Draft Document for Review February 23, 2023 11:05 am SG24-8537-00

Redbooks
ibm.com/redbooks

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

Dino Quintero
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
Tim Simon

Cloud

Power Systems

IBM

Redbooks

IBM Redbooks

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

February 2023

Note: Before using this information and the product it supports, read the information in “Notices” on page 13.

First Edition (February 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 v4.8.23,
- Red Hat OpenShift Data Foundation (Previously OCS) v4.8
- Red Hat OpenShift Container Platform 4.10
- IBM Cloud Pak for Data Version 4.6
- IBM Cloud Pak for Business Automation Version 22.0.2
- IBM Cloud Pak for Integration Version 2021.4
- IBM Cloud Pak for Integration Version 2022.4
- IBM Cloud Pak for Watson AIOps Version 3.6.1
- IBM Cloud Pak for WebSphere Hybrid Edition version 5.1.0
- IBM Storage Scale (previously Spectrum Scale) 5.1.5
- IBM Instana Observability 1.0.232

This document was created or updated on February 23, 2023.

Contents

Notices

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

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

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

AIX®
Cognos®
DataStage®
DB2®
Db2®
DS8000®
FileNet®
IBM®
IBM Cloud®
IBM Cloud Pak®
IBM Elastic Storage®
IBM Spectrum®
IBM Watson®
IBM Z®
Instana®
POWER®
Power Architecture®
POWER8®
POWER9™
PowerHA®
PowerVM®
Redbooks®
Redbooks (logo) ®
Spectrum Fusion™
System z®
SystemMirror®
Turbonomic®
WebSphere®
z/OS®
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.

Ansible, Ceph, CloudForms, OpenShift, Red Hat, are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries 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 utilizing Hybrid Cloud solutions to meet new business requirements can be a challenge. This Redbooks publication is designed to show you how to implement a Hybrid Cloud solution utilizing 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, scalable Cloud environment. We provide hints and tips on how to install your Red Hat OpenShift cluster and also provide guidance on how to size and ultimately tune your environment to meet your end user's expectations.

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 (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.

Tushar Agrawal is a Customer Success Leader in the United States. He has 20 years of experience in enterprise architecture and solution design in eCommerce and Supply Chain for large-scale data processing and high availability using microservices and event-driven architecture. Tushar has successfully launched 50+ complex solutions for customers across industry verticals and segments. Tushar passionately leads a team of architects to drive the adoption of IBM's Hybrid Cloud vision using Red Hat OpenShift and develop reusable assets to reduce time to market. Tushar has filed 100+ patent applications with USPTO and continuously works on innovation with emerging technologies to co-create intellectual property for IBM. Tushar is originally from India. Tushar holds an MBA from North Dakota State University, Fargo, ND, and a BS 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, presales solution architect at AWS and as an SRE and onboarding ISVs for Google marketplace at a partner. He brings multifaceted experience to the table, a continuous learner, and a strong supporter of STEM. In his free time, he coaches K-12 students for FIRST LEGO league competitions and inspiring the young minds to take up STEM careers.

Shahid Ali is a Cloud Solution Lead for MEA Region. At the time of this publication, he is based in Riyadh, Saudi Arabia, and leading hybrid multi-cloud solutions in MEA region. Shahid is an experienced Enterprise Architect and joined IBM around 5 years back as an Enterprise Architect. He has 28 years of experience as an architect and consultant. Before joining IBM, he has provided consultancy services in some of the largest projects in Saudi

Arabia in 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 courses platforms. Shahid has several IBM and industry certifications and is also a member of IBM Academy of Technology.

Daniel Casali is a Thought Leader Information Technology Specialist working for 15 years at IBM with Power Systems, High Performance Computing, Big Data, and Storage. His role at IBM is to bring to reality solutions that address client's needs by exploring new technologies and for different workloads. He is also fascinated by real multicloud implementations, and always trying to abstract and simplify the new challenges of the heterogeneous architectures that are intrinsic to this new consumption model, be it 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 engineering, design, and architecture of PaaS and cloud infrastructures. At the time of this publication, 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 Systems 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 & AI field with multi-industry knowledge and vast experience in Sales. At the time of this publication, 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. He is also a technology enthusiast, knowledge-sharing obsessed, and is continually learning new technologies such as Quantum computing.

Shrirang Kulkarni is a LinuxONE and Cloud Architect and 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 also worked with various clients in over 25 countries worldwide from IBM Dubai as a Lab services consultant for System Z in the Middle East and Africa. He has achieved "IBM Expert Level IT specialist" and "The Open Group Certified Master IT Specialist" certifications. He has coauthored the Redbook *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 System z®, IBM z/VM®, Cloud Solution, zCX, OpenShift, Architecture Design and Solution for z/VM and Linux on System z, Performance tuning Linux on System z, IBM z/VM, Oracle, System P, and System x.

Nick Lawrence is an IT Management Consultant on the IBM Technology Lifecycle Services team (IBM Power Systems). Nick specializes in emerging technologies, cloud, and AI applications. Prior to joining Technology Services in the spring of 2022, Nick was a software developer responsible for building healthcare solutions 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, OpenShift, Infrastructure as Code and GitOps. He has written extensively on performance monitoring, software defined infrastructure and OpenShift ecosystem sections of this publication.

Gabriel Padilla is the Linux Test Architect for IBM Power Systems focused on Hardware Assurance. He has a Bachelor's Degree as an Electronic Engineer and also a Master's Degree in Information Technology. He has been in IBM more than 10 years and his experience ranges from Test Development (Design) to Supply Chain process. Gabriel is considered as a technical leader for IBM Systems on subject matters including Linux, Red Hat OpenShift and Cloud.

Gustavo Santos is an IBM Brand Technical Specialist and Power Systems Consultant. He has been with IBM since 1997. He has 25 years of experience in IBM Power Systems, Cognitive Solutions and Hybrid Cloud Architecture. He holds a degree in Systems Engineering from Universidad Abierta Interamericana. During the last 7 years he was working as a Power Systems Consultant and during the last year he was working as Brand Technical Specialist, to create new solutions for the clients and add value to the IBM Solutions portfolio. This is his eighth residency for a Redbook project and he has written extensively on IBM Power infrastructure.

Shiv Tiwari is a seasoned Information Technology professional with an impressive 18 years of technical experience. As a Data & 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 track record of success working with leading enterprises across various industries, including banking, telecommunications, insurance, and retail. He has demonstrated his knowledge and expertise through his co-authorship of the Redbook IBM AIX® and Enterprise Cloud Solutions. His areas of expertise include Data Engineering, Data Science, AI and Machine Learning, Data Governance, Data Quality, and Data Observability. Aside from his professional achievements, He is a passionate technology enthusiast who is always striving to stay up-to-date with the latest advancements in the field.

Sundaragopal Venkatraman (Sundar) is a CTO - Center of Excellence, IBM Expert Labs. He has diversified skills on Platform, IBM Cloud® Paks, Migration & Modernization. A trusted advisor to customers on enterprise Modernization and Automation. 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 has delivered key notes on WW conferences on Technology Transformation & Modernization. He is a co-chair for the IT specialist board in Asia-Pacific.

Tim Simon is a Redbooks Project Leader in Tulsa, Oklahoma, USA. 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 using IBM Power, IBM Storage, and IBM System z throughout his career.

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 (CP4MCM, CP4WAIOPS, IBM Monitoring, APM/Instana)

Attila Grósz, Storage Systems Technical Sales,
IBM Sales, Hungary

Ashwin Srinivas, Senior AI & 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

1

Introduction

This chapter provides an introduction to the concepts and considerations needed to implement your workloads in a cloud environment using Red Hat OpenShift on IBM Power System servers - either on-premises or in the cloud. The following topics are addressed:

1.1 Adapting to a new infrastructure paradigm

It seems to be a business imperative for enterprises to embrace cloud resources and infrastructure. In order to compete in the modern world, an enterprise needs to adapt. The journey to cloud is undertaken because business's see many benefits from cloud computing infrastructures and platforms such as those 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 mix multiple clouds in a hybrid cloud environment is being done because business users are demanding a more adaptive IT infrastructure which can support the new business requirements of agile programming and flexible infrastructure that can allow the business to quickly take advantage of new business opportunities that make better use of the enterprise's assets. Enterprises are looking to cloud for the following benefits.

Flexibility

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

Scalability: Cloud infrastructure scales on demand to support fluctuating workloads.

Storage options: Users can choose public, private, or hybrid storage offerings, depending on security needs and other considerations.

Control choices: Organizations can determine their level of control with as-a-service options. These 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, encryption, and 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:

Accessibility: Cloud-based applications and data are accessible from virtually any internet-connected device.

Speed to market: Developing in the cloud enables users to 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, saving organizations the cost of servers and other equipment.

Pay structure: A "utility" pay structure means users only pay for the resources they use.

Strategic value

Cloud services give enterprises a competitive advantage by providing the most innovative technology available. This provides benefits through:

Streamlined work: Cloud service providers (CSPs) manage underlying infrastructure, enabling organizations to 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 teams can collaborate from widespread locations.

Competitive edge: Organizations can move more nimbly than competitors who must devote IT resources to managing infrastructure.

As you can see, the benefits of cloud can be significant, and when properly utilized can provide a business advantage in today's extremely competitive world. However, moving to a cloud environment requires a new way of thinking in terms of how to get the most our of your infrastructure in the new cloud world.

1.1.2 New performance paradigm

Enterprises have grown quite adapt 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 all interact.

When moving to a cloud infrastructure – be it private, public or hybrid – the tried and true processes and techniques are most likely not going to be 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 quickly start, scale, stop, and move based on the current workloads. This provides challenges in many ways and new tools and techniques need to be developed to monitor and manage your cloud environment to provide the appropriate end user experience.

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

1.2 Red Hat OpenShift on IBM Power Systems

There are many options available for running your cloud workloads, both in the world of private clouds and in public clouds. This book is addressing the use of Red Hat OpenShift on IBM Power System servers either in your enterprise or in a public cloud environment such as what is provided by the IBM Power Virtual Server offering - an Infrastructure as a Service offering running on IBM Power Servers in IBM data centers.

This section provides an overview of Red Hat OpenShift – an industry leading cloud offering – running on IBM Power Systems.

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 is built on top of Kubernetes. It takes care of integrated scaling, monitoring, logging, and metering functions. Red Hat OpenShift includes everything you need for hybrid cloud, like a container runtime, networking, monitoring, container registry, authentication, and authorization.

The next section provides a high level overview of Red Hat OpenShift. This is intended to be an introduction - more detail on Red Hat OpenShift is provided in Chapter 4, "Red Hat OpenShift architecture and design" on page 107.

Red Hat OpenShift architecture and components

To make the most of Red Hat OpenShift, it helps to understand its architecture. Figure 1-1 provides an overview of Red Hat OpenShift.

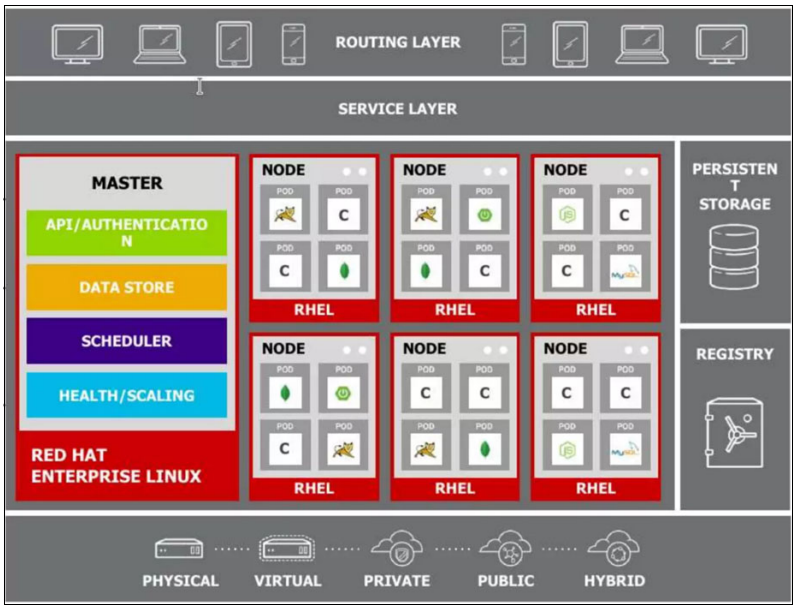


Figure 1-1 Red Hat OpenShift architecture

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

Infrastructure layer

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

This book is focused on utilizing IBM Power System Servers with built in virtualization functionality that can allow you to run your traditional workloads and your new cloud workloads on the same physical servers in different logical partitions (LPARs).

Using your existing IBM Power Systems infrastructure can allow you to efficiently integrate your new cloud applications with your traditional applications and allow you to continue to get business value from the current applications while quickly building new workflows to take advantage of 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 – they are discussed more in “Containers” on page 23.

The service layer provides a permanent IP address and host name to the PODs; connects applications together, and allows simple internal load balancing by distributing tasks across application components.

There are mainly 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. It is recommended that you have multiple control nodes for availability. You can also have multiple worker nodes in the cluster; adding worker nodes allows you to scale the environment or even provide isolation for sensitive workloads. The worker nodes are where all your coding adventures happen.

Control nodes

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

1. API and authentication: Any administration request goes through the API; these requests are SSL-encrypted and authenticated to ensure the security of the cluster.
2. Data Store: Stores the state and information related to environment and application.
3. Scheduler: Determines POD placements while considering current memory, CPU, and other environment utilization.
4. Health/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 is not restarted for a temporary time.

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 is to process data 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 runtimes and libraries required to run your software services.

Keep in mind that containers are ephemeral, so saving data in a container risks the loss of data. To prevent that, you need to use persistent storage to save data for applications and databases.

All containers in one POD share the same IP Address and same volume. In the same POD, you can also have a sidecar container, which can be a service mesh or for security analysis – it must be defined in the same POD sharing the same resources as other containers. Applications can be scaled horizontally or vertically by adding additional 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 of your data is saved and connected to containers. It is important to have persistent storage because containers are ephemeral, which means when they are restarted or deleted, any saved data is lost. Therefore, persistent storage prevents any loss of data and allows the use 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 Systems

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

IBM Power Systems are designed from the ground up 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 in mind with industry leading connectivity and scalability to handle large numbers of concurrent users and work with large data sources effectively.

IBM Power Systems provides a flexible platform with cloud like scalability and pricing and are also available as a hybrid cloud offering in the IBM Power Virtual Server offering. An advantage of choosing IBM Power Systems 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 Systems servers can be the right solution for your cloud requirements and this book is designed to help you in designing, implementing and tuning those applications running in your IBM Power Systems Server cloud.

More details on the advantages of using IBM Power Systems Servers in your hybrid cloud are provided in Chapter 3, “IBM Power Systems performance capabilities” on page 57.

1.4 Summary

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

We highly recommend that the platform you choose be an IBM Power System solution running Red Hat OpenShift.

2

Performance and tuning

In this chapter we will discuss some techniques to maximize the performance of your applications in a Red Hat OpenShift cluster. We will explore how to define performance, and some of the tools that can be used to measure the different aspects of performance, in your cluster. We will provide some specific recommendations to help you for 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” on page 26 defines terms that are important to understand when discussing performance.
- “Models” on page 28 describes some different performance theories that can help you as you plan for and monitor the performance of your environment.
- “An example use case scenario” on page 30 is a case study of how a specific customer tuned their Red Hat OpenShift cluster and the results they were able to achieve.
- “Red Hat OpenShift performance baseline” on page 33 provides guidance on some settings that you can use as you set up your cluster for best performance.
- “Red Hat OpenShift starting configuration” on page 40 provides a recommended starting configuration for your cluster. This can then be adjusted and scaled to meet your specific requirements.
- “Tools” on page 41 describes some of the tools that can be used to measure and monitor the performance of your cluster.

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 discuss the components involved.

2.1.1 Performance components

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

Optimized, lightweight images

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

Dependency

An Operator may have a dependency on another Operator being present 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 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 Custom Resource Definition (CRD) API, and is not related to packages or bundles.

Memory utilization

Memory is an important component to allow the system to do work. If there is not enough memory available, the system will have to swap out some sections of memory to load new content. This causes delays in starting new tasks and may create additional 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 which can help identify PODs that exceed memory usage limits and terminate them.
- Monitoring the node level which 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 free space available on the disk can have implications on system performance. Low disk space on the root volume can lead to issues with scheduling PODs. Once 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 in performance of your Red Hat OpenShift cluster. High CPU utilization can cause extended latency to your user as the infrastructure is not able to perform required tasks in a timely manner. Monitoring CPU utilization via Grafana or any

other monitoring tool can help identify if CPU utilization is related to health issues in the cluster.

Performance monitoring plan

Detailed document that describes your indicators, measures, and approach to data collection, acquisition, analysis, use, and reporting.

Results

Changes that happen because of what a project or program does. Includes outcomes and outputs.

2.1.2 Performance tuning terminology

Table 2-1 provides a list of terms that are used when discussing performance tuning. Understanding these terms can help you appropriately 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 a particular element such as a process, at a particular time.
Latency	Delay 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. This 10 seconds is 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 will be 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.
Response time	Amount of time the Container takes to complete an operation. Therefore, it is an aggregate of time incurred by all server processing and transmission latency for an operation. Response time may be as experienced by an application end user, or may be the amount of time needed for some other operation that is unrelated or indirectly related to end user sessions.
Throughput	Typically expressed in transactions per second (TPS), expresses how many operations or transactions can be processed in a set amount of time.

Term	Definition
Thread	An operating system feature for performing a given 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 SLA/SLO/SLI

When planning for performance and availability of a system, it is important to have objectives that are agreed upon prior to implementation that describe the expectations users will have for the availability of the system. SLIs, SLAs, and SLOs represent different concepts describing the promises we make to the users about the availability of your system to ensure that the users expectations are being met. These components address things like:

- 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.

SLA

An SLA (service level agreement) 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 usually legal documents and represent both the expectations and the consequences of failure in meeting those expectations. Consequences could include financial penalties or service credits for example.

SLO

An SLO (service level objective) is a statement about an specific metric within an SLA. This could be related to uptime, response time or other measurable metric important to the user. Where SLAs are only really relevant 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 on what goals they need to set and measure themselves against.

SLI

The SLI (service level indicator) is the specific metric that shows compliance (or non-compliance) for the SLOs that are set up. These need to be specific and measurable indicators. To meet an SLO the SLI for that component needs to equal to or higher than the SLO target.

2.2 Models

Applications running in Red Hat OpenShift, or for that matter any Kubernetes flavored cluster, are distributed systems. When analyzing performance and tuning the workloads in such clusters you need to understand a few theoretical models. In this section we discuss some of these models and provide some examples. These model concepts were used by the authors of this paper 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 was credited for discovering Queuing theory in 1920. Still today, this theory has applications in several fields ranging from designing call centers, optimizing restaurant operations and in understanding the bottlenecks in a distributed system. Whether you are running applications on-premise or in a cloud, our systems, interconnects, and applications are becoming increasingly distributed. With the advent of microservices paradigm, where you may find an average of 100 microservices or more, this complexity further increases.

In the context of performance testing and tuning, large scale distributed systems are very difficult to test to assess capacity as the testing requires a production copy of the infrastructure and simulated load. By understanding queuing theory, you can build a mathematical model of the system and using commonly available tools to find an optimal architecture without huge cost overlay. An excellent talk on this topic was presented at KubeCon 2017.

2.2.2 Little's Law

John Little, an operation research professor at MIT discovered this law, stating that the average number of tasks in a queue can be obtained by a product of arrival rate of tasks and their holding/processing time. This Law is an intuitive way to derive the relationship between Latency and Throughput. Please see IBM WebSphere® Performance Cookbook on how to use Little's law to calculate sizing for a web server.

2.2.3 The 4 Golden Signals

Google distilled their years of experience of running millions of commodity servers into the Server Reliability Engineering practice. Google's methodology proposes monitoring the four Golden Signals, i.e. Latency, Traffic, Errors and Saturation.

Latency is defined as how long a request took to process.

Traffic is how much traffic a given service is receiving. This could be described by the volume of data being processed or the network or disk i/o rates. This includes both the application and the subsystem level.

Errors is how many requests are resulted in errors. This includes both the errors a given service is returning and errors that occur when a request results in an exception that causes application failures.

Saturation is the measure of how busy the servers are in terms of compute, storage and networking. In other words, are the applications starving for resources meaning 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 and these are available, free of charge, to help others implement their best practices.

2.2.4 USE method

Brendan Gregg, a Solaris engineer is a well-known Performance expert for his work in tools such as Dtrace for Solaris and newer eBPF in Linux. He pioneered two performance models. One of those performance models is a resource focused *USE* method. It is important to understand this how 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 for us to determine their **U**tilization, **S**aturation and **E**rrors.

Utilization determines how busy a given resource is for example a CPU could be 100% utilized or disk i/o could be 100% so beyond which the system cannot service new requests anymore.

Saturation denotes how many processes are queued to access the resources. If you recall, earlier we introduced queuing theory and Little's law, they 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 presence of an issue could have a domino affect on applications that rely on a specific resource.

For more information see Brendan Gregg's page on the USE method.

2.2.5 RED method

For the microservices architecture, Weave Works, the people behind the Flux GitOps coined the term RED for measuring microservices performance where:

R is **R**ate: the number of requests being served per second (or a duration) or throughput.

E for **E**rrors i.e. number of failed requests.

D denotes the **D**uration or latency of each request.

The RED method, as its author states, has its origins based on the 4 golden signals. Nevertheless, it provides a different abstraction to help measure and troubleshoot performance issues.

2.3 An example use case scenario

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

Products deployed

- CP4I v2021.4 - App Connect and API connect
- Red Hat OpenShift v4.8.23
- OpenShift Data Foundation (previously named OpenShift Container Storage) v4.8

Compute resources

Table 2-2 on page 31 shows the compute resources and storage assigned to the different components used in our use case.

Table 2-2 Server details with compute & resources

Nodes	Type	Server	Server IP Address	Hostname	CPU	Memory	Disk Space
Boot	VM (temp)		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
Infra	VM	Server IF1	10.198.34.20	infra0.ocpuat.domainname.com	16	32	300 + Additional disk 1TiB (RAW - unformatted) (OCS)
	VM	Server IF2	10.198.34.21	infra1.ocpuat.domainname.com	16	32	300 + Additional disk 1TiB (RAW - unformatted) (OCS)
	VM	Server IF3	10.198.34.22	infra2.ocpuat.domainname.com	16	32	300 + Additional disk 1TiB (RAW - unformatted) (OCS)
HA Proxy	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: 1.5s
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 (ex. Integration server) and changed the max and min CPU values.
- We changed the failover using replicas for the application PODs.
- We used Grafana to monitor the complete metrics in and out of Red Hat OpenShift system.
- We monitored the subsystem using external tool to identify the utilization.
- We made the POD crash scenario and changed the values of threshold.
- We re-sized the utilization cap to 50%.
- We observed the worker plane utilization was less then 40% as per customer requirements.
- We also changed a few application product parameters (primarily Liberty and nginx).

The test case was run then run on a UAT environment where we monitored the parameters below:

- 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 API server (master) and etcd processes were reduced.

Recommendations

After gathering the additional data shown above, we were able to make the following recommendations:

- Looking at Figure 2-1 we see the sizes of I/O operations done by a 3-node cluster of etcd 3.1 (using storage v3 mode and with quorum reads enforced). As there is a large number of small size writes it is recommended that etcd be run on a system with SSD storage to optimize performance.

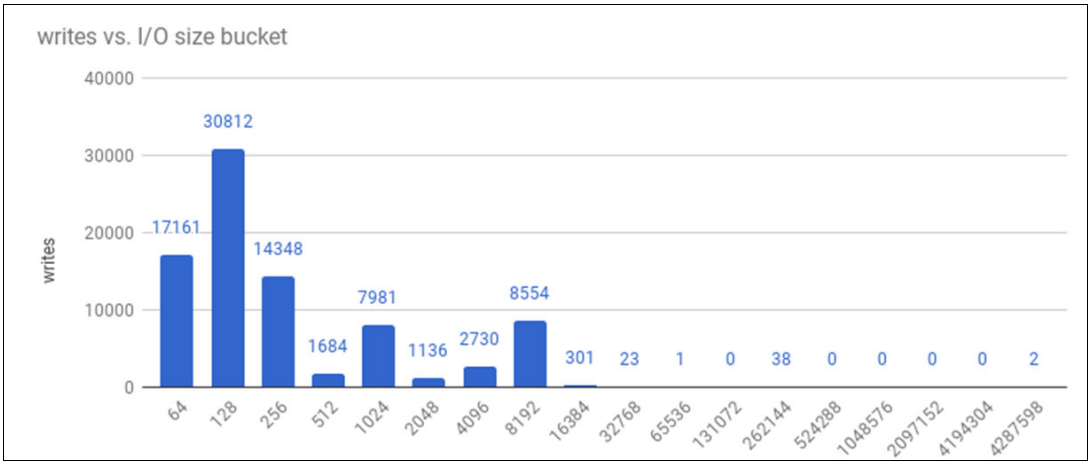


Figure 2-1 etcd write sizes

- As etcd processes are typically memory intensive and Master / API server processes are CPU intensive, this makes them a reasonable fit for co-location on a single machine or virtual machine (VM).
- Optimize communication between etcd and master hosts either by co-locating them on the same host, or providing a dedicated network.

Tip: After the profiling of etcd under Red Hat OpenShift Container Platform as shown in Figure 2-1, it can be seen that etcd frequently performs small reads and writes. Using etcd with storage that handles small read/write operations quickly, such as SSD, is highly recommended.

Additional tuning actions

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

- Changed the parameters of Liberty for heap.
- Changed application logging level required.
- Changed the access controls.
- Changed the CPU size for the PODs.
- Changed the replica count.
- We started with 50 users then the test continue to grow with 500, 1000, 2000, 3800 and finally landed at 4000 users.
- We 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 did we were able to achieve the response time that was required by the customer. The response time includes the time the user experienced starting from the user's computer and includes transitioning through the HAproxy and routers.

2.4 Red Hat OpenShift performance baseline

In this section, we discuss Red Hat OpenShift container platform performance. We provide information on baseline considerations and provide estimates on expected performance.

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

A **node** provides the runtime environments for containers. Each node in a Kubernetes cluster runs the required services to be 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 recommendations.

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

General baseline recommendations

As you consider running your OCP environment consider the following general guidelines:

- Do the proper capacity planning. Identify and document your requirements.
- Install using the install 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.
- Don't change any CoreOS values without a recommended instructions from product service either Red Hat or IBM.
- Consider externalizing logs via external tools.
- Ensure the proper User Governance is set up.
- Plan for your CI/CD deployment pipeline and consider using fully automated environments.

2.4.2 Cluster performance considerations

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

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

Important: The default cluster network address setting, 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 larger node counts beyond 500 nodes.

2.4.3 Recommended node host 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:

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

Modifying the number of unavailable worker nodes

You always have the flexibility to change the type and size of worker nodes. By default, only one machine is allowed to 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 following control plane node size recommendations shown in Table 2-3 are based on the results of control plane density focused testing.

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

Table 2-3 Control plane sizing recommendations[1]

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 usage will spike up when one of the nodes is stopped, rebooted or fails. The failures can be due to unexpected issues with power, network 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 in order to be highly available which leads to increase in the resource usage. This is also expected during upgrades because the masters are cordoned, drained, and rebooted serially to apply the operating system updates, as well as the control plane Operators update.

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

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

2.4.5 Recommended *etcd* 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 up 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 particularly 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 top of a block device that can write at least 50 IOPS of 8000 bytes long sequentially. That is, with a latency of 20ms, keep in mind that uses *fdatasync* 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:

- Run *etcd* on machines that are backed by SSD or NVMe disks with low latency and high throughput.
- Avoid NAS setups and spinning 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 only run infrastructure components such as the default router, the registry and components for monitoring, These are defined as infrastructure nodes and they are not counted toward the number of subscriptions required to run your environment.

Separating these infrastructure functions from your application worker nodes can provide a better environment for your applications as it reduces “overhead” on the application nodes. In a production deployment it is recommended 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 information on infrastructure nodes and which components can run on infrastructure nodes, see the "Red Hat OpenShift control plane and infrastructure nodes" section in the 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, you create and apply a custom config map.

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, as these factors can lead to an increase in the number of metrics or time series in Prometheus. The following infrastructure node size recommendations 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 LAN (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. This allows for all PODs behind a service to communicate with each other, even if they are running on different systems.

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

Routing optimization

Data traffic needs to move in and out of your OpenShift cluster as well as between different PODs and nodes running different services in the cluster. Traffic need to be quickly and efficiently routed in order for the cluster services to be available to your end users.

Scaling Red Hat OpenShift Container Platform HAProxy router

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

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

- HTTP keep-alive/close mode
- route type
- TLS session resumption client support
- number of concurrent connections per target route
- number of target routes
- backend server page size
- underlying infrastructure (network/SDN solution, CPU, and so on)

In general, 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 language used or if using static or dynamic content.

Ingress, or router, sharding should be used to serve more routes towards applications and help horizontally scale the routing tier. Sharding allows you to 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 will serve those routes. Ingress sharding is useful in cases where you want to load balance incoming traffic across multiple Ingress Controllers, when you want to isolate traffic to be routed to a specific Ingress Controller.

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

2.4.9 Storage considerations

Figure 2-2 on page 39 summarizes the recommended and configurable storage technologies for the given Red Hat OpenShift Container Platform cluster application. It provides general guidelines to help determine what storage types are avaiable and what the recommended use case 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

Notes forFigure 2-2

1. ReadWriteOnce.
2. ReadOnlyMany.
3. ReadWriteMany.
4. This does not apply to physical disk, VM physical disk, VMDK, loopback over NFS, AWS EBS, Azure Disk and Cinder (the latter for block).
5. For monitoring components, using file storage with the ReadWriteMany (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's PVs or PVCs. Apps must integrate with the object storage REST API.

2.4.10 Other considerations

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

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 may not have access to the resources it requires when it needs them, which will result in reduced performance.

However, this may be an acceptable trade-off in favor of increased density and reduced costs. For example, development, quality assurance (QA), or test environments may be overcommitted, whereas production might not be.

- Red Hat OpenShift Container Platform implements resource management through the compute resource model and quota system. See the documentation for more information about the Red Hat OpenShift resource model.

For more information and strategies for overcommitting, see “Placing PODs onto overcommited nodes”.

Using a pre-deployed image to improve efficiency

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

Pre-pulling images

To efficiently produce images, you can pre-pull any necessary container images to all node hosts. This means the image does not have to be initially pulled, which saves time and performance over slow connections, especially for images, such as Source-to-image (S2I)[2], metrics, and logging, which 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 in an efficient manner.

2.5 Red Hat OpenShift starting configuration

Table 2-5 is an example hardware requirement 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 as per questionnaire		Local Storage/node	Number of machines
		CPU	RAM		
Bootstrap (Temp)	RHCOS	8	16	120 GB	1
Bastion / Installation (for triggering deployment)	RHEL 7/8	2	8	200 GB	1
Control	RHCOS	4	16	300 GB	3
*Compute	RHCOS	8	32	120 GB	3
Infra	RHCOS	4	16	120 GB	3
**Storage (OCS/ODF)	RHCOS	4	16	120 GB	3

Notes for Table 2-5:

- Compute node resource requirement is based on sizing requirements.
- Infra nodes are based on customer requirements and optional.
- Storage is based on customer requirement with additional ** external storage.

The resources 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 UPI or IPI with the given scenario.

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

2.6 Tools

In this section we show, what tools can be used to monitor performance related characteristics of a system and applications. Later we show IBM Power Systems server and Power Virtual Server specific tools as well.

2.6.1 Observability

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

Observability can discover and collect telemetry data, which can be logs, metrics, traces and dependencies. This data can help Site Reliability Engineers (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 environments. One of the main differentiator 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 “unknown unknowns”.

2.6.2 Instana

IBM Instana® provides an Enterprise Observability Platform with automated application performance monitoring capabilities to businesses operating complex, modern, cloud native applications no matter where they reside – on premises or in public and private clouds, including mobile devices or IBM Z mainframe computers. You can control modern hybrid applications with Instana’s 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 needed for customers as they optimize application performance, enable innovation and mitigate risk, helping DevOps increase efficiency and add value to software delivery pipelines while meeting their service-level and business-level objective.

Gartner rated IBM Instana as a leader in the Magic Quadrant for application performance monitoring and observability tools space. See the Gartner report here.

Features

Instana provides the following:

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

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

Instana uses sensors to provide automated infrastructure and Application Monitoring with no Plug-ins or Application restarts. – each sensor suppors an application component, middleware component, Operating System or other integration point to enable you to manage and monitor your infrastructure. At the moment of writing this document, Instana has the following number of integrations:

- AI Ops Integrations (18):
  - CI/CD Automation (7)
  - DevOps Tools (9)
- Cloud Operations (24)
- Containers and Orchestration (23)
- End 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, Frameworks (34):
  - Application Frameworks (7)
  - Application Monitoring (17)
  - Proxies and Service Meshes (4)
  - Tracing Technology (7)

Instana has a graphical user interface which can be used via a Web browser.

Instana architecture

Instana provides automatic, continuous discovery of your application stack. A single, lightweight agent per host continually discovers all components and deploys sensors 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 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 using their Dynamic Graph. The Dynamic Graph provides a dependency map allowing you to get to root cause of issues quickly.

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

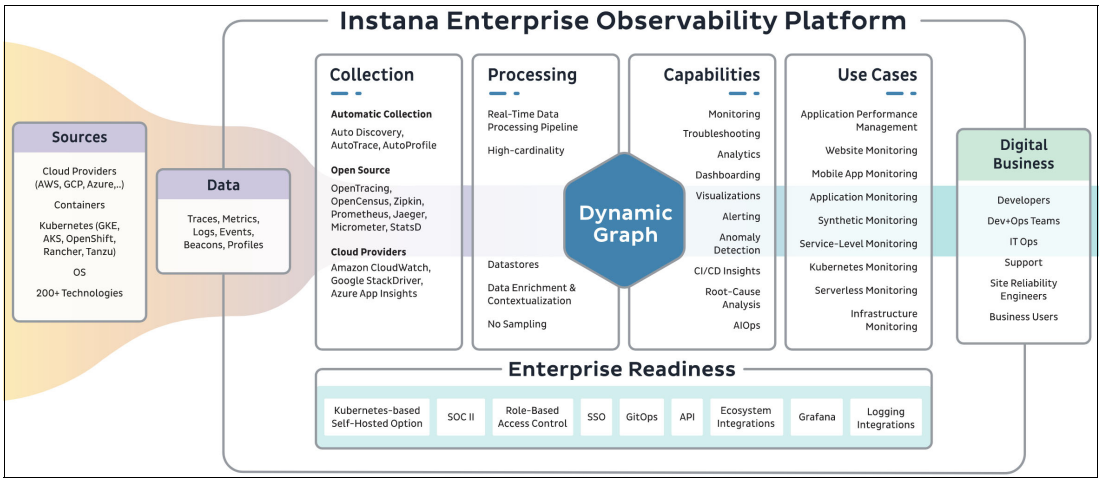


Figure 2-3 Instana platform architecture

Install methods

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 install methods. Compared to Kubernetes there are additional prerequisites for Red Hat OpenShift, so read the documentation before starting the installation.

Note: At the time of writing this publication Operator based install is not supported on Power servers as there is no image for instana-agent-operator available for ppc64le architecture on the OperatorHub.

On Red Hat OpenShift and Kubernetes clusters Instana agents are defined and running as PODs managed by a DaemonSet, which means that all worker nodes will 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 on Instana's Infrastructure window as shown in the Figure 2-4.

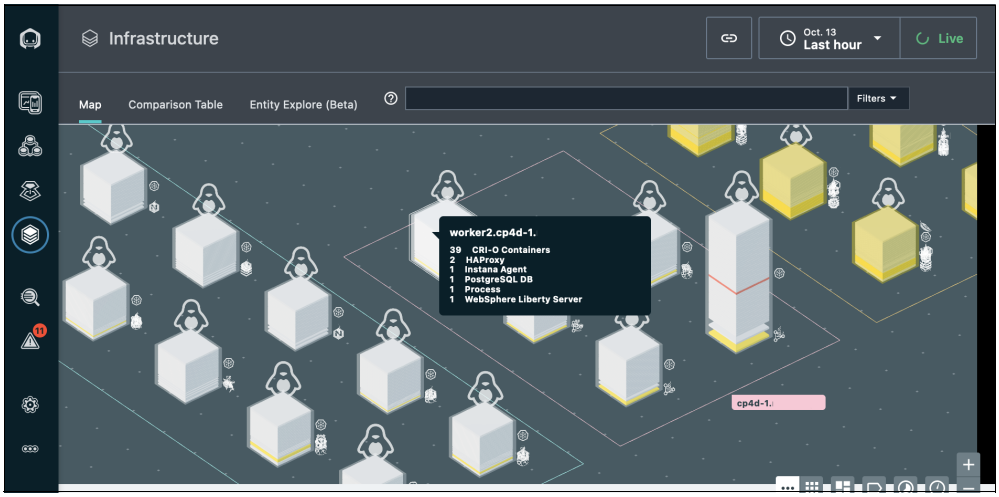


Figure 2-4 Instana - Infrastructure screen

[3] https://instanaimg.imgix.net/media/ObservabilityGraph-01.svg

Monitoring Power Systems servers CPU utilization in Instana

The Figure 2-5 shows CPU utilization statistics for a Power Systems based Red Hat OpenShift node.

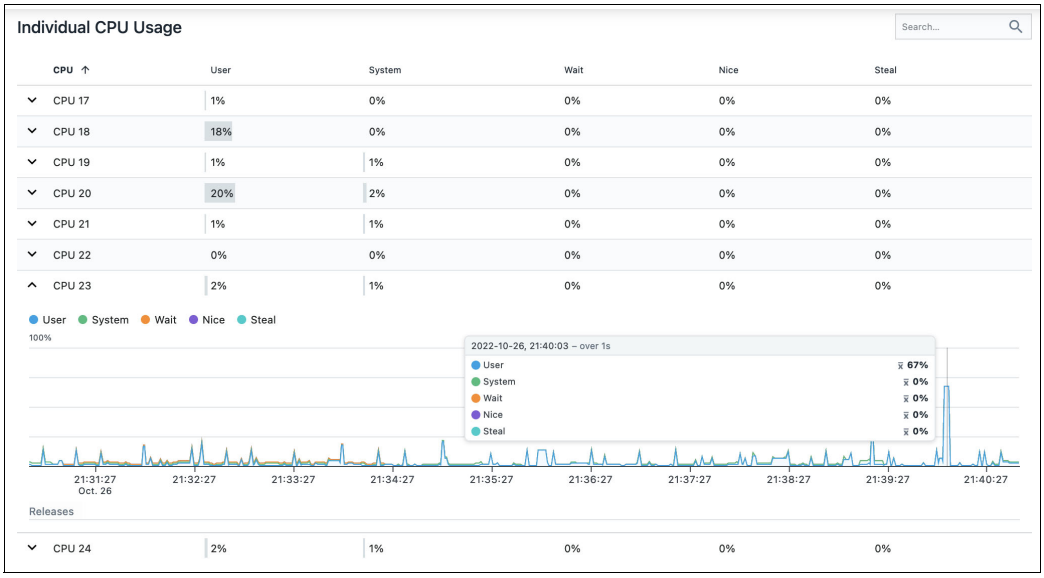


Figure 2-5 CPU usage

The number of CPUs are based on actual virtual CPUs assigned to the LPAR and the SMT settings. As our system has 4 vCPUs and SMT is set to 8 by default we have 32 CPUs shown in Instana and by the following commands also. The commands are started in Terminal window of the Red Hat OpenShift GUI or could be used after running 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 but not on bare metal servers.

Example 2-1 lscpu and lparstat

```
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 Systems HMC Instana sensor

IBM Power Systems HMC provides interfaces to monitor utilization of physical and virtual resources of Power Systems. These are REST API interfaces for Performance and Capacity Monitoring or PCM.

Instana has a sensor which makes use of this REST API to collect performance data and uncover any performance related anomaly. The sensor is supported from Version 10 Release 1 Service Pack 1010 of HMC software. Figure 2-6 shows the architecture of the IBM Power Systems HMC Instana sensor.

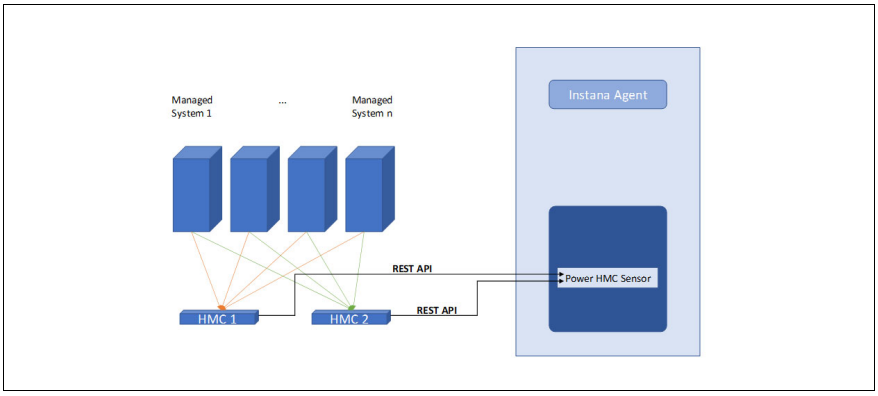


Figure 2-6 IBM Power Systems HMC Instana sensor architecture

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

Prerequisites

The following contains the prerequisites for IBM Power Systems HMC Sensor:

- The user which is set in the sensor configuration must have *hmcviewer* role on HMC.
- *Performance and Capacity Monitoring* must be enabled for the managed system to monitor.
- Enable the following flags:
  - *LongTermMonitoringEnabled.*
  - *AggregationEnabled.*
  - *EnergyMonitorEnabled.*

Setup and configuration

As the sensor is a remote sensor the Instana Agent has to be installed first on a system that can access the HMC.

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

`<agent_install_dir>/etc/instana.`

On a Linux box this is `/opt/instana/agent/etc/instana`.

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

Example 2-2 IBM Power Systems HMC Sensor configuration settings in 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 can not 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 can not 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 followings:

- Processor, memory and network metrics for power managed systems.
- Processor and memory metrics for hypervisor.
- Processor, memory, network and storage metrics for LPARs and vios.

CPU and memory utilization, power consumption logical partition data and many more. The default collection granularity is 300 seconds. Please see the full list of collected metrics here.

Troubleshooting

If the monitored IBM Power Systems HMC uses self signed certificates then this certificate has to be imported into the Instana agent's trusted certificate store. The message in Example 2-3 points to this problem.

Example 2-3 Self signed certificate error

```
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's certificate file, but check where the certificate store is located on the actual installation before using the suggested command to import it.

The certificate file can be downloaded via a web browser or using the command shown in Example 2-4. Before running the commands set the HMC, PORT, SERVERNAME variables according the actual setup.

Example 2-4 Get and import HMC certificate into Instana agent's 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: We are storing the actual location of the **`cacerts`** file because the documentation points to a different location and the command would fail with that.

Performance tuning related considerations

Instana provides real-time monitoring with 1-second collection granularity for metrics, logs, and traces. It then provides Unbound Analytics[4] to speed up root cause analysis. The collection and combination with Power Systems based metrics from HMC sensor extends the base capabilities to uncover performance related problems. Figure 2-7 shows a view of an Instana instance with a Power HMC being managed.

