageserver	4.11.3	True	False	False	29m
service-ca	4.11.3	True	False	False	35d

Verifying node status

Example 4-10 shows the command to validate the status of the nodes in your cluster.

Example 4-10 Node status and sample output

```
$ oc get no
NAME                STATUS    ROLES    AGE   VERSION
crc-wkzjw-master-0 Ready    master,worker 35d   v1.24.0+b62823b
```

Using your web browser to manage your cluster

You can log in to your Red Hat OpenShift Local instance by using your web browser, as shown in Figure 4-6.

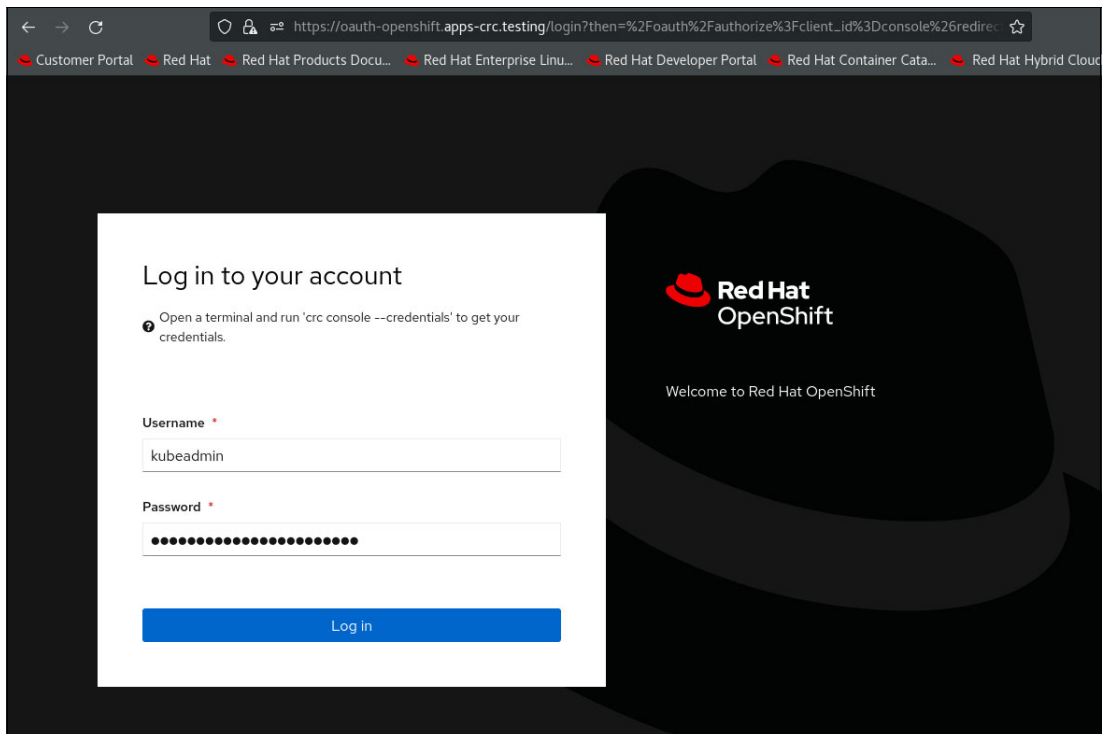


Figure 4-6 Red Hat OpenShift login

Then, you can verify your cluster operators status, as shown in Figure 4-7.

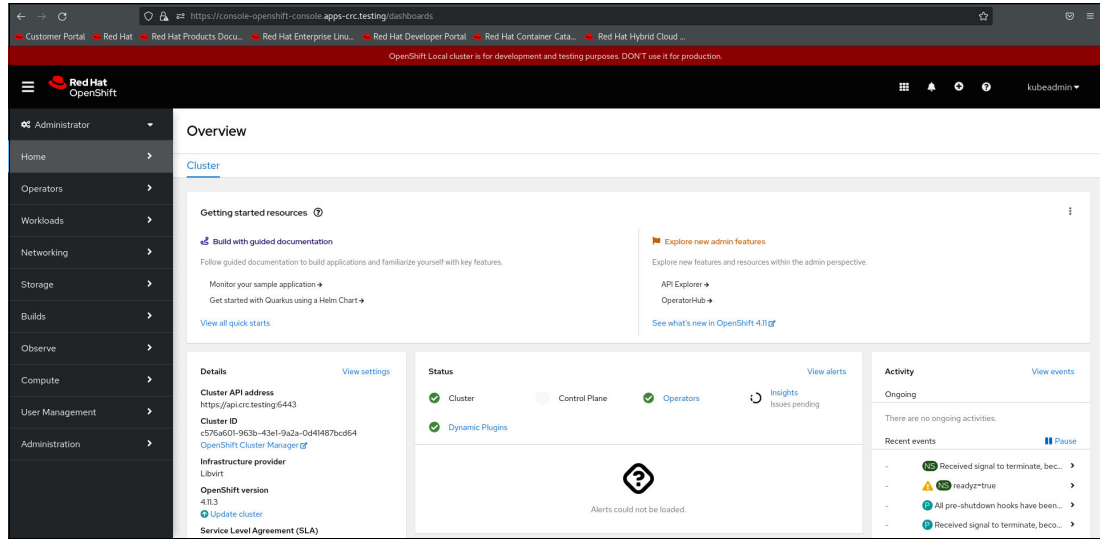


Figure 4-7 Cluster overview

Starting, stopping, or deleting the Red Hat OpenShift Local instance

To start, stop, or delete your local cluster instance:

- ▶ To start the new Red Hat OpenShift Local instance, run `$ crc start`.
- ▶ To stop the Red Hat OpenShift Local instance and container run time, run `$ crc stop`.
- ▶ To delete the existing Red Hat OpenShift Local instance, run `$ crc delete`.

4.4.4 High availability for master nodes

The Red Hat OpenShift cluster architecture consists of multiple types of roles for the Red Hat OpenShift nodes, which are the master or control plane nodes, worker or compute nodes, and infrastructure nodes, which are special worker nodes that are designated for infrastructure workloads.

Master node or control plane

The Red Hat OpenShift Container Platform master node is a server that performs control functions for the entire Red Hat OpenShift Container Platform cluster environment. The master nodes form the control plane that is responsible for the creation, scheduling, and management of all objects that are specific to the Red Hat OpenShift Container Platform cluster. The key components of the Red Hat OpenShift Container Platform that run on the Master node include the following components:

- ▶ Kubernetes apiserver
- ▶ Scheduler
- ▶ Cluster Management
- ▶ Red Hat OpenShift apiserver
- ▶ Operator Lifecycle Management
- ▶ Web Console
- ▶ etcd

Worker node or compute node

The Red Hat OpenShift worker nodes run containerized applications that are created and deployed by developers. Some worker nodes can be used for specific workloads that are considered an infrastructure-related workload. For example:

- ▶ Some Red Hat OpenShift worker nodes can be used explicitly for container storage solutions, such as Red Hat OpenShift Data Foundation.
- ▶ Red Hat OpenShift worker nodes can be used explicitly for logging, monitoring, and image registry.
- ▶ Red Hat OpenShift worker nodes can be used explicitly for a default router or sharding.

Other uses of master nodes

The Red Hat OpenShift Container Platform master nodes can run applications like a worker node while retaining its master role. For example, you can run the Red Hat OpenShift Container Platform infrastructure components on the master nodes to reduce the number of worker nodes that are required to run infrastructure components.

Three-node Red Hat OpenShift Container Platform

You can define a Red Hat OpenShift cluster with only three nodes. This cluster consists of three control plane or master nodes that also act as worker and infrastructure nodes. This smaller, 3-node Red Hat OpenShift cluster provides a resource efficient cluster for cluster administrators and developers to use for testing, development, and small production environments. For more information, see this [Red Hat OpenShift document](#).

Why the master node is critical

The master node forms the control plane that is responsible for the creation, scheduling, and management of all objects that are specific to Red Hat OpenShift. It includes the API, the controller manager, and the scheduler capabilities in one Red Hat OpenShift binary file.

For your cluster to operate, you must have at least one healthy control plane host that has a master node.

High availability for the master node

The Red Hat OpenShift Container Platform cluster can be installed and configured in HA mode, which uses multiple master nodes, or in non-HA mode, which uses a single master node. A single-node cluster has more restrictive resource constraints, and might not be supported on all hardware platforms.

The Red Hat OpenShift Container Platform cluster HA mode requires exactly three master nodes, which are used to maintain three replica copies of the critical components, for example, apiserver, etcd, and controller manager across three master nodes.

Note: Exactly three control plane nodes must be used for all production deployments in Red Hat OpenShift Container Platform cluster HA mode. At the time of writing, it is not possible to run with two masters or five masters. For more information, see the following resources:

- ▶ [Is it possible to scale master / etcd nodes in Red Hat OpenShift 4?](#)
- ▶ [Cluster masters](#)

4.4.5 Disaster recovery

Disaster recovery (DR) is one of key requirements when planning business continuity in your Red Hat OpenShift environment. Define and document plans for recovering your Red Hat OpenShift environment when failures of infrastructure or other site-related failures cause your environment to go down. Some of the failures in your Red Hat OpenShift Container Platform to consider and plan for are the following ones:

- ▶ Red Hat OpenShift Container Platform perspective:
 - Lose most of your control plane hosts (master node), leading to etcd quorum loss and the cluster going offline.
 - Accidental deletion of critical cluster data or configuration.
- ▶ Application workload perspective while running on Red Hat OpenShift Container Platform:
 - Less than the minimum worker node threshold nodes are available and your application goes offline.
 - Accidental deletion of critical configuration for a stateless application.
 - Accidental deletion of critical data and configuration for a stateful application.
- ▶ Data center outages, either planned or unplanned.

Overcoming different disaster scenarios

The different disaster situations in Red Hat OpenShift Container Platform require different precautions to overcome the disaster situations. The following documents provide solutions for recovering from the issues that are described above:

- ▶ [How to replace all master nodes in Red Hat OpenShift Container Platform 4](#)
- ▶ [Ignition fails adding new nodes to UPI cluster after upgrading to Red Hat OpenShift Container Platform 4.6+](#)

Backup and restore

Backup and restore is a traditional and effective method of recovery. To recover your cluster with a restore, back up all the critical configuration data for the Red Hat OpenShift Container Platform and the application data.

To back up cluster data, run the following command:

```
$ /usr/local/bin/cluster-backup.sh /home/core/assets/backup
```

Alternatively, you can use any backup and restore utility that supports Red Hat OpenShift Container Platform. IBM Spectrum Protect Plus is one option that you can use to back up Red Hat OpenShift container data directly to cloud storage. This data can include persistent volumes (PVs), namespace-scoped resources, and application-consistent data.

Replacing services and nodes in Red Hat OpenShift Container Platform

You can replace unhealthy service nodes and other nodes that are not running or are in the NotReady state in the Red Hat OpenShift Container Platform. For example, you might need to replace an unhealthy bare metal etcd member whose node is not running or is in a NotReady state.

Disaster recovery site

A DR site or secondary site must provide a location to recover your environment in a primary site failure due to planned maintenance or perhaps a natural disaster such as fire, tornado, or hurricane. The DR site must have the required infrastructure to run your environment, and it might be another site in your enterprise or even a site that is provided by a cloud vendor.

Most Red Hat OpenShift environments require persistent storage to store the data that is required for the applications to run. Recovering your environment requires that your data is at your recovery site. For faster recovery, the data must be consistently replicated from your primary site to your secondary location, which requires the appropriate storage infrastructure to support that replication. Backup and restore of the data can be an option for those applications that have lower recovery time objectives (RTOs).

Reference architectures

The IBM Academy of Technology created some reference articles to use in your Red Hat OpenShift solution design, found at this [IBM Cloud Architecture](#) website.

Reference architecture 1: Starter environment

This topology is the minimal one for a Red Hat OpenShift Container Platform 4.x for stateless and ephemeral workloads. It is shown in Figure 4-8.

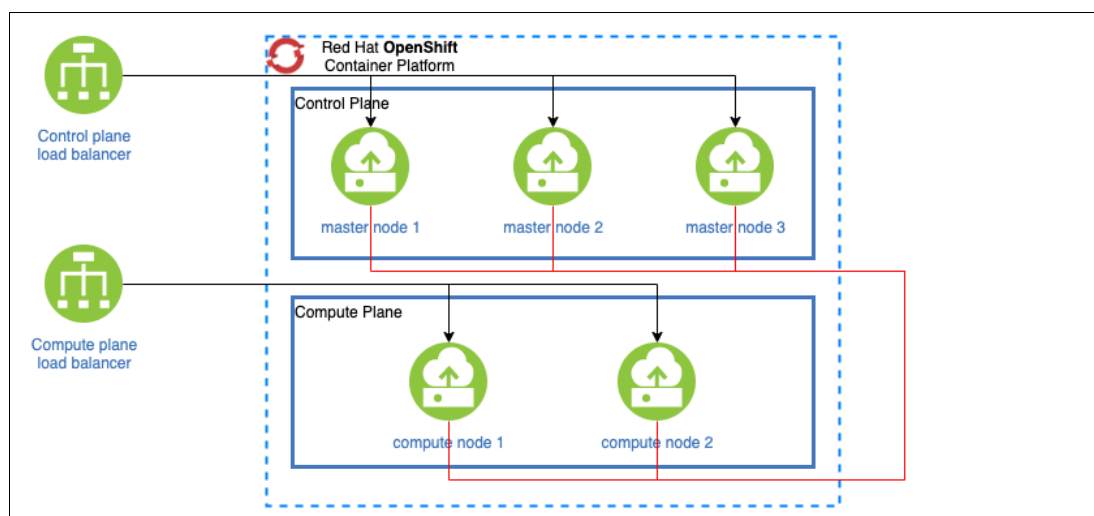


Figure 4-8 Starter environment

- ▶ Number of nodes that are required:
 - Three master nodes
 - Two worker nodes
- ▶ Application workloads:
 - Stateless
 - Ephemeral workloads
- ▶ Use cases:
 - Development
 - Test

Consideration: There is no persistent storage in this solution and no HA.

Reference architecture 2: 3-node cluster

This reference architecture, which is shown in Figure 4-9, was first available in Red Hat OpenShift Container Platform 4.5. At the time of writing, it is supported only for deployment on bare metal.

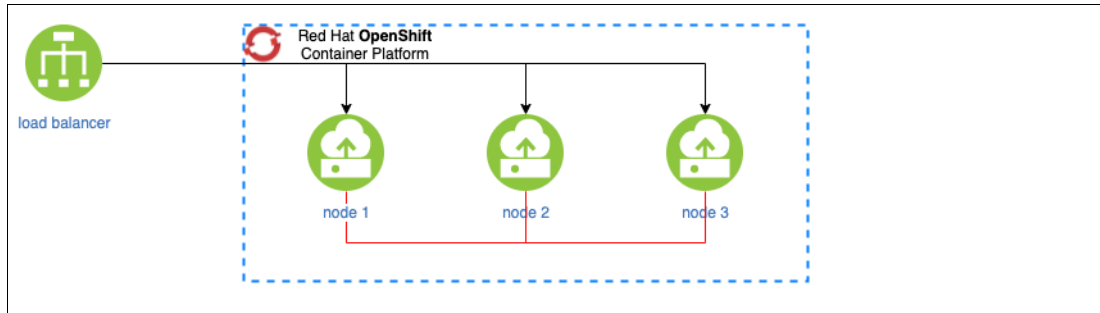


Figure 4-9 Three-node cluster

- ▶ Number of nodes that are required: Three master nodes. The master and work functions are colocated.
- ▶ Application workloads:
 - Stateless
 - Ephemeral workloads
- ▶ Use cases:
 - Requirement for limited footprint:
 - Development
 - Test

Consideration: There is no persistent storage in this solution and no HA.

Reference architecture 3: On-premises cluster

This cluster, which is shown in Figure 4-10, is deployed in one availability zone (AZ) or data center.

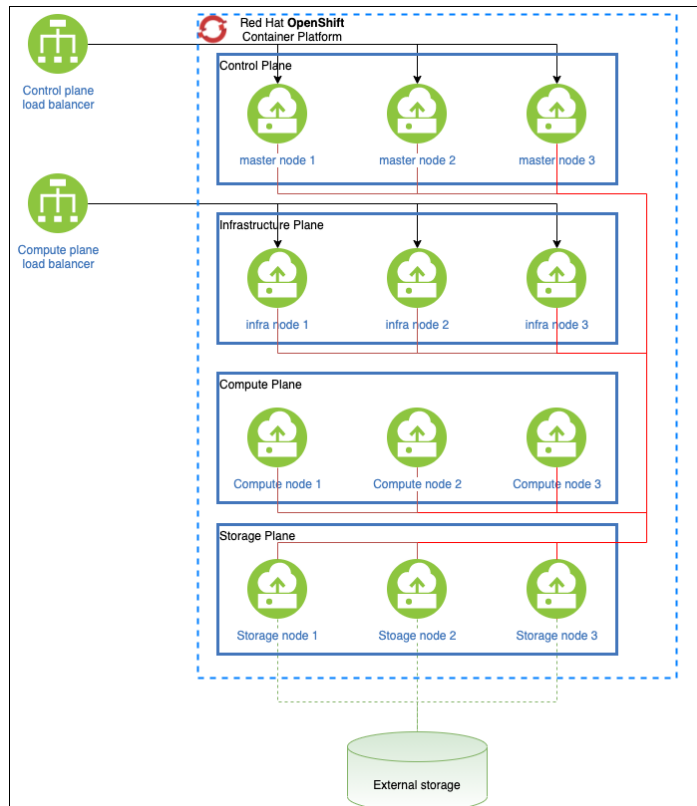


Figure 4-10 On-premises cluster

- ▶ Number of nodes required:
 - Three master nodes
 - Three worker nodes
 - Three infrastructure nodes (Elasticsearch requires three instances, that is, one per node)
 - Three storage nodes
- ▶ Application workloads: Any (subject to a limited SLA)
- ▶ Use cases:
 - Development
 - Test
 - Integration
 - Production (subject to a limited SLA)

Consideration: There is no HA in this design.

Note: The number of nodes and their sizing can be adjusted depending on the workload target.

Reference architecture 4: Cloud cluster

This design, which is shown in Figure 4-11, provides an HA cluster. The nodes must be spread across multiple AZs, which are locations with different power or storage connections that are designed to not have a single point of failure in common with the other AZs.

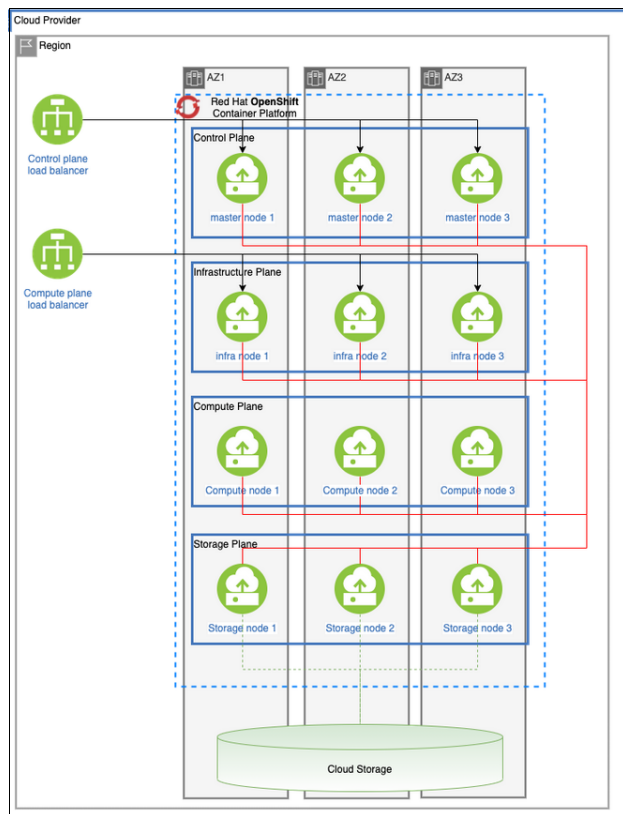


Figure 4-11 Cloud cluster

- ▶ Deploys a cluster over three AZs.
- ▶ Number of nodes that are required:
 - Three master nodes
 - Three worker nodes
 - Three infrastructure nodes (Elasticsearch requires three instances, with one per node)
- ▶ Application workloads: Any (subject to a limited SLA for DR)
- ▶ Use cases:
 - Development
 - Test
 - Integration
 - Production (subject to a limited SLA for DR)

Consideration: Requires an infrastructure with different AZs. This architecture provides for a HA cluster but does not consider DR with a second site.

Note: The number of nodes and their sizing can be adjusted depending on the workload target.

Reference architecture 5: Two clusters on-premises

This configuration, which is shown in Figure 4-12, provides a HA solution for your on-premises environment. It requires two data centers to provide the availability. The distance between the data centers influences the options that are available for data replication and the recovery point objective (RPO) of the solution. In each data center, the architecture looks like the one that is used in “Reference architecture 3: On-premises cluster” on page 119. Whether this architecture satisfies DR requirements depends on the separation of the two data centers.

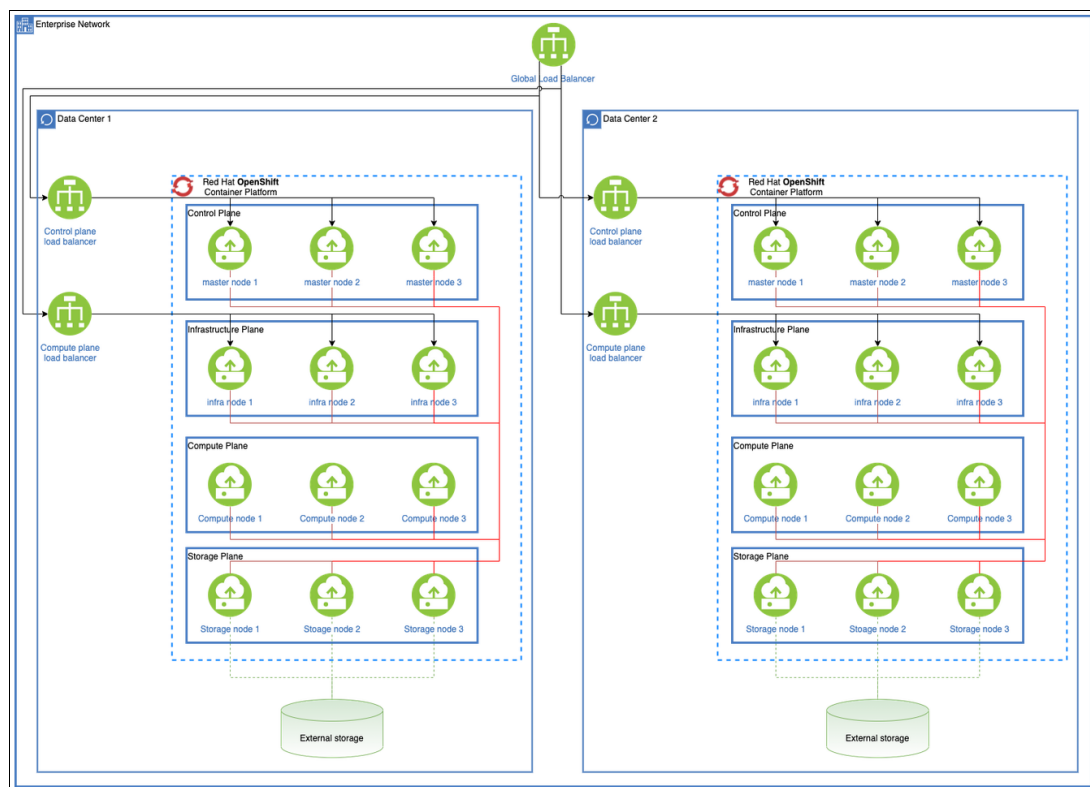


Figure 4-12 Two clusters on-premises

- ▶ Deploy clusters in different data centers. Two data centers required. Minimum of two clusters, with one per data center.
- ▶ Number of nodes per clusters:
 - Three master nodes
 - Three worker nodes
 - Three infrastructure nodes (Elasticsearch requires three instances, with one per node)
 - Three storage nodes
- ▶ Application workloads: Any.
- ▶ Use cases: Production: On-premises. Development and test can coexist.

Consideration: The replication methods, whether using middleware or product-native replication or storage-based replication, and the DR topology (Active/Active or Active/Passive) should be decided based on the capabilities of the deployed products, workload, middleware, and storage. Supported backup and restore can be done, but it reflects on your RTO.

Reference architecture 6: two clusters on two regions

This configuration, which is shown in Figure 4-13, is the cloud equivalent of “Reference architecture 5: Two clusters on-premises” on page 121. It requires a cloud provider with two distinct regions, and uses two clusters, with one cluster in each region.

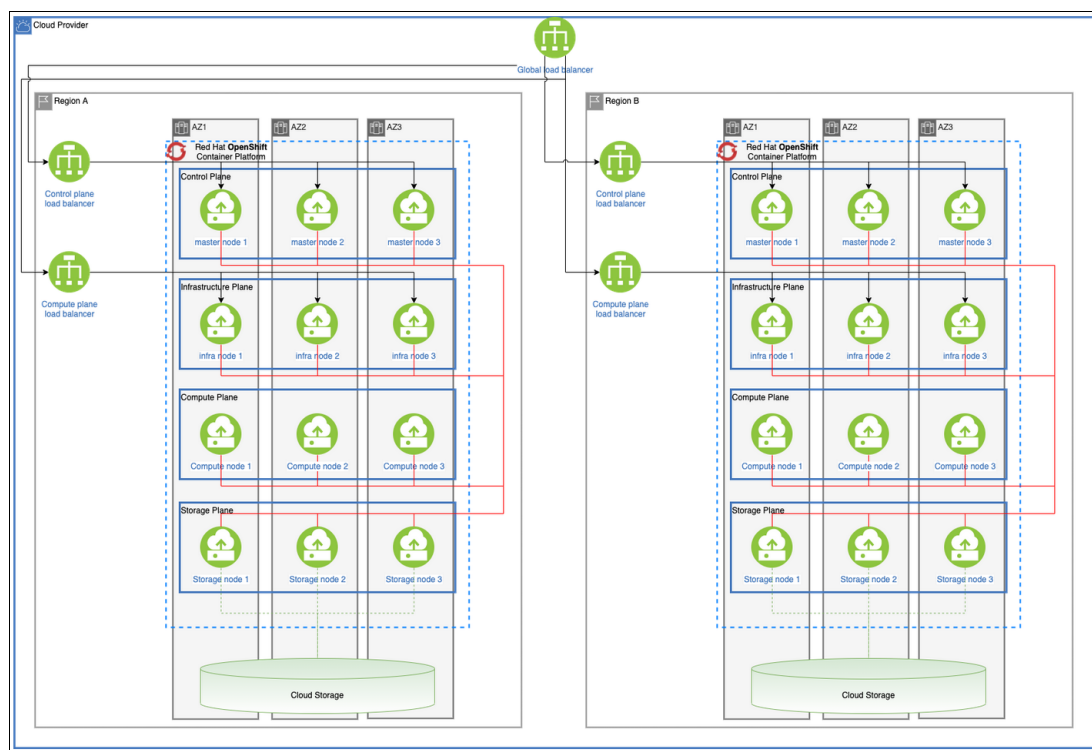


Figure 4-13 Two clusters cross-region

- ▶ Deploy clusters in different regions. Two regions are required. Minimum of two clusters, with one per region.
- ▶ Number of nodes per clusters:
 - Three master nodes
 - Three worker nodes
 - Three infrastructure nodes (Elasticsearch requires three instances, with one per node)
- ▶ Application workloads: Any.
- ▶ Use cases: Production: On-premises. Development and test can coexist.

