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

First Edition (July 2020)

This edition applies to:

- Red Hat Enterprise Linux V7
- Red Hat OpenShift Container Platform for Power Enterprise V4.3
- Red Hat CoreOS V4.3
- IBM AIX V7.2

© Copyright International Business Machines Corporation 2020. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
This document was created or updated on July 7, 2020.
Contents

Notices .................................................... vii
Trademarks ................................................ viii

Preface ..................................................... ix
Authors ...................................................... ix
Now you can become a published author, too! ........................................ xi
Comments welcome ........................................ xi
Stay connected to IBM Redbooks ........................................ xii

Chapter 1. Introduction to Red Hat OpenShift V4.3 .............................. 1
1.1 Introduction .............................................. 2
1.2 Red Hat OpenShift V4.3 .................................. 2
1.3 Overview of this publication ................................ 3

Chapter 2. Supported configurations and sizing guidelines ......................... 5
2.1 IBM Power Systems ........................................ 6
  2.1.1 Mission-critical workloads .......................... 7
  2.1.2 Big data workloads ................................... 9
  2.1.3 Enterprise AI workloads ............................ 10
2.2 Red Hat OpenShift Container Platform V4.3 on IBM Power Systems ......... 11
  2.2.1 Differences between Red Hat OpenShift Container Platform V4.3 and V3.11 ... 14
2.3 Supported configurations .................................. 16
  2.3.1 Supported configurations and recommended hardware ..................... 16
  2.3.2 Getting started with Red Hat OpenShift V4.3 on Power Systems with a minimal configuration ......................... 17
  2.3.3 Installation on restricted networks ........................ 19
2.4 Red Hat OpenShift V4.3 sizing guidelines ................................ 20
  2.4.1 Red Hat OpenShift sizing guidelines ................... 22
  2.4.2 Sizing for IBM Cloud Paks .......................... 23
2.5 Storage guidelines ......................................... 25
  2.5.1 NFS storage .......................................... 25
  2.5.2 NFS dynamic provisioning ............................ 26
  2.5.3 IBM Spectrum Scale .................................. 30
  2.5.4 IBM Spectrum Scale and NFS ........................ 30
  2.5.5 PowerVC Container Storage Interface driver (CSI) ...................... 30
  2.5.6 Backing up and restore your Red Hat OpenShift cluster and applications .... 31

Chapter 3. Reference installation guide for Red Hat OpenShift V4.3 on Power Systems servers ................................................. 33
3.1 Introduction .............................................. 34
3.2 Red Hat OpenShift V4.3 deployment with Internet connection stand-alone installation ............................................. 34
  3.2.1 PowerVM configuration for network installation ....................... 35
  3.2.2 BOOTP infrastructure server for installing CoreOS using the network ... 35
  3.2.3 Prerequisites on network infrastructure for Red Hat OpenShift Container Platform installation in a production environment .......... 38
  3.2.4 Creating a network infrastructure for a test environment ............ 39
  3.2.5 Installing Red Hat OpenShift on PowerVM LPARs using the BOOTP network install created ........................................ 44
  3.2.6 Configuring the internal Red Hat OpenShift registry .................. 49
3.3 Using NIM as your BOOTP infrastructure ........................................... 50
  3.3.1 Prerequisites and components .................................................. 51
3.4 Installing on scale-out servers bare metal ...................................... 56
3.5 NVMe 4096 block size considerations ............................................ 57
3.6 Offline Red Hat OpenShift V4.3 deployment .................................... 58

Chapter 4. IBM Cloud Paks ................................................................. 59
  4.1 Why IBM Cloud Paks, an introduction ........................................... 60
  4.2 IBM Cloud Paks ............................................................................ 60
  4.3 IBM Cloud Paks offerings ............................................................ 61
    4.3.1 IBM Cloud Pak for Application ............................................. 61
    4.3.2 IBM Cloud Pak for Data ....................................................... 61
    4.3.3 IBM Cloud Pak for Integration ............................................. 62
    4.3.4 IBM Cloud Pak for Automation ............................................ 62
    4.3.5 IBM Cloud Pak for Multicloud Management ......................... 62
  4.4 IBM Cloud Pak for Data ............................................................... 63
    4.4.1 IBM Cloud Pak for Data features and enhancements ............. 64
    4.4.2 IBM Cloud Pak for Data requirements ................................. 65
    4.4.3 Installing IBM Cloud Pak for Data on Power Systems servers..... 65
    4.4.4 IBM Cloud Pak for Data backup ........................................... 71

Appendix A. Configuring Red Hat CoreOS .............................................. 73
  CoreOS machine configuration management and machineconfig objects........ 74
  Using CoreOS tuned operator to apply the sysctl parameters ................. 82

Appendix B. Booting from System Management Services ......................... 85
  Entering SMS mode ........................................................................... 86
  Option 1: Boot directly to SMS from the HMC .................................. 87
  Option 2: Enter SMS from the startup console .................................. 88
  Configure booting from network ...................................................... 89
  Configure boot device order ............................................................. 102

Appendix C. Additional material ......................................................... 111
  Locating the web material ............................................................... 111
  Using the web material .................................................................... 111
    System requirements for downloading the web material ................. 111
    Downloading and extracting the web material ................................ 112

Related publications ........................................................................... 113
  IBM Redbooks ................................................................................. 113
  Online resources ............................................................................ 113
  Help from IBM ................................................................................. 113
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®
- IBM®
- IBM Cloud®
- IBM Cloud Pak®
- IBM Spectrum®
- IBM Watson®
- InfoSphere®
- Interconnect®
- OpenCAPI™
- POWER®
- POWER8®
- POWER9™
- PowerHA®
- PowerVM®
- Redbooks®
- Redbooks (logo) ®
- SPSS®
- SystemMirror®
- Watson™

The following terms are trademarks of other companies:

ITIL is a Registered Trade Mark of AXELOS Limited.

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.

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

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

RStudio, and the RStudio logo are registered trademarks of RStudio, Inc.

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

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

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

This IBM® Redpaper publication describes how to deploy Red Hat OpenShift V4.3 on IBM Power Systems servers.

This book presents reference architectures for deployment, initial sizing guidelines for server, storage, and IBM Cloud® Paks. Moreover, this publication delivers information about initial supported Power System configurations for Red Hat OpenShift V4.3 deployment (bare metal, IBM PowerVM® LE LPARs, and others).

This book serves as a guide on how to deploy Red Hat OpenShift V4.3 and provide start guidelines and recommended practices for implementing it on Power Systems and completing it with the supported IBM Cloud Paks.

The publication addresses topics for developers, IT architects, IT specialists, sellers, and anyone who wants to implement a Red Hat OpenShift V4.3 and IBM Cloud Paks on IBM Power Systems. This book also provides technical content to transfer how-to skills to the support teams, and solution guidance to the sales team.

This book compliments the documentation that is available at IBM Knowledge Center, and also aligns with the educational offerings that are provided by the IBM Systems Technical Education (SSE).

Authors

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

Dino Quintero is an IT Management Consultant and an IBM Level 3 Senior Certified IT Specialist with IBM Redbooks® in Poughkeepsie, New York. He has 24 years of experience with IBM Power Systems 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, cloud computing, artificial intelligence including machine and deep learning, and cognitive solutions. He also is a Certified Open Group Distinguished IT 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.

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 that on-premises or in the public cloud.

Alain Fisher is an IT specialist and DevOps Developer and supports many IBM development teams at the IBM Hursley Lab, UK. He holds a B.Sc. (Hons) degree in Computer Science from The Open University, England. He joined IBM in 2001 supporting DB2® middleware products, such as DB2 before moving to the Hursley Lab. His areas of expertise include the
OpenStack cloud platform, cloud deployments, automation and virtualization. He has participated on two previous IBM Redbooks since 2005.

**Federico Fros** is an IT Specialist who works for IBM Global Technologies Services leading the UNIX and Storage team for the IBM Innovation center in Uruguay. He has worked in IBM for more than 15 years, including 12 years of experience in IBM Power Systems and IBM Storage. He is an IBM Certified Systems Expert for UNIX and High Availability. His areas of expertise include IBM AIX®, Linux, PowerHA® SystemMirror®, IBM PowerVM, SAN Networks, Cloud Computing, and IBM Storage Family, including IBM Spectrum® Scale.

**Miguel Gomez Gonzalez** is an IBM Power Systems Integration engineer based in Mexico with over 11 years experience in Linux and IBM Power Systems technologies. He has extensive experience in implementing several IBM Linux and AIX clustering solutions. His areas of expertise are Power Systems performance boost, virtualization, high availability, system administration, and test design. Miguel holds a Master in Information Technology Management from ITESM.

**Felix Gonzalez** is an IT Specialist who currently works in UNIX and Storage Team for IBM Global Technologies Services in Uruguay. He is studying Electronic Engineering at Universidad de la República in Uruguay. He is a Red Hat Linux Certified Specialist with 5 years of experience in UNIX systems, including FreeBSD, Linux and AIX. His areas of expertise include AIX, Linux, FreeBSD, Red Hat Ansible, IBM PowerVM, SAN, Cloud Computing, and IBM Storage.

**Paulo Sergio Lemes Queiroz** is a Systems Consultant for AI Solutions on Power Systems with IBM in Brazil. He has 19 years of experience in UNIX and Linux, ranging from systems design and development to systems support. His areas of expertise include AIX Power Systems, AIX, GPFS, RHCS, KVM, and Linux. He is a Certified Advanced Technical Expert for Power Systems with AIX (CATE) and a Red Hat Certified Engineer (RHCE).

**Sudipto Pal** is an Application Architect and Technical Project manager in GBS with over 20 years of leadership experience in designing innovative business solution for clients from various industry, domain and geographical locations. Skilled in Cloud computing, Big Data, Application development and virtualization. He successfully delivered several critical projects with IBM clients from USA and Europe. He led Cognos® administration competency and mentor several candidates. He co-authored Exploiting IBM PowerVM Virtualization Features with IBM Cognos 8 Business Intelligence, SG24-7842 publication.

**Bogdan Savu** is a Cloud Infrastructure Architect at IBM POWER® IaaS and works for IBM Systems in Romania. He has over 13 years of experience in designing, developing, and implementing Cloud Computing, Virtualization, Automatization, and Infrastructure solutions. Bogdan holds a bachelor’s degree in Computer Science from the Polytechnic University of Bucharest. He is an IBM Certified Advanced Technical Expert for Power Systems, ITIL 4 Managing Professional Certificate, TOGAF 9 Certified, Red Hat Certified Engineer, Red Hat Certified Specialist in Ansible Automation, Red Hat Certified Specialist in Red Hat OpenShift Administration, and VMware Certified Professional. His areas of expertise include Cloud Computing, Virtualization, DevOps, and scripting.

**Richard Wale** is a Senior IT Specialist, supporting many IBM development teams at the IBM Hursley Lab, UK. He holds a B.Sc. (Hons) degree in Computer Science from Portsmouth University, England. He joined IBM in 1996 and has been supporting production IBM AIX systems since 1998. His areas of expertise include IBM Power Systems, PowerVM, AIX, and IBM i. He has participated in co-writing many IBM Redbooks publications since 2002.

Thanks to the following people for their contributions to this project:
Now you can become a published author, too!

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

Find out more about the residency program, browse the residency index, and apply online at: ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

- Use the online Contact us review Redbooks form found at: ibm.com/redbooks
- Send your comments in an email to: redbooks@us.ibm.com
- Mail your comments to:
  IBM Corporation, IBM Redbooks
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400
Stay connected to IBM Redbooks

- Find us on Facebook:
  http://www.facebook.com/IBMRedbooks
- Follow us on Twitter:
  http://twitter.com/ibmredbooks
- Look for us on LinkedIn:
  http://www.linkedin.com/groups?home=&gid=2130806
- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
- Stay current on recent Redbooks publications with RSS Feeds:
  http://www.redbooks.ibm.com/rss.html
Introduction to Red Hat OpenShift V4.3

This chapter provides an overview of the scope of this publication.

This chapter contains the following:

- Introduction
- Red Hat OpenShift V4.3
- Overview of this publication
1.1 Introduction

This publication provides an overview of the 4.3 release of Red Hat OpenShift Platform on IBM Power Systems. It is interim successor to the publication Red Hat OpenShift and IBM Cloud Paks on IBM Power Systems: Volume 1, SG24-8459. That publication provided a summary overview of containers, Kubernetes and the introduction of Red Hat OpenShift onto the IBM Power Systems platform.

With over 25 years since the release of the original models, IBM Power Systems continues to be designed for both traditional Enterprise workloads and the most demanding, data-intensive computing. The range of models offer flexibility, scalability and innovation.

We made a statement of intent within the previously mentioned publication, that subsequent volumes be published in due course. We felt this approach better served the agile nature of the Red Hat OpenShift product. The window is always moving, the next release is already on the horizon. At the time of writing, Volume 2 is in development; however with some of the changes and improvements provided by Red Hat OpenShift V4.3, we felt there was a need for an interim publication to follow on the heels of the product release.

Red Hat OpenShift is one of the most reliable enterprise-grade containers. It is designed and optimized to easily deploy web applications and services. Categorized as a cloud development Platform as a Service (PaaS) and based on industry standards, such as Docker and Kubernetes.

This publication explains what is new with Red Hat OpenShift on IBM Power Systems, and provides updated installation instructions and sizing guides.

1.2 Red Hat OpenShift V4.3

The Red Hat OpenShift and IBM Cloud Paks on IBM Power Systems: Volume 1, SG24-8459 publication relates to Red Hat OpenShift V3.11. This book focuses on Red Hat OpenShift V4.3. Although there are V4.x releases for other hardware platforms, V4.3 is the first official V4.x release for IBM Power Systems.

Note: Red Hat OpenShift V4.3 for IBM Power Systems was officially announced by IBM on 28th April 2020. It was subsequently announced two days later by Red Hat on their Red Hat OpenShift blog. For more details of either announcement respectively refer to the following URLs:


https://www.openshift.com/blog/openshift-4.3-now-available-on-ibm-power-systems-1

Red Hat OpenShift V4.3 provides a number of important changes compared to the previous V3.x release written about in Volume 1. The most significant change is the move from Red Hat Enterprise Linux to Red Hat CoreOS for the operating system used for the cluster nodes. CoreOS is a lightweight Linux specifically designed for hosting containers across a cluster of nodes; as it is patched and configured as part of Red Hat OpenShift, it bring consistency to the deployed environment and reduces the overhead of ongoing ownership.

For further information about Red Hat OpenShift V4.3 on IBM Power Systems, refer to Chapter 2, “Supported configurations and sizing guidelines” on page 5.
1.3 Overview of this publication

Within the subsequent chapters in this publication, we discuss Red Hat OpenShift V4.3 on IBM Power Systems; detail the different installation methods (compared to the previous V3.11 release); and document supported environments and suggested sizing considerations.

**Note:** Documented information regarding supported environments, configurations and sizing guides are correct at the time of writing this publication. Be aware that due to the agile nature of Red Hat OpenShift it is entirely possible that elements and aspects can change with subsequent V4.3.x updates. If major changes are required, a revised edition of this IBM RedPaper can be published. However we always recommend to check the given official resources (release notes, online documentation) for any changes or deviations to what is written here.
Supported configurations and sizing guidelines

This chapter provides information about the supported configurations, sizing guidelines and recommended practices to help you size and deploy Red Hat OpenShift on Power Systems.

This chapter contains the following:
- IBM Power Systems
- Red Hat OpenShift Container Platform V4.3 on IBM Power Systems
- Supported configurations
- Red Hat OpenShift V4.3 sizing guidelines
- Storage guidelines
2.1 IBM Power Systems

Over the years, the IBM Power Systems family has grown, matured, been innovated and pushed the boundaries of what clients expect and demand from the harmony of hardware and software.