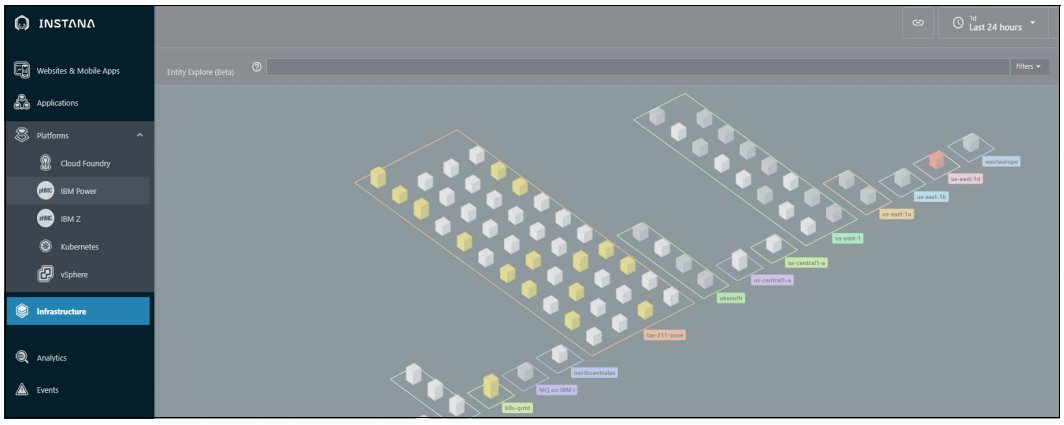


Figure 2-7 Instana's Platform menu with Power HMC entry

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

The following figures are example of Instana metrics gathered from the IBM Power Systems system through the HMC. Figure 2-8 shows the IBM Power Systems servers managed by the HMC.

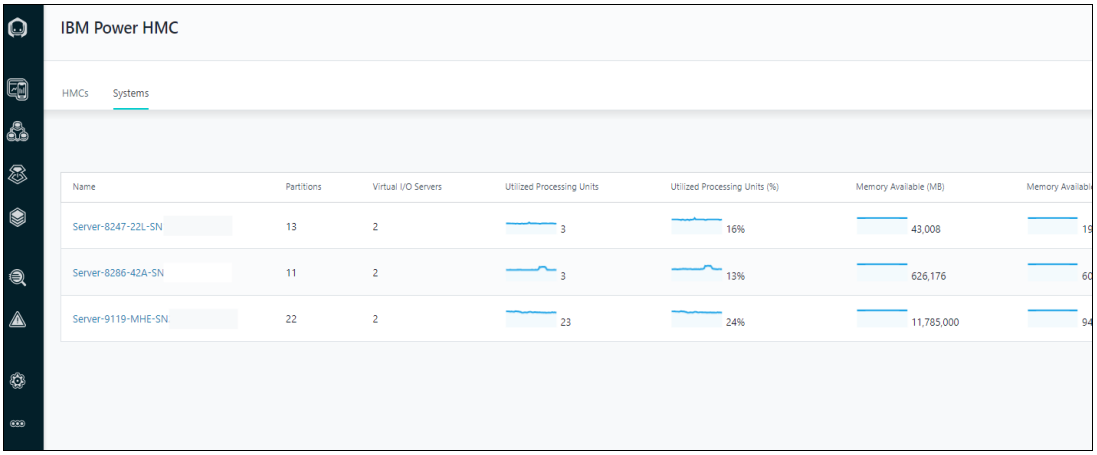


Figure 2-8 IBM Power Systems servers shown in IBM Power Systems HMC dashboard

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

Figure 2-9 IBM Power Systems server summary window

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

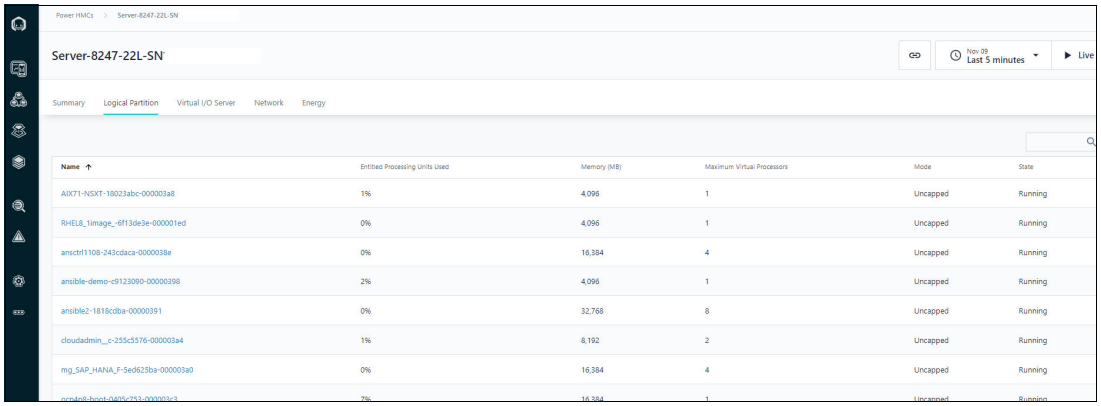


Figure 2-10 LPARs of a specific IBM Power Systems servers

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

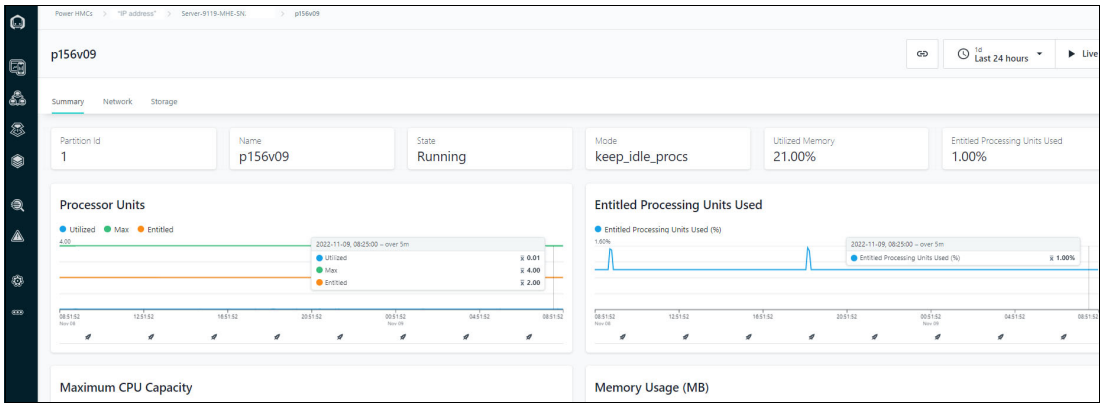


Figure 2-11 VIO Server processor utilization shown in Instana

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

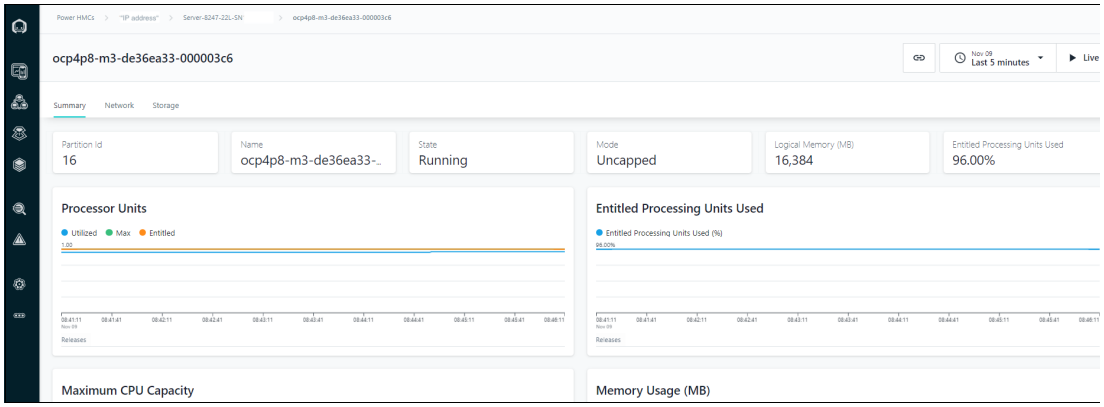


Figure 2-12 LPAR processor utilization shown in Instana

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

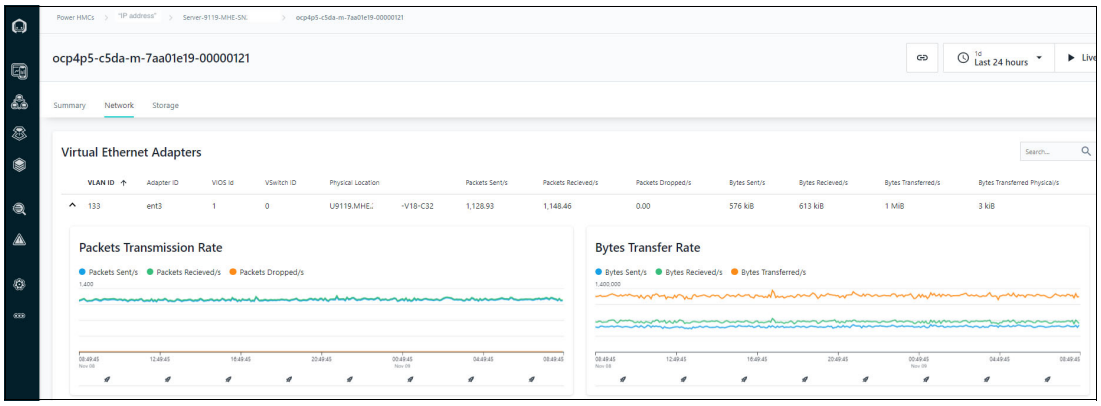


Figure 2-13 Network utilization

2.6.3 IBM Turbonomic Application Resource Management

In todays hybrid world we still have two main problems when provision infrastructure for applications and services. We can over provision or we can request and provision less

resources what is really needed for our application at peak time. Over provisioning is expensive as we are not using the resources what we are paying for, but we still can not take the risk not of not being able to support the peak load of our systems. This still leaves us with times where we have requests which can not 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 allowing us to keep the cost of our solution at an optimal level.

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

- Full-stack visualization using an application centric top-down approach. This will help in discovering 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 preventative, preemptive, and precise and can be automated.
- With the help of intelligent automation, we gain the value of 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:

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

Turbonomic architecture

Figure 2-14 shows the architecture of Turbonomic.

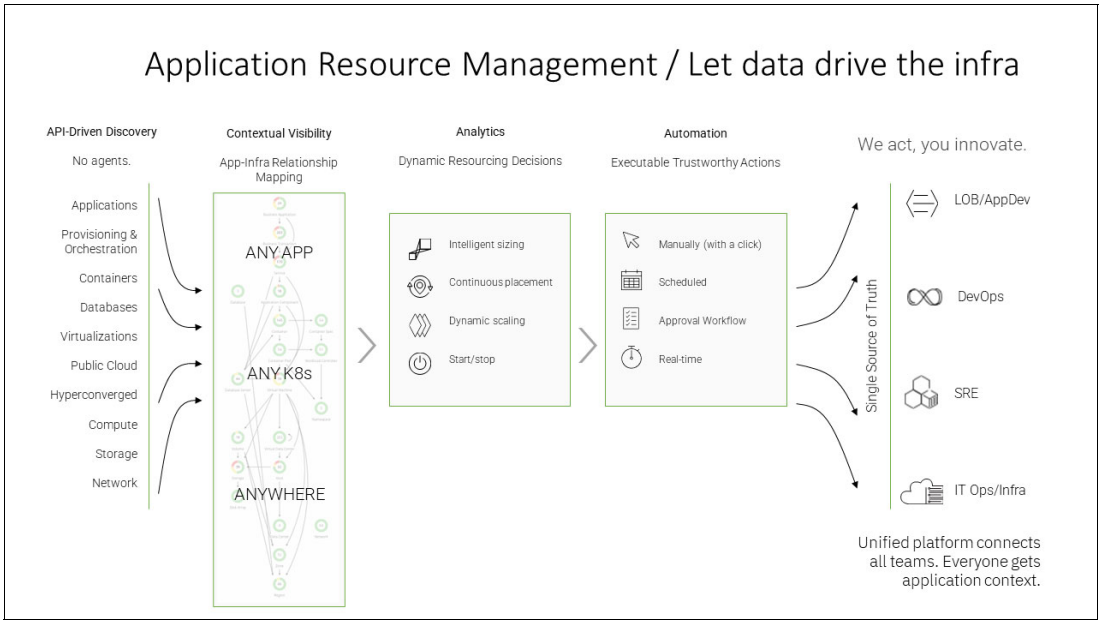


Figure 2-14 Turbonomic architecture

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

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.

Main features of Sysdig:

- Infrastructure as Code (IaC) security: shift security further left.
- Cloud Security Posture Management: continuous security of cloud based services via flagging misconfiguration, suspicious activities and excessive, unnecessary permissions.
- Vulnerability management: consolidate container and host security scanning, 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 via utilizing and automating Kubernetes-native network policy.
- Monitoring and troubleshooting: provide monitoring with deep insight using Kubernetes-native Prometheus.
- Compliance: validate compliance with major security standards like PCI, SOC2 and NIST for containers and host as well.

Architecture

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

Figure 2-15 shows the architecture of the platform.

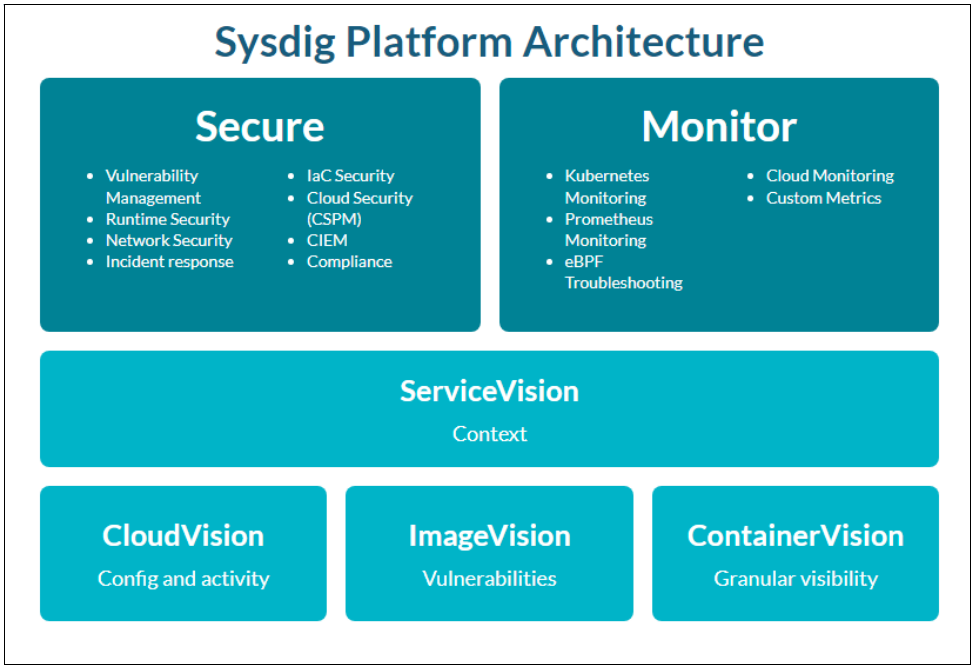


Figure 2-15 Sysdig architecture

Sysdig is built on open-source security stack which provides accelerated innovation and standardization. Its monitoring capabilities are built on Prometheus, so Red Hat OpenShift workloads on IBM Power Systems 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. As such it is integrated into IBM Cloud Monitoring and is the base of the capabilities of that service.

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

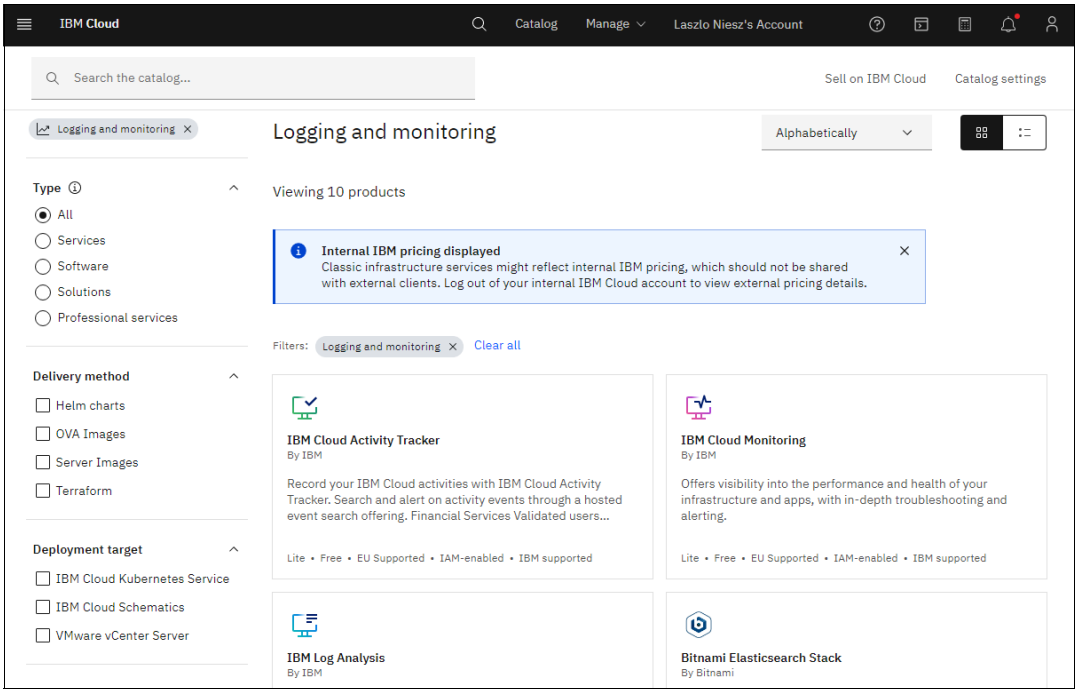


Figure 2-16 IBM Cloud Monitoring in the IBM Cloud catalog

This service is able to monitor Red Hat OpenShift and other Kubernetes clusters running in IBM Cloud. There is a free version for 30 days with some restrictions, but this is can provide an option to learn about IBM Cloud Monitoring and Sysdig.

2.6.5 Prometheus

Prometheus[5] is an open-source systems monitoring and alerting toolkit originally built at SoundCloud. Since its inception in 2012, many companies and organizations have adopted Prometheus and the project has a very active developer and user community. It is now a standalone open source project and maintained independently of any company. To emphasize this and to clarify the project's governance structure, Prometheus joined the Cloud Native Computing Foundation in 2016 as the second hosted project, after Kubernetes.

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

[5] https://prometheus.io/docs/introduction/overview/#what-is-prometheus

Red Hat OpenShift Monitoring

Red Hat OpenShift contains a full feature monitoring stack, which can be used to monitor the cluster health by default, but can be enabled for user projects as well. A key part of this stack is Prometheus which is shown in Figure 2-17.

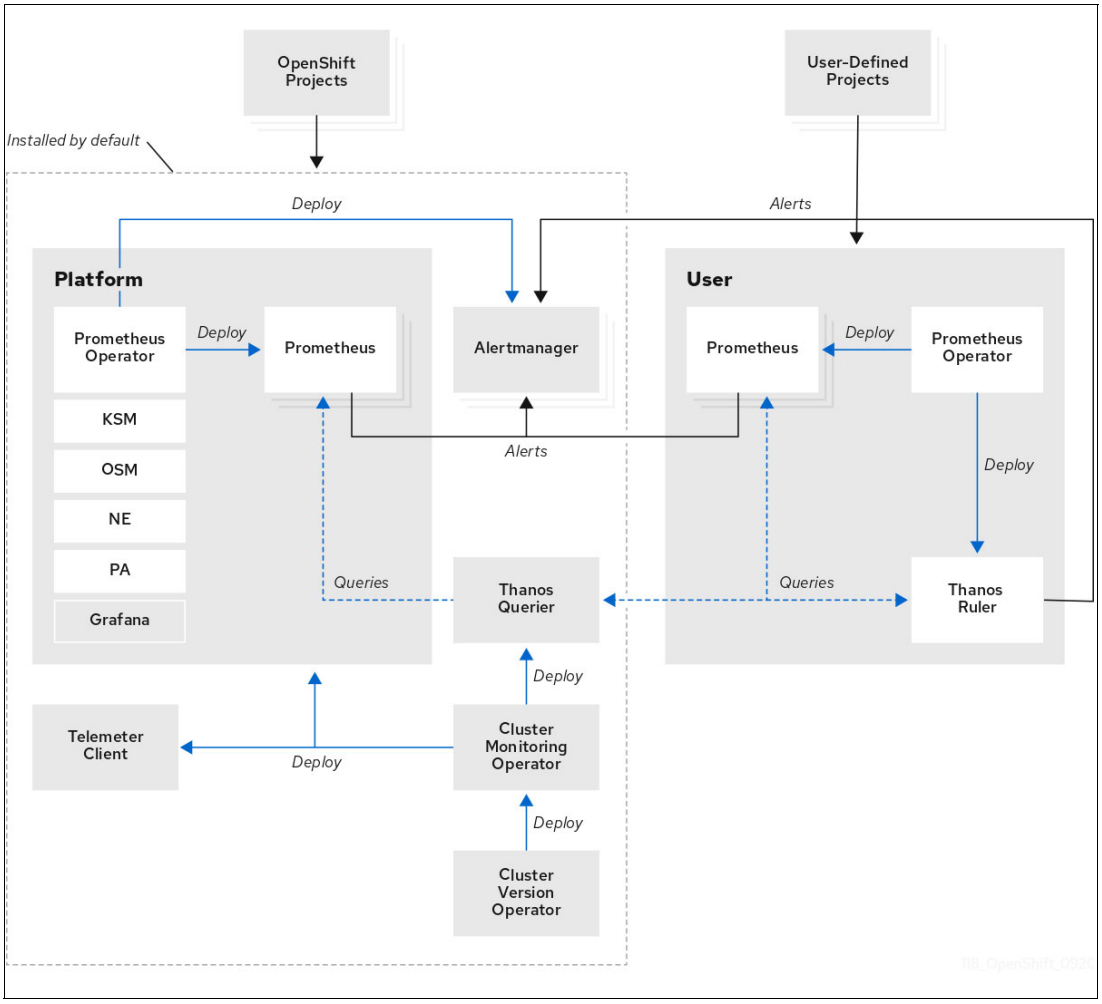


Figure 2-17 Red Hat OpenShift Monitoring stack

Figure 2-17 shows the parts of the monitoring stack which is installed by default,. These are used to monitor the Red Hat OpenShift platform base and controlling components. These components are deployed into the *openshift-monitoring* namespace.

Additionally user-defined projects and applications can be monitored if enabled. In this case there will be a new namespace created 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 along with managing Prometheus instances will also automatically generate monitoring target configurations based on Kubernetes labels. The monitoring of operator system metrics and nodes are done by a node-exporter agent.

After enabling user-defined project monitoring a separate instance of Prometheus is created in namespace *openshift-user-workload-monitoring*.

As node related monitoring is implemented in the base Prometheus instance will we concentrate on that in this chapter.

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

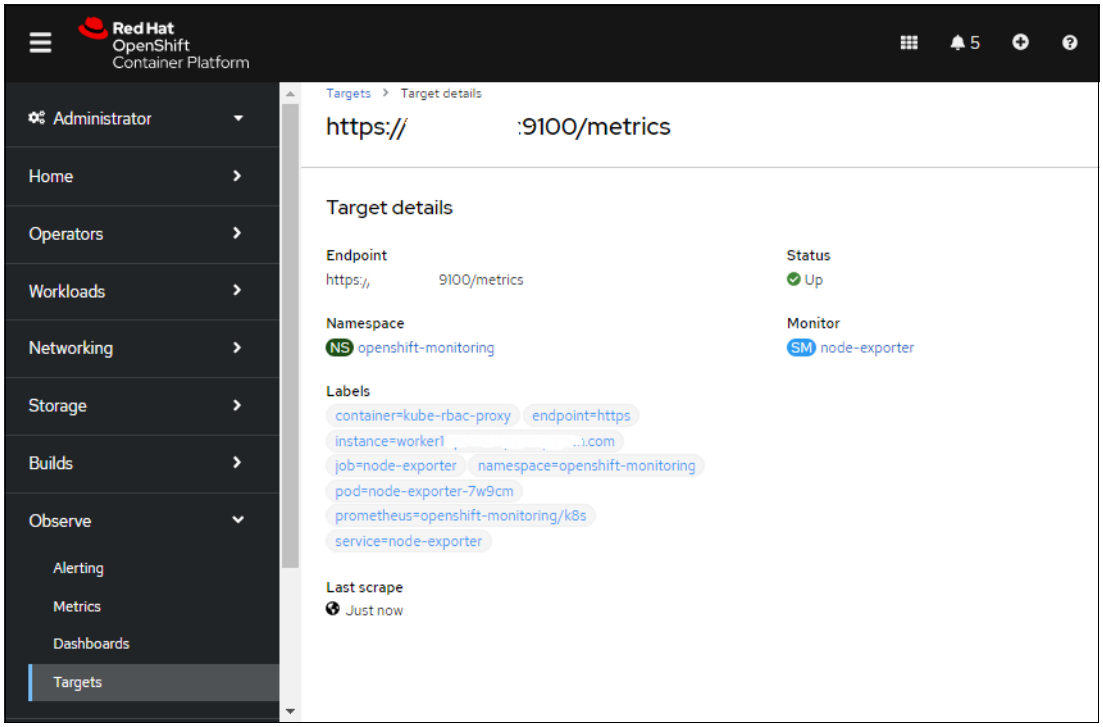


Figure 2-18 Red Hat OpenShift node as node-exporter target

Red Hat OpenShift provides an observability framework and the GUI has the following features for this under the menu *Observe*:

- Alerting: here we can see alerting rules based on metrics collected by Prometheus, silencers (to silence a predefined alerting rule) and the alerts fired by alerting rules.
- Metrics: this page provides a GUI to build custom queries based on the metrics collected by Red Hat OpenShift Monitoring.
- Dashboards: preconfigured dashboards for monitoring, with the possibility to dig deeper and show / edit the query of each dashboard element via pushing the *Inspect* button.
- Targets: shows all monitoring targets supported by the platform.

The Observe → Dashboard has two dashboards 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 chosen node.

The dashboard, by default, does not have any IBM Power Systems architecture specific metrics.

Figure 2-19 shows the Cluster dashboard.

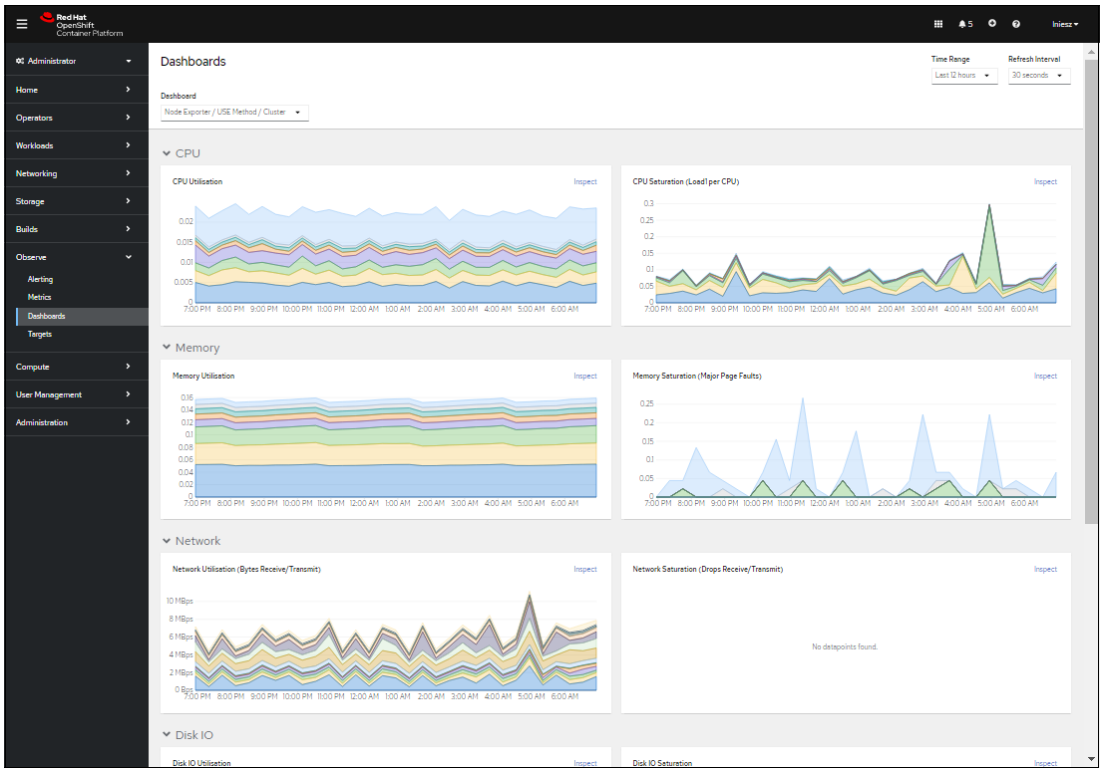


Figure 2-19 Node Exporter / USE Method / Cluster

Prometheus has a user interface (UI), however this is deprecated in Red Hat OpenShift version 4.10 and will be removed from version 4.11 as Red Hat OpenShift has its own Metrics UI available from Observe → Metrics menu.

The detailed configuration of predefined alerts and recored metrics can be seen in ConfigMap *prometheus-k8s-rulefiles-0* in the *openshift-monitoring* namespace. This is mounted into the Prometheus POD as */etc/prometheus/rules/prometheus-k8s-rulefiles-0* which is set to be used when the POD is starting.

2.6.6 Grafana

Grafana is a widely used operational dashboard which can be connected with many metric collection systems such as Prometheus.

Grafana provided the dashboard service in earlier versions of Red Hat OpenShift, however it is deprecated in version 4.10 and will be removed from version 4.11. Similar functionality is available using Red Hat OpenShift's own Observe → Dashboards service.

To configure Grafana we first have to define data sources, which in case of Red Hat OpenShift is normally the integrated Prometheus. This configuration in Red Hat OpenShift is done via a secret named *grafana-datasources-v2*. The default data source in Red Hat OpenShift looks like in what is shown in Example 2-5 on page 56.