Consideration: The replication methods, whether using middleware or product-native replication or storage-based replication, and the DR topology (Active/Active or Active/Passive) should be decided based on the capabilities of the deployed products, workload, middleware, and storage. Supported backup and restore can be done, but it reflects on your RTO.

Anti-patterns

Anti-patterns are architectures that generate more issues than they solve problems. Their implementation focuses on a small set of requirements and does not consider the full picture.

Anti-pattern 1: Two data center stretch cluster

This configuration, which is shown in Figure 4-14, is *not recommended* because of the following potential issues:

- ▶ Network latency
- ▶ Network issues
- ▶ Loss of quorum

These issues lead to inconsistent performance and an overall unstable environment due to difficulty diagnosing and solving any issues.

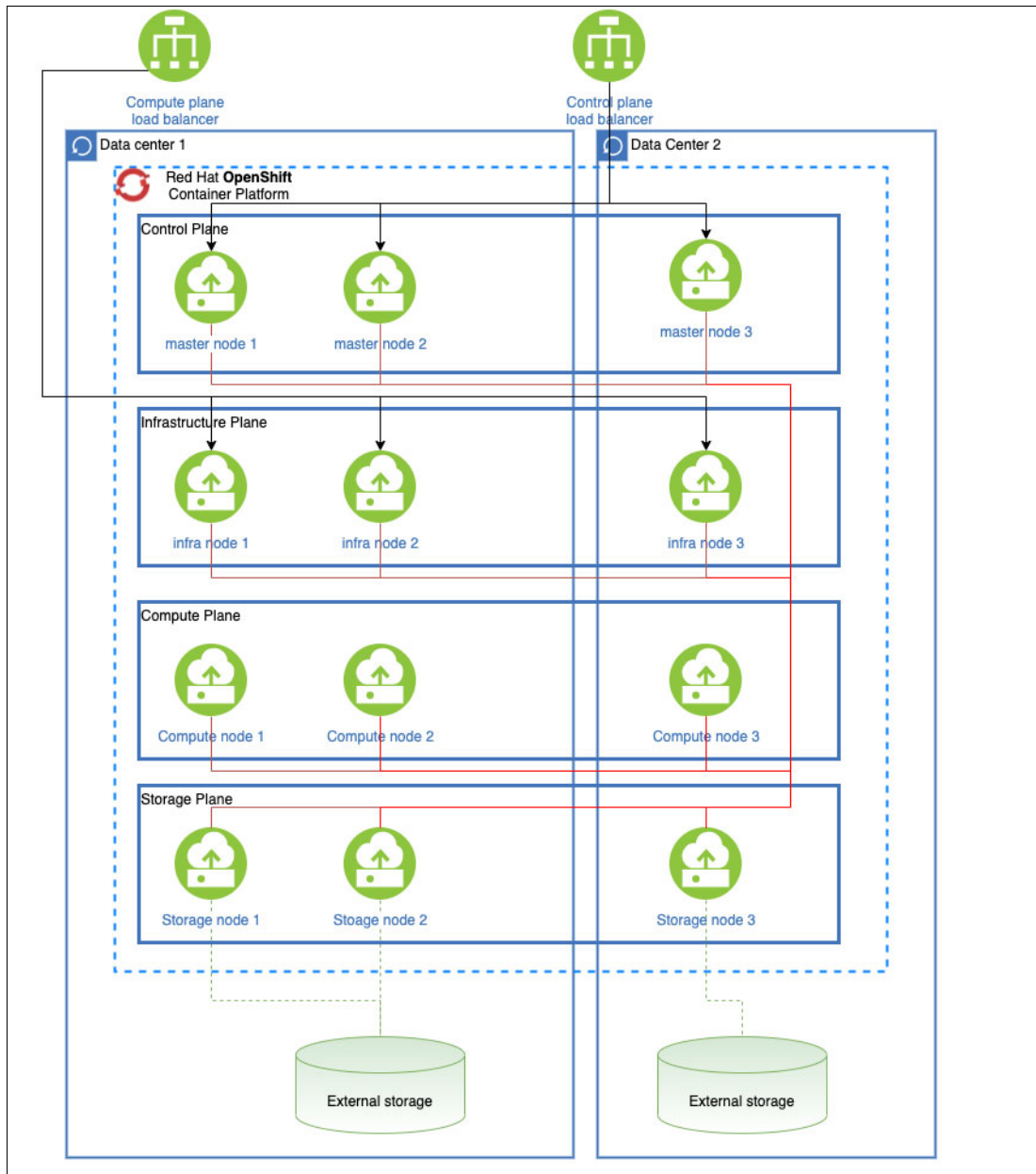


Figure 4-14 Stretch cluster

Anti pattern 2: Hybrid stretch cluster

Like “Anti-pattern 1: Two data center stretch cluster” on page 123, this configuration, shown in Figure 4-15, is *not recommended* because of similar problems with management and stability:

- ▶ Network latency
- ▶ Network issues
- ▶ Scheduling and workload placement

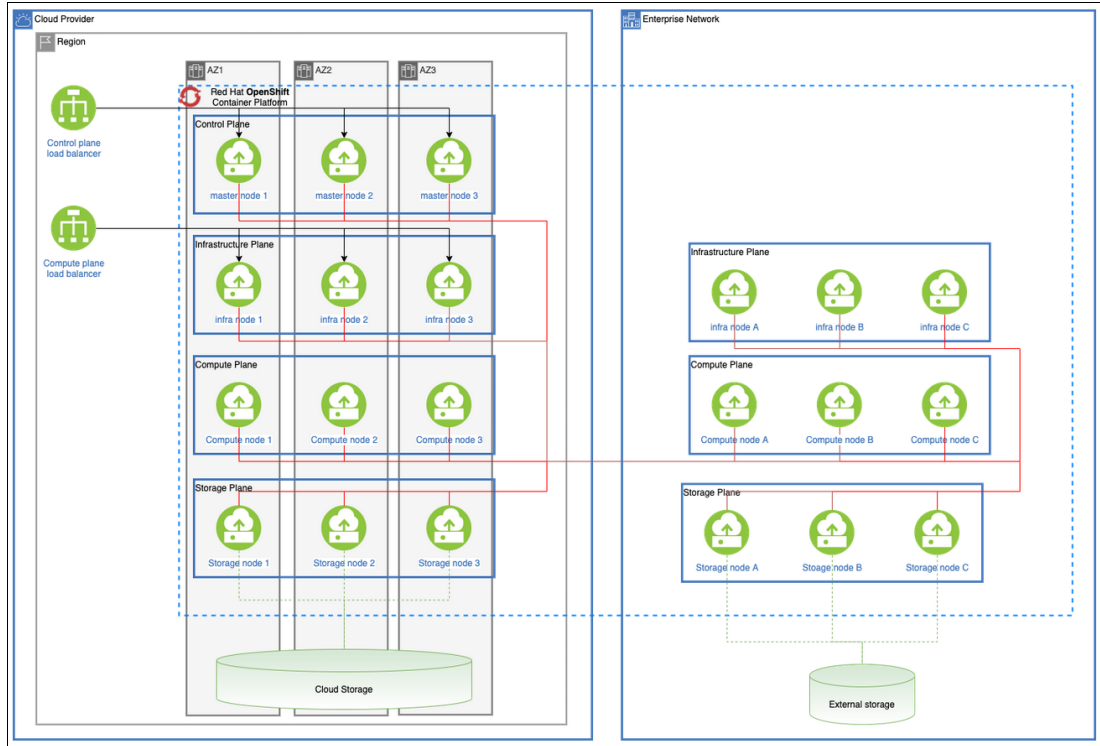


Figure 4-15 Hybrid stretch cluster

Conclusion

In summary, designing and planning your Red Hat OpenShift architecture and infrastructure is critical to the success you have in deploying and running your applications in that new environment. You must understand your business objectives and the application workload that you intend to run in the environment. Understanding them helps you choose the right platform to support the environment.

Table 4-5 summarizes the factors for consideration.

Table 4-5 Factors to consider in planning your Red Hat OpenShift infrastructure

Factor to consider	Some choices
Hardware resource and platform	IBM Power or IBM PowerVS IBM VPC IBM zSystems Generic x86 Infrastructure
Application workload	Stateful or stateless Microservices AI and machine learning

Factor to consider	Some choices
Environment	Development Production DR Site On-premises Public, private, or hybrid cloud
Business objectives	RPO RTO SLA HA requirements Compliance requirements

4.5 Red Hat OpenShift ecosystem

An *ecosystem*² is a complex of living organisms, their physical environment, and all their interrelationships in a particular unit of space. In this section, we describe the Red Hat OpenShift ecosystem. We show how the product helps developers and integrators deploy and maintain workloads, and how they can manage the infrastructure to meet your enterprise compute requirements.

Red Hat OpenShift is an enterprise container orchestration platform. It is a software product that includes components of the Kubernetes container management project but adds productivity and security features that are important for large-scale workloads. With Kubernetes orchestration, you can build application services that span multiple containers, schedule containers across a cluster, scale those containers, and manage their health over time. Kubernetes eliminates many of the manual processes that are involved in deploying and scaling containerized applications.

Red Hat OpenShift also incorporates various features that are required for an enterprise to manage and run applications better and faster. Red Hat OpenShift offers consistent security, built-in monitoring, centralized policy management, and compatibility with Kubernetes container workloads. It is fast, enables self-service provisioning, and integrates with various tools. There is *no* vendor lock-in, which is demonstrated by the various environments, both hardware-based and software-based (virtualized) environments, where Red Hat OpenShift can run on-premises or in a cloud solution. The flexibility also is shown by all the services and solutions that you can run on Red Hat OpenShift itself.

There are many solutions and tools that can help you operate the ecosystem and deploy new services on it. One of the new solutions that can provide the physical environment and help deploy and operate the Red Hat OpenShift cluster is IBM Storage Fusion (previously named IBM Spectrum Fusion™). IBM Storage Fusion is a container-native data platform for Red Hat OpenShift with enterprise-grade data storage and protection services. It offers an agile way to manage, recover, and access your mission-critical data. For more information about IBM Storage Fusion, see [IBM Storage Fusion](#).

Other elements of this ecosystem help create and deploy solutions like DevOps and CI/CD tools. Another component in the ecosystem is Service Mesh by MuleSoft, which can help you with governance, security and discoverability, and API management. Another major component is GitOps, which helps with security and operability, and to implement infrastructure as code (IaC) practices.

² <https://www.britannica.com/science/ecosystem>

Who uses Red Hat OpenShift

The Red Hat enterprise production ecosystem gives DevOps a consistent application platform to manage hybrid cloud, multi-cloud, and edge deployments so that your business can innovate quickly.

Why Red Hat OpenShift

Red Hat OpenShift is one of the fastest growing enterprise products. Red Hat OpenShift manages hybrid technologies by running enterprise applications, which helps your enterprise modernize existing or legacy applications, and accelerates new cloud-native application development and delivery at scale across any infrastructure.

The following sections describe that advantages that your enterprise can get from using Red Hat OpenShift.

Scalability

Any enterprise that adopts Red Hat OpenShift and deployed apps can scale to thousands of instances across hundreds of nodes in seconds. On cloud environments, scalability is a best in class option for enterprises.

Flexibility

Integration makes the product lifecycle flexible. Red Hat OpenShift simplifies the deployment and management of a hybrid infrastructure so that you have the flexibility of a self-managed or fully managed service running an on-premises, cloud, or hybrid environment.

Open-source standards

Red Hat OpenShift incorporates Open Container Initiative (OCI) containers and Cloud Native Computing Foundation (CNCF)-certified Kubernetes for container orchestration, in addition to other open source technologies.

Container portability

Container images that are built on the OCI industry standard ensure portability between developer workstations and Red Hat OpenShift production environments.

Enhanced developer experience

Red Hat OpenShift offers a comprehensive set of developer tools, multi-language support, and CLI and integrated development environment (IDE) integrations. Features include CI/CD pipelines based on Red Hat products and third-party CI/CD solutions, Service Mesh, serverless capabilities, and monitoring and logging capabilities.

Automated installation and upgrades

Automated installation and over-the-air platform upgrades are supported in cloud with Amazon Web Services, Google Cloud Platform, IBM Cloud, and Microsoft Azure. Also, for on-premises solutions that use vSphere, there are Red Hat OpenStack Platform, Red Hat Virtualization, or bare metal. Services that are used from the Operator-Hub can be deployed configured, and are they upgradeable with one click.

Automation

Streamlined and automated container and application builds, deployments, scaling, health management, and more are included in Red Hat OpenShift. Ansible or any other automation products provide integration capability, which helps automate activities easily.

Edge architecture support

Red Hat OpenShift enhances support of smaller-footprint topologies in edge scenarios that include 3-node clusters, single-node Red Hat OpenShift clusters, and remote worker nodes, which better map to varying physical sizes, connectivity, and availability requirements of different edge sites. The edge use cases are further enhanced with support for Red Hat OpenShift clusters on an ARM architecture, which is commonly used for low-power-consumption devices.

Multicluster management

Red Hat OpenShift with Red Hat Advanced Cluster Management for Kubernetes can easily deploy apps, manage multiple clusters, and enforce policies across clusters at scale.

Advanced security and compliance

Red Hat OpenShift offers core security capabilities like access controls, networking, and an enterprise registry with a built-in scanner. Red Hat Advanced Cluster Security for Kubernetes enhances security with capabilities like runtime threat detection, full lifecycle vulnerability management, and risk profiling.

Red Hat OpenShift also comes with a prebuilt operator for compliance (Compliance Operator), which helps close all vulnerable doors tightly.

Persistent storage

Red Hat OpenShift supports a broad spectrum of enterprise storage solutions, including Red Hat OpenShift Container Storage and Data Foundation, Elastic File storage, and other options for running both stateful and stateless apps. Existing block storage solutions can be integrated through the Container Storage Interface (CSI) driver.

Robust ecosystem

An expanding ecosystem of partners provides many integrations. Third parties deliver more storage and network providers, IDE, CI, integrations, independent software vendor (ISV) solutions, and more.

OperatorHub

OperatorHub is the web console interface in Red Hat OpenShift Container Platform that cluster administrators use to discover and install Operators.

4.5.1 Operator Lifecycle Manager

OLM is part of the open-source Operator Framework, which is designed to manage Operators in an effective, automated, and scalable way. OLM helps install, update, and manage the whole lifecycle of container-native applications and associated services in Red Hat OpenShift clusters.

Operators provide more functions than older tools like Helm and base Red Hat OpenShift resources, which managed applications and services. Basically, OLM can manage many Red Hat OpenShift resources and their whole lifecycle.

Table 4-6 compares OLM and Operators to Helm for installation automation.

Table 4-6 Compare Helm and Operators

Capability	Helm charts	Operators
Packaging	Standard	Standard
Installation	Standard	Standard
Updates by using Kubernetes manifests	Standard	Standard
Upgrades by using data migration and sequential tasks	N/A	Available
Backup and recovery	N/A	Available
Auto-tuning and self-healing with workload and log analysis	N/A	Available
Integration with external cloud services and APIs	N/A	Available
Event-based automation	N/A	Available
Stepwise automation	N/A	Available

OLM is built from two Operators:

- ▶ **OLM Operator:** Responsible for deploying applications that are defined by cluster service version (CSV) resources, by watching predefined requirements and running the installation strategy that is defined.
- ▶ **Catalog Operator:** Responsible for resolving and installing CSVs and the required resources that they specify. Also monitors the defined catalog sources and manages the updates of packages based on configuration.

Table 4-7 shows the Custom Resource Definitions (CRDs) that are managed by the two Operators.

Table 4-7 Custom Resource Definitions that are managed by OLM Operators

Resource	Short name	Owner	Definition
ClusterServiceVersion (CSV)	csv	OLM	Application metadata: Name, version, icon, required resources, installation, and so on.
InstallPlan	ip	Catalog	Calculated list of resources to create to automatically install or upgrade a CSV.
CatalogSource	catsrc	Catalog	A repository of CSVs, CRDs, and packages that define an application.

Resource	Short name	Owner	Definition
Subscription	sub	Catalog	Used to keep CSVs up to date by tracking a channel in a package.
OperatorGroup	og	OLM	Configures all Operators that are deployed in the same namespace as the OperatorGroup object to watch for their custom resource (CR) in a list of namespaces or cluster-wide.

There are default Catalog Sources when Red Hat OpenShift is installed, but developers and vendors can create sources that can be added to Red Hat OpenShift. The default Catalog Sources are the following ones:

- ▶ certified-operators: Products from leading ISVs. Red Hat partners with ISVs to package and ship them. Supported by the ISV.
- ▶ community-operators: Optionally, visible software that is maintained by relevant representatives in the operator-framework/community-operators GitHub repository. No official support.
- ▶ redhat-marketplace: Certified software that can be purchased from [Red Hat Marketplace](#).
- ▶ redhat-operators: Red Hat products that are packaged and shipped by Red Hat. Supported by Red Hat.

Red Hat OpenShift provides the OperatorHub dashboard to search for and install Operators from the defined catalog sources. It is possible to disable certain catalog sources, which can be useful to prevent users from installing Operators from community sources in a production environment.

Because Operators manage the standard and custom resources (CRs) of a package, installing an Operator creates only CRDs (if necessary) and sets the update channels. After installation, you may chose which update channel you are following, based on which OLM can do automatic updates, as shown in Figure 4-16.

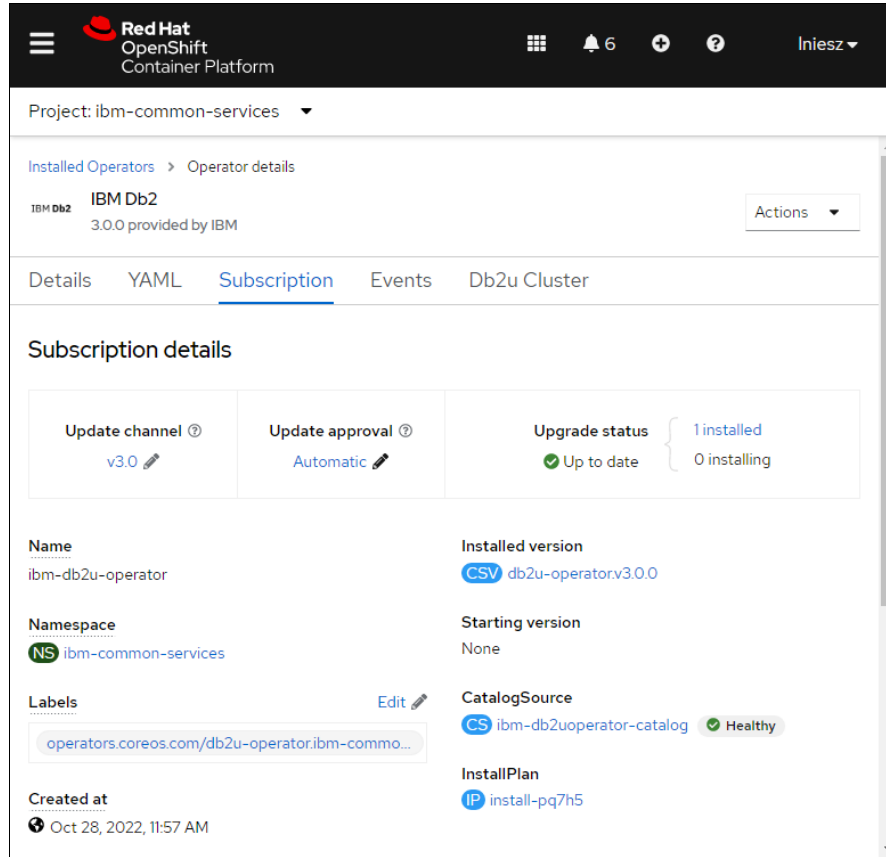


Figure 4-16 IBM Db2 Operator Subscription details

Operator SDK

Operator SDK is a component of the Operator Framework. It provides a CLI-based tool to build, test, and deploy an Operator.

Operators that are built by the Operator SDK watch the resources and process events based on the changes of resources in a handler. The handler acts to reconcile the state of the application package.

Operators can be developed that are based on Go, Ansible, and Helm, but with different capabilities, as shown in Figure 4-17.³

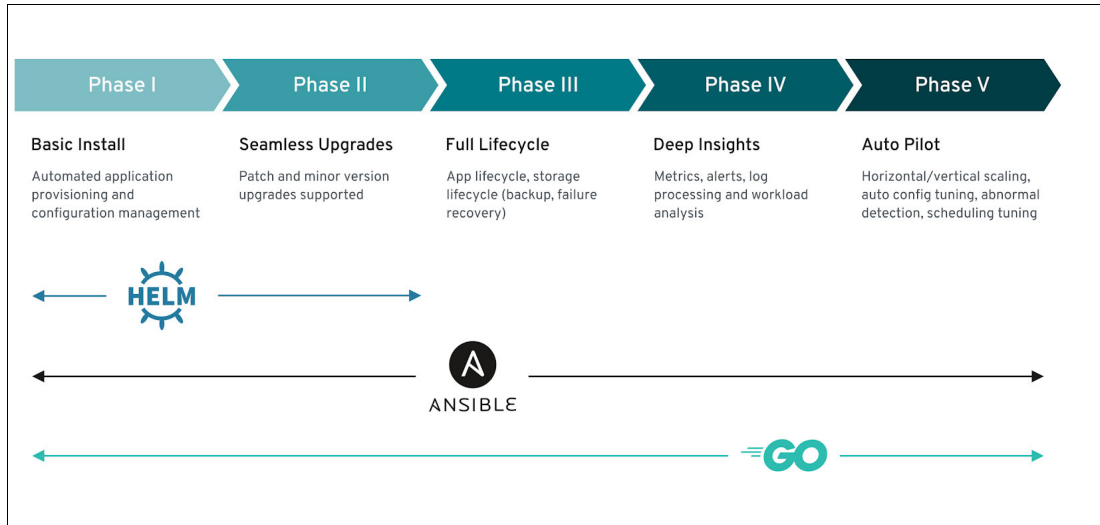


Figure 4-17 Operator capability level

Cluster operators

Red Hat OpenShift itself uses the Operator managing key platform elements, which are called cluster operators. Example 4-11 shows the installed operators and their states in our test cluster.

Example 4-11 Red Hat OpenShift cluster operators

```
(py39) [root@build-cp4d-1 ~]# oc get co
```

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
authentication	4.10.34	True	False	False	171m
baremetal	4.10.34	True	False	False	39d
cloud-controller-manager	4.10.34	True	False	False	39d
cloud-credential	4.10.34	True	False	False	39d
cluster-autoscaler	4.10.34	True	False	False	39d
config-operator	4.10.34	True	False	False	39d
console	4.10.34	True	False	False	21d
csi-snapshot-controller	4.10.34	True	False	False	39d
dns	4.10.34	True	False	False	39d
etcd	4.10.34	True	False	False	39d
image-registry	4.10.34	True	False	False	24h
ingress	4.10.34	True	False	False	39d
insights	4.10.34	True	False	False	39d
kube-apiserver	4.10.34	True	False	False	39d
kube-controller-manager	4.10.34	True	False	False	39d
kube-scheduler	4.10.34	True	False	False	39d
kube-storage-version-migrator	4.10.34	True	False	False	24h
machine-api	4.10.34	True	False	False	39d
machine-approver	4.10.34	True	False	False	39d
machine-config	4.10.34	True	False	False	16h
marketplace	4.10.34	True	False	False	39d
monitoring	4.10.34	True	False	False	39d
network	4.10.34	True	False	False	39d
node-tuning	4.10.34	True	False	False	29d
openshift-apiserver	4.10.34	True	False	False	30d
openshift-controller-manager	4.10.34	True	False	False	6d21h
openshift-samples	4.10.34	True	False	False	30d

³ <https://redhat-connect.gitbook.io/certified-operator-guide/>

operator-lifecycle-manager	4.10.34	True	False	False	39d
operator-lifecycle-manager-catalog	4.10.34	True	False	False	39d
operator-lifecycle-manager-packageserver	4.10.34	True	False	False	21d
service-ca	4.10.34	True	False	False	39d
storage	4.10.34	True	False	False	39d

The version of the Operator and the whole Red Hat OpenShift cluster is controlled by the ClusterVersion configuration, which is watched by the Cluster Version Operator. This Operator is where parameters that are related to automatic updates can be set.

Figure 4-18 shows the version and the possible update channels of our test cluster.

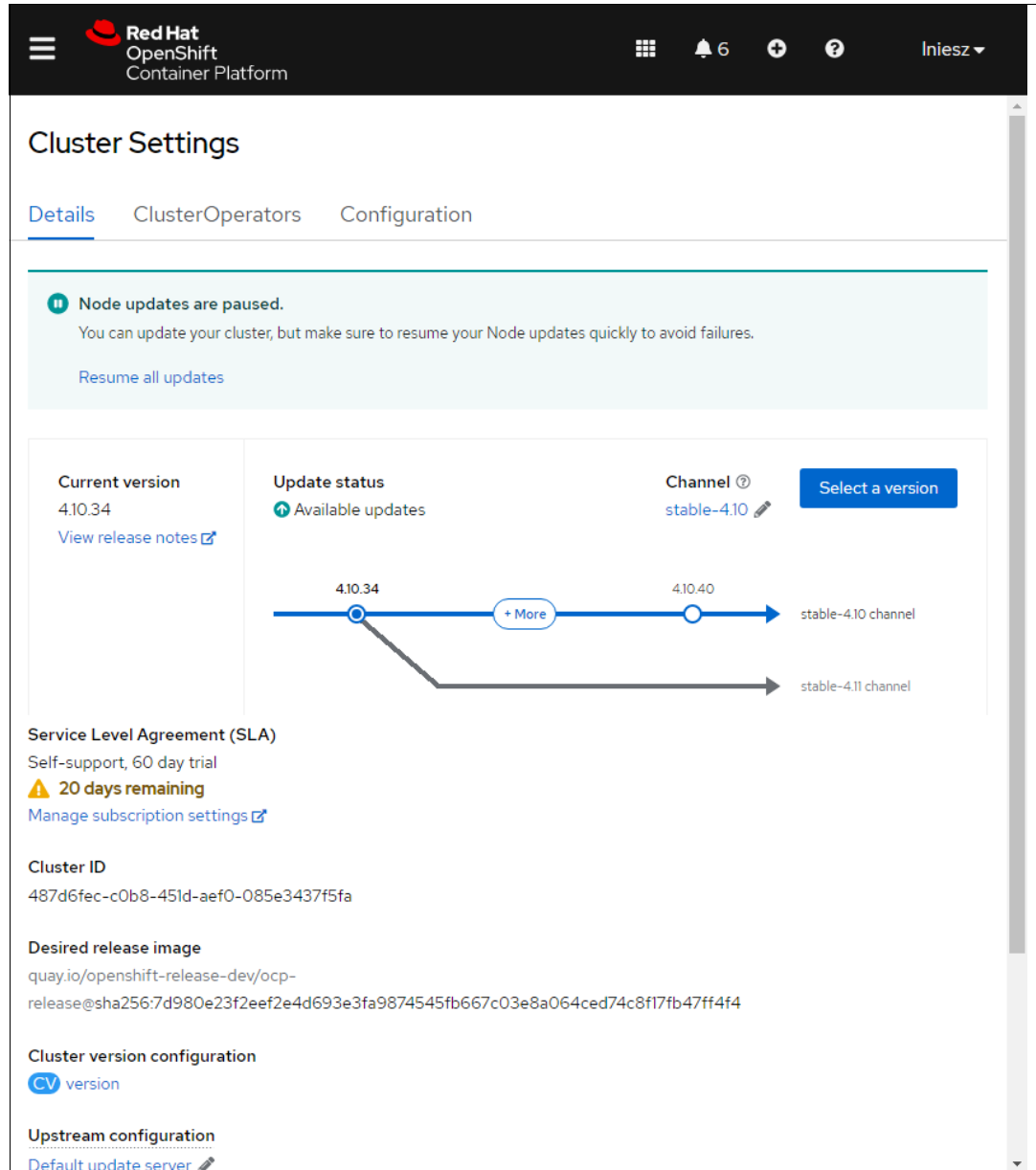


Figure 4-18 Cluster version and update channels in the GUI

The Operators are implemented as Red Hat OpenShift pods, which use other resources to manage the cluster. The related resources can be viewed in Red Hat OpenShift GUI or the CLI, as shown for the authentication cluster operator in Example 4-12.