With the advent of the POWER4 processor in 2001, IBM introduced logical partitions (LPARs) outside of their mainframe family to another audience. What was seen as radical then, has grown into the expected today. The term virtualization is now common-place across most platforms and operating systems. These days virtualization is the core foundation for Cloud Computing.

IBM Power Systems are built for the most demanding, data-intensive, computing on earth. The servers are cloud-ready and help you unleash insight from your data pipeline, from managing mission-critical data, to managing your operational data stores and data lakes, to delivering the best server for cognitive computing.

IBM POWER9, the foundation for the No. 1 and No. 2 supercomputers in the world, is the only processor with state-of-the-art I/O subsystem technology, including next generation NVIDIA NVLink, PCIe Gen4 and OpenCAPI™.

POWER9 processor-based servers can be found in three product families: Enterprise servers, scale-out servers and accelerated servers. Each of these three families is positioned for different types of client requirements and expectations.

IBM Power Systems servers based on POWER9 processors are built for today’s most advanced applications from mission-critical enterprise workloads to big data and AI as shown in Figure 2-1.

**Mission-critical workloads** - Robust scale-out and enterprise servers can support a wide range of mission-critical applications running on IBM AIX, IBM i and Linux operating systems, and they can provide building blocks for private and hybrid cloud environments.

**Big data** - Scale-out servers deliver the performance and capacity required for big data and analytics workloads.

**Enterprise AI** - Servers provide intelligent, accelerated infrastructure for modern analytics, high performance computing (HPC) and AI workloads. They are advanced enterprise servers that deliver fast machine learning performance.
2.1.1 Mission-critical workloads

To handle the most data-intensive mission-critical workloads, organizations need servers that can deliver outstanding performance and scalability. Whether they are supporting small business groups or building large private and hybrid cloud environments, they need no-compromise infrastructure.

Enterprise (scale-up) servers

IBM Power Systems Enterprise (scale-up) servers offer the highest levels of performance and scale for the most data-intensive, mission-critical workloads as shown in Table 2-1. They can also serve as building blocks for growing private and hybrid cloud environments. Support for AIX, Linux and IBM i (for the IBM Power Systems E980 server) gives organizations the flexibility to run a wide range of applications.

The IBM Power Systems E950 server is the correct fit for growing midsize businesses, departments and large enterprises looking for a building-block platform for their data center. The IBM Power Systems E980 server is designed for large enterprises that need flexible, reliable servers for a private or hybrid cloud infrastructure.

Table 2-1  IBM Power Systems: Enterprise servers

<table>
<thead>
<tr>
<th>Key features</th>
<th>E950</th>
<th>E980</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine type and model (MTM)</td>
<td>9040-MR9</td>
<td>9080-M9S</td>
</tr>
<tr>
<td>Form factors</td>
<td>4U</td>
<td>5U system node and 2U system controller unit.</td>
</tr>
<tr>
<td>Sockets</td>
<td>2 to 4</td>
<td>4 per node</td>
</tr>
<tr>
<td>Processor cores</td>
<td>• Up to 48 cores – 12 core processor sockets at 3.15 to 3.80 GHz (max).&lt;br&gt;• Up to 44 cores – 11 core processor sockets at 3.2 to 3.80 GHz (max).&lt;br&gt;• Up to 40 cores – 10 core processor sockets at 3.40 to 3.80 GHz (max).&lt;br&gt;• Up to 32 cores – 8 core processor sockets at 3.60 to 3.80 GHz (max).&lt;br&gt;One node: 4x POWER9 CPUs; 8, 10, 11 or 12 cores each System maximum: 16x POWER9 CPUs; 8, 10, 11 or 12 cores each</td>
<td></td>
</tr>
<tr>
<td>Memory slots</td>
<td>128</td>
<td>128 per node</td>
</tr>
<tr>
<td>Memory max.</td>
<td>16 TB</td>
<td>64 TB per node</td>
</tr>
<tr>
<td>PCIe G4 slots</td>
<td>10</td>
<td>8 per node; max. 32</td>
</tr>
<tr>
<td>Supported operating systems</td>
<td>AIX and Linux</td>
<td>AIX, IBM i and Linux</td>
</tr>
</tbody>
</table>
Scale-out servers

IBM Power Systems scale-out servers for mission-critical workloads offer a strong alternative to commodity x86 servers as shown in Table 2-2. They provide a robust, reliable platform to help maximize performance and help ensure availability.

Scale-out AIX, IBM i and Linux servers - These servers are designed to scale out and integrate into an organization’s cloud and AI strategy, delivering exceptional performance and reliability.

Table 2-2 IBM Power Systems: Scale-out servers

<table>
<thead>
<tr>
<th>Machine type and model (MTM)</th>
<th>S914</th>
<th>S922</th>
<th>S924</th>
<th>L922</th>
</tr>
</thead>
</table>
| Key features                 | • Entry-level offering.  
   • Industry-leading integrated security and reliability.  
   • Cloud-enabled. | • Strong price-performance for mission-critical workloads.  
   • Dense form factor with large memory footprint.  
   • Cloud-enabled with integrated virtualization. | • Industry-leading price-performance for mission-critical workloads.  
   • Large memory footprint.  
   • Strong security and reliability.  
   • Cloud-enabled with integrated virtualization. | • Industry-leading price-performance for mission-critical Linux workloads.  
   • Dense form factor with large memory footprint. |
| Form factors                 | 9009-41A   | 9009-22A   | 9009-42A   | 9008-22L   |
| Sockets                      | 4U and tower | 2U         | 4U         | 2U         |
| Microprocessors              | 1x POWER9 CPUs; 4, 6 or 8 cores. | Up to 2x POWER9 CPUs; 4, 8 or 10 cores. | 2x POWER9 CPUs; 8, 10 or 12 cores. | Up to 2x POWER9 CPUs; 8, 10 or 12 cores. |
| Memory slots                 | 16         | 32         | 32         | 32         |
| Memory max.                  | 1 TB       | 4 TB       | 4 TB       | 4 TB       |
| PCIe G4 slots                | 2          | 4          | 4          | 4          |
| Supported operating systems  | AIX, IBM i and Linux | AIX, IBM i and Linux | AIX, IBM i and Linux | Linux |

Scale-out servers for SAP HANA - These servers are designed to deliver outstanding performance and a large memory footprint of up to 4 TB in a dense form factor as shown in Table 2-3 on page 9. These servers help deliver insights fast at the same time maintaining high reliability. They are also scalable: When it is time to grow, organizations can expand database capacity and the size of their SAP HANA environment without having to provision a new server.
2.1.2 Big data workloads

Across industries, organizations are poised to capitalize on big data to generate new business insights, improve the customer experience, enhance efficiencies and gain competitive advantage. But to make the most of growing data volumes, they need servers with the performance and capacity for big data and AI workloads.

IBM Power Systems Scale-out servers for big data deliver the outstanding performance and scalable capacity for intensive big data and AI workloads as shown in Table 2-4. Purpose-built with a storage-rich server design and industry-leading compute capabilities, these servers are made to explore and analyze a tremendous amount of data—all at a lower cost than equivalent x86 alternatives.

Table 2-4  IBM Power Systems: Scale-out servers for Big Data

<table>
<thead>
<tr>
<th>Key features</th>
<th>LC921</th>
<th>LC922</th>
</tr>
</thead>
<tbody>
<tr>
<td>• High performance in a space-saving design.</td>
<td>• Highest storage capacity in the IBM Power Systems portfolio.</td>
<td></td>
</tr>
<tr>
<td>• Industry-leading compute in a dense form factor.</td>
<td>• Up to 44 cores and 2 TB of memory.</td>
<td></td>
</tr>
<tr>
<td>• High performance for SAP HANA.</td>
<td>• High performance at lower cost than comparable x86 systems.</td>
<td></td>
</tr>
<tr>
<td>Machine type and model (MTM)</td>
<td>9006-12P</td>
<td>9006-22P</td>
</tr>
<tr>
<td>Form factors</td>
<td>1U</td>
<td>2U</td>
</tr>
<tr>
<td>Sockets</td>
<td>1 upgradeable or 2</td>
<td>2</td>
</tr>
</tbody>
</table>
2.1.3 Enterprise AI workloads

AI holds tremendous promise for facilitating digital transformations, accelerating innovation, enhancing the efficiency of internal processes, identifying new marketplace opportunities and more. For organizations to take advantage of AI and cognitive technologies such as machine learning and deep learning, they need powerful, accelerated servers that can handle these data-intensive workloads. Accelerated servers can also play a vital role in HPC and supercomputing. With the correct accelerated servers, researchers and scientists can explore more complex, data-intensive problems and deliver results faster than before.

The IBM Power Systems Accelerated Compute Server helps reduce the time to value for enterprise AI initiatives. The IBM PowerAI Enterprise platform combines this server with popular open source deep learning frameworks and efficient AI development tools to accelerate the processes of building, training and inferring deep learning neural networks. Using PowerAI Enterprise, organizations can deploy a fully optimized and supported AI platform with blazing performance, proven dependability and resilience.

The new IBM Power System IC922 server is built to delivers powerful computing, scaling efficiency, and storage capacity in a cost-optimized design to meet the evolving data challenges of the artificial intelligence (AI) era. Refer to Table 2-5.

The IC in IC922 stands for inference and cloud. The I can also stand for I/O.

Table 2-5 IBM Power Systems: Accelerated compute servers

<table>
<thead>
<tr>
<th>Key features</th>
<th>AC922</th>
<th>IC922</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Unprecedented performance for modern AI, analytics and HPC workloads.</td>
<td>• The Power IC922 server is engineered to put your AI models to work and unlock business insights. It uses cooptimized hardware and software to deliver the necessary components for AI inference.</td>
<td></td>
</tr>
<tr>
<td>• Proven deployments from small clusters to the world’s largest supercomputers, with near-linear scaling.</td>
<td>• Simple GPU acceleration.</td>
<td></td>
</tr>
<tr>
<td>Machine type and model (MTM)</td>
<td>8335-GTH</td>
<td>9183-22X</td>
</tr>
<tr>
<td>Form factors</td>
<td>2U</td>
<td>2U</td>
</tr>
<tr>
<td>Sockets</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
2.2 Red Hat OpenShift Container Platform V4.3 on IBM Power Systems

Red Hat OpenShift V4.3 for Power Systems was announced by IBM on April 28, 2020. For more information see the announcement letter for Red Hat OpenShift V4.3 on Power Systems:


Red Hat OpenShift V4.3 for Power Systems is an enterprise-grade platform providing a secure, private platform-as-a-service cloud on IBM Power Systems servers.

Some of the key features of Red Hat OpenShift V4.3 include:

- Red Hat Enterprise Linux CoreOS, offering a fully immutable, lightweight, and container-optimized Linux OS distribution.
- Cluster upgrades and cloud automation.
- IBM Cloud Paks support on Power Systems platforms. IBM Cloud Paks are a containerized bundling of IBM middleware and open source content.

The Red Hat OpenShift architecture builds on top of Kubernetes and is consists of three types of roles for the nodes:

- **Bootstrap**: Red Hat OpenShift Container Platform uses a temporary bootstrap node during initial configuration to provide the required information to the master node (control plane). It boots by using an Ignition configuration file that describes how to create the cluster. The bootstrap node creates the master nodes, and master nodes creates the worker nodes. The master nodes install additional services in the form of a set of Operators. The Red Hat OpenShift bootstrap node runs CoreOS V4.3.

- **Master**: Red Hat OpenShift Container Platform master is a server that performs control functions for the whole cluster environment. The master machines are the control plane. It is responsible for the creation, scheduling, and management of all objects specific to Red Hat OpenShift. It includes API, controller manager, and scheduler capabilities in one Red Hat OpenShift binary. It is also a common practice to install an `etcd` key-value store on

<table>
<thead>
<tr>
<th>Microprocessors</th>
<th>AC922</th>
<th>IC922</th>
</tr>
</thead>
<tbody>
<tr>
<td>2x POWER9 with NVLink CPUs: 16 or 20 cores; or 18 or 22 cores with liquid cooling.</td>
<td>12-core (2.8 - 3.8 GHz), 16-core (3.35 - 4.0 GHz), and 20-core (2.9 - 3.8 GHz) POWER9.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GPUs</th>
<th>AC922</th>
<th>IC922</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 or 6 NVIDIA Tesla GPU processors.</td>
<td>Up to six NVIDIA T4 GPU accelerator.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory slots.</th>
<th>AC922</th>
<th>IC922</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>32</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory max.</th>
<th>AC922</th>
<th>IC922</th>
</tr>
</thead>
<tbody>
<tr>
<td>1TB</td>
<td>2 TB</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PCIe G4 slots</th>
<th>AC922</th>
<th>IC922</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>• Multiple I/O options in the system with the standard Peripheral Component Interconnect® Express (PCIe) Riser.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Supported operating systems</th>
<th>AC922</th>
<th>IC922</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>Linux</td>
<td></td>
</tr>
</tbody>
</table>
Red Hat OpenShift masters to achieve a low-latency link between etcd and Red Hat OpenShift masters. The Red Hat OpenShift master node runs either CoreOS V4.3 or Red Hat Enterprise Linux V7.x.

**Important:** Because of the consensus required by the RAFT\(^a\) algorithm, the etcd service must be deployed in odd numbers to maintain quorum. For this reason, the minimum number of etcd instances for production environments is three.

a. The Raft Consensus Algorithm: [https://raft.github.io/](https://raft.github.io/)

- **Worker:** Red Hat OpenShift worker nodes run containerized applications created and deployed by developers. A Red Hat OpenShift worker node contains the Red Hat OpenShift node components, including the container engine CRI-O, container workloads running and stopping executor Kubelet, and a service proxy managing across worker nodes communication for Pods. A Red Hat OpenShift application node runs CoreOS V4.3 or Red Hat Enterprise Linux V7.x.

A deployment host is any virtual or physical host that typically required for the installation of Red Hat OpenShift. The Red Hat OpenShift installation assumes that many, if not all the external services like DNS, load balancing, HTTP server, DHCP are already available in an existing data center and therefore there is no need to duplicate them on a node in the Red Hat OpenShift cluster. However, current experience has shown that creating a deployment host node and consolidating the Red Hat OpenShift required external services on it greatly simplifies installation. After installation is complete, the deployment host node can continue to serve as a load balancer for the Red Hat OpenShift API service (running on each of the master nodes) and the application ingress controller (also running on the three master nodes). As part of providing this single front door to the Red Hat OpenShift cluster it can serve as a jump server controlling access between some external network and the Red Hat OpenShift cluster network.
Figure 2-2 shows the high level view of the Red Hat OpenShift Container Platform components for the various IBM Power Systems hardware platforms.

![Diagram of Red Hat OpenShift Container Platform](image)

**Note:** Standalone KVM is for development purposes only.

Nodes can run on top of PowerVC, PowerVM, Red Hat Virtualization, KVM or run bare metal environment. Table 2-6 shows the IBM Power Systems infrastructure landscape for Red Hat OpenShift Container Platform V4.3.