Example 2-5 prometheus.yaml

```
{
    "apiVersion": 1,
    "datasources": [
        {
            "access": "proxy",
            "basicAuth": true,
            "secureJsonData": {
                "basicAuthPassword":
"8BCcK+MXnQUwzgYGcKEq5k+InSNWXZkueMRbZQvn8T/DwdFQ4XpfcKv5g76gjKfZgxQZBAJajyiQkRKJw
kRJk4hecJUHwlSCuOwGTml6/NCxVomCWvnMNb7pYZYWSDvOQGDpK6VCO66RdLhX7vF+SUZEe33/x4A+5iw
EN78wWtoMx+Ehb7WboET7jESxqpr1Yanj5Nr+bLeLslcwj2qYHOgxkrwKZBmO6YqKrYSeNqbJfmOkvwuFa
/QPQJBt8hEI4xBFWchTSOBrHaSjKhC/8zd+Z7mtK/i9VGioKEqEXOzYRAxGBh4GqdjXlTn233tFBgypScD
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. We can choose these to create a dashboard from the collected data.

The data source can be local, in which case we 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 specified by another config map: *grafana-dashboards*. Many specialized dashboard configurations can be downloaded from this Grafana Dashboards site.

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

3

IBM Power Systems performance capabilities

This chapter describes how you can take advantage of the performance features that are integrated by design in the IBM Power Systems processor when you are running your containerized workloads.

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

The following topics are covered in this chapter:

- "IBM Power Systems hardware" on page 58 describes the IBM Power Systems server architecture and its performance advantages.
- "IBM Power Systems Virtual Server" on page 79 describes how those components are utilized in the IBM PowerVS architecture.
- "Components" on page 84 breaks down the performance advantages by component.

3.1 IBM Power Systems hardware

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

IBM Power Systems has been leading the industry in infrastructure reliability with 25% lower downtime vs. comparable high-end server[1]. With IBM Power Systems E1080 we are making 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 operations of today's in-memory systems depend on memory reliability because of their large memory footprint. IBM Power Systems Power10 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 capability, high speed memory interconnections, and high speed data and networking connections the IBM Power Systems processor family can run more processes per core than competing architecture – resulting in a decreased hardware and software investment required. The IBM Power Systems Power10 family provides a range of options, from small entry servers like S1014 to enterprise architectures like E1080.

3.1.1 IBM Power Systems 10 capabilities

The IBM Power Systems Power10 processor was introduced to the general public on August 17, 2020 at the 32nd HOT CHIPS[3] semiconductor conference. At that meeting, the new capabilities and features of the latest IBM Power Systems processor microarchitecture and the IBM Power Systems Instruction Set Architecture (ISA) v3.1B were revealed and categorized according to the following IBM Power Systems 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.

- Energy efficiency focus area

  Up to threefold energy efficiency improvement in comparison to IBM POWER9 processor technology.

- Artificial intelligence (AI)-infused core focus area

[1] https://www.ibm.com/downloads/cas/VQ5B65YZ
[2] Based on IBM's internal analysis of the IBM product failure rate of DDIMMS vs Industry Standard-DIMMs
[3] https://hotchips.org/

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 Power Systems 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.

1. **Artificial Intelligence to provide rapid adoption of AI to realize growth and operational benefits.**

   Faster adoption of AI technologies as compared to IBM POWER9 from enhanced in-core AI inferencing capability in every server without requiring GPUs or any additional specialized hardware reducing the solution cost.

2. **Sustainability focus growing because of positive and negative drivers.**

   The IBM Power Systems Power10 processor delivers new levels of performance as compared to IBM POWER9 with 33% lower energy consumption for the same workload in IBM Power Systems E1080 vs IBM Power Systems E980.

3. **Resiliency driven by accelerated digitization and expectation of 24/7 access.**

   IBM Power Systems family of servers historically offer at least 25% less downtime vs comparable high-end servers designed to reduce cost of downtime and improve business result.

4. **Security to protect against current growth in scale and frequency of cyber-attacks and data breaches.**

   IBM Power Systems Power10 cores brings transparent memory encryption, this means all data in memory remains encrypted when in transit between memory and processor. This is designed to reduce risk and cost of potential data breaches and security incidents.

5. **Hybrid Cloud to mix on-premise powerful with cloud flexibility.**

   IBM Power Systems Power10 cores deliver a frictionless experience in extending mission-critical workloads across hybrid cloud because can offer up to 2.5x more value than a public cloud-only approach.

IBM Power Systems Power10 processor design

IBM Power Systems Power10 processor is built for cloud, whether private cloud, public cloud or hybrid cloud. One of the most significant improvements in the architecture is an advanced data plane for data-centric workloads, this improves the data bandwidth and capacity. IBM Power Systems Power10 processors are one of the first in the industry to support PCIe generation 5 devices and IBM Power Systems Power10 also adds a new Open Memory Interface (OMI) for better flexibility in supporting current and future memory technologies.

The processor’s core architecture is built on 7nm technology, improving performance by adding larger caches and reduced latencies. In terms of security, it brings an enhanced end-to-end encryption capability providing 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 added to each core to improve AI inferencing.

All these features and capabilities takes advantages of the energy efficiency for enterprise hybrid cloud with substantial scaling improvements for the largest partitions running mission critical workloads such as Oracle DB, SAP Hana, EPIC healthcare software and more by giving same power with less energy consumption.

IBM Power Systems Power10 processor chip

- Technology and Packaging:

  - 602mm2 7nm Samsung (18B devices)
  - 18 layer metal stack, enhanced device
  - Single-chip or Dual-chip sockets-computational Capabilities:
  - Up to 15 SMT8 Cores (2 MB L2 Cache / core) (Up to 120 simultaneous hardware threads)
  - Up to 120 MB L3 cache (low latency NUCA mgmt)
  - Improved energy efficiency relative to IBM POWER9
  - Enterprise thread strength optimizations
  - AI and security focused ISA additions
  - 2x general, 4x matrix SIMD relative to IBM POWER9
  - EA-tagged L1 cache, 4x MMU relative to IBM POWER9
- Open Memory Interface:
  - 16 x8 at up to 32 GT/s (1 TB/s)
  - Technology agnostic support: near/main/storage tiers
  - Minimal (< 10ns latency) add vs DDR direct attach
- PowerAXON Interface:
  - 16 x8 at up to 32 GT/s (1 TB/s)
  - SMP interconnect for up to 16 sockets
  - OpenCAPI attach for memory, accelerators, I/O
  - Integrated clustering (memory semantics)
- PCIe Gen 5 Interface:
  - x64 / DCM at up to 32 GT/s

A more detailed description of the IBM Power Systems Power10 Processor and its features is provided starting in section 3.1.4, “IBM Power Systems Power10 processor core” on page 66.

3.1.2 IBM Power Systems Power10 packaging

IBM Power Systems Power10 based servers are designed to create business agility with a flexible and secure hybrid cloud infrastructure, modernizing applications to maximize value from data. These servers integrate new cloud-native microservices to innovate with existing applications keeping in mind the concept of build once, deploy anywhere for optimized workload placement.

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

IBM Power Systems Power10 based servers offers dynamic agility to seamlessly adjust to changing business needs and are delivered with flexible consumption options with built-in cost optimization.

IBM Power Systems Power10 scale-out servers

IBM Power Systems Power10 scale-out servers provide enhanced performance and scale. The IBM Power Systems Power10 scale-out server's family includes:

  - Six new 1 and 2-socket, 2U and 4U height server models.
  - Up to 48 cores and 8TB memory footprints.
  - Up to 50% performance per price increase and 1.4X more system performance vs. IBM POWER9.
  - Expanded Dynamic Capacity consumption features with CUoD and PEP 2.0.
  - Value-driven solutions and higher technical standards.

Details on each of the models is shown below.

IBM Power Systems S1014 highlights

- Rack and tower form factors
- IBM Power Systems Power10 processors with 4 or 8 cores per server
- 8 DDIMM slots that provide up to 1 TB max memory capacity* (GA: 512GB)
- Main memory encryption for added security
- Five PCIe FHHL slots (4 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 module
- Supports external PCIe I/O Expansion Drawer
- Supports external SAS Storage Expansion Drawer
- Titanium power supplies to meet EU Efficiency Directives:
  - 2x 220 VAC (rack only) or 4x 1200W 110 VAC with C14 inlet
- Enterprise BMC managed

IBM Power Systems S1022s highlights

- IBM Power Systems Power10 processors with 4, 8, or 16 total cores per server
- 1-Hop flat CPU interconnect for maximum scalability
- 6 DDIMM slots that provide up to 2 TB max memory capacity* (GA: 1TB)
- Main memory encryption for added security
- Active memory mirroring support to reduce unplanned outages
- Ten PCIe HHHL slots (8 are Gen 5 capable), all slots are concurrently maintainable
- Up to 8 NVMe U.2 Flash Bays provide up to 51.2 TB of storage
- Secure and Trusted Boot with TPM module
- Supports external PCIe I/O Expansion Drawer
- Supports external SAS Storage Expansion Drawer
- Titanium power supplies to meet EU Efficiency Directives:
  - 2x 220 VAC with C14 inlet
- Enterprise BMC managed

IBM Power Systems S1022 and L1022 highlights

- IBM Power Systems Power10 processors with 12, 24, 32, or 40 total cores per server
- 1-Hop flat CPU interconnect for maximum scalability
- 32 DDIMM slots that provide up to 4 TB max memory capacity* (GA: 2TB)
- Main memory encryption for added security
- Active memory mirroring support to reduce unplanned outages
- Shared Capacity Utility support
- Ten PCIe HHHL slots (8 are Gen 5 capable), all slots are concurrently maintainable
- Up to 8 NVMe U.2 Flash Bays provide up to 51.2 TB of storage
- Secure and Trusted Boot with TPM module
- Supports external PCIe I/O Expansion Drawer
- Supports external SAS Storage Expansion Drawer
- Titanium power supplies to meet EU Efficiency Directives:
  - 2x 220 VAC with C14 inlet
- Enterprise BMC managed

IBM Power Systems S1024 and L1024 highlights

- IBM Power Systems Power10 processors with 12, 24, 32, or 48 total cores per server
- 1-Hop flat CPU interconnect for maximum scalability
- 32 DDIMM slots that provide up to 8 TB max memory capacity* (GA: 2TB)
- Main memory encryption for added security
- Active memory mirroring support to reduce unplanned outages
- Shared Capacity Utility support
- Ten PCIe FHHL slots (8 are Gen 5 capable), all slots are concurrently maintainable

- Up to 16 NVMe U.2 Flash Bays provides up to 102.4 TB of storage
- Secure and Trusted Boot with TPM module
- Supports external PCIe I/O Expansion Drawer
- Supports external SAS Storage Expansion Drawer
- Titanium power supplies to meet EU Efficiency Directives:
  - 4x 220 VAC with C14 inlet
- Enterprise BMC managed

IBM Power Systems Power10 enterprise servers

The IBM Power Systems Power10 enterprise servers add additional scalability and virtualization capabilities to address the most challenging workloads. In addition they add levels of hardware redundancy that allow the system to dynamically recover from hardware errors without an outage. Along with the enterprise scalability, the enterprise servers come with an enhanced level of support - IBM Power Expert Care. The E1050 provides up to 96 cores and 16 TB of RAM in a rack mounted format. The E1050 can start with one 4u enclosure and dynamically scale to a maximum of four enclosures. The E1080 can scale up to 240 8 way SMT cores and 64TB of RAM. It 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 Systems E1050 highlights

- 4U Server – 19” Rack Enclosure
- IBM Power Systems Power10 DCM processor w/ 12, 18, or 24 cores/socket, delivers up to 96 cores (doubling E950)
- 1-Hop flat CPU interconnect for maximum scalability and efficiency
- 64 DDIMM slots that provide up to 16TB max memory capacity* (GA: 8TB)
- Main Memory Encryption for added security:
  - Active Memory Mirroring support to reduce unplanned outages
  - 11 PCIe slots (8 are GEN5 capable), all slots are concurrently maintainable
- Up to 10 NVMe U.2 Flash Bays provides up to 64 TB of internal storage:
  - Secure and Trusted Boot with TPM module
  - Supports external PCIe I/O Expansion Drawer
  - Supports external SAS Storage Expansion Drawer
- Titanium power supplies to meet EU Efficiency Directives
- Enterprise BMC managed (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

IBM Power Systems E1080 highlights

- Follow-on to E980 enterprise system
- Modular rack mounted design to scale up to four 5U node drawers + one 2U control unit
- Max of 240 IBM Power Systems Power10 SMT8 cores (10, 12, 15 core offerings SCM package)
- New 32Gb SMP Cables (low latency) with Concurrent Maintenance capability
- Secure and Trusted Boot with Redundant TPM modules
- 2U System Control Unit Drawer
- Up to 64TB total memory (16TB per drawer)
- New OMI DDIMMs that provide increased memory bandwidth of 409 GB/s per socket
- Main Memory Encryption for added Security
- Ports available for future support of Memory Inception/Clustering
- Eight PCIe slots per drawer that are Blindswap with GEN5 support for future I/O

- Internal Storage - 4 NVMe Flash 7mm U.2 Bays per drawer
- Up to 16 I/O Expansion Drawers (4 Drawers per CEC Drawer)
- Enterprise BMC managed (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 Power Systems Power10 processor

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

The *IBM's Power10 Processor* session material as presented at the 32nd HOT CHIPS conference is available through the HC32 conference proceedings archive at this web page.

IBM Power Systems Power10 processor overview

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

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

The processor supports the following three distinct functional interfaces which all are capable to run with a signaling rate of up to 32 GTps[4]:

- Open memory interface

  The IBM Power Systems Power10 processor has eight memory controller unit (MCU) channels that support one open memory interface (OMI) port with two OMI links each[5]. One OMI link aggregates 8 lanes running at 32 GTps speed and connects to one memory buffer based differential DIMM (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 Power Systems Power10 processor to facilitate the connectivity to other processors in a symmetric multiprocessing (SMP) architecture configuration. Each SMP connection requires 18 lanes, eight data lanes plus one spare lane per direction (2 x(8+1)). In this way 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 also allows the use of 128 data lanes to potentially connect accelerator or memory devices through the OpenCAPI protocols. Also, it can support memory cluster and memory interception architectures.

[4] Giga transfers per second (GTps)
[5] The OMI links are also referred to as OMI sub-channels.

Because of the versatile characteristic of the technology, it is also referred to as *PowerAXON* interface (Power A-bus/X-bus/OpenCAPI/Networking[6]). The OpenCAPI and the memory clustering and memory interception use cases can be pursued in the future and are currently not use by available technology products.

- PCIe Version 5.0 interface

  To support external I/O connectivity and access to internal storage devices, the IBM Power Systems Power10 processor provides differential Peripheral Component Interconnect Express version 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:

  – 1 x16 PCIe Gen 4
  – 2 x8 PCIe Gen 4
  – 1 x8, 2 x4 PCIe Gen 4
  – 1 x8 PCIe Gen 5, 1 x8 PCIe Gen 4
  – 1 x8 PCIe Gen 5, 2 x4 PCIe Gen 4

Figure 3-1 shows the IBM Power Systems Power10 processor die with several functional units labeled. Note, 16 SMT8 processor cores are shown, but only 10-, 12-, or 15-core processor options are available for IBM Power Systems E1080 server configurations.

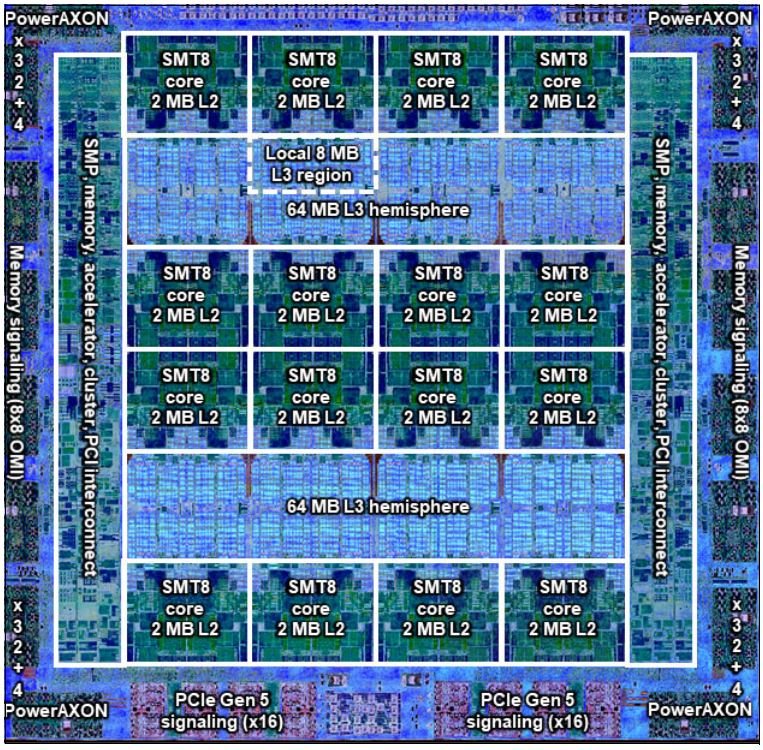


Figure 3-1 The IBM Power Systems Power10 processor chip (Die photo courtesy of Samsung Foundry)

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

Table 3-1 Summary of the IBM Power Systems10 processor chip and processor core technology

Technology	IBM Power Systems Power10 processor
Processor die size	602 mm^2

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

Technology	IBM Power Systems Power10 processor
Fabrication technology	► CMOS[a] 7-nm lithography ► 18 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[b] of POWER8 and POWER9

a. Complimentary metal-oxide-semiconductor (CMOS)
b. IBM Power instruction set architecture (Power ISA)

The IBM Power Systems Power10 processor is packaged as single-chip module (SCM) for exclusive use in the IBM Power Systems E1080 servers. The SCM contains the IBM Power Systems 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 IBM Power Systems E1080 servers.

Figure 3-2 shows the logical diagram of the IBM Power Systems Power10 SCM.

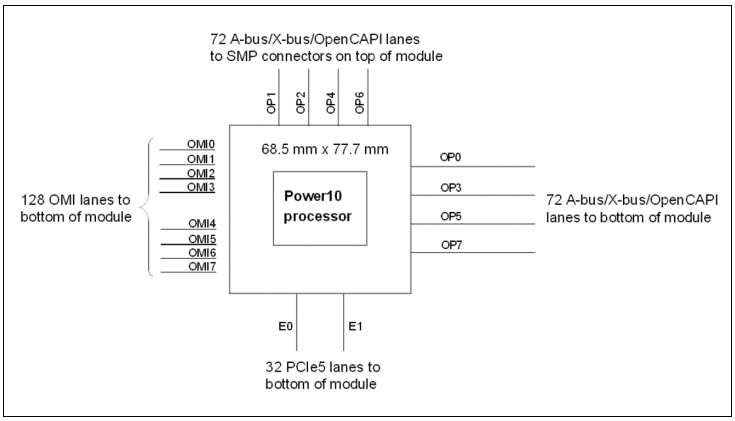


Figure 3-2 IBM Power Systems Power10 SCM logical diagram

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 open memory interface ports (OMI0 to OMI7) with two OMI links each provide access to the buffered main memory differential DIMMs (DDIMMs). The 32 PCIe Gen 5 lanes are grouped in two PCIe host bridges (E0, E1).

Figure 3-3 on page 66 shows a physical diagram of the IBM Power Systems Power10 SCM. The eight SMP connectors (OP1A, OP1B, OP2A, OP2B, OP4A, OP4B, OP6A, and OP6B) externalize 4 SMP busies, which are used to connect system node drawers in 2-, 3-, and 4-node IBM Power Systems E1080 configurations. The OpenCAPI connectivity options are also indicated, although they are not used by any commercially available product.

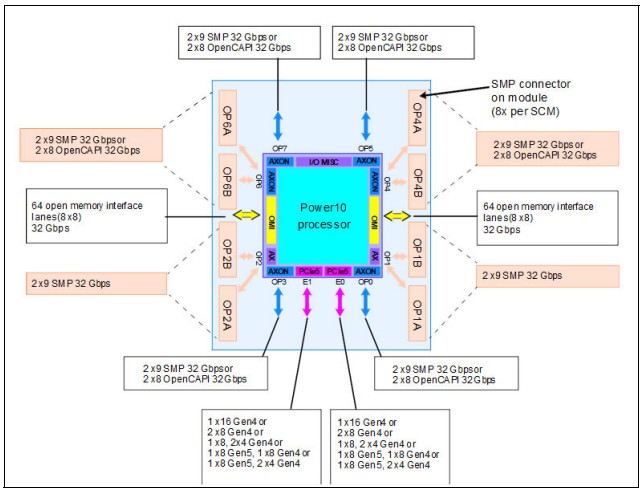


Figure 3-3 IBM Power Systems Power10 single chip module

3.1.4 IBM Power Systems Power10 processor core

The IBM Power Systems Power10 processor core inherits the modular architecture of the IBM POWER9 processor core, but the re-designed and enhanced micro-architecture significantly 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 significant performance gains for machine learning, particularly for AI inferencing workloads.

The IBM Power Systems E1080 server uses the IBM Power Server Power10 enterprise-class processor variant in which each core can run with up to eight essentially independent hardware threads. If all threads are active, the mode of operation is referred to as 8-way simultaneous multithreading (SMT8) mode. An IBM Power System Power10 core with SMT8 capability is named Power10 SMT8 core or SMT8 core for short. The IBM Power Systems Power10 core also supports modes with four active threads (SMT4), two active threads (SMT2) and one single active thread (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 on page 67 shows the functional units of a SMT8 core where all 8 threads are active. The two execution resource domains are highlighted with colored backgrounds in two different shades of blue.

Each of the two execution resource domains supports between one and four threads and includes 4 vector scalar units (VSU) of 128-bit width, two matrix-multiply assist (MMA) accelerators, and one quad-precision floating-point (QP) and decimal floating-point (DF) unit.

One VSU and the directly associated logic is called an execution *slice*. Two neighboring slices can also be used as a combined execution resource which is then named *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, one to two threads each share a four slices resource domain. Figure 3-4 on page 67 also indicates other essential resources that are shared among the SMT threads, such as instruction cache, instruction buffer and L1 data cache.

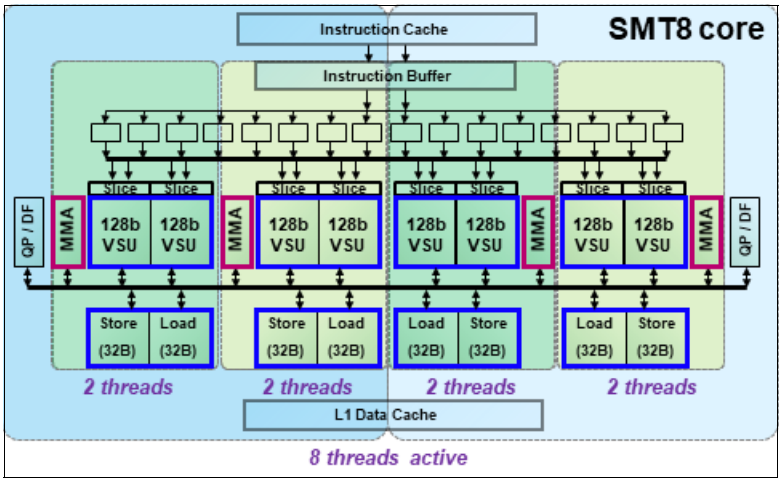


Figure 3-4 IBM Power Systems Power10 SMT8 core

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 executions resources.

Micro-architecture performance and efficiency optimization lead to a significant 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 IBM Power Systems Power10.

The IBM Power Systems 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 next. The change in relation to the IBM POWER9 processor core implementation is provided in parentheses.

Enhanced computation resources

The following are major computational resource enhancements:

- Eight vector scalar unit (VSU) execution slices, each supporting 64-bit scalar or 128-bit single instructions multiple data (SIMD) +100% for permute, fixed-point, floating-point, and crypto (Advanced Encryption Standard (AES)/SHA) +400% operations.
- Four units for matrix-math assist (MMA) acceleration each capable of producing a 512-bit result per cycle (new), +400% Single and Double precision FLOPS plus support for reduced precision AI acceleration).
- Two units for quad-precision floating-point and decimal floating-point operations additional instruction types.

Larger working sets

The following major changes were implemented in working set sizes:

- L1 instruction cache: 2 x 48 KB 6-way (96 KB total) (+50%)
- L2 cache: 2 MB 8-way (+400%)
- L2 translation lookaside buffer (TLB): 2 x 4K entries (8K 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)
- Translation lookaside buffer (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 yield ERAT access only on a cache miss.

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

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

3.1.5 Simultaneous multithreading

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

If more than one hardware thread is active, the processor runs in simultaneous multi-threading (SMT) mode. In addition to the ST mode, the IBM Power Systems 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. This ability significantly enhances

the throughput and scalability of multi-threaded applications and optimizes the compute density for single-threaded 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 POWER4.

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

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

3.1.6 Matrix-multiply assist AI workload acceleration

The matrix-multiply assist (MMA) facility was introduced by the IBM Power Instruction Set Architecture (ISA) v3.1. The related instructions implement numerical linear algebra operations on small matrices and are meant to accelerate computation-intensive kernels, such as matrix multiplication, convolution, and discrete Fourier transform.

To efficiently accelerate MMA operations, the IBM Power Systems 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 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. See Figure 3-5 on page 70 for a MMA diagram.

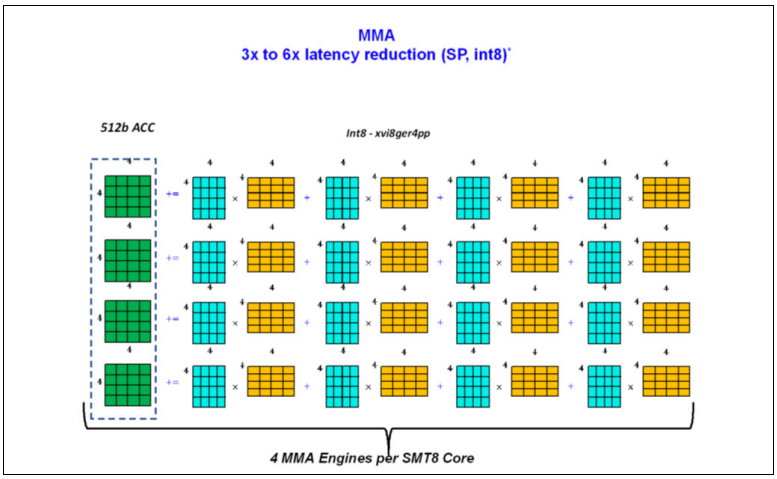


Figure 3-5 MMA engines per SMT8 Core

IBM Power Systems Power10 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. See Figure 3-6 the for a complete distribution from the Open Memory Interface to the MMA accumulators, passing through the cache levels.

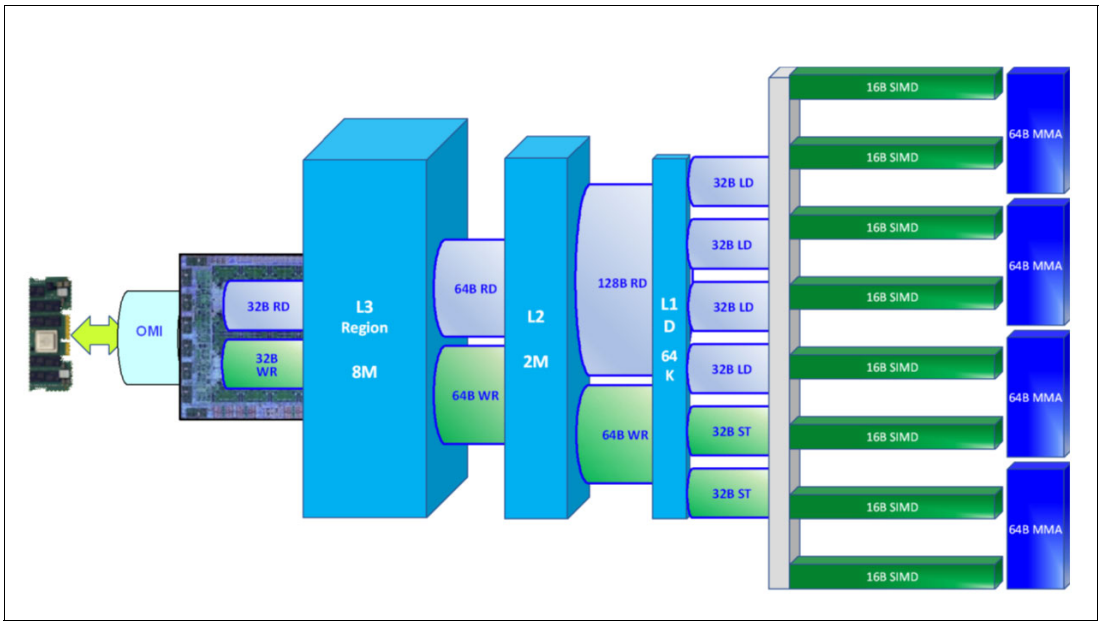


Figure 3-6 Open Memory Interface to MMA

In order to leverage the performance advantages of the MMA capabilities in the IBM Power Systems Power10 processor, you should utilize libraries that have been built for IBM Power Systems Power10 MMA.

Of special importance for AI inferencing is the OpenBLAS library. This library is used by popular frameworks such as PyTorch, Tensorflow and ONNX Runtime.Chapter 6.1, “AI Inferencing with Red Hat OpenShift and IBM Power Systems Power10” on page 184 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 Power Systems Power10 processor's high throughput math engine, see the white paper *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 Power Systems Power10 processor includes a large on-chip L3 cache of up to 120 MB with a non-uniform cache access (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 8MB of L3 cache. It also can access 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 and also can provide aggregate storage for the on-chip cache footprint.

Intelligent L3 cache management enables the IBM Power Systems Power10 processor to optimize the access to L3 cache lines and minimize cache latencies. The L3 includes a replacement algorithm with data type and reuse awareness. It 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 Power Systems Power10 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 Power Systems Power10 processor introduces a new and innovative open memory interface (OMI). The OMI is driven by 8 on-chip memory controller units (MCUs) and is implemented in two separate physical building blocks that lie in opposite areas at the outer edge of the IBM Power Systems Power10 die. Each area supports 64 OMI lanes that are grouped in four ports. One port in turn consists of two links with 8 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 x 512 GBps = 1 TBps per IBM Power Systems Power10 single chip module (SCM).

The OMI physical interface enables low latency, high-bandwidth, technology-agnostic host memory semantics to the processor and allows attaching established and emerging memory elements. With the IBM Power Systems E1080 server. OMI initially supports one main tier, low-latency, enterprise-grade Double Data Rate 4 (DDR4) differential DIMM (DDIMM) per

OMI link. This configuration yields a total memory capacity of 16 DDIMMs per SCM and 64 DDIMMs per IBM Power Systems E1080 server node. The memory bandwidth depends on the DDIMM density configured for a specific IBM Power Systems 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 slightly reduced to 375 GBps per SCM if 128 GB or 256 GB DDIMMs running at 2933 MHz are used.

In summary, the IBM Power Systems Power10 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 single chip module (16 x 8 lanes)

3.1.9 Pervasive memory encryption

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

The IBM Power Systems Power10 processor supports the following modes of operation:

- AES XTS mode

  XTS is an abbreviation for the xor–encrypt–xor based tweaked-codebook mode with ciphertext stealing. AES XTS provides a block cipher with strong encryption, which is particularly useful to encrypt 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 Power Systems Power10 processor is supported for future use if persistent memory solutions become available for IBM Power Systems servers.

- AES CTR mode

  CTR stands for *Counter* mode of operation and designates a low-latency AES bock 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 IBM Power Systems E1080 servers support the AES CTR mode for pervasive memory encryption. Each IBM Power Systems Power10 processor holds a 128-bit encryption key that is used by the processor's MCU to encrypt the data of the differential DIMMs that are attached to the OMI links.

The MCU crypto engine is transparently 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 switched off through any administrative interface.

Note: The pervasive memory encryption of the IBM Power Systems 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 memory controller unit 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 Power Systems Power10 processor has an on-chip accelerator called nest accelerator unit or NX unit. The coprocessor features that are available on the IBM Power Systems 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 Secure Hash Algorithm (SHA) cryptography.
- Random number generation.

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

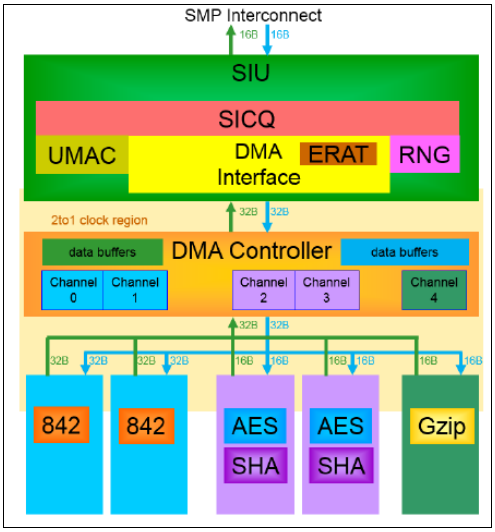


Figure 3-7 Block diagram of the NX unit

Each one of the AES/SHA engines, data compression, and Gzip units consist of a coprocessor type and the NX unit features three coprocessor types. The NX unit also includes more support hardware to support coprocessor invocation by user code, use of effective addresses, 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 Power Systems Power10 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 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 Power Systems Power10 systems implements a high throughput engine that can perform the equivalent work of multiple cores. The system performance can benefit by off-loading these expensive operations to on-chip accelerators, which in turn can greatly 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 way of 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:

- IBM support article *Using the POWER9 NX (gzip) accelerator in AIX.*
- IBM Power Systems community article *POWER9 GZIP Data Acceleration with IBM AIX.*
- AIX community article *Performance improvement in openssh with on-chip data compression accelerator in power9*.
- IBM Documentation: *nxstat Command.*

3.1.11 SMP interconnect and accelerator interface

The IBM Power Systems 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, symmetric multiprocessing link interface, enabling up to 16 IBM Power Systems Power10 processors to be combined into a large, robustly scalable, 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 Power Systems Power10 systems to directly use memory throughout the cluster.

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

Because of the versatile nature of signaling technology, the 32 Gbps interface is also referred to as Power/A-bus/X-bus/OpenCAPI/Networking (*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 Power Systems Power10 processor die. The IBM Power Systems E1080 server makes use of this interface to implement single-drawer chip-to-chip and drawer-to-drawer chip interconnects.

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

- Three (2 x 9)-bit on planar buses per IBM Power Systems Power10 single chip module (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, an increase of 33% compared to the IBM Power Systems E980 single-drawer implementation.
- 4-way SMP architecture implementations build out of four IBM Power Systems Power10 SCMs per drawer in 1-hop topology.

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

- Three (2 x 9)-bit buses per IBM Power Systems Power10 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 2-hop topology.

Figure 3-8 on page 76 shows the SMP connections for a fully configured 4-node 16-socket IBM Power Systems E1080 system. The blue lines represent the chip-to-chip connections within one system node. The green lines represent the drawer-to-drawer SMP connections.

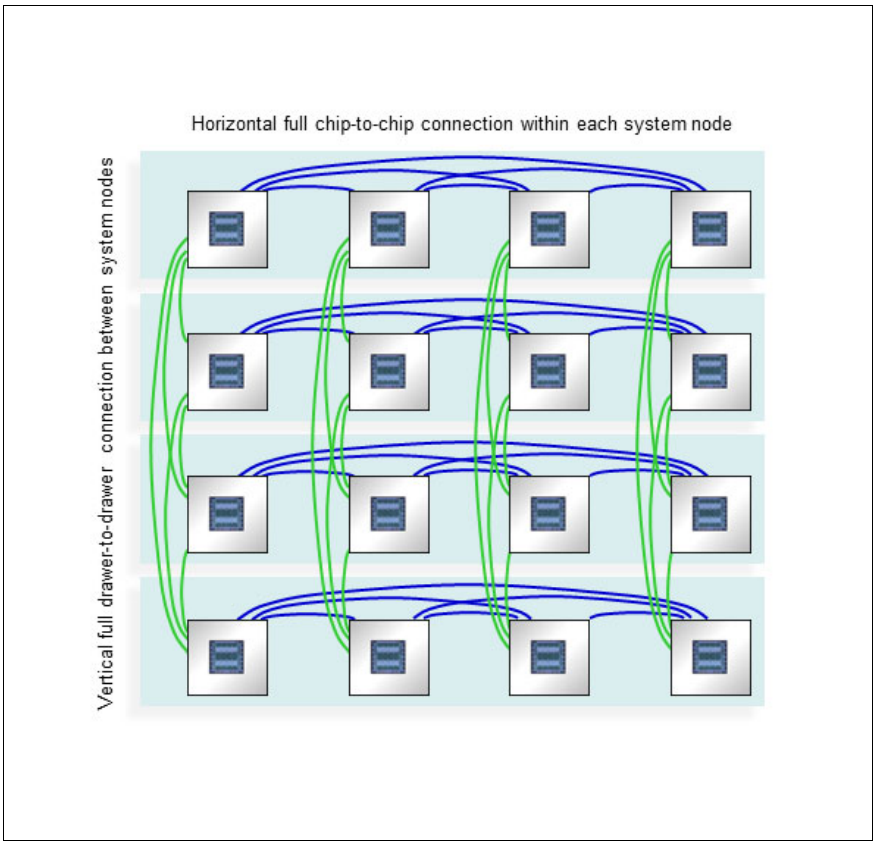


Figure 3-8 SMP interconnect in a 4-node 16-socket IBM Power Systems E1080 system

From the drawing that is shown in Figure 3-8, it is easy to 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 Systems and performance management

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

As in the previous IBM POWER9 EnergyScale implementation, the IBM Power Systems Power10 EnergyScale technology supports dynamic processor frequency changes that are dependent 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 Power Systems Power10 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 maximum performance mode (MPM). Because of this redundant nature of characteristics, the DPM for IBM Power Systems Power10 processor-based systems was removed in favor of an enhanced MPM. For example, the maximum frequency is now

achievable in the IBM Power Systems Power10 enhanced maximum performance mode (regardless of the number of active cores), which was not always the case with IBM POWER9 processor-based servers.

The IBM Power Systems Power10 processor-based IBM Power Systems E1080 system features MPM enabled by default. This mode dynamically adjusts processor frequency to maximize performance and enable a much higher processor frequency range. Each of the power saver modes deliver consistent system performance without any variation if the nominal operating environment limits are met.

For IBM Power Systems Power10 processor-based systems 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 that a processor module can run at:

- 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 are enabled to run at higher frequencies.

The following IBM Power Systems Power10 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*.

- Maximum performance mode

  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 well above the nominal frequency and the upper limit is given by the system's maximum frequency.

  The MTFR is published as part of the system specifications of a specific IBM Power Systems Power10 system 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 in turn 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 the IBM Power Systems E1080.

- Idle power saver mode (IPS)

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

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

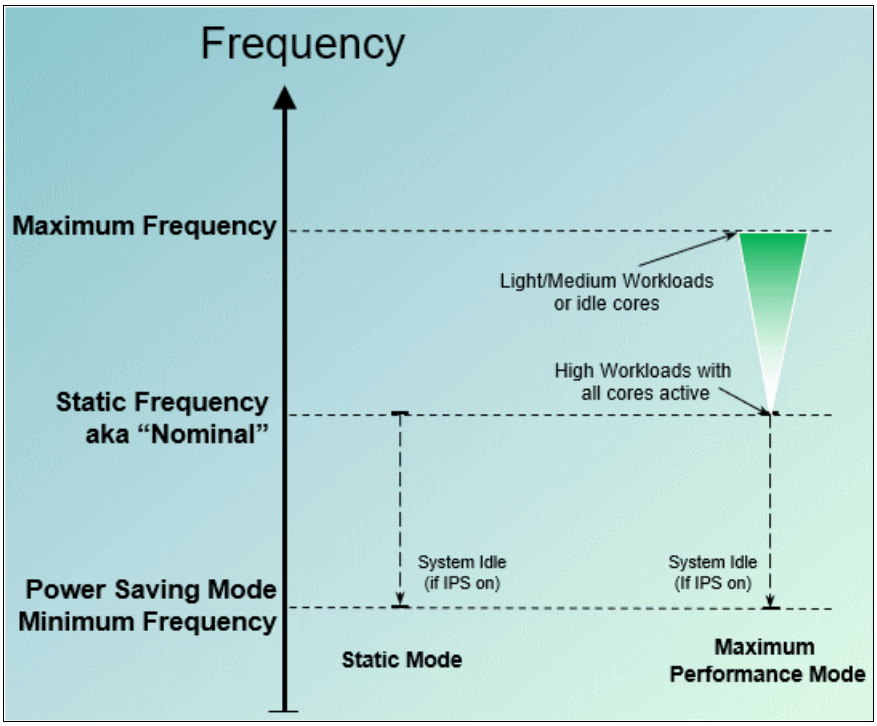


Figure 3-9 IBM Power Systems Power10 power management modes and related frequency ranges

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 IBM Power Systems E1080 server.

Table 3-3 Characteristic frequencies and frequency ranges for IBM Power Systems 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 to 3.90 GHz (max)
EDP3	12	3.40	3.60	3.60 to 4.15 GHz (max)
EDP4	15	3.25	3.55	3.55 to 4.00 GHz (max)

For IBM Power Systems 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 also 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 IBM Power Systems E1080 server.

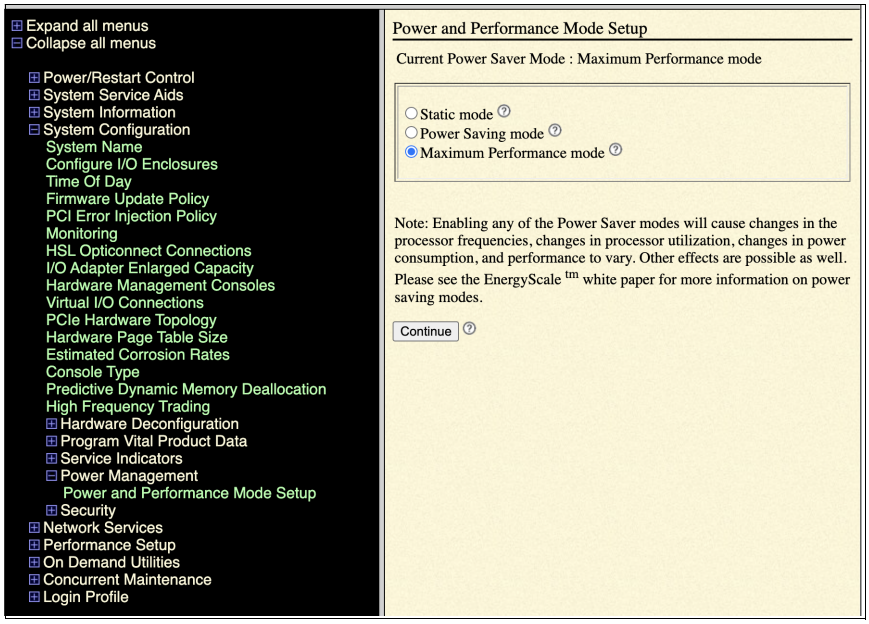


Figure 3-10 IBM Power Systems E1080 ASMI menu for Power and Performance Mode setup

Figure 3-11 shows the HMC menu for power and performance mode setup.

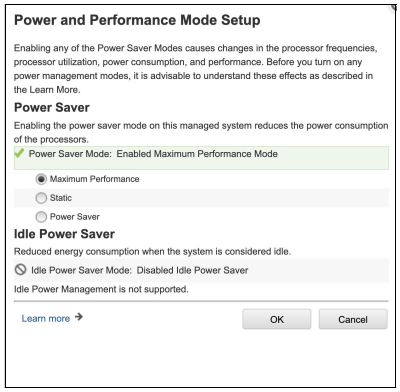


Figure 3-11 IBM Power Systems E1080 HMC menu for Power and Performance Mode setup

3.2 IBM Power Systems Virtual Server

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

As PowerVS runs on IBM Power Systems processors, it integrates seamlessly with workloads that you are running on premise today allowing you an easy entry point to moving your existing workloads to the cloud should you desire to do so. With PowerVS You get fast,

self-service provisioning, flexible management both on-premises and off-premises, and similar to on-premises it can be connected to access a stack of enterprise services from IBM – all with pay-as-you-use billing that lets you easily scale up and out. You can quickly deploy an IBM PowerVS infrastructure to meet your specific business needs and easily control workload demands.

3.2.1 Architecture

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

On IBM Cloud, the Identity and access management (IAM) service provides ability to securely authenticate users, control access to PowerVS resources with resource groups, and allow access to specific resources for a set of users with access groups. PowerVS requires additional access for IBM Cloud features such as Direct Link, Transit Gateway service, Virtual Private Cloud, and so on.

Compute resources and operating systems

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 IBM Cloud team, but the operating system, middleware, runtime and the software applications and data are to be managed by client as depicted in Figure 3-12.

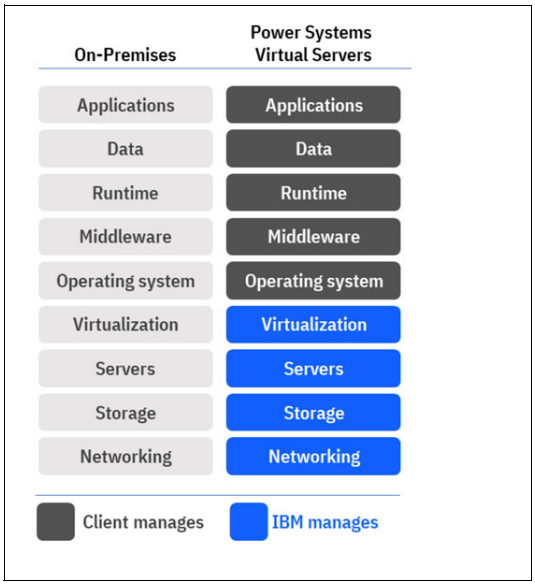


Figure 3-12 PowerVS RACI matrix (Ref: IBM Cloud Docs)

IBM provides the stock AIX and IBM i images during PowerVS provisioning. The clients can also bring their own custom AIX, IBM i or Linux image. 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 to an IBM Cloud Object Storage (COS) location. The COS bucket should be created and the custom image should be uploaded there. The PowerVS offering allows provisioning a new virtual server based on an OVA image.

Storage and Networking

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 generally not suitable for production workloads. Your storage tier choice should consider not just the average I/O load, but also the peak IOPS of the storage workload. The Tier 3 storage is currently set to 3 IOPS/GB, and the Tier 1 storage is currently set to 10 IOPS/GB, but these numbers can change over time for PowerVS. It is important to remember that after the IOPS limit is reached for the storage volume, the I/O latency increases.

PowerVS instances support both private or public network interfaces. Public network provides an easy and quick method to connect to a PowerVS instance. In this case, IBM configures the network environment to enable a secure public network connection from the internet to the 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 firewall and supports various secure network protocols. On the other hand, the private network allows the 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 PowerVS connections. Connecting a PowerVS instance to IBM Cloud Transit Gateway network grants access to all networks connected on the transit gateway such as providing connectivity between PowerVS environments located at two different data centers as well as interconnecting PowerVS to the IBM Cloud classic and VPC infrastructures, keeping traffic within the IBM Cloud network.

For example, the network architecture shown in Figure 3-13 allows connectivity between multiple PowerVS locations for high availability (HA) and disaster recovery (DR) solutions, as well as connectivity to IBM Cloud classic infrastructure environment and IBM Cloud VPC environment.

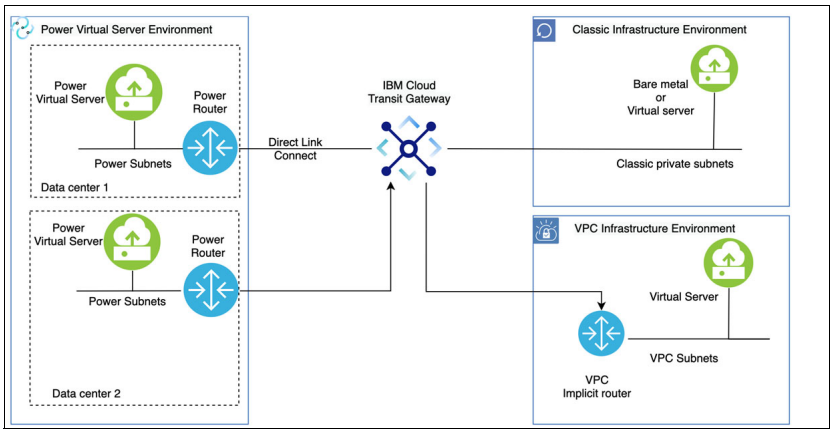


Figure 3-13 Transit Gateway connectivity across different environments

PowerVS instances support VM pinning to hosts, as well as affinity or anti-affinity rule. The affinity or anti-affinity rule are also supported for storage volumes.

High Availability and Disaster Recovery

To support basic High Availability (HA) capabilities, the PowerVS instance will restart the virtual servers on a different host system if a hardware failure occurs. For more advanced HA, the IBM PowerHA® SystemMirror® for IBM AIX running in the PowerVS environment can be used. For PowerVS instances that are part of the IBM PowerHA SystemMirror cluster, PowerVS allows selecting a different server using Colocation Rules. PowerVS does not provide access to the HMC, 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 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 disaster recovery mechanisms between two IBM i virtual server instances can be implemented by using IBM PowerHA geographic mirroring. Disaster recovery solutions for Linux workloads can be done using various application and database replication technologies.

Figure 3-14 illustrates the major HA and DR options for PowerVS servers.

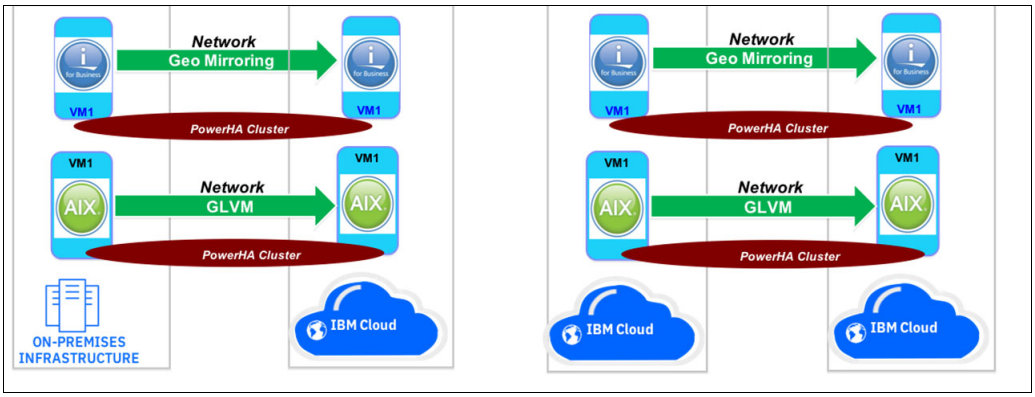


Figure 3-14 HA and DR options for PowerVS

IBM Power Systems Virtual Server also offers Global Replication Service (GRS), which provides asynchronous data replication between two regions. GRS is a valuable feature for disaster recovery as a copy of your data is kept in consistent form in a second IBM Power Systems Virtual Server site that is hundreds or thousands of km 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 replicated to the secondary site, rather than continuously. This requires less network bandwidth, is less costly and has less impact on the active volumes.

Backup and Recovery

The PowerVS configuration and data are not backed up automatically. However, your PowerVS instance can be backed up and restored from a Cloud Object Storage bucket. Any y 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 (ICC) 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 API for Data Protection (OADP).OADP backs up and restores Kubernetes resources and internal images, at the granularity of a namespace, by using Velero 1.7. OADP backs up and restores persistent volumes (PVs) by using snapshots or Restic.

3.2.2 Capabilities

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

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

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

While provisioning PowerVS instance, the user can specify number of cores, amount of memory, network interfaces, and data volume size and type. PowerVS processors can be either dedicated or shared (capped or uncapped). 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. Besides the stock AIX and IBM i images, the clients can always bring their own custom AIX, IBM i or Linux image that has been tested and deployed.

PowerVS instance can support SAP NetWeaver applications on versions of the IBM-provided AIX or Linux stock operating system images. For the 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. The Red Hat OpenShift Cluster on PowerVS is supported as well. To support easier installation, the IBM provides automation to create the entire cluster of servers and install Red Hat OpenShift.

PowerVS instances run in a multi-tenant environment. Dedicated processors provide the best overall performance. Shared uncapped processors are slightly flexibility 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 used in a month.

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 Systems environment, Oracle database in IBM PowerVS, as well as cloud native development and application modernization by using Red Hat OpenShift on PowerVS.

3.2.3 Ecosystem

For enterprise customers, the IBM Power Systems are an important tool and enterprise customers have been consistently looking for options to run IBM Power Systems in the cloud due to IBM Power Systems ability to support high performance and mission critical workloads such as SAP applications and Oracle databases. IBM Power Systems VMs are available on other clouds besides 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 Power Systems VMs to host Linux, AIX, and IBM i workloads. Additionally, 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 Microsoft Azure's global cloud infrastructure. Google Cloud also offers IBM Power Systems as a service on Google Cloud for running AIX, IBM i, or Linux on IBM Power Systems.

Several services and products are available to accelerate adoption and migration of workloads to IBM Power Systems VMs 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 Systems environments on-premises.

3.3 Components

The following sections explain technologies which are either built into the IBM Power Systems servers or can be implemented as part of 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 via software defined manner.

Software defined resource types help in automation, scaling and flexibility, however it changes the way development, application and operation team works together. This change of work responsibility started with virtualization, especially as many of IBM PowerVM’s 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 IBM Power Systems platform team. When working with virtual networking in IBM PowerVM and configuring Shared Ethernet Adapters and virtual switches, it required that the platform operation team learn about VLANs, switches. and Software Defined Storage. Software Defined Networking features will continue this evolution.

3.3.1 Software defined Storage

IBM provides the following Software Defined Storage solutions for Red Hat OpenShift on IBM Power Systems. 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
  - NFS via IBM Storage Scale - Cluster Export Services without dynamic provisioning
- Block
  - IBM Spectrum Virtualize and IBM DS8000® (CSI)

  - Red Hat OpenShift Data Foundation - CephRBD
- Object
  - IBM Cloud Object Storage
  - Red Hat OpenShift Data Foundation - NooBaa

File access type provides Read Write Many (RWX) mode while block type provides Read Write Once (RWO) mode to access the storage in Red Hat OpenShift PODs. See Chapter 2., “Performance and tuning”Figure 2-2 on page 39 for recommended uses of Block, File and Object in your Red Hat OpenShift environment.

Above this, other providers also provide software defined storage solution and there is the widely used NFS protocol to mount exported filesystems from remote servers. NFS could 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 we can use the solution on this repository: https://github.com/kubernetes-sigs/nfs-subdir-external-provisioner.

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

1. Red Hat OpenShift Data Foundation which is discussed in “Red Hat OpenShift Data Foundation (ODF)” on page 85.
2. IBM Storage Scale which is described starting in “IBM Storage Scale (previously IBM Spectrum Scale)” on page 87.

Container Storage Interface (CSI)

Managing container storage in different container orchestration systems like Kubernetes and Red Hat OpenShift is usually done via a standard API specification called CSI. This enables managing storage volumes without the need to know how exactly the underlying storage infrastructure works.

Red Hat OpenShift Data Foundation (ODF)

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

ODF as a storage cluster can be implemented in several ways. The basic building blocks of an ODF cluster is a storage node. For availability we have to 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.

ODF can be installed on IBM Power Systems server based Red Hat OpenShift clusters in internal mode, so the storage nodes will use SSD storage devices assigned to the virtual machines (LPARs). These SSDs are accessed via a storage class provided by the Local Storage Operator. Currently SSDs with capacities of 4TB or less are supported, and we can have maximum 9 devices per storage node. The architecture of ODF is shown in Figure 3-15 on page 86.

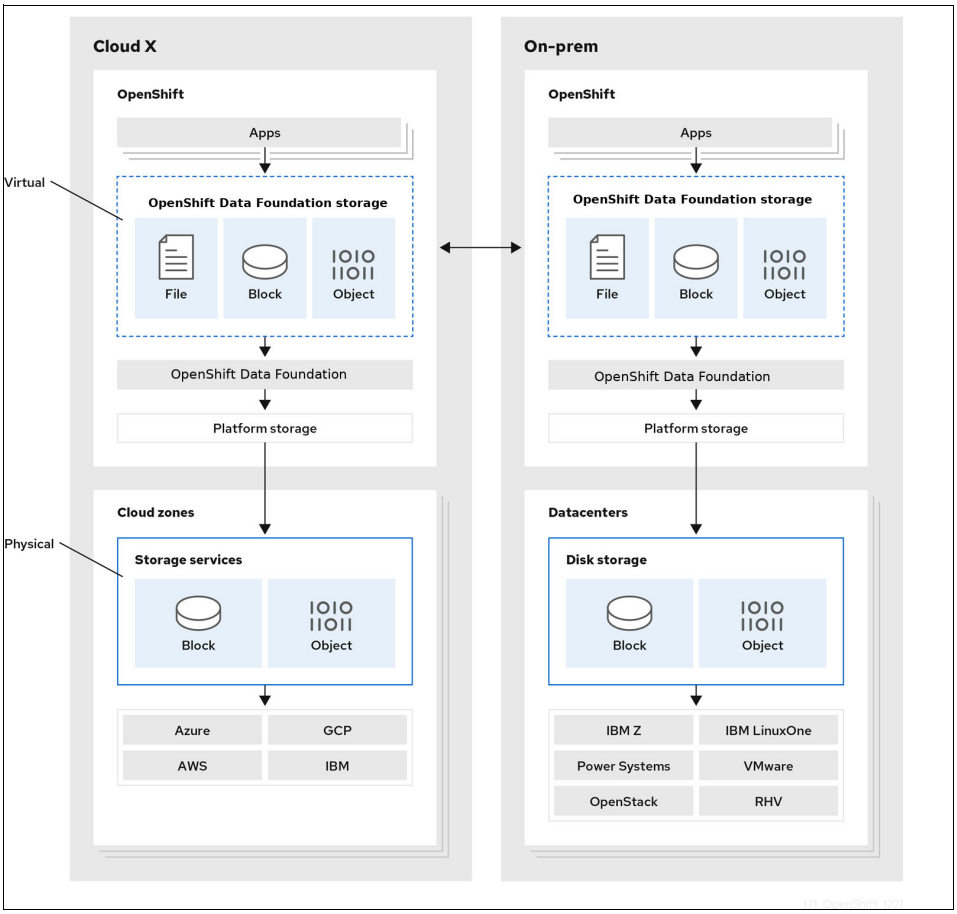


Figure 3-15 Red Hat OpenShift Data Foundation architecture

On IBM Power Systems Power10 servers we 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 then be used as the storage backing for Red Hat ODF.

Table 3-4 NVMe drives per IBM Power Systems Power10 server types

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

After a successful deployment of ODF we will have the following storage classes:

- ocs-storagecluster-ceph-rbd: block based storage class for RWO and RWX mode and RWO mode for filesystem volume mode access of CephRBDs.
- ocs-storagecluster-cephfs: file based storage class for RWO and RWX mode.

- openshift-storage.noobaa.io: storage class for Object Buckets, 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 based on Ceph Object Gateway native object storage interface.

When using the object gateway the endpoint will determine which storage class and interface will handle 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 is used nearly 20 years in demanding enterprise environments.

Key features of IBM Storage Scale:

- Connect applications via providing unified data fabric and single namespace.
- Access data independently of underlying storage technology.
- Scalability and optimization.
- Policy engine and active file management (AFM).
- Increase security via encryption immutability.
- Eliminate data loss and data corruption.
- Integration with backup solutions.
- Single point of management with an intuitive graphical user interface (GUI).
- High performance S3 interface.

IBM Storage Scale can be implemented as a storage cluster in which locally or Storage Area Network (SAN) attached storage devices can be 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 a containerized mode as well.

In a IBM Power Systems system based environment, NVMe drives can be used as locally attached disks for the best performance. The NVMe drives can be attached to a virtual machine (LPAR in IBM Power Systems servers) in pairs or one by one as was shown in Table 3-4 on page 86.

Container Native access of data stored in IBM Storage Scale

IBM Storage Scale container native is a containerized version of IBM Storage Scale. To provide container native access for data on IBM Storage Scale cluster a containerized IBM Storage Scale cluster has to be installed in the Red Hat OpenShift cluster. This cluster and its nodes do not have local storage attached, but enable the access of remote data via IBM Storage Scale remote mount option.

Note: Check the documentation for architecture specific prerequisites before installing CNSA.

Applications can access remote data as persistent volumes via the IBM Storage Scale CSI driver. See the architecture in Figure 3-16.

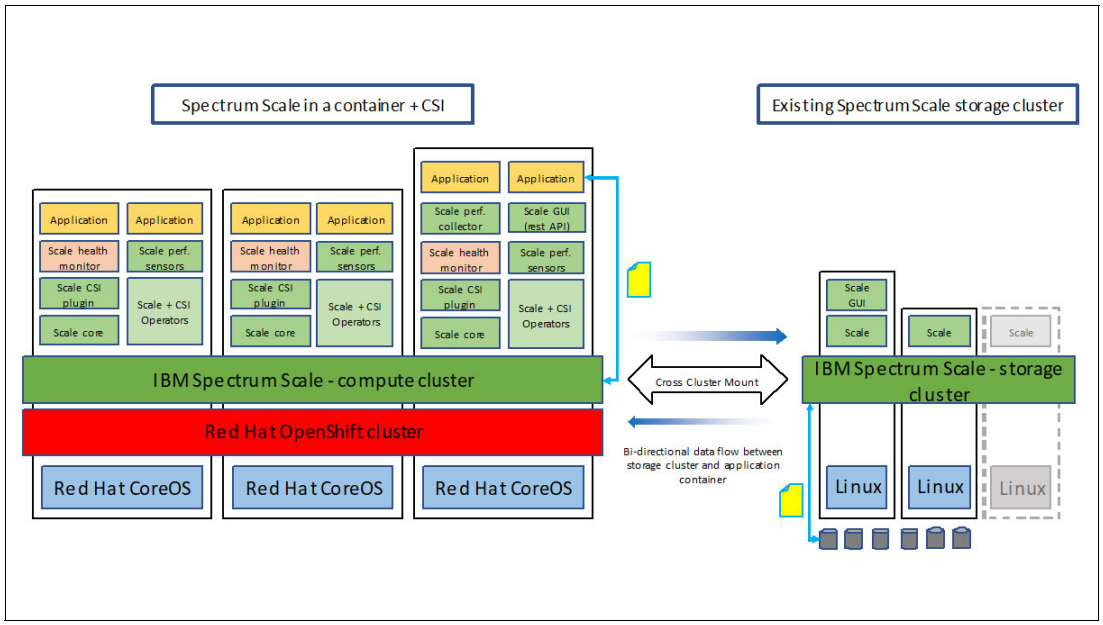


Figure 3-16 Container Native Storage Access architecture

After installing CNSA on an Red Hat OpenShift cluster we can see default Storage Classes. Example 3-1 shows the default Storage Classes when CNSA and NFS client storage provisioner is installed.

Example 3-1 List Storage Classes in Red Hat OpenShift

```
(py39) [root@build-cp4d-1 ~]# oc get sc
NAME                                PROVISIONER                    RECLAIMPOLICY
VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
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 created at CNSA configuration time to allocate PVs and PVCs or create IBM Storage Scale fileset based or lightweight (directory based) Storage Classes.