Example 4-12 Listing the authentication Operator-related objects

```
(py39) [root@build-cp4d-1 ~]# oc get co authentication -o jsonpath="{.status.relatedObjects}" | jq
[
  {
    "group": "operator.openshift.io",
    "name": "cluster",
    "resource": "authentications"
  },
  {
    "group": "config.openshift.io",
    "name": "cluster",
    "resource": "authentications"
  },
  {
    "group": "config.openshift.io",
    "name": "cluster",
    "resource": "infrastructures"
  },
  {
    "group": "config.openshift.io",
    "name": "cluster",
    "resource": "oauths"
  },
  {
    "group": "route.openshift.io",
    "name": "oauth-openshift",
    "namespace": "openshift-authentication",
    "resource": "routes"
  },
  {
    "group": "",
    "name": "oauth-openshift",
    "namespace": "openshift-authentication",
    "resource": "services"
  },
  {
    "group": "",
    "name": "openshift-config",
    "resource": "namespaces"
  },
  {
    "group": "",
    "name": "openshift-config-managed",
    "resource": "namespaces"
  },
  {
    "group": "",
    "name": "openshift-authentication",
    "resource": "namespaces"
  },
  {
    "group": "",
    "name": "openshift-authentication-operator",
    "resource": "namespaces"
  },
  {
    "group": "",
    "name": "openshift-ingress",
    "resource": "namespaces"
  },
  {
    "group": "",
    "name": "openshift-oauth-apiserver",
    "resource": "namespaces"
  }
]
```

Here you can see the configuration resource `cluster.oauth.config.openshift.io`, which you can use to add authentication methods into the cluster.

Example 4-13 shows that `htpasswd` authentication is configured in our test cluster.

Example 4-13 OAuth configuration

```
apiVersion: config.openshift.io/v1
kind: OAuth
metadata:
  annotations:
    include.release.openshift.io/ibm-cloud-managed: 'true'
    include.release.openshift.io/self-managed-high-availability: 'true'
    include.release.openshift.io/single-node-developer: 'true'
    release.openshift.io/create-only: 'true'
  creationTimestamp: '2022-10-09T08:41:13Z'
  generation: 2
  managedFields: ...
name: cluster
ownerReferences:
- apiVersion: config.openshift.io/v1
  kind: ClusterVersion
  name: version
  uid: 8fd427af-8579-44d6-8e6b-fdfd65948698
resourceVersion: '3466695'
uid: e9a2ff78-90f4-44e8-a414-1f02b70388f0
spec:
  identityProviders:
  - htpasswd:
      fileData:
        name: htpasswd-bd4bw
      mappingMethod: claim
      name: htpasswd
      type: HTPasswd
```

4.5.2 Service Mesh

Red Hat OpenShift Service Mesh⁴ addresses various problems in a microservices architecture by creating a centralized point of control in an application. It adds a transparent layer on existing distributed applications without requiring any changes to the application code.

Service Mesh, which is based on the open-source Istio project, provides a way to create a network of deployed services that provides discovery, load-balancing, service-to-service authentication, failure recovery, metrics, and monitoring. A service mesh also provides more complex operational functions, including A/B testing, canary releases, access control, and end-to-end authentication.

Red Hat OpenShift Service Mesh provides several key capabilities uniformly across a network of services:

- ▶ **Traffic Management:** Controls the flow of traffic and API calls between services, makes calls more reliable, and makes the network more robust in the face of adverse conditions.
- ▶ **Service Identity and Security:** Provides services in the mesh with a verifiable identity, and protects service traffic as it flows over networks of varying degrees of trustworthiness.
- ▶ **Policy Enforcement:** Applies an organizational policy to the interaction between services, ensures that access policies are enforced and resources are fairly distributed among consumers. Policy changes are made by configuring the mesh, not by changing the application code.
- ▶ **Telemetry:** You gain an understanding of the dependencies between services and the nature and flow of traffic between them to quickly identify issues.

For more information about the preparation and installation of Red Hat OpenShift Service Mesh on your Red Hat OpenShift Container Platform, see the [Service Mesh Documents](#).

⁴ https://docs.openshift.com/container-platform/4.11/service_mesh/v2x/ossm-about.html

4.5.3 DevOps and CI/CD pipelines

Development and operations (DevOps)⁵ represent a set of ideas and practices larger than those two terms alone or together. DevOps speeds up how an idea goes from development to deployment. At its core, DevOps relies on automating routine operational tasks and standardizing environments across an app's lifecycle.

- ▶ **DevOps culture:** DevOps relies on a culture of collaboration that aligns with open-source principles and transparent, agile approaches to work. The culture of open-source software projects can be a blueprint for building a DevOps culture. Sharing information is the default approach to collaboration in open-source communities.
- ▶ **DevOps process:** Developing modern applications requires different processes than the approaches of the past. Many teams use agile approaches to software development by using different software architectures, for example, a microservices architecture. For these teams, DevOps is not an afterthought. In fact, “Customer satisfaction through early and continuous software delivery” is the first of 12 principles in the Agile Manifesto,⁶ which is why continuous integration and continuous deployment (CI/CD) is so important to DevOps teams.
- ▶ **DevOps platform and tools:** Selecting tools that support your processes is critical for DevOps to be successful. If your operations are going to keep pace with rapid development cycles, they must use highly flexible platforms and treat their infrastructure like dev teams treat code. Manual deployments are slow and leave room for error.

Platform provisioning and deployment can be simplified through automation. Site Reliability Engineering (SRE) takes these manual operations tasks and manages them by using software and automation. An SRE approach can further support the goals of a DevOps team.

DevOps versus SRE

DevOps is an approach to culture, automation, and platform design to deliver increased business value and responsiveness through rapid, high-quality service delivery. SRE can be considered an implementation of DevOps.⁷

Like DevOps, SRE is about team culture and relationships. Both SRE and DevOps work to bridge the gap between development and operations teams to deliver services faster.

Faster application development lifecycles, improved service quality and reliability, and reduced IT time per application that is developed are benefits that can be achieved by both DevOps and SRE practices.

However, SRE differs from DevOps because it relies on Site Reliability Engineers within the development team (who also have an operations background) to remove communication and workflow problems.

Continuous Integration, Continuous Delivery, and Continuous Deployment

CI/CD⁸ is a method to frequently deliver apps to customers by introducing automation into the stages of app development. The main concepts attributed to CI/CD are continuous integration, continuous delivery, and continuous deployment. CI/CD is a solution to the problems that integrating new code can cause for development and operations teams.

⁵ <https://www.redhat.com/en/topics/devops>

⁶ <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>

⁷ <https://www.redhat.com/en/topics/devops/what-is-sre>

⁸ <https://www.redhat.com/en/topics/devops/what-is-ci-cd>

Specifically, CI/CD introduces ongoing automation and continuous monitoring throughout the lifecycle of apps, from integration and testing phases to delivery and deployment. Taken together, these connected practices are often referred to as a “CI/CD pipeline” and are supported by development and operations teams working together in an agile way with either a DevOps or SRE approach.

CI/CD tools

CI/CD tools can help a team automate their development, deployment, and testing:

- ▶ One of the best-known open-source tools for CI/CD is the automation server Jenkins. Jenkins can handle anything from a simple CI server to a complete CD hub.
- ▶ Tekton Pipelines is a CI/CD framework for Kubernetes platforms that provides a standard cloud-native CI/CD experience with containers.
- ▶ Beyond Jenkins and Tekton Pipelines, here are other open-source CI/CD tools that you might want to investigate:
 - Spinnaker, a CD platform that is built for multicloud environments.
 - GoCD, a CI/CD server with an emphasis on modeling and visualization.
 - Concourse, “an open-source continuous thing-doer.”⁹
 - Screwdriver, a build platform that is designed for CD.

Also, any tool that is foundational to DevOps is likely to be part of a CI/CD process. Tools for configuration automation (such as Ansible, Chef, and Puppet), container run times (such as Docker, rkt, and cri-o), and container orchestration (Kubernetes) are not strictly CI/CD tools, but they show up in many CI/CD workflows.

Tekton Pipelines is available in Red Hat OpenShift through the Red Hat OpenShift Pipelines Operator.

4.5.4 GitOps for Red Hat OpenShift node tuning and configuration

One option that can be used for tuning your Red Hat OpenShift cluster node is GitOps. This section describes that process. An example is shown in 6.3, “GitOps for system configuration” on page 190.

What is GitOps

GitOps is a set of principles that originally was defined for operating and managing software systems, but these principles can be useful in managing software-defined infrastructures and running those software systems. The principles are derived from existing best practices from other IT fields, like software development and delivery.

Open GitOps Working Group for standardization

Open GitOps is a collection of open-source standards that is related to standardizing GitOps definitions and implementation. It is managed by the GitOps Working Group under CNCF. For more information, see [GitOps](#).

GitOps principles

This section shows how you can use these principles in practice to help the operation and especially the tuning and performance configuration of Red Hat OpenShift clusters on IBM Power servers. It also shows the available tools that can be used to put the principles in practice.

⁹ <https://concourse-ci.org/>

The principles are the following ones:

- ▶ **Declarative:** The wanted state of a GitOps managed system must be expressed in declarative forms.
- ▶ **Versioned and Immutable:** The storage of the wanted state must provide immutability and provide versioning, so the whole version history must be stored.
- ▶ **Pulled Automatically:** The wanted state of the elements of the system is pulled by agents automatically.
- ▶ **Continuously Reconciled:** The software agents are continuously monitoring the changes of the state, and they apply the wanted state on the system.

These principles have common ground with other modern software development and delivery methods and practices, like DevSecOps and CI/CD. GitOps principles and the tools to implement them especially are used in the deployment phase of CI/CD. The principles can be applied at the application lifecycle management and the underlying software-defined infrastructure itself. Thinking about the subject of our book, which is the tuning and management of workloads on Red Hat OpenShift on IBM Power servers, this approach is possible because of the way Red Hat OpenShift works.

Red Hat OpenShift: Tuning related resources to manage with GitOps

In Red Hat OpenShift, the application and related resources (secrets, configurations, and ingress routes), and the underlying infrastructure elements, are defined by YAML files. This declarative way of defining all the cluster elements makes it possible to store and handle node configuration and tuning parameters in YAML files, which can be applied on the cluster to change these otherwise immutable parameters of the nodes and the running operating systems on cluster. The collection of these YAML files can be handled as a configuration database, but when moving toward the GitOps way of working, they can be stored in a Git repository.

Using Red Hat CoreOS as the operating system of the nodes, and the operating system updates that are implemented through Red Hat OpenShift, provides an immutable and secure way to handle the node configurations. Direct login to the nodes can be disabled, but if it is not, then direct modification of operating system changes are reverted by Red Hat OpenShift at the next update, which might cause problems for the operation.

A node, which can be either a master, infrastructure, or compute node in an Red Hat OpenShift cluster, is basically a running operating system on a virtual server or in the case of bare metal, a full physical server.

In the following list, we show the resource types in Red Hat OpenShift that can be used for a tuning-related configuration:

- ▶ **Node:** Manages node grouping, and labels that are used to assign other configuration resources.
- ▶ **MachineConfig:** Can be used to configure the following node and operating system settings:
 - **config:**
 - **storage-files:** Can be used to push configuration files to nodes, which are used by systemd daemons, for example.
 - **systemd:** Can be used to define users and to send SSH public keys for remote access of nodes.
 - **passwd:** Can be used to distribute SSH keys.

- extensions: Configures host OS extensions.
- FIPS: Enables running the node in FIPS mode.
- kernelArguments: Configures kernel arguments.
- kernelType: Can be used to run the host OS in real-time kernel mode.
- osImageURL: Defines the source of the operating system image.
- ▶ MachineConfigPool: Can be used to group settings in MachineConfig resources and assign them to nodes.
- ▶ Tuned: Can be used to create profiles of node- and operating system-related tuning and assign the configurations to nodes.
- ▶ KubeletConfig: Can be used to configure one of the main Red Hat OpenShift daemons.

Red Hat OpenShift GitOps

Red Hat provides Red Hat OpenShift GitOps for the automatic pulling and reconciliation of configurations, which can be stored in a Git-based external environment to provide storage and version control of the YAML files. Red Hat OpenShift GitOps is based on ArgoCD, which is a CNCF Incubating project. For more information, see [Red Hat OpenShift GitOps](#).

In Red Hat OpenShift GitOps, a dex server provides the authentication and authorization configuration, which enables it to use a role-based access control (RBAC) configuration that is set up in Red Hat OpenShift and also used in GitOps. To enable GitOps to configure resources in an Red Hat OpenShift namespace, the namespace should be labeled by using the following key and value:

```
argocd.argoproj.io/managed-by=<ArgoCD instance name>
```

Some of the tuning-related Red Hat OpenShift resource types are not namespace-scoped resources, so the normal Red Hat OpenShift GitOps authorization method is not working. The Red Hat OpenShift GitOps Service Accounts must be assigned with elevated roles because we show them in our GitOps use case, as shown in 6.3, “GitOps for system configuration” on page 190.

The possible configurations are changing and evolving with new Red Hat OpenShift and supported hardware versions and types.

Note: Applying Red Hat OpenShift configuration changes to a live cluster might restart cluster nodes, which can cause the pod to stop and restart on other nodes. This event might result in application outages.

The following list shows the main configuration elements in an Red Hat OpenShift GitOps configuration:

- ▶ Red Hat OpenShift GitOps operator
- ▶ ArgoCD instance
- ▶ Project:
 - Cluster
 - Source repository

► Application

When you use GitOps for infrastructure management, you do not define real software applications, but instead define a collection of Red Hat OpenShift resources, which can be defined by YAML files that are stored on a Git repository and applied to the live Red Hat OpenShift clusters.

► ApplicationSet

The storage and secure management of the Red Hat OpenShift resource definitions can be done by using GitHub or a compatible repository. This approach provides versioning and approval and review processes for changes before applying them to the live cluster.

The definitions can be either stand-alone YAML files, Helm charts, or Kustomize-based configurations.

GitOps work flow

The GitOps work flow consists of the following steps:

1. Setup:

- a. Create a Git repository for the configuration YAML files. Set up the approval process and the necessary security configuration.
- b. Create subdirectories to group the Red Hat OpenShift resource definition YAML files. A subdirectory is assigned to a GitOps application, and all YAML files, Helm charts, or Kustomize-based resources are applied to the defined GitOps “Application”.
- c. Collect the initial YAML files from the working cluster under the subdirectories. Push the individual YAML files, Helm charts, or Kustomize configuration in the appropriate subdirectory.
- d. Set up Red Hat OpenShift GitOps in the target clusters:
 - i. Install the operator.
 - ii. Create an ArgoCD instance.
 - iii. Configure secure authentication and authorization for GitOps procedures.
 - iv. Create Red Hat OpenShift Roles and RoleBindings to GitOps ServiceAccounts to enable the management of namespace-scoped resources.
- e. In the ArgoCD GUI or CLI, create a project for handling configuration changes. Specify a destination cluster, lists of cluster resources that may be modified, and source repositories that can be used for configuration YAML files.
- f. Create an application that specifies the project. Provide the necessary setup by specifying the project, source repository, and target. Set up automatic synchronization if necessary, but be careful because some node reconfigurations might restart the pods running on the node because the node’s operating system might restart to get the new settings.

2. Operation:

- a. Change the configuration YAML files in the Git repository as necessary by using Git pull requests and going through necessary approvals.
- b. Check the changes in the ArgoCD application, and synchronize the configuration if it is not set to synchronize automatically.
- c. Check the applied configuration in the Red Hat OpenShift cluster.

We show a sample setup about how to use GitOps with Red Hat OpenShift to configure and tune the cluster nodes, and to store the configurations in GitHub, in 6.3, “GitOps for system configuration” on page 190.

4.6 Running Red Hat OpenShift on IBM Power

This section focuses on the benefits of running Red Hat OpenShift on IBM Power. It also describes how you can start your deployment of Red Hat OpenShift Container Platform on IBM Power, either in an on-premises bare metal system or in IBM PowerVS on IBM Cloud.

4.6.1 Red Hat OpenShift on IBM Power

Modern application development and modernization of existing applications require a robust platform that ensures scalability, agility, portability, security, and resiliency. It also must incorporate new ideas, features, and benefits into applications at a faster pace than traditional applications.

For many application workloads, Red Hat OpenShift on IBM Power is an excellent platform for your containerized applications, providing a superior user experience in a highly secure, high-performance environment that can be sized to meet your budgetary requirements. Red Hat OpenShift running on IBM Power can support more users per server than competing technologies, and it can be dynamically scaled up or down to meet your workload requirements.

In this section, we provide a deeper view into why you should choose Red Hat OpenShift for your cloud platform, and show you how running Red Hat OpenShift on IBM Power provides more benefits for your containerized applications.

Benefits of Red Hat OpenShift

Red Hat OpenShift full-stack automated operations and self-service provisioning for DevOps work together to more efficiently move ideas from development to production. Red Hat OpenShift is an enterprise-grade product full of useful features and capabilities that combine software development and IT operations.

Red Hat OpenShift manages hybrid technologies and applications to help you modernize existing applications and accelerate new, cloud-native application development and delivery at scale across any infrastructure. Red Hat OpenShift gives DevOps a consistent app platform to manage hybrid cloud, multi-cloud, and edge deployments so that your business can innovate quickly. Red Hat OpenShift features and benefits include the following items:

- ▶ Scalability
- ▶ Flexibility
- ▶ Open-source standards
- ▶ Container portability
- ▶ Enhanced developer experience
- ▶ Automated installation and upgrades
- ▶ Automation
- ▶ Edge architecture support
- ▶ Multi-cluster management
- ▶ Advanced security and compliance
- ▶ Persistent storage
- ▶ Robust ecosystem

Red Hat OpenShift includes the following capabilities:

- ▶ Backup and recovery
- ▶ CI/CD
- ▶ GitOps
- ▶ Helm
- ▶ HA
- ▶ Managing security
- ▶ Operators
- ▶ Sandboxed containers
- ▶ Serverless
- ▶ Service mesh
- ▶ Virtualization
- ▶ Windows containers

For more information about the Red Hat OpenShift capabilities, features, and benefits, see [Red Hat OpenShift features and benefits](#).

Benefits of Red Hat OpenShift on IBM Power

Capacity planning for Red Hat OpenShift is important because it determines the number of worker nodes in the cluster and how many pods are expected to fit per node. This number depends on the application because the application's memory, CPU, and storage requirements must be considered.

Red Hat OpenShift license pricing is based on CPU, which is reflected in the total number of CPUs of all the worker nodes or compute nodes in the Red Hat OpenShift cluster. Red Hat guidelines for the maximum number of pods per node is 250. Do not exceed this number because it results in lower overall performance.

x86 system versus IBM Power

An x86 core running with hyperthreading is equivalent to two Kubernetes CPUs. Therefore, when running with x86 hyperthreading, a Kubernetes CPU is equivalent to half of an x86 core. This conversion factor is shown in Table 4-8.

Table 4-8 CPU conversion

	vCPU	x86 cores (SMT2)	Physical SMT2 IBM Power cores	Physical SMT4 IBM Power cores
vCPUs and x86 or IBM Power cores	2	1	1	0.5

A PowerVM core can be defined to be 1, 2, 4, or 8 threads with the simultaneous multi-threading (SMT) setting. Therefore, when running on PowerVM with SMT4, a PowerVM core is equivalent to four Kubernetes CPUs, and when running with SMT8, the same PowerVM core is equivalent to eight Kubernetes CPUs. Therefore, when running with SMT4, a Kubernetes CPU is equivalent to a quarter of a PowerVM core, and when running with SMT8, a Kubernetes CPU is equivalent to one-eighth of a PowerVM core.

If your pod CPU resource was defined to run on x86, you must consider the effects of the IBM Power performance advantages and the effects of Kubernetes resources that are assigned on a thread basis. For example, for a workload where IBM Power has a 2X advantage over x86 when running with PowerVM SMT4, you can assign the same number of Kubernetes CPUs to IBM Power that you do to x86 to get equivalent performance. This conversion factor is shown in Table 4-9.

Table 4-9 vCPUs to physical cores conversion

vCPU	Physical x86 cores	Physical SMT2 IBM Power cores	Physical SMT4 IBM Power cores	Physical SMT8 IBM Power cores
56	28	28	14	7

For more information and some lessons learned along with some tips and tricks, see 2.4 “Red Hat OpenShift V4.3 sizing guidelines” in *Red Hat OpenShift V4.3 on IBM Power Systems Reference Guide*, REDP-5599.

On-premises IBM Power servers or in IBM Cloud

Cloud computing has grown in popularity, and adoption of cloud is also widespread depending on business needs, application workload, and other factors. But, organizations still need an on-premises infrastructure to run their core systems and applications. The hybrid cloud connects an organization’s on-premises private cloud services and third-party public cloud services into a single, flexible infrastructure for running critical applications and workloads.

Some reasons that on-premises computing continues to thrive include the following ones:

- ▶ Data residency
- ▶ Data gravity
- ▶ Existing on-premises capacity
- ▶ Complete control
- ▶ Security and compliance requirements

But with on-premises computing, you are responsible for maintaining server hardware and software, data backups, storage, and DR regarding SLA. This situation can be an issue for smaller companies who have limited budgets and technical resources. Cloud computing for smaller companies can help them to reduce operational costs and gain other benefits:

- ▶ Increased ability to optimize DR and business continuity
- ▶ Increased ability to scale capacity up and down based on demand
- ▶ Ability to maintain better levels of control of critical workloads
- ▶ Improved speed and decreased effort that are associated with updates
- ▶ Better IT infrastructure management and flexibility

For example, Figure 4-19 shows that Red Hat OpenShift on IBM Cloud has more uptime (99.99%) compared to other cloud providers.





	 Red Hat AWS	 Red Hat Microsoft Azure	 Red Hat OpenShift Dedicated	 Red Hat IBM
DETAILS	Red Hat OpenShift Service on AWS	Microsoft Azure Red Hat OpenShift	Red Hat OpenShift Dedicated	Red Hat OpenShift on IBM Cloud
Service level agreement (SLA)	99.95% uptime	99.95% uptime	99.95% uptime	99.99% uptime

Figure 4-19 SLA uptime (99.99%) comparison

4.6.2 How to install Red Hat OpenShift Container Platform in IBM Cloud

This section focuses on how to deploy and start Red Hat OpenShift on IBM Power either in an on-premises, bare-metal system or on IBM PowerVS on IBM Cloud. For step-by-step installation instructions, see Chapter 6, “Deployment scenarios”, in *Red Hat OpenShift V4.X and IBM Cloud Pak on IBM Power Systems Volume 2*, SG24-8486.

Deploying Red Hat OpenShift Container Platform 4.x in IBM Cloud

The Red Hat OpenShift Container Platform 4.x deployment process in IBM PowerVS on IBM Cloud uses Red Hat Ansible and Terraform.

The installation process uses the code that is found at this [GitHub repository](#).

At a high level, the Red Hat OpenShift Container Platform 4.x deployment process in IBM PowerVS on IBM Cloud includes the following steps:

1. Setting up the IBM Cloud environment:
 - a. Set up the IBM PowerVS service.
 - b. Set up the private network.
 - c. Import Red Hat images.
 - d. Set up the user API key.
2. Setting up the deployment host:
 - a. Install Terraform.
 - b. Install IBM Cloud Terraform Provider.
 - c. Install the IBM PowerVS CLI.
3. Deploy the Red Hat OpenShift Container Platform:
 - a. Clone the `ocp4-upi-powervs` Git repository.
 - b. Set up the Terraform variables.
 - c. Install Red Hat OpenShift Container Platform.

Note: Terraform is not required to install Red Hat OpenShift Container Platform. Multiple approaches are available for deploying the infrastructure. This section demonstrates how to use IaC with Terraform to simplify the deployment.

Deploying Red Hat OpenShift Container Platform 4.x on IBM Power

The high-level Red Hat OpenShift Container Platform 4.x deployment process in IBM PowerVS includes the following steps:

1. Configure PowerVM for network installation.
2. Prepare your environment:
 - a. Configure the DHCP server.
 - b. Configure the TFTP server.
 - c. Configure the DNS server.
 - d. Configure the web server.
 - e. Configure the load balancer.
3. Start the servers to install RHCOS by using one of the following methods:
 - a. Install from ISO instead of the TFTP server.
 - b. Install from DHCP and th TFTP server.
4. Create the SSH key.
5. Perform the installation.
6. Check the installation.
7. Back up your cluster.

For more information, see Chapter 6, “Installing Red Hat OpenShift V4.3 and V4.4: Tips and tricks” in *Red Hat OpenShift V4.X and IBM Cloud Pak on IBM Power Systems Volume 2*, SG24-8486, and Chapter 3, “Reference installation guide for Red Hat OpenShift V4.3 on Power Systems servers” in *Red Hat OpenShift V4.3 on IBM Power Systems Reference Guide*, REDP-5599.

Note: You can use Ansible and prebuilt playbooks that were developed by the Red Hat Conference of the Parties (COP) to prepare your environment. This playbook assumes the following points:

- ▶ You can use Red Hat Enterprise Linux 7/8 System for the Ansible playbooks.
- ▶ You are on a network that has access to the internet.
- ▶ The ocp4-helper node is your LB, DHCP, PXE, DNS, and HTTP Server.
- ▶ You can disable installing DHCP on the helper, if required.
- ▶ You still must perform the Red Hat OpenShift installation steps manually.

You run the `openshift-install` command from the ocp4-helper node.

The prebuilt playbook is found at this [GitHub repository](#).

Advanced deployment of Red Hat OpenShift Container Platform

The process for installing Red Hat OpenShift 4.x on IBM Power servers might be different based on your architecture and the purpose of your cluster, such as development and testing, production, DR, or other use cases.

Static IP address or DHCP server

As a best practice, use a DHCP server for long-term management of the cluster machines. If a DHCP service is not available for your user-provisioned infrastructure (UPI) because of single points of failure, dependencies, or compliance policies, use static IP addresses for all the nodes.

To set up static IP addresses or configure special settings such as bonding, you can use one of the following approaches:

- ▶ Pass special kernel parameters when you start the live installer, as described in Chapter 6, “Installing from ISO instead of TFTP” in *Red Hat OpenShift V4.X and IBM Cloud Pak on IBM Power Systems Volume 2*, SG24-8486.
- ▶ Use a machine config to copy networking files to the installed system. To do so, set up a static IP configuration for an RHCOS node, as shown in this [Red Hat solution](#).
- ▶ Configure networking from a live installer shell prompt by using available Red Hat Enterprise Linux tools, such as `nmcli` or `nmtui`, as shown in “Configuring networking from a live installer shell prompt by using nmcli or nmtui” on page 145.