<table>
<thead>
<tr>
<th>IaaS</th>
<th>PowerVC*</th>
<th>N/A</th>
<th>RHV-M</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypervisor</td>
<td>PowerVM</td>
<td>PowerVM</td>
<td>KVM/RHV</td>
<td>Bare metal</td>
</tr>
<tr>
<td>Guest OS</td>
<td>CoreOS V4.3 or later</td>
<td>CoreOS V4.3 or later</td>
<td>CoreOS V4.3 or later</td>
<td>CoreOS V4.3 or later</td>
</tr>
<tr>
<td>Systems</td>
<td>E980, E950, S924, S922, S914, L922</td>
<td>E980, E950, S924, S922, S914, L922</td>
<td>LC922, AC922, IC922</td>
<td>LC922, AC922, IC922</td>
</tr>
<tr>
<td>File storage</td>
<td>NFS, Spectrum Scale**</td>
<td>NFS, Spectrum Scale**</td>
<td>NFS, Spectrum Scale**</td>
<td>NFS, Spectrum Scale**</td>
</tr>
</tbody>
</table>
* PowerVC V1.4.4.1 added support for CoreOS and Red Hat Enterprise Linux V8 VMs and guests.

** Spectrum Scale on Red Hat OpenShift V4.3 requires Red Hat Enterprise Linux V7 worker nodes.

### 2.2.1 Differences between Red Hat OpenShift Container Platform V4.3 and V3.11

This section highlights some of the high-level differences between Red Hat OpenShift V4.3 and Red Hat OpenShift V3.11 on IBM Power Systems:

- **One key difference is the change in the base OS, as it transitions from Red Hat Enterprise Linux V7 to CoreOS for the master nodes. CoreOS is a stripped-down version of Red Hat Enterprise Linux that is optimized for container orchestration. CoreOS comes bundled with Red Hat OpenShift V4.x and is not separately charged.**

- **The container runtime moves from Docker to CRI-O. The difference in the base OS is that CRI-O is a lightweight alternative to using Docker as the runtime for Kubernetes.**

- **The container CLI also transitions into Podman. The key difference between Podman and Docker for CLI is that Podman does not require a daemon to be running, and it shares many of the same underlying components with other container engines, including CRI-O.**

- **Installation and configuration is done using openshift-install (ignition-based) deployment for Red Hat OpenShift V4.3 and replacing Red Hat OpenShift-Ansible based on Ansible tool.**

A side-by-side comparisons of the two stacks is shown in Table 2-7.

#### Table 2-7  Red Hat OpenShift Container Platform V4.3 versus V3.11 stack differences

<table>
<thead>
<tr>
<th></th>
<th>Red Hat OpenShift V4.3</th>
<th>Red Hat OpenShift V3.11</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Base OS</strong></td>
<td>CoreOS</td>
<td>Red Hat Enterprise Linux V7</td>
</tr>
<tr>
<td><strong>Container Run-time</strong></td>
<td>CRI-O</td>
<td>Docker</td>
</tr>
<tr>
<td><strong>Container CLI</strong></td>
<td>Podman, Buildah, Skopeo</td>
<td>Docker</td>
</tr>
<tr>
<td><strong>Installation</strong></td>
<td>openshift-Install</td>
<td>OpenShift-Ansible</td>
</tr>
<tr>
<td><strong>Operational tools</strong></td>
<td>Prometheus</td>
<td>Hawkular, Cassandra</td>
</tr>
<tr>
<td><strong>Z-stream update</strong></td>
<td>Automated (every week)</td>
<td>Manual (six weeks)</td>
</tr>
<tr>
<td><strong>Content Update</strong></td>
<td>Automated with operators</td>
<td>Manual</td>
</tr>
</tbody>
</table>
In Figure 2-3, the solid boxes represent nodes that are required at a minimum to run Red Hat OpenShift Container Platform v3.11. The dashed lines represent recommended configuration for production. When deploying Red Hat OpenShift V3.11 on Power Systems, it is only required to have one master, one infrastructure, and one worker node. It is also common practice to consolidate the master and infrastructure nodes on a single VM. Although the worker node can also be consolidated onto a single VM, it is not recommended because Red Hat OpenShift core licenses are determined by the number of cores on the worker nodes and not the control nodes. Although, this is the minimal install, when moving into a production level deployment, it is recommended that there be at least 3 copies of this on 3 separate systems for high availability and fault tolerance.

In Red Hat OpenShift V4.3, the three master nodes become a requirement, and at least two worker nodes must be present in the cluster. Red Hat OpenShift environments with multiple workers usually require a distributed shared file system. In most cases the developer of the application does not have any control over which worker in the Red Hat OpenShift cluster the application Pod is dispatched. So, regardless of which worker the application can be deployed to, the persistent storage needed by the application needs to be available. One supported storage provider for Red Hat OpenShift V4.3 on POWER architecture is an NFS server. PowerVC CSI driver is also supported and can be used for block storage. To use NFS you need to create the NFS server that in this book was done using Spectrum Scale CES. For a provisioner, export the NFS to each of the Red Hat OpenShift worker nodes. You can create static persistent volumes and use a different export point for each desired Red Hat OpenShift persistent volume. However, you do not explicitly mount the exports on the Red Hat OpenShift workers. The mounts are done by Red Hat OpenShift as needed by the containers when they request an associated PVC for the predefined persistent volumes. Also a persistent volume provisioner for NFS is available and used in our IBM Cloud Pak® for Data scenario.

In Figure 2-4 on page 16, the solid boxes represent nodes that are required at a minimum to run Red Hat OpenShift Container Platform V4.3 with NFS storage.
2.3 Supported configurations

This section provides guidance on the supported configurations for installing Red Hat OpenShift V4.3 on Power Systems.

2.3.1 Supported configurations and recommended hardware

Red Hat OpenShift V4.3 on Power systems can be deployed in the supported configurations as shown in Table 2-8. Both PowerVM and KVM based Power systems are supported for Red Hat OpenShift V4.3. On KVM based systems such as the AC922 and IC922, the host operating system is Red Hat Enterprise Linux and CoreOS is the guest OS. CoreOS as the host OS is for the bare metal deployment.

Table 2-8 Supported configurations

<table>
<thead>
<tr>
<th>POWER9 server</th>
<th>Guest OS</th>
<th>Host OS</th>
<th>Cloud mgmt.</th>
<th>File storage</th>
<th>Block storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>E980/E950</td>
<td>CoreOS V4.3</td>
<td>N/A</td>
<td>PowerVC V1.4.4.1</td>
<td>NFS Spectrum Scale</td>
<td>Spectrum Virtualize CSI PowerVC CSI</td>
</tr>
<tr>
<td>S9xx/L9xx</td>
<td>CoreOS V4.3</td>
<td>N/A</td>
<td>PowerVC V1.4.4.1</td>
<td>NFS Spectrum Scale</td>
<td>Spectrum Virtualize CSI PowerVC CSI</td>
</tr>
</tbody>
</table>
Figure 2-5 on page 18 provides further clarity on the supported host OS and guest OS for KVM based Power systems. For PowerVM based systems such as the enterprise E/S/L systems there is no host OS, only a guest OS. Red Hat Enterprise Linux V7 as a guest OS is only supported in Red Hat OpenShift V3.11. Red Hat OpenShift V4.3 requires CoreOS.

Note: At the time this publication was written, Spectrum Scale support is limited to deployment as an NFS server. IBM intends to support Spectrum Scale on CoreOS in future releases.

### 2.3.2 Getting started with Red Hat OpenShift V4.3 on Power Systems with a minimal configuration

This section describes the minimal configuration of Red Hat OpenShift V4.3 on Power Systems to get you started using the container platform. This initial sizing helps you to get a Red Hat OpenShift V4.3 instance (stand-alone) without a large capital investment. This option also helps you to scale to an HA production level deployment in the future.

To deploy in a minimal configuration of Red Hat OpenShift V4.3, you need the following machines:

- A temporary bootstrap machine.
- Three masters or control plane machines.
- Two workers or compute machines.

The relative sizing for these nodes is shown in Table 2-9.

<table>
<thead>
<tr>
<th>Machine</th>
<th>Operating system</th>
<th>vCPU</th>
<th>RAM</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bootstrap</td>
<td>Red Hat CoreOS</td>
<td>4</td>
<td>16 GB</td>
<td>120 GB</td>
</tr>
<tr>
<td>Control Plane</td>
<td>Red Hat CoreOS</td>
<td>4</td>
<td>16 GB</td>
<td>120 GB</td>
</tr>
<tr>
<td>Compute</td>
<td>Red Hat CoreOS</td>
<td>2</td>
<td>8 GB</td>
<td>120 GB</td>
</tr>
</tbody>
</table>

The bootstrap node is only required during the installation step but is not needed after the cluster is up and running.

Note: Given this sizing (Table 2-9), the number of cores required will depend on the configured SMT level. Remember to account for that to stand up a minimal instance of Red Hat OpenShift V4.3.
PowerVM, bare metal and KVM based Power Systems are supported with Red Hat OpenShift V4.3:

- On PowerVM based systems such as the enterprise E/S/L systems, there is no host OS and CoreOS is the guest OS.
- On KVM based systems such as the AC922 and IC922, the host OS is Red Hat Enterprise Linux and CoreOS is the guest OS.
- CoreOS as the host OS is for the bare metal deployments.

Figure 2-5 shows an example production level deployment on scale-out (IC922) systems. This configuration attempts to be as cost efficient as possible. One thing to note is that the 22 cores 240 GB is to represent the total amount of compute resources that are available for worker nodes on that system. It must not be interpreted as a single VM with 20 cores and 240 GB allocated but to what it can be scaled up to.

**Important:** One thing to keep note of when sizing with an IC922 is that for PowerVC GA 1.5, only 4 VMs can be supported on a single system.
Figure 2-6 shows an example production level deployment on scale-up (E950) systems.

![Minimal Red Hat OpenShift V4.3 deployment on POWER scale-Up architecture](image)

**2.3.3 Installation on restricted networks**

With Red Hat OpenShift V4.3, you can perform an installation that does not require an active connection to the Internet to obtain software components. You can complete an installation in a restricted network only on an infrastructure that you provision, not with the infrastructure that the installation program provisions, hence your platform selection is limited.

To complete a restricted network installation, you must create a registry that mirrors the contents of the Red Hat OpenShift registry and contains the installation media. You can create this mirror on a bastion host, which can access both the Internet and your closed network, or by using other methods that meet your restrictions.

**Note:** It is worth noting that restricted network installations always use user-provisioned infrastructures. Because of the complexity of the configuration for user-provisioned installations, consider completing a standard user-provisioned infrastructure installation before you attempt a restricted network installation. Completing this test installation can make it easier to isolate and troubleshoot any issues that might arise during your installation in a restricted network.

For more detailed information about supported configurations, see installing Red Hat OpenShift on IBM Power Systems at the following website:

https://docs.openshift.com/container-platform/4.3/installing/installing_ibm_power/installing-ibm-power.html

For more information about use cases, refer to the following website:

https://developer.ibm.com/linuxonpower/power-hybrid-cloud/
2.4 Red Hat OpenShift V4.3 sizing guidelines

This section gives an idea on how containers consume CPU and the benefits that Power Systems can bring to you.

Power Systems can have up to 8 threads per core. This is a benefit for containers that consume CPU based on the OS CPU count as shown in Example 2-1. You can see that this is based on the number of threads available in the system.

Example 2-1  lscpu output

```
[root@client ~]# 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:    1
Socket(s):             2
NUMA node(s):          2
Model:                 2.1 (pvr 004b 0201)
Model name:            POWER8 (architected), altivec supported
Hypervisor vendor:     pHyp
Virtualization type:   para
L1d cache:             64K
L1i cache:             32K
L2 cache:              512K
L3 cache:              8192K
NUMA node0 CPU(s):     0-3,8-11
NUMA node3 CPU(s):     0-3,8-11
```

This means you can have 4x more containers per core, maintaining the required and limiting settings of your YAML files when compared to x86.

You can find examples of different workloads running on Power Systems 2.5X (two and a half times) faster when running 8 threads on a PowerVM system when compared to the same amount of cores on x86:

▶ https://www.youtube.com/watch?v=KHHNrlv5qoxs

Although performance is not a simple 2.5 to 1 comparison and workloads can vary, this section tries to explain how to use this performance advantage.

With Kubernetes, Pods are assigned CPU resources on a CPU thread basis. "CPU" on deployment and Pod definitions refers to an OS CPU which maps to a CPU hardware thread. For an x86 system, an x86 core running with hyperthreading is equivalent to 2 Kubernetes CPUs. This means when running with x86 hyperthreading, a Kubernetes CPU is equivalent to 1/2 of an x86 core. 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 4 Kubernetes CPUs whereas when running with SMT-8, the same PowerVM core is equivalent to 8 Kubernetes CPUs. This means when running with SMT-4, a Kubernetes CPU is equivalent to 1/4th of a PowerVM core and when running with SMT-8 a Kubernetes CPU is equivalent to 1/8th of a PowerVM core. If your Pod CPU resource was defined to run on x86, you need to consider the effects of the POWER's performance advantage and the effects of
Kubernetes resources being assigned on a thread basis. For example, for a workload where POWER has a 2X advantage over x86 when running with PowerVM SMT-4, you can assign the same number of Kubernetes CPUs to POWER that you do to x86 to get equivalent performance. From a hardware point of view, you are assigning the performance of 1/2 the number of cores to POWER that you assigned to x86. Whereas, for a workload where POWER has a 2X advantage over x86 when running with PowerVM SMT-8, you need to assign twice the number of Kubernetes CPUs to POWER that you do to x86 to get equivalent performance. Even though you are assigning twice the number of Kubernetes CPUs to POWER, from a hardware point of view, you are assigning the performance of 1/2 the number of cores to POWER that you assigned to x86.

**Note:** IBM Cloud Pak for Data V3.0.1 defaults to running in SMT-4 mode on POWER. This can change in the future.

Transforming this abstract concept on core performance divided on threads is difficult, and the summary is shown in Figure 2-7.

![Pods consume OS CPU = Threads](image)

*Figure 2-7  Threads on a core*

As seen in Figure 2-7, for the workload shown, because the PowerVM system can deliver 2X the performance of x86 when running with SMT-4, 2 Kubernetes CPUs for PowerVM (which is equivalent to 1/2 of a physical core) can deliver the same performance as 2 Kubernetes CPUs for x86 (which is equivalent to 1 physical core).
Figure 2-8 shows the container on a Pod limited (throttled) to 1 CPU, in yellow. You see the behavior is different because of the relative performance from the different SMT modes (ppc64le servers).

All development performance testing done on PowerVM systems are done using SMT4, and are capable of using half of the cores than x86 using the same assumptions. On OpenPower processor systems, all the tests for IBM Cloud Pak for Data are done with SMT2. Using these tests as baseline, we provide directions for sizing guidelines on 2.4.2, “Sizing for IBM Cloud Paks” on page 23.

Appendix A, “Configuring Red Hat CoreOS” on page 73 shows steps for controlling SMT on a node using labels. This enables different SMT across your cluster for different purposes. You can use the label to select where your application runs depending on needs. Using this method you can have a super packing configuration of nodes: nodes running SMT8 and nodes running a specific SMT because of application constrains.

### 2.4.1 Red Hat OpenShift sizing guidelines

Table 2-10 shows the conversion between vCPU, x86 physical cores, and IBM POWER9™ physical cores. Power Systems performance is normally higher than that of x86 but this performance is workload dependent. For initial sizing, a conservative 1-to-1 mapping of x86 cores to Power Systems Opal only cores and 2-to-1 for Power Systems PowerVM capable have been used.

<table>
<thead>
<tr>
<th>vCPUs and x86/POWER cores</th>
<th>vCPU</th>
<th>x86 Cores (SMT-2)</th>
<th>Physical SMT-2 POWER cores</th>
<th>Physical SMT-4 POWER cores</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Let's begin by first highlighting the master node resource sizing requirements. The master node sizing requirements depend on the number of worker nodes in the cluster. Table 2-11 shows recommended master node sizing by worker node density.