- For fileset based PVCs IBM Storage Scale will create a separate fileset for each PVCs and the underlying PV.
- For lightweight volumes a directory has to be created in the remotely mounted IBM Storage Scale file system and the CSI driver will create directories for the PVCs under that directory. The lightweight volumes are not mounted to the containers the same ways as fileset based volumes as shown in Example 3-2 on page 89.

Example 3-2 Attached fileset 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 /testpvclw1
drwxrws--x. 2 root 1000760000 4096 Nov  9 13:34 /testpvclw2
```

Note: Since lightweight volume does not enforce quota, it can grow beyond defined size, which may result in consuming whole file system. To avoid this, you must manually create or use an existing fileset to host the lightweight PVC volumes. This can be done by specifying the directory inside fileset for volDirBasePath option.

Moving PODs between nodes with PVCs attached

Persistent Volumes and Persistent Volume Claims can be used in read write many and read write once mode. Read write many mode enables mounting volumes from multiple nodes at the same time, while read write once will not allow this. This has to be taken in account even if we will use it from one POD but we want to move the POD to a new node, which means changing the deployment configuration.

Kubernetes provides a configuration option for deployments defining what happens when we are upgrading the configuration. This 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 RollingUpdate strategy new PODs are started before the old ones are stopped and this can cause a problem, when we are using PVCs in RWO mode.

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

Example 3-4 Moving a POD with 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 trying to move a POD from one node to another using RWO volume. This will fail and the new POD remains in Pending state as Red Hat OpenShift can not attach the volume to it. In this case we have to 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 with 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 between nodes can be necessary for performance tuning, maintenance and spread the load between nodes.

Monitoring throughput of PVCs on IBM Storage Scale

We have several ways to monitor the IBM Storage Scale throughput, but have to keep in mind that CNSA based IBM Storage Scale volumes are network attached volumes. Because of this we will not see local disk load on the nodes where the application PODs are running, but the network traffic will increase.

The following examples show Red Hat OpenShift and IBM Storage Scale based monitoring screen shots to check the load generated by **dd** commands.

Figure 3-17 on page 92 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*.

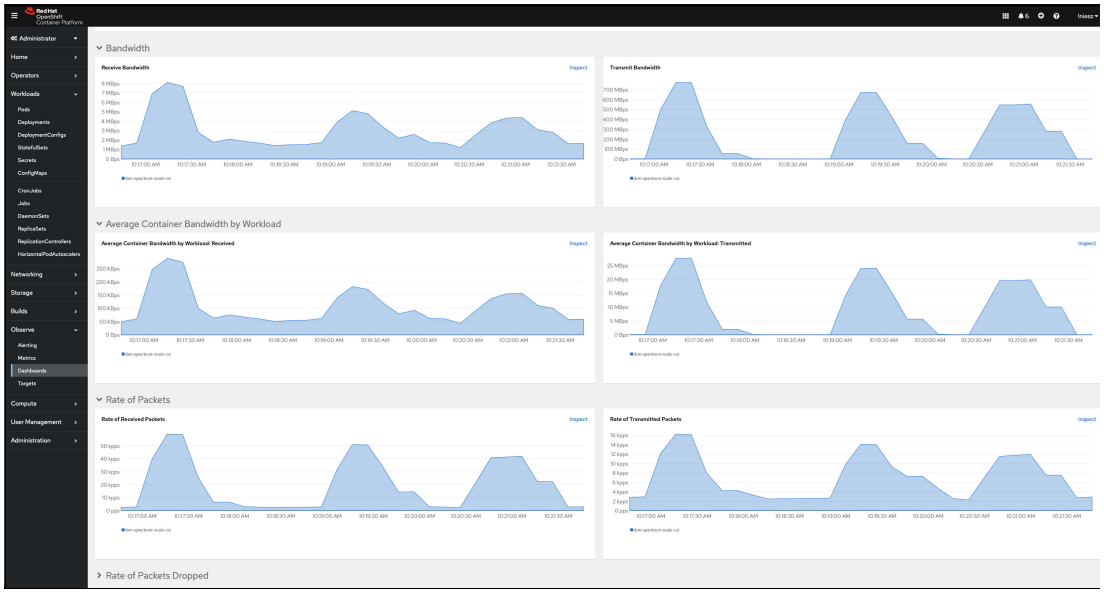


Figure 3-17 Network traffic increased on nodes where CNSA

We can check the metric *node_network_transmit_byte_excluding_lo* for the IBM Storage Scale nodes as well as shown in Figure 3-18.

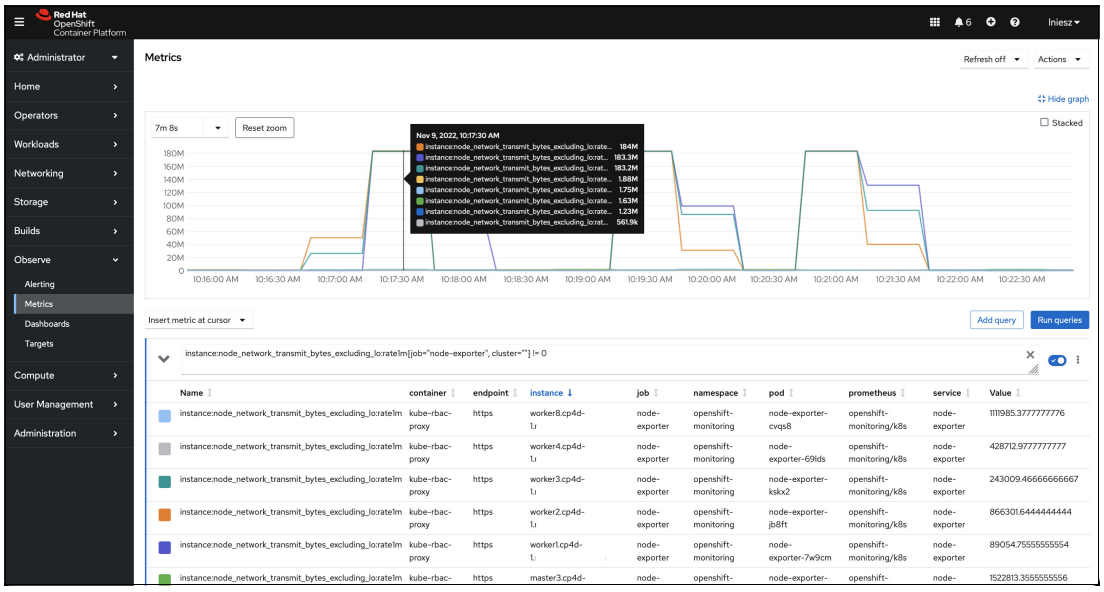


Figure 3-18 IBM Storage Scale node transmit metrics in Red Hat OpenShift Observability

IBM Storage Scale on Red Hat OpenShift also has a GUI which is accessible through the auto created Red Hat OpenShift route: *ibm-spectrum-scale-gui*. This user interface has a monitoring menu in which we can see dashboards, statistics, events, thresholds and audit logs.

In Figure 3-19 we show the Client throughput to disk statistics.

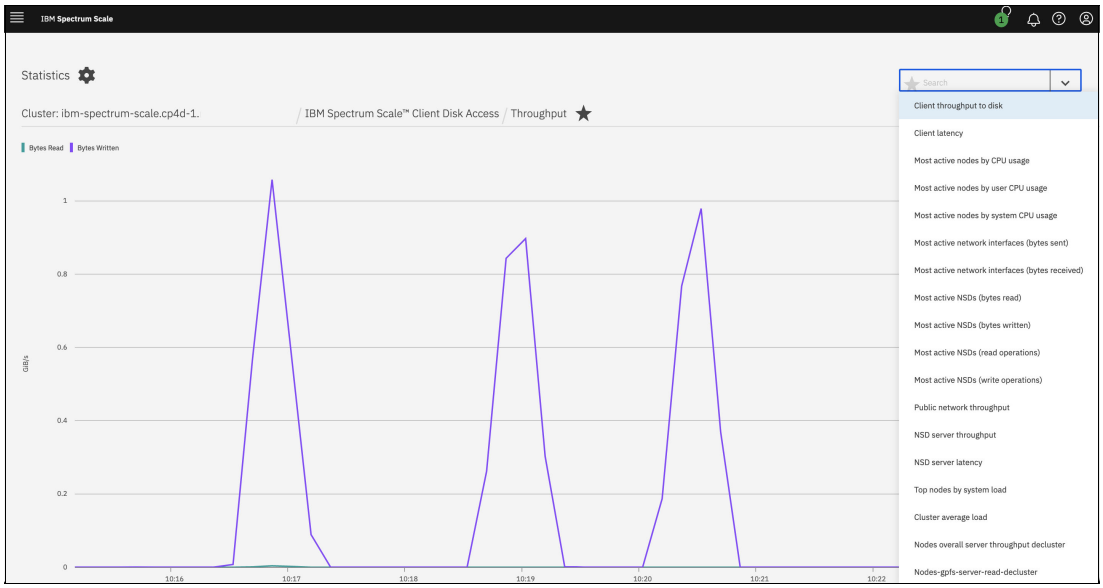


Figure 3-19 Client throughput to disk while dd in Red Hat OpenShift POD

IBM Storage Scale Data Access Services

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.

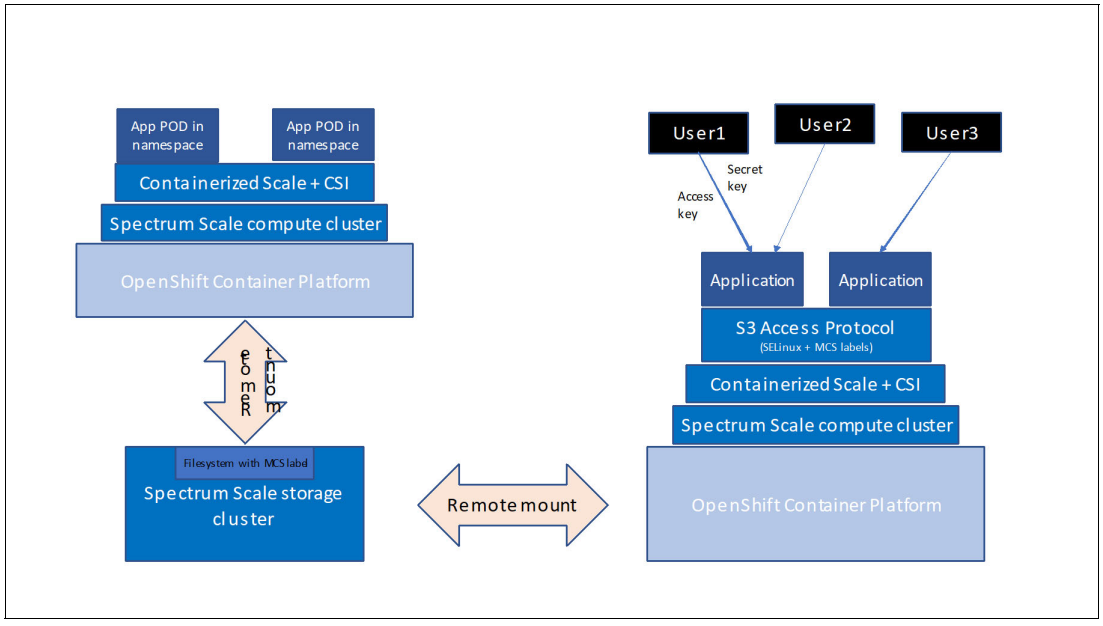


Figure 3-20 CSI + DAS cluster

IBM Storage Scale enables the access of the same data both as objects using the S3 protocol and as normal file system files using containerized applications at the same time.

Note: The Data Access Service cluster has to be a dedicated CNSA cluster on x86_64 based bare metal servers and the remotely mounted IBM Storage Scale cluster must be based on IBM Elastic Storage® Systems (ESS).

IBM Storage Scale DAS includes an embedded license for Red Hat OpenShift Data Foundation. The IBM Storage Scale DAS operator therefore implicitly deploys Red Hat OpenShift Data Foundation. The use of Red Hat OpenShift Data Foundation is limited to the features that can be configured with the IBM Storage Scale DAS management interfaces.

See the following document here in which IBM published benchmark results for IBM Storage Scale DAS.

COSBench using objects with a size of 1 GB running against a three-node IBM Storage Scale DAS cluster and using IBM Elastic Storage System 3200 as the back-end storage:

- More than 60 GB/s aggregated throughput for read workloads.
- More than 20 GB/s aggregated throughput for write workloads.

3.3.2 Software defined networking

Software-defined networking (SDN) is the decoupling of the network control logic from the devices performing the function, such as routers, which control the movement of information in the underlying network. This approach simplifies the management of infrastructure, which may be specific to one organization or partitioned to be shared among several[7].

SDN features controllers that overlay above the network hardware in the cloud or on-premises, offering policy-based management. Technically speaking, the network control plane and forwarding plane are separated from the data plane (or underlying infrastructure), enabling the organization to program network control directly. This differs significantly from traditional data center environments. In a traditional environment, a router or switch – whether in the cloud or physically in the data center – will only be aware of the status of network devices adjacent to it. With SDN, the intelligence is centralized and prolific; it can view and control everything.

In SDN we have three main components:

- Applications, which needs information about the network capabilities and requests resources from the network.
- SDN controllers that communicates with the application and determines the destination of data packets, and also plays load balancing roles.
- Networking devices which are controlled by the controllers to route the traffic.

SDN works well in virtualized environment as well and enables policy-based network management.

SDN types:

- Open SDN uses open protocol to manage virtual and physical devices.
- API SDN are API based solution.
- Overlay Model SDN creates virtual network over existing physical or virtualized network devices providing tunnels for traffic channels.
- Hybrid Model SDN combines traditional networking with SDN features allowing the optimal protocol for each type of traffic.

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

[7] https://www.ibm.com/cloud/blog/software-defined-networking

IBM PowerVM virtual Ethernet in CoreOS

IBM PowerVM provides hypervisor-based network virtualization. The client virtual machines (LPARs) will see virtual Ethernet adapters similar way as in bare metal configurations.

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

Example 3-6 Virtual Ethernet adapter in CoreOS

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

path:
   parent="vio"
   physloc="U9080.HEX.785EDA8-V188-C2-T0"
   connection="30000002"
…
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
```

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

Example 3-7 nmcli command

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

The command "**nmcli device show**" shows detailed configuration of the physical or virtual Ethernet device and the dynamic SDN configuration based on Open vSwitch, which is the default Software-defined networking 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 Machine Config Operator using MachineConfig custom resources. The network and SDN related configuration is placed in the *00-master* and *00-work* MachineConfigs for master and worker nodes. At install time and when a new MachineConfig resource is created the operator will combine all configuration into a rendered configuration and the nodes are restarted with this new configuration, so changing an existing or creating a new config can result application outages as the PODs will not be available while the nodes are rebooting.

Open vSwitch

Open vSwitch (OVS) is a multilayer software switch licensed under the open source Apache 2 license[8].

OVS is well suited to function as a virtual switch in VM environments. In addition to exposing standard control and visibility interfaces to the virtual networking layer, it was designed to support distribution across multiple physical servers. OVS using Virtual extensible Local Area Network (VXLAN) technology is an overlay network to transport L2 network over an existing L3 network. See details at https://www.rfc-editor.org/rfc/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 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 binaries, 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]
```

[8] https://github.com/openvswitch/ovs

```
Description=Open vSwitch
Before=network.target network.service
After=network-pre.target ovsdb-server.service ovs-vswitchd.service
PartOf=network.target
Requires=ovsdb-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
```

The command to check OVS configuration on Red Hat OpenShift nodes is **`ovs-vsctl`**.

Red Hat OpenShift and Open vSwitch

Red Hat OpenShift Container Platform uses a software-defined networking (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 using OVS.

The Red Hat OpenShift SDN uses Open vSwitch, virtual extensible LAN (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 Geneve (Generic Network Virtualization Encapsulation) 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. This allows for all PODs behind a service to 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 leads to increased CPU utilization. Both these outer- and inner-packets are subject to normal checksumming rules to guarantee data is not corrupted during transit. Depending on CPU performance, this additional processing overhead can cause a reduction in throughput and increased latency when compared to traditional, non-overlay networks.

Optimizing networking

We can use VXLAN-offload capable network adapters which moves the packet checksum calculation and associated CPU overhead off of the system CPU and onto dedicated hardware on the network adapter, which could enable pushing network throughput over Gbps.

Another way of optimization is to increase the maximum transmission units (MTU). This must be done via the whole traffic route of network packets:

- Physical Ethernet adapter ports in IBM Power Systems server.
- Etherchannel or Link Aggregation adapter on VIO Server if used.
- Shared Ethernet Adapter on VIO Server.
- VIO Server virtual Ethernet adapter.
- LPAR / 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 here.

Using IPsec can further decrease performance as it can prevent using some network interface acceleration features like NIC offloading.

OVS related logs can be viewed as shown in Example 3-9.

Example 3-9 List OVS logs on 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 binaries, 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 vethbdfd1b7e 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 Input Output operations per second

Input output 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 are not always directly related to a user transaction as each transaction can result in one to many interactions with the storage device. Applications utilizing databases often require more IOPS than simple web applications, so understanding your application is also 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 will be an important factor as you decide what type of storage to provision for the applications and services you are running. Storage can generally be characterized by Tiers - see section 3.3.4, “Tier1 and Tier3 storage” for a discussion of storage tiers. If you have higher IOPS requirements, you should consider utilizing Tier1 storage to provide a good user experience. Lower IOPS requirements can often be satisfied by lower storage tiers.

3.3.4 Tier1 and Tier3 storage

The type of storage that is used to back the Persistent Volume Claims (PVCs) in your cluster will heavily influence the performance of the cluster and the experience that your user have with the applications running there. The different performance characteristics of the available storage technologies are often used to provide generic “Tiers” of storage.

Tiers are a broad methodology of defining the capabilities of your storage and are not standardized across the industry, but in general Tier1 represents the highest performance devices and Tier3 represents the lower lever performance devices. Tier1 technology tends to be more expensive to procure and operate and are usually lower in capacity - this 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 due to your being willing to accept a lower level of service for that application.

Storage - both internal and external - 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 discussed above, but you can be assured that devices with lower performance capabilities will be used to provide storage in the lower Storage tiers.

Storage can be provided by spinning disk devices which tend to be relatively slow as the device has to wait for the spinning platter inside to rotate to the correct location of the data needed. Newer technology for storage includes solid state devices where there are no moving parts so these devices can provide lower latency and better performance than the old spinning disks, they also tend to be more reliable since they do not rely on moving parts to operate. Even in SSD storage devices there are a range of capabilities and connection types that differentiate the performance of those devices.

As you design your cluster, take into account the user requirements of each of the applications/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 match the cost vs performance that you need to provide in to your user.

Fiber channel is technology that can allow you to efficiently provide storage in a shared environment and may allow you to reduce the cost of storage by sharing across multiple clusters. Fiber channel is discussed in section 3.3.5, “Fiber channel”.

3.3.5 Fiber channel

Fiber channel (FC) is a high speed storage area network connection that is used for connecting storage, disk and tape, to the processors. The FC specification provides for line speeds from 1 Gb to the current generation 64 Gb with the planned introduction of 128 Gb in the near future.

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

Fiber channel protocol provides for a switched environment (storage area network or SAN), allowing the sharing of ports with multiple devices and allowing flexible distances between connected devices. This also allows sharing of devices between different processors for easy scalability and migration capability. A fiber channel network can provide very low latency connections to shared storage – the latencies are often as low or lower than direct connected

disk in your servers. By providing the option of connecting to external storage, FC provides an additional layer of flexibility and availability and can move some of the processing power required for data replication out of the processor into the SAN storage – releasing that processing power to be better used by your applications.

FC block devices can be connected to your Red Hat OpenShift cluster through the use of the CSI driver provided in Red Hat OpenShift and supported by the storage devices. Using the CSI driver allows you to take advantage of the flexibility and availability features of the SAN storage seamlessly in your Red Hat OpenShift cluster.

3.3.6 Network File System

Network File System (NFS) is a mechanism for sharing files across multiple servers/clusters across a network connection. NFS is generally a low cost solution for sharing files between different applications as it does not require special connections to the 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 normally significantly less than any class of storage (see 3.3.3, “Input Output operations per second” and 3.3.4, “Tier1 and Tier3 storage” for a discussion of storage tiers and requirements) and should be used with caution for high volume applications.

3.3.7 Network

The capability of your network connections is also a significant contributor to the experience that your users have higher network speeds provide lower latency and can support more users. However, higher network speeds are more expensive to procure and operate.

Choose network connections that allow you to meet your user’s expectations. It is also important that you monitor network utilizations as you scale users and applications and be prepared to add additional network capacity as needed.

Technologies like SR-IOV provided by IBM Power Systems servers can provide flexibility in designing your network connectivity. SR-IOV is discussed in section 3.3.8, “Single Root - I/O Virtualization (SR-IOV)”.

3.3.8 Single Root - I/O Virtualization (SR-IOV)

Single root I/O virtualization (SR-IOV) allows multiple virtual servers and the running operating systems on them to 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 and is a successor of the proprietary solution named Integrated Virtual Ethernet adapter (IVE) which was available in some of the IBM Power7 servers. SR-IOV is a newer technology based on the physical adapters capability, but in many IBM Power Systems processor-based servers the IBM PowerVM provided Shared Ethernet Adapter is the standard way of virtualizing network adapters.

It is also possible to leverage SR-IOV technology via VIO servers in which case client partitions are using virtual Network Interface Controllers (vNIC) based on SR-IOV capable adapters configured in VIO servers.

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

Table 3-5 Ethernet network virtualization technologies

Technology	Live Partition Mobility	Quality of service (QoS)	Direct access performance.	Redundancy Options	Server Side Failover	Requires VIOS
SR-IOV	No[1]	Yes	Yes	Yes[2]	No	No
vNIC	Yes	Yes	No[3]	Yes[2]	vNIC Failover	Yes
SEA / vEth	Yes	No	No	Yes	SEA Failover	Yes
Hybrid Network Virtualization	Yes	Yes	Yes	Yes	No	No

Notes:

1. SR-IOV can optionally be combined with VIOS and virtual Ethernet to use higher-level virtualization functions like Live Partition Mobility (LPM); however the client partition will not receive the performance or QoS benefit.
2. Some limitations apply. See FAQ on link aggregation.
3. Generally better performance and requires fewer system resources when compared to SEA/virtual Ethernet.

See *IBM Power Systems SR-IOV: Technical Overview and Introduction*, REDP-5065 for more details on SR-IOV in IBM Power Systems servers. Also see this FAQ document for additional details.

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 Systems servers SR-IOV provides QoS controls to set the desired capacity values for each logical ports, which allow prioritization of partition traffic.

Sharing physical adapters can reduce the cost due to consolidation and there is no additional CPU and memory usage comparing to shared Ethernet adapters.

Red Hat OpenShift and SR-IOV

SR-IOV is supported on IBM Power System based Red Hat OpenShift clusters on specific physical network interfaces[9]. In IBM Power System servers with Red Hat OpenShift running on them there are multiple ways to use SR-IOV.

- One way is having a bare metal server as an Red Hat OpenShift node and share the SR-IOV capable PCI adapter between PODs.
- The other way is to have a VIO server and configure virtual servers (LPARs) as OpenShift nodes using vNICs, which are configured on shared SR-IOV capable physical adapters set up in VIO servers.

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

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

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

Red Hat OpenShift also supports Ethernet and Infiniband device attachment using SR-IOV.

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

- Discovers the SR-IOV network devices available in nodes.
- Generates *NetworkAttachementDefinition* custom resources (CR) for SR-IOV Container Network Interface (CNI).
- Creates and updates the configuration of the SR-IOV network device plugin.
- Creates node specific *SriovNetworkNodeState* CR.
- Updates the spec.interfaces field in each *SriovNetworkNodeState* CR.

There is a daemonset deployed to every worker node by the operator to discover and initialize the SR-IOV network devices. Additional plugins discover, advertise and allocate virtual function (VF) resources into PODs.

For installation and initial configuration see the Red Hat OpenShift documentation.

After installing the operator, we can configure whether it will drain the nodes and inject the configuration automatically.

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

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

```
(py39) [root@build-cp4d-1 ~]# oc get SriovNetworkNodeState -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 SriovNetworkNodeState worker1.cp4d-1.rtp.raleigh.ibm.com -n
openshift-sriov-network-operator -o yaml
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodeState
metadata:
  creationTimestamp: "2022-11-15T08:58:39Z"
  generation: 1
  name: worker1.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: "33330348"
  uid: 2a6d0ed4-9818-42ef-8049-eea2bbfa08f1
spec:
  dpConfigVersion: "33329739"
status: {}

(py39) [root@build-cp4d-1 ~]# oc get SriovNetworkNodeState worker8.cp4d-1.rtp.raleigh.ibm.com -n
openshift-sriov-network-operator -o yaml
apiVersion: sriovnetwork.openshift.io/v1
kind: SriovNetworkNodeState
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 102 shows: if the node is not supported for SR-IOV, then there are no interfaces listed in the *Status* section. The operator will set the following label for supported nodes: "*feature.node.kubernetes.io/network-sriov.capable: true*".

The actual configuration of virtual functions to PODs are controlled by the *SriovNetworkNodePolicy* custom resource.

Note: When a configuration is applied to a *SriovNetworkNodePolicy* CR the nodes can be drained and rebooted depending on the *SriovOperatorConfig* custom resource.

With this custom resource we can select on what nodes what type of adapters or specifically which adapters will be configured for SR-IOV. Here we can set if remote direct memory access (RDMA) will be enabled or not. There is a parameter resourceName also set, which will help attaching the later created *SriovNetwork* and POD to the virtual functions of SR-IOV.

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

With the creation of *SriovNetwork* custom resource we will 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 will create 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 Add POD to additional networks

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

Red Hat OpenShift will create an annotation: *k8s.v1.cni.cncf.io/network-status* based on the additional network configuration we set.

3.3.9 Partition mobility

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

The following types of migrations are available.

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

Use cases for partition mobility

Partition mobility provides systems management flexibility and is designed to improve system availability. Some examples in which partition mobility can help:

- Server consolidation: migrate and consolidate partitions from many servers to less with higher capacity.
- Workload balancing: distribute partition between servers to share and balance the load, or migrate partition to server with specific HW resource needed be applications.
- Evacuating servers for planned maintenance: as preparation for HW, firmware or VIOS upgrades or changes, which would require outage, the partitions can be migrated uninterrupted to other servers, then migrate back after successful upgrade, this way avoiding planned outage.
- Migrating from older technology (IBM POWER8 and above) to newer technology, however certain new HW features could require operating system reboots to take advantage.

Note: Partition mobility does not provide automatic workload balancing and is not a replacement of high availability or disaster recovery solution.

Using virtual machine remote restart in case of a failed partitions or complete systems can provide higher availability for our systems and this feature is available in IBM PowerVC. See details about this product in IBM PowerVC Documentation.

Processor compatibility of LPM source and target

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

See the supported scenarios in the IBM Power Systems Documentation.

Prerequisites

The best way to summarize the prerequisites is to tell that both source and target systems must have IBM PowerVM Enterprise Edition license activated and both source and target partitions must be fully virtualized (no physical adapter or interface attached) at the time of the migration.

The target system must provide the same virtual resources as the source system and must be connected to the same network with synchronized Time of Day clocks of the VIO servers. It is also possible to migrate a partition to another system managed by different HMC. The target VIO servers 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 has to be prepared also to be able to allocate the same storage volumes to the migrated partition on the target system. Virtual adapters cannot be marked as required and shouldn't be marked for “any client”. The processor mode must be compatible between the source and target systems. A processor compatibility mode is a value assigned to a

logical partition by the hypervisor that specifies the processor environment in which the logical partition can successfully operate.

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

See the following for active partition migration compatibility mode combinations:

- For active partitions
- For inactive partitions

Additional documentation is found in configuration validation.

Note: If a partition has physical resources attached, then they have to be deallocated before the migration, but they can be added back on the target system, so with manual steps we can create a workaround.

Migration phases

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

Active partition migration phases:

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

Inactive partition migration phases:

1. Validate configuration.
2. Create new LPAR on target server.
3. Create new virtual resources on target server.
4. Remove the old LPAR configuration from the source server.
5. Free up the old resource 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 3 master nodes, which is quite sensitive to the connectivity between the master nodes. This is necessary to keep the etcd cluster in sync. See details about Red Hat OpenShift HA in section 4.4.5, “Disaster recovery” on page 127.

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

4

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/Continuous Delivery (CI/CD) pipelines, and source-to-image build capability. This chapter explores the architecture and layout of a Red Hat OpenShift environment. We will cover the following topics:

4.1 Design considerations for Red Hat OpenShift

Container based computing is an important trend in today's environment and there are many options available when you are choosing the infrastructure to run your cloud native applications. While container based infrastructures are designed to 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.

That's why, Red Hat OpenShift Container Platform architecture planning and design is very important to fulfill your business requirements. In the rest of this chapter we present information intended to inform you of how Red Hat OpenShift and IBM Power Systems servers can be used to implement a cloud solution on one of the best platforms in order to meet your business requirements.sider in designing your Red Hat OpenShift environment.

4.2 Red Hat OpenShift capabilities on IBM Power Systems

Clients are demanding exceptional customer experiences which is driving organizations to develop new applications to meet customer expectations and also modernize existing applications to accelerate their cloud-native journey. DevOps teams require a flexible and agile development approach and 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 Systems provides developers a consistent and secure platform to innovate continuously with skill set that is common across various platforms including IBM Power Systems with additional reliability, adaptability and performance. Red Hat OpenShift on IBM Power Systems offers flexibility and choice for a variety of cloud consumption models across physical, virtual, private, and public clouds. It also provides scalability and takes advantage of the added security built into IBM Power Systems servers to provide highly secure cloud based environments in a hybrid cloud for cloud-native development.

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

- Red Hat OpenShift enables scalability to thousands of instances across hundreds of nodes in seconds, This scalability is enhanced as IBM Power Systems can scale the underlaying infrastructure up and down based on demand.
- With built-in virtualization, IBM Power Systems provides capability to dynamically add or remove memory and CPUs allocated to worker node virtual machines.
- BM Power provides a pay-per-use consumption model in both on-premise and off-premise environments.
- By collocating cloud-native apps with existing VM-based apps running on AIX, IBM i, or Linux environments, IBM Power Systems servers enable low-latency connection between apps and data.
- IBM Power Virtual Server provides support to run leading business applications like SAP HANA in an IBM Power Systems based cloud.

Red Hat OpenShift on IBM Power Systems provides a strong foundation built for security and reliability. For example, live partition mobility can be used to provide uninterrupted access to critical data and applications. The IBM Power Systems compute infrastructure reduces unplanned downtime with less than two minutes per year. This results in several advantages

including improved productivity for IT teams and reducing impact on critical business processes as well as end-users.

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

The built-in agility of Red Hat OpenShift and IBM Power Systems is extended to a truly hybrid cloud model through the IBM Power Virtual Server. IBM PowerVS is an enterprise Infrastructure-as-a-Service offering built around IBM Power Systems servers colocated in IBM Cloud data centers and offering access to over hundreds of IBM Cloud services. Red Hat OpenShift is available on IBM Power Virtual Servers via a platform-agnostic installer.

For workloads on Red Hat OpenShift on IBM Power Systems, the solution offers a lower overall total cost of ownership and greater throughput for SLAs[1].

4.3 IBM Cloud Paks capabilities

Building containerized applications from scratch requires a significant investment in cloud resources, talent and management tools. With a known shortage of cloud-native skills and short project time lines (businesses want solutions delivered yesterday), IBM customers are seeking enterprise-grade and preintegrated software to accelerate digital transformation and innovation. IBM Cloud Paks are AI-powered software designed for the hybrid cloud landscape.

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

IBM Cloud Paks are:

- Portable and can run anywhere: The portability of hybrid cloud solutions built with IBM Cloud Paks means that they are built to run on any hybrid cloud environment. They can run on-premises infrastructure, on public hybrid cloud infrastructure or in an integrated system 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.

IBM Cloud Paks on IBM Power Systems take advantage of 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 designed to accelerate application modernization with preintegrated data, automation and security capabilities.

[1] https://www.ibm.com/downloads/cas/26A6ONY

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

The Paks are the following:

- IBM Cloud Pak® for WebSphere Hybrid Edition
- IBM Cloud Pak for Integration
- IBM Cloud Pak for Watson AIOps
- IBM Cloud Pak for Business Automation
- IBM Cloud Pak for Data.

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

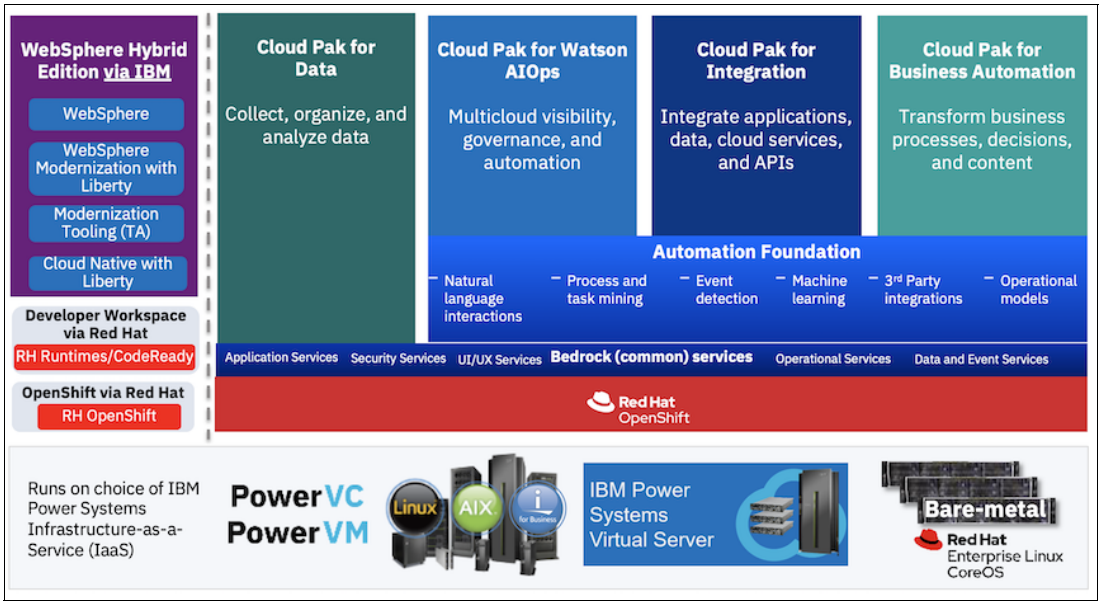


Figure 4-1 Cloud Paks on IBM Power Systems

Table 4-1 lists the IBM Cloud Paks capabilities available on IBM Power Systems.

Table 4-1 IBM CloudPak capabilities available on IBM POWER®

IBM Cloud Pak	Capabilities available on IBM Power Systems
IBM Cloud Pak for Data	IBM Db2® Warehouse
	DB2 Advanced
	Data Management Console
	Watson Machine Learning Accelerator (GPU support)
IBM Cloud Pak for WebSphere Hybrid Edition (WSHE)	Transformation Advisor (tool)
	Mobile Foundation (Traditional)
	WebSphere Application Server
	WebSphere Application Server Liberty
	WebSphere Application Server ND

IBM Cloud Pak	Capabilities available on IBM Power Systems
IBM Cloud Pak for Integration	MQ, MQ Advanced
	App Connect Enterprise
	Platform Navigator
	Event Streams (Kafka)
	App Connect Designer
IBM Cloud Pak for Watson AIOps	RHACM: Manage-to IBM Power Systems
	RHACM: Manage-from IBM Power Systems
	CP4WAIOps: Infra Automation: Manage-to IBM Power Systems
	CP4WAIOps: Infra Automation: Manage-from IBM Power Systems (CP4MCM 2.3)
	Instana: VM observability for AIX, IBM i and Linux
	Instana: Container observability via OCP on IBM Power Systems
	Turbonomic: Manage to IBM Power Systems (containers)
IBM Cloud Pak for Business Automation	ODM: Operational Decision Manager
	BAW: Business Automation Workflow
	FNCM: IBM FileNet® Content Manager
	ER: Enterprise Records
	BAI: Business Automation Insights
	BAS: Business Automation Studio
	AD: Application Designer
	ADS: Automation Decision Services

IBM Cloud Paks are covered in more detail in Chapter 5., “IBM Cloud Paks on Red Hat OpenShift running on IBM Power Systems” on page 159. In addition more information about the IBM Cloud Paks can be found at:

https://www.ibm.com/cloud-paks

https://www.ibm.com/products/cloud-pak-for-integration

https://www.ibm.com/products/cloud-pak-for-watson-aiops

https://www.ibm.com/products/cloud-pak-for-business-automation

https://www.ibm.com/products/cloud-pak-for-data

4.4 Red Hat OpenShift Architecture

This section provides a view of Red Hat OpenShift and its underlying architecture. We will provide an overall view about the its architecture and its underlying components. Running Red Hat OpenShift on IBM Cloud or on premise provides your developers with a fast and secure way to containerize and deploy cloud ready workloads in Kubernetes clusters.

Red Hat OpenShift Container Platform (OCP) is a cloud-based Kubernetes container platform. The foundation of Red Hat OpenShift Container Platform is based on Kubernetes. Red Hat OpenShift Container Platform provides a platform for developing and running containerized applications. It is designed to allow applications and the hosting providers that support them to scale from just a small cluster with a few machines and applications to a cluster of thousands of machines that serve millions of end customers.

With its foundation in 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 in open Red Hat technologies lets you extend your containerized applications beyond a single cloud to on-premise and multi-cloud environments.

Red Hat OpenShift Container Platform provides enterprise-ready enhancements to Kubernetes, including the following:

- Hybrid model cloud deployments: You can deploy Red Hat OpenShift Container Platform clusters to a variety of public cloud platforms or in your data center privately.
- 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 intense testing and certification initiatives for Red Hat's 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.

While 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 on page 113 shows the various components of OPC and how developers and administrators interact with the cluster. There are various components in OCP which all run on the underlying physical or cloud infrastructure. These are divided into the compute plane which provides application support, monitoring, and networking for example, and the control plan which provides management and security among other services.