Configuring networking from a live installer shell prompt by using nmcli or nmtui

To configure an ISO installation, complete the following steps:

1. Start the IBM Power server by using an ISO installer. When the system is successfully running, you see the live system shell prompt, as shown in Figure 4-20.

```

Red Hat Enterprise Linux CoreOS 410.84.202210040010-0 (Dotp) 4.10
Ignition: ran on 2022/11/02 02:49:01 UTC (this boot)
Ignition: no config provided by user
SSH host key: SHA256:w6yfsKwc7K6qA/IX0QH1/kw1ya28TYEFzi9PkjeLIao (ECDSA)
SSH host key: SHA256:muggghGNP8+D5KDLMbaTq.jPyuevbNBG7u0HtggNg9IW4 (ED25519)
SSH host key: SHA256:5S6erMJ9wP9fFtX+d0.j7ourPIa.jZ.jGSNJ0AU6ucK7m0 (RSA)
localhost login: core (automatic login)

#####
Welcome to the CoreOS live environment. This system is running completely
from memory, making it a good candidate for hardware discovery and
installing persistently to disk. Here is an example of running an install
to disk via coreos-installer:

sudo coreos-installer install /dev/sda \
--ignition-url https://example.com/example.ign

You may configure networking via 'sudo nmcli' or 'sudo nmtui' and have
that configuration persist into the installed system by passing the
'--copy-network' argument to 'coreos-installer install'. Please run
'coreos-installer install --help' for more information on the possible
install options.
#####

[core@localhost ~]#

```

Figure 4-20 CoreOS live system shell prompt

2. From the live system shell prompt, configure networking for the live system by using available Red Hat Enterprise Linux tools, such as `nmcli` or `nmtui`. Example 4-14 shows some example `nmcli` commands.

Example 4-14 The nmcli command and sample output

```

[core@localhost ~]$ sudo nmcli connection show
NAME                                UUID
TYPE      DEVICE
Wired connection 1                  99615422-096b-4aa1-9e5d-cd1e4587c9ec  ethernet
enp1s0
[core@localhost ~]$
[core@localhost ~]$ sudo nmcli connection modify
99615422-096b-4aa1-9e5d-cd1e4587c9ec ipv4.addresses ip_address/subnet_mask

```

```
[core@localhost ~]$ sudo nmcli connection modify
99615422-096b-4aa1-9e5d-cd1e4587c9ec ipv4.gateway gateway_ip_address
[core@localhost ~]$ sudo nmcli connection modify
99615422-096b-4aa1-9e5d-cd1e4587c9ec ipv4.dns dns_ip_address
[core@localhost ~]$ sudo nmcli connection modify
99615422-096b-4aa1-9e5d-cd1e4587c9ec ipv4.method manual
[core@localhost ~]$ sudo nmcli connection up 99615422-096b-4aa1-9e5d-cd1e4587c9ec
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/2)

[core@localhost ~]$ ping gateway_ip_address
```

3. Run the **coreos-installer** command to install the system, as shown in Example 4-15, and add the **--copy-network** option to copy the networking configuration.

Example 4-15 The coreos-installer command and sample output

```
[core@localhost ~]$ sudo coreos-installer install --copy-network
--ignition-url=http://web_server_ip:port/ign/bootstrap1.ign /dev/vda
--insecure-ignition
Installing Red Hat Enterprise Linux CoreOS 410.84.202210040010-0 (Ootpa)
.....
> Read disk 3.8 Gib/3.8 Gib (100%)
Writing Ignition config
Copying networking configuration from /etc/NetworkManager/system-connections/
Copying /etc/NetworkManager/system-connections/Wired connection 1.nmconnection to
installed system.
Install complete.
```

Note: The **--copy-network** option copies only the networking configuration that is under `/etc/NetworkManager/system-connections`. It does not copy the system hostname.

For the static hostname, you can create separate ignition files (*.ign) with a customized hostname configuration for all the nodes. You can modify ignition files manually by using tools like `filetranspile` or `butane`. For more information, see this [Red Hat document](#).

4. Restart into the installed system.
5. Copy these settings to the installed system so that they take effect when the installed system first starts. Repeat the process for each of your nodes. You also can configure a bonding interface at the live system shell prompt and verify the network settings before you install the system.

For more information about how to use the `nmcli` and `coreos-installer` command, see this [Red Hat OpenShift document](#), and this [Red Hat Enterprise document for Red Hat Enterprise Linux 8](#).

Configuring the load balancer

A best practice for production environments is to have two load balancers. Depending on the architecture, you might need different types of load balancers for different scopes.

For example:

- ▶ Local Traffic Managers (LTMs) or Enterprise Load Balancers (ELBs): Provide load-balancing services between two or more servers or applications if there is a local system failure. Required for single Red Hat OpenShift clusters.
- ▶ Global Traffic Managers (GTMs): Provide load-balancing services between two or more sites or geographic locations. Required for two or more Red Hat OpenShift clusters across multiple geographic locations.

You can install two Red Hat Enterprise Linux partitions running HAProxy and keepalived for your production environments, and you must configure the HAProxy and keepalived software in the load-balancer-defined machine. The keepalived is used to implement a dedicated active and passive load balancer across two load-balancer servers, which forward traffic to a pool of two real servers and share a virtual IP address for the client system.

For more information, see the following resources:

- ▶ [Set up HAProxy as a load balancer.](#)
- ▶ [Configure a load balancer by using keepalived.](#)

Creating an automated etcd backup in Red Hat OpenShift 4.x

When your cluster is running, it is a best practice to take a backup so that if something fails in a later operation, you do not need to reinstall your cluster. For more information, see Chapter 6, “Backing up your cluster” in *Red Hat OpenShift V4.X and IBM Cloud Pak on IBM Power Systems Volume 2*, SG24-8486. Automated backups can help with the recovery of one or more master node clusters on Red Hat OpenShift 4.x while providing a minimum RPO.

Here are the necessary resources that you need to create automatic backups by using cronjob from Red Hat OpenShift:

- ▶ Namespace
- ▶ Service Account
- ▶ Cluster Role
- ▶ Cluster Role Binding
- ▶ Set Privileges for Service Account
- ▶ cronjob

For step-by-step instructions, see [How to Create Automated etcd Backup in Red Hat OpenShift 4.x](#). Another option is the [etcd backup cronjob](#), which uses Network File System (NFS) persistent storage to store the backup files.



IBM Cloud Paks on Red Hat OpenShift running on IBM Power

IBM Cloud Paks are pre-certified, containerized software, and foundational services that provide customers with a common operations and integration framework. Cloud Paks are built on Red Hat OpenShift, so you can build once and deploy anywhere. This chapter describes these Cloud Paks that run on IBM Power:

- ▶ IBM Cloud Pak for Data
- ▶ IBM Cloud Pak for Business Automation (IBM CP4BA)
- ▶ IBM Cloud Pak for Integration (IBM CP4I)
- ▶ IBM Cloud Pak for Watson AIOps
- ▶ IBM Cloud Pak for WebSphere Hybrid Edition

This chapter contains the following topics:

- ▶ Introduction
- ▶ IBM Cloud Paks
- ▶ IBM Cloud Paks offerings on IBM Power
- ▶ IBM Cloud Pak for Watson AIOps and IBM Cloud Pak for Data
- ▶ IBM Db2 workloads on IBM Cloud Pak for Data on IBM Power

5.1 Introduction

The 2020s are an age of uncertainty. Rapid change shifted market dynamics and upended old business models, which revealed vast disconnects between digital investments and customer needs.

Your business must adapt to these changes. Firms with the technological and strategic ability to proactively rethink core business concepts and change ahead of competitors are growing over three times their industry averages.¹ To keep up, it is time to rethink how you use technologies like the cloud, artificial intelligence (AI), and automation to accelerate innovation, speed time to market, and meet your evolving customer expectations.

IBM Cloud Paks are your path toward this digital transformation. This chapter looks at how IBM Cloud Paks can help you as you move forward on your journey to hybrid multicloud.

5.2 IBM Cloud Paks

IBM Cloud Paks are AI-powered software for hybrid cloud that is designed to help you advance digital transformation with prediction, security, automation, and modernization capabilities. By using them, you can develop applications once and deploy them anywhere, integrate security across your IT landscape, and automate operations with intelligent workflows. Deploy your applications across any cloud to accelerate development, deliver seamless integration, and enhance collaboration and efficiency.

IBM Cloud Paks are pre-integrated containerized software that is built on Red Hat OpenShift and designed to help you develop and consume cloud services anywhere and from any cloud to modernize and make your data work for you wherever you are. Flexibly and quickly consume and manage all deployments with a governed, protected, and unified platform that delivers consistency across software tools and that is continuously available (from the data center all the way to the edge).

With the deployment and configuration of IBM Cloud Paks, enterprises can rapidly and reliably accelerate the journey to hybrid cloud. IBM Cloud Paks provides an open, fast, and more secure way to move core business applications to any cloud. IBM Cloud Paks is a full-stack, converged infrastructure with a virtualized cloud hosting environment that helps to extend applications to cloud.

Here are some of the features of IBM Cloud Paks:

- ▶ **Portable:** IBM Cloud Paks can be run anywhere. Its applications can run on any hybrid cloud environment. Applications can run on an on-premises infrastructure, on a public hybrid cloud infrastructure, or in an integrated system that leverages a common set of Kubernetes skills.
- ▶ **Secure:** IBM Cloud Paks are certified by IBM, with up-to-date vulnerability scanning software to provide cloud security protection of sensitive data and full-stack support from hardware to applications.
- ▶ **Expandable:** IBM Cloud Paks are pre-integrated to deliver use cases like application deployment and process automation.

Using IBM Cloud Paks to build your enterprise environment provides several benefits:

- ▶ Provides a foundation to rapidly address business requirements and build new capabilities into your applications to provide your business with a stronger competitive position.
- ▶ Builds applications to run where they provide the best benefit to your business: on-premises, private cloud, public cloud, or hybrid multicloud.
- ▶ Provides more efficiency by automating operations in hybrid multicloud environments.
- ▶ Integrates common industry components to build, move, and manage quickly your applications and data.
- ▶ Provides full software stack support, which provides ongoing security, compliance, and version compatibility.

The following sections provide a high-level overview of the IBM Cloud Pak offerings.

IBM Cloud Pak for Data

The IBM Cloud Pak for Data platform helps improve productivity and reduce complexity. IBM Cloud Pak for Data helps you build a data fabric that connects the siloed data that is distributed across your hybrid cloud landscape. This product offers a wide selection of IBM and third-party services that span the entire data lifecycle.

IBM Cloud Pak for Business Automation

IBM CP4BA is a modular set of integrated software components that is built for any hybrid cloud, and designed to automate work and accelerate business growth. This end-to-end automation platform helps you analyze workflows, design AI-infused apps with low-code tools, assign tasks to bots, and track performance. With this offering, you can transform fragmented workflows to stay competitive, boost efficiency, and reduce operational costs.

IBM Cloud Pak for Watson AIOps

Innovate faster, reduce operational cost, and transform IT operations across a changing landscape with an AIOps platform that delivers visibility into performance data and dependencies across environments. Embrace AI, machine learning, and automation to help IT operations managers and Site Reliability Engineers (SREs) address incident management and remediation. IBM Cloud Pak for Watson AIOps integrates the Infrastructure Management and Monitoring capabilities in IBM Cloud Pak for Multicloud Management as part of the IBM strategy to enable AI-powered automation for IT operations and management.

IBM Cloud Pak for Integration

IBM CP4I is a hybrid integration platform that applies the functions of closed-loop AI automation to support multiple styles of integration. The platform provides a comprehensive set of integration tools within a single, unified experience to connect applications and data across any cloud or on-premises environment. IBM CP4I integration software unlocks business data silos and assets as application programming interfaces (APIs), connects cloud and on-premises apps, and protects in-flight data integrity with enterprise messaging.

IBM Cloud Pak for Network Automation

IBM Cloud Pak for Network Automation is an intelligent cloud platform that enables the automation and orchestration of network operations so that Communication Service Providers and Managed Service Providers can transform their networks, evolve to zero-touch operations, reduce OpEx, and deliver services faster.

IBM Cloud Pak for Security

IBM Cloud Pak for Security can help you gain deeper insights, mitigate risks, and accelerate responses. With an open security platform that can advance your zero trust strategy, you can use your existing investments while leaving your data where it is, which helps your team become more efficient and collaborative.

Although at the time of writing all the Cloud Paks are certified to run on Intel based cloud services (private, public, or hybrid), not all of them are certified to run on IBM Power processor-based servers. In the following section, we describe which of the IBM Cloud Pak offerings are certified to run on IBM Power processor-based cloud environments. This list will change as more IBM Cloud Paks are tested and certified on IBM Power processor-based platforms.

5.3 IBM Cloud Paks offerings on IBM Power

At the time of writing, IBM certified a portion of the IBM Cloud Pak solutions to run on IBM Power. Figure 5-1 shows the IBM Cloud Paks capabilities that are supported on IBM Power.

IBM WebSphere Hybrid Edition <small>(Formerly)</small> IBM Cloud Pak for Applications	IBM Cloud Pak for Integration	IBM Cloud Pak for Watson for AI Ops	IBM Cloud Pak for Data
<ul style="list-style-type: none"> • IBM WebSphere Application Server • IBM WebSphere Liberty • Network Deployment <p>+ Application Modernization tools</p> <ul style="list-style-type: none"> • Transformation Advisor 	<ul style="list-style-type: none"> • App Connect Designer • Platform Navigator • App Connect Enterprise • IBM MQ Advanced • IBM MQ Native HA • IBM Event Streams (Kafka) <p>IBM Cloud Pak for Business Automation</p> <ul style="list-style-type: none"> • FileNet Content Manager • Business Automation Workflow • Business Automation Studio • Automation Decision Services • Enterprise Records • Operation Decision Manager • Application Designer 	<p>Infrastructure Automation</p> <ul style="list-style-type: none"> • Deployment automation for VM environments • Monitoring of VM env • Service Library extension <p>IBM Instana Observability</p> <ul style="list-style-type: none"> • Monitor Hybrid Multicloud • Support for IBM AIX, IBM I, and Linux <p>IBM Turbonomics</p> <ul style="list-style-type: none"> • Optimize Red Hat OpenShift Hybrid Multicloud <p>Red Hat Advanced Cluster Manager</p> <ul style="list-style-type: none"> • Manage Hybrid Multicloud Red Hat OpenShift Container Platform • Governance, Risk, and Compliance 	<ul style="list-style-type: none"> • IBM Db2 Advanced • IBM Db2 Data Management Console • IBM Db2 Warehouse

Figure 5-1 IBM Cloud Paks offering on IBM Power

5.3.1 IBM Cloud Pak for Data

IBM Cloud Pak for Data is IBM offering to modernize customer environments. IBM Cloud Pak provides data lakes with the latest analytics innovations, security, and the flexibility of hybrid cloud. This offering provides a full suite of IBM analytics capabilities that are available with IBM Cloud Pak for Data without moving data or rewriting any of your existing applications.

The integration of IBM Data and AI recognized solutions, such as IBM DataStage®, IBM Cognos®, IBM Watson Studio, and IBM Watson Knowledge Catalog, in to IBM Cloud Pak for Data reduces the time-to-value for business, lowers the total cost of ownership (TCO), and helps ensure compliance, security, and governance. This approach addresses the following Data Fabric use cases in a unique and collaborative platform:

- ▶ Multi-cloud Data Integration
- ▶ Data Governance and Privacy
- ▶ Customer 360
- ▶ Machine Learning Operations (MLOps) and Trustworthy AI
- ▶ Data Observability

Customers can query all data sources (both inside and outside) their data lake and see a seamless single view of all data with complete security and without data movement. Then, they can use these data sets for data science, and machine learning at petabyte scale to gain faster insights and make better decisions.

Figure 5-2 provides an overview of the capabilities of IBM Cloud Pak for Data.

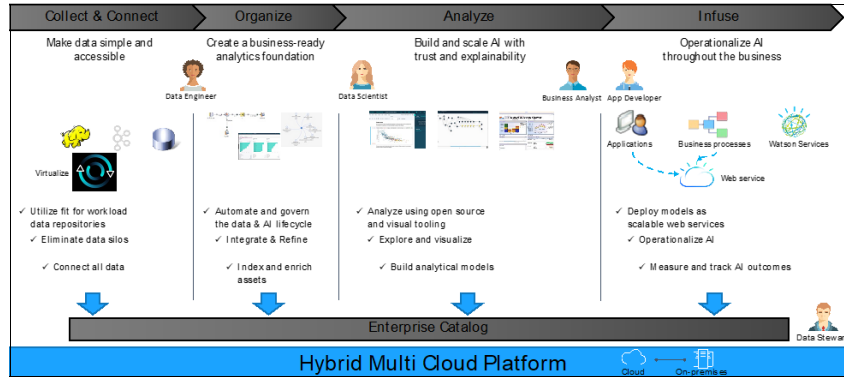


Figure 5-2 IBM Cloud Pak for Data overview

IBM Cloud Pak for Data provides automated data preparation, model development, and feature engineering. Developing insights with analytics and integrating insights into business operations can be time-intensive and difficult to scale. A solution that connects data, analytics and AI, and operations is required for enterprises to unlock the full potential of their data to serve business needs.

For more information about IBM Cloud Pak for Data, see 5.4.2, “IBM Cloud Pak for Data” on page 158.

For more information about installation assistance, see 6.2.2, “Installing IBM Cloud Pak for Data on Red Hat OpenShift” on page 184.

For more information about an example use case of IBM Cloud Pak for Data, see 5.5.4, “Additional Db2 use cases” on page 172 which provides implementation details.

For more information about the requirements of IBM Cloud Pak for Data 4.5, see [Hardware Requirements](#) and [Software Requirements](#).

5.3.2 IBM Cloud Pak for Business Automation

IBM CP4BA enables process automation through the broadest set of AI-powered automation software. IBM CP4BA brings together process mining, Robotic Process Automation, operational intelligence, and a core set of automation capabilities to automate all types of work. The containerized software is powered by AI and runs in hybrid cloud environments so that you can deploy it anywhere. The software can run the workload in any environment that best suits the customer’s needs, such as a private cloud or local cloud.

IBM CP4BA capabilities include the following ones:

- Document processing** IBM CP4BA extracts data from structured, semi-structured, and unstructured documents. It automatically detects and corrects data that is extracted incorrectly.
- Content services** IBM CP4BA allows for the classification, management, and access to these digital assets. It provides secure access to enterprise content from anywhere.
- Decision management** IBM CP4BA automates the decisions with business rules. IBM CP4BA rapidly adapts to change with business-friendly tools and increases the consistency and auditability of business policies. It integrates with predictive analytics for real-time response.
- Workflow automation** IBM CP4BA is designed for human and automated activities to better manage internal workloads. Improve consistency across business operations with increased visibility and reduced cycle time to increase straight-through processing.

Figure 5-3 provides an overview of IBM Cloud Pak for Automation.

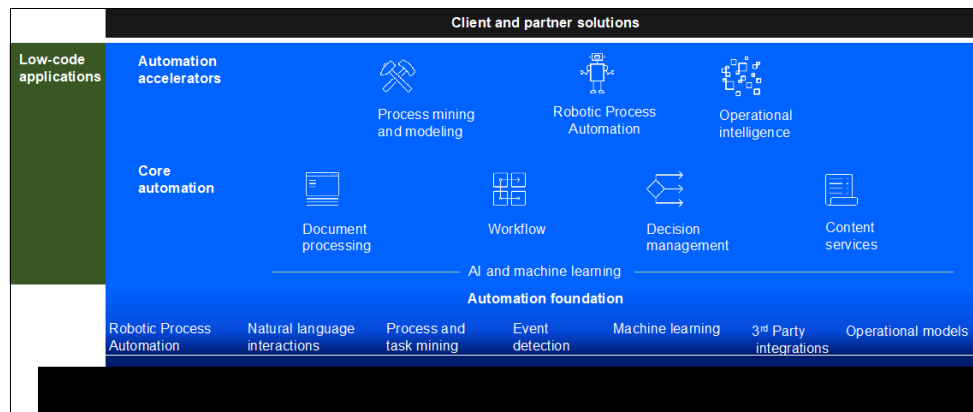


Figure 5-3 IBM Cloud Pak for Automation

5.3.3 IBM Cloud Pak for Integration

IBM CP4I is a hybrid integration solution. IBM CP4I combines applications, APIs, messaging, events, high-speed data transfer, and secure gateway capabilities with AI-powered automation. It comes with a pre-integrated set of capabilities, which include API lifecycle management, application and data integration, messaging and events, high-speed transfer, and integration security.

IBM CP4I runs on Red Hat OpenShift. It is a software solution that streamlines operations, workloads, and clusters across multiple cloud environments. The software solution integrates with commonly used tools and applications that you already might have for operations. IBM CP4I minimizes the workflow interruption to the cloud. IBM CP4I leverages an Automation Foundation set of capabilities that is consistent across all IBM Cloud Paks.

IBM CP4I includes the following capabilities:

- ▶ API Management
- ▶ Application integration
- ▶ End-to-end security
- ▶ Enterprise messaging

- ▶ Event streaming
- ▶ High-speed data transfer

IBM CP4I Create quickly exposes data, events, microservices, enterprise applications, and software as a service (SaaS) services as APIs through open standards. It organizes, creates versions, curates, and publishes any API through a full lifecycle.

IBM CP4I Secure applies built-in and extensible policies to secure, control, and mediate the delivery of APIs with unmatched scale.

IBM CP4I Socialize allows developers to easily find, understand, try, and subscribe to your APIs through a branded self-service portal.

Figure 5-4 provides an overview of IBM CP4I.

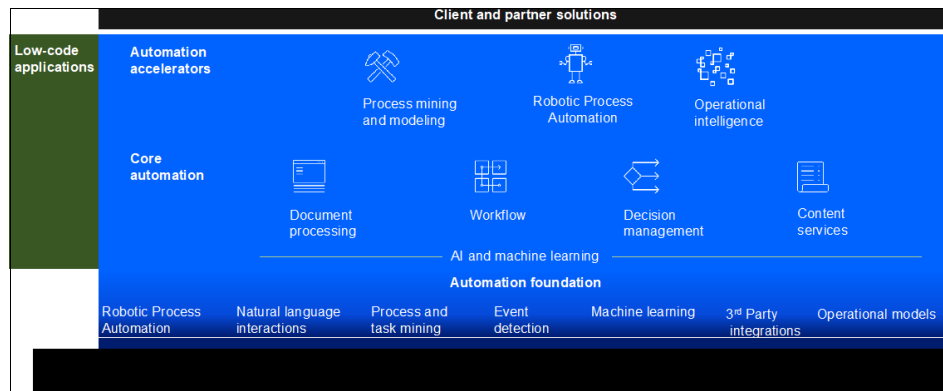


Figure 5-4 IBM Cloud Pak for Integration

5.3.4 IBM Cloud Pak for Watson AIOps

IBM Cloud Pak for Watson AIOps is a solution that help users innovate faster, reduce operational cost, and transform IT operations across a changing landscape with an AIOps solution that delivers visibility into performance data and dependencies.

Organizations are turning to AI-powered automation to improve speed, utilization, and service delivery to solve for the problem of increasingly stretched IT operations resources. AI for IT operations is a mixture of AI and existing IT processes, such as incident and problem management, which provides operational benefits, such as predictive alerts and outage avoidance. AI for IT operations provides a better digital experience for your customers.

IBM Cloud Pak for Watson AIOps simplifies operations management as follows:

- ▶ Accurately identifying and resolving emerging IT outages
- ▶ Assigning IT incidents properly and with context
- ▶ Diagnosing problems faster in complex environments

IBM Cloud Pak for Watson AIOps provides Infrastructure Automation, which is a stand-alone capability module. Infrastructure Automation is built on open-source Terraform and ManageIQ.

Infrastructure Management provides two distinct features:

- ▶ *Managed services use* Cloud Automation Manager and its self-service capabilities to orchestrate resources. It uses Terraform and Ansible technology, and it provides a standardized and compliant environment for DevOps teams with a built-in self-service catalog that integrates with enterprise-wide catalogs through APIs.

Service Composer is a comprehensive IT workflow orchestration capability that you use to drag and publish services by using integration with Ansible. Workflows connect Terraform with Ansible playbooks for configuration management that supports infrastructure as code (IaC) and GitOps best practices. Service Composer includes supported versions of Terraform that work across multiple cloud environments like RHEV, VMWare, and OpenStack, and across multiple cloud providers, like AWS, Microsoft Azure, Google, and IBM Cloud.

- ▶ Infrastructure management, previously known as IBM Red Hat CloudForms, discovers, manages, and automates the deployment of virtual machines (VMs), cloud services, and Kubernetes clusters. It addresses the challenges of managing hybrid IT environments. Infrastructure management is based on ManageIQ and CloudForms. Based on client feedback, IBM focused on increasing the flexible integration capabilities of the Hybrid Cloud infrastructure. For organizations that want to use hybrid cloud, infrastructure management helps them accomplish that task. It integrates with clouds, containers, on-premises, or a virtual infrastructure. Infrastructure management discovers inventory across virtualization, container, network, and storage management systems; and maps relationships and listens for changes to build a rich model. It scans the contents of VMs, hosts, and containers, and combines with auto-discovery data to create advanced security and compliance policies.

Figure 5-5 shows the capabilities of IBM Cloud Pak for Watson AIOps.

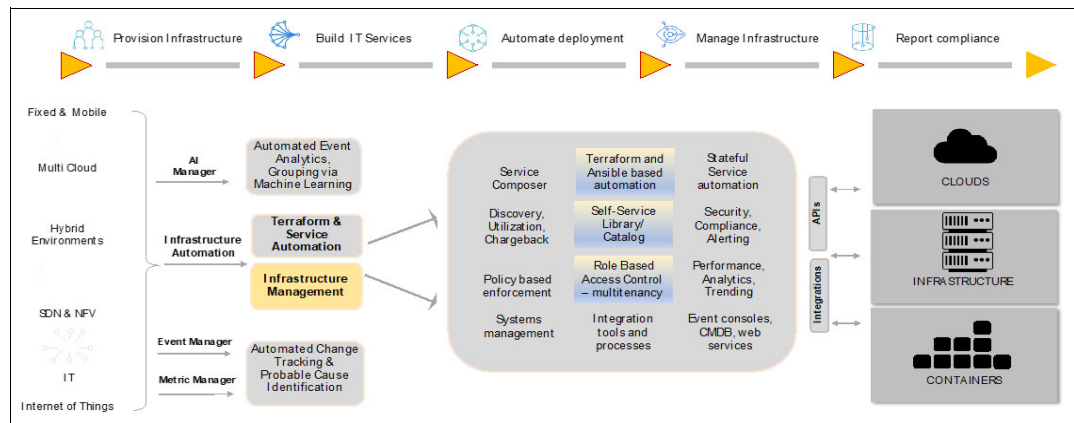


Figure 5-5 IBM Cloud Pak for Watson AIOps

5.3.5 IBM Cloud Pak for WebSphere Hybrid Edition

IBM WebSphere Hybrid Edition combines all the products in the IBM WebSphere portfolio with application modernization tools. It is designed to help your business in its digital transformation initiatives, from optimization and modernization to cloud enablement. IBM Cloud Pak for WebSphere Hybrid Edition replaces Cloud Pak for Applications.

IBM Cloud Pak for Watson AIOps powers automation by using diverse data sets from an entire range of hybrid environments, from cloud to on-premises. It brings the information together across IT operations. With this IBM Cloud Pak, you can tap into shared automation services to get insight into how the processes run. Visualize hotspots and bottlenecks, and pinpoint what to fix by using event detection to prioritize what issues to address first.