**Table 2-11  Master node initial sizing**

<table>
<thead>
<tr>
<th>Worker nodes</th>
<th>vCPU</th>
<th>Physical SMT-2 POWER cores</th>
<th>Physical SMT-4 POWER cores</th>
<th>Memory (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>100</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>32</td>
</tr>
<tr>
<td>250</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>64</td>
</tr>
</tbody>
</table>

To figure out the number of worker nodes in the cluster, it is important to determine how many Pods are expected to fit per node. This number is dependent on the application itself as the application's memory, CPU, and storage requirements need to be considered. Red Hat has also provided guidelines for the maximum number of Pods per node, which is 250. It is recommended to not exceed this number as it results in lower overall performance.

### 2.4.2 Sizing for IBM Cloud Paks

This section gives a high level overview of the sizing guidelines for deploying the various IBM Cloud Paks on Red Hat OpenShift V4.3 on Power Systems. For more in depth information about each IBM Cloud Pak see Chapter 4, “IBM Cloud Paks” on page 59.

Table 2-12 outlines the number of VMs that are required for deployment and sizing of each VM for IBM Cloud Pak for Multicloud Manager. Table 2-13 shows the conversion of the number of vCPUs to physical cores.

**Table 2-12  IBM Cloud Pak for Multicloud Manager**

<table>
<thead>
<tr>
<th>Node</th>
<th># of VM’s</th>
<th>vCPU per node</th>
<th>Memory (GB per node)</th>
<th>Disk (GB per node)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bastion</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>100</td>
</tr>
<tr>
<td>Master</td>
<td>3</td>
<td>16</td>
<td>32</td>
<td>300</td>
</tr>
<tr>
<td>Worker</td>
<td>8</td>
<td>4</td>
<td>16</td>
<td>200</td>
</tr>
<tr>
<td>Storage</td>
<td>1</td>
<td>4</td>
<td>16</td>
<td>2x100, 3x200</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>13</strong></td>
<td><strong>86</strong></td>
<td><strong>244</strong></td>
<td><strong>3200</strong></td>
</tr>
</tbody>
</table>

Table 2-13 shows the conversion of the number of vCPUs to physical cores.

**Table 2-13  vCPUs to physical cores conversion**

<table>
<thead>
<tr>
<th>vCPU</th>
<th>Physical x86 cores</th>
<th>Physical SMT-2 POWER cores</th>
<th>Physical SMT-4 POWER cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>86</td>
<td>43</td>
<td>43</td>
<td>22</td>
</tr>
</tbody>
</table>

The recommended sizing for installing IBM Cloud Pak for Applications is shown in Table 2-14 on page 24.
Table 2-14  IBM Cloud Pak for Applications

<table>
<thead>
<tr>
<th>Node</th>
<th># of VM’s</th>
<th>vCPU per node</th>
<th>Memory (GB per node)</th>
<th>Disk (GB per node)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master</td>
<td>3</td>
<td>8</td>
<td>16</td>
<td>200</td>
</tr>
<tr>
<td>Worker</td>
<td>3</td>
<td>8</td>
<td>16</td>
<td>100</td>
</tr>
<tr>
<td>Shared Service</td>
<td>1</td>
<td>8</td>
<td>16</td>
<td>100</td>
</tr>
<tr>
<td>Total</td>
<td>7</td>
<td>56</td>
<td>128</td>
<td>1000</td>
</tr>
</tbody>
</table>

Table 2-15 shows the conversion of the number of vCPUs to physical cores.

Table 2-15  vCPUs to physical cores conversion

<table>
<thead>
<tr>
<th>vCPU</th>
<th>Physical x86 cores</th>
<th>Physical SMT-2 POWER cores</th>
<th>Physical SMT-4 POWER cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>28</td>
<td>28</td>
<td>14</td>
</tr>
</tbody>
</table>

The recommended sizing for installing IBM Cloud Pak for Integration is shown in Table 2-16.

Table 2-16  IBM Cloud Pak for Integration

<table>
<thead>
<tr>
<th>Node</th>
<th># of VM’s</th>
<th>vCPU per node</th>
<th>Memory (GB per node)</th>
<th>Disk (GB per node)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master</td>
<td>3</td>
<td>8</td>
<td>16</td>
<td>200</td>
</tr>
<tr>
<td>Worker</td>
<td>8</td>
<td>8</td>
<td>16</td>
<td>200</td>
</tr>
<tr>
<td>Storage</td>
<td>1</td>
<td>4</td>
<td>16</td>
<td>2x100, 3x200</td>
</tr>
<tr>
<td>Total</td>
<td>12</td>
<td>92</td>
<td>192</td>
<td>2000</td>
</tr>
</tbody>
</table>

Table 2-17 shows the conversion of the number of vCPUs to physical cores.

Table 2-17  vCPUs to physical cores conversion

<table>
<thead>
<tr>
<th>vCPU</th>
<th>Physical x86 cores</th>
<th>Physical SMT-2 POWER cores</th>
<th>Physical SMT-4 POWER cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>50</td>
<td>50</td>
<td>25</td>
</tr>
</tbody>
</table>

The recommended sizing for installing the IBM Cloud Pak for Data is shown in Table 2-18. IBM Cloud Pak for Data is the most variable in terms of the sizing as it is highly dependent on the add-ons that are required for a given project. The Power Systems release has almost 50 add-on modules for AI (IBM Watson®), Analytics, Dashboarding, Governance, Data Sources, Development Tools, and Storage. Part of this effort must include sizing for add-ons, but for now the information provided Table 2-18 is used for a base installation.

Table 2-18  IBM Cloud Pak for Data

<table>
<thead>
<tr>
<th>Node</th>
<th># of VM’s</th>
<th>vCPU per node</th>
<th>Memory (GB per node)</th>
<th>Disk (GB per node)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master</td>
<td>3</td>
<td>8</td>
<td>32</td>
<td>n/a</td>
</tr>
<tr>
<td>Worker</td>
<td>3</td>
<td>16</td>
<td>64</td>
<td>200</td>
</tr>
<tr>
<td>Storage</td>
<td>1</td>
<td>n/a</td>
<td>n/a</td>
<td>800</td>
</tr>
</tbody>
</table>
Table 2-19 shows the conversion of the number of vCPUs to physical cores.

<table>
<thead>
<tr>
<th>vCPU</th>
<th>Physical x86 cores</th>
<th>Physical SMT-2 POWER cores</th>
<th>Physical SMT-4 POWER cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>40</td>
<td>40</td>
<td>20</td>
</tr>
</tbody>
</table>

The recommended sizing for installing the IBM Cloud Pak for Automation is shown in Table 2-20.

<table>
<thead>
<tr>
<th>Node</th>
<th># of VM’s</th>
<th>vCPU per node</th>
<th>Memory (GB per node)</th>
<th>Disk (GB per node)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master</td>
<td>3</td>
<td>8</td>
<td>16</td>
<td>200</td>
</tr>
<tr>
<td>Worker</td>
<td>5</td>
<td>8</td>
<td>16</td>
<td>100</td>
</tr>
<tr>
<td>Shared Service</td>
<td>1</td>
<td>8</td>
<td>32</td>
<td>400</td>
</tr>
<tr>
<td>Total</td>
<td>7</td>
<td>72</td>
<td>160</td>
<td>1500</td>
</tr>
</tbody>
</table>

Table 2-21 shows the conversion of the number of vCPUs to physical cores.

<table>
<thead>
<tr>
<th>vCPU</th>
<th>Physical x86 cores</th>
<th>Physical SMT-2 POWER cores</th>
<th>Physical SMT-4 POWER cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>72</td>
<td>36</td>
<td>36</td>
<td>18</td>
</tr>
</tbody>
</table>

### 2.5 Storage guidelines

This section discusses the storage options available for Red Hat OpenShift V4.3 on Power Systems. Red Hat OpenShift environments with multiple workers usually require a distributed shared file system. This requirement stems from the fact that most applications require some sort of persistent store. In most cases, the developer of the application does not have any control over which worker in the Red Hat OpenShift cluster the application Pod is dispatched. Hence, regardless of which worker the application can be deployed to, the persistent storage required by the application needs to be available. However, there are some environments that do not require the added complexity required to provision a shared, distributed storage environment.

#### 2.5.1 NFS storage

At the time this publication was written, the only supported ReadWriteMany (RWX) storage provider for Red Hat OpenShift V4.3 on POWER architecture is an NFS server. You need to
install the required NFS components on each of the Red Hat OpenShift worker nodes, in addition to the node designated as the NFS server. For each desired Red Hat OpenShift persistent volume an explicit NFS export must be created. However, you do not explicitly mount the exports on the Red Hat OpenShift workers. The mounts are done by Red Hat OpenShift as needed by the containers when they issue an associated PVC for the predefined persistent volumes. A YAML file created to define a static persistent volume based on NFS is shown in Example 2-2.

Example 2-2  Static NFS persistent volume provisioning

```yaml
apiVersion: v1
class: PersistentVolume
metadata:
  name: nfs-pv01
spec:
  capacity:
    storage: 300Gi
  accessModes:
    - ReadWriteMany
  PersistentVolumeReclainPolicy: Retain
  nfs:
    path:/nfsFileSystem/pv01
    nfsServer.domain.com
    readOnly: False
```

2.5.2  NFS dynamic provisioning

This section uses the provisioner with the deployment files as seen on the following website:

https://github.com/kubernetes-incubator/external-storage/tree/master/nfs-client/deploy

You use the deployment.yaml file but change it to use the ppc64le image docker.io/ibmcom/nfs-client-provisioner-ppc64le:latest as shown in Example 2-3. You need to have a preexisting NFS export to use this provisioner. Change `<NFS_SERVER>` and `<NFS_BASE_PATH>` to match your exported NFS. You can use Spectrum Scale as your NFS server as shown in 2.5.4, “IBM Spectrum Scale and NFS” on page 30. There are some advantages of Spectrum Scale over a kernel NFS, for example the use of snapshots. This is due to the possibility to use Active File Management for DR and multicloud purposes, and more advanced features that Spectrum Scale provides.

Example 2-3  NFS provisioner deployment file

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nfs-client-provisioner
  labels:
    app: nfs-client-provisioner
  # replace with namespace where provisioner is deployed
  namespace: default
spec:
  replicas: 1
  strategy:
    type: Recreate
  selector:
```
matchLabels:
  - app: nfs-client-provisioner

template:
  metadata:
    labels:
      - app: nfs-client-provisioner

spec:
  serviceAccountName: nfs-client-provisioner
  containers:
    - name: nfs-client-provisioner
      image: docker.io/ibmcom/nfs-client-provisioner-ppc64le:latest
      volumeMounts:
        - name: nfs-client-root
          mountPath: /persistentvolumes
      env:
        - name: PROVISIONER_NAME
          value: fuseim.pri/ifs
        - name: NFS_SERVER
          value: <NFS_SERVER>
        - name: NFS_PATH
          value: <NFS BASE PATH>
  volumes:
    - name: nfs-client-root
      nfs:
        server: <NFS_SERVER>
        path: <NFS BASE PATH>

You also need the rbac.yaml file as shown in Example 2-4.

Example 2-4  Role based access for the provisioner in rbac.yaml file

apiVersion: v1
kind: ServiceAccount
metadata:
  name: nfs-client-provisioner
  # replace with namespace where provisioner is deployed
  namespace: default

---
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: nfs-client-provisioner-runner
rules:
  - apiGroups: [""]
    resources: ["persistentvolumes"]
    verbs: ["get", "list", "watch", "create", "delete"]
  - apiGroups: [""]
    resources: ["persistentvolumeclaims"]
    verbs: ["get", "list", "watch", "update"]
  - apiGroups: ["storage.k8s.io"]
    resources: ["storageclasses"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["""]
    resources: ["events"]
    verbs: ["create", "update", "patch"]

---
kind: ClusterRoleBinding
display: rbac.authorization.k8s.io/v1
metadata:
  name: run-nfs-client-provisioner
subjects:
  - kind: ServiceAccount
    name: nfs-client-provisioner
    namespace: default
roleRef:
  kind: ClusterRole
  name: nfs-client-provisioner-runner
  apiGroup: rbac.authorization.k8s.io

---

kind: Role
display: rbac.authorization.k8s.io/v1
metadata:
  name: leader-locking-nfs-client-provisioner
  namespace: default
rules:
  - apiGroups: [""
    resources: ["endpoints"
    verbs: ["get", "list", "watch", "create", "update", "patch"]

---

kind: RoleBinding
display: rbac.authorization.k8s.io/v1
metadata:
  name: leader-locking-nfs-client-provisioner
  namespace: default
subjects:
  - kind: ServiceAccount
    name: nfs-client-provisioner
    namespace: default
roleRef:
  kind: Role
  name: leader-locking-nfs-client-provisioner
  apiGroup: rbac.authorization.k8s.io

And finally for the storage class creation, use the class.yaml file as shown in Example 2-5.

Example 2-5   Storage class definition class.yaml

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: managed-nfs-storage
provisioner: fuseim.pri/ifs # or choose another name, must match deployment's env
parameters:
  archiveOnDelete: "false"
To define the provisioner apply the rbac file, the security constrains, the deployment and the class as shown in Example 2-6.

**Note:** We are defining the provisioner on the default project. We recommend to create a new project and define the provisioner. Remember to change all namespaces from default to your namespace. Also remember to change from default to your namespace on the **add-scc-to-user** command as shown in Example 2-6.

```
Example 2-6  Defining the provisioner

[root@client ~]# oc apply -f rbac.yaml
serviceaccount/nfs-client-provisioner created
clusterrole.rbac.authorization.k8s.io/nfs-client-provisioner-runner created
clusterrolebinding.rbac.authorization.k8s.io/run-nfs-client-provisioner created
role.rbac.authorization.k8s.io/leader-locking-nfs-client-provisioner created
rolebinding.rbac.authorization.k8s.io/leader-locking-nfs-client-provisioner created
[root@client ~]# oc adm policy add-scc-to-user hostmount-anyuid
system:serviceaccount:default:nfs-client-provisioner
securitycontextconstraints.security.openshift.io/hostmount-anyuid added to:
["system:serviceaccount:default:nfs-client-provisioner"]
[root@client ~]# oc apply -f deployment.yaml
deployment.apps/nfs-client-provisioner created
[root@client ~]# oc apply -f class.yaml
storageclass.storage.k8s.io/managed-nfs-storage created
[root@client ~]#
```

You can use the provisioner now.

**Guidelines for IBM Cloud Pak for Data storage performance**

IBM Cloud Pak for Data relies on dynamic persistent volumes provisioned during installation. The method described in 2.5.2, “NFS dynamic provisioning” on page 26 has been used in this book as the base for IBM Cloud Pak for Data install.

To ensure correct behavior of your system, check your NFS exports meet latency and bandwidth specifications. To test latency, check your result is comparable or better than the result as shown in Example 2-7.

```
Example 2-7  Test latency for IBM Cloud Pak for Data

[root@client ~]# dd if=/dev/zero of=/mnt/testfile bs=4096 count=1000 oflag=dsync
1000+0 records in
1000+0 records out
4096000 bytes (4.1 MB, 3.9 MiB) copied, 1.5625 s, 2.5 MB/s
```

To test bandwidth check your result is comparable or better than the result as shown in Example 2-8.