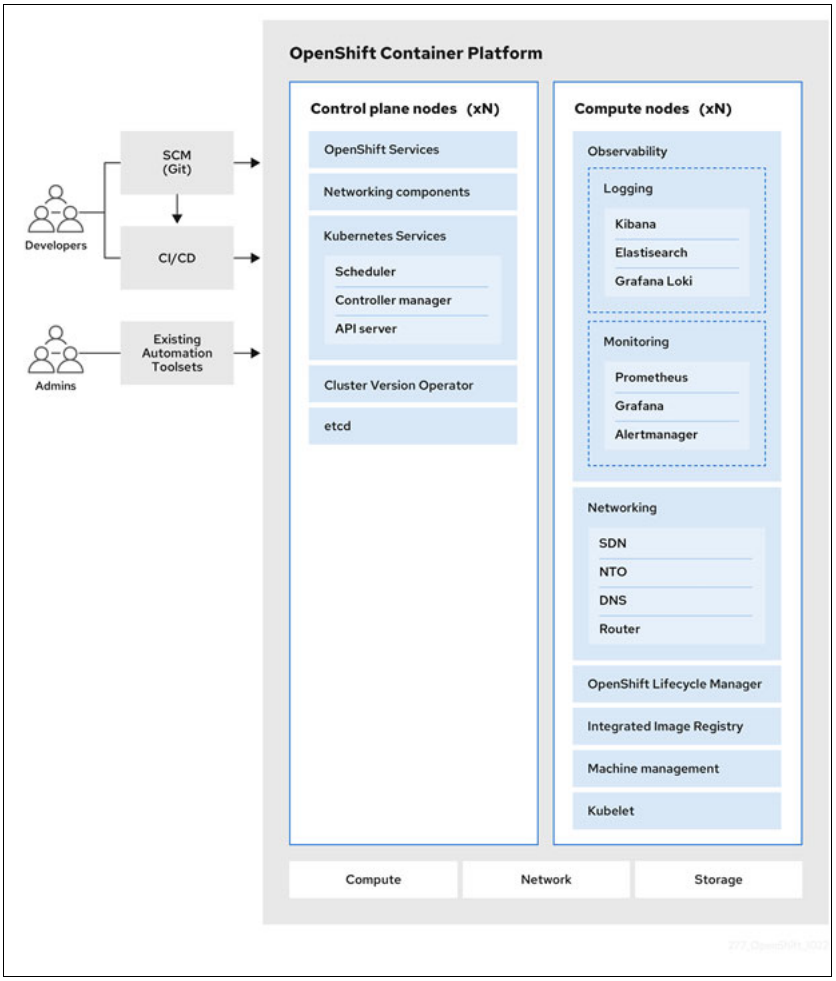


Figure 4-2 Red Hat OpenShift Container Platform architecture

4.4.1 Enterprise Kubernetes

Red Hat OpenShift is Enterprise Kubernetes. A platform that is built by experts in Kubernetes and containers 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, to run 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 fairly 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. Using PODs provides extra metadata with the container and offers the ability to 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 allows 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 at a time. 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-premise adoption. The open source development model allows 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. Where applications were once expected to be installed on operating systems that included all their dependencies, containers let an application carry their dependencies with them. 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 allows 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 carries its own dependent software and manages its own interfaces, such as networking and file systems, so applications never need to 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 also deploy and test a new version of an application alongside the existing version. If the container passes your tests, simply 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. This capability allows you to scale only the required services instead of the entire application, which can allow you to 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 specifically 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:

- Ignition, which Red Hat OpenShift Container Platform uses as a firstboot system configuration for initially bringing up 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 fully

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 of 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 also 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 completely 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 allow simplified, cluster-wide management of those critical components.

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 to 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 CLI (oc) interface.

Red Hat OpenShift Container Platform lifecycle

Figure 4-3 on page 116 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.

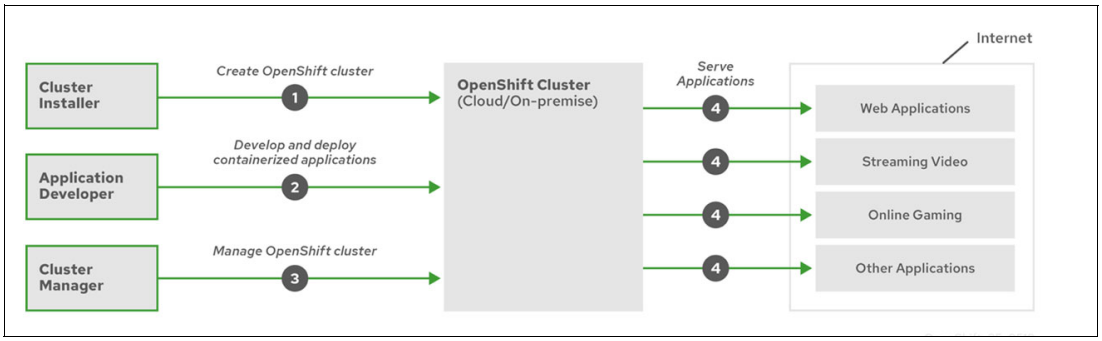


Figure 4-3 Red Hat OpenShift Container Platform Lifecycle

In Red Hat OpenShift on IBM Cloud, your clusters comprise an IBM-managed master that secures components such as the API server and etcd, and customer-managed worker nodes that you configure to run you app workloads, as well as 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 version 4 Components

Review the architecture diagram shown in Figure 4-4 and then scroll through the following tables for a description of master and worker node components in Red Hat OpenShift on IBM Cloud clusters that run version 4 on classic infrastructure.

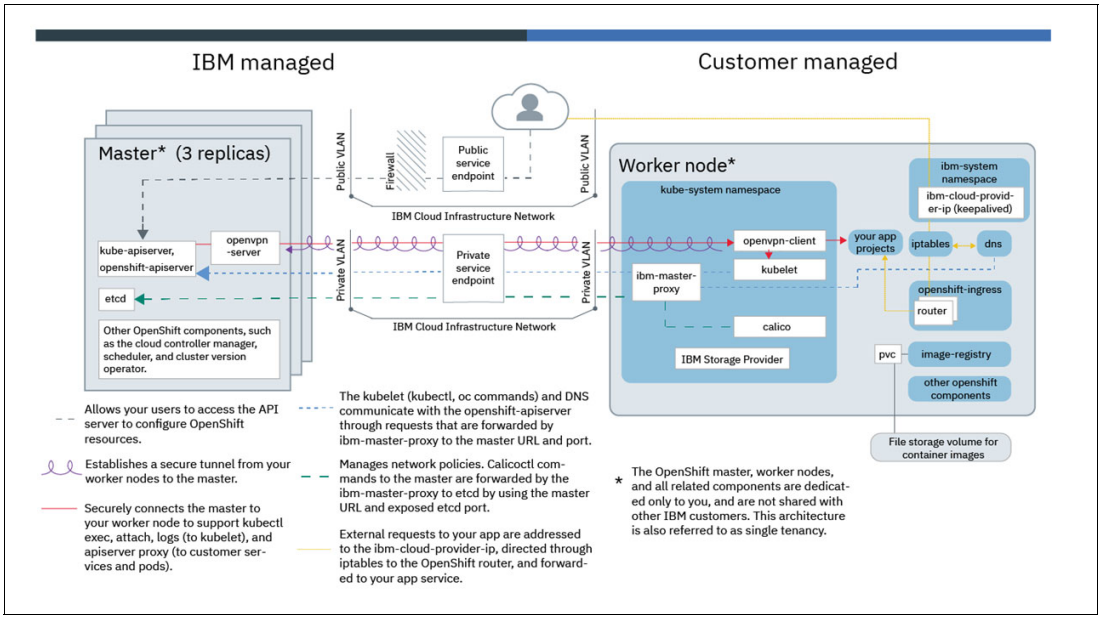


Figure 4-4 Red Hat OpenShift version 4 master components

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 don't 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 version 4 master components

Review the following components in the IBM-managed master of your Red Hat OpenShift on IBM Cloud cluster. You can't modify these components. IBM manages the components and automatically updates them during master patch updates.

In Red Hat OpenShift Container Platform 4, many components are configured by a corresponding operator for ease of management. The following table discusses these operators and components together, to focus on the main functionality the component provides to the cluster.

Master components, including the API server and etcd, have three replicas and are spread across zones for even higher availability. Masters include the same components as described in the Classic cluster architecture for version 4 clusters. The master and all the master components are dedicated only to you, and are not shared with other IBM customers.

Red Hat OpenShift version 4 worker node components

Review the following components in the customer-managed worker nodes of your Red Hat OpenShift on IBM Cloud cluster.

These components run on your worker nodes because you are able to 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 virtual machines 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 console. Unlike classic clusters, you don't see VPC compute worker nodes in your infrastructure portal or separate infrastructure bill, but instead manage all maintenance and billing activity for the worker nodes from Red Hat OpenShift on IBM Cloud.

Worker nodes include the same components as described in the Classic cluster architecture for version 4 clusters. Red Hat OpenShift on IBM Cloud worker nodes run on the Red Hat Enterprise Linux 7 operating system.

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 don't 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.

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 virtual machines, 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, however 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.

Other major components

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 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 are automatically 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 are automatically 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 Public or Private app networking for VPC clusters. 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) could be a good choice for developers to get started building 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 of the tools needed to develop container-based applications.

Red Hat OpenShift Local Components

The latest version of the Red Hat OpenShift Local 2.5 will ship 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 Latest
Red Hat OpenShift client binary (oc)	v4.10.18 or Latest
Podman binary	4.1.0 or Latest

Minimum System Requirements

Red Hat OpenShift Local has the minimum hardware and operating system requirements seen in Table 4-3.

Table 4-3 Minimum System Requirements - Red Hat OpenShift Local

operating system	CPU	memory	Disk Storage	Hardware Architectures
Microsoft Windows 10 Fall Creators Update (version 1709) or later	4	9GB	35GB	AMD64 and Intel 64 (x86_64)
macOS 11 Big Sur or later.	4	9GB	35GB	Intel 64 (x86_64), ARM-based M1 (aarch64)
Red Hat Enterprise Linux/CentOS 7, 8 and 9 minor releases	4	9GB	35GB	AMD64 and Intel 64 (x86_64)

The Podman container runtimes are shown in Table 4-4,

Table 4-4 Minimum System Requirements - Podman

operating system	CPU	memory	Disk Storage	Hardware Architectures
Microsoft Windows 10 Fall Creators Update (version 1709) or later	2	2GB	35GB	AMD64 and Intel 64 (x86_64)
macOS 11 Big Sur or later.	2	2GB	35GB	Intel 64 (x86_64), ARM-based M1 (aarch64)
Red Hat Enterprise Linux/CentOS 7, 8 and 9 minor releases	2	2GB	35GB	AMD64 and Intel 64 (x86_64)

See the official at Getting Started Guide for Red Hat OpenShift Local for the most current information.

Installing Red Hat OpenShift Local

In this example, we will show you how to use Red Hat OpenShift Local in your local system. This can be a laptop or a Virtual Machine (VM) running with Red Hat Enterprise Linux 8.

1. Download the latest release of Red Hat OpenShift Local from the Red Hat download site as shown in Figure 4-5 on page 120.
2. Copy the downloaded file to the ~/ directory and extract the contents of the archive file as shown in Example 4-1.

Example 4-1 Download and extract Red Hat Local install file

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

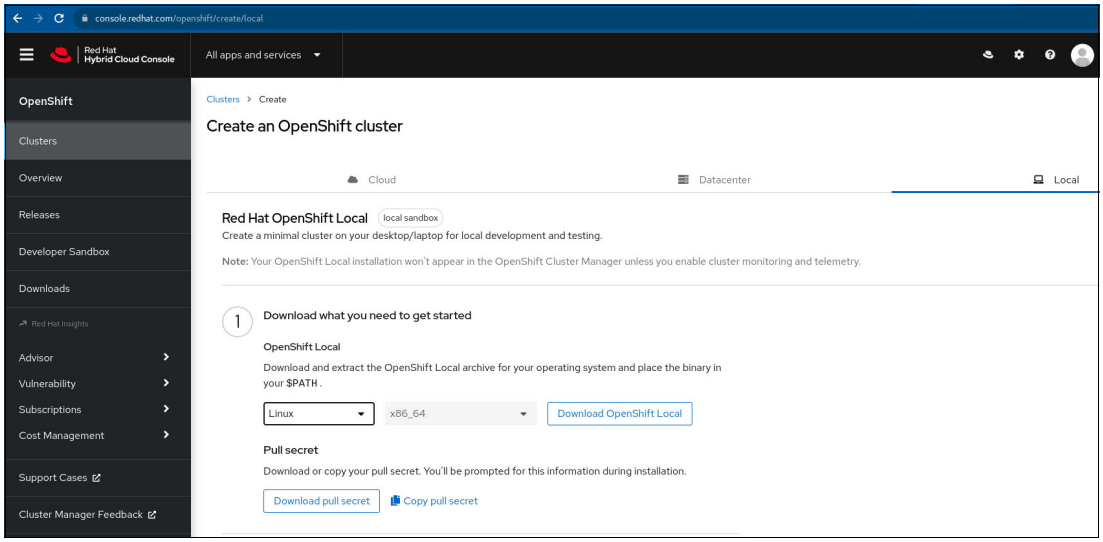


Figure 4-5 Create cluster

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

Example 4-2 Set $PATH variable

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

Initial Setup for the Red Hat OpenShift Local

Before you start the initial setup, make sure your system is configured with a local rpm repository, so that it may install the required rpms as required.

1. The crc setup command performs operations to set up the environment of your system for the Red Hat OpenShift Local instance. This is shown in Example 4-3.

Example 4-3 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 desired 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 run for the first time, the **"crc setup"** command will ask you to enable telemetry for the usage data collection. But you can disable or enable it back at anytime using the ***"crc config set consent-telemetry no"*** or ***"crc config set consent-telemetry yes"*** command. The changes to telemetry consent do not modify a running instance. The change will take effect next time you run the **crc start** command.

Tip: Make sure the DNS client is configured in the file /etc/resolv.conf and is able to resolve the required Red Hat web sites, otherwise you will 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/c
rc_libvirt_4.11.7_amd64.crcbundle": dial tcp: lookup mirror.openshift.com: no such
host
```

2. Example 4-4 shows how to start the Red Hat OpenShift Local instance once the setup completes successfully.

Example 4-4 crc setup command and sample output

```
3.15 GiB / 3.15 GiB
[--------------------------------------------------------------------------------]
100.00% 1.07 MiB p/s
INFO Uncompressing /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
```

Using Red Hat OpenShift Local

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

Example 4-5 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 you run the crc start command, you will be asked to provide the pull secret that you can copy from the same Red Hat download site.

2. Once you start the Red Hat OpenShift Local instance it will provide you the login credentials and the URLs for both command line interface (CLI) or web based graphical user interface (GUI) access at the end. Sample output is shown in Example 4-6.

Example 4-6 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 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' command line interface:
  $ eval $(crc oc-env)
  $ oc login -u developer https://api.crc.testing:6443
```

If the Cluster is not ready you will receive the error messages as 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 you start a Red Hat OpenShift Local instance it may fail because of the predefined waiting time for the Red Hat OpenShift Local instance readiness. The time taken by the system to get your Red Hat OpenShift Local instance ready may take longer than the predefined waiting time depending on your system. You will still be able to login to the Red Hat OpenShift Local instance and verify which particular Cluster Operators are not ready yet.

Log to Red Hat OpenShift Local

You can login to the Red Hat OpenShift Local instance either from command line or from the web browser.

Examples of using the command line to work on your cluster

The following are examples of the commands that can be used in the command line to work on your Red Hat OpenShift Local cluster.

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

Logging in via the command line is shown in Example 4-8.

Example 4-8 Login to Red Hat OpenShift cluster using oc command and sample output

```
$ eval $(crc oc-env)
$ oc login -u kubeadmin https://api.crc.testing:6443
Logged into "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 cluster operators status

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

Example 4-9 Cluster Operator (CO) 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 the your web browser to manage your cluster

You can login to your Red Hat OpenShift Local instance using your web browser connection as shown in Figure 4-6 on page 125.

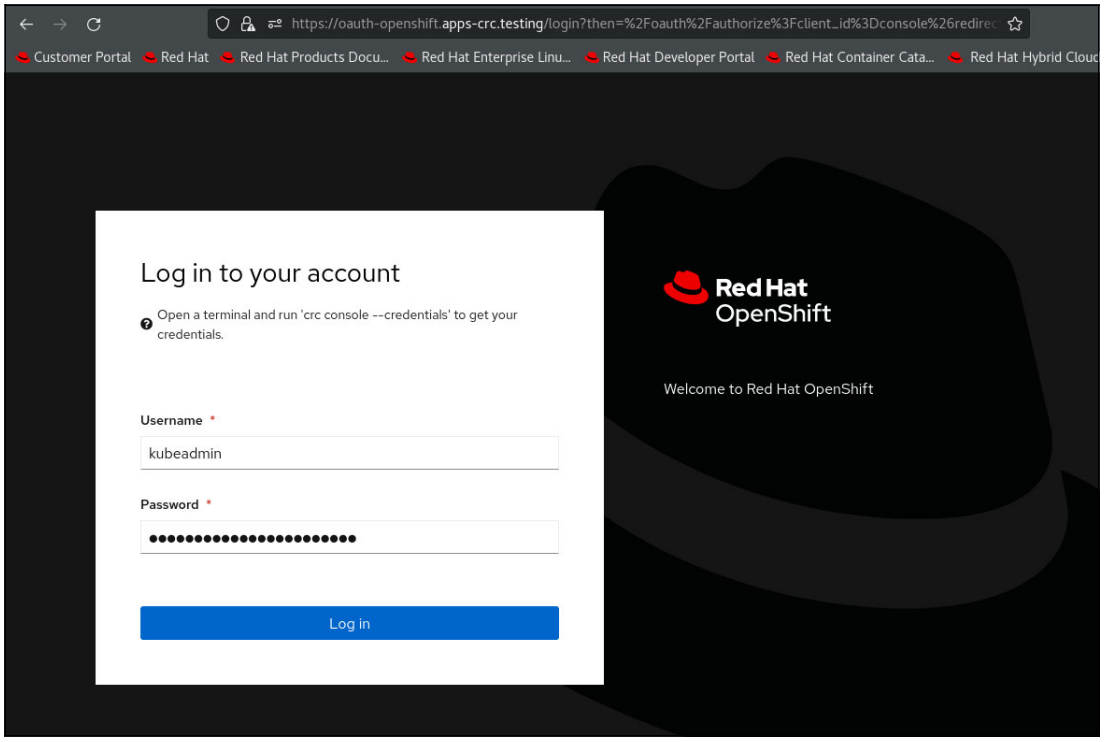


Figure 4-6 Red Hat login

You can then verify your cluster operators *status as seen in* Figure 4-7.

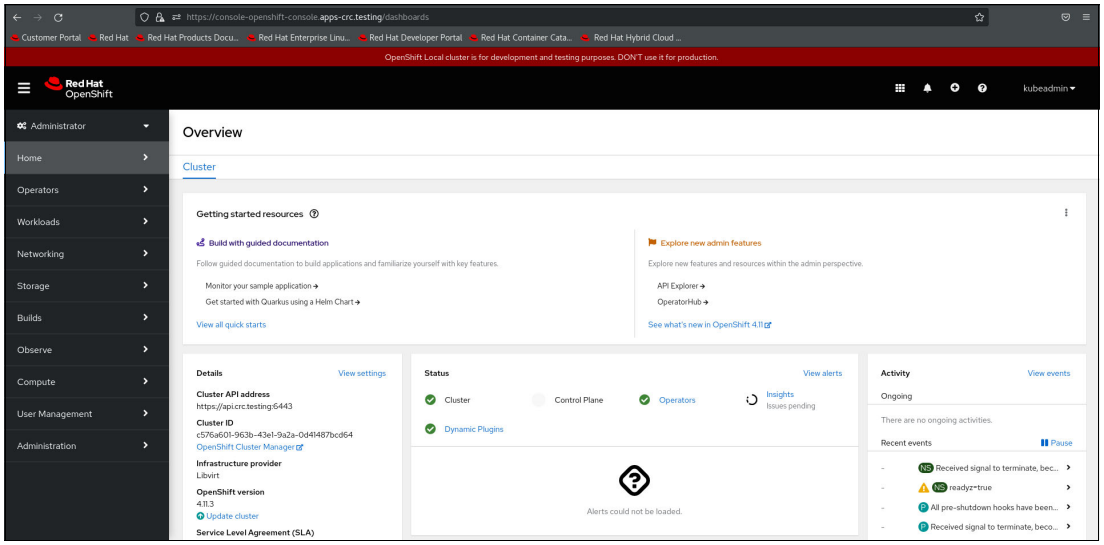


Figure 4-7 Cluster overview

Start, Stop or Delete the Red Hat OpenShift Local instance

The commands to Start, Stop, or Delete your local cluster instance are:

- To start the new Red Hat OpenShift Local instance: **`$ crc start.`**
- To stop the Red Hat OpenShift Local instance and container runtime: **`$ crc stop.`**
- To delete the existing Red Hat OpenShift Local instance: **`$ 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. These are the master or control plane nodes, worker or compute nodes, and infrastructure nodes which are special worker nodes for designated for infrastructure workload.

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:

- Kubernetes API server
- Scheduler
- Cluster Management
- Red Hat OpenShift API server
- 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. But some worker nodes can be used for specific workloads that are considered as an infrastructure related workload. For example:

- Some Red Hat OpenShift worker nodes could be explicitly used for container storage solutions. Like: Red Hat OpenShift Data Foundation.
- Red Hat OpenShift worker nodes could be explicitly used for logging, monitoring, image registry etc.
- Red Hat OpenShift worker nodes could be explicitly used for default router or sharding.

Other uses of master nodes

The Red Hat OpenShift Container Platform master nodes can also run applications like a worker node along 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 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 three-node Red Hat OpenShift cluster provides a resource efficient cluster for cluster administrators and developers to use for testing, development, and small production environments. This is documented in this 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 application programming interface (API), the controller manager, and the scheduler capabilities in one Red Hat OpenShift binary file.

In order for your cluster to operate you must have at least one healthy control plane host which has a master node.

High availability for the master node

The Red Hat OpenShift Container Platform cluster can be installed and configured in high availability mode, which uses multiple master nodes, or in non-HA mode, which uses a single master node. A single-node cluster generally has more restrictive resource constraints may not be supported on all hardware platforms.

The Red Hat OpenShift Container Platform cluster high-availability (HA) mode requires exactly 3 master nodes which are used to maintain three replica copies of the critical components, for example API server, etcd, controller manager and so on across three master nodes.

Note: Exactly three control plane nodes must be used for all production deployments in Red Hat OpenShift Container Platform cluster high-availability (HA) mode. That means, it's not possible to run with two masters or five masters as of now. See the official documentation on these sites:

- https://access.redhat.com/solutions/4833531
- https://docs.openshift.com/container-platform/4.11/architecture/control-plane.html#defining-masters_control-plane .

4.4.5 Disaster recovery

Disaster recovery is one of key requirements when planning for Business Continuity in your Red Hat OpenShift environment. You need to 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 that you need to consider and plan for are:

- Red Hat OpenShift Container Platform perspective.
  - Lost the majority of your control plane hosts (that is master node), leading to etcd quorum loss and the cluster going offline.
  - Accidental deletion of critical cluster data or configuration.
- Application workload perspective that 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 will require different precautions to overcome the disaster situations. The following documents provide solutions for recovering from the issues described above.

- https://access.redhat.com/articles/6270901
- https://access.redhat.com/solutions/5514051

Backup And Restore

Backup and restore is a traditional and effective method of recovery. In order to be able to recover your cluster with a restore you need to back-up all the critical configuration data for the Red Hat OpenShift Container Platform as well as for the application data.

The following command can be used to back up cluster data.

```
$ /usr/local/bin/cluster-backup.sh /home/core/assets/backup
```

Alternatively, you can use any backup/restore utility that supports the Red Hat OpenShift Container Platform. IBM Spectrum Protect Plus is one option that allows you to back up Red Hat OpenShift container data directly to cloud storage. This can include persistent volumes, 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 may need to replace an unhealthy bare metal etcd member whose node is not running or is in a NotReady state.

Disaster Recovery Site

A Disaster Recovery site or secondary site is required to provide a location to recover your environment in case of 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 could be another site in your enterprise or even a site provided by a cloud vendor.

Most Red Hat OpenShift environments require persistent storage to store the data required for the applications to run. Recovering your environment will require that your data be at your recovery site. For faster recovery the data will need to be consistently replicated from your primary site to your secondary location which will require 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 (RTO).

Reference architectures:

The IBM Academy of Technology has created some reference articles for use in your Red Hat OpenShift solution design. These can be found at this IBM Cloud Architecture site.

Reference architecture 1 starter environment

This is the minimal topology for a Red Hat OpenShift Container Platform v4.x designed for stateless and ephemeral workloads. This is shown in Figure 4-8 on page 129.

- Number of nodes required:
  - 3 master nodes
  - 2 worker nodes
- Application workloads:
  - Stateless
  - Ephemeral workloads.
- Use cases:
  - Development
  - Test

Consideration: There is no persistent storage in this solution and no high availability.

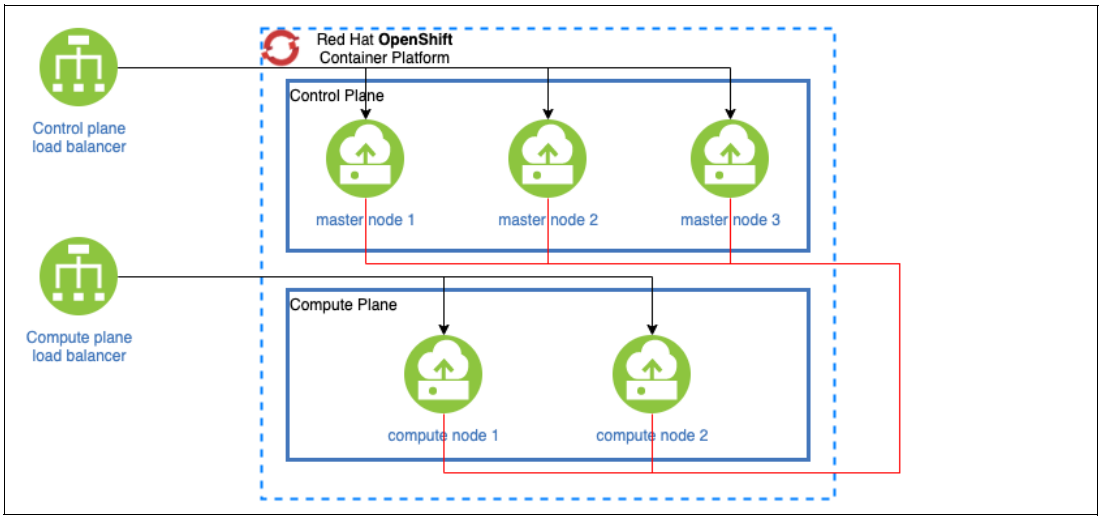


Figure 4-8 Starter environment

Reference architecture 2 three node cluster

This reference architecture which is shown in Figure 4-9 was first available in Red Hat OpenShift Container Platform 4.5. Currently it is only supported for deployment on bare metal.

- Number of nodes required:
  - 3 master nodes – master and work function 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 high availability.

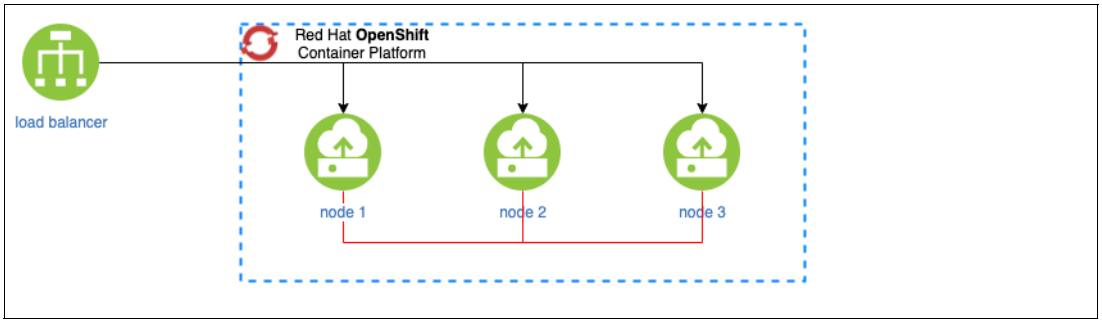


Figure 4-9 Three node cluster

Reference architecture 3 on-premises cluster

This cluster, which is shown in Figure 4-10, is deployed in one availability zone or Data Center.

- Number of nodes required:
  - 3 master nodes
  - 3 worker nodes
  - 3 infrastructure nodes (Elasticsearch requires 3 instances – one per node)
  - 3 storage nodes
- Application workloads:
  - Any - subject to limited SLA
- Use cases:
  - Development
  - Test
  - Integration
  - Production - subject to limited SLA

Consideration: There is no high availability in this design.

Note: The number of nodes and their sizing can be adjusted depending on the workload target.

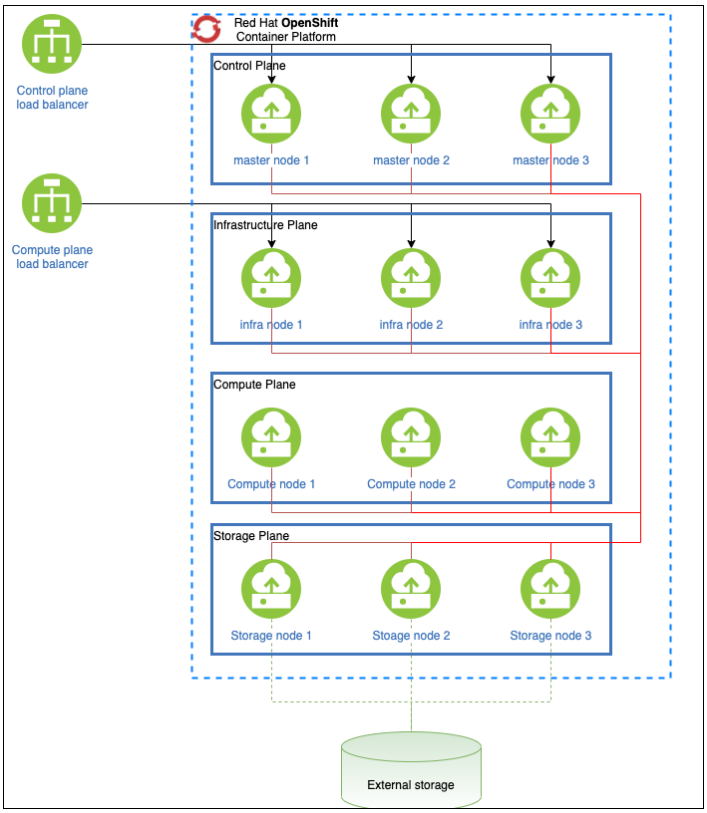


Figure 4-10 On-premises cluster

Reference architecture 4 cloud cluster

This design provides a highly available cluster and is shown in Figure 4-11. It requires that the nodes be spread across multiple availability zones - locations with different power or storage connections designed to not have a single point of failure in common with the other availability zones.

- Deploy cluster over availability zones (AZ), three availability zones required.
- Number of nodes required:
  - 3 master nodes
  - 3 worker nodes
  - 3 infrastructure nodes (Elasticsearch requires 3 instances – one per node)
- Application workloads:
  - Any - subject to limited SLA for DR
- Use cases:
  - Development
  - Test
  - Integration
  - Production - subject to limited SLA for DR

Consideration: Requires infrastructure with different availability zones. It provides for a highly available cluster but does not consider disaster recovery with a second site.

Note: The number of nodes and their sizing can be adjusted depending on the workload target.

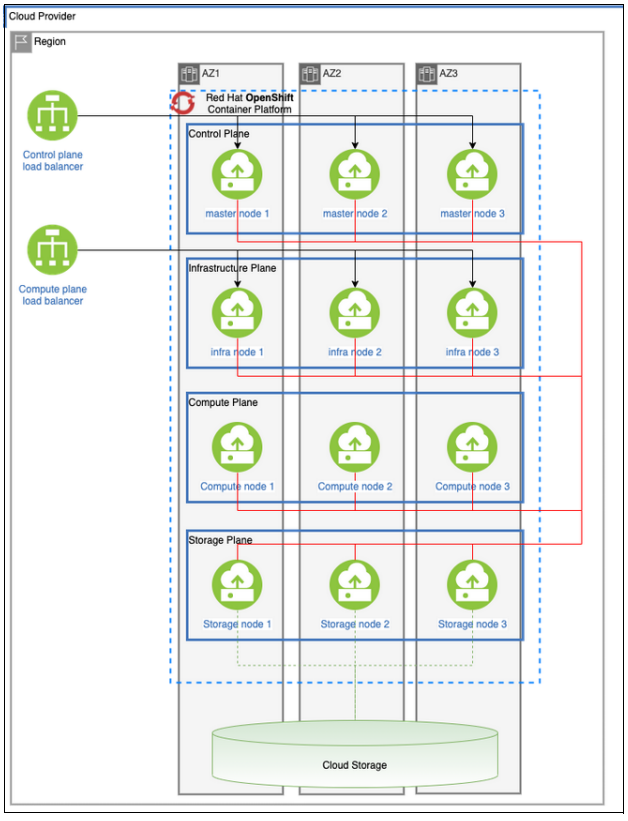


Figure 4-11 Cloud cluster

Reference architecture 5 two clusters on-premises

This configuration, shown in Figure 4-12, provides a highly available solution for your on-premises environment. It requires two data centers to provide the availability. The distance between the data centers will influence the options available for data replication and the RPO of the solution. In each data center the architecture will look like the one used in “Reference architecture 3 on-premises cluster” on page 130. Whether or not this satisfies disaster recovery requirements depends on the separation of the two data centers.

- Deploy clusters in different data centers. Two data centers required.
  - Minimum of 2 clusters, one per data center.
- Number of nodes per clusters:
  - 3 master nodes
  - 3 worker nodes
  - 3 infrastructure nodes (Elasticsearch requires 3 instances, one per node)
  - 3 storage nodes
- Application workloads:
  - Any
- Use cases:
  - Production - on-premises
    - Development/Test can coexist

Consideration: The replication methods, whether using middleware/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 will reflect on your Recovery Time Objective (RTO).

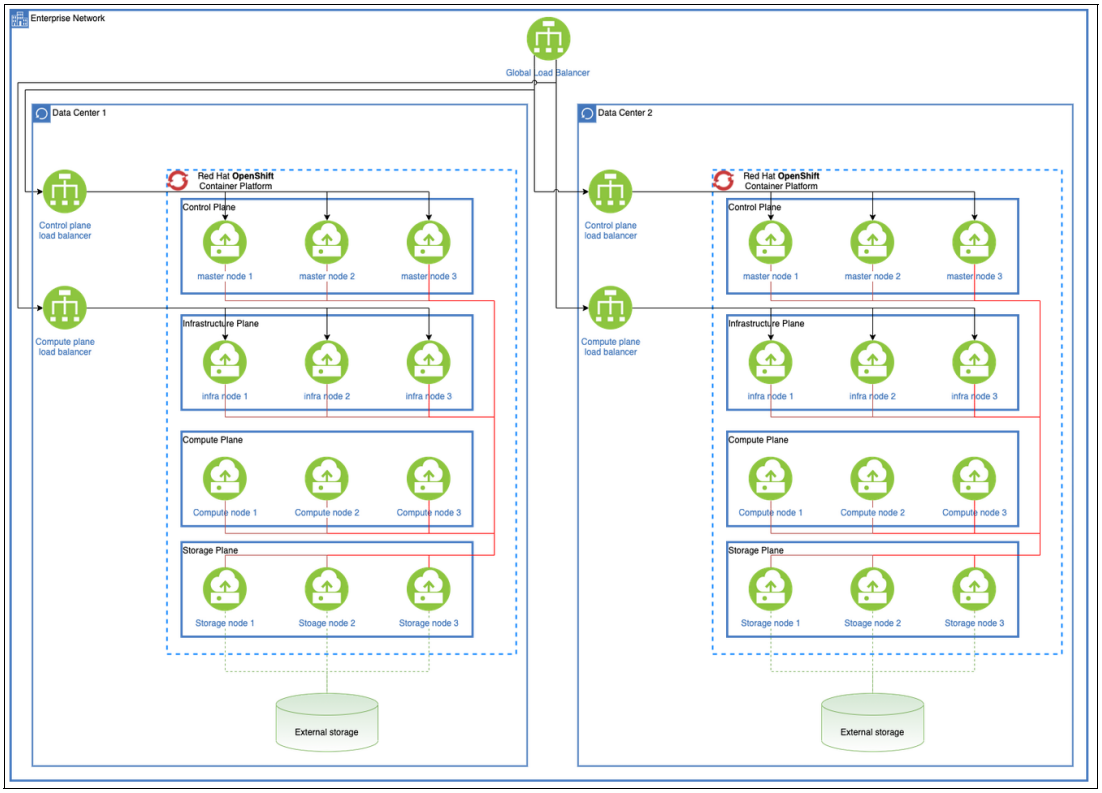


Figure 4-12 Two clusters on premises

Reference architecture 6 two clusters on two regions

This configuration, shown in Figure 4-13, is the cloud equivalent of “Reference architecture 5 two clusters on-premises” on page 132. It requires a cloud provider with two distinct regions and utilizes two clusters – one cluster in each region.

- Deploy clusters in different regions. Two regions required.
  - Minimum of 2 clusters, one per region.
- Number of nodes per clusters:
  - 3 master nodes
  - 3 worker nodes
  - 3 infrastructure nodes (Elasticsearch requires 3 instances, one per node)
- Application workloads:
  - Any
- Use cases:
  - Production - on-premises
    - Development/Test can coexist

Consideration: The replication methods, whether using middleware/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 will reflect on your Recovery Time Objective (RTO).

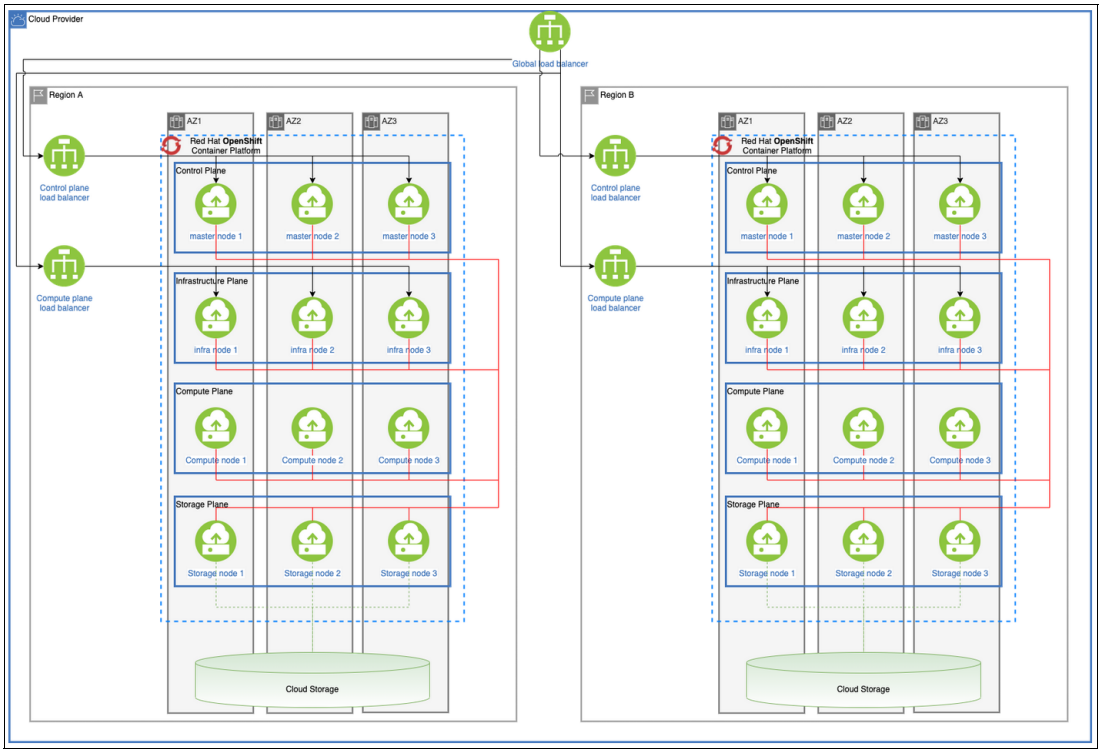


Figure 4-13 Two clusters cross region

Anti-patterns

The anti-patterns are architectures that may seem seductive but they should not be implemented. At the end, they generated more issues than they solve problems. Their

implementation focuses on a small set of requirements and they don't consider the full picture.

Anti-patterns 1 two data center stretch cluster

This configuration, shown in Figure 4-14, looks interesting but is *not recommended*. It is not recommended due to potential issues due to:

- Network latency
- Network issues
- Loss of quorum

These issues will lead to inconsistent performance and an overall unstable environment due to difficultly diagnosing and solving any issues.

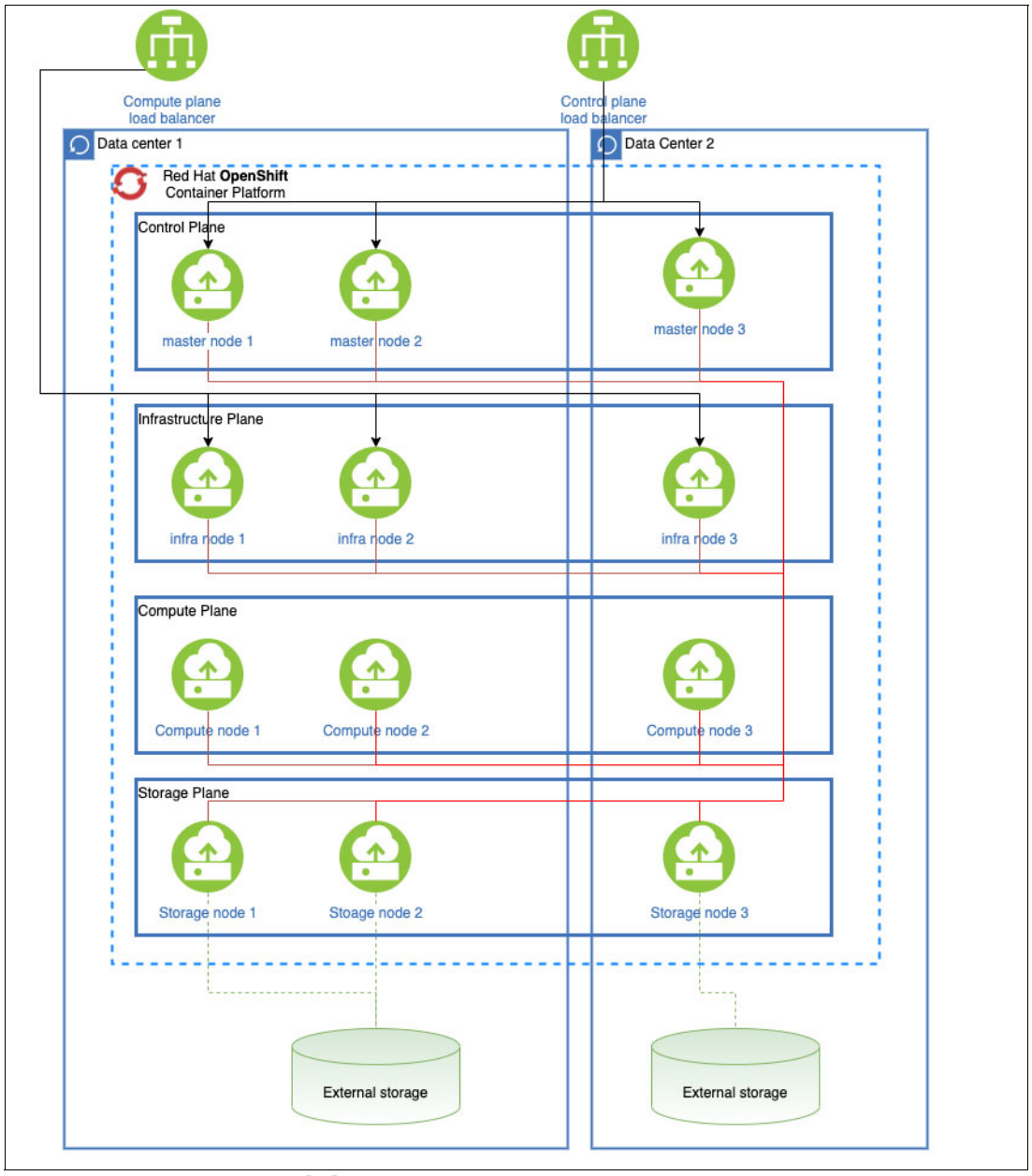


Figure 4-14 Not recommended - stretch cluster

Anti patterns 2: Hybrid stretch cluster To not be deployed

Just like “Anti-patterns 1 two data center stretch cluster” on page 134, this configuration, shown in Figure 4-15, is *not recommended*. This is subject to similar problems with management and stability due to:

- Network latency
- Network issues
- Scheduling and workload placement

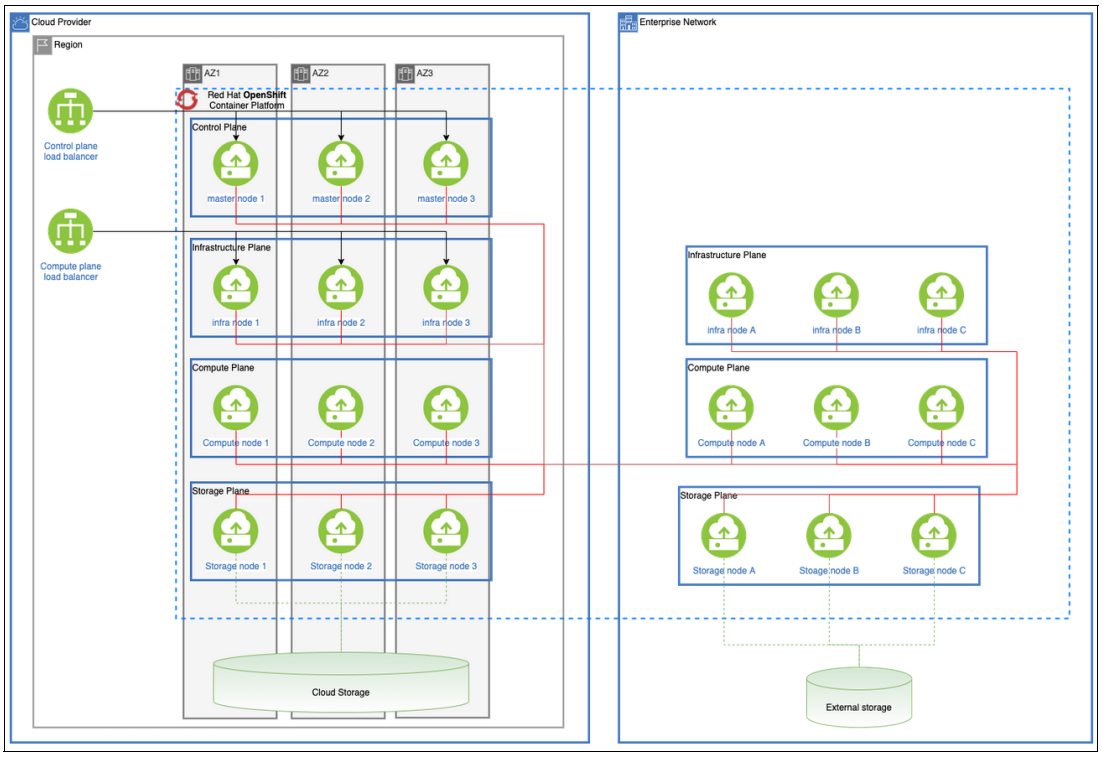


Figure 4-15 Not recommended hybrid stretch cluster

Conclusion

In summary, designing and planning your Red Hat OpenShift architecture and infrastructure is critical to the success you will have in deploying and running your applications in that new environment. You need to understand your business objectives and the application workload that you intend to run in the environment. Understanding those will help 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 OpenShift infrastructure

Factor to consider	Some Choices
Hardware Resource and Platform	IBM Power System or IBM Power Virtual Server IBM Virtual Private Cloud (VPC) IBM System Z Generic x86 Infrastructure
Application Workload	Stateful or Stateless Microservices Artificial Intelligence / Machine Learning

Factor to consider	Some Choices
Environment:	Development Production Disaster Recovery Site, on-premises public, private or hybrid cloud
Business objectives	Recovery Point Objective (RPO) Recovery Time Objective (RTO) Service Level Agreement (SLA) Highly Availability Requirements Compliance Requirements

4.5 Red Hat OpenShift Ecosystem

An Ecosystem according to Britannica[2] is the 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 as well as 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. Kubernetes orchestration allows you to 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 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's fast, enables self-service provisioning, and integrates with a variety of tools. In other words, there's no vendor lock-in. This is demonstrated by the variety of environments, both HW and SW based (virtualized) environments where Red Hat OpenShift can run, on premise or in a cloud solution. It is also shown by all the services and solutions we can run on Red Hat OpenShift itself.

There are many solutions and tools which can help in operating the ecosystem and which can help in deploying new services on it. One of the new solutions which can provide the physical environment and also help in deploying and operating 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 as needed. For information on IBM Storage Fusion see https://www.ibm.com/products/spectrum-fusion.

Other elements of this ecosystem help creating and deploying solutions like DevOps and CI/CD tools. Another player in the ecosystem is Service mesh by MuleSoft which can assist in governance, security and discoverability and API management. Another major player is GitOps whose main benefit is the security and operabiliy which helps to implement Infrastructure as Code practices.

[2] https://www.britannica.com/science/ecosystem

Who uses Red Hat OpenShift?

Red Hat's Complete Enterprise Production Eco System gives developers and IT operators a consistent application platform to manage hybrid cloud, multi-cloud, and edge deployments so 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 running Enterprise applications, helping your Enterprise to modernize existing or legacy applications, and accelerate new cloud-native application development and delivery at scale across any infrastructure. The following are 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 Environment scalability is best in class option for Enterprises.

Flexibility

Integration makes complete product lifecycle very flexible. Red Hat OpenShift simplifies deployment and management of a hybrid infrastructure, giving you the flexibility to have a self-managed or fully managed service, running on-premise or in cloud and hybrid environments.

Open source standards

Red Hat OpenShift incorporates Open Container Initiative (OCI) containers and Cloud Native Computing Foundation-certified Kubernetes for container orchestration, in addition to other open source technologies.

Container portability

Container images built on the OCI industry standard ensure portability between developer workstations and Red Hat OpenShift production environments.

Enhanced developer experience

Always best in choice. Red Hat OpenShift offers a comprehensive set of developer tools, multi-language support, and command line and integrated development environment (IDE) integrations. Features include continuous integration/continuous delivery (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-premise solutions using vSphere, Red Hat OpenStack Platform, Red Hat Virtualization, or bare metal. Services used from the Operator-Hub can be deployed fully configured and are upgradeable with 1 click.

Automation

Streamlined and automated container and application builds, deployments, scaling, health management, and more are included. Ansible or any other automation product provide integration capability, helping 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, and remote worker nodes, which better map to varying physical size, connectivity, and availability requirements of different edge sites. The edge use cases are further enhanced with support for Red Hat OpenShift clusters on ARM architecture, 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 enterprise registry with built-in scanner. Red Hat Advanced Cluster Security for Kubernetes enhances this with security capabilities like runtime threat detection, full life cycle vulnerability management, and risk profiling.

Red Hat OpenShift also comes with prebuilt operator for compliance which is 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 via the Computer Storage Interface driver.

Robust ecosystem

An expanding ecosystem of partners provides a wide variety of integrations. Third parties deliver additional storage and network providers, IDE, CI, integrations, independent software vendor solutions, and more.

Operator hub:

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

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 much more than older tools like Helm and base Red Hat OpenShift resources, which managed applications and services. Basically it can manage all kind of 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

Capability	Helm charts	Operators
Updates using Kubernetes manifests	Standard	Standard
Upgrades 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 defined by CSV resources, via watching predefined requirements and running the install strategy defined.
- Catalog Operator: responsible for resolving and installing cluster service versions (CSVs) and the required resources they specify and also monitors the defined catalog sources and manage updates of packages based on configuration.

Table 4-7 shows the Custom Resource Definitions which are managed by the two Operators.

Table 4-7 Custom Resource Definition 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 be created to automatically install or upgrade a CSV.
CatalogSource	catsrc	Catalog	A repository of CSVs, CRDs, and packages that define an application.
Subscription	sub	Catalog	Used to keep CSVs up to date by tracking a channel in a package.
OperatorGroup	og	OLM	Configures all Operators 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 which can be added into Red Hat OpenShift. The default Catalog Sources are the following:

- certified-operators: Products from leading independent software vendors (ISVs). Red Hat partners with ISVs to package and ship. Supported by the ISV.
- community-operators: Optionally visible software 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 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 installing Operators from community sources in a production environment.

As Operators manages the standard and custom resources of a Package, so installing an Operator will only create CRDs if necessary and set the update channels. After install we can chose which update channel we are following, based on which OLM can do automatic updates as shown in Figure 4-16.

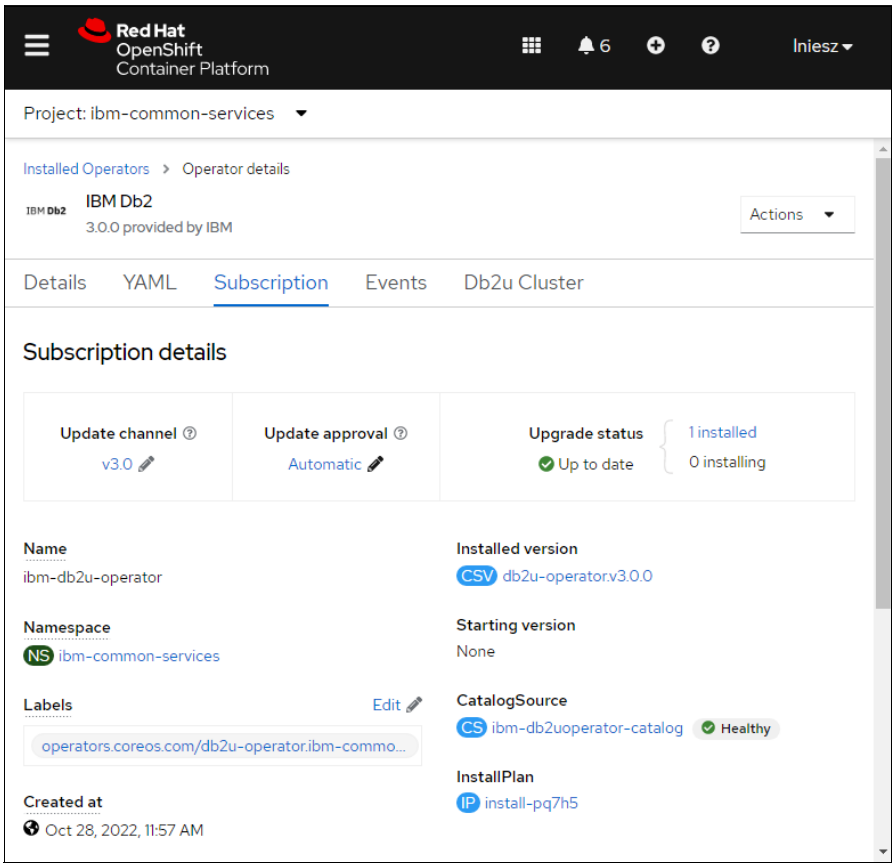


Figure 4-16 IBM DB2® Operator Subscription details

Operator SDK

Operator SDK is a component of the Operator Framework. It provides a Command Line Interface (CLI) based tool to build, test and deploy an Operator.

Operators built by Operator SDK watch the resources and process events based on changes of resources in a handler. The handler takes actions to reconcile the state of the application package.

Operators can be developed based on Go, Ansible and Helm but with different capabilities as shown in Figure 4-17[3].

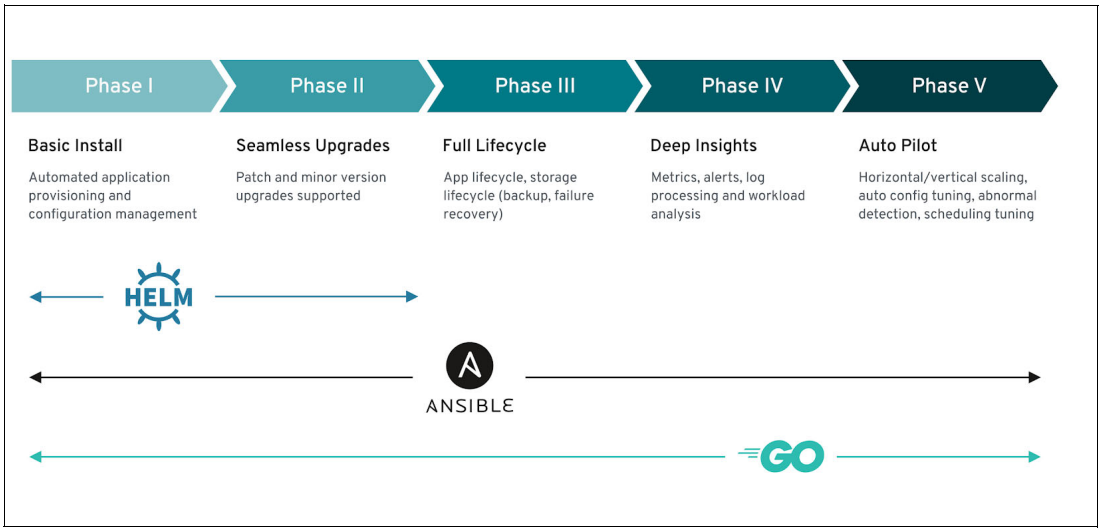


Figure 4-17 Operator Capability Level

Cluster Operators

Red Hat OpenShift itself is using Operators managing key platform elements, and they 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   MESSAGE
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
operator-lifecycle-manager                 4.10.34   True        False         False      39d
operator-lifecycle-manager-catalog         4.10.34   True        False         False      39d
```