WebSphere Hybrid Edition includes six IBM Solutions, as illustrated in Figure 5-6:

- ▶ IBM WebSphere Application Server
- ▶ IBM WebSphere Liberty
- ▶ IBM WebSphere Application Server Network Deployment
- ▶ IBM Cloud Transformation Advisor
- ▶ IBM Mono2Micro
- ▶ IBM Cloud Foundry Migration Runtime

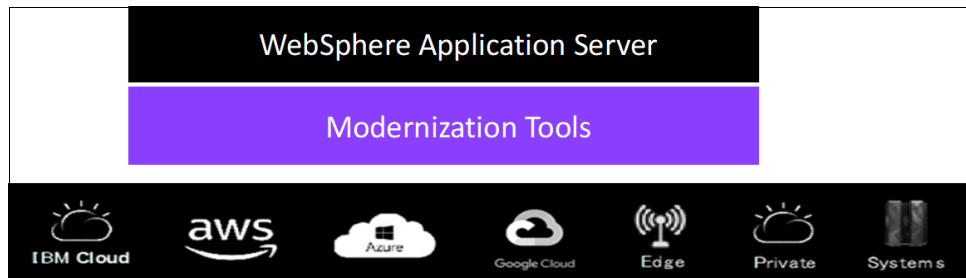


Figure 5-6 IBM Cloud Pak for WebSphere Hybrid Edition

WebSphere Hybrid Edition is unique because it offers the flexibility of all WebSphere editions and is designed for on-premises, cloud, and hybrid cloud deployments. WebSphere Hybrid Edition is the solution for growing the existing WebSphere Application Server installation base and providing for application modernization and new cloud native applications by using the Liberty run time. WebSphere Migration Toolkit can help you move to new versions of WebSphere Application Server or move from traditional WebSphere Application Server to the Liberty profile.

5.4 IBM Cloud Pak for Watson AIOps and IBM Cloud Pak for Data

Section 5.3, “IBM Cloud Paks offerings on IBM Power” on page 152 provided an overview of all the IBM Cloud Paks that are certified to run on IBM Power servers. This section provides a deeper dive into two of those IBM Cloud Paks that can be especially useful in your journey to cloud. The infrastructure management and monitoring capabilities of IBM Cloud Pak for Watson AIOps brings AI-powered automation to help you better manage your hybrid cloud environment. To complement these capabilities, IBM Cloud Pak for Data provides a unique and collaborative platform that enables Multi-cloud Data Integration, data governance and privacy, Customer 360, MLOps and Trustworthy AI, and data observability, while also ensuring compliance, security, and governance.

This section describes both IBM Cloud Paks in more detail and their uses.

5.4.1 IBM Cloud Pak for Watson AIOps

IBM Cloud Pak for Watson AIOps provides a unique set of tools to help you design and run AIOps on a cloud infrastructure. The functions that are provided are described in this section.

Infrastructure management and monitoring

IBM Cloud Automation Manager automates the provisioning of the infrastructure and VM applications across multiple cloud environments with optional workflow orchestration.

Application management

The application development and enhancement process is DevOps-based, unified, and simplified, and it is made more efficient by using the application management functions of IBM Cloud Pak for Watson AIOps. This capability is built on a Kubernetes resource-based application model, along with a channel- and subscription-based deployment model. The model unifies and simplifies application management across single and multi-cluster scenarios.

The application management capability uses a channel- and subscription-based model to optimize continuous and automated delivery in managed clusters. Application release management is automated through DevOps platform for operations like deployment, manage, and monitor.

Application model

Application development and deployment are two phases in a project's lifecycle. Application development is often restricted within fewer instances. However, in a production deployment, multiple instances are made available when scalability becomes a major factor. In a DevOps environment, roles can be defined for development and deployment. A development team must focus more on application development and defining application resources. A DevOps admin can set up a channels and a subscription model for a faster, smoother, and more efficient deployment of the application to achieve high scalability in a managed cluster environment.

Application resource

In IBM Cloud Pak for Watson AIOps, resources are classified as application resources and deployable resources. These resources are further divided into channel, subscription, and placement rule resources to facilitate deploying, updating, and managing applications that are spread across clusters. Both single and multi-cluster applications use the same Kubernetes specifications, but multi-cluster applications involve more automation of the deployment and application management lifecycle.

5.4.2 IBM Cloud Pak for Data

As companies become data-driven and expand the potential of AI, they must use data from diverse and complex sources across multi-hybrid-cloud environments, and deal with different data formats and standards. With huge amounts of data that is produced every day, enterprises are challenged to understand what data matters for their business. They also are challenged to process, govern, and manipulate all this data to ensure trust, and accessibility to the entire enterprise for both technical and business users.

To simplify the usage of the data, IBM introduced IBM Cloud Pak for Data to implement the data fabric approach to accelerate governance and the journey to AI.

IBM Cloud Pak for Data brings together all the critical cloud, data, and AI capabilities as containerized microservices over Red Hat OpenShift to deliver the unified hybrid multi-cloud platform.

IBM AI ladder

The IBM AI ladder begins with data. You get higher business value when you perform business-assisted functions such as analytics, machine learning, or AI on top of the data. The IBM AI ladder provides a prescriptive approach for gathering, preparing, and using data.

The AI ladder consists of four rungs:

- ▶ **Collect:** Make it easier to consume and access data.
- ▶ **Organize:** Create a trusted analytics foundation on data with business meaning.
- ▶ **Analyze:** Scale business insights with AI everywhere.
- ▶ **Infuse:** Operationalize AI with trust and transparency.

The AI ladder is designed to simplify and automate how an enterprise turns data into insights by unifying the collection, organization, and analysis of data, regardless of where it is within a secure hybrid cloud platform.

The following priorities are built into the IBM technologies to support the AI ladder:

- ▶ **Simplicity:** Different kinds of users can use tools that support their skill levels and goals, from “no code” to “low code” to programmatic.
- ▶ **Integration:** As users go from one rung of the ladder to the next, the transitions are seamless.
- ▶ **Automation:** The most common and important tasks have intelligence that is included so that users focus on innovation rather than repetitive tasks.

IBM Cloud Pak for Data and Data Fabric

Data has been growing fast over the past few years, and it will grow at a continually expanding rate. The data complexity problem is huge and increases with data volume growth. The data growth and management challenge severely inhibits the ability of an enterprise to become data-driven, which makes it difficult for the enterprise to get the full value out of their data.

The ability to use data effectively drives innovation in an enterprise, and the most innovative and best performing enterprises are data-driven. Enterprises must establish an architecture that simplifies data access to enable them to connect the right data to the right people at the right time.

A data fabric can provide a new architecture that an enterprise can use to become data-driven. The data fabric provides an abstraction layer so that data can be shared and used across a hybrid multicloud landscape, connecting data from various sources, such as on-premises data lakes and data warehouses, multiple cloud providers, and existing applications, including both legacy and SaaS solutions, while maintaining data observability.

Figure 5-7 shows some of the benefits of using a data fabric to connect your data.

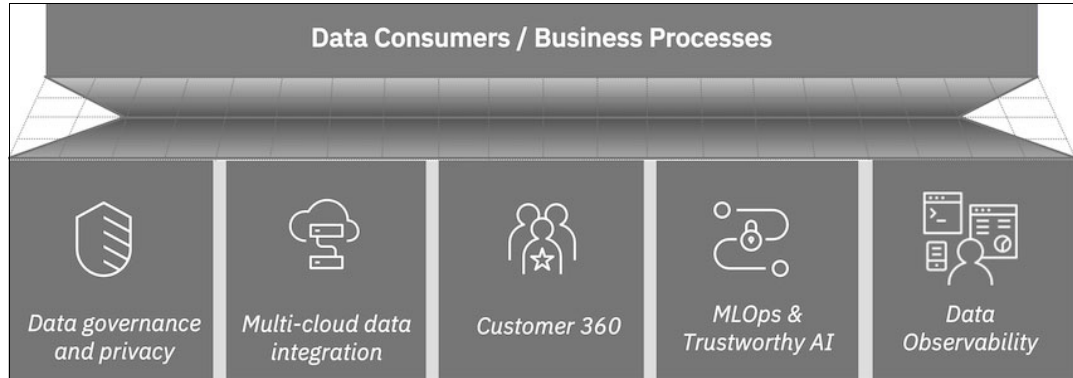


Figure 5-7 IBM data fabric approach

The data fabric approach enables enterprise to manage, govern, and use data to provide agility, gain speed, and maintain trust with deep enforcement of governance, security, and regulatory compliance. It also reduces the costs of integration, including reduced bandwidth costs and processing requirements, by keeping and processing the data where it is and by supporting the full DataOps lifecycle.

IBM Cloud Pak for Data helps you to implement a data fabric to support the use cases that are shown in Figure 5-7 by bringing a unique data platform to your enterprise, which results in a faster time-to-value for your business.

Data fabric use cases

This section describes some of the most common use cases for a data fabric in your enterprises.

Data governance and privacy

To ensure that data consumers are connected to the right data at the right time (and ensure data trust), they must find the required data and access it.

It is difficult to find data within the enterprise. There are many different data sources with different data admin and access rules. There are legacy systems (sometimes without documentation), and there are spreadsheets and data sets that are not in a system but in business users machines. Your data often looks like the first image in Figure 5-8 on page 161, that is, spread across multiple tables and rooms. Finding important and current data is nearly impossible in this scenario.

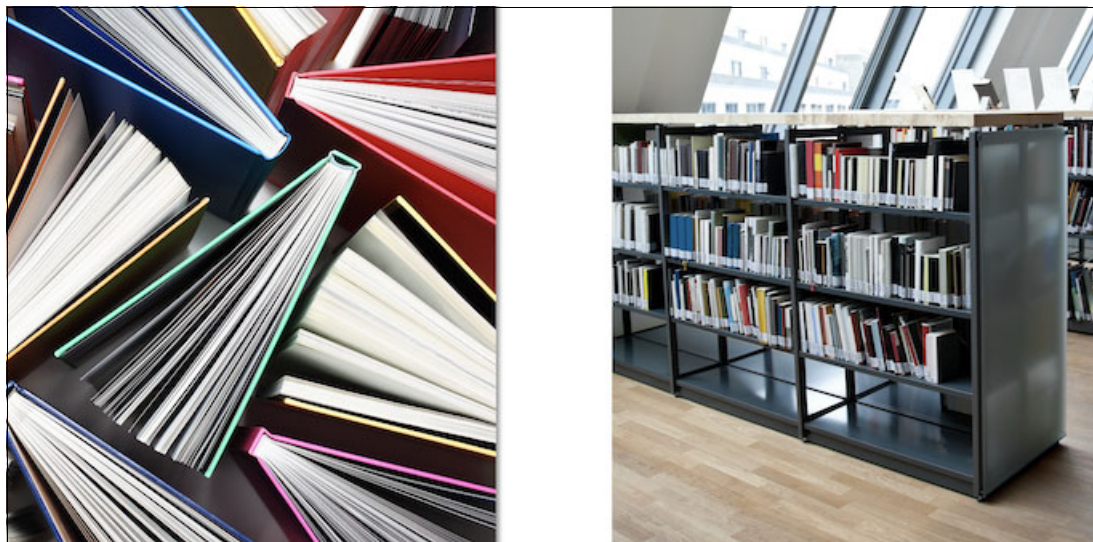


Figure 5-8 The importance of a data catalog

A *data catalog* can help by registering data assets across the enterprise and organizing access to that data. A data catalog works like the bookshelves in a library with books that are organized on the shelves, and the readers can use the library's catalog to find the right book, as shown in the second image in Figure 5-8.

The data catalog is used by the users, both technical and non-technical business users, to find and access the right data at the right time to be used in their applications. The catalog creates a collaborative environment where the data can be identified, including details and reviews of the data. You can run data quality analysis and even enrich the data if needed. IBM Cloud Pak For Data helps your enterprise users understand the available data while establishing an environment to support highly automated data processes and maintaining consistent governance.

IBM Cloud Pak for Data can automatically apply industry-specific regulatory policies and rules to data assets; apply your enterprise-specific policies and rules; provide automated data governance and privacy to ensure data trust; maintain privacy; enable protection; and enable security and compliance.

Having a metadata and governance layer applied to all data, analytics, and AI initiatives increases visibility and collaboration on any hybrid multi-cloud environment, and it facilitates the anonymization of training data and test sets by maintaining the integrity of the data that is used.

IBM Cloud Pak for Data implements an AI-augmented data catalog that business users can use to easily understand, collaborate with, enrich, and access the right data from a unique and centralized platform that is shareable by the entire enterprise while enabling access to data without having to move or copy it. This catalog simplifies your data management processes and helps to ensure data trust and governance.

Multi-cloud Data Integration

Enterprises continue to move to hybrid multicloud solutions and require many integration styles and techniques to access their data. Data must be extracted, ingested, streamed, virtualized, and transformed by using an extensive library of hybrid cloud data sources, which are driven by automation and data policies that maximize performance while minimizing storage and egress costs.

IBM Cloud Pak for Data integrates data across hybrid multi-cloud environments to accelerate time-to-value by democratizing data for AI, business intelligence, and applications. This approach creates a unified view of your enterprise data to enable consistency across your operations and applications, and to intelligently integrate and automate data engineering tasks while enhancing data integration to support your business requirements.

Figure 5-9 shows the Multi-cloud Data Integration approach.

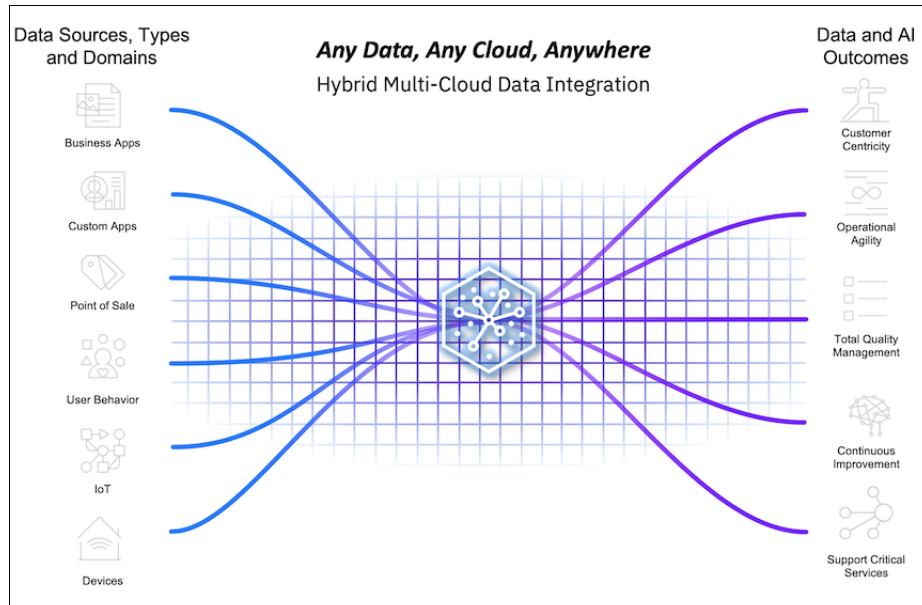


Figure 5-9 Multi-cloud Data Integration

Different data sources, data types, and domains; data from business applications and custom applications that are collected from devices and IoT; and data about user behavior is processed by IBM Cloud Pak for Data. IBM Cloud Pak for data helps extract, ingest, prepare, transform, and deliver data across the enterprise. This approach results in improved outcomes for different initiatives by using data and AI.

Any data access or delivery process is automated and streamlined to help speed up the data delivery process. IBM Cloud Pak for Data provides automatic workload balancing and elastic scaling capabilities to run jobs in any environment and with any amount of data. In addition, resiliency and continuous integration are built in. Delivery automation and continuous analysis can be performed automatically in real time wherever the data is, which minimizes storage and egress costs.

To consolidate and simplify IT infrastructures, IBM Cloud Pak for Data can run anywhere (on-premises or any cloud) and automate data operations to deliver trusted data to business users by integrating data and cataloging it, which prevents delays and disruptions of mission-critical data through data resilience and easy data access.

Improved data integration flows use IBM DataStage for Extract, Transform, and Load (ETL), Data Virtualization, and real-time capture to optimize access to many diverse data sources by using extensive native connectors. Quality analysis and remediation can be natively added into data pipelines to avoid costly downstream processing. IBM Cloud Pak for Data supports the full DataOps lifecycle (governance, quality, master data, integration, and collaboration), which is integrated into a unique data platform.

Customer 360

Enterprises have different challenges with customer data. They have multiple systems and applications, data silos, multiple domains, and lack of data quality. The customer uses all these applications.

When the customer with an overdue mortgage receives a special offer to borrow more money from an enterprise, it is a signal that the enterprise needs a single view of the customer, that is, a customer 360. This is one single example, but enterprises have complex challenges that are based on the lack of data consistency and standardization. With multiples copies of customer data in different formats, you can spend days or even weeks analyzing customer data.

With multiple systems and local processes, enterprises must use manual reconciliation and manual remediation to improve the quality of the customer's data. An enterprise also requires complex data integration processes regarding the movement of data. Also, inconsistent business rules can cause a lack of trust among users.

Customer 360 drives different enterprise initiatives to reduce costs and optimize productivity from the following perspectives:

- ▶ **Analytical:** Focus on customer care and on the single view of the customer so that businesses can answer deeper and more complex questions about the customer.
- ▶ **Governance:** Prepare and maintain customer high and trustworthy data quality.
- ▶ **Compliance:** Accelerate compliance and fraud prevention, and ensure data privacy.
- ▶ **Operational:** Infuse the single view of the customer into the applications, tools, and systems so that it is at the right time by the business, and to provide hyper-personalization.
- ▶ **Prescriptive:** Design for better outcomes, for example, predict churn or define next best offer or next best action by using AI powered patterns and algorithms.

To provide a comprehensive Customer 360 view, IBM Cloud Pak for Data integrates and matches data across multiple systems and domains to break down data silos and create an integrated view of data. Then, it applies entities resolution algorithms and ML-powered probabilistic matching to increase the results. Users spend more time applying AI and analytics to business challenges versus wasting time on hunting for quality data and consolidating the customer view.

IBM Cloud Pak for Data provides a centralized platform to govern the data and apply the business rules; automatically apply remediation and reconciliation based on data quality analysis and defined rules; and provides mechanisms to infuse the single view of the customer into various applications, tools, and systems.

Machine Learning Operations and Trustworthy AI

Ensuring AI operation and trusting machine learning models is a challenge. Enterprises must deal with the AI lifecycle and ensure the quality of the AI models that are deployed. Also, the quality of the test data sets that are used during the training phase and the fairness of the model to avoid bias are important.

The IBM data fabric approach to MLOps and Trustworthy AI is based on three important AI lifecycle phases that are implemented by IBM Cloud Pak for Data (see Figure 5-10):

- ▶ **Data:** Collect and prepare data by ensuring governance, quality, and trust.
- ▶ **Model:** Build, deploy, and monitor models by ensuring fairness, robustness, and explainability.
- ▶ **Process:** Use automation to drive consistency, efficiency, and transparency for AI.

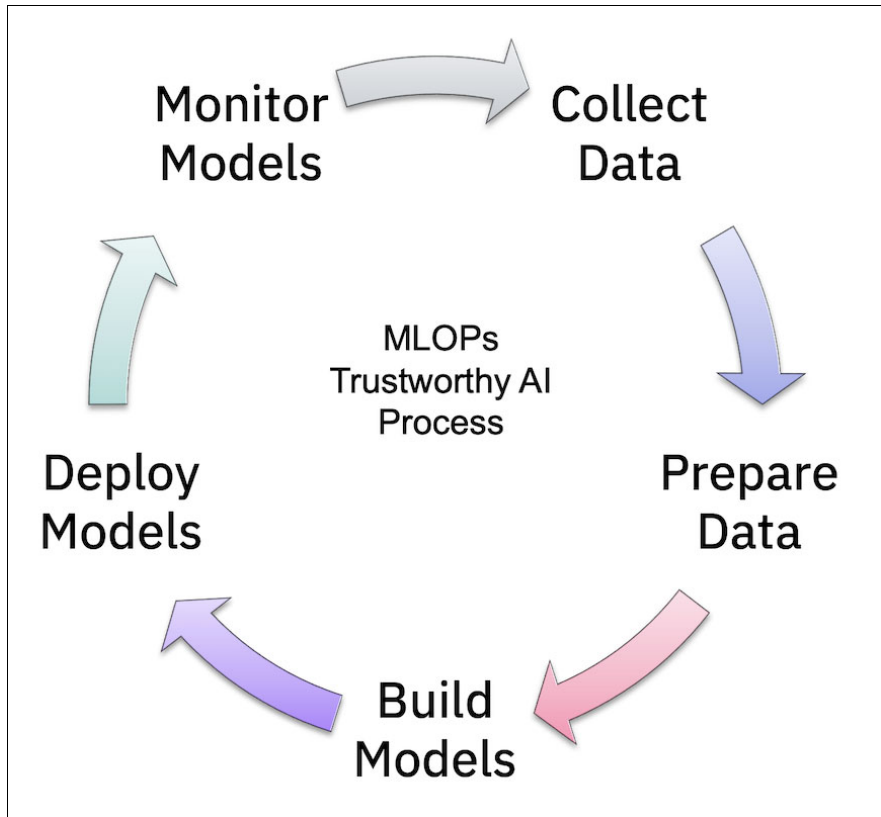


Figure 5-10 Machine Learning Operations and Trustworthy AI process

The data catalog that is provided by IBM Cloud Pak for Data provides automatic application of policies and rules while ensuring privacy and security, and the ability to access and connect data. The “collect and prepare data” phase is easily achieved while providing the data scientist access to quality and trustworthy data in a collaborative platform that enables data exploration, data refinery, and data visualization with minimal code or even no code.

To build the AI models, IBM Cloud Pak for Data provides no-code and low-code services, and it supports open-source tools. AutoAI automates several aspects of the MLOps lifecycle, including feature transformation, feature engineering, algorithm selection, and model training to accelerate the models that are built while providing efficiency improvements for data scientists.

Several end-to-end tools enable one-click deployment of the model into production, with version control, auto-retrain, and pre-packaged dependencies. The fairness of the model is improved by bias-detection monitoring, and bias can be removed from an unfair model to ensure that only fair models are used in production. This approach provides equitable outcomes from models across all groups.

Explainability is provided by IBM Cloud Pak for Data so that you can understand the model outcomes and decisions that are made from it. The features that most influence the prediction are shown, and you can see the what-if analysis, which better explains the results.

Model drift is monitored over time, and you can automatically retrain the model. This adaptation to changing model parameters can ensure trust on models while ensuring that business objectives are met. All these tools and features enable trusted models that can be deployed quickly into production and continuously improved, and stay reliable over time.

Pipelines enable automation of the AI flow and orchestration of entire lifecycle by retrieving fresh training data, retraining the model, and deploying the model into production. You can automatically capture model metadata and lineages to ensure that you understand the model and its governance and its risks, which increases efficiency and transparency for AI.

Data Observability

Data Observability is the ability to detect data changes and anomalies and to provide proactive awareness of data health while interactively correcting and resolving data quality issues.

A Data Observability framework expands Data Fabric use cases by including remediation of data quality issues in data governance, such as the following instances:

- ▶ Monitoring data-in-motion in existing data pipelines
- ▶ Data monitoring
- ▶ Model monitoring for the Trustworthy AI use case

The main goal is to detect data issues earlier and resolve them faster before they impact the business, and to enhance reliability.

IBM Databand.ai solution for Data Observability is based on four main steps:

Collect	Collect metadata automatically from diverse solutions to gain visibility into mission-critical metadata.
Profile	Build profiles of a historical baseline by using the common data pipeline's behavior, which is continuously compared with the behavior of the data-in-motion.
Alert	Alert you when deviations of rules or other anomalies are detected. Alerts are created and sent to the data managers.
Resolve	Resolve issues by creating smart workflows to remediate data quality issue. Scale-up the resources to process more data if the volume increases. Route bad data with low quality to a staging area, Retrain models to improve accuracy based on the performance monitoring.

These steps are designed to automatically resolve issues in your Data Fabric.

When you combine Data Observability with IBM Instana for enterprise application observability (see 2.6.2, "IBM Instana" on page 26 for more detail) and MLOps and Trustworthy AI for monitoring models, you can enable end-to-end enterprise observability and reliability.

For more information about IBM Data Fabric, use cases for IBM Cloud Pak For Data, and sign up for a no-charge trial, see [IBM Data Fabric Solutions](#).

5.5 IBM Db2 workloads on IBM Cloud Pak for Data on IBM Power

IBM Db2 is a world-class, enterprise relational database management system (RDBMS). Db2 provides advanced data management and analytics capabilities for your online transactional processing (OLTP) workloads. Traditionally, enterprises run the IBM Db2 database in a dedicated on-premises environment, which often is hosted on a logical partition (LPAR) running on an IBM Power server that uses either the IBM AIX or Red Hat Enterprise Linux for IBM Power Little Endian (Red Hat Enterprise Linux ppc64le) operating systems. Using IBM Cloud Pak for Data, you can change the deployment architecture to a container-native environment by deploying one or more Db2 databases on the IBM Cloud Pak for Data software running on Red Hat OpenShift.

This approach is not a good option for all Db2 workloads, but there are some workloads that can benefit from a cloud-native containerized approach. IBM also provides tools to help you integrate a Db2 instance running on IBM Cloud Pak for Data with your existing enterprise Db2 databases. The IBM Db2 Data Gate service provides a gateway to synchronize data from Db2 for IBM z/OS that is hosted on IBM zSystems to any IBM Cloud Pak for Data environment.

This gateway can extract, load, synchronize, and propagate your mission-critical data to a target database on IBM Cloud Pak for Data for quick access to your high-volume, read-only transactional, and analytic applications. The IBM Db2 Data Gate service is described in more detail in Chapter 2, “Cloud Pak for Data services overview”, in *IBM Cloud Pak for Data Version 4.5: A practical, hands-on guide with best practices, examples, use cases, and walk-throughs*, SG24-8522. For more information, see this [IBM Documentation web page](#).

Integrating a Db2 database with IBM Cloud Pak for Data can be useful in the following situations:

- ▶ You need your transactional data to be governed, such as data from a website, bank, or retail store.
- ▶ You want to create a replica of your transactional database so that you can run analytics without affecting regular business operations.
- ▶ Ensure the integrity of your data by using an ACID-compliant database.
- ▶ You need a low-latency database.
- ▶ You need real-time insight into your business operations.

By using the Db2 operator and containers in IBM Cloud Pak for Data, you can deploy Db2 by using a cloud-native model, which can provide the following benefits:

- ▶ Less operating system patching needed due to the reduction of the infrastructure,
- ▶ Faster time to value when deploying Db2 databases.
- ▶ Improved lifecycle management:
 - Similar to a cloud service, it is easy to install, upgrade, and manage Db2.
 - Ability to deploy your Db2 database in minutes.
 - Faster backup and restore through snapshot-based mechanisms.
- ▶ A rich ecosystem that includes a Data Management Console, REST, and Graph.
- ▶ Extended availability of Db2 with a multitier resiliency strategy.

- ▶ Support for software-defined storage, such as Red Hat OpenShift Data Foundation, IBM Storage Scale Container Storage Interface (CSI), and other world-leading storage providers.
- ▶ Reduction of the amount of infrastructure, that is, number of LPARs and amount of storage.

You can create a Db2 database in your Red Hat OpenShift environment by using IBM Cloud Pak for Data, or you can quickly move an existing on-premises Db2 Linux, UNIX, or Windows (LUW) database that is running on an IBM PowerLinux server to IBM Cloud Pak for Data by using the Db2-click-to-containerize automation tools. An example of how to install Db2 with IBM Cloud Pak for Data is shown in 6.2, “Running Db2 workloads on IBM Cloud Pak for Data on IBM Power” on page 183.

The rest of this chapter explores the features of Db2 in IBM Cloud Pak for Data and the benefits that your enterprise might gain.

5.5.1 IBM Db2

The scalability of Db2, which includes the number of cores, memory size, and storage capacity, provides an RDBMS that can handle any type of workload. These capabilities are available in the Db2 service that is deployed as a set of microservices that is running in a container environment. This containerized version of Db2 for Cloud Pak for Data makes it highly secure, available, and scalable without any performance compromises.