```
Example 2-8  Test bandwidth for IBM Cloud Pak for Data

[root@client ~]# dd if=/dev/zero of=/mnt/testfile bs=1G count=1 oflag=dsync
1+0 records in
1+0 records out
1073741824 bytes (1.1 GB) copied, 5.14444 s, 209 MB/s
```
2.5.3 IBM Spectrum Scale

IBM Spectrum Scale is a cluster file system that provides concurrent access to a single file system or set of file systems from multiple nodes. The nodes can be SAN attached, network attached, a mixture of SAN attached and network attached, or in a shared nothing cluster configuration. This enables high performance access to this common set of data to support a scale-out solution or to provide a high availability platform.

**Note:** Native Spectrum Scale support for CoreOS is intended for all architectures, s390x, ppc64le and x86_64, and can be a great option for persistent storage.

IBM Spectrum Scale has many features beyond common data access including data replication, policy based storage management, and multi-site operations. You can create a cluster of AIX nodes, Linux nodes, Windows server nodes, or a mix of all three. IBM Spectrum Scale can run on virtualized instances providing common data access in environments to take advantage of logical partitioning, or other hypervisors. Multiple IBM Spectrum Scale clusters can share data within a location or across wide area network (WAN) connections.

For more information about the benefits and features of IBM Spectrum Scale, refer to the following website:


2.5.4 IBM Spectrum Scale and NFS

IBM Spectrum Scale provides extra protocol access methods. Providing these additional file and object access methods and integrating them with Spectrum Scale offers several benefits. It enables users to consolidate various sources of data efficiently in one global namespace. It provides a unified data management solution and enables not only efficient space utilization but also avoids having to make unnecessary data moves just because access methods can be different.

Protocol access methods that are integrated with Spectrum Scale are NFS, SMB, and Object. Although each of these server functions (NFS, SMB and Object) uses open source technologies, this integration adds value by providing scaling and high availability by using the clustering technology in Spectrum Scale. The NFS support for IBM Spectrum Scale enables clients to access the Spectrum Scale file system by using NFS clients with their inherent NFS semantics.

For further information about setting up IBM Spectrum Scale as an NFS server refer to the following website:


2.5.5 PowerVC Container Storage Interface driver (CSI)

The Container Storage Interface (CSI) allows Red Hat OpenShift to consume storage from storage backends that implement the CSI interface as persistent storage. The PowerVC CSI driver is a standard for providing storage ReadWriteOnce (RWO) functionality to containers. The PowerVC CSI pluggable driver interacts with PowerVC storage for operations such as create volumes, delete volumes, attach or detach volumes.
For more information about the Container Storage Interface in Red Hat OpenShift refer to the following website:

https://docs.openshift.com/container-platform/4.3/storage/persistent_storage/persistent-storage-csi.html

For more information about the PowerVC CSI and how to configure it, refer to the following website:


2.5.6 Backing up and restore your Red Hat OpenShift cluster and applications

There are many parts of the cluster that must be backed up so it can be restored. This section describes it on a high level.

**Backing up the etcd to restore the cluster state**

The state of the cluster can be backed up to the master node. This backup can be used to restore the cluster to this previous state if anything happens to the cluster.

The procedure to backup the `etcd` can be read at the following website:

https://docs.openshift.com/container-platform/4.3/backup_and_restore/backing-up-etcd.html

Check the two files generated are backed up in a safe place outside of the cluster so it can be retrieved and used in case it is needed.

The procedure to restore to a previous state can be read at the following website:

https://docs.openshift.com/container-platform/4.3/backup_and_restore/disaster_recovery/scenario-2-restoring-cluster-state.html

**Backing up application consistent persistent volumes**

The backup of the persistent volumes is highly dependent on the application being used. Each application likely has a way to correctly backup the data residing on the persistent volume.

Some databases like DB2 have their own backup tool (`db2 backup`) to create online backups, and taken off the container to be used if need it. This tool is also used on the containers as you can read at the following website:


If you are using NFS or Spectrum Scale, you can use any other backup tool to backup the directory after the backup completed successfully. Just check you backup to a PV that is available externally and has enough space to perform the operation.

Other type of applications, that are easily scaled down and scaled up, can be scaled down to replicas=0, taking a snapshot of the volume, (can be a fileset if you are using Spectrum Scale for example), and returning the applications back to their former replica number.

Some applications, for example IBM Cloud Pak for Data, provide tools to quiesce the workload (`cpdbr quiesce` and `cpdbr unquiesce`). You can also combine this tools to create a backup strategy for your PVC.
Some applications do not need to maintain consistency, and a backup of the PV files is enough.

**Backing up crash consistent persistent volumes**
If you are using Spectrum Scale, you can always snapshot the file system and back it up.

**Note:** This is not advisable if you intend to always maintain an application consistent method of restore.

Some applications can fail to come up with this method considering they need consistency.
Reference installation guide for Red Hat OpenShift V4.3 on Power Systems servers

This chapter describes the install procedure and dependencies to install Red Hat OpenShift Container Platform V4.3 on Power Systems servers. The documentation here is based on the Red Hat install documentation using a user provisioned infrastructure that most enterprise system administrators are used to, and can use to provision to their developers and application environments.

This chapter contains the following:

- Introduction
- Red Hat OpenShift V4.3 deployment with Internet connection stand-alone installation
- Using NIM as your BOOTP infrastructure
- Installing on scale-out servers bare metal
- Offline Red Hat OpenShift V4.3 deployment
3.1 Introduction

This chapter focuses on the user provided infrastructure method of installation aiming on enterprise customers that have their network infrastructure to provide load balancing, firewall and DNS.

On all architectures, the control plane runs only Red Hat CoreOS and does not permit the use of Red Hat Linux for this role, unlike the previous Red Hat OpenShift V3.x release. The installation of this operating system (OS) depends on a file called ignition that acts on the configuration of the CoreOS. CoreOS is an immutable OS so there is no rpm package management like you expect on a Red Hat Enterprise Linux OS. The minimum number of nodes required for a production Red Hat OpenShift cluster are 3 master nodes and 2 worker nodes. The only option for worker nodes on ppc64le environments is CoreOS so you have a flat environment with CoreOS on master and worker nodes. The collection of the master nodes is called control plane. A bootstrap node is needed to bring the control plane up, but it is destroyed after the control plane is fully up.

The documentation in which this chapter has been based on is available at the following link: https://docs.openshift.com/container-platform/4.3/installing/installing_ibm_power/installing-ibm-power.html

This chapter highlights the differences, and uses the BOOTP for the boot process instead of PXE boot as suggested in the documentation for PowerVM LPARs. Although we use the defined PXE boot for bare metal installation.

The minimum resource requirement to start a Red Hat OpenShift V4.3 cluster on Power Systems servers is described in Table 3-1.

<table>
<thead>
<tr>
<th>Machine</th>
<th>Operating system</th>
<th>vCPU</th>
<th>Virtual RAM</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bootstrap</td>
<td>Red Hat CoreOS</td>
<td>4</td>
<td>16 GB</td>
<td>120 GB</td>
</tr>
<tr>
<td>Control plane</td>
<td>N/A</td>
<td>2</td>
<td>16 GB</td>
<td>120 GB</td>
</tr>
<tr>
<td>Compute</td>
<td>N/A</td>
<td>2</td>
<td>16 GB</td>
<td>120 GB</td>
</tr>
</tbody>
</table>

3.2 Red Hat OpenShift V4.3 deployment with Internet connection stand-alone installation

This section explains how to use the network boot method translating the PXE boot present on the official document to BOOTP installation for PowerVM LPARs. Most existing customers are using BOOTP on NIM. This section shows how to create a Linux environment for that to segregate the NIM server from the Red Hat OpenShift install server. You can also unify them and use an existing NIM server to be your network installation infrastructure (BOOTP and TFTP) as shown in 3.3, “Using NIM as your BOOTP infrastructure” on page 50.
3.2.1 PowerVM configuration for network installation

The client process to install CoreOS using the network boot is the same as in any AIX NIM installation. For example, this installation cover the install of Red Hat OpenShift Container Platform V4.3 across three IBM POWER® or POWER9 L922 with enough hardware for a starter kit for a basic Spectrum Scale environment. Figure 3-1 shows the network connections for a minimal workload with Spectrum Scale as the storage backend for Read-Write-Many (RWX) persistent volumes.

![PowerVM network architecture example](image)

This test environment uses other servers to act as the DNS and load balancer considering we do not have control over the enterprise network resources, only to the Power Systems servers. Considering it is a single point of failure, do not use this method on a production environment. Use instead highly available network services provided by the enterprise networking, infrastructure and security teams. The DHCP/BOOTP in this case is another single server and it is fine for production environments if the nodes are installed with fixed IP given that the server is only used during installation. The install server can be your NIM server, if you already have one.

3.2.2 BOOTP infrastructure server for installing CoreOS using the network

During the initial boot, the machines require either a DHCP/TFTP server or a static IP addresses be set to establish a network connection to download their ignition configuration files. For this installation all our nodes have direct Internet access, and have installed DHCP/TFTP on the client server to serve IP addresses. We are also serving HTTP and for this environment the IP address of this server is 192.168.122.5. We used a ppc64le Red Hat Enterprise Linux V7.7 for simplicity of configuration of the grub network environment.

This document does not go into great deal of setting up a boot environment. If you want more understanding of how this is done, read Red Hat’s documentation for Configuring Network Boot on IBM Power Systems Using GRUB2 at the following website:

Install a DHCP server, a TFTP server and a HTTP server using YUM.

Follow the procedure on Example 3-1 to create a GRUB2 network boot directory inside the tftp root.

**Example 3-1 Creating netboot directory**

```
[root@jinete ~]# grub2-mknetdir --net-directory=/var/lib/tftpboot
Netboot directory for powerpc-ieee1275 created. Configure your DHCP server to point to /var/lib/tftpboot/boot/grub2/powerpc-ieee1275/core.elf
```

Now you need to configure the grub load for each server on /var/lib/tftpboot/boot/grub2/grub.cfg. Example 3-2 shows entries for each kind. Check the ignition files for each.

**Example 3-2 Example for each entry on /var/lib/tftpboot/boot/grub2/grub.cfg**

```bash
if [ ${net_default_mac} == 52:54:00:af:db:b6 ]; then
default=0
fallback=1
timeout=1
menuentry "Bootstrap CoreOS (BIOS)" {
    echo "Loading kernel Bootstrap"
    linux "/rhcos-4.3.18-ppc64le-installer-kernel-ppc64le" rd.neednet=1
    nameserver=192.168.122.2 console=tty0 console=ttyS0 coreos.inst=yes
    coreos.inst.install_dev=vda
    coreos.inst.image_url=http://192.168.122.5:8080/rhcos-4.3.18-ppc64le-metal.ppc64le.raw.gz coreos.inst.ignition_url=http://192.168.122.5:8080/bootstrap.ign
    echo "Loading initrd"
    initrd "/rhcos-4.3.18-ppc64le-installer-initramfs.ppc64le.img"
}
fi

if [ ${net_default_mac} == 52:54:00:02:23:c7 ]; then
default=0
fallback=1
timeout=1
menuentry "Master1 CoreOS (BIOS)" {
    echo "Loading kernel Master1"
    linux "/rhcos-4.3.18-ppc64le-installer-kernel-ppc64le" rd.neednet=1
    nameserver=192.168.122.2 console=tty0 console=ttyS0 coreos.inst=yes
    coreos.inst.install_dev=vda
    coreos.inst.image_url=http://192.168.122.5:8080/rhcos-4.3.18-ppc64le-metal.ppc64le.raw.gz coreos.inst.ignition_url=http://192.168.122.5:8080/master.ign
    echo "Loading initrd"
    initrd "/rhcos-4.3.18-ppc64le-installer-initramfs.ppc64le.img"
}
fi

if [ ${net_default_mac} == 52:54:00:68:5c:7c ]; then
default=0
fallback=1
timeout=1
menuentry "Worker1 CoreOS (BIOS)" {
```
echo "Loading kernel Worker1"
linux "/rhcos-4.3.18-ppc64le-installer-kernel-ppc64le" rd.neednet=1
nameserver=192.168.122.2 console=tty0 console=ttyS0 coreos.inst=yes
coreos.inst.install_dev=vda
coreos.inst.image_url=http://192.168.122.5:8080/rhcos-4.3.18-ppc64le-metal.ppc64le.raw.gz
coreos.inst.ignition_url=http://192.168.122.5:8080/worker.ign
echo "Loading initrd"
intrd "/rhcos-4.3.18-ppc64le-installer-initramfs.ppc64le.img"
fi

Example 3-2 on page 36 shows the installation image and the ignition file point to an HTTP server. Use `httpd` and change the port to run on 8080. Leave in the directory the default `/var/www/html`.

**Note:** Remember to repeat the entries for all nodes as these are only examples for each node type to bootstrap the cluster. You have at least three masters and two worker nodes. In our case, we have three masters and two worker nodes. The `env2` entry on the `IP` parameter is the one in our environment and changes with the virtual ID you give to it on the LPAR.

Example 3-2 on page 36 shows some referenced files. You can download these installation files from the following site: [https://cloud.redhat.com/openshift/install/power/user-provisioned](https://cloud.redhat.com/openshift/install/power/user-provisioned)

You find the client, the installer and the CoreOS assets. Remember to get all the referenced files in `grub.cfg`: `rhcos*metal.ppc64le.raw.gz`, `rhcos*initramfs.ppc64le.img`, and `rhcos*kernel-ppc64le`.

The `rhcos*metal.ppc64le.raw.gz` file must be placed on the root directory of the HTTP server, and both `rhcos*initramfs.ppc64le.img` and `rhcos*kernel-ppc64le` must be on the TFTP root directory.

Install `dhcpcd` and configure `dhcpcd.conf` to match your configuration as shown in Example 3-3.

**Example 3-3**  **Contents of `/etc/dhcp/dhcp.conf`**