[3] https://redhat-connect.gitbook.io/certified-operator-guide/

```
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 Operators, and in this way the whole Red Hat OpenShift cluster, is controlled by the ClusterVersion configuration, which is watched by the Cluster Version Operator. This is where parameters related to automatic updates can be set. Figure 4-18 shows the actual version and the possible update channels of our test cluster.

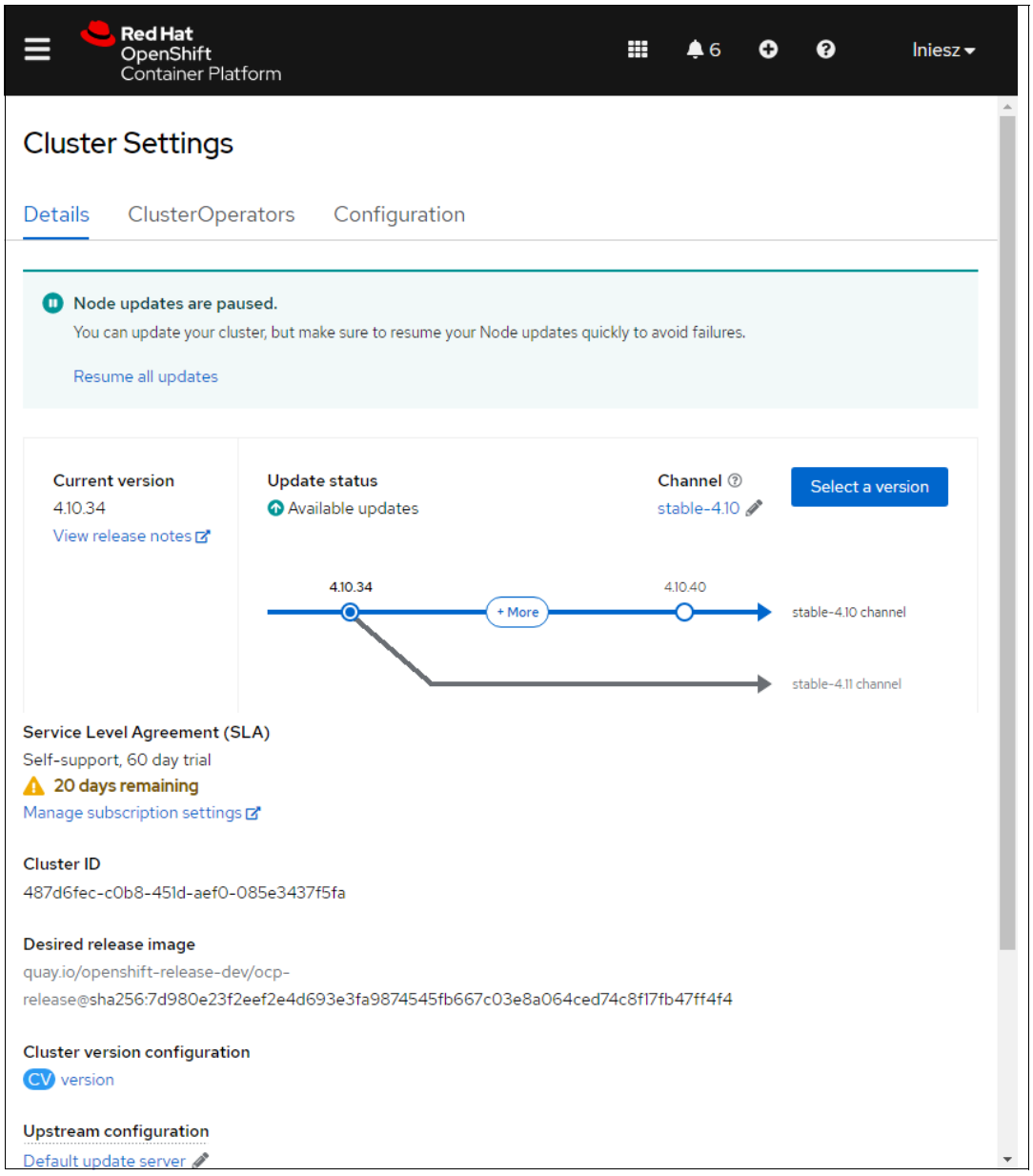


Figure 4-18 Cluster version and update channels in GUI

The Operators are implemented as Red Hat OpenShift PODs which are using other resources to manage the cluster. The related resources can be viewed in Red Hat OpenShift GUI as well as in CLI, as shown for the authentication cluster Operator in Example 4-12 on page 143.

Example 4-12 List 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"
  }
]
```

Actually here we can see the configuration resource: **`cluster.oauth.config.openshift.io`** which we can use to add authentication methods into the cluster. In Example 4-13 we can see 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 Mesh[4] addresses a variety of 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 an easy 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 functionality, including A/B testing, canary releases, access control, and end-to-end authentication.

Red Hat OpenShift Service Mesh provides a number of key capabilities uniformly across a network of services:

- **Traffic Management** - Control the flow of traffic and API calls between services, make calls more reliable, and make the network more robust in the face of adverse conditions.
- **Service Identity and Security** - Provide services in the mesh with a verifiable identity and provide the ability to protect service traffic as it flows over networks of varying degrees of trustworthiness.
- **Policy Enforcement** - Apply organizational policy to the interaction between services, ensure access policies are enforced and resources are fairly distributed among consumers. Policy changes are made by configuring the mesh, not by changing application code.
- **Telemetry** - Gain understanding of the dependencies between services and the nature and flow of traffic between them, providing the ability to quickly identify issues.

For more details on preparation and the installation of Red Hat OpenShift Service Mesh your Red Hat OpenShift Container Platform refer to the Service Mesh Documents.

4.5.3 DevOps and CI/CD pipelines

The word “DevOps”[5] is a mashup of “development' and “operations” but it represents a set of ideas and practices much larger than those two terms alone, or together. DevOps speeds up

[4] https://docs.openshift.com/container-platform/4.11/service_mesh/v2x/ossm-about.html

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 opensource principles and transparent, agile approaches to work. The culture of opensource software projects can be a blueprint for how to build a DevOps culture. Freely 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 using different software architecture, for example 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. That's why continuous integration and continuous deployment (CI/CD) is so important to DevOps teams.

DevOps platform & 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'll need to 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 manually operations tasks and manages them 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 intended to deliver increased business value and responsiveness through rapid, high-quality service delivery. SRE can be considered an implementation of DevOps[6].

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 life cycles, improved service quality and reliability, and reduced IT time per application 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/CD)

CI/CD[7] 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 integrating new code can cause for development and operations teams (AKA “integration hell”).

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

[5] https://www.redhat.com/en/topics/devops
[6] https://www.redhat.com/en/topics/devops/what-is-sre
[7] https://www.redhat.com/en/topics/devops/what-is-ci-cd

supported by development and operations teams working together in an agile way with either a DevOps or site reliability engineering (SRE) approach.

CI/CD tools

CI/CD tools can help a team automate their development, deployment, and testing. A few of them are described below:

- One of the best-known opensource tools for CI/CD is the automation server Jenkins. Jenkins is designed to 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, other opensource CI/CD tools you may wish to investigate include:
  - Spinnaker, a CD platform built for multicloud environments.
  - GoCD, a CI/CD server with an emphasis on modeling and visualization.
  - Concourse, “an open-source continuous thing-doer.”
  - Screwdriver, a build platform designed for CD.

Additionally, any tool that's foundational to DevOps is likely to be part of a CI/CD process. Tools for configuration automation (such as Ansible, Chef, and Puppet), container runtimes (such as Docker, rkt, and cri-o), and container orchestration (Kubernetes) aren't strictly CI/CD tools, but they'll show up in many CI/CD workflows.

Tekton Pipelines is available in Red Hat OpenShift via 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 OpenShif cluster node is to use GitOps. This section describes how to do that. An example of this is shown in 6.3, “GitOps for system configuration” on page 200.

What is GitOps

GitOps is a set of principles originally defined for operating and managing software systems, but these principles could be very useful in managing software defined infrastructures as well in running those software systems. The principles are derived from already 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 related to standardizing GitOps definitions and implementation. It is managed by GitOps Working Group under Cloud Native Computing Foundation (CNCF). Read more about GitOps here.

GitOps principles

In this chapter we show how could these principles be used in practice to help the operation and especially the tuning and performance configuration of Red Hat OpenShift clusters on IBM Power Systems servers. We also show the available tools that can be used to put the principles in practice.

The principles are the following:

- Declarative: The desired state of a GitOps managed system must be expressed in declarative forms.

- Versioned and Immutable: The storage of the desired state must provide immutability and must provide versioning, so the whole version history must be stored.
- Pulled Automatically: The desired 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 will apply the desired state on the system.

As we read the principles it is very clear that these are on common grounds 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 however can be applied not only at the application life cycle management but on the underlying software defined infrastructure itself. Thinking about the subject of our book, namely tuning and management of workloads on Red Hat OpenShift on IBM Power Systems servers, this is possible because of the way Red Hat OpenShift is working.

Red Hat OpenShift tuning related resources to manage with GitOps

In Red Hat OpenShift not only are the application and related resources – like secrets, configurations, and ingress routes – are defined by YAML files, but the underlying infrastructure elements also. 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 it. The collection of these YAML file can be handled as configuration database, but when moving towards the GitOps way of working these can be stored in a Git repository.

The usage of Red Hat CoreOS as the operating system of the nodes and the operating system updates implemented via Red Hat OpenShift provides an immutable and secure way to handle the node configurations. Direct login to the nodes can be disabled, if it is not then direct modification of operating system changes will be reverted by Red Hat OpenShift at the next update, which could cause problems for the operation.

A node – which can be either master, infrastructure, or compute node in an Red Hat OpenShift cluster – is basically a running operating system on a virtual server or in case of bare metal a full physical server. In the following we show the resource types in Red Hat OpenShift which can be used for tuning related configuration:

- **Node**: Manages node grouping and labels, to be 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 will be 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**
    - Define 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 the following node and operating system related tuning and assign them to nodes.
- **KubeletConfig**: Can be used to configure one of the main Red Hat OpenShift daemon.

Red Hat OpenShift GitOps

Red Hat provides Red Hat OpenShift GitOps for the automatic pulling and reconciliation of the configurations, which can be stored a Git based external environment providing the storage and version control of the YAML files. Red Hat OpenShift GitOps is based on ArgoCD, a CNCF Incubating project. See the official documentation of Red Hat OpenShift GitOps here.

In Red Hat OpenShift GitOps a DEX server provides the authentication and authorization configuration, which enables it to use RBAC configuration set up in Red Hat OpenShift to be used in GitOps as well. To enable GitOps configuring resources in an Red Hat OpenShift namespace the namespace should be labeled 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 as we will show it in our GitOps use case as shown in section 6.3, “GitOps for system configuration” on page 200.

The possible configurations are changing and evolving with new Red Hat OpenShift and supported HW versions and types.

Note: Applying Red Hat OpenShift configuration changes to a live cluster can result in restarting cluster nodes, which can cause the POD to stop and restart on other nodes. This could 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:
  - Using GitOps for infrastructure management we do not define real SW applications, but a collection of Red Hat OpenShift resources, which can be defined by YAML files stored on 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 using GitHub or a compatible repository. This will provide versioning and even approval and review processes for changes before applying them onto the live cluster.

The definitions can be either standalone YAML files, HELM charts or Kustomize based configurations.

GitOps Workflow

1. Setup:
   a. Create a Git repository for the configuration YAML files. Setup approval process and necessary security configuration.
   b. Create subdirectories to group together Red Hat OpenShift resource definition YAML files. A subdirectory will be assigned to a GitOps application and all YAML files, HELM charts or Kustomize based resources will be applied to the defined GitOps “Application”.
   c. Collect 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. Setup Red Hat OpenShift GitOps in target clusters:
      i. Install the operator.
      ii. Create 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 ArgoCD GUI or CLI create a project for handling configuration changes, specifying destination cluster, lists of cluster resources to allow modifications on them, allowed source repositories which can be source for configuration YAML files.
   f. Create “Application” specifying the project. Provide all necessary setup specifying project, source repository and target. Setup automatic synchronization, if necessary, but be careful that some node reconfiguration can cause restarting of the PODs running on the node as the node’s operating system can reboot to get new settings.
2. Operation:
   a. Change configuration YAML files in Git repository as necessary. This is done normally via Git pull requests and going through necessary approvals.
   b. Check the changes in ArgoCD application and synchronize the configuration if it is not set to synchronize automatically.
   c. Check applied configuration in Red Hat OpenShift cluster.

We will show a sample setup how to use GitOps with Red Hat OpenShift to configure and tune the cluster nodes and to store the configurations in GitHub in Section 6.3, “GitOps for system configuration” on page 200.

4.6 Running Red Hat OpenShift on IBM Power Systems

This section focuses on the benefits of running Red Hat OpenShift on IBM Power Systems. Also, how you can start your deployment of Red Hat OpenShift Container Platform on a IBM

Power System either in an on-premises bare-metal system or in an IBM Power Systems Virtual Server on IBM Cloud.

4.6.1 Red Hat OpenShift on IBM Power Systems

The modern application development and modernization of existing applications require a robust platform that ensures scalability, agility, portability, security and resiliency. And on the other hand, incorporates the new ideas, features and benefits in their application at a much faster pace than traditional applications.

The software development practice and culture that integrates the tasks of development and IT operations teams to shorten the application development life cycle. For example: DevOps (that combines software development (Dev) and IT operations (Ops).) or SRE (Site Reliability Engineering).

For many application workloads, Red Hat OpenShift on IBM Power Systems 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 Systems can support more users per server than competing technologies and can be dynamically scaled up or down to meet your workload requirements.

In the rest of this sections, 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 Power provides even more benefits for your containerized applications.

Benefits of Red Hat OpenShift

Red Hat OpenShift's full-stack automated operations and self-service provisioning for developers and IT operators to work together more efficiently to move ideas from development to production. In other words, OpenShift is an enterprise-grade product full of useful features and capabilities that combines software development and IT operations.

Red Hat OpenShift manages hybrid technologies and applications, helping you modernize existing applications and accelerate new cloud-native application development and delivery at scale across any infrastructure. Red Hat OpenShift gives developers and IT operators a consistent app platform to manage hybrid cloud, multi-cloud, and edge deployments so your business can innovate quickly. Red Hat OpenShift features and benefits include the following:

- 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
- High availability (HA)
- Managing security
- Operators
- Sandboxed containers
- Serverless
- Service mesh
- Virtualization
- Windows containers

You can visit this Red Hat official site for more details of the Red Hat OpenShift capabilities, features and benefits.

Benefits of Red Hat OpenShift on IBM Power Systems

As we have discussed, the Capacity Planning for the Red Hat OpenShift is important and it will determine 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. As per Red Hat provided guidelines for the maximum number of Pods per node, which is 250. It is recommended to not exceed this number because it results in lower overall performance.

x86 system versus IBM Power System

An x86 core running with hyper threading is equivalent to two Kubernetes CPUs. Therefore, when running with x86 hyper threading, a Kubernetes CPU is equivalent to half of an x86 core. This conversion factor is shown in Table 4-8.

A PowerVM core can be defined to be 1, 2, 4, or 8 threads with the SMT setting. Therefore, when running on PowerVM with SMT-4, a PowerVM core is equivalent to four Kubernetes CPUs whereas when running with SMT-8, the same PowerVM core is equivalent to eight Kubernetes CPUs. Therefore, when running with SMT-4, a Kubernetes CPU is equivalent to a quarter of a PowerVM core and when running with SMT-8 a Kubernetes CPU is equivalent to one eighth of a PowerVM core.

Table 4-8 CPU conversion

	vCPU	x86 Cores (SMT-2)	Physical SMT-2 POWER cores	Physical SMT-4 POWER cores
vCPUs and x86 or IBM Power Systems cores	2	1	1	0.5

If your POD CPU resource was defined to run on x86, you must consider the effects of the IBM Power's performance advantage and the effects of Kubernetes resources being assigned on a thread basis. For example, for a workload where IBM Power Systems has a 2X advantage over x86 when running with PowerVM SMT-4, you can assign the same number of Kubernetes CPUs to IBM Power Systems that you do to x86 to get equivalent performance. This conversion factor is shown in Table 4-9 on page 152.

Table 4-9 vCPUs to physical cores conversion

vCPU	Physical x86 cores	Physical SMT-2 POWER cores	Physical SMT-4 POWER cores	Physical SMT-8 POWER cores
56	28	28	14	7

You can find more details and some lessons learned along with some tips and tricks in Section 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 System 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 on-premises Infrastructure to run their core systems and applications. The hybrid cloud connects an organizations 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 will continue to thrive include:

- 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 disaster recovery with respect to service level agreement (SLA).

But this can be an issue for smaller companies who have limited budgets and technical resources. Cloud computing for a smaller companies can help them to reduce operational costs and benefit from other benefits such as:

- Increased ability to optimize disaster recovery and/or 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 associated with updates.
- Better IT infrastructure management and flexibility.

For example, Figure 4-19 shows that Red Hat OpenShift on IBM Cloud will have more uptime (99.99%) compared to other cloud providers.

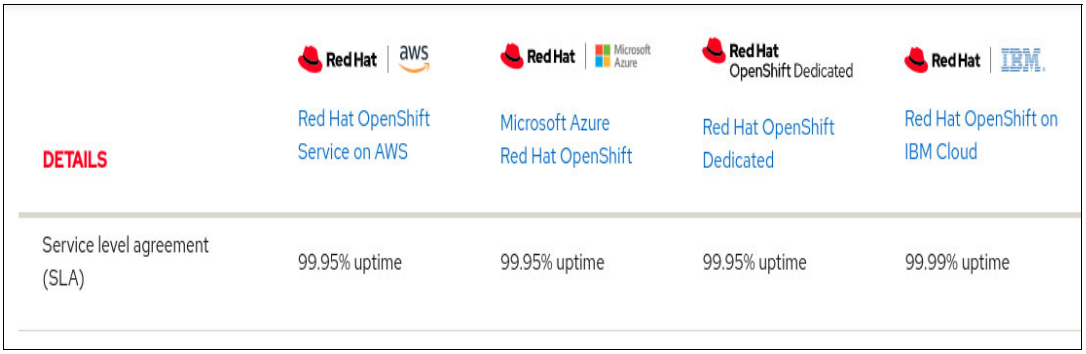


Figure 4-19 SLA uptime (99.99%) comparison

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 an IBM Power System either in an on-premises bare-metal system or on an IBM Power Systems Virtual Server on IBM Cloud. For step-by-step installation instruction refer to Chapter 6, “Deployment scenarios”, in *Red Hat OpenShift V4.X and IBM Cloud Pak on IBM Power Systems Volume 2*, SG24-8486 along with other tips and considerations.

Deploying Red Hat OpenShift Container Platform V4.x in IBM Cloud

The Red Hat OpenShift Container Platform V4.x deployment process in IBM Power Systems Virtual Server on IBM Cloud uses Red Hat Ansible and Terraform.

The installation process uses the code that is found at the following GitHub repository.

At a high level, the Red Hat OpenShift Container Platform V4.x deployment process in IBM Power Systems Virtual Server on IBM Cloud includes the following steps:

- Setting up the IBM Cloud environment:
  - IBM Power Systems Virtual Server Service.
  - Private network.
  - Import Red Hat images.
  - User application programming interface (API) key.
- Setting up the deployment host:
  - Install Terraform.
  - Install IBM Cloud Terraform Provider.
  - Install the IBM Power Systems Virtual Server command-line interface (CLI).
- Deploy the Red Hat OpenShift Container Platform:
  - Clone the ocp4-upi-powervs Git repository.
  - Set up the Terraform variables.
  - Install the 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 infrastructure as code with Terraform to simplify the deployment.

Deploying Red Hat OpenShift Container Platform V4.x on IBM Power Systems Servers

The high-level Red Hat OpenShift Container Platform V4.x deployment process in IBM Power Systems Virtual Server includes the following steps:

- PowerVM configuration for network installation.
- Preparing your environment:
  - Configuring the DHCP server.
  - Configuring the tftp server.
  - Configuring the DNS server.
  - Configuring the web server.
  - Configuring the load balancer.
- Starting the servers to install Red Hat Enterprise Linux CoreOS using one of the following methods:
  - Installing from ISO instead of tftp server
  - Installing from DHCP and tftp server
- Creating the SSH key.
- Going through the installation process.

- Checking the installation.
- Backing up your cluster.

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 for a detailed step by step guide.

Also see 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 for some lessons learned along with some tips and tricks.

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:

- You can use Red Hat Enterprise Linux 7/8 System for this Ansible Playbooks.
- You're on a Network that has access to the internet.
- The ocp4-helpernode will be your LB/DHCP/PXE/DNS and HTTP server.
- You can disable installing DHCP on the helper, if required.
- You still have to do the Red Hat OpenShift Install steps by hand.

You will be running the openshift-install command from the ocp4-helpernode.

The prebuilt playbook is found at the following GitHub repository.

Advanced deployment of Red Hat OpenShift Container Platform

The process for installing Red Hat OpenShift V4.x on IBM Power Systems servers may be slightly different based on your architecture and the purpose of your cluster. Whether it is for development and testing, production, Disaster Recovery, or other use cases.

Static IP Address or DHCP server

It is recommended to use a DHCP server for long-term management of the cluster machines. but if a DHCP service is not available for your user-provisioned infrastructure (UPI) due to considerations about single points of failure, dependencies, or compliance policies you can use static IP addresses for all of the nodes.

To set up static IP addresses or configure special settings, such as bonding, you can do one of the following:

- Pass special kernel parameters when you boot the live installer.

  See 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.

  Set up static IP configuration for an RHCOS node as shown in this Red Hat solution.

- Configure networking from a live installer shell prompt using available Red Hat Enterprise Linux tools,

  Using nmcli or nmtui.

Configure networking from a live installer shell prompt using nmcli or nmtui.

To configure an ISO installation, use the following procedure.

1. Boot up the IBM Power System using an ISO installer. Once the system is successfully running up you will see the live system shell prompt as shown in Figure 4-20.

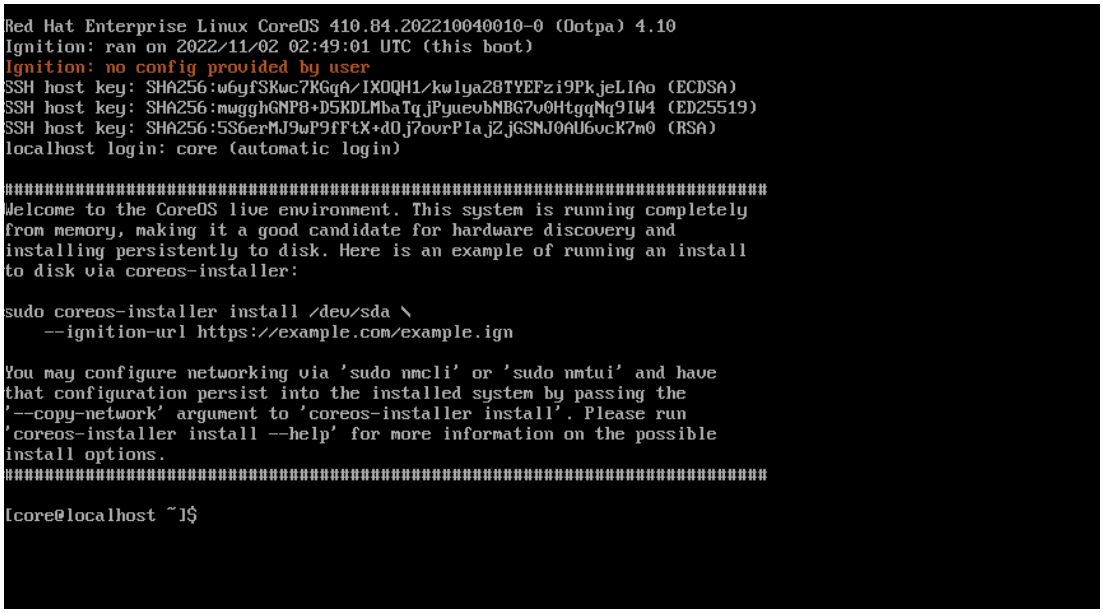


Figure 4-20 CoreOS live system shell promptt

2. From the live system shell prompt, configure networking for the live system using available Red Hat Enterprise Linux tools, such as nmcli or nmtui. Example 4-14 shows example nmcli commands.

Example 4-14 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 on page 156, adding the --copy-network option to copy networking configuration.

Example 4-15 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 only copies networking configuration found under /etc/NetworkManager/system-connections. In particular, it does not copy the system hostname.

For the static hostname, you can create separate ignition files (*.ign) with customized hostname configuration for all of the nodes. You can modify ignition files manually and can use tools like filetranspile or butane. For more details refer to this Red Hat document.

4. Reboot into the installed system.
5. Now copy those settings to the installed system so that they take effect when the installed system first boots. You must repeat the process for each of your nodes. You can also configure a bonding interface at the live system shell prompt and verify the network settings before you install the system.

For more details on how to use nmcl*i and coreos-installer comm*and refer to this Red Hat OpenShift Document, and this Red Hat Enterprise document for Red Hat Enterprise Linux8.

Configuring the load balancer

A best practice for production environments is to have two load balancers and depending on the architecture may need different type load balancers for different scope. For example:

- ***Local Traffic Managers (LTM) or Enterprise Load Balancers (ELB):*** provide load balancing services between two or more servers/applications in the event of a local system failure. Usually required for single Red Hat OpenShift clusters.
- ***Global Traffic Managers (GTM):*** provide load balancing services between two or more sites or geographic locations. Usually 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:

- How to set up HAProxy as a load balancer.
- How to configure a Load Balancer using keepalived.

Creating Automated etcd Backup in Red Hat OpenShift 4.x

Once your cluster is up and 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. 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 for details. Automated backups can assist in the recovery of one or more Master Node Clusters on Red Hat OpenShift 4.x while providing a minimum Recovery Point Objective (RPO).

Below are the necessary resources you will need to create automatic backups using *CronJob* from Red Hat OpenShift:

Namespace
Service Account
Cluster Role
Cluster Role Binding
Set Privileges for Service Account
CronJob

See the Red Hat official blog post on How to Create Automated etcd Backup in Red Hat OpenShift 4.x for step-by-step instructions. Another option is the Red Hat solutions on Red Hat OpenShift Container Platform 4.x: etcd backup cronjob which uses NFS persistent storage to store the backup files.

5

IBM Cloud Paks on Red Hat OpenShift running on IBM Power Systems

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. In this chapter we discuss these Cloud Paks which run on IBM Power Systems:

- IBM Cloud Pak for Data
- IBM Cloud Pak for Business Automation
- IBM Cloud Pak for Integration
- IBM Cloud Pak for Watson AIOps
- IBM Cloud Pak for WebSphere Hybrid Edition

Topics covered in this chapter are:

5.1 Introduction

The 2020s have ushered in an age of uncertainty. Rapid change has shifted market dynamics and upended old business models, revealing vast disconnects between digital investments and customer needs.

Your business must adapt to these changes. Already 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's time to rethink how you use technologies like the cloud, 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 are designed to help you advance digital transformation with prediction, security, automation and modernization capabilities. They let you develop applications once and deploy them anywhere, integrate security across your IT landscape and automate operations with intelligent workflows. Deploy them across any cloud to accelerate development, deliver seamless integration and enhance collaboration and efficiency.

IBM Cloud Paks are pre-integrated containerized software built on Red Hat OpenShift that are designed to help you develop and consume cloud services anywhere and from any cloud, so you can modernize with ease 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 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. It provides an open, fast, and more secure way to move core business applications to any cloud. It is a full stack, converged infrastructure with a virtualized cloud hosting environment that helps to extend applications to cloud.

IBM Cloud Paks are:

- **Portable:** IBM Cloud Pak are portable can be run anywhere. The portability of IBM Cloud Paks brings that applications are built to run on any hybrid cloud environment. Applications can run on-premises infrastructure, on public hybrid cloud infrastructure or in an integrated system leveraging 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:

- They provide a foundation to rapidly address business requirements and build new capabilities into your applications to provide your business a stronger competitive position.

- Build applications to run where they provide the best benefit to your business, on premise, private cloud, public cloud or hybrid multicloud.
- Provide additional efficiency by automating operations in hybrid multicloud environments.
- Integrate common industry components to quickly build, move, and manage your applications and data.
- Provide full software stack support, providing 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. Cloud Pak for Data helps you build a data fabric which connects the siloed data distributed across your hybrid cloud landscape. This product offers a wide selection of IBM and third-party services which span the entire data lifecycle.

IBM Cloud Pak for Business Automation

IBM Cloud Pak for Business Automation is a modular set of integrated software components, built for any hybrid cloud, 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 tooling, assign tasks to bots and track performance. With this offering, you can transform fragmented workflows, allowing you to stay competitive, boost efficiency and reduce operational costs.

IBM Cloud Pak for Watson AIOps

Innovate faster, reduce operational cost and transform IT operations (ITOps) across a changing landscape with an AIOps platform that delivers visibility into performance data and dependencies across environments. Embrace artificial intelligence, machine learning and automation to help ITOps 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 (MCM) as part of the IBM strategy to enable AI-powered Automation for IT operations and management.

IBM Cloud Pak for Integration

IBM Cloud Pak for Integration is a hybrid integration platform that applies the functionality 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. Cloud Pak for Integration's integration software unlocks business data silos and assets as APIs, connects cloud and on-premise 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 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 response. 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 – helping your team become more efficient and collaborative.

While all of the above Cloud Paks are certified to run on Intel based cloud services - whether private, public or hybrid – not all of them have been certified to run on IBM Power Systems based servers at this time. In the following section we detail which of the IBM Cloud Pak offerings are certified to run on IBM Power Systems based cloud environments. This list will change over time as additional Cloud Paks are tested and certified on IBM Power Systems server platforms.

5.3 IBM Cloud Paks Offerings on IBM Power Systems

As of the writing of this book, IBM has certified a portion of the IBM Cloud Pak solutions to run on IBM Power Systems servers. Figure 5-1 shows the IBM Cloud Paks capabilities currently supported on IBM Power Systems.