Db2 databases are fully integrated in IBM Cloud Pak for Data, which enables them to work seamlessly with data governance and AI services to provide secure in-depth analysis of your data.

Working with a Db2 database

After you create a Db2 database, you can use the integrated database console to perform common activities to manage and work with the database. From the console, you can perform the following tasks:

- ▶ Explore the database through its schemas, tables, views, and columns, which include viewing the privileges for these database objects.
- ▶ Monitor databases through key metrics, such as Availability, Responsiveness, Throughput, Resource usage, Contention, and Time Spent.
- ▶ Manage access to the objects in the database.
- ▶ Load data from flat files that are stored on various storage types.
- ▶ Run SQL and maintain scripts for reuse.

For more information about the integrated database console, see 5.5.3, “IBM Db2 Data Management Console” on page 170.

Initial setup and configuration considerations

Setting up the Db2 service and databases in IBM Cloud Pak for Data requires some extra steps and considerations compared to some of the other IBM Cloud Pak for Data services.

Before installing the Db2 service, consider using dedicated compute nodes for the Db2 database. In a Red Hat OpenShift cluster, compute nodes or worker nodes run the applications.

Installing Db2 on a dedicated compute node is a best practice for production, and it is important for databases that are performing heavy workloads. Setting up dedicated nodes for your Db2 database involves Red Hat OpenShift taints and tolerations to provide node exclusivity. You also must create a custom security context constraint (SCC) that is used during the installation.

After you install the Db2 service and before you create your database, consider disabling the default automatic setting of interprocess communication (IPC) kernel parameters so that you can set the kernel parameters manually. Also, consider enabling the `hostIPC` option for the cluster so that kernel parameters can be tuned for the worker nodes in the cluster. Now, you can use the Red Hat OpenShift Machine Config Operator to tune the worker IPC kernel parameters from the control or the master nodes.

Now, you can create your database in your IBM Cloud Pak for Data cluster. You can specify the number of nodes that can be used by the database, including the cores per node and memory per node. You also can specify the usage of dedicated nodes by specifying the label for those dedicated nodes.

You also can set the page size for the database to 16 K or 32 K. One of the last steps is to set the storage locations for your system data, user data, backup data, transactional logs, and temporary table space data. This data can be stored together in a single storage location, but it is a best practice to use separate locations, especially among the user data, transactional logs, and backup data.

Section 6.2, “Running Db2 workloads on IBM Cloud Pak for Data on IBM Power” on page 183 demonstrates how to install a Db2 database on IBM Power by using a container-native platform that uses IBM Cloud Pak for Data on Red Hat OpenShift.

Learning more

For more information about the Db2 service, see the following resources:

- ▶ IBM Documentation:
 - [Db2 on Cloud Pak for Data](#)
 - [Preparing to install the Db2 service](#)
 - [Installing the Db2 service](#)
 - [Postinstallation setup for the Db2 service](#)
- ▶ Video: [Db2 on IBM Cloud Pak for Data platform](#)
- ▶ Blog: [The Hidden History of Db2](#)

5.5.2 IBM Db2 Warehouse

IBM Db2 Warehouse is an enterprise-ready data warehouse that is used globally. Db2 Warehouse provides in-memory data processing, columnar data storage, and in-database analytics for online analytical processing (OLAP) workloads.

The scalability and performance of Db2 Warehouse through its massively parallel processing (MPP) architecture provides a data warehouse that can handle any type of analytical workloads. These workloads include complex queries, and predictive model building, testing, and deployment.

IBM Cloud Pak for Data automatically creates a suitable data warehouse environment. For a single node, the warehouse uses a symmetric multiprocessing (SMP) architecture for cost-efficiency. For two or more nodes, the warehouse uses an MPP architecture for high availability (HA) and improved performance.

By using the Db2 Warehouse operator and containers in IBM Cloud Pak for Data, you can deploy a Db2 Warehouse instance that uses a cloud-native model and provides the following benefits:

- ▶ Lifecycle management: Similar to a cloud service, it is easy to install, upgrade, and manage Db2 Warehouse.
- ▶ Ability to deploy your Db2 Warehouse database in minutes.
- ▶ A rich ecosystem of available tools and interfaces, including a Data Management Console, REST API, and graphing tools.
- ▶ Extended availability of Db2 Warehouse with a multitier resiliency strategy.
- ▶ Support for software-defined storage, such as Red Hat OpenShift Data Foundation, IBM Storage Scale CSI, and other world-leading storage providers.

Using the Db2 Warehouse database

Using a Db2 Warehouse database that is integrated with IBM Cloud Pak for Data can be useful in the following situations:

- ▶ You have developers who must create small-scale database management systems for development and test work. For example, you must test new applications and data sources in a development environment before you move them to a production environment.
- ▶ You want to accelerate line-of-business analytics projects by creating a data mart service that combines a governed data source with analytic techniques.
- ▶ You want to deliver self-service analytics solutions and applications that use data that is generated from new sources and imported directly into the private cloud warehouse.
- ▶ You want to migrate a subset of applications or data from an on-premises data warehouse to a private cloud.
- ▶ You want to save money and improve performance by migrating on-premises data marts or an on-premises data warehouse to a cloud-native data warehouse.
- ▶ You want to support data scientists who are designing queries, must store data locally, and need to use a logical representation.
- ▶ You want to reduce network traffic and improve analytic performance by storing your data near your Analytics Engine.
- ▶ You have multiple departments, and each department requires their own database management system.

After you create a Db2 Warehouse database, you can use the integrated database console to perform the following common tasks to manage and work with the database:

- ▶ Explore the database through its schemas, tables, views, and columns, which include viewing the privileges for these database objects.
- ▶ Monitor databases through key metrics, such as Availability, Responsiveness, Throughput, Resource usage, Contention, and Time Spent.
- ▶ Manage access to the objects in the database.
- ▶ Load data from flat files that are stored on various storage types.
- ▶ Run SQL and maintain scripts for reuse.

For more information about the integrated database console, see 5.5.3, “IBM Db2 Data Management Console” on page 170.

Initial setup and configuration considerations

Setting up the Db2 Warehouse service and data warehouse databases in IBM Cloud Pak for Data requires some extra steps and considerations compared to some of the other IBM Cloud Pak for Data services.

Before installing the Db2 Warehouse service, consider using dedicated worker nodes for the Db2 Warehouse database, which is important for data warehouse databases. Setting up dedicated nodes for your Db2 Warehouse database involves taints and tolerations to provide node exclusivity.

If you plan to use an MPP configuration, you must designate specific network communication ports on the worker nodes, and ensure that these ports are not blocked. You also can improve performance in an MPP configuration by establishing an inter-pod communication network. Also, create a custom SCC that is used during the installation.

After you install the Db2 Warehouse service and before you create your data warehouse database, consider disabling the default automatic setting of the IPC kernel parameters so that you can set the kernel parameters manually. Also, consider enabling the `hostIPC` option for the cluster so that you can tune kernel parameters for the worker nodes in the cluster. Now, you can use the Red Hat OpenShift Machine Config Operator to tune the worker IPC kernel parameters from the master nodes.

Now, you can create your data warehouse database in your IBM Cloud Pak for Data cluster. You can choose to use the SMP or MPP architectures with the following configurations:

- ▶ Single physical node with one LPAR (default)
- ▶ Single physical node with multiple LPARs
- ▶ Multiple physical nodes with multiple LPARs

These configurations can be deployed on dedicated nodes by specifying the label for the dedicated nodes.

One of the last steps is to set the storage locations for your system data, user data, backup data, transactional logs, and temporary table space data. This data can be stored together in a single storage location, but as a best practice, consider using separate locations, especially among the user data, transactional logs, and backup data.

After the data warehouse database is created, you can start using the database by creating your first set of tables and loading data into the tables.

Learn more

For more information about the Db2 Warehouse service, see the following resources:

- ▶ [Db2 Warehouse on Cloud Pak for Data](#)
- ▶ [Preparing to install the Db2 Warehouse service](#)
- ▶ [Installing the Db2 Warehouse service](#)
- ▶ [Postinstallation setup for the Db2 Warehouse service](#)

5.5.3 IBM Db2 Data Management Console

The IBM Db2 Data Management Console service is a database management tool platform that you can use to administer and optimize the performance of your integrated IBM Db2 databases on IBM Cloud Pak for Data. These integrated databases include Db2, Db2 Warehouse, IBM Db2 Big SQL, and Data Virtualization, which you can manage and monitor from a single user interface (UI) console.

By using this console, you can perform the following tasks for your integrated databases:

- ▶ Administer databases.
- ▶ Work with database objects and utilities.
- ▶ Develop and run SQL scripts.
- ▶ Move and load large amounts of data into databases for in-depth analysis.
- ▶ Monitor the performance of your IBM Cloud Pak for Data integrated Db2 database.

Using the Db2 Data Management Console

The console home page provides an overview of all the IBM Cloud Pak for Data integrated databases that you are monitoring. This home page includes the status of database connections and monitoring metrics that you can use to analyze and improve the performance of your databases.

Figure 5-11 shows the summary page of the Db2 Data Management Console.

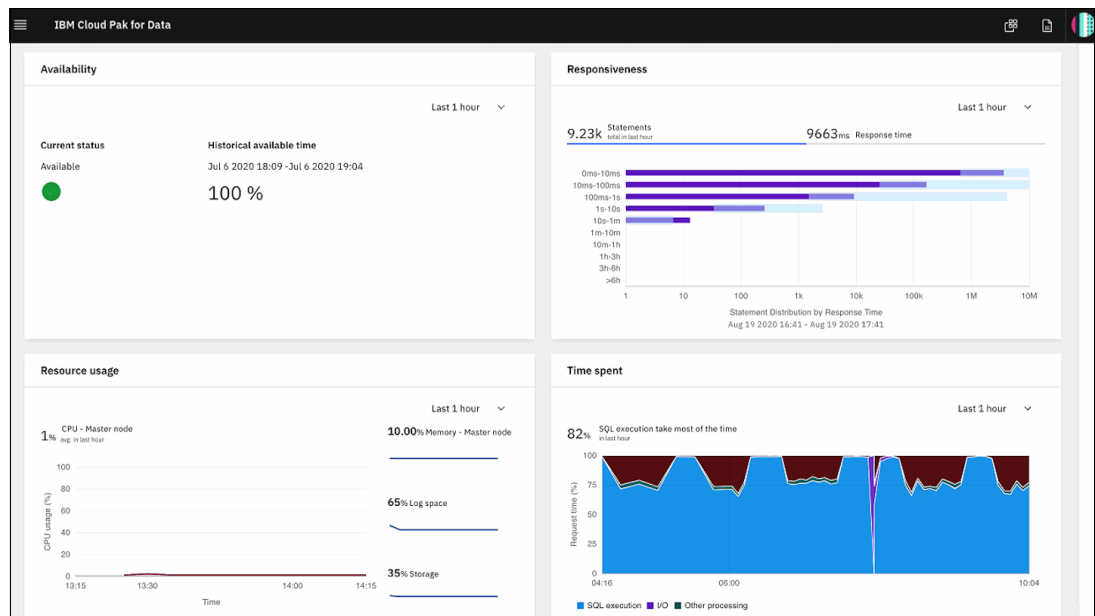


Figure 5-11 Db2 Data Management Console summary page

From the console, you can perform the following tasks:

- ▶ Explore integrated databases through schemas, tables, views, and columns.
- ▶ Monitor integrated databases through key metrics, such as Availability, Responsiveness, Throughput, Resource usage, Contention, and Time Spent.
- ▶ Run SQL and maintain scripts for reuse.
- ▶ Load data from flat files that are stored on various storage types.
- ▶ Tune single SQL statements and query workloads.
- ▶ Create and schedule jobs.
- ▶ Manage alerts.
- ▶ Create monitoring reports to compare and analyze different data sets.
- ▶ Set up and manage monitor profiles and event monitor profiles.

Initial setup and configuration considerations

After installing the Db2 Management Console, you must provision an instance of the service. Only one instance of the console can exist in an IBM Cloud Pak for Data deployment.

To provision an instance, first select the plan size for the compute resources: small, medium, or large. Then, configure the storage resources by providing the storage class and the amount of storage for your persistent storage.

When the console instance is provisioned, use the console to manage and maintain your integrated databases.

Learn more

For more information about the Db2 Data Management Console service, see the following resources:

- ▶ [IBM Db2 Data Management Console on Cloud Pak for Data](#)
- ▶ [Installing Db2 Data Management Console](#)
- ▶ [Provisioning the service \(Db2 Data Management Console\)](#)
- ▶ [Db2 Data Management Console for Cloud Pak for Data demonstration](#)
- ▶ [APIs](#)

5.5.4 Additional Db2 use cases

There are many other use cases for using Db2 in an IBM Cloud Pak for Data environment that we have not explored. Db2 can be integrated into many workflows within the IBM Cloud Pak for Data, including Data Virtualization, IBM Watson AI, and IBM Cognos reporting. For more information, see 6.2, “Running Db2 workloads on IBM Cloud Pak for Data on IBM Power” on page 183.

Many of these use cases are covered in *IBM Cloud Pak for Data Version 4.5: A practical, hands-on guide with best practices, examples, use cases, and walk-throughs*, SG24-8522.



Use cases

This chapter presents several use cases for running applications on IBM Power that leverage the performance capabilities of the platform.

This chapter contains the following topics:

- ▶ Artificial intelligence inferencing with Red Hat OpenShift and IBM Power10 processor-based servers
- ▶ Running Db2 workloads on IBM Cloud Pak for Data on IBM Power
- ▶ GitOps for system configuration

6.1 Artificial intelligence inferencing with Red Hat OpenShift and IBM Power10 processor-based servers

Artificial intelligence (AI) inferencing is the process of using a trained machine learning model to make predictions. Applications rely on inferencing to apply predictions to business problems in real time. High-speed inferencing is challenging because it requires numerous, computationally expensive multiplications of large matrices, especially for deep learning models that are used for tasks such as image recognition, speech to text, natural language processing, and time series forecasting.

The IBM Power E1080 (Power E1080) features a state-of-the-art processor that delivers 4.3X containerized throughput per core when compared to x86. This section describes how AI inferencing workloads can be optimized by running on IBM Power10 processor-based nodes and leveraging the processor's Matrix Math Accelerator (MMA) capabilities.

6.1.1 Matrix Math Accelerator

MMA is a set of instructions and data types that accelerates matrix multiplication on the IBM Power10 processor. Packages that are optimized to leverage MMA deliver 5X faster AI inferencing per socket on Power E1080 compared to IBM Power 980 (Power 980).¹

Existing AI inferencing workloads do not need source code changes to leverage MMA on IBM Power10 processor-based servers, but to see a performance improvement from MMA, you must ensure packages such as OpenBLAS, PyTorch, ONNX Runtime, and TensorFlow are obtained from a distribution that enables the MMA capabilities.

6.1.2 Optimized AI libraries

The Open Cognitive Environment (open-ce) is a community driven set of packages for machine learning on IBM Power. The packages are installed within a Conda environment. For more information about Conda, see [Conda documentation](#).

By going to the main [Open-CE GitHub Page](#), users can build the open-ce packages themselves. For those users that want the precompiled binary files, there are multiple organizations that distribute precompiled Conda packages.

One company that distributes precompiled packages is Rocket Software. A benefit of the Rocket Software RocketCE distribution is that these packages are built to use the MMA capabilities of the IBM Power10 processor. For more information about the latest releases, see the [RocketCE for IBM Power forum](#).

6.1.3 ONNX Runtime

ONNX Runtime is a cross-platform accelerator for AI inferencing. It can be used with models that are built from using PyTorch, TensorFlow, and many other frameworks. The accelerator provides performance improvements by leveraging both hardware capabilities and graph rewrites.

When ONNX Runtime is obtained from RocketCE, it leverages the MMA capabilities that are included in the IBM Power10 processor.

¹ <https://dach.tdsynnex.com/ch/blog/wp-content/uploads/2021/10/IBM-Power-E1080-Client-Presentation-Switzerland.pdf>

6.1.4 Inferencing engine tutorial

This section describes how to build and deploy a classifier that identifies an image as one of a thousand classes. The classifier is deployed as an Red Hat OpenShift container on an IBM Power10 processor-based node. It uses ONNX Runtime from RocketCE to leverage the IBM Power10 processor-based server MMA technology.

The files that are referenced in this tutorial are available at [GitHub](#).

The classifier uses a pretrained ResNet model. For more information about the pretrained ResNet model, see [GitHub](#).

Conda environment

The classifier runs within a Conda environment. To use MMA, ensure that ONNX Runtime is obtained from RocketCE.

Example 6-1 shows the `environment.yaml` file for the Conda environment. Because RocketCE is listed first, packages in that library are preferred over the ones in `conda-forge`. In addition, the dependency for `onnxruntime` explicitly states that `rocketce-1.6.0` should be used.

Example 6-1 Conda YAML file

```
name: onnxruntime
channels:
  - rocketce
  - conda-forge
  - nodefaults
dependencies:
  - python=3.9
  - pip
  - rocketce/label/rocketce-1.6.0::onnxruntime
  - numpy
  - pillow
  - flask
  - scipy
```

The `environment.yaml` file is used to create the Conda environment as part of the container build. The classifier runs within the environment.

Container file

The container build initializes a Conda environment. The complete container file is a *Dockerfile*, and it is available in the classifier path at [GitHub](#). You can download the project from GitHub and build the container by using `podman`. A sample build command is shown here:

```
podman build -f Dockerfile -t inference:latest
```

The container file contains instructions to download and install Conda within the image. The commands are shown in Example 6-2 on page 176. Miniconda is used for this example because it includes fewer packages by default, which decreases the size of the container. The URL to download the latest version of Miniconda for IBM Power is documented [here](#).

The `-b` option of the installer script causes all the agreements to be accepted without prompting, and the `-p` option specifies the directory to install Miniconda information.

Because each **RUN** statement creates a layer, the commands to download the installer, run the installation, and remove the installer program from the file system must be included in the same **RUN** command to reduce the size of the container image.

Example 6-2 Conda init

```
RUN wget "$MINICONDA_REPO/$MINICONDA_VERSION" -O installer.sh && \  
    chmod u+x ./installer.sh && \  
    ./installer.sh -b -p $HOME/miniconda && \  
    rm ./installer.sh  
  
RUN eval "$($HOME/miniconda/bin/conda shell.bash hook)" && \  
    conda init
```

The container file also includes commands to create the Conda environment by using the `onnxruntime_env.yaml` file (from Example 6-1 on page 175), which are shown in Example 6-3.

The **bash --login** is necessary because Conda commands require a login shell to initialize Conda.

Example 6-3 Creating a Conda environment

```
COPY onnxruntime_env.yaml .  
RUN bash --login -c 'conda env create -f onnxruntime_env.yaml'
```

The container has a few other commands to build the image. These commands are straightforward and can be reviewed by looking at the complete file in GitHub.

- ▶ The model file and class labels are downloaded and saved in the container image.
- ▶ The Python scripts are copied in to the image.
- ▶ The port that is used by the classifier service is exposed.

The container's command activates the Conda environment and starts the classifier service. A login shell is required to use the **conda activate** command. The command is shown in Example 6-4.

Example 6-4 Activating the Conda environment

```
CMD bash --login -c 'conda activate onnxruntime && python app.py'
```

Classifier service

The classifier service is implemented by using the Flask web framework and ONNX Runtime. Because the Conda environment was created with packages that support MMA, the classifier receives a performance benefit without any code modifications for IBM Power10 processor-based servers.

When the service is initialized, the trained model is loaded and ONNX Runtime is initialized. The initialization is shown in Example 6-5.

Example 6-5 Initializing the service

```
import onnxruntime as ort  
INFERENCE_SESSION = ort.InferenceSession(  
    ONNX_MODEL_FILE_PATH, providers=["CPUExecutionProvider"]  
)
```

When Conda is installed, the `conda init` command (shown in Example 6-2 on page 176) is invoked to initialize Conda when the interactive shells begin.

The logic for making a prediction has a few steps:

1. Load the image in the HTTP request.
2. Preprocess the image so that it can be passed to the model.
3. Use the ONNX Runtime inference session to make predictions.
4. Choose the top five predictions with scores greater than zero.
5. Package the predictions in a JSON format and send the response to the client.

This logic is shown in Example 6-6.

Example 6-6 Running the model to make predictions

```
@app.route("/infer", methods=["POST"])
def infer() -> flask.Response:
    """
    Runs the inference on the provided JPG file and returns a json of
    the top 5 predictions where the score is greater than zero.
    """
    # Load and preprocess the image (scale/resize/normalize)
    input_image = Image.open(io.BytesIO(flask.request.data))
    image_array = preprocess(input_image)

    # Make the single image into a batch
    batch = np.expand_dims(image_array, axis=0)

    # Make Predictions.
    # There is only one input (named "data") for this model, and with
    # a batch size of 1, there is only one row of scores as output.
    output = INFERENCE_SESSION.run([], {"data": batch})[0].flatten()
    scores = softmax(output)

    top5 = [
        Prediction(label=CLASS_LABELS[p], score=round(float(scores[p]), 3))
        for p in np.argsort(-scores)[:5]
    ]

    # Send response json
    return flask.Response(
        json.dumps({"predictions": [p.dict() for p in top5 if p.score > 0]}),
        content_type="application/json",
        status=200,
    )
```

When a container image is built for the classifier, the container can be deployed to an Red Hat OpenShift Container Platform.

Labeling Red Hat OpenShift nodes with MMA capabilities

In many environments, only a subset of the cluster's nodes have IBM Power10 processor-based server MMA capabilities. Labels and node selectors are used to ensure that the containers for AI inferencing run on nodes that support MMA to ensure that these workloads run as fast as possible.

A label can be added by using the Red Hat OpenShift user interface (UI) by selecting **Compute** → **Nodes**. After clicking the IBM Power10 processor-based server node, click the actions drop-down and click **Edit node** to modify the labels for the node, as shown in Figure 6-1. For our example, we added the label “ai.inference.accelerator=mma” to the IBM Power10 processor-based server node.

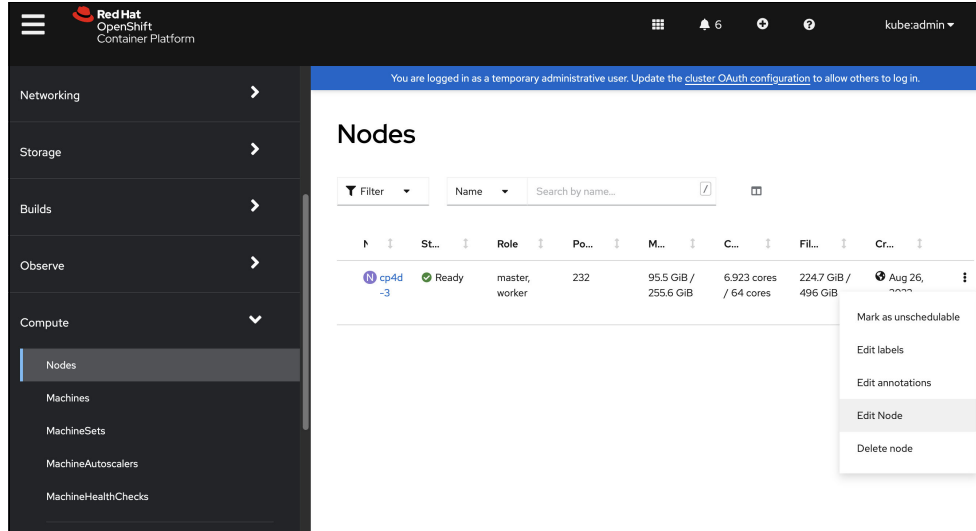


Figure 6-1 Compute node edit

Figure 6-2 shows the “mma” label that is attached to a node.

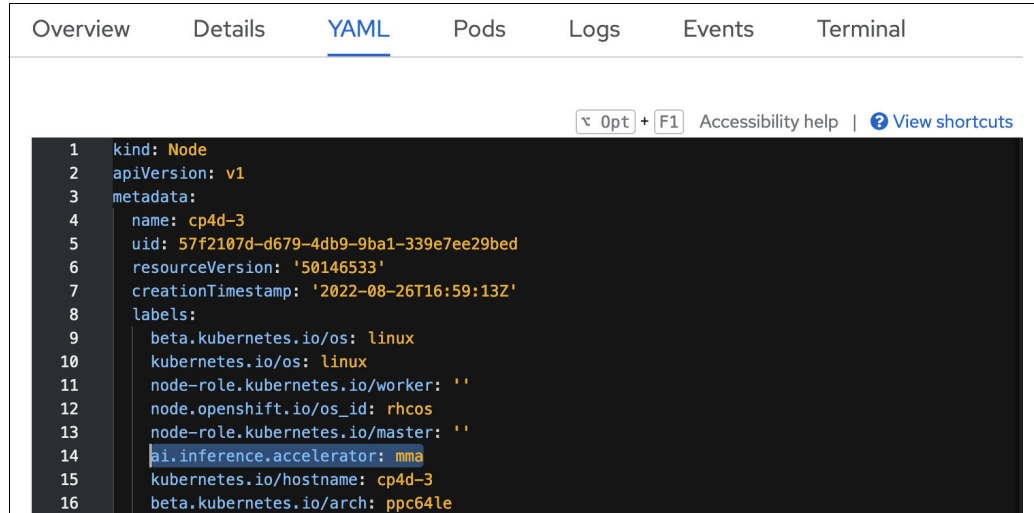


Figure 6-2 Label showing an mma example

Red Hat OpenShift deployment

To deploy, clone this [Git repository](#), which includes the YAML files that are needed. The deployment.yaml file, which is shown in Example 6-7 on page 179, shows the description of what is used to deploy the inference container. The container includes the nodeSelector, which ensures that the pods that are created are deployed to nodes that are labeled as supporting MMA.

Example 6-7 Deployment.yaml file

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: inference
  namespace: inferencep
spec:
  selector:
    matchLabels:
      app: inference
  replicas: 3
  template:
    metadata:
      labels:
        app: inference
    spec:
      containers:
        - name: inference
          image: quay.io/ntlawrence/inference:latest
          ports:
            - containerPort: 5000
      nodeSelector:
        ai.inference.accelerator: mma
```

Create the project by running `cli oc`, and then create the deployment by using the `deployment.yaml` file from the cloned Git repository, as shown in Example 6-8.

Example 6-8 Creating a project and deployment

```
# git clone https://github.com/gpadillax/ai-inference-p10.git
Cloning into 'ai-inference-p10'...
remote: Enumerating objects: 16, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 16 (delta 5), reused 16 (delta 5), pack-reused 0
Receiving objects: 100% (16/16), 6.38 KiB | 6.38 MiBps, done.
Resolving deltas: 100% (5/5), done.
```

```
# cd ai-inference-p10/
```

```
# oc new-project inferencep
Now using project "inferencep" on server
"https://api.cp4d-3.rtp.raleigh.ibm.com:6443".
```

You can add applications to this project with the `'new-app'` command. For example, try:

```
oc new-app rails-postgresql-example
```

to build a new example application in Ruby. Or use `kubectl` to deploy a simple Kubernetes application:

```
kubectl create deployment hello-node
--image=k8s.gcr.io/e2e-test-images/agnhost:2.33 -- /agnhost serve-hostname
```

```
# oc create -f deployment.yaml
deployment.apps/inference created
```

Creating a service

Services for this example are created with NodePort, which maps to the container's port 5000 so that clients can address the service without addressing individual pods.

The YAML for the service is shown in Example 6-9.

Example 6-9 Creating a service YAML

```
apiVersion: v1
kind: Service
metadata:
  name: inference
  namespace: inferencecp
spec:
  selector:
    app: inference
  type: NodePort
  ports:
    - protocol: TCP
      port: 5000
      targetPort: 5000
```

To create the service by using a CLI, run **oc create -f** with the file name `service.yaml`, as shown in Example 6-10.

Example 6-10 The create -f service command

```
# oc create -f service.yaml
service/inference created
```

Creating a route

To access the service from outside the cluster, create a route. Creating the route assigns a URL to the service, which you can find by examining the location in the route details page. The YAML to create a route is shown in Example 6-11.

Example 6-11 Creating a route

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: inference
  namespace: inferencecp
spec:
  path: /
  to:
    kind: Service
    name: inference
  port:
    targetPort: 5000
```

Create the route by running `cli oc create` with the `route.yaml` file, as shown in Example 6-12.