```plaintext
default-lease-time 900;
max-lease-time 7200;
subnet 192.168.122.0 netmask 255.255.255.0 {
  option routers 192.168.122.1;
  option subnet-mask 255.255.255.0;
  option domain-search "ocp4.ibm.lab";
  option domain-name-servers 192.168.122.2;
  next-server 192.168.122.5;
  filename "boot/grub2/powerpc-ieee1275/core.elf";
}

allow bootp;
option conf-file code 209 = text;
host bootstrap {
  hardware ethernet 52:54:00:af:db:b6;
  fixed-address 192.168.122.10;
  option host-name "bootstrap.ocp4.ibm.lab";
}
```
allow booting;
}
host master1 {
    hardware ethernet 52:54:00:02:23:c7;
    fixed-address 192.168.122.11;
    option host-name "master1.ocp4.ibm.lab";
    allow booting;
}
host master2 {
    hardware ethernet 52:54:00:06:c2:ee;
    fixed-address 192.168.122.12;
    option host-name "master2.ocp4.ibm.lab";
    allow booting;
}
host master3 {
    hardware ethernet 52:54:00:df:15:3e;
    fixed-address 192.168.122.13;
    option host-name "master3.ocp4.ibm.lab";
    allow booting;
}
host worker1 {
    hardware ethernet 52:54:00:07:b4:ec;
    fixed-address 192.168.122.14;
    option host-name "worker1.ocp4.ibm.lab";
    allow booting;
}
host worker2 {
    hardware ethernet 52:54:00:68:5c:7c;
    fixed-address 192.168.122.15;
    option host-name "worker2.ocp4.ibm.lab";
    allow booting;
}
host worker3 {
    hardware ethernet 52:54:00:68:ac:7d;
    fixed-address 192.168.122.16;
    option host-name "worker3.ocp4.ibm.lab";
    allow booting;
}

The MAC address must match both grub.cfg and dhcpd.conf files for correct installation.

**Note:** The MAC addresses used differ as they are normally randomly generated upon LPAR definition. You can use your preferred method to define the LPARs and collect the MAC address from the configuration to use on these files.

### 3.2.3 Prerequisites on network infrastructure for Red Hat OpenShift Container Platform installation in a production environment

The network prerequisites can be found in the Red Hat OpenShift Container Platform installation documentation on Power Systems at the following website:

https://docs.openshift.com/container-platform/4.3/installing/installing_ibm_power/installing-ibm-power.html
Note: Use highly available enterprise services available in your infrastructure to provide the services indicated on this section (load balancer, DNS, and network connectivity). This is normally done before the installation by the enterprise networking, security, and active directory teams.

This section shows copies of the documentation tables for easier reference. Check for updates on the source documentation.

### 3.2.4 Creating a network infrastructure for a test environment

If you want to perform a test or get acquainted with Red Hat OpenShift without creating a full network infrastructure for it, follow the procedures to create the prerequisites if permitted on your network.

**Important:** Do not follow this procedure for production environments unless you have created a full network infrastructure.

#### Provision DNS with dnsmasq on test environments

Table 3-2 is from the Red Hat OpenShift documentation, shown for reference, and it contains needed DNS entries used by the cluster nodes and Pods.

<table>
<thead>
<tr>
<th>Component</th>
<th>Record</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kubernetes external API</td>
<td>api.&lt;cluster_name&gt;.&lt;base_domain&gt;.</td>
<td>This DNS A/AAAA or CNAME record must point to the load balancer for the control plane machines. This record must be resolvable by both clients external to the cluster and from all the nodes within the cluster.</td>
</tr>
<tr>
<td>Kubernetes internal API</td>
<td>api-int.&lt;cluster_name&gt;.&lt;base_domain&gt;.</td>
<td>This DNS A/AAAA or CNAME record must point to the load balancer for the control plane machines. This record must be resolvable from all the nodes within the cluster. The API server must be able to resolve the worker nodes by the host names that are recorded in Kubernetes. If it cannot resolve the node names, proxied API calls can fail, and you cannot retrieve logs from Pods.</td>
</tr>
</tbody>
</table>
To provision the DNS, enter all the details as shown in Table 3-2 on page 39 in your DNS. Ensure it forwards the requests to the actual network DNS servers, in this case represented

<table>
<thead>
<tr>
<th>Component</th>
<th>Record</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routes</td>
<td>*.&lt;cluster_name&gt;.&lt;base_domain&gt;.</td>
<td>A wildcard DNS A/AAAA or CNAME record that points to the load balancer that targets the machines that run the Ingress router Pods, which are the worker nodes by default. This record must be resolvable by both clients external to the cluster and from all the nodes within the cluster.</td>
</tr>
<tr>
<td>etcd Name Record</td>
<td>etcd-&lt;index&gt;.&lt;cluster_name&gt;.&lt;base_domain&gt;.</td>
<td>Red Hat OpenShift Container Platform requires DNS A/AAAA records for each etcd instance to point to the control plane machines that host the instances. The etcd instances are differentiated by &lt;index&gt; values, which start with 0 and end with n-1, where n is the number of control plane machines in the cluster. The DNS record must resolve to an unicast IPv4 address for the control plane machine, and the records must be resolvable from all the nodes in the cluster.</td>
</tr>
</tbody>
</table>
| etcd Service Record | _etcd-server-ssl._tcp.<cluster_name>.<base_domain>. | For each control plane machine, Red Hat OpenShift Container Platform also requires a SRV DNS record for etcd server on that machine with priority 0, weight 10 and port 2380. A cluster that uses three control plane machines requires the following records:

<table>
<thead>
<tr>
<th>Name</th>
<th>TTL</th>
<th>Class</th>
<th>Priority</th>
<th>Weight</th>
<th>Port</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>_etcd-server-ssl._tcp.&lt;cluster_name&gt;.&lt;base_domain&gt;</td>
<td>86400</td>
<td>IN</td>
<td>0</td>
<td>10</td>
<td>2380</td>
<td>etcd-0.&lt;cluster_name&gt;.&lt;base_domain&gt;</td>
</tr>
<tr>
<td>_etcd-server-ssl._tcp.&lt;cluster_name&gt;.&lt;base_domain&gt;</td>
<td>86400</td>
<td>IN</td>
<td>0</td>
<td>10</td>
<td>2380</td>
<td>etcd-1.&lt;cluster_name&gt;.&lt;base_domain&gt;</td>
</tr>
<tr>
<td>_etcd-server-ssl._tcp.&lt;cluster_name&gt;.&lt;base_domain&gt;</td>
<td>86400</td>
<td>IN</td>
<td>0</td>
<td>10</td>
<td>2380</td>
<td>etcd-2.&lt;cluster_name&gt;.&lt;base_domain&gt;</td>
</tr>
</tbody>
</table>
by 10.124.0.1 and 10.124.0.2. In this way, all other addresses can also be resolved when pulling containers from the registries. Example 3-4 shows our dnsmasq configuration file.

**Example 3-4  dnsmasq configuration file**

```plaintext
address=/client.ocp4.ibm.lab/192.168.122.5

address=/bootstrap.ocp4.ibm.lab/192.168.122.10
ptr-record=10.122.168.192.in-addr.arpa,bootstrap.ocp4.ibm.lab

address=/master1.ocp4.ibm.lab/192.168.122.11
address=/etcd-0.ocp4.ibm.lab/192.168.122.11
srv-host=_etcd-server-ssl._tcp.ocp4.ibm.lab,etcd-0.ocp4.ibm.lab,2380
ptr-record=11.122.168.192.in-addr.arpa,master1.ocp4.ibm.lab

address=/master2.ocp4.ibm.lab/192.168.122.12
address=/etcd-1.ocp4.ibm.lab/192.168.122.12
srv-host=_etcd-server-ssl._tcp.ocp4.ibm.lab,etcd-1.ocp4.ibm.lab,2380
ptr-record=12.122.168.192.in-addr.arpa,master2.ocp4.ibm.lab

address=/master3.ocp4.ibm.lab/192.168.122.13
address=/etcd-2.ocp4.ibm.lab/192.168.122.13
srv-host=_etcd-server-ssl._tcp.ocp4.ibm.lab,etcd-2.ocp4.ibm.lab,2380
ptr-record=13.122.168.192.in-addr.arpa,master3.ocp4.ibm.lab

address=/worker1.ocp4.ibm.lab/192.168.122.14
ptr-record=14.122.168.192.in-addr.arpa,worker1.ocp4.ibm.lab

address=/worker2.ocp4.ibm.lab/192.168.122.15
ptr-record=15.122.168.192.in-addr.arpa,worker2.ocp4.ibm.lab

address=/worker3.ocp4.ibm.lab/192.168.122.16

address=/api.ocp4.ibm.lab/192.168.122.3
address=/api-int.ocp4.ibm.lab/192.168.122.3
address=/apps.ocp4.ibm.lab/192.168.122.3

# Listen on lo and env2 only
bind-interfaces
interface=lo,env2
server=10.124.0.1
server=10.124.0.2
```

**Provision load balancing with haproxy on test environments**
The next prerequisite mentioned on the documentation is the load balancing. We use haproxy to accomplish this prerequisite for the test environments as shown in Table 3-3 on page 42.
Table 3-3  Load balancing entries requirement

<table>
<thead>
<tr>
<th>Port</th>
<th>Machines</th>
<th>Internal</th>
<th>External</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6443</td>
<td>Bootstrap and controlplane. You remove the bootstrap machine from the load balancer after the bootstrap machine initializes the cluster control plane.</td>
<td>X</td>
<td>X</td>
<td>Kubernetes API server.</td>
</tr>
<tr>
<td>22623</td>
<td>Bootstrap and controlplane. You remove the bootstrap machine from the load balancer after the bootstrap machine initializes the cluster control plane.</td>
<td>X</td>
<td></td>
<td>Machine config server.</td>
</tr>
<tr>
<td>443</td>
<td>The machines that run the Ingress router Pods, compute, or worker, by default.</td>
<td>X</td>
<td>X</td>
<td>HTTPS traffic.</td>
</tr>
<tr>
<td>80</td>
<td>The machines that run the Ingress router Pods, compute, or worker, by default.</td>
<td>X</td>
<td>X</td>
<td>HTTP traffic.</td>
</tr>
</tbody>
</table>

Example 3-5 shows the configuration file of what the installation documentation describes when using haproxy to implement it.

Example 3-5  haproxy configuration file

global
log 127.0.0.1 local2
chroot /var/lib/haproxy
pidfile /var/run/haproxy.pid
maxconn 4000
user haproxy
group haproxy
daemon

# turn on stats unix socket
stats socket /var/lib/haproxy/stats

defaults
  mode http
  log global
  option httplog
  option dontlognull
  option http-server-close
  option forwardfor except 127.0.0.0/8
  option redispachat
  retries 3
  timeout http-request 10s
  timeout queue 1m
  timeout connect 10s
  timeout client 1m
  timeout server 1m
  timeout http-keep-alive 10s
  timeout check 10s
  maxconn 3000

frontend openshift-api
  bind *:6443
  default_backend openshift-api
  mode tcp
  option tcplog

backend openshift-api
  balance source
  mode tcp
  server ocp43-bootstrap 192.168.0.10:6443 check
  server ocp43-master01 192.168.0.11:6443 check
  server ocp43-master02 192.168.0.12:6443 check
  server ocp43-master03 192.168.0.13:6443 check

frontend openshift-configserver
  bind *:22623
  default_backend openshift-configserver
  mode tcp
  option tcplog

backend openshift-configserver
  balance source
  mode tcp
  server ocp43-bootstrap 192.168.0.10:22623 check
  server ocp43-master01 192.168.0.11:22623 check
  server ocp43-master02 192.168.0.12:22623 check
  server ocp43-master03 192.168.0.13:22623 check

frontend openshift-http
  bind *:80
  default_backend openshift-http
  mode tcp
  option tcplog

backend openshift-http
  balance source
  mode tcp
  server ocp43-worker01 192.168.0.14:80 check
  server ocp43-worker02 192.168.0.15:80 check
server ocp43-worker02 192.168.0.16:80 check
frontend openshift-https
  bind *:443
  default_backend openshift-https
  mode tcp
  option tcplog
backend openshift-https
  balance source
  mode tcp
  server ocp43-worker01 192.168.0.14:443 check
  server ocp43-worker02 192.168.0.15:443 check
  server ocp43-worker03 192.168.0.16:443 check

After you are done with your configuration, check all services are started. You must be aware you might need to change SELinux boolean configurations to get haproxy to serve on any port.

3.2.5 Installing Red Hat OpenShift on PowerVM LPARs using the BOOTP network install created

This section uses the BOOTP process to install CoreOS. Red Hat OpenShift Container Platform is installed with all the configurations to be passed to CoreOS install with the bootstrap ignition file.

Check you downloaded the Red Hat OpenShift install package, the Red Hat OpenShift client from https://cloud.redhat.com/openshift/install/power/user-provisioned, and the binaries are on your path (you need to uncompress the packages you download). Also, you already downloaded the CoreOS assets to build your BOOTP infrastructure and the files are correctly placed as directed. On this same page you find your pull secret that you need to configure your install-config.yaml file.

The pull secret is tied to your account and you can use your licenses. At the time this publication was written, we were tied to a 60-day trial. If you intend to maintain the cluster for more than 60 days, check you have a valid subscription.

1. Follow the steps for installing a Power Systems cluster until the point you have your YAML file ready. Do not create the ignition files at this moment. Create the sample file as shown in the following website:

https://docs.openshift.com/container-platform/4.3/installing/installing_ibm_power/installing-ibm-power.html#installation-initializing-manual_installation-ibm-power

Our YAML file is shown in Example 3-6.

Example 3-6 install-config.yaml file

apiVersion: v1
baseDomain: ibm.lab
compute:
  - hyperthreading: Enabled
    name: worker
    replicas: 0
controlPlane:
  hyperthreading: Enabled
  name: master
  replicas: 3
The creation of the manifests which are a set of YAML files that are used to create the ignition files that configure the CoreOS installation. The creation of the manifests is done using the YAML file you prepared (Example 3-6 on page 44), and the openshift-install command downloaded from https://cloud.redhat.com/openshift/install/power/user-provisioned. Place both openshift-installer and install-config.yaml file into a single directory and create a backup of the install-config.yaml because it is deleted after use. For any install, these are the only three files in the directory. Our install directory is shown in Example 3-7.

Example 3-7   install directory

```
[root@client install]# ls
install-config.yaml install-config.yaml.bak openshift-install
[root@client install]# 
```

After you prepared the install directory, run the manifest creation as shown in Example 3-8.

Example 3-8   Manifest creation

```
[root@client install]# ./openshift-install create manifests
INFO Consuming Install Config from target directory 
WARNING Making control-plane schedulable by setting MastersSchedulable to true for
Scheduler cluster settings
[root@client install]# ls -la
```

```
total 318028
drwxr-xr-x.  4 root root       163 Jun  1 08:26 .
drwxrwxrwt. 11 root root      4096 Jun  1 08:18 ..
-rw-r--r--.  1 root root      3549 Jun  1 08:18 install-config.yaml.bak
drwxr-x---.  2 root root      4096 Jun  1 08:26 manifests
```

Example 3-8 creates a new structure. Per the documentation, change the manifests/cluster-scheduler-02-config.yml files to mark the masters not schedulable as shown in Example 3-9 on page 46.
Example 3-9 Change cluster-scheduler-02-config.yml

```yaml
apiVersion: config.openshift.io/v1
kind: Scheduler
metadata:
  creationTimestamp: null
  name: cluster
spec:
  mastersSchedulable: false
  policy:
    name: 
  status: {}
```

Now create the ignition files as shown in Example 3-10.

Example 3-10 Creating ignition files

```
[root@client install]# ./openshift-install create ignition-configs
INFO Consuming OpenShift Install (Manifests) from target directory
INFO Consuming Worker Machines from target directory
INFO Consuming Openshift Manifests from target directory
INFO Consuming Common Manifests from target directory
INFO Consuming Master Machines from target directory
[root@client install]# ls -la
total 319416
drwxr-xr-x.  3 root root       219 Jun  1 12:20 .
drwxrwxrwt. 11 root root      4096 Jun  1 08:18 ..
drwxr-x---.  2 root root        50 Jun  1 12:20 auth
-rw-r-----.  1 root root    293828 Jun  1 12:20 bootstrap.ign
-rw-r--r--.  1 root root      3459 Jun  1 08:18 install-config.yaml.bak
-rw-r-----.  1 root root      1829 Jun  1 12:20 master.ign
-rw-r-----.  1 root root        96 Jun  1 12:20 metadata.json
-rw-r-r--.  1 root root 325462976 Jun  1 08:21 openshift-install
-rw-r--r--.  1 root root 73018 Jun  1 12:20 .openshift_install.log
-rw-r-----.  1 root root 1226854 Jun  1 12:20 .openshift_install_state.json
-rw-r-----.  1 root root 1829 Jun  1 12:20 worker.ign
```

Copy the ignition files (bootstrap.ign, master.ign and worker.ign) to your httpd file server.

**Note:** You need the files inside the auth directory to access your cluster. Copy the file `auth/kubeconfig` to `/root/.kube/config`. Find the password for the console inside the `kubeadmin-password` file.

Verifying all components to install your cluster

The following steps help to verify all components are ready to install your cluster:

1. The web server is up on the correct port (for example, 8080).
2. Check the CoreOS raw image and the ignition files have the correct permissions.
3. Check TFTP has been started.
4. Check you have the initramfs and the kernel files available.
5. Check you created the grub netboot directory inside your tftp server root. Also that you created the grub.cfg file and it is correctly configured.
6. DHCP needs to point to the tftp server and to the correct boot file.
7. DNS must have all needed entries including the etcd entries.

Confirm that the LoadBalancer correctly points to the control-plane (22623 and 6443), and to the worker nodes (80 and 443).

**Install your CoreOS operating systems on the LPARs**

Follow the minimum requirements to install the LPARs. Start the boot from the bootstrap. The first time you boot you have blank disks assigned, hence there is no point in changing the boot list. Although many of you are familiar with the Power Systems boot, Appendix B, “Booting from System Management Services” on page 85, has been added with the process for first time users.

Use a network boot of your choice to install the servers. 3.3, “Using NIM as your BOOTP infrastructure” on page 50 shows how to do it if you are not familiar with it.

After you start all servers, wait for the bootstrap to complete. This section does not discuss troubleshoot during the boot process. To get information about the bootstrap completion use the command as shown in example Example 3-11.