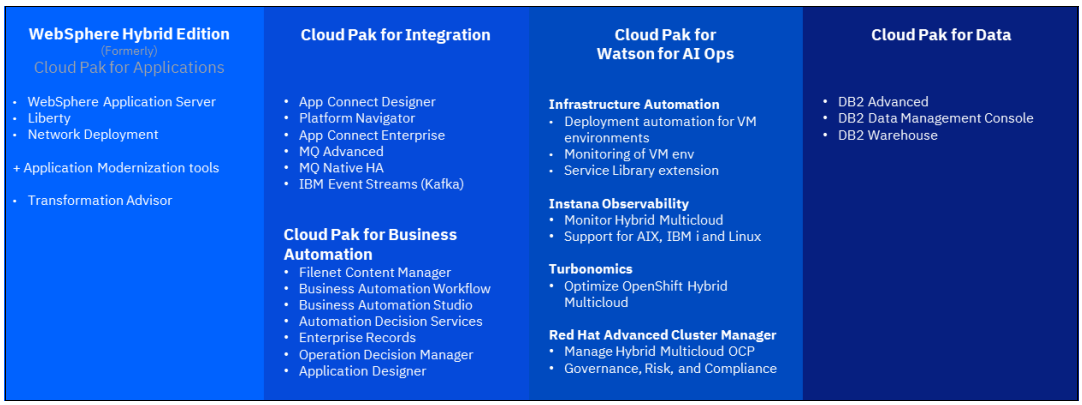


Figure 5-1 Cloud Pak offering on IBM Power Systems

Additional components are in the roadmap for support and certification and should be released in the near future. For further information about the current support for IBM Power Systems servers by Cloud Paks, contact your IBM representative.

5.3.1 IBM Cloud Pak for Data

Cloud Pak for Data is IBM offering to modernize customer environment. Cloud Pak provides their data lake with the latest analytics innovations, security, and the flexibility of hybrid cloud. This offering provides full suite of IBM analytics capabilities available with Cloud Pak for Data, without having to move data or rewrite any of your existing applications.

The integration of IBM Data & AI recognized solutions – such as IBM DataStage®, IBM Cognos, IBM Watson Studio, IBM Watson Knowledge Catalog, and more – into the IBM Cloud Pak for Data, reduces the time-to-value for business, lowers the TCO (Total Cost of Ownership), and helps ensure compliance, security and governance. This addresses the following Data Fabric use cases in a unique and collaborative platform:

- Multi-Cloud Data Integration
- Data Governance & Privacy
- Customer 360
- MLOps & Trustworthy AI
- Data Observability

Customers can query all data sources both inside and outside their data lake giving them a seamless single view of all data with complete security and without the need for data movement. They can then use these data sets for data science, machine learning at petabyte scale to gain faster insights and make better decisions. Figure 5-2 provides an overview of the capabilities of Cloud Pak for Data.

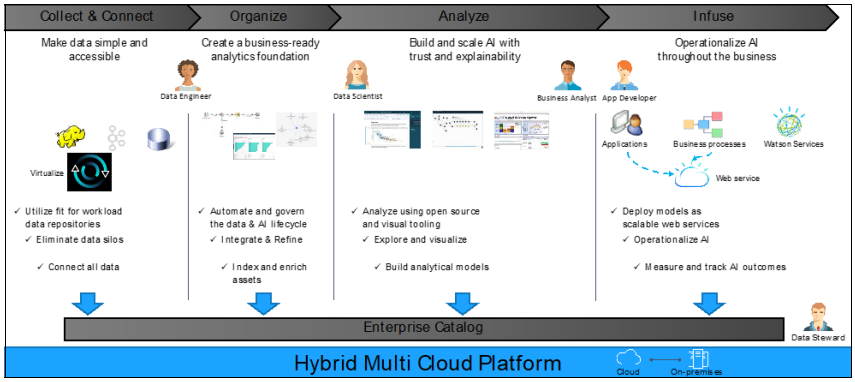


Figure 5-2 Cloud Pak for Data overview

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 additional detail on IBM Cloud Pak for Data see 5.4.2, “Cloud Pak for Data” on page 168.

For installation assistance see 6.2.2, “Installing Cloud Pak for Data on Red Hat OpenShift” on page 194.

For an example use case of IBM Cloud Pak for Data see 5.5.4, “Additional Db2 Use cases” on page 181 which provides implementation details.

More information about the requirements of the Cloud Pak for Data version 4.5 can be found in Hardware Requirements and here for Software Requirements.

5.3.2 IBM Cloud Pak for Business Automation

The IBM Cloud Pak for Business Automation enables process automation through the broadest set of AI-powered automation software. Cloud Pak for Business Automation 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. It provides the option of running the workload in any environment which best suits the customer needs private cloud, local cloud.

IBM Cloud Pak for Business Automation (CP4BA) capabilities include:

- Document processing.
- Content services.
- Decision management.
- Workflow automation.

Document processing

IBM Cloud Pak for Business Automation extracts data from structured, semi-structured, and unstructured documents. It automatically detects and corrects data extracted incorrectly.

Content services

IBM Cloud Pak for Business Automation allows for the classification, management, and access to these digital assets. It provides secure access to enterprise content from anywhere.

Decision management

IBM Cloud Pak for Business Automation automates the decisions with business rules. CP4BA rapidly adapts to change with business-friendly tooling and increases consistency and auditability of business policies. It integrates with predictive analytics for real-time response.

Workflow automation

IBM Cloud Pak for Business Automation is designed for human and automated activities to better manage internal workloads. Improve consistency across business operations with increased visibility and reduce cycle time to Increase straight-through processing.

Figure 5-3 provides an overview of Cloud Pak for Automation.

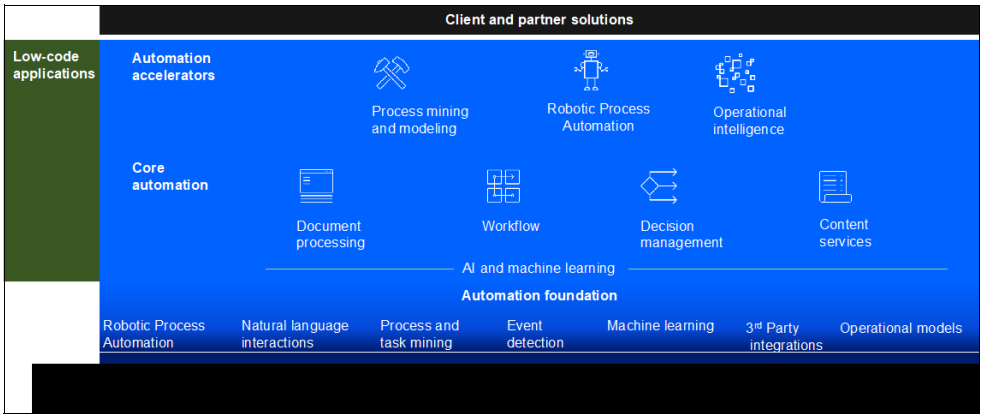


Figure 5-3 Cloud Pak for Automation

5.3.3 IBM Cloud Pak for Integration

IBM Cloud Pak For Integration (CP4I) is a hybrid integration solution. Cloud Pak For Integration 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 Cloud Pak for Integration runs on Red Hat OpenShift. It is a software solution offering that enables streamlining operations, workloads and clusters across multiple cloud environments. The software solution integrates with commonly used tools and applications that you may already have in place for operations. CP4I minimizes the workflow interruption to the cloud. CP4I takes advantage of an automation foundation set of capabilities that is consistent across all the Cloud Paks offered by IBM.

CP4I includes the following capabilities:

- API Management.
- Application integration.

- End-to-end security.
- Enterprise Messaging.
- Event Streaming.
- High speed data transfer.

IBM Cloud Pak for Integration *Create* quickly exposes data, events, microservices, enterprise applications and SaaS services as APIs through open standards. It manages rapidly organize, version, curate and publish any API through a full-lifecycle. CP4I Secure apply built-in and extensible policies to secure, control and mediate the delivery of APIs with unmatched scale. *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 Cloud Pak for Integration.

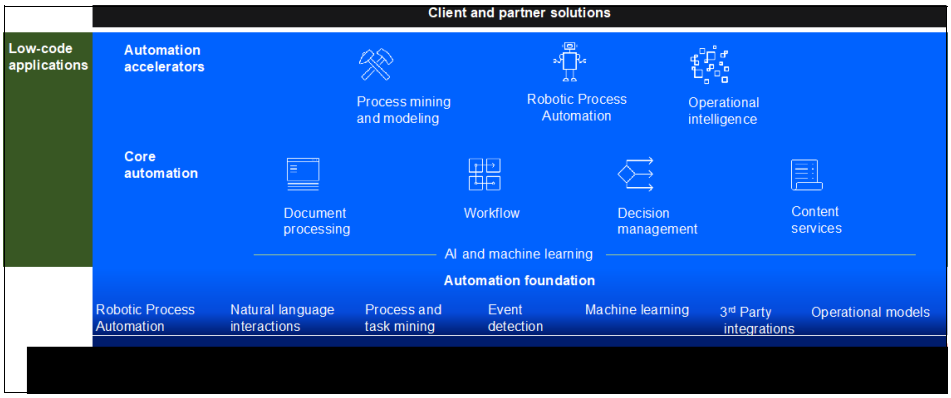


Figure 5-4 Cloud Pak for Integration

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 the mixture of AI and existing IT processes – like incident and problem management – which provides operational benefits such as predictive alerts and outage avoidance. Ultimately providing a better digital experiences for your customers.

IBM Cloud Pak for Watson AIOps provides simplification to operations management by:

- 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, it is a stand-alone capability module. Infrastructure Automation is built on Opensource Terraform and ManageIQ.

As it brings two distinct features:

- Managed services at its core have Cloud Automation Manager and its self-service capabilities to orchestrate resources which is now coupled with Terraform and Service Automation technology.

– Infrastructure management, previously known as IBM Red Hat CloudForms to discover, manage and automate the deployment of VMs, cloud services and Kubernetes clusters.

Managed services use Terraform and Ansible technology. 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 allows for drag-and-drop to publish services using out of the box integration with Ansible. Workflows connect Terraform with Ansible playbooks for configuration management that supports infrastructure-as-code (IaC) and GitOps best practices. It includes supported versions of Terraform that work across multiple cloud environments like RHEV, VMWare, OpenStack and across multiple cloud providers, like AWS, Microsoft Azure, Google and IBM Cloud.

Infrastructure management addresses the challenges of managing hybrid IT environments. It is based on ManageIQ and CloudForms. Based on client feedback, IBM focused on increasing the flexible integration capabilities given the Hybrid Cloud infrastructure. The organizations which want to get started with hybrid cloud, Infrastructure management capability helps them do exactly that. It integrates with Clouds, Containers, on premise or virtual infrastructure. It Discover inventory across virtualization, container, network and storage management systems, map relationships and listen for changes to build a rich model. It scan the contents of VMs, hosts, and containers, and combine with auto-discovery data to create advanced security and compliance policies.

Figure 5-5 shows the capabilities of Cloud Pak for Watson AIOps.

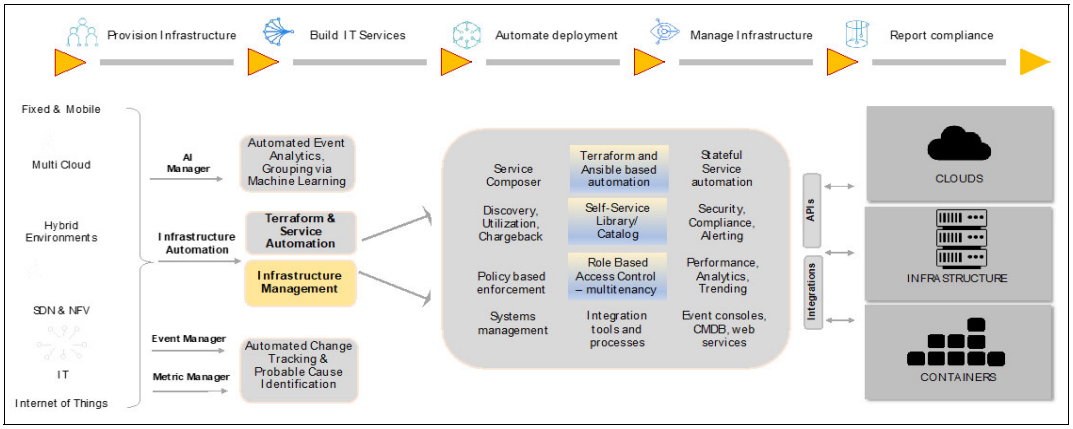


Figure 5-5 Cloud Pak for Watson AIOps

5.3.5 IBM Cloud Pak for WebSphere Hybrid Edition

IBM WebSphere Hybrid Edition combines all of the products in our WebSphere portfolio with application modernization tools. It's 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 ITOps. With this Cloud Pak, we can tap into shared automation services to get insight into how processes run. Visualize hotspots and bottlenecks, and pinpoint what to fix with event detection to prioritize which 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

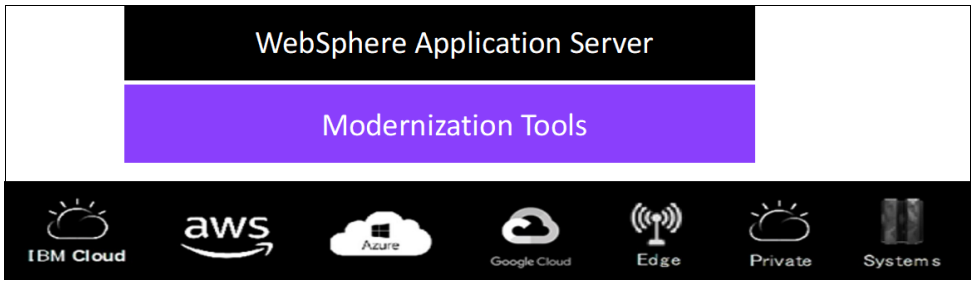


Figure 5-6 IBM Cloud Pak for WebSphere Hybrid Edition

WebSphere Hybrid Edition is unique in that 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 growth of the existing WebSphere Application Server (WAS) install base as well as providing for application modernization and new cloud native applications using the Liberty runtime. WebSphere Migration Toolkit can help move to new versions of WAS or move from traditional WAS to the Liberty profile.

5.4 Cloud Pak for Watson AIOps and Cloud Pak for Data

This previous section provided an overview of all of the Cloud Paks that are certified to run on IBM Power Systems servers. In this section, we take a deeper dive into two of those 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, as well as data observability, while also ensuring compliance, security, and governance.

In this section we will describe both Cloud Paks in more detail and their uses.

5.4.1 Cloud Pak for Watson AIOps

The IBM Cloud Pak for Watson AIOps provides a unique set of tools to assist you in designing and running AIOps on a cloud infrastructure. The functions provided are discussed in this section.

Infrastructure Management and Monitoring

The IBM Cloud Automation Manager solution automates 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 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 Cloud Pak for Data

As companies needs to become data driven and expand the potential of AI, they need to use data from diverse and complex sources, across multi-hybrid-cloud environments, also deal with different data formats and standards. With huge amounts of data produced every day, enterprise are challenged to understand what data really matters for their business. They are also challenged to process, govern, and manipulate all this data to guarantee trust and accessibility to entire enterprise, both technical and business users.

To simplify the use of the data, IBM introduced the IBM Cloud Pak for Data to implement the data fabric approach to accelerate the governance and the journey to AI.

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 multi-hybrid-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 artificial intelligence 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**: to make it easier to consume and access data.

- **Organize**: to create a trusted analytics foundation on data with business meaning.
- **Analyze**: to scale business insight with artificial intelligence everywhere.
- **Infuse**: to operationalize artificial intelligence 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 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 included so that users focus on innovation rather than repetitive tasks.

IBM Cloud Pak for Data and Data Fabric

We already know the data has being growing fast over the past few years and we expect this behavior will not change, instead, we believe it will grow at a continually expanding rate. The data complexity problem is huge, and increases with the data volume growth. The data growth and management challenge severely inhibits the ability of an enterprise to become data-driven and makes it difficult to get 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 need to 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 which will allow an enterprise to become data-driven. The data fabric provides an abstraction layer allowing data to be shared and used across a hybrid multicloud landscape, connecting data from various sources, such as on-premise data lakes and data warehouses, multiple cloud providers, and existing applications including both legacy and SaaS solutions, all while maintaining data observability. Figure 5-7 shows some of the benefits of utilizing a data fabric to connect your data together.

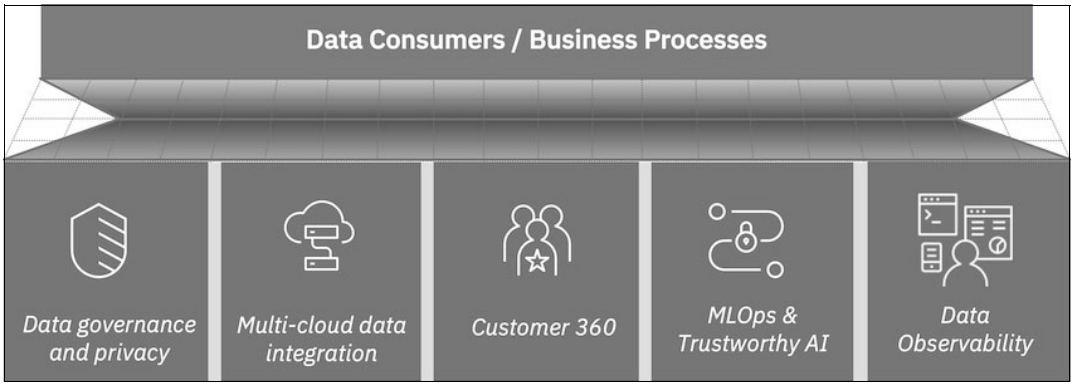


Figure 5-7 IBM data fabric approach

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 while reducing the costs of integration, including reduced bandwidth

costs and processing requirements by keeping and processing the data where it is and providing support for the full DataOps life cycle.

IBM Cloud Pak for Data helps you to implement a data fabric to support the use cases shown in Figure 5-7 on page 169 by bringing a unique data platform to your enterprise that 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 & Privacy

To ensure that data consumers are connected to the right data at the right time – and guarantee data trust – they need to have a be able to find the required data and be able to access that data.

We all understand the difficulty of finding data within the enterprise. There are a lot of different data sources with different data admin and accesses rules. There are legacy systems – sometimes without documentation – and there are also spreadsheets and datasets not even in a system but in business users machines. Your data often looks like the first image in Figure 5-8 – spread across multiple tables and rooms. Obviously finding important and current data is nearly impossible in this scenario.

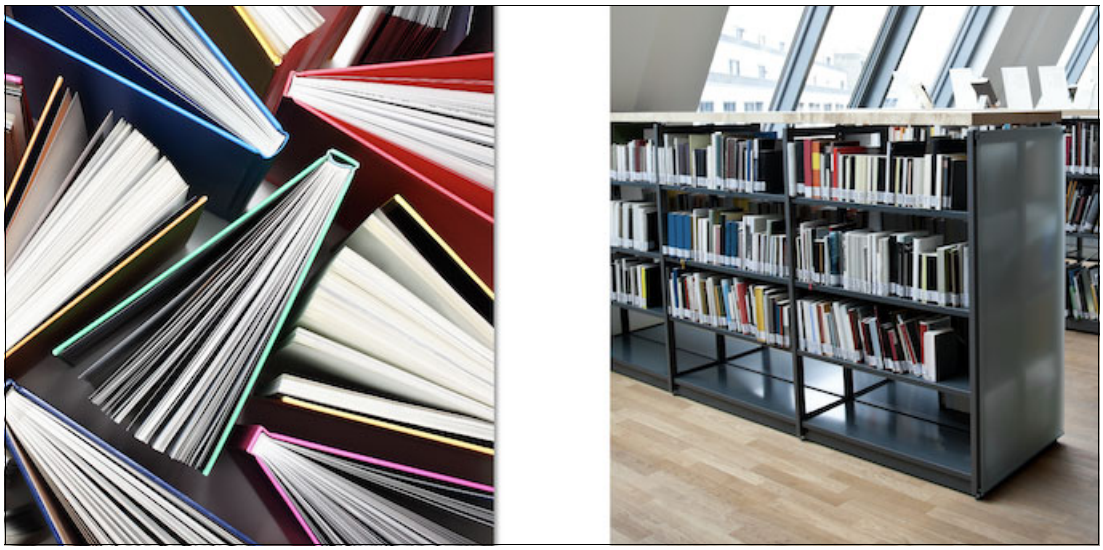


Figure 5-8 The Catalog importance

A data catalog can help by registering the data assets across the enterprise and organizing access to that data. The data catalog works similar to the bookshelves in a library with books organized on the shelves where the book readers can use the library's catalog to find the right book as seen in the second image in Figure 5-8.

The data catalog is designed to be 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. This call allow you to 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.

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 also facilitates anonymization of training data and test sets by to maintain the integrity of the data used.

Cloud Pak for Data implements an AI-augmented data catalog which allows business users to easily understand, collaborate, enrich and access the right data from a unique and centralized platform shareable by the entire enterprise while enabling access to data without having to move or copy it. This simplifies your data management processes and helps to ensure data trust and governance.

Multi-Cloud Data Integration

Enterprises are continuing to move to hybrid multicloud solutions and require a wide range of integration styles and techniques to access their data. Data needs to be extracted, ingested, streamed, virtualized and transformed using an extensive library of hybrid cloud data sources, all driven by automation and data policies that maximize performance while minimizing storage and egress costs.

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 creates a unified view of your enterprise data to enable consistency across your operations and. applications. This provides the ability to intelligently integrate and automate data engineering tasks while enhancing data integration to support your business requirements. This is illustrated in Figure 5-9.

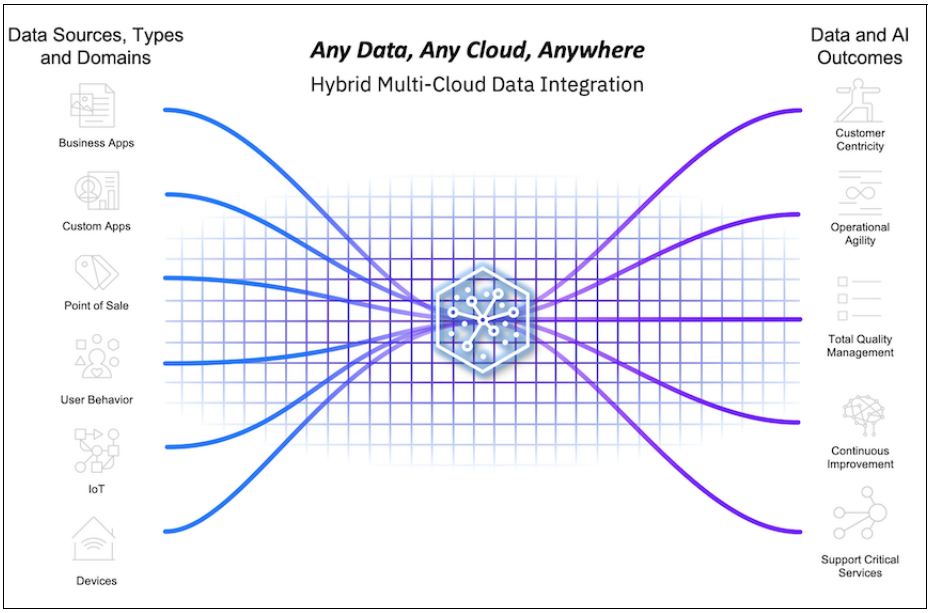


Figure 5-9 Multi-cloud Data Integration

Different data sources, data types and domains, data from business applications, custom applications, collected from devices, IoT, data about users behavior are processed by the Cloud Pak for Data. Cloud Pak for data helps extract, ingest, prepare, transform, and deliver

data across the enterprise. This all results in improved the outcomes for different initiatives utilizing data and AI.

Any data access or delivery process is automated and streamlined to help speed up the data delivery process. 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 resides, minimizing storage and egress costs.

To consolidate and simplify IT infrastructures, the 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, preventing delays and disruptions of mission-critical data through data resilience and easy data access.

Improved data integration flows making the best use of IBM DataStage for ETL (Extract, Transform, and Load), Data Virtualization and real-time capture to optimize access to many diverse data sources using extensive native connectors. Quality analysis and remediation can be natively added into data pipelines to avoid costly downstream processing. Cloud Pak for Data supports the full DataOps life cycle – governance, quality, master data, integration, and collaboration – integrated into a unique data platform.

Customer 360

Enterprise has different challenges with the customer data to be addressed, they have multiple systems and applications, data silos, multiple domains and lack of data quality and the customer is one single person that uses all these applications from this one single enterprise and suppose they have a complete view and understand about them.

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 – a customer 360. This is one single example, but enterprises have complex challenges based on the lack of data consistency and standardization and by having multiples copies of customers' data, in different formats, resulting in having to spend days or even weeks to analyze customer data.

Multiple systems and local processes requires enterprise to use manual reconciliation and manual remediation to improve the quality of the customer's data, also requires complex data integration processes to movement data, additionally they face lack of trust in data caused by the inconsistent business rules.

Customer 360 drives different enterprise iniatives to reduce costs and optimize productivity, by following perspectives:

- **Analytical**: Focused on Customer Care and on the single view of the customer to enable business answer deeper and more complex questions about the customer.
- **Governance**: Prepare and maintain customer data quality high and trustworthy.
- **Compliance**: Accelerate the compliance enforcement and fraud prevent, also guarantee data privacy.
- **Operational**: Infuse the single view of the customer into the applications, tools and systems, to be used at the right time by the business and providing hyper personalization.
- **Prescriptive**: Design for better outcomes e.g. predict churn, define next best offer or next best action, by using AI powered patterns and algorithms.

To provide a comprehensive Customer 360 view, 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 and applies entities resolution algorithms and ML-powered probabilistic matching to increases the results. It enables users spend more time on applying AI and Analytics to business challenges versus wasting time on hunting quality data and consolidating the customer view. Also 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 vary applications, tools and systems.

MLOps & Trustworthy AI

Ensure AI operationalization and trust in machine learning models is a challenge today, enterprises must deal with AI lifecycle and needs to guarantee the quality of the AI models deployed. Additionally, the quality of the test data sets used during the training phase and the fairness of the model to avoid bias is also important.

The IBM data fabric approach to ML Ops & Trustworthy AI is based on three important AI lifecycle phases, implemented by the Cloud Pak for Data (see Figure 5-10).

- **Data**: Collect and prepare data by ensuring the governance, the quality and the trust.
- **Model**: Build, deploy and monitor models by guarantying fairness, robustness and explainability.
- **Process**: Use automation to drive consistency, efficiency, and transparency for AI.

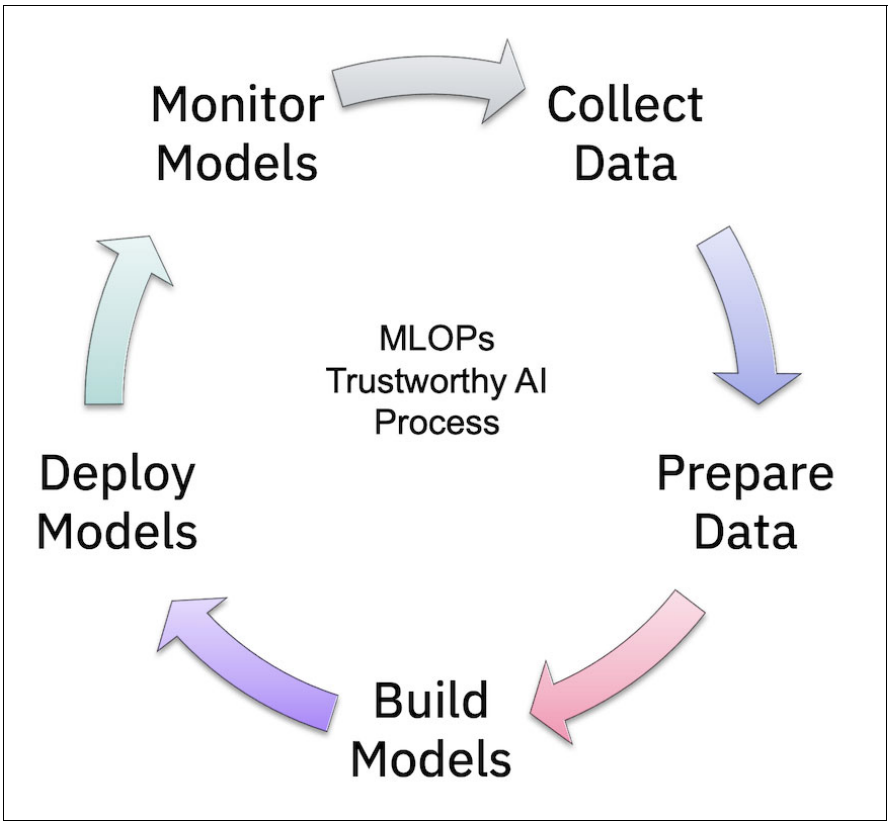


Figure 5-10 MLOps Trustworthy AI process

The data catalog provided by Cloud Pak for Data provides automatic application of policies and rules, while ensuring the privacy, and security. while allowing you 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 start to build the AI models, Cloud Pak for Data provides no code and low code services, as well as the support of 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 built while providing efficiency improvements for data scientists.

Several end-to-end tools enables the 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 ensuring only fair models are used in production. This provides equitable outcomes from models across all groups.

Explainability is provided by Cloud Pak for Data to understand the model outcomes and decisions made from it, the features that most influence on the prediction are shown, also the ability for what-if analysis are available to better explain the results.

The model drift is monitored over time with the ability to automatically retrain the model. This allows the adaptation to changing model parameters that can guarantee trust on models while ensuring that business objectives are being met. All these tools and features put together, enables trusted models that can be deployed quickly into production, are continuously improved, and stay reliable over time.

Pipelines enable automation of the AI flow and orchestration of entire lifecycle, retrieving fresh training data, retraining the model, and deploying into production. Automatically capture model metadata and lineage to guarantee understanding and governance of the model and risks, increasing 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.

Data Observability framework expands Data Fabric use cases by including remediation of data quality issues in data governance. This includes:

- Monitoring in motion data 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 historical baseline with the common data pipeline's behavior, which is continuously compared with the behavior of the data in motion.
Alert	**Alert** 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 issues. **Scale-up** the resources to process more data if volume increases. **Route** bad-data with low quality to staging area,

Retrain models to improve accuracy based on the performance monitoring.

These steps are designed to automatically resolve issues in your Data Fabric automatically through automation.

When you combine Data Observability with Instana for enterprise application observability (see 2.6.2, “Instana” on page 41 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 see: https://www.ibm.com/data-fabric. You can also find more information on use cases for IBM Cloud Pak For Data and sign up for a free trial.

5.5 Db2 workloads on Cloud Pak for Data on IBM Power Systems

IBM Db2 database is a world class, enterprise relational database management system (RDBMS). Db2 provides advanced data management and analytics capabilities for your online transactional workloads (OLTP). Traditionally enterprises run the IBM Db2 database in a dedicated on-premises environment, often hosted on a LPAR running on an IBM Power Systems server using either IBM AIX or Red Hat Enterprise Linux for IBM Power Systems Little Endian (Red Hat Enterprise Linux ppc64le) operating systems. Utilizing 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 likely to be a good option for all Db2 workloads, but there are some workloads that would benefit from a cloud native containerized approach. IBM also provides tooling 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 Z 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 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 *IBM Cloud Pak for Data Version 4.5: A practical, hands-on guide with best practices, examples, use cases, and walk-throughs*, SG24-8522 chapter 2. For additional information you can also reference this IBM Documentation web page.

Integrating a Db2 database with 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.
- You need to 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 Cloud Pak for Data, you can deploy Db2 by using a cloud-native model, which may provide the following benefits:

- Less operating system patching needed due to the reduction of 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 etc.
- A rich ecosystem that includes a Data Management Console, REST, and Graph.
- Extended availability of Db2 with a multi tier resiliency strategy.
- Support for software-defined storage, such as Red Hat OpenShift Data Foundation, IBM Storage Scale CSI, and other world leading storage providers.
- Reduction of the amount of infrastructure, i.e. number of LPARs, amount of storage, etc.,

You can create a new Db2 database in your Red Hat OpenShift environment using IBM Cloud Pak for Data or you can quickly move an existing on-premises Db2 LUW database which is running on an IBM Power Systems Linux server to IBM Cloud Pak for Data using the Db2-click-to-containerize automation tooling. An example of how to install Db2 with IBM Cloud Pak for Data is shown in section “Running Db2 workloads on IBM Cloud Pak for Data on IBM Power Systems” on page 193.

In the rest of this chapter we explore the features of Db2 in the IBM Cloud Pak for Data and the benefits your enterprise may see.

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 Cloud Pak for Data, which enables them to work seamlessly with the 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.

See 5.5.3, “Db2 Data Management Console” on page 180 for more information on the integrated database console.

Initial setup and configuration considerations

Setting up the Db2 service and databases in Cloud Pak for Data requires some extra steps and considerations compared to some of the other Cloud Pak for Data services.

Before installing the Db2 service, consider the use of 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 recommended for production and is important for databases that are performing heavy workloads. Setting up dedicated nodes for your Db2 database involves Red Hat OpenShift taint and toleration to provide node exclusivity. You also must create a custom security context constraint (SCC) that is used during the installation.

After installing the Db2 service and before creating 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. Doing so allows you to 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 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 use 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 advised to consider the use of separate locations, especially among the user data, transactional logs, and backup data.

In section 6.2, “Running Db2 workloads on IBM Cloud Pak for Data on IBM Power Systems” on page 193 we demonstrate how to install a Db2 database on IBM Power Systems using a container-native platform using IBM Cloud Pak for Data on Red Hat OpenShift.

Learn 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
  - Post installation setup for the Db2 service
- Video: Db2 on IBM Cloud Pak for Data platform
- Blog: The Hidden History of Db2

5.5.2 Db2 Warehouse

IBM Db2 Warehouse is an enterprise ready data warehouse that is used globally. Db2 Warehouse provides in-memory data processing, columnar data store, and in-database analytics for online analytical processing workloads (OLAP).

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 the suitable data warehouse environment. For a single node, the warehouse uses symmetric multiprocessing (SMP) architecture for cost-efficiency. For two or more nodes, the warehouse is deployed by using an MPP architecture for high availability and improved performance.

By using the Db2 Warehouse operator and containers in Cloud Pak for Data, you can deploy Db2 Warehouse 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, RESTAPI, and Graphing tools.
- Extended availability of Db2 Warehouse with a multi tier 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 integrated with 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, if you need to 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 is 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.

See 5.5.3, “Db2 Data Management Console” on page 180 for more information on the integrated database console.

Initial setup and configuration considerations

Setting up the Db2 Warehouse service and data warehouse databases in Cloud Pak for Data requires some extra steps and considerations compared to some of the other Cloud Pak for Data services.

Before installing the Db2 Warehouse service, consider the use of 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 taint and toleration 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 be establishing an inter-POD communication network. Also, create a custom security context constraint (SCC) that is used during the installation.

After installing the Db2 Warehouse service and before creating your data warehouse database, consider disabling the default automatic setting of interprocess communication (IPC) kernel parameters so that you can set them 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. Doing so allows you to 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 Cloud Pak for Data cluster. You can choose to use the SMP or MPP architectures with the following configurations:

- Single physical node with one logical partition (default).
- Single physical node with multiple logical partitions.
- Multiple physical nodes with multiple logical partitions.

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 it is advised to consider the use of separate locations, especially among the user data, transactional logs, and backup data.

After the data warehouse database is created, you can now 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:

- IBM Documentation:
  - 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 Db2 Data Management Console

The 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 Cloud Pak for Data. These integrated databases include Db2, Db2 Warehouse, Db2 Big SQL, and Data Virtualization, which you can manage and monitor from a single user interface 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 Cloud Pak for Data integrated Db2 database.

Using the Db2 Data Management Console

The console home page provides an overview of all of the 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 Data Management Console.

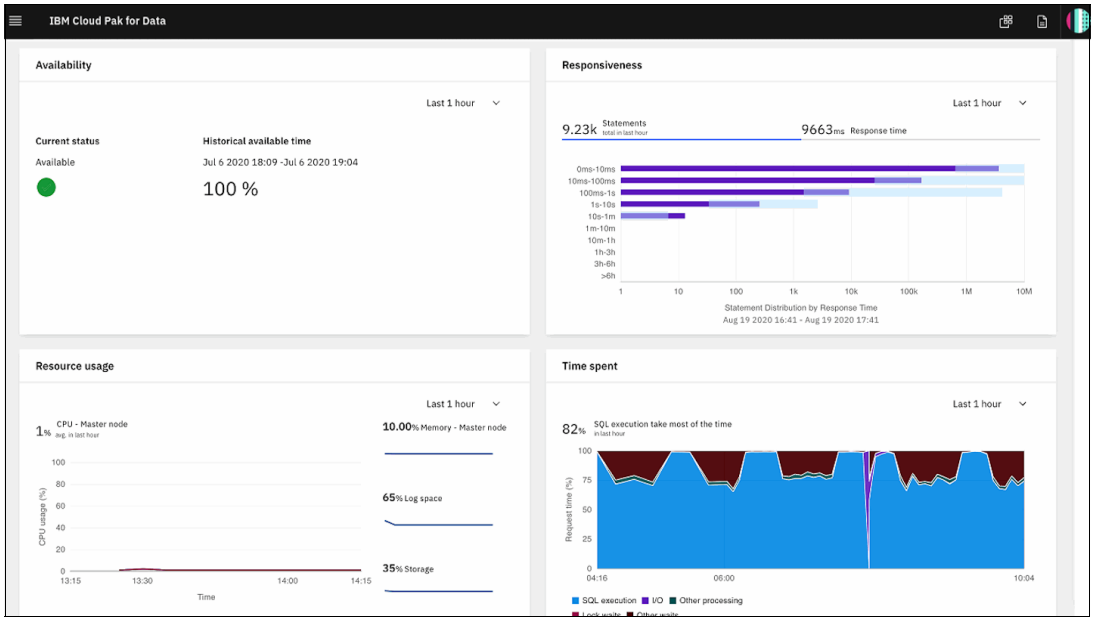


Figure 5-11 Db2 Data Management Console Summary Page

From the console, you can perform the following tasks:

- Explore integrated databases through its 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 a Cloud Pak for Data deployment.

To provision an instance, you first select the plan size for the compute resources: small, medium, or large. Then, you configure the storage resources by providing the storage class and the amount of storage for your persistent storage.

When the console instance is provisioned, you can start to 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 Documentation:
  - 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 utilizing DB2 in an IBM Cloud Pak for Data environment that we have not explored. DB2 can be integrated into many workflows within the Cloud Pak for Data including Data Virtualization, Watson AI and IBM Cognos® reporting. Section 6.2, “Running Db2 workloads on IBM Cloud Pak for Data on IBM Power Systems” on page 193

Many of these 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

6

Use Cases

This chapter presents a number of use cases for running applications on IBM Power systems that take advantage of the performance capabilities of the platform.

The use cases shown are:

6.1 AI Inferencing with Red Hat OpenShift and IBM Power Systems Power10

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 usually requires numerous, computationally expensive multiplications of large matrices. This is especially true for deep learning models that are commonly used for tasks such as image recognition, speech to text, natural language processing, and time series forecasting.

The IBM Power Systems 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 Power Systems 10 nodes and leveraging the processor's Matrix-Multiply Assist (MMA) capabilities.

6.1.1 Matrix-Multiply Assist

Matrix-Multiply Assist (MMA) is a set of instructions and data types that are designed to accelerate matrix multiplication on the IBM Power Systems Power10 processor. Packages that are optimized to leverage MMA deliver 5X faster AI inferencing per socket on IBM Power Systems E1080 compared to IBM Power Systems 980[1].

Existing AI inferencing workloads do not need source code changes to leverage MMA on IBM Power Systems Power10; however to see a performance improvement from MMA you need to 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 popular packages for machine learning on IBM Power Systems. The packages are designed to be installed within a Conda environment. More detailed information about Conda is available at this link – https://docs.conda.io/en/latest/.

The main Open-CE Github page (https://github.com/open-ce) provides users the ability to build the open-ce packages themselves. For those users that want the precompiled binaries, there are multiple organization that distribute precompiled Conda packages.

One company that distributes precompiled packages is Rocket Software. A benefit of Rocket Software's RocketCE distribution is that these packages are built to use the MMA capabilities of the IBM Power Systems Power10 processor. More information about the latest releases can be found in the RocketCE for IBM Power Systems forum.

6.1.3 ONNX Runtime

ONNX Runtime is a cross-platform accelerator for AI inferencing. It can be used with models built from PyTorch, TensorFlow and many other frameworks. The accelerator provides performance improvements by leveraging both hardware capabilities and graph rewrites.

[1] https://dach.tdsynnex.com/ch/blog/wp-content/uploads/2021/10/IBM-Power-E1080-Client-Presentation-Switzerland.pdf

When ONNX Runtime is obtained from RocketCE, it will take advantage of the MMA capabilities that are included in the IBM Power Server Power10 processor.

6.1.4 Inferencing Engine Tutorial

This section describes how to build and deploy a 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 Power Systems Power10 node; it uses ONNX Runtime from RocketCE to leverage the IIBM Power Systems Power10 MMA technology.

The files referenced in this tutorial are available here:
https://github.com/gpadillax/ai-inference-p10/

The classifier uses a pretrained ResNet model. More information about the pretrained ResNet model is located at:
https://github.com/onnx/models/tree/main/vision/classification/resnet.

Conda environment

The classifier runs within a Conda environment. In order to utilize MMA, the environment needs to ensure that ONNX Runtime is obtained from RocketCE.

Example 6-1 shows the environment.yaml for the Conda environment. Because RocketCE is listed first, packages in that library will be preferred over those in conda-forge. In addition, the dependency for *onnxruntime* explicitly states that the rocketce-1.6.0 version 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 will be used to create the Conda environment as part of the container build. The classifier will execute within the environment.

Container file

The container build initializes a Conda environment. The complete container file is named ‘Dockerfile’, and is available from our github repo at the classifier path in https://github.com/gpadillax/ai-inference-p10/. You can download the project from GitHub and build the container 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. 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 Systems is documented here.

The -b option of the installer script causes all of the agreements to be accepted without prompting, and the -p option specifies the directory to install miniconda information.

Because each RUN statement creates a new layer, the commands to download the installer, execute the installation, and remove the installer program from the file system must be included in the same RUN command. This reduces 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 using the *onnxruntime_env.yaml* file (from Example 6-1 on page 185) 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 Create 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. They are:

- The model file and class labels are downloaded and saved in the container image.
- The python scripts are copied into the image.
- The port 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 Activate conda environment

```
CMD bash --login -c 'conda activate onnxruntime && python app.py'
```

Classifier service

The classifier service is implemented using the Flask web framework and ONNX Runtime. Because the Conda environment has been created with packages that support MMA, the classifier will receive a performance benefit without any code modifications for IBM Power Systems Power10.

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 Initialize service

```
import onnxruntime as ort
INFERENCE_SESSION = ort.InferenceSession(
    ONNX_MODEL_FILE_PATH, providers=["CPUExecutionProvider"]
)
```

Once Conda is installed, the conda init command (shown in Example 6-2 on page 186) is invoked to initialize Conda when the interactive shells begin.