Example 6-12 Creating the route

```
# oc create -f route.yaml
route.route.openshift.io/inference created
```

Example Inference REST call

After creating the route, the classifier can be addressed from outside of the cluster.

Example 6-13 shows a `wget` command to download a sample image of a typewriter (shown in Figure 6-3) for classification.

Example 6-13 The wget command

```
wget
https://upload.wikimedia.org/wikipedia/commons/thumb/c/c2/Macchina_per_scrivere_elettromeccanica_-_Museo_scienza_tecnologia_Milano_10947.jpg/512px-Macchina_per_scrivere_elettromeccanica_-_Museo_scienza_tecnologia_Milano_10947.jpg -O /tmp/tw.jpg
```



Figure 6-3 Photo to be classified

Now, you can run a rest call to the classifier. The URL for the service is defined by the route details. The `/infer` path is defined by the Python application, as shown in Example 6-14.

Example 6-14 REST call to the classifier

```
# oc get route
NAME          HOST/PORT                                PATH  SERVICES
PORT  TERMINATION  WILDCARD
inference  inference-inferencecp.apps.cp4d-3.rtp.raleigh.ibm.com  /
inference  5000                                None

# curl -s -X POST --data-binary @/tmp/tw.jpg --header "Content-Type: image/jpg"
http://inference-inferencecp.apps.cp4d-3.rtp.raleigh.ibm.com/infer
```

Example 6-15 shows the response from the service.

Example 6-15 Service response

```
{
  "predictions": [
    {
      "Label": {
        "synset_id": "n04505470",
        "names": [
          "typewriter keyboard"
        ]
      },
      "score": 0.572
    },
    {
      "Label": {
        "synset_id": "n04264628",
        "names": [
          "space bar"
        ]
      },
      "score": 0.428
    }
  ]
}
```

6.1.5 Model lifecycle

This example shows how to deploy a simple classifier that optimizes performance by using MMA on IBM Power10 processor-based server. In real enterprise applications, where more sophisticated workflows are needed to train, deploy, and monitor AI solutions, Kubeflow is an open-source toolkit for Machine Learning Operations (MLOps) in containerized environments, such as Red Hat OpenShift. A brief overview of the solution is described in [MLOps with Kubeflow on IBM Power](#).

If you are interested in advanced technologies for AI and deep learning, IBM offers AI workshops and consulting services to help clients maximize the capabilities of IBM Power. We can be contacted at [Contact IBM Technology Expert Labs](#).

6.1.6 Summary

In this tutorial, we optimized the performance of a simple classifier by:

- ▶ Using the ONNX format for the model.
- ▶ Evaluating the model with a distribution of ONNX Runtime that was built to leverage MMA.
- ▶ Deploying the container to an Red Hat OpenShift cluster by targeting an IBM Power10 processor-based server with MMA capabilities.

6.2 Running Db2 workloads on IBM Cloud Pak for Data on IBM Power

This section describes a new way of running Db2 databases on IBM Power by using a container-native platform.

Instead of the traditional way of hosting one or more Db2 databases on an LPAR on IBM Power running IBM AIX or Red Hat Enterprise Linux for IBM Power Little Endian (Red Hat Enterprise Linux ppc64le) operating systems, we change the deployment architecture to a container-native environment by deploying one or more Db2 databases on the IBM Cloud Pak for Data software running on the Red Hat OpenShift Container Platform.

Switching to a container-native platform such as IBM Cloud Pak for Data on Red Hat OpenShift brings various benefits:

- ▶ Reduction of the amount of infrastructure, such as the number of LPARs or amount of storage.
- ▶ Less operating system patching is needed due to the reduction of the infrastructure.
- ▶ Faster time to value when deploying Db2 databases.
- ▶ Faster backup and restore through snapshot-based mechanisms.

Db2 running on IBM Cloud Pak for Data is described in 5.5.1, “IBM Db2” on page 167.

6.2.1 Lab environment

Our lab environment for this chapter consists of a couple of infrastructure components that host the software that we are going to use throughout this chapter:

- ▶ A Red Hat OpenShift 4.10 cluster running on eight IBM Power10 processor-based server LPARs:
 - Three master nodes, four worker nodes, and a bootstrap node.
 - The IBM Power10 processor-based server LPARs are running on dedicated cores for the master and worker nodes.
 - The bastion and the bootstrap node are running with a 1:10 PU to VP ratio.
- ▶ A bastion host running on an IBM Power10 processor-based server LPAR is running Red Hat Enterprise Linux 8.5 ppc64le Linux, which provides DNS, load balancer, Network File System (NFS), and DHCP services to the Red Hat OpenShift cluster.
- ▶ An IBM Storage Scale (Previously Spectrum Scale) 5.1.5 storage cluster:
 - Running on four IBM Power10 processor-based server LPARs, each running Red Hat Enterprise Linux 8.6 ppc64le Linux.
 - Two GUI nodes and two NSD nodes providing two GPFS file systems running on seven LUNs with 500 GB each.
 - gpfs0 has 2 TB of storage (coming from four LUNs), and gpfs1 has 1.5 TB of storage (coming from three LUNs).
- ▶ IBM Storage Scale Container Native Storage Access 5.1.5 is set up on the Red Hat OpenShift cluster and connected to external IBM Storage Scale storage cluster, and the gpfs0 file system is remotely mounted.
- ▶ A storage class that is named `ibm-spectrum-scale-csi-fileset` for the IBM Storage Scale Container Native Storage Access.

- ▶ An x86-based virtual machine (VM) running Red Hat Enterprise Linux 8.6 that we use to run the IBM Cloud Pak for Data installer.
- ▶ All the LPARs have internet connectivity.

We set up a Db2 Linux, UNIX, or Windows (LUW) 11.5 instance on one of the LPARs of the IBM Storage Scale storage clusters and create the Db2 SAMPLE database on the instance. We clone this Db2 SAMPLE database into a Db2 instance running in a container on IBM Cloud Pak for Data on Red Hat OpenShift.

6.2.2 Installing IBM Cloud Pak for Data on Red Hat OpenShift

Installing Cloud Pak for Data and its Db2 services on an Red Hat OpenShift cluster is a 4-step process. This section describes each step.

You can skip these explanations and go directly to the step if you want to type in the commands directly and you do not need the background information on why we are doing these steps.

Step 1: Setting up a client workstation

Set up an x86-based client workstation that runs the IBM Cloud Pak for Data command-line interface (CLI) (**cpd-cli**) tools and the **olm-utils** Ansible-based containerized software package that it includes. (At the time of writing, IBM Power and Linux on zSystems based client workstations are not supported for **olm-utils**). With the **cpd-cli** CLI, you can install the IBM Cloud Pak for Data software on the Red Hat OpenShift cluster.

Note: The **cpd-cli** CLI pulls **olm-utils** during the setup. The **olm-utils** container image packages Ansible-based installation scripts to administer and install IBM Cloud Pak for Data and its services, such as Db2.

To run **olm-utils**, you must have a container environment that is installed on the client workstation, such as **podman** or **docker**. In our example, we use **podman**.

Finally, set up the Red Hat OpenShift CLI (**oc**) to interact with the Red Hat OpenShift cluster directly.

Step 2: Collecting required information

Gather some required information before installing the IBM Cloud Pak for Data software by completing the following steps:

1. Obtain a valid IBM Cloud Pak for Data software license, which is an IBM entitlement application programming interface (API) key. There are two ways to obtain this license key:
 - For up to 60 days, you can obtain a 60-day trial key for IBM Cloud Pak for Data from the following URL:

<https://www.ibm.com/account/reg/us-en/signup?formid=urx-42212>
 - If you need more time, work with your IBM Sales representative to obtain an IBM standard evaluation license for IBM Cloud Pak for Data.

When either of the requests are processed, log in with your IBMid at the following URL:

<https://myibm.ibm.com/products-services/containerlibrary>

2. On the **Get entitlement key** tab, select **Copy key** to copy the entitlement key to the clipboard. Save the API key in a text file. An example is shown in Example 6-16.

Example 6-16 Example of an IBM entitlement key

```
eyJhbGciOiJIUzI1NiJ9.eyJkpc3MiOiJJQk0gTWYya2V0cGxhY2UiLCJpYXQiOiJlE2MTc3OTE2M0csImp0aSI6IjM1MDU2NzBjMTI4NTRiZmM4MTQyN2E5OFJlOWF1NjUwIn0.pCc4KoA22c9n6goFsw1R5GVrff3nyRnqNOTBYN6P-cg
```

3. Choose which IBM Cloud Pak for Data components that you want to install on your cluster. Because this chapter deals with running Db2 on IBM Cloud Pak for Data, we choose all the Db2 related services (db2oltp, db2wh, and dmc), the mandatory services (cpfs and cpd_platform), plus one optional service (scheduler), which is the scheduler service that you use to set and enforce quotas for services running on the IBM Cloud Pak for Data platform. An example list of components is shown in Example 6-17.

Example 6-17 Example list of components

```
COMPONENTS=cpfs,scheduler,cpd_platform,db2oltp,db2wh,dmc
```

4. Set up more installation variables:
 - Red Hat OpenShift cluster access details.
 - Red Hat OpenShift projects (namespaces) to create for the IBM Cloud Pak for Data cluster.
 - Storage classes to use (in our case, we are using Storage Scale Container Native, but NFS is also supported).
 - The IBM Entitlement key (see Example 6-16).
 - The IBM Cloud Pak for Data version to install (Version 4.6.0).
 - The list of components to install (see Example 6-17).

Step 3: Preparing the Red Hat OpenShift cluster

Prepare your Red Hat OpenShift cluster before you can start the installation of IBM Cloud Pak for Data. You do this step once. The preparation step completes the following actions:

- ▶ Updates the global image pull secret of the Red Hat OpenShift cluster by using the information from the IBM entitlement API key so that the Red Hat OpenShift cluster has the necessary credentials to pull the IBM Cloud Pak for Data container images that are hosted at the IBM Container registry (icr.io).
- ▶ Updates the CRI-O settings of the worker nodes on the Red Hat OpenShift cluster so that the prerequisites for Cloud Pak for Data (the pids_limit is equal or higher to 12288) of the CRI-O container run time are met.
- ▶ Updates the Db2 kubelet settings of the worker nodes on the Red Hat OpenShift cluster.

Step 4: Installing Cloud Pak for Data and Db2 services

Start the installation of IBM Cloud Pak for Data and the Db2 related services on the Red Hat OpenShift cluster.

Setting up a client workstation

1. Obtain an x86-based Linux VM or bare-metal server. In this example, we use a Red Hat Enterprise Linux 8.6 VM.
2. Log in to the VM as the root user and verify the Red Hat Enterprise Linux version and x86_64 architecture, as shown in Example 6-18.

Example 6-18 Verifying the Red Hat Enterprise Linux version

```
# cat /etc/redhat-release
Red Hat Enterprise Linux release 8.6 (Ootpa)

# uname -a
Linux clientforpowerinstall1.fyre.ibm.com 4.18.0-372.32.1.el8_6.x86_64 #1 SMP Fri
Oct 7 12:35:10 EDT 2022 x86_64 x86_64 x86_64 GNU/Linux
```

3. Install **podman** and **jq**, as shown in Example 6-19.

Example 6-19 Installing podman and jq

```
# yum -y install podman jq
# podman version
Client:      Podman Engine
Version:     4.2.0
API Version: 4.2.0
Go Version:  go1.18.7
Built:       Wed Oct 26 12:23:47 2022
OS/Arch:    linux/amd64

# jq --version
jq-1.6
```

4. Install **screen**, as shown in Example 6-20.

Example 6-20 Installing screen

```
# yum install -y --nogpgcheck
https://dl.fedoraproject.org/pub/epel/8/Everything/x86_64/Packages/s/screen-4.6
.2-12.el8.x86_64.rpm
```

5. Install the **oc** client, as shown in Example 6-21.

Example 6-21 installing the oc client

```
# wget https://mirror.openshift.com/pub/openshift-v4/x86_64/
clients/ocp/4.10.34/openshift-client-linux.tar.gz
# tar -xzf openshift-client-linux.tar.gz
# mv oc kubectl /usr/local/bin
# rm -f openshift-client-linux.tar.gz
# rm -f README.md
# oc version
Client Version: 4.10.34
Kubernetes Version: v1.23.5+8471591
```

6. Create a user that is named cp4d and change to it, as shown in Example 6-22.

Example 6-22 Creating a user

```
# useradd cp4d
# su - cp4d
```

7. Install **cpd-cli** as user cp4d, as shown in Example 6-23.

Example 6-23 Installing cpd-cli

```
$ wget
https://github.com/IBM/cpd-cli/releases/download/v11.3.0/cpd-cli-linux-EE-11.3.0.t
gz
$ tar -xzvf cpd-cli-linux-EE-11.3.0.tgz
```

8. Add the two lines that are shown in Example 6-24 to your `~/.bash_profile` file and source the file.

Example 6-24 Sourcing cpd-cli

```
PATH=$PATH:~/cpd-cli-linux-EE-11.3.0-52
export PATH
$ source ~/.bash_profile
$ cpd-cli version
cpd-cli
Version: 11.0
Build Date: 2022-09-30T15:20:03
Build Number: 52
CPD Release Version: 4.5.3
```

Collecting required information

To collect the required information to install IBM Cloud Pak for Data and the Db2 services, complete the following steps:

1. Obtain your IBM entitlement API key. To do so, log in with your IBMid at <https://myibm.ibm.com/products-services/containerlibrary>. Click the **Get entitlement key** tab, and then click **Copy key** to copy the entitlement key to the clipboard. Save the API key in a text file.
2. Set up a `cpd_vars.sh` file with the installation environment variables and source the file. Adapt the values for `OCF_URL`, `OCF_PASSWORD`, `OCF_TOKEN`, and `IBM_ENTITLEMENT_KEY` to match with your environment, as shown in Example 6-25.

Example 6-25 Sourcing the cpd-vars file

```
$ cat cpd_vars.sh
export OCF_URL="api.cp4d-1.rtp.raleigh.ibm.com:6443"
export OPENSIFT_TYPE="self-managed"
export OCF_USERNAME="kubeadmin"
export OCF_PASSWORD="3rgHV-9XtKz-383hW-r9bgt"
export OCF_TOKEN="sha256~3_klXKIIdWvkoNQTHMi6axxxxxxxxxxxx"
export PROJECT_CPFS_OPS=ibm-common-services
export PROJECT_CPD_OPS=ibm-common-services
export PROJECT_CATSRC=openshift-marketplace
export PROJECT_CPD_INSTANCE=zen
export STG_CLASS_BLOCK=ibm-spectrum-scale-csi-fileset
export STG_CLASS_FILE=ibm-spectrum-scale-csi-fileset
```

```
export IBM_ENTITLEMENT_KEY=eyJhbGciOiJIUzI1NiJ9.eyJpc3Mixxxxxxx
export VERSION=4.6.0
export COMPONENTS=cpfs,scheduler,cpd_platform,db2oltp,db2wh,dmc
$ source ./cpd_vars.sh
Preparing the Red Hat OpenShift cluster
Open a new screen session. If you happen to lose the ssh session, you can
reconnect
later via screen -r command.
$ screen
```

3. Log in to the Red Hat OpenShift cluster by running the **manage login-to-ocp** command, as shown in Example 6-26.

Example 6-26 Logging in to the Red Hat OpenShift cluster

```
$ cpd-cli manage login-to-ocp --token=${OCP_TOKEN} \
--server=${OCP_URL}
KUBECONFIG is /opt/ansible/.kubeconfig
Logged in to "https://api.cp4d-1.rtp.raleigh.ibm.com:6443" as
"kube:admin" using the token provided.
```

4. Modify the global pull secret by using the command that is shown in Example 6-27.

Example 6-27 Modifying the pull secret

```
$ cpd-cli manage add-icr-cred-to-global-pull-secret \ $
{IBM_ENTITLEMENT_KEY}
Saved credentials for cp.icr.io
secret/pull-secret data updated
```

5. Modify the **crio** settings of the worker nodes, as shown in Example 6-28. Wait until all worker nodes restart.

Example 6-28 Modifying the crio settings

```
$ cpd-cli manage apply-crio --openshift-type=${OPENSIFT_TYPE}
[SUCCESS] 2022-09-08T12:00:19.296528Z The apply-crio command ran successfully.
```

6. Modify the Db2 kubelet settings for the worker nodes, as shown in Example 6-29. Wait until all worker nodes restart.

Example 6-29 Modifying the Db2 kubelet settings

```
$ cpd-cli manage apply-db2-kubelet \
--openshift-type=${OPENSIFT_TYPE}
[SUCCESS] 2022-07-01T00:22:31.576217Z The apply-db2-kubelet
command ran successfully.
Installing the CPD platform and Db2 services
```

7. Apply the olm artifacts, as shown in Example 6-30.

Example 6-30 Applying the olm artifacts

```
$ cpd-cli manage apply-olm --release=${VERSION} \
--components=${COMPONENTS}
[SUCCESS] 2022-07-01T00:32:23.468211Z The apply-olm command ran
successfully.
```

8. Create the custom resources (CRs), as shown in Example 6-31.

Example 6-31 Creating the custom resources

```
$ cpd-cli manage apply-cr \  
  --components=${COMPONENTS} \  
  --release=${VERSION} \  
  --cpd_instance_ns=${PROJECT_CPD_INSTANCE} \  
  --file_storage_class=${STG_CLASS_FILE} \  
  --block_storage_class=${STG_CLASS_BLOCK} \  
  --license_acceptance=true  
[SUCCESS] 2022-07-01T01:30:46.046375Z The apply-cr command ran  
successfully.
```

9. Get the IBM Cloud Pak for Data web GUI URL and the initial admin password by running the command that is shown in Example 6-32.

Example 6-32 Getting the GUI URL and admin password

```
$ cpd-cli manage get-cpd-instance-details \  
  --cpd_instance_ns=${PROJECT_CPD_INSTANCE} \  
  --get_admin_initial_credentials=true  
CPD Url: cpd-zen.apps.cp4d-1.rtp.raleigh.ibm.com  
CPD Username: admin  
CPD Password: j8Ug400eK8vN
```

10. Log in to the IBM Cloud Pak for Data web GUI URL with the user name and password from step 9, as shown in Figure 6-4.

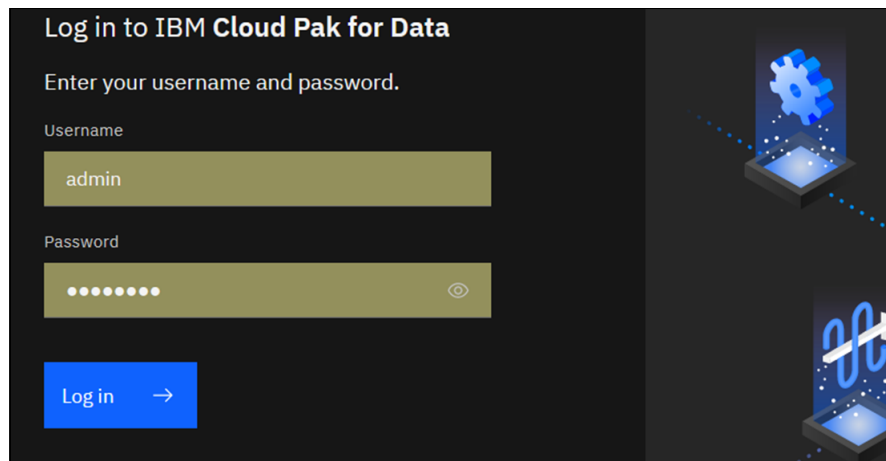


Figure 6-4 IBM Cloud Pak for Data login page

The IBM Cloud Pak for Data home page opens, as shown in Figure 6-5.

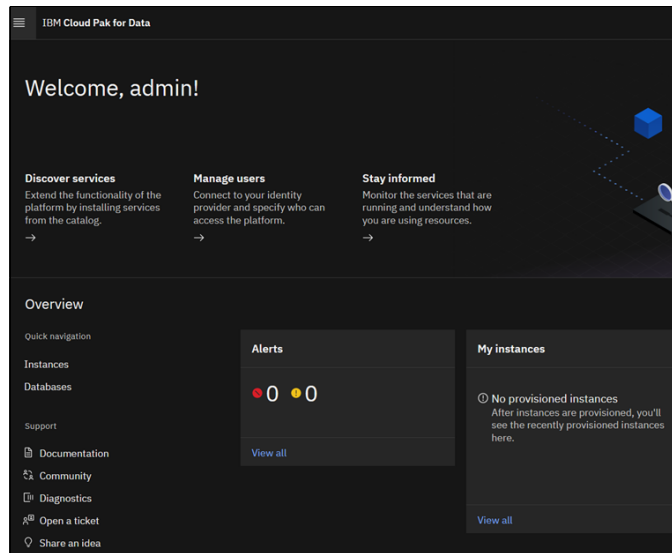


Figure 6-5 IBM Cloud Pak for Data home page

6.3 GitOps for system configuration

GitOps principles help provide continuous deployment, which also can be used to implement infrastructure as code (IaC). For more information, see 4.5.4, “GitOps for Red Hat OpenShift node tuning and configuration” on page 136.

This section shows an example of using GitOps to manage node configuration and performance tuning.

Installing Red Hat OpenShift GitOps

GitOps in Red Hat OpenShift is provided by Red Hat OpenShift GitOps Operator, which is based on ArgoCD.

Install Red Hat OpenShift GitOps Operator from OperatorHub by using the installation method “All namespaces on the cluster (default)”. The operator is installed in to the openshift-operators namespace.

The operator creates an ArgoCD server instance with the name of openshift-gitops in the openshift-gitops namespace.

The default configuration sets up the DEX server to authenticate from Red Hat OpenShift OAuth so that the Red Hat OpenShift users can log in to ArgoCD from the beginning.

Example 6-33 shows a section of the automatically created ArgoCD resource configuration.

Example 6-33 Default role-based access control configuration for openshift-gitops

```
apiVersion: argoproj.io/v1alpha1
kind: ArgoCD
metadata:
  name: openshift-gitops
  namespace: openshift-gitops
...
```

```

    finalizers:
      - argoproj.io/finalizer
spec:
  ...
  grafana:
    enabled: false
    ingress:
      enabled: false
    resources:
      limits:
        cpu: 500m
        memory: 256Mi
      requests:
        cpu: 250m
        memory: 128Mi
    route:
      enabled: false
  ...
  prometheus:
    enabled: false
    ingress:
      enabled: false
    route:
      enabled: false
  ...
  sso:
    dex:
      openShiftOAuth: true
      resources:
        limits:
          cpu: 500m
          memory: 256Mi
        requests:
          cpu: 250m
          memory: 128Mi
      provider: dex
  ...
  rbac:
    policy: |
      g, system:cluster-admins, role:admin
      g, cluster-admins, role:admin
    scopes: '[groups]'

```

Based on this default configuration, the users in Red Hat OpenShift `cluster-admins` group have an admin role in ArgoCD, so, for example, they can create projects and applications.

Creating a GitOps project

To separate different application groups and the IaC configuration in GitOps, create a project with the name of power.

Example 6-34 shows the YAML file for this project.

Example 6-34 AppProject power

```
apiVersion: argoproj.io/v1alpha1
kind: AppProject
metadata:
  creationTimestamp: '2022-10-24T10:16:56Z'
  generation: 4
  name: power
  namespace: openshift-gitops
  resourceVersion: '11234518'
  uid: 879f95ce-318e-4258-a0bd-a764624d68c7
spec:
  destinations:
    - name: '*'
      namespace: '*'
      server: 'https://kubernetes.default.svc'
  sourceRepos:
    - 'https://github.com/lniesz/power.git'
status: {}
```

This project specifies the Git repository for the YAML files that contain the Red Hat OpenShift related configurations, like MachineConfig, Node, and Tuned. It could delimit the destination clusters and namespaces too, but in this case we specified only the local cluster and allow any target namespace.

Also, the following resource type limitations may be configured per project:

- ▶ Cluster-scoped resource allowlist
- ▶ Cluster-scoped resource deny list
- ▶ Namespace-scoped resource allowlist
- ▶ Namespace-scoped resource deny list

Authorizing GitOps to work with Red Hat OpenShift Node and Tuned resources

In our example, we configure automated Node and Tuned patching and modification, so we must authorize the auto-created Red Hat OpenShift GitOps related service accounts to patch and modify these resources. We create the Red Hat OpenShift roles and rolebindings that are shown in Example 6-35.

Example 6-35 Node authorization

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: patch-node
rules:
  - verbs:
    - patch
    apiGroups:
```



```

- ''
resources:
- nodes

kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: openshift-gitops-argocd-application-controller-patch-node
subjects:
- kind: ServiceAccount
  name: openshift-gitops-argocd-application-controller
  namespace: openshift-gitops
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: patch-node

```

Example 6-36 shows the result of the tuning.

Example 6-36 Tuned authorization

```

kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: manage-tuneds
  namespace: openshift-cluster-node-tuning-operator
rules:
- verbs:
  - get
  - watch
  - list
  - create
  - update
  - patch
  apiGroups:
  - tuned.openshift.io
  resources:
  - tuneds

kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: openshift-gitops-argocd-application-controller-manage-tuneds
  namespace: openshift-cluster-node-tuning-operator
subjects:
- kind: ServiceAccount
  name: openshift-gitops-argocd-application-controller
  namespace: openshift-gitops
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: manage-tuneds

```

Without these role-based access control (RBAC) settings, you get the errors that are shown in Figure 6-6 at application synchronization.

OPERATION Sync

PHASE Failed

MESSAGE one or more objects failed to apply, reason: tuned.tuned.openshift.io is forbidden: User "system:serviceaccount:openshift-gitops:openshift-gitops-argocd-application-controller" cannot create resource "tuned" in API group "tuned.openshift.io" in the namespace "openshift-cluster-node-tuning-operator", nodes "master-0" is forbidden: User "system:serviceaccount:openshift-gitops:openshift-gitops-argocd-application-controller" cannot patch resource "nodes" in API group "" at the cluster scope

STARTED AT a few seconds ago (Fri Oct 21 2022 11:07:18 GMT+0200)

DURATION 00:02 min

FINISHED AT a few seconds ago (Fri Oct 21 2022 11:07:20 GMT+0200)

REVISION 3c6a8fd

INITIATED BY cecuser

RESULT

KIND	NAMESPACE	NAME	STATUS	HOOK	MESSAGE
machine...	power-tuning	99-master-powersmt	♥ Synced		machineconfig.machineconfiguration.openshift.io/99-master-powersmt unchanged
tuned.op...	openshift-cluster-node...	cp4d-wkc-ipc	♥ SyncF...		tuned.tuned.openshift.io is forbidden: User "system:serviceaccount:openshift-gitops:openshift-gitops-argocd-application-controller" cannot create resource "tuned" in API group "tuned.openshift.io" in the namespace "openshift-cluster-node-tuning-operator"
v1/Node	power-tuning	master-0	♥ SyncF...		nodes "master-0" is forbidden: User "system:serviceaccount:openshift-gitops:openshift-gitops-argocd-application-controller" cannot patch resource "nodes" in API group "" at the cluster scope

Figure 6-6 Insufficient authorization setting results

Creating a GitHub repository for the GitOps application

Create a repository and a folder for the Red Hat OpenShift configuration YAML files, as show in Figure 6-7 on page 195. An example GitHub repository was set up and contains the appropriate YAML files. You can find the GitHub repository at [GitHub](#).

The content of the repository, including the `99-master-powersmt.yaml` and `cp4dworkertuned.yaml` files, is based on the examples in Appendix A, “Configuring Red Hat CoreOS”, of *Red Hat OpenShift V4.3 on IBM Power Systems Reference Guide*, REDP-5599.

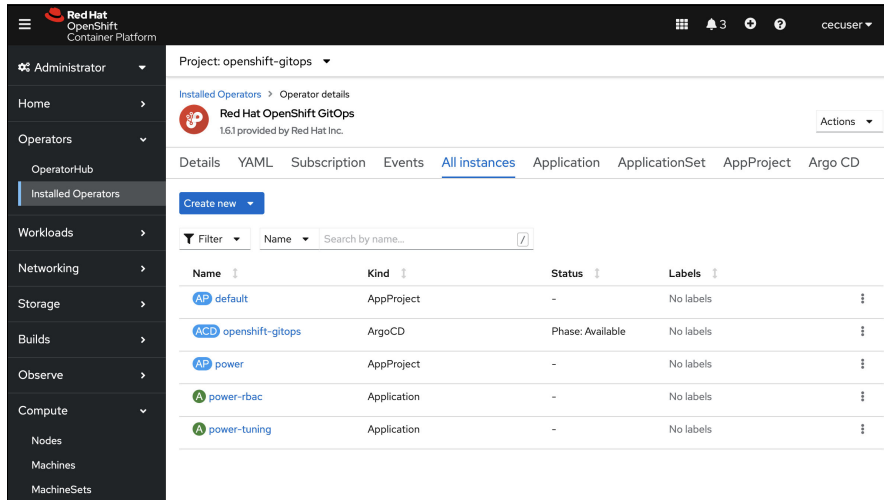


Figure 6-7 Git repository for power-tuning application

The node definition YAML file should be based on the actual nodes in your cluster and have the simultaneous multi-threading (SMT) label, as shown in Example 6-37.

Example 6-37 Node YAML file

```

kind: Node
apiVersion: v1
metadata:
  name: master-0.example.com
  labels:
    beta.kubernetes.io/arch: ppc64le
    beta.kubernetes.io/os: linux
    kubernetes.io/arch: ppc64le
    kubernetes.io/hostname: master-0.example.com
    kubernetes.io/os: linux
    node-role.kubernetes.io/master: ''
    node-role.kubernetes.io/worker: ''
    node.openshift.io/os_id: rhcos
    SMT: '8'
  annotations:
    machineconfiguration.openshift.io/controlPlaneTopology: SingleReplica
    volumes.kubernetes.io/controller-managed-attach-detach: 'true'
spec: {}

```

In this case, the SMT setting is set to 8 because this Red Hat OpenShift node is on an IBM Power10 processor-based server LPAR.

Creating a power-tuning GitOps application

After the preparations, you can create an application that does the following tasks:

- ▶ Monitor and synchronize a MachineConfig definition, which sets the SMT configuration of a node that is based on the Node label: SMT. This labeling also can be done by using GitOps, and the label is based on a Node definition in the same repository.
- ▶ Monitor and synchronize Tuned definition to set the kernel arguments of the selected nodes.

Example 6-38 shows the GitOps application definition, which can be saved and then applied by running the `oc apply -f "filename"` command.

Example 6-38 GitOps application power-tuning

```
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: power-tuning
  finalizers: []
spec:
  destination:
    name: ''
    namespace: power-tuning
    server: 'https://kubernetes.default.svc'
  source:
    path: tuning
    repoURL: 'https://github.com/lniesz/power'
    targetRevision: HEAD
  project: power
  syncPolicy:
    syncOptions:
      - CreateNamespace=true
```

After this application is defined, you can see the application state in the ArgoCD GUI. A Red Hat OpenShift route is auto-created when Red Hat OpenShift GitOps is installed.

The following screen captures show the different views of the application.

Figure 6-8 shows the application overview.

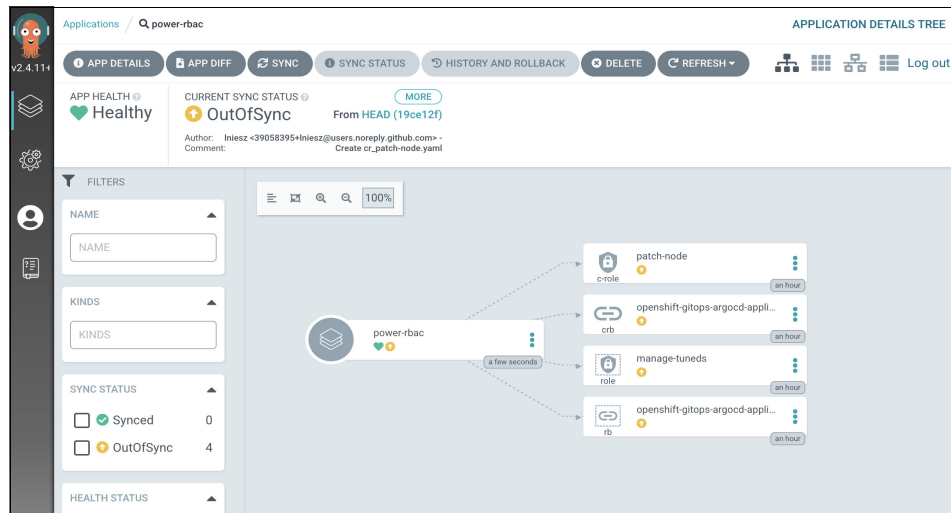


Figure 6-8 Application overview

Figure 6-9 shows the detailed view of the application.

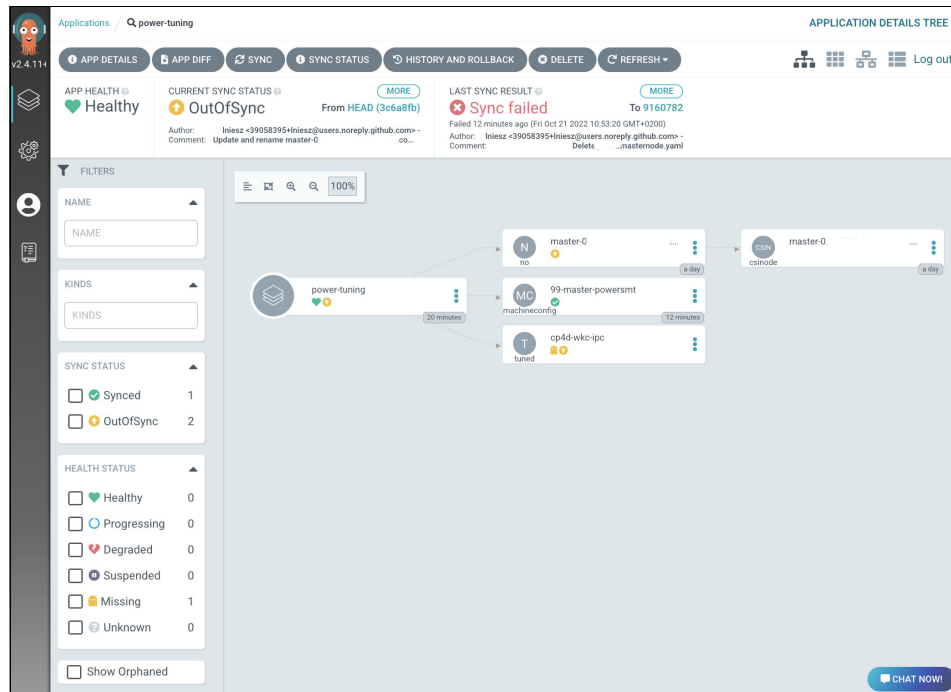


Figure 6-9 Detailed view of application

Figure 6-10 shows the dialog window after you click **SYNC**. Here, you choose which resources to synchronize with special cases for pruning, replacing resources, and the optional namespace creation, if you have namespace-scoped resources in the application.

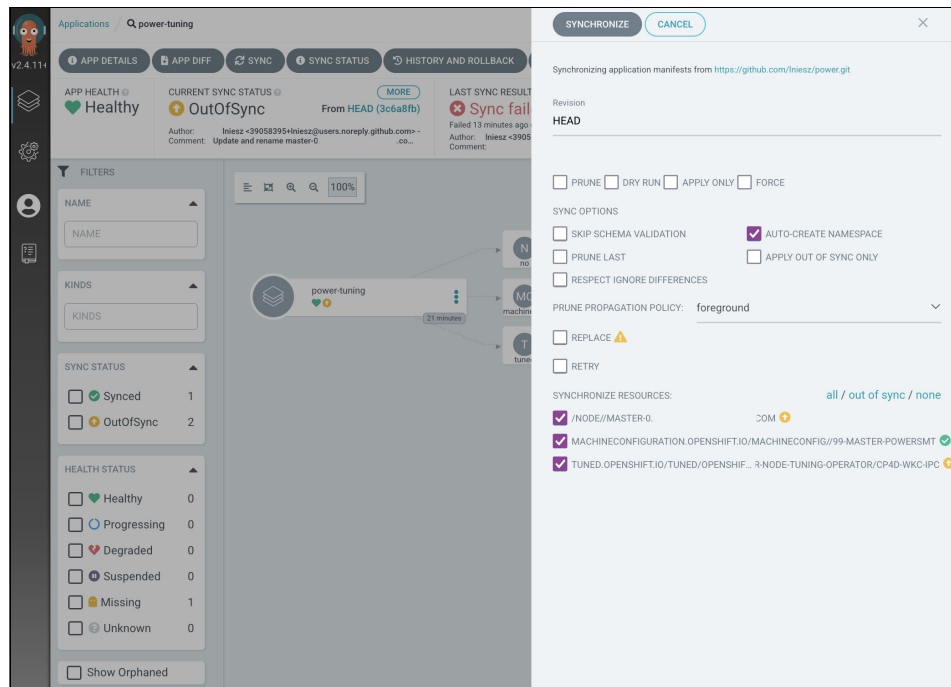


Figure 6-10 Choosing options after clicking SYNC

Synchronizing the power-tuning application

After a successful synchronization, all resources are available, as shown in Figure 6-11.

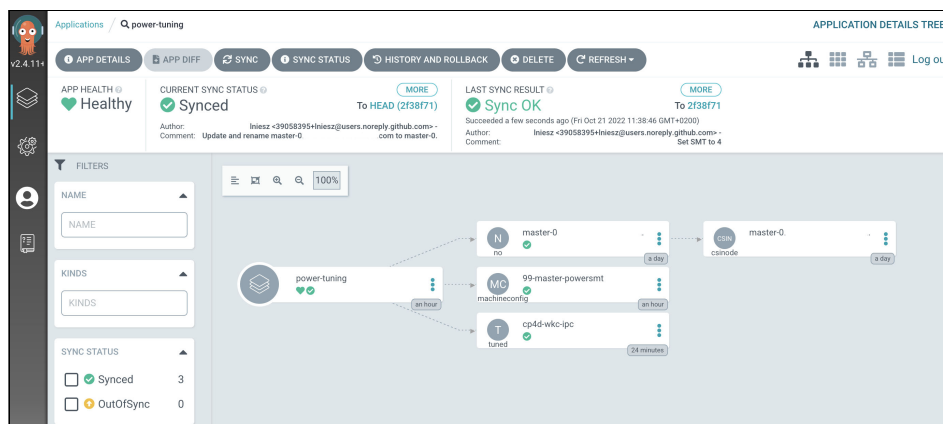


Figure 6-11 Successfully synced application

After successful synchronization, you can check what Red Hat OpenShift GitOps related resource instances are shown in the Red Hat OpenShift GUI. Click **Installed Operators**, and in the `openshift-gitops` namespace, click **Red Hat OpenShift GitOps**. In the All instances pane, you can see your instances, as shown in Figure 6-12.

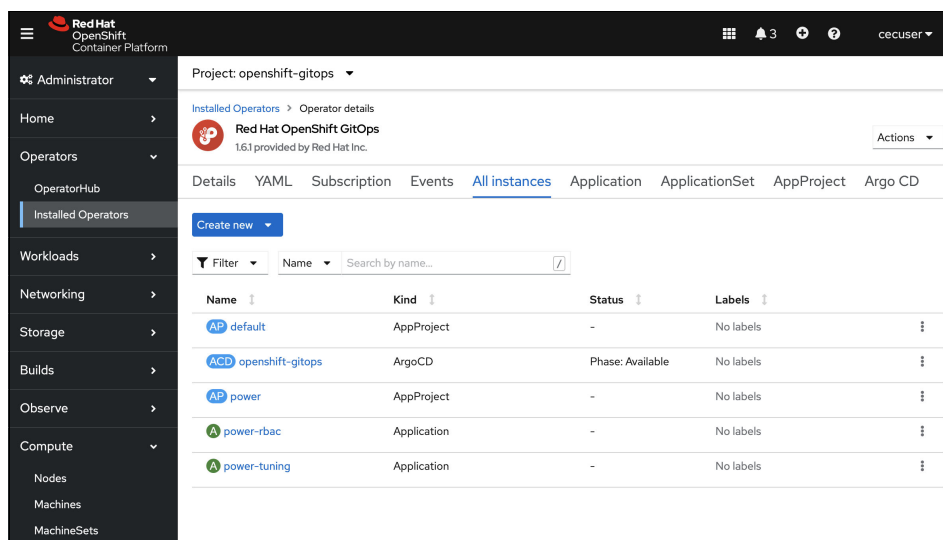


Figure 6-12 Successfully synced application resources

Checking the configuration on the Red Hat OpenShift node

The MachineConfig operator creates a service with the name `powersmt`. Example 6-39 shows the debug window for the node, where you can check the new service.

Example 6-39 Checking the powersmt service on the Red Hat OpenShift node

```
(base) $ oc get node
NAME                                STATUS    ROLES    AGE    VERSION
master-0.example.com                Ready    master,worker    23h    v1.23.5+8471591

(base) $ oc debug node/master-0.example.com
Starting pod/master-0examplecom-debug ...
To use host binary files, run `chroot /host`
```

Pod IP: 172.20.11.150
If you don't see a command line, try pressing enter.

```
sh-4.4# systemctl list-units
Running in chroot, ignoring request: list-units
```

```
sh-4.4# chroot /host
```

```
sh-4.4# systemctl status powersmt
? powersmt.service - POWERSMT
   Loaded: loaded (/etc/systemd/system/powersmt.service; enabled; vendor preset:
disabled)
   Active: active (running) since Fri 2022-10-21 08:56:27 UTC; 24min ago
   Main PID: 2646 (powersmt)
     Tasks: 2 (limit: 417034)
    Memory: 70.7M
       CPU: 40.901s
   CGroup: /system.slice/powersmt.service
           ?? 2646 /bin/bash /usr/local/bin/powersmt
           ??76427 /bin/sleep 30
```

```
...
Oct 21 09:20:28 master-0.example.com powersmt[2646]: /bin/grep:
ion.openshift.io/reason: No such file or directory
Oct 21 09:20:59 master-0.example.com powersmt[2646]: /bin/grep:
ion.openshift.io/reason: No such file or directory
```

```
sh-4.4# cat /etc/systemd/system/powersmt.service
[Unit]
Description=POWERSMT
After=network-online.target
[Service]
ExecStart="/usr/local/bin/powersmt"
[Install]
WantedBy=multi-user.target
```

```
sh-4.4# ps -ef|grep powersmt
root      2646      1  0 08:56 ?          00:00:00 /bin/bash
/usr/local/bin/powersmt
root      77953    76515  0 09:21 pts/0    00:00:00 grep powersmt
```

The initial SMT setting on an IBM Power10 processor-based server node is 8, which means that each core has eight hardware threads. This setting can be checked on the node, as shown in Example 6-40.

Example 6-40 Checking the SMT setting on a node

```
sh-4.4# ppc64_cpu --smt
SMT=8
```

```
sh-4.4# lscpu
Architecture:      ppc64le
Byte Order:        Little Endian
CPUs:              16
Online CPUs list:  0-15
Threads per core: 8
```

```

Cores per socket: 2
Sockets:          1
NUMA nodes:      1
Model:           2.2 (pvr 004e 0202)
Model name:      POWER9 (architected), altivec supported
Hypervisor vendor: pHyp
Virtualization type: para
L1d cache:       32K
L1i cache:       32K
NUMA node0 CPU(s): 0-15
Physical sockets: 2
Physical chips:  1
Physical cores/chip: 10

```

Changing the node label in Git and synchronizing the power-tuning application

To check that the example IaC implementation through GitOps works, change the node YAML file in the Git repository that is found at [GitHub](#). Then, commit the change. In this example, we change only the SMT label in the YAML file to 4.

The commit message is set to “Set SMT to 4”. This change also appears in the ArgoCD GUI, which shows that the app in the running Red Hat OpenShift cluster is not synchronized with the Git repository, as shown in Figure 6-13.

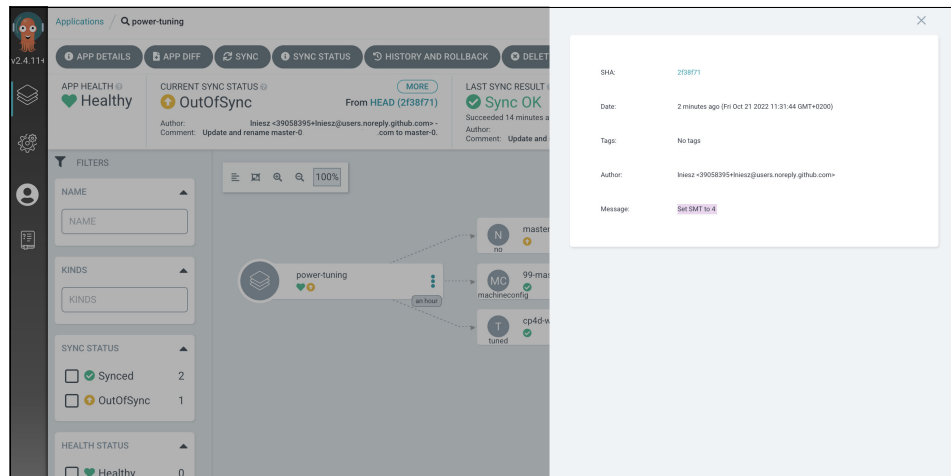


Figure 6-13 Application is out of sync after a change in Git

After you click **SYNC**, ArgoCD applies the configuration, which in this case sets the node label SMT to 4, and changes the running powersmt service on the node to the real SMT setting on the node to 4.

You can check that the application is available in ArgoCD, and that the SMT setting changed on the node, as shown in Example 6-41.

Example 6-41 SMT is set to 4 on the node

```

sh-4.4# ppc64_cpu --smt
SMT=4
sh-4.4# lscpu
Architecture:      ppc64le

```



```
Byte Order:          Little Endian
CPU(s):             16
Online CPUs list:   0-3,8-11
Offline CPUs list: 4-7,12-15
Threads per core: 4
Cores per socket:   2
Sockets:            1
NUMA nodes:         1
Model:              2.2 (pvr 004e 0202)
Model name:         POWER9 (architected), altivec supported
Hypervisor vendor: pHyp
Virtualization type: para
L1d cache:          32K
L1i cache:          32K
NUMA node0 CPU(s): 0-3,8-11
Physical sockets:   2
Physical chips:     1
Physical cores/chip: 10
```

Additional possibilities

This use case is the basis for the following development ideas:

- ▶ Because security is more important, check and set up fine-grained RBAC rules for applications, users, and managed Red Hat OpenShift resource types.
- ▶ Limit target namespaces and resources in the GitOps project.
- ▶ Review the **powersmt** service when there are developments in CoreOS in the handling of IBM Power processor-based server-based commands. In this publication, we review the service by running **ppc64_cpu**, which can be incorporated into the **powersmt** script, as suggested by *Red Hat OpenShift V4.3 on IBM Power Systems Reference Guide*, REDP-5599.
- ▶ Create a separate application for the Red Hat OpenShift role and rolebinding configurations.
- ▶ Automate the creation of the Node YAML files based on the actual installation.
- ▶ You can create an application from other applications in GitOps to combine related application groups.

GitOps and ArgoCD work with HELM and Kustomize, so you can create the YAML files in the Git repositories that are based on these tools.

Abbreviations and acronyms

AD	Application Designer	ELB	Enterprise Load Balancer
ADS	Automation Decision Services	ER	Enterprise Records
AFM	active file management	ERAT	effective-to-real address translation
AI	artificial intelligence	ESS	Elastic Storage System
API	application programming interface	FC	Fibre Channel
APM	Application and Performance Monitoring	FHE	Fully Homomorphic Encryption
ASMI	Advanced System Management Interface	FIFO	first in - first out
AZ	availability zone	FNCM	IBM FileNet Content Manager
BAI	Business Automation Insights	Geneve	Generic Network Virtualization Encapsulation
BAS	Business Automation Studio	GLVM	Geographic Logical Volume Manager
BAW	Business Automation Workflow	GRS	Global Replication Service
BRMS	Backup, Recovery, and Media Services	GTM	Global Traffic Manager
CI/CD	continuous integration and continuous delivery	GTPs	giga transfers per second
CLI	command-line interface	HA	high availability or highly available
CMOS	complementary metal-oxide-semiconductor	HADR	high availability and disaster recovery
CNCF	Cloud Native Computing Foundation	HMC	Hardware Management Console
CNI	Container Network Interface	HPC	high-performance computing
CoD	Capacity on Demand	laaS	infrastructure as a service
COP	Conference of the Parties	laC	infrastructure as code
CR	custom resource	IAM	identity and access management
CRD	Custom Resource Definition	IBM	International Business Machines Corporation
CSI	Container Storage Interface	IBM CP4BA	IBM Cloud Pak for Business Automation
CSV	ClusterServiceVersion or cluster service version	IBM CP4I	IBM Cloud Pak For Integration
CUoD	Capacity Upgrade on Demand	IBM PowerVS	IBM Power Systems Virtual Server
DDIMM	differential DIMM	IDE	integrated development environment
DDR4	Double Data Rate 4	IOPS	I/O operations per second
DEXCR	Dynamic Execution Control Register	IPC	interprocess communication
DFP	decimal floating-point	IPI	Installer-provisioned Infrastructure
DMA	direct memory access	IPS	idle power saver mode
DME	dense math engine	ISA	Instruction Set Architecture
DPM	dynamic performance mode	ISV	independent software vendor
DR	disaster recovery	IVE	Integrated Virtual Ethernet Adapter
EA	effective address	LPAR	logical partition
eBMC	Enterprise BMC managed	LPM	Live Partition Mobility
		LTM	Local Traffic Manager
		LUW	Linux, UNIX, or Windows

MCC	Machine Config Controller	SaaS	software as a service
MCU	memory controller unit	SAN	storage area network
MMA	Matrix Math Accelerator	SCC	security context constraint
MPM	maximum performance mode	SCM	single-chip module
MPP	massively parallel processing	SDN	software-defined networking
MTRF	MPM typical frequency range	SEA	Shared Ethernet Adapter
MTU	maximum transmission unit	SHA	Secure Hash Algorithm
NFS	Network File System	SIMD	single instructions multiple data
NIC	network interface card or network interface controller	SIU	SMP interconnect unit
NUCA	non-uniform cache access	SLA	service-level agreement
NX	nest accelerator	SLI	service-level indicator
OCI	Open Container Initiative	SLO	service-level objective
ODM	Operational Decision Manager	SMP	symmetric multiprocessing
OLAP	online analytical processing	SMT	simultaneous multi-threading
OLM	Operator Lifecycle Manager	SMT8	8-way simultaneous multithreading
OLTP	online transactional processing	SRE	Site Reliability Engineering or Site Reliability Engineer
OMI	Open Memory Interface	ST	single-threaded
Ops	operations	TCO	total cost of ownership
OVS	Open vSwitch	TLB	translation lookaside buffer
PaaS	platform as a service	TPM	Trust Platform Module
PCM	Performance and Capacity Monitoring	TPS	transactions per second
PEP	IBM Power Enterprise Pools	UAT	user acceptance testing
PowerAXON	Power A-bus/X-bus/OpenCAPI/Networking	UDP	user datagram protocol
PV	persistent volume	UI	user interface
PVC	persistent volume claim	UPI	user-provisioned infrastructure
QA	quality assurance	USE	Utilization, Saturation, and Errors
QoS	quality of service	VF	virtual function
QP	quad-precision floating-point	VIOS	Virtual I/O Server
RBAC	role-based access control	VM	virtual machine
RDBMS	relational database management system	vNIC	virtual Network Interface Controller
RDMA	remote direct memory access	VPC	Virtual Private Cloud
RED	Rate, Errors, and Duration	VRA	Virtual Router Appliance
RHCOS	Red Hat Enterprise Linux CoreOS	VSU	vector scalar unit
ROHA	Resource Optimized High Availability	VSX	Vector Scalar eXtension
ROP	return-oriented programming	VXLAN	virtual extensible local area network
RPO	recovery point objective	zCX	IBM z/OS Container Extensions
RTO	recovery time objective		
RWO	Read Write Once		
RWX	Read Write Many		
S2I	source-to-image		

Related publications

The publications that are listed in this section are considered suitable for a more detailed description of the topics that are covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide more information about the topics in this document. Some publications that are referenced in this list might be available in softcopy only.

- ▶ *IBM Cloud Pak for Data Version 4.5: A practical, hands-on guide with best practices, examples, use cases, and walk-throughs*, SG24-8522
- ▶ *IBM Power Systems SR-IOV: Technical Overview and Introduction*, REDP-5065
- ▶ *Matrix-Multiply Assist Best Practices Guide*, REDP-5612
- ▶ *Red Hat OpenShift V4.X and IBM Cloud Pak on IBM Power Systems Volume 2*, SG24-8486
- ▶ *Red Hat OpenShift V4.3 on IBM Power Systems Reference Guide*, REDP-5599

You can search for, view, download, or order these documents and other Redbooks, Redpapers, web docs, drafts, and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- ▶ How to Create Automated etcd Backups in Red Hat OpenShift 4.x:
<https://cloud.redhat.com/blog/ocp-disaster-recovery-part-1-how-to-create-automated-etcd-backup-in-openshift-4.x>
- ▶ IBM Documentation: `nxstat` command:
<https://www.ibm.com/docs/en/aix/7.2?topic=n-nxstat-command>
- ▶ OpenCapi website:
<https://opencapi.org/>
- ▶ *Performance improvement in OpenSSH with on-chip data compression accelerator in IBM POWER9*:
<https://community.ibm.com/community/user/power/blogs/swetha-narayana/2021/07/27/performance-improvement-in-openssh-with-on-chip-da>
- ▶ IBM Power10 processor FAQ
<https://community.ibm.com/community/user/power/viewdocument/sr-iov-vnic-and-hnv-information?CommunityKey=71e6bb8a-5b34-44da-be8b-277834a183b0&tab=librarydocuments>

- ▶ IBM POWER9 gzip Data Acceleration with IBM AIX:
<https://community.ibm.com/community/user/power/blogs/brian-veale1/2020/11/09/power9-gzip-data-acceleration-with-ibm-aix>
- ▶ Using the POWER9 NX (gzip) accelerator in AIX:
<https://www.ibm.com/support/pages/using-power9%E2%84%A2-nx-gzip-accelerator-aix>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

IBM Technology Lifecycle Services

ibm.com/services/technology-support



SG24-8537-00

ISBN 0738461121

Printed in U.S.A.

Get connected