```
Example 3-11   Wait for bootstrap completion
[root@client install]# ./openshift-install wait-for bootstrap-complete
INFO Waiting up to 30m0s for the Kubernetes API at https://api.ocp4.ibm.lab:6443...
INFO API v1.16.2 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove the bootstrap resources
```

After completion, it is safe to shutdown the bootstrap server as it is not needed anymore in the cluster lifecycle, and everything is done using the control plane. Even adding new nodes is done without a bootstrap.

To get configurations ready for use, we show how to change CoreOS parameters.

The supported way of making configuration changes to CoreOS is through the machineconfig objects (see the login message of CoreOS) as shown in Example 3-12.

```
Example 3-12   Message when logging into CoreOS
Red Hat Enterprise Linux CoreOS 43.81.202005200338.0
Part of OpenShift 4.3, RHCOS is a Kubernetes native operating system
managed by the Machine Config Operator ("clusteroperator/machine-config").
WARNING: Direct SSH access to machines is not recommended; instead,
make configuration changes via `machineconfig` objects:

https://docs.openshift.com/container-platform/4.3/architecture/architecture-rhcos.html

---
Last login: Sun May 31 11:58:27 2020 from 20.0.1.37
[core@worker1 ~]$ 
```

Appendix A, “Configuring Red Hat CoreOS” on page 73 shows how to create machineconfig objects and how to add them.
Important: The configuration shown in Appendix A, “Configuring Red Hat CoreOS” on page 73 is important when using VIOS Shared Ethernet Adapters (SEA), and also for IBM Cloud Pak for Data. It also simplifies SMT management across the cluster making it possible to run different levels of SMT to take advantage of the parallel threading offered by the POWER processor. But also makes it possible for applications sensitive to context switching to run at their best. If you do not apply this configuration, you can end up with authentication operator in unknown state if using SEA, as in our case due to our network configuration. The command to apply the files is `oc apply -f <FILE>`.

With time, the range of operations will startup and become available as shown in Example 3-13.

**Example 3-13   Waiting for all cluster operators to become available**

```bash
[root@client install]# oc get co
NAME                      VERSION AVAILABLE PROGRESSING DEGRADED SINCE
authentication            4.3.19 True False False 1h
cloud-credential          4.3.19 True False False 1h
cluster-autoscale         4.3.19 True False False 1h
console                   4.3.19 True False False 1h
dns                       4.3.19 True False False 1h
image-registry            4.3.19 True False False 1h
ingress                   4.3.19 True False False 1h
insights                  4.3.19 True False False 1h
kube-apiserver            4.3.19 True False False 1h
kube-controller-manager   4.3.19 True False False 1h
kube-scheduler            4.3.19 True False False 1h
machine-api               4.3.19 True False False 1h
machine-config            4.3.19 True False False 1h
marketplace               4.3.19 True False False 1h
monitoring                4.3.19 True False False 1h
network                   4.3.19 True False False 1h
node-tuning               4.3.19 True False False 1h
openshift-apiserver       4.3.19 True False False 1h
openshift-controller-manager 4.3.19 True False False 1h
openshift-samples         4.3.19 True False False 1h
operator-lifecycle-manager 4.3.19 True False False 1h
operator-lifecycle-manager-catalog 4.3.19 True False False 1h
service-ca 4.3.19         4.3.19 True False False 1h
service-catalog-apiserver 4.3.19 True False False 1h
service-catalog-controller-manager 4.3.19 True False False 1h
storage                   4.3.19 True False False 1h
```

The process to check when the cluster is ready can also be monitored using the `openshift-install` command as shown in Example 3-14.

**Example 3-14   Waiting for install completion**

```bash
[root@client install]# ./openshift-install wait-for install-complete
INFO Waiting up to 30m for the cluster at https://api.ocp4.ibm.lab:6443 to initialize...
INFO Waiting up to 10m for the openshift-console route to be created...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export KUBECONFIG=/root/install/auth/kubeconfig'
```
3.2.6 Configuring the internal Red Hat OpenShift registry

Red Hat OpenShift has an internal registry. IBM Cloud Pak for Data documentation mentions the use of this registry to simplify operations. Hence this section shows how to configure it.

Export the NFS server as shown in the documentation at the following website:
https://docs.openshift.com/container-platform/4.3/registry/configuring_registry_storage/configuring-registry-storage-baremetal.html

Example 3-15 shows a sample YAML file describing how to define a static persistent volume in the NFS storage backend.

Example 3-15   YAML file to define persistent value with NFS

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: imageregistry
spec:
  capacity:
    storage: 100Gi
  accessModes:
  - ReadWriteMany
  nfs:
    path: /ocpgpfs/imageregistry
    server: ces.ocp4.ibm.lab
  persistentVolumeReclaimPolicy: Retain
```

Apply the file by issuing the command `oc apply -f <FILE>` as shown in Example 3-16.

Example 3-16   Apply the file

```
[root@client ~]# oc apply -f imageregistrypv.yaml
persistentvolume/imageregistry created
[root@jinete nfs]# oc get pv
NAME     CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM
imageregistry 100Gi RWX Retain Available
```

Now just enable the internal image registry by changing the `managementState` parameter from `Deleted` to `Managed` as shown in Example 3-17. Issue the command `oc edit configs.imageregistry.operator.openshift.io` to open a text editor and make the change.

Example 3-17   Edit Image registry operator

```yaml
apiVersion: imageregistry.operator.openshift.io/v1
kind: Config
metadata:
  creationTimestamp: "2020-05-27T17:04:58Z"
finalizers:
  - imageregistry.operator.openshift.io/finalizer
generation: 4
```
Exposing the internal registry

To push content to the internal registry, you can optionally enable the default route as shown in Example 3-18.

**Example 3-18  Enable the default route**

```
[root@client ~]# oc patch configs.imageregistry.operator.openshift.io/cluster
--patch '{"spec":{"defaultRoute":true}}' --type=merge
config.imageregistry.operator.openshift.io/cluster patched
```

### 3.3 Using NIM as your BOOTP infrastructure

Network Installation Management (NIM) is used to boot, install and maintain AIX operating systems up to date. NIM uses TFTP and BOOTP protocols to boot and install AIX servers from the network. This section shows how to configure a NIM server running on an AIX operating system to boot and install a CoreOS LPAR from the network.

For more information about NIM, visit IBM Knowledge Center at the following link:


**Tip:** This step-by-step guide can also be used with Red Hat Linux V7 and later versions.
3.3.1 Prerequisites and components

This section describes the prerequisites

Steps required to setup the NIM environment and boot
These are the steps required to setup the NIM environment:

- Install NIM master files.
- Install and configure Apache web server for AIX.
- Copy the ignition files and the CoreOS bare metal file to the http server root directory.
- Configure tftp and bootp.
- Copy the CoreOS image files to tftp directory.
- Copy the ignition files to the Apache web server root directory.
- Netboot the LPAR in the Power Systems Hardware Management Console (HMC).

Install NIM master files
For NIM server installation in AIX, the nim.master and nim.spot files from AIX install image needs to be installed. Refer to the IBM knowledge Center site for more information about how to install and configure NIM at the following website:

https://www.ibm.com/support/knowledgecenter/ssw_aix_72/install/basic_config.html

Install Apache web server in AIX
The Apache RPM package httpd for AIX can be downloaded (Example 3-19) and manually installed with all of the package dependencies from the IBM Toolbox for Linux Applications link:


Example 3-19  Download and install apache for AIX from aixtoolbox IBM site

--2020-05-25 20:59:09--
Resolving public.dhe.ibm.com... 170.225.15.112
Connecting to public.dhe.ibm.com|170.225.15.112|:443... connected.
Self-signed certificate encountered.
HTTP request sent, awaiting response... 200 OK
Length: 4279776 (4.1M) [text/plain]
Saving to: 'httpd-2.4.41-1.aix6.1.ppc.rpm'
httpd-2.4.41-1.aix6.1.ppc.rpm
100%[=====================================================>] 4.08M 310KB/s in 27s
2020-05-25 20:59:37 (154 KB/s) - 'httpd-2.4.41-1.aix6.1.ppc.rpm' saved [4279776/4279776]

[ocp43nimserver@root:/bigfs:] rpm -ivh httpd-2.4.41-1.aix6.1.ppc.rpm
error: Failed dependencies:
apr >= 1.5.2-1 is needed by httpd-2.4.41-1.ppc
apr-util >= 1.5.4-1 is needed by httpd-2.4.41-1.ppc
expat >= 2.2.6 is needed by httpd-2.4.41-1.ppc
libapr-1.so is needed by httpd-2.4.41-1.ppc
libaprutil-1.so is needed by httpd-2.4.41-1.ppc
libgcc >= 6.3.0-1 is needed by httpd-2.4.41-1.ppc
libibter.a(libibter-2.4.so.2) is needed by httpd-2.4.41-1.ppc
libldap.a(libldap-2.4.so.2) is needed by httpd-2.4.41-1.ppc
libpcre.a(libpcre.so.1) is needed by httpd-2.4.41-1.ppc
openldap >= 2.4.40 is needed by httpd-2.4.41-1.ppc
pcre >= 8.42 is needed by httpd-2.4.41-1.ppc

Example 3-19 on page 51 shows package dependencies need to be installed first, before the httpd package can be installed. Another option is to install the YUM package manager for AIX and let YUM resolve and install the dependencies.

Installing with YUM

Other installation method can be by installing YUM for AIX and then use YUM to install the Apache httpd package. This can be the preferred method because YUM can automatically download and install all package dependencies for Apache.

YUM install AIX:


After YUM is installed, you can use it to install the Apache web server and all package dependencies as shown in Example 3-20.

**Example 3-20  Install Apache web server in AIX using YUM**

```bash
#yum search httpd
============== N/S Matched: http ==========================================
httpd.ppc : Apache HTTP Server
httpd-devel.ppc : Development tools for the Apache HTTP server.
httpd-manual.ppc : Documentation for the Apache HTTP server.
libnghttp2.ppc : A library implementing the HTTP/2 protocol
libnghttp2-devel.ppc : Files needed for building applications with libnghttp2
mod_http2.ppc : Support for the HTTP/2 transport layer
----------------- output removed -------------
Name and summary matches only, use "search all" for everything.

#yum install httpd.ppc
Setting up Install Process
Resolving Dependencies
--> Running transaction check
----> Package httpd.ppc 0:2.4.41-1 will be installed
Installed:
httpd.ppc 0:2.4.41-1
Dependency Installed:
apr.ppc 0:1.5.2-1   apr-util.ppc 0:1.5.4-1   cyrus-sasl.ppc 0:2.1.26-3
libiconv.ppc 0:1.14-2   libstdc++.ppc 0:8.3.0-2   libunistring.ppc 0:0.9.9-2
libxml2.ppc 0:2.9.9-1
ncurses.ppc 0:6.2-1   openldap.ppc 0:2.4.48-1   pcre.ppc 0:8.43-1
xz-libs.ppc 0:5.2.4-1
Dependency Updated:
bzip2.ppc 0:1.0.8-2   expat.ppc 0:2.2.9-1   gettext.ppc 0:0.19.8.1-5   glib2.ppc
0:2.56.1-2   info.ppc 0:6.6-2   libgcc.ppc 0:8.3.0-2   readline.ppc 0:8.0-2
sqlite.ppc 0:3.28.0-1
Complete!
```
Basic Apache configuration and service start

Edit the Apache configuration file as shown in Example 3-21 to change the service port that listens and verifies the root directory.

The Apache default configuration file is /opt/freeware/etc/httpd/conf/httpd.conf.

Example 3-21   Apache config file modifications

#vi /opt/freeware/etc/httpd/conf/httpd.conf
Listen 8080
#
# DocumentRoot: The directory out of which you will serve your
doctors. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
# DocumentRoot "/var/www/htdocs"
<Directory "/var/www/htdocs">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#  Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.4/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks
#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#  AllowOverride FileInfo AuthConfig Limit
#
AllowOverride None
#
# Controls who can get stuff from this server.
#
Require all granted
</Directory>

Copy ignition files and CoreOS boot file to the Apache root directory

After generating the ignition files as described in Example 3-10 on page 46, these files need to be copied to the Apache root directory in the NIM server.

Ignition files: There is no Red Hat OpenShift installer for AIX, so the ignition files must be created using the openshift-install script in a Linux Red Hat or MacOS operating system.

Copy the bootstrap.ign, master.ign and worker.ign files to the NIM server as shown in Example 3-22 on page 54.
Example 3-22  Copy the ignition files and CoreOS boot file to the NIM server

deploy01# scp /root/openshift-install/*.ign ocp43nimserver:/var/www/htdocs/
deploy01# scp rhcos-4.3.18-ppc64le-metal.ppc64le.raw.gz ocp43nimserver:/var/www/htdocs/

Copy the rhcos-4.3.18-ppc64le-metal.ppc64le.raw.gz file to the Apache root directory as shown in Example 3-22. Example 3-23 shows the contents of the Apache RootFS directory.

Example 3-23  Apache RootFS directory content

ls -la /var/www/htdocs

```
total 1517040
drwxr-xr-x    6 root system          256 May 26 11:09 ..
-rwxrwxr-x    1 root system       295410 May 28 21:44 bootstrap.ign
-rwxrwxr-x    1 root system          45 Jun 11 2007 index.html
-rwxrwxr-x    1 root system       1829 May 28 21:44 master.ign
-rwxrwxr-x    1 root system 776412494 May 28 21:43 rhcos-4.3.18-ppc64le-metal.ppc64le.raw.gz
-rwxrwxr-x    1 root system       1829 May 28 21:44 worker.ign
```

Start and verify the Apache service

Start the httpd service to publish the content in the rootfs as shown in Example 3-24.

Example 3-24  Apache start service command

```
#/opt/freeware/sbin/apachectl -k start
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.16.140.120. Set the 'ServerName' directive globally to suppress this message
```

NOTE: The AH00558: ServerName Warning message can be ignored.

Verify Apache is running and listening on port 8080 as shown in Example 3-25.

Example 3-25  Apache server status

```
# ps -ef | grep -i http
apache  5636450 14221670   0 18:42:49   -  0:00 /opt/freeware/sbin/httpd -k start
apache  5701922 14221670   0 18:42:49   -  0:00 /opt/freeware/sbin/httpd -k start
apache  8323540 14221670   0 18:42:49   -  0:00 /opt/freeware/sbin/httpd -k start
apache  9765306 14221670   0 18:42:49   -  0:00 /opt/freeware/sbin/httpd -k start
root   14221670   1 18:42:49   -  0:00 /opt/freeware/sbin/httpd -k start
apache 15073660 14221670   0 18:42:49   -  0:00 /opt/freeware/sbin/httpd -k start

# netstat -an | grep 8080
tcp 0 0 *.8080 *.* LISTEN
```

Bootp and tftp configuration

The /etc/bootptab file needs to be manually updated to permit network communication to the LPAR you want to boot from the network.

Add the following, according to your configuration, to the end of the bootptab file as shown in Example 3-26 on page 55:

- 172.16.140.95 is the LPAR IP address.
- fa698d138b20 is the LPAR HW address.
172.16.140.120 is the NIM server IP address.
255.255.255.0 is the network mask.

*Example 3-26  bootptab file content*

```
bootstrap:bf=/tftpboot/coreos/boot/grub2/powerpc-ieee1275/core.elf:ip=172.16.140.95:ht=ethernet:ha=fa698d138b20:sa=172.16.140.120:sm=255.255.255.0:
```

**Note:** After the installation is completed, the bootptab lines must be removed manually.

Copy all files from powerpc-ieee1275 directory including the core.elf from the Red Hat deploy01 LPAR to the NIM server as shown in Example 3-27.

*Example 3-27  ieee1275 files copy to the NIM server*

```
deploy01#scp /var/lib/tftpboot/boot/grub2/powerpc-ieee1275/* ocp43nimserver:/tftpboot/coreos/boot/grub2/powerpc-ieee1275/
```

**Grub file configuration**

CoreOS uses grub as the boot file configuration. Example 3-28 shows the boot file only to netboot the bootstrap LPAR.

*Example 3-28  grub2 file options*

```
cat /tftpboot/coreos/boot/grub2/powerpc-ieee1275/grub.cfg
set default=0
set fallback=1
set timeout=10
echo "Loading kernel bootstrap test"
menuentry "bootstrap CoreOS (BIOS)" {
echo "Loading kernel bootstrap"
insmod linux
linux  "/tftpboot/coreos/rhcos-4.3.18-installer-kernel-ppc64le"
systemd.journald.forward_to_console=1 rd.neednet=1
ip=172.16.140.95::172.16.140.1:255.255.255.0:bootstrap.demolab.uy.ibm.com::none
nameserver=172.16.140.70    console=tt0 console=tt0S0 coreos.inst=yes
coreos.inst.install_dev=sda
coreos.inst.image_url=http://172.16.140.120:8080/rhcos-4.3.18-ppc64le-metal.ppc64le.raw.gz
coreos.inst.ignition_url=http://172.16.140.120:8080/bootsap.ign
echo "Loading initrd"
initrd "/tftpboot/coreos/rhcos-4.3.18-installer-initramfs.ppc64le.img"
}
```

**Configure tftp access**

The last config file change is the `/etc/tftpaccess.ctl` file that permits tftp process to some directories on the server as shown in Example 3-29.

*Example 3-29  tftpaccess configuration file to permit boot process*

```
cat /etc/tftpaccess.ctl
# NIM access for network boot
allow:/tftpboot/coreos
```

Chapter 3. Reference installation guide for Red Hat OpenShift V4.3 on Power Systems servers  55
Netboot from the HMC

Finally, the LPAR can be started and booted from the network. This can be done from the HMC command line as shown in Example 3-30 or using the HMC graphical user interface as described in Appendix B, “Booting from System Management Services” on page 85.

- S is the NIM server IP address.
- G is the default gateway.
- C is the LPAR IP address.
- K is the network mask.

Example 3-30  LPAR netboot command in the HMC

```
lpar_netboot -t ent -s auto -d auto -D -S 172.16.140.120 -G 172.16.140.1 -C 172.16.140.95 -K 255.255.255.0 ocp43bootstrap default_profile dc4p824
```

Tip: If more verbose debug is needed for the lpar_netboot command, the –x flag can be specified.

3.4 Installing on scale-out servers bare metal

The ideal container implementation creates an isolation layer for workloads better than server virtualization, saving memory and CPU needed to maintain different workloads. PowerVM can provide you with a layer of separation better than x86 virtualization because of its tight integration with the hardware. But you can also deploy bare metal servers when you want large worker nodes.

This section shows how to add a bare metal server (this is also known by Opal or PowerNV servers) to your existing PowerVM cluster. This can be the most complex case because you are mixing different ways to provision nodes into your cluster. The entire cluster can also be PowerNV only.

Petitboot is the operating system bootloader for scale-out PowerNV systems, and it is based on Linux kexec. Petitboot can use PXE boot to ease the process booting with a simple configuration file. It can load any operating system image that supports the Linux kexec reboot mechanism like Linux and FreeBSD. Petitboot can load images from any device that can be mounted by Linux, and can also load images from the network using the HTTP, HTTPS, NFS, SFTP, and TFTP protocols.

You can still use the same DHCP, HTTP and TFTP servers used for the KVM install and with just a few changes provide the necessary environment to also install PowerNV servers as shown in Example 3-31.

Example 3-31  dhcp.conf file for mixed PowerVM and PowerNV environment

```
default-lease-time 900;
max-lease-time 7200;
subnet 192.168.122.0 netmask 255.255.255.0 {
    option routers 192.168.122.1;
    option subnet-mask 255.255.255.0;
    option domain-search "ocp4.ibm.lab";
    option domain-name-servers 192.168.122.1;
    next-server 192.168.122.5;
    filename "boot/grub2/powerpc-ieee1275/core.elf";
}
allow bootp;
```
option conf-file code 209 = text;
.
.
host powervmworker {
    hardware ethernet 52:54:00:1a:fb:b6;
    fixed-address 192.168.122.30;
    option host-name "powervmworker.ocp4.ibm.lab";
    next-server 192.168.122.5;
    filename "boot/grub2/powerpc-ieee1275/core.elf";
    allow booting;
}

host powervmworker {
    hardware ethernet 98:be:94:73:cd:78;
    fixed-address 192.168.122.31;
    option host-name "powervmworker.ocp4.ibm.lab";
    next-server 192.168.122.5;
    option conf-file "pxelinux/pxelinux.cfg/98-be-94-73-cd-78.cfg";
    allow booting;
}

After changing the dhcp.conf file, restart the dhcpd server. The PXE boot does not point to the grub.cfg file created for the PowerVM hosts but to the configuration file stated on the dhcpd entry.

**Note:** We identified the config file to be `pxelinux/pxelinux.cfg/98-be-94-73-cd-78.cfg`. This means the file has to be on your tftp server. In our case, `/var/lib/tftpboot/pxelinux/pxelinux.cfg/98-be-94-73-cd-78.cfg`.

Create the configuration file as in Example 3-32.

**Example 3-32**  Configuration file example for PXE boot of a PowerNV worker node

```
DEFAULT pxeboot
TIMEOUT 20
PROMPT 0
LABEL pxeboot
KERNEL http://192.168.122.5:8080/rhcos-4.3.18-ppc64le-installer-kernel-ppc64le
APPEND
initrd=http://192.168.122.5:8080/rhcos-4.3.18-ppc64le-installer-initramfs.ppc64le.img
rd.neednet=1 ip= dhcp nameserver=192.168.122.1 console
```

### 3.5 NVMe 4096 block size considerations

At the time this publication was written, there is no 4K Block size image available for CoreOS. If you want to install an NVMe with block size of 4096, you need to first format it to 512 as shown in Example 3-33.

**Example 3-33**  Formatting NVMe to 512 block size

```
[root@localhost ~]# nvme format /dev/nvme0n1 -b 512
Success formatting namespace:1
[root@localhost ~]# nvme list ns
```
<table>
<thead>
<tr>
<th>Node</th>
<th>SN</th>
<th>Model</th>
<th>Format</th>
<th>FW Rev</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/nvme0n1</td>
<td>S46JNE0M700050</td>
<td>PCIe3 1.6TB NVMe Flash Adapter III x8</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

1.60 TB / 1.60 TB 512 B + 0 B MA24MA24

### 3.6 Offline Red Hat OpenShift V4.3 deployment

If you do not have an Internet connection to your Enterprise even via Proxy, you can still install Red Hat OpenShift Container Platform on Power Systems.

Find a bastion host that has access to the Internet which your cluster can reach. This server is not used as a router or any other network service other than the registry for the Red Hat OpenShift Platform. Hence it is not in the production path and it is used for maintenance only.

The online documentation on the Red Hat website explains the concept:


**Note:** The most important part of this documentation is where it shows you how to configure the mirror registry on the bastion node:


These instructions assume that you perform the instructions on a x86 architecture. If you are performing it on a ppc64le, the only change is to change the registry container image to one supported in ppc64le when performing Step 5 of the section in this link:

Chapter 4. IBM Cloud Paks

This chapter covers briefly the IBM Cloud Paks offerings available to use with Red Hat OpenShift. This chapter also provides our experiences installing IBM Cloud Pak for Data, and points the user to the documentation with additional information about supported services and prerequisites.

This chapter contains the following sections:

► Why IBM Cloud Paks, an introduction
► IBM Cloud Paks
► IBM Cloud Paks offerings
► IBM Cloud Pak for Data
4.1 Why IBM Cloud Paks, an introduction

IT world is primarily divided into two major segments. The first segment is new development and the second is run operations where new features and enhancements are deployed to production environments. Both phases have their own set of requirements, constrains and challenges. This is the situation during migration to the cloud. Cloud migration process for a production environment is significantly different from a development environment. This leads to new terminologies like cloud migration and cloud modernization.

Some of the key performance indexes are:

- Easy to manage orchestration service.
- Improved security.
- Efficiency in governance.
- Reduced cost to maximize the return on the investment (ROI).

IBM Cloud Paks are enterprise-ready, containerized middleware and common software solutions, hosted on Kubernetes based platform, that give clients an open, faster and more secure way to move core business applications to any Cloud. This is a full stack, converged infrastructure with virtualized cloud hosting environment that helps to extend applications to the Cloud.

The benefits of IBM Cloud Paks are:

- Market ready: IBM Cloud Paks with Red Hat OpenShift is a flexible combination to ensure faster deployment with high scalability. API based micro services ensure faster adoption of changes. Cloud Native applications are quickly developed using container and microservices which can take advantage of capabilities of middleware database through DevOps practices.
- Run anywhere: IBM Cloud Paks are portable. They can run on-premises, on public clouds, or in an integrated system.

IBM Cloud Paks have been certified by IBM with up-to-date software to provide full-stack support, from hardware to applications.

4.2 IBM Cloud Paks

IBM Cloud Paks are built on the Kubernetes based portable platform leveraging a common container platform from Red Hat OpenShift. This enterprise ready containerized software solution provides several key benefits to different segment of users (Figure 4-1 on page 61) during:

- Build phase: Packaged with open platform components to take advantage of several API services available from different sources. This is easy to build and distribute. Developers take advantage of single application environment, configured with all required tools for planning the modernization process, building Cloud native API and runtime platform to deploy the solution.
- Move phase: Run-anywhere model allows same software to be ported to on-premises or private or public cloud based on the client’s requirement. This is also a transition from a large monolithic application to an API based micro service model.
- Run phase: Available inbuilt services like logging, monitoring, metering, security, identity management and image registry. Each business is unique with a set of key business values which can lead the application to operate from on-premises, private and public
clouds. The unified Kubernetes platform provides such flexibility to deploy and run the same application in any desired platform.

Figure 4-1  IBM Cloud Paks helps users looking to enable workloads

4.3 IBM Cloud Paks offerings

This section describes the different IBM Cloud Paks, descriptions, and capabilities.

4.3.1 IBM Cloud Pak for Application

Accelerates the modernization and building of applications by using built-in developer tools and processes. This includes support for analyzing existing applications and guiding the application owner through the modernization journey. In addition, it supports cloud-native development microservices functions and serverless computing. Customers can quickly build cloud apps, although existing IBM middleware clients gain the most straightforward path to modernization.

Transformation of a traditional application is one the key features in this scope. Development, testing and redeployment, are some of the phases where most of the effort and challenges experienced in traditional development model. Agile DevOps based development model is the of the potential solutions to break this challenge. To complement Agile development process, IBM Cloud Pak for Application extends Kubernetes features for consistent and faster development cycle which helps IBM clients to build cost optimized smarter application.

4.3.2 IBM Cloud Pak for Data

Unifies and simplifies the collection, organization, and analysis of data. Enterprises can turn data into insights through an integrated cloud-native architecture. IBM Cloud Pak for Data is extensible and can be customized to a client's unique data and AI landscapes through an integrated catalog of IBM, open source, and third-party microservices add-ons.
IBM Cloud Pak for Data is covered in more detail in 4.4, “IBM Cloud Pak for Data” on page 63.

**4.3.3 IBM Cloud Pak for Integration**

Integrates applications, data, cloud services and APIs. Supports the speed, flexibility, security, and scale required for modern integration and digital transformation initiatives. It comes with a pre-integrated set of capabilities, which includes API lifecycle, application and data integration, messaging and events, high-speed transfer, and integration security.

Personalize customer experience is the primary business focus which needs integrated view of all scattered data. IBM Cloud Pak for Integration facilitates rapid integration of data along with security, compliance and version capability. Some of the key capabilities are:

- API lifecycle management.
- Application and data integration.
- Enterprise messaging.
- Event streaming.
- High-speed data transfer.
- Secure gateway.

**4.3.4 IBM Cloud Pak for Automation**

Transforms business processes, decisions, and content. A pre-integrated set of essential software that enables clients to easily design, build, and run intelligent automation applications at scale. The three major KPI are:

- Improved efficiently and productivity.
- Enhanced customer experience.
- Operational insight.

IBM Cloud Pak for Automation helps to automate business operations with an integrated platform. Kubernetes makes it easier to configure, deploy, and manage containerized applications. Compatible with all types of projects small and large to deliver better end-to-end customer journeys with improved governing of content and processes.

Automation capabilities empower to work more effectively in the following cases:

- With limited work force to manage higher workload from new applications or services, rising customer demand, or seasonal fluctuations.
- Create enhanced and personalized customer experiences that increase loyalty by drawing insights instantly from multiple sources of information.
- When you need to scale operations to help maximize revenue and customer service.

**4.3.5 IBM Cloud Pak for Multicloud Management**

Provides consistent visibility, governance, and automation across a range of multicloud management capabilities such as infrastructure management, application management, multicluster management, edge management, and integration with existing tools and processes.

---

Note: Although all IBM Cloud Paks are intended to be supported on the ppc64le architecture, at the time of this writing, IBM Cloud Pak for Data was readily available for us to test, hence we worked more with this IBM Cloud Pak specifically.
4.4 IBM Cloud Pak for Data

IBM Cloud Pak for Data is a native cloud solution that provides a single unified interface for your team to connect to your data no matter where it lives, govern it, find it, and use it for analysis.

Data management:
- Connect to data.
- Data governance.
- Identify desired data.
- Data transformation for analytics.

User access and collaboration capability:
- Single unified interface.
- Services to data management and collaboration.
- Data readiness for ready use in analytics.
- Access and connection are inbuilt features.

Functionality with fully integrated data and AI platform:
- Data collection.
- Data organization.
- Data analysis.
- Infuse AI into the business.
- Support multcloud architecture.

Benefits:
- Cost saving.
- Inbuilt services.
- Tools those are integrated.
- Scope for customize solution.
4.4.1 IBM Cloud Pak for Data features and enhancements

IBM Cloud Pak for Data V3.0.1 is the exclusive offering for Red Hat OpenShift. Clients can selectively install and activate required services only. IBM Cloud Pak for Data has several enhancements for usability and more integrated service orientation as shown in Figure 4-2.

![Figure 4-2  IBM Cloud Pak for Data enhancements](image)

These enhancements (Figure 4-2) are classified into the following sections:

Platform:
- Modular services-based platform setup for efficient and optimized utilization of resources.
- Inbuilt dashboard with meaningful KPI