The logic for making a prediction is made up of 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 Run model to make prediction

```
@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 has been built for the classifier, the container can be deployed to an Red Hat OpenShift Container Platform.

Label Red Hat OpenShift nodes with MMA capabilities

In many environments, only a subset of the cluster's nodes have IBM Power Systems Power10 MMA capabilities. Labels and node selectors are used to ensure that the containers for AI inferencing run on nodes that support MMA. This ensures that these workloads run as fast as possible.

A label can be added using the Red Hat OpenShift UI by going to the Compute → Nodes tab. After clicking on the IBM Power Systems Power10 node, the "Edit node" option from the actions drop down can be used to modify the labels for the node. See Figure 6-1.

For our example, we added the label "ai.inference.accelerator=mma" to the IBM Power Systems Power10 node.

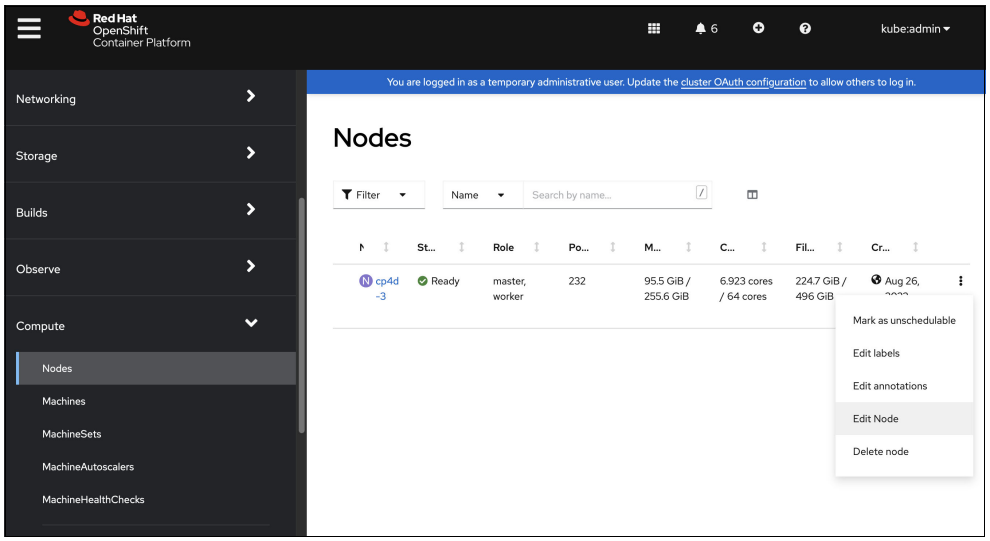


Figure 6-1 Computenode edit

Figure 6-2 shows the "mma" label attached to a node.

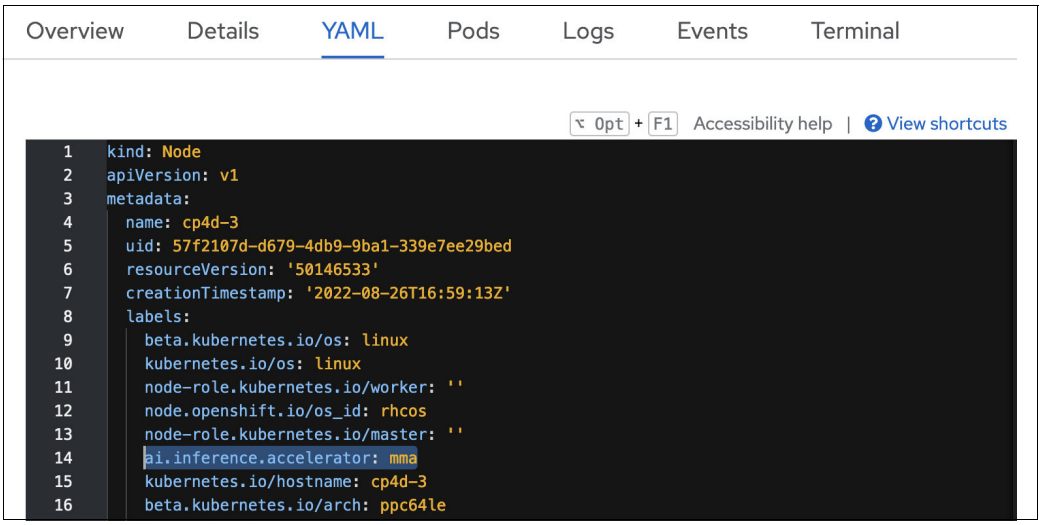


Figure 6-2 Label showing mma example

Red Hat OpenShift Deployment

To deploy, clone the git repo https://github.com/gpadillax/ai-inference-p10 which includes the yaml files needed. These will be described in the following sections. The deployment.yaml file, shown in Example 6-7, shows the description of what is used to deploy the inference container. This includes the nodeSelector which ensures that the PODs created will be deployed to nodes that have been 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
```

Now, create the new project using **cli oc** and then proceed to create the deployment using the deployment.yaml file from cloned git repo as seen in Example 6-8.

Example 6-8 Create project and create 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 MiB/s, 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
```

Create Service

Services for this example are created with NodePort that maps to the container's port 5000. This allows clients address the service, without having to address individual PODs.

The yaml for the service is shown in Example 6-9.

Example 6-9 Create service yaml

```
apiVersion: v1
kind: Service
metadata:
  name: inference
  namespace: inferencep
spec:
  selector:
    app: inference
  type: NodePort
  ports:
    - protocol: TCP
      port: 5000
      targetPort: 5000
```

To create the service using cli, use **oc create -f** and the file name, service.yaml as shown in Example 6-10.

Example 6-10 Create service command

```
# oc create -f service.yaml
service/inference created
```

Create a Route

In order to access the service from outside the cluster, a route must be created. Creating the route will cause a URL to be assigned to the service, which can be found by examining the location in the route details page. The yaml to create a new route is shown in Example 6-11.

Example 6-11 Create route

```
apiVersion: route.openshift.io/v1
kind: Route
metadata:
  name: inference
  namespace: inferencep
spec:
  path: /
  to:
```

```
    kind: Service
    name: inference
  port:
    targetPort: 5000
```

Create the route using **cli oc create** and route.yaml file as shown in Example 6-12.

Example 6-12 Create 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 wget command

```
wget
https://upload.wikimedia.org/wikipedia/commons/thumb/c/c2/Macchina_per_scrivere_el
ettromeccanica_-_Museo_scienza_tecnologia_Milano_10947.jpg/512px-Macchina_per_scri
vere_elettromeccanica_-_Museo_scienza_tecnologia_Milano_10947.jpg -O /tmp/tw.jpg
```

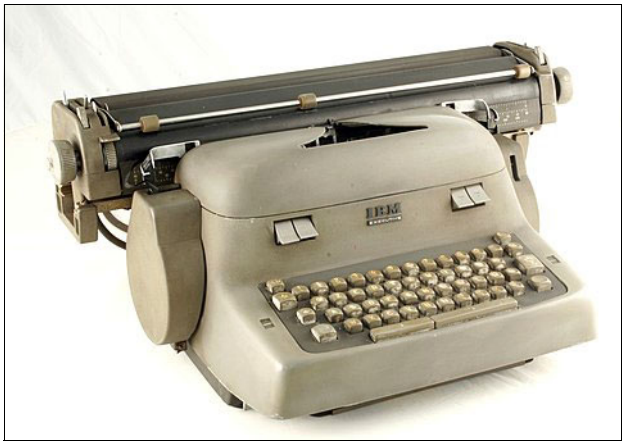


Figure 6-3 Photo to be classified

Now you can execute 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. This is shown in Example 6-14.

Example 6-14 Rest call to classifier

```
# oc get route
NAME        HOST/PORT                                          PATH   SERVICES
PORT   TERMINATION   WILDCARD
inference   inference-inferencep.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-inferencep.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 Power Systems Power10 In real enterprise applications, more sophisticated workflows are needed to train, deploy, and monitor AI solutions. Kubeflow is an open source toolkit for Machine Learning Operations in containerized environments such as Red Hat OpenShift. A brief overview of the solution is described here.

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 Systems. We can be contacted here.

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, targeting an IBM Power Systems Power10 system with MMA capabilities.

6.2 Running Db2 workloads on IBM Cloud Pak for Data on IBM Power Systems

In this section we will explore a new way of running Db2 databases on IBM Power Systems by using a container-native platform.

Instead of the traditional way of hosting one or more Db2 databases on a LPAR on IBM Power Systems running IBM AIX or Red Hat Enterprise Linux for IBM Power Systems Little Endian (Red Hat Enterprise Linux ppc64le) operating systems, we are changing 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 (OCP).

Switching to a container-native platform such as IBM Cloud Pak for Data on Red Hat OpenShift brings various benefits such as:

- Reduction of the amount of infrastructure, i.e. number of LPARs, amount of storage, etc.
- Less operating system patching needed due to the reduction of infrastructure.
- Faster time to value when deploying Db2 databases.
- Faster backup and restore through snapshot-based mechanisms etc.

Db2 running on Cloud Pak for Data was discussed earlier in section 5.5.1, “IBM Db2” on page 176.

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, namely: -

- A Red Hat OpenShift 4.10 cluster running on 8 IBM Power10 LPARs:
  - Three master nodes, four worker nodes, and a bootstrap node.
  - The IBM Power Systems Power10 LPARs are running on dedicated cores for the master and worker nodes.
  - The bastion and the bootstrap node is running with 1:10 PU to VP ratio.
- A bastion host running on a IBM Power Systems Power10 LPAR:
  - Running Red Hat Enterprise Linux 8.5 ppc64le Linux OS that provides DNS, load balancer, NFS and DHCP services to the Red Hat OpenShift cluster.
- An IBM Storage Scale (Previously Spectrum Scale) 5.1.5 storage cluster:
  - Running on 4 IBM Power Systems Power10 LPARs each with Red Hat Enterprise Linux 8.6 ppc64le Linux.
  - 2 GUI nodes and 2 NSD nodes providing 2 GPFS file systems running on 7 LUNs with 500 GB each.
  - gpfs0 has 2TB of storage (coming from 4LUNs), and gpfs1 has 1.5 TB of storage (coming from 3 LUNs).
- Storae Scale CNSA 5.1.5 has been setup on the Red Hat OpenShift cluster and been connected to external Storage Scale storage cluster and the gpfs0 file system has been remotely mounted.
- A storage class named ibm-spectrum-scale-csi-fileset for the Storage Scale CNSA.
- An x86-based VM running Red Hat Enterprise Linux 8.6 that we will use to run the Cloud Pak for Data installer.
- All the LPARs have Internet connectivity.

Finally, we have setup a Db2 LUW 11.5 instance on one of the LPARs of the Storage Scale storage clusters and created the Db2 SAMPLE database on the instance. We are going to clone this Db2 SAMPLE database into a Db2 instance running in a container on the Cloud Pak for Data on Red Hat OpenShift.

6.2.2 Installing 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. In the following we will explain, what is going to happen in each step.

You can skip these explanations and go directly to sections below if you want to type in the commands directly and if you don't need the background information on why we are doing these steps.

Step 1 - Setting up a client workstation

First, we need to setup an x86-based client workstation that runs the Cloud Pak for Data CLI (cpd-cli) command line tooling and the olm-utils Ansible-based containerized software package it includes (IBM Power Systems and z Linux based client workstations are not supported at this time for the olm-utils). The cpd-cli command line interface will enable us to install the Cloud Pak for Data software on the Red Hat OpenShift cluster in a convenient way.

Note: The cpd-cli command line will pull a containerized software package called olm-utils at first time. The olm-utils container image packages Ansible-based installation scripts that make it very easy to administrate and install Cloud Pak for Data and its services such as Db2.

To be able to run the olm-utils, we need a container environment installed on the client workstation, such as podman or docker in this example we are going to use the podman.

Finally, we need to setup the Red Hat OpenShift CLI (oc) to being able to interact with the Red Hat OpenShift cluster directly outside of the cpd-cli tooling.

Step 2 - collecting required information

Second, we need to gather some required information prior to be installing the Cloud Pak for Data software.

We first need a valid Cloud Pak for Data software license, i.e. an IBM entitlement API key. There are two ways to obtain this license key.

For a period of up to 60 days, you can obtain a 60-days trial key for Cloud Pak for Data from this URL:

`https://www.ibm.com/account/reg/us-en/signup?formid=urx-42212`.

If you need a longer time, please work with your IBM Sales representative to get an IBM standard evaluation license for Cloud Pak for Data.

Once either of the requests has been processed, login with your IBM ID to the following URL: `https://myibm.ibm.com/products-services/containerlibrary`.

Then 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 on page 195.

Example 6-16 Example of an IBM entitlement key

```
eyJhbGciOiJIUzI1NiJ9.eyKpc3MiOiJJQkOgTWFya2VOcGxhY2UiLCJpYXQiOjE2MTc3OTE2MOcsIm
pOaSI6IjM1MDU2NzBjMTI4NTRiZmM4MTQyN2E5OFJlOWFlNjUwInO.pCc4KoA22c9n6goFsw1R5GVrf
f3nyRnqNOTBYN6P-cg
```

Now, we need to choose which Cloud Pak for Data components we want to install on our cluster. As this chapter deals with running Db2 on Cloud Pak for Data, we are going to chose all the Db2-related services (db2oltp, db2wh, dmc), the mandatory services (cpfs, cpd_platform) plus one optional service (scheduler) i.e. the scheduler service that will allow us to set and enforce quotas for services running on the CPD platform. An example list of components is shown in Example 6-17.

Example 6-17 Example for the list of components

```
COMPONENTS=cpfs,scheduler,cpd_platform,db2oltp,db2wh,dmc
```

Finally, we need to setup a couple of more installation variables such as:

- Red Hat OpenShift cluster access details.
- Red Hat OpenShift projects (aka namespaces) we are going to create for the CPD cluster.
- Storage classes we are going to use (in our case we are going to use Storage Scale Container Native, but NFS is also supported).
- The IBM Entitlement key (see Example 6-16).
- The CPD version we are going to install (4.6.0).
- The list of components we are going to install (see Example 6-17 above).

Step 3 - preparing the Red Hat OpenShift cluster

Now, we are going to **prepare** our Red Hat OpenShift cluster before we can start the installation of Cloud Pak for Data.

This only needs to be done once. The preparation step does the following actions.

- Updates the global image pull secret of our Red Hat OpenShift cluster using the information from our 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 (i.e. pids_limit is equal or higher to 12288) of the CRI-O container runtime 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

Having done steps 1 to 3, we are now ready to start the installation of Cloud Pak for Data and the Db2 related services on our Red Hat OpenShift cluster.

Setting up a client workstation

1. Get an x86-based Linux VM or bare-metal server. In this example we use a Red Hat Enterprise Linux 8.6.

2. Login to the VM as root user and verify the Red Hat Enterprise Linux version and x86_64 architecture see Example 6-18.

Example 6-18 Verify 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 Install 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 Install screen

```
# yum install -y --nogpgcheck
   https://dl.fedoraproject.org/pub/epel/8/Everything/x86_64/Packages/s/scre
   en-4.6.2-12.el8.x86_64.rpm
```

5. Install oc client. See Example 6-21.

Example 6-21 install oc client

```
# wget https://mirror.openshift.com/pub/openshift-v4/x86_64/
   clients/ocp/4.10.34/openshift-client-linux.tar.gz
# tar -xzvf 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 new user cp4d and change to the new user. See Example 6-22.

Example 6-22 create user

```
# useradd cp4d
# su - cp4d
```

7. Install cpd-cli as user cp4d as shown in Example 6-23.

Example 6-23 install cpd-cli

```
$ wget
https://github.com/IBM/cpd-cli/releases/download/v11.3.0/cpd-cli-linux-EE-11
.3.0.tgz
$ tar -xzvf cpd-cli-linux-EE-11.3.0.tgz
```

8. Add the two lines shown in Example 6-24 to your ~/.bash_profile file and source the file.

Example 6-24 Source 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

1. Obtain your IBM entitlement API key. Login with your IBM ID to the following URL: https://myibm.ibm.com/products-services/containerlibrary. Then 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.
2. Setup a cpd_vars.sh file with the installation environment variables and source the file.

   You need to adapt the values for OCP_URL, OCP_PASSWORD, OCP_TOKEN and IBM_ENTITLEMENT_KEY to match with your environment as seen in Example 6-25.

Example 6-25 Source cpd-vars file

```
$ cat cpd_vars.sh
export OCP_URL="api.cp4d-1.rtp.raleigh.ibm.com:6443"
export OPENSHIFT_TYPE="self-managed"
export OCP_USERNAME="kubeadmin"
export OCP_PASSWORD="3rgHV-9XtKz-383hW-r9bgt"
export OCP_TOKEN="sha256~3_klXKIdDwVkoNQTHMi6axxxxxxxxxx"
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.eyJpc3Mixxxxxxxx
export VERSION=4.6.0
export COMPONENTS=cpfs,scheduler,cpd_platform,db2oltp,db2wh,dmc
$ source ./cpd_vars.sh
Preparing the 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. Login to the Red Hat OpenShift cluster via the CLI **manage login-to-ocp** command. See Example 6-26.

Example 6-26 Login to the Red Hat OpenShift cluster

```
$ cpd-cli manage login-to-ocp --token=${OCP_TOKEN} \
--server=${OCP_URL}
KUBECONFIG is /opt/ansible/.kubeconfig
Logged into "https://api.cp4d-1.rtp.raleigh.ibm.com:6443" as
"kube:admin" using the token provided.
```

4. Modify the global pull secret using the command in Example 6-27.

Example 6-27 Modify 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 seen in Example 6-28. Wait until all worker nodes have been rebooted.

Example 6-28 modify crio setting

```
$ cpd-cli manage apply-crio --openshift-type=${OPENSHIFT_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 have been rebooted.

Example 6-29 modify db2 kubelet settings

```
$ cpd-cli manage apply-db2-kubelet \
   --openshift-type=${OPENSHIFT_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 seen in Example 6-30.

Example 6-30 apply 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 as shown in Example 6-31.

Example 6-31 create 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 with the command shown in Example 6-32.

Example 6-32 get 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: j8Ug4OOeK8vN
```

10. Login to the IBM Cloud Pak for Data web GUI URL with the username and password from above as shown in Figure 6-4.

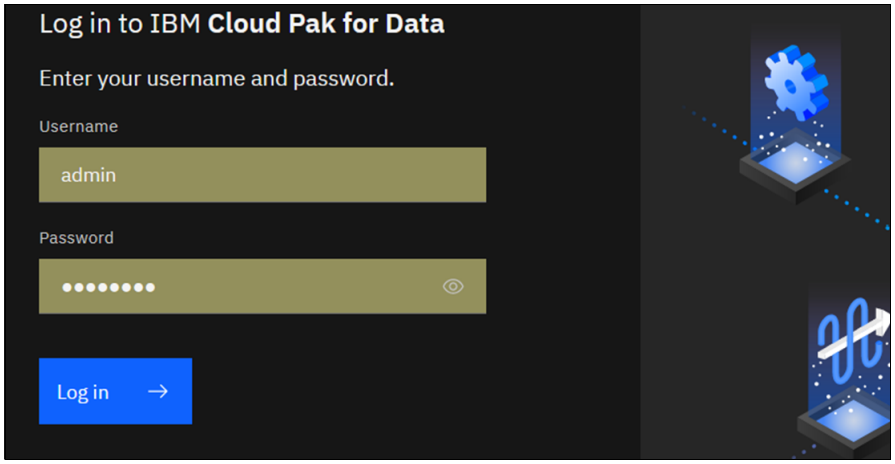


Figure 6-4 IBM Cloud Pak for Data login page

You will arrive at the IBM Cloud Pak for Data home page seen in Figure 6-5.

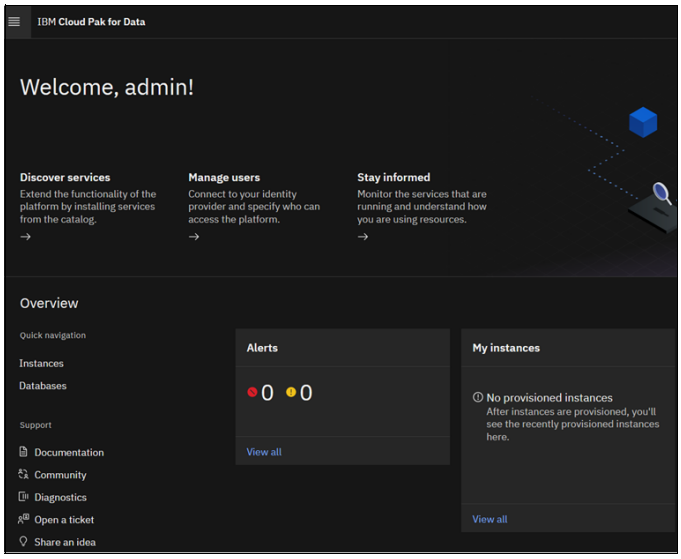


Figure 6-5 CPD home page

6.3 GitOps for system configuration

GitOps principles helps providing continuous deployment, which can also be used to implement Infrastructure as Code. See “GitOps for Red Hat OpenShift node tuning and configuration” on page 146 for more information.

In this section we show an example of using GitOps to manage node configuration and performance tuning.

Installation of Red Hat OpenShift GitOps

GitOps in Red Hat OpenShift is provided by Red Hat OpenShift GitOps Operator, which is based on ArgoCD.

First we have to install Red Hat OpenShift GitOps Operator from OperatorHub. The install method has to be “**`All namespaces on the cluster (default)`**”.

The operator will be installed into the **`openshift-operators`** namespace.

The operator creates an ArgoCD server instance with the name of **`openshift-gitops`** in **`openshift-gitops`** namespace.

The default configuration sets up the DEX server to authenticate from Red Hat OpenShift OAuth, so the Red Hat OpenShift users can login to ArgoCD right from the beginning. Example 6-33 shows a section of the automatically created ArgoCD resource configuration.

Example 6-33 Default RBAC 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 will have admin role in ArgoCD, so for example they can create projects and application.

Create a GitOps project

To separate different application groups and our infrastructure of code configuration in GitOps we create a new project with the name of **power**.

Figure 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 which will contain our Red Hat OpenShift related configurations, like MachineConfig, Node, Tuned.

It could limit the destination clusters and namespaces as well, but in this case we specified only the local cluster and allow any target namespace.

Additionally the following resource type limitations could be configured per project:

- Cluster scoped resource allow list.
- Cluster scoped resource deny list.
- Namespace scoped resource allow list.
- Namespace scopes resource deny list.

Authorize GitOps to work with Red Hat OpenShift Node and Tuned resources

As we will configure automated Node and Tuned patching and modification we have to authorize the auto created Red Hat OpenShift GitOps related service accounts to patch and modify these resources. For this we create the following Red Hat OpenShift **`roles`** and **`rolebindings`**. See Example 6-35 for details.

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 RBAC settings we would get the errors shown in Figure 6-6 at application synchronization.

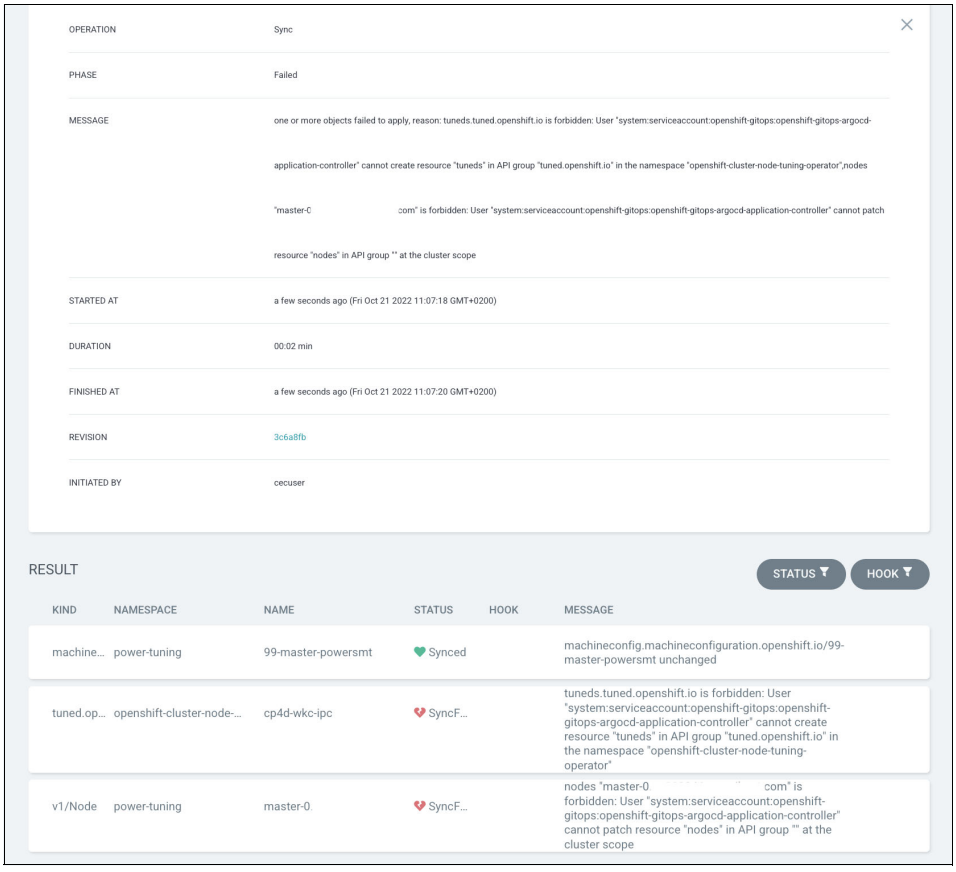


Figure 6-6 Insufficient authorization setting results

Create a GitHub repository for the GitOps application

Create a repository and a folder for the Red Hat OpenShift configuration yaml files. See Figure 6-7. An example GitHub repository has been set up and contains the appropriate YAML files. You can find the GitHub repository at https://github.com/lniesz/power.

The content of the repository is based on the examples in Appendix A of the *Red Hat OpenShift V4.3 on IBM Power Systems Reference Guide*, REDP-5599.

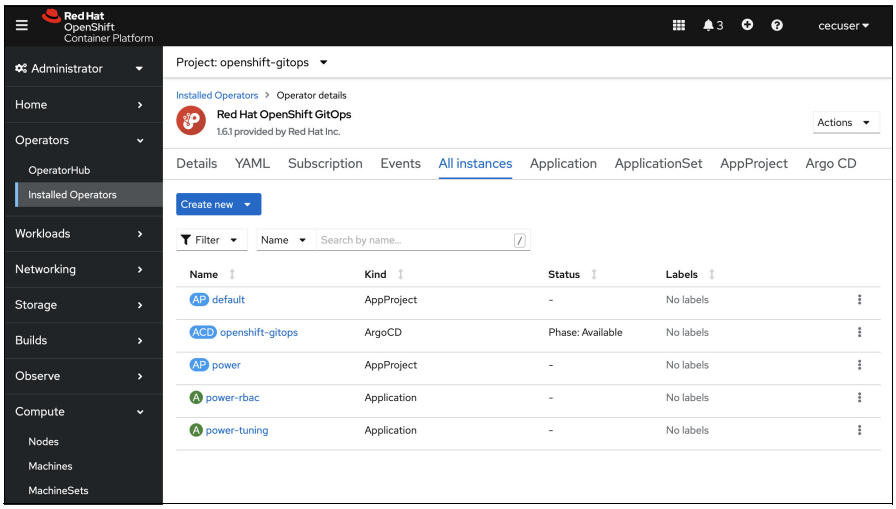


Figure 6-7 Git repository for power-tuning application

The files **99-master-powersmt.yaml** and **cp4dworkertuned.yaml** are based on the IBM Redpaper.

The node definition yaml file should be based on the actual nodes in your cluster and should have the SMT label as shown below 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 as this Red Hat OpenShift node is on an IBM Power Systems Power10 based LPAR.

Create power-tuning GitOps application

After the preparations we can create an application which will do the following:

- Monitor and synchronize a MachineConfig definition, which sets the SMT configuration of a node based on the Node label: SMT. This labeling is also done via GitOps, 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 after that applied with 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 we can see the application state in the ArgoCD GUI. An Red Hat OpenShift route is auto created when Red Hat OpenShift GitOps is installed.

The following screen shots show the different views of the application. Figure 6-8 shows the application overview.

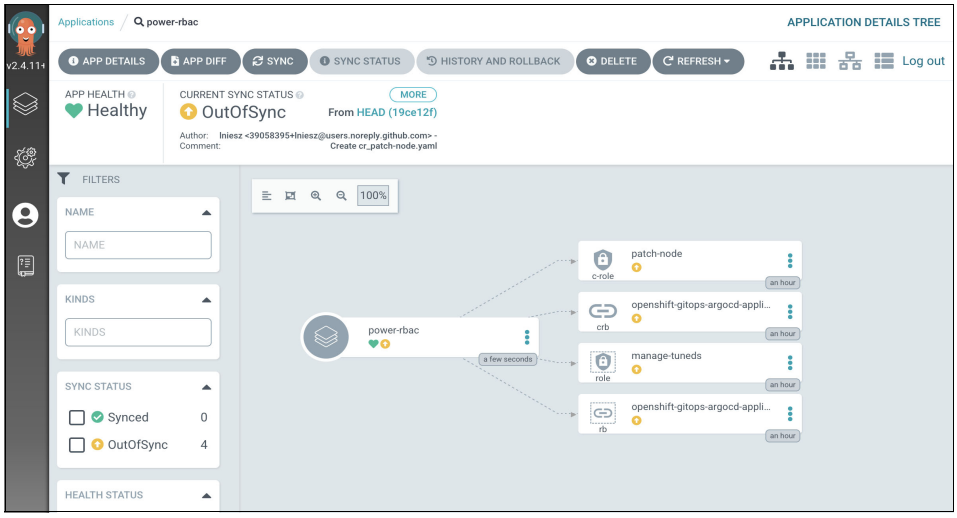


Figure 6-8 Application overview

Figure 6-9 shows the detailed view of the application.

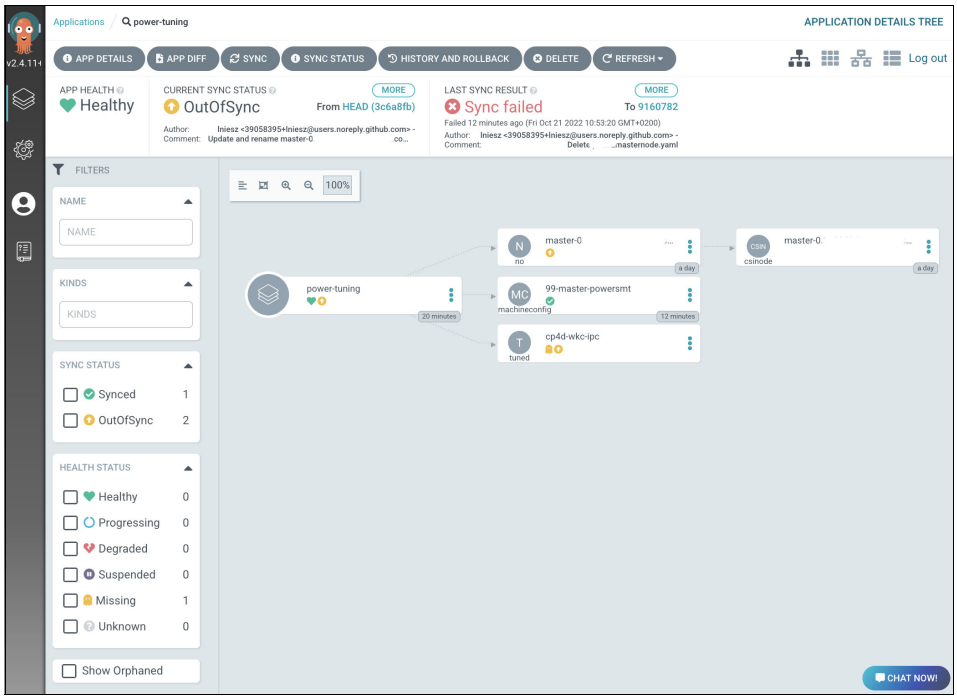


Figure 6-9 Detailed view of application

Figure 6-10 shows the dialog window, when we push the SYNC button. Here we can chose which resources to synchronize with special cases, for pruning, replacing resources and the optional namespace creation, if we have namespace scoped resources in the application.

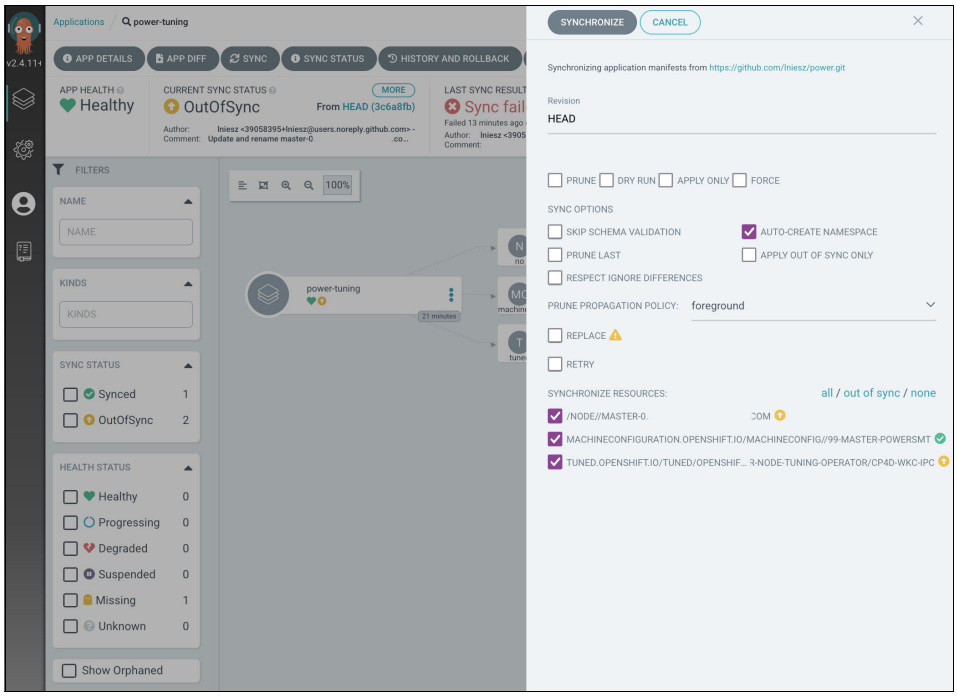


Figure 6-10 Choosing options when pressing SYNC

Synchronize power-tuning application

After successful synchronization all resources are in green, as it is shown in Figure 6-11.

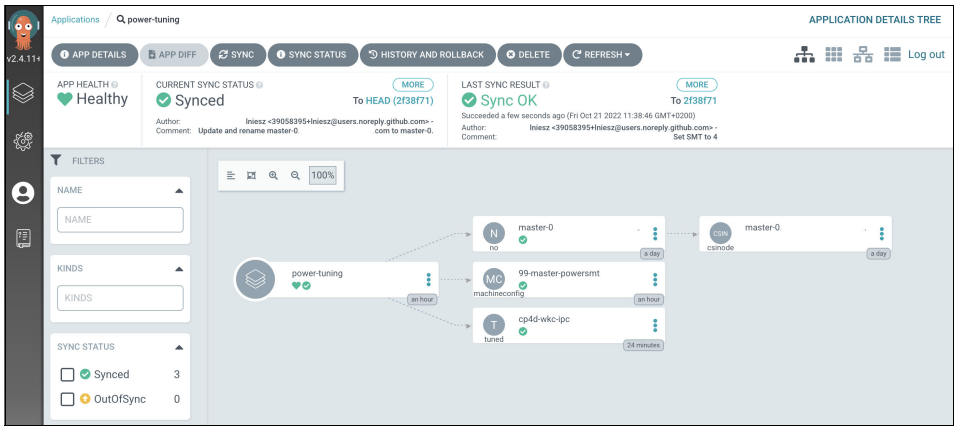


Figure 6-11 Successfully synced application

After successful synchronization we can check what Red Hat OpenShift GitOps related resource instances are shown in Red Hat OpenShift GUI. Open the Installed Operators menu and in the namespace of openshift-gitops choose Red Hat OpenShift GitOps. In the **All instances** pane we can see our instances as shown in Figure 6-12.

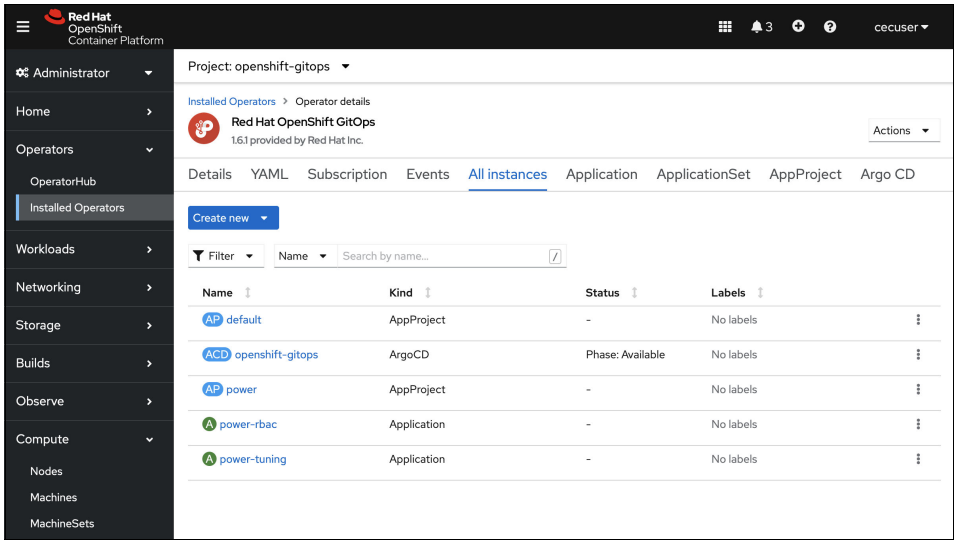


Figure 6-12 Successfully synced application resources

Check configuration on Red Hat OpenShift node

The MachineConfig operator will create a new service with the name of *powersmt*. See Example 6-39 to open debug window to the node and check the new service.

Example 6-39 Check powersmt service on 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 binaries, run `chroot /host`
```

```
Pod IP: 172.20.11.150
If you don't see a command prompt, 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 Power Systems Power10 node is 8, which means each core has 8 hardware thread. This can be checked on the node as shown in Example 6-40.

Example 6-40 Check SMT setting on node

```
sh-4.4# ppc64_cpu --smt
SMT=8

sh-4.4# lscpu
Architecture:        ppc64le
Byte Order:          Little Endian
CPU(s):              16
On-line CPU(s) list: 0-15
Thread(s) per core:  8
```

```
Core(s) per socket:  2
Socket(s):           1
NUMA node(s):        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
```

Change node label in Git and sync the power-tuning application

To check the working of our simple IaC implementation via GitOps change the node yaml file in the Git repository located at:
https://github.com/lniesz/power/blob/main/tuning/master-0.example.com.yaml.

Then commit the change. We only change the SMT label in the yaml file to 4.

The commit message is set to “Set SMT to 4”. This will be shown in the ArgoCD GUI also, showing that our app in the running Red Hat OpenShift cluster is not in sync with our Git repository. See Figure 6-13.

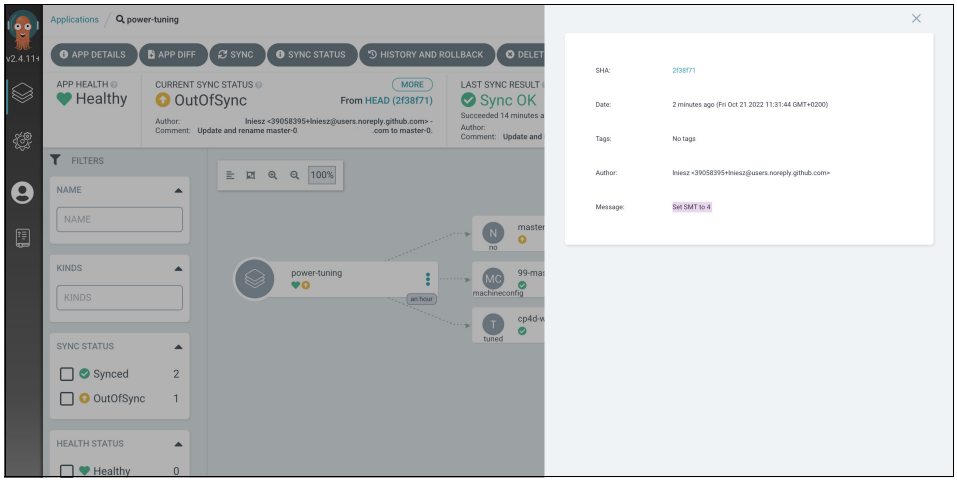


Figure 6-13 Application out of synch after change in Git

After pushing the **SYNC** button ArgoCD will apply the configuration, so in this case set the node label SMT to 4, and the running powersmt service on the node will change the real SMT setting on the node to 4.

We can check that the application will be green in ArgoCD and the SMT setting is changed on node as it is shown in Figure 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
On-line CPU(s) list:  0-3,8-11
Off-line CPU(s) list: 4-7,12-15
Thread(s) per core:   4
Core(s) per socket:   2
Socket(s):            1
NUMA node(s):         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

We see the following development ideas based on our use case:

- As security is getting more and more important check and setup fine grained RBAC rules, for applications, users, managed Red Hat OpenShift resource types.
- Limit target namespaces and resources in GitOps project.
- Review the powersmt service as there are developments in CoreOS in the handling of IBM Power Systems server based commands. In this book we use a simple command ppc64_cpu, which could be incorporated into the powersmt script suggested by the referenced IBM Redpaper.
- 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.
- It is possible to create an application from other applications in GitOps to combine related application groups.

GitOps and ArgoCD works well with HELM and Kustomize as well, so create the yaml files in the Git repo based on your choice of these tools.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- *Matrix-Multiply Assist Best Practices Guide, REDP-5612*
- *Red Hat OpenShift V4.3 on IBM Power Systems Reference Guide, REDP-5599*
- *IBM Power Systems SR-IOV: Technical Overview and Introduction, REDP-5065*
- *Red Hat OpenShift V4.X and IBM Cloud Pak on IBM Power Systems Volume 2, SG24-8486*
- *IBM Cloud Pak for Data Version 4.5: A practical, hands-on guide with best practices, examples, use cases, and walk-throughs*, SG24-8522

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- Using the POWER9 NX (gzip) accelerator in AIX

  https://www.ibm.com/support/pages/using-power9%E2%84%A2-nx-gzip-accelerator-aix
- 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
- AIX community article: *Performance improvement in openssh with on-chip data compression accelerator in power9*

  https://community.ibm.com/community/user/power/blogs/swetha-narayana/2021/07/27/performance-improvement-in-openssh-with-on-chip-da
- IBM Documentation: nxstat Command

  https://www.ibm.com/docs/en/aix/7.2?topic=n-nxstat-command
- OpenCapi Website

  https://opencapi.org/
- How to Create Automated etcd Backup 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)

- Power 10 FAQ

  https://community.ibm.com/community/user/power/viewdocument/sr-iov-vnic-and-hnv-information?CommunityKey=71e6bb8a-5b34-44da-be8b-277834a183b0&tab=librarydocuments

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

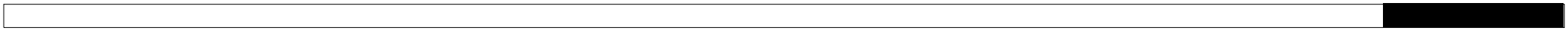
To determine the spine width of a book, you divide the paper PPI into the number of pages in the book. An example is a 250 page book using Plainfield opaque 50# smooth which has a PPI of 526. Divided 250 by 526 which equals a spine width of .4752". In this case, you would use the .5" spine. Now select the Spine width for the book and hide the others: **Special>Conditional Text>Show/Hide>SpineSize(-->Hide:)>Set** . Move the changed Conditional text settings to all files in your book by opening the book file with the spine.fm still open and **File>Import>Formats** the Conditional Text Settings (ONLY!) to the book files.

(1.5" spine)
1.5"<-> 1.998"
789 <->1051 pages

(1.0" spine)
0.875"<->1.498"
460 <-> 788 pages

(0.5" spine)
0.475"<->0.873"
250 <-> 459 pages

(0.2"spine)
0.17"<->0.473"
90<->249 pages

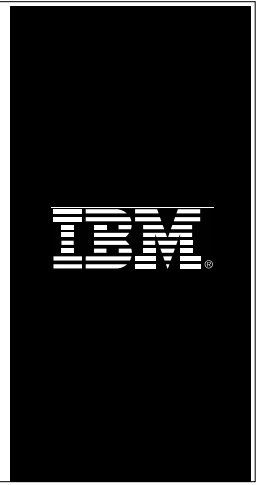


(0.1"spine)
0.1"<->0.169"
53<->89 pages

Implementing, Tuning, and Optimizing

SG24-8537-00

ISBN DocISBN

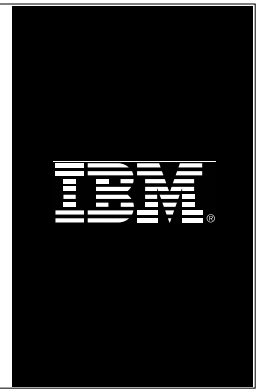


(2.5” spine)
2.5”<->nnn.n”
1315<-> nnnn pages

Implementing, Tuning, and Optimizing Workloads with Red Hat

SG24-8537-00

ISBN DocISBN



(2.0” spine)
2.0” <-> 2.498”
1052 <-> 1314 pages

SG24-8537-00

ISBN DocISBN

Printed in U.S.A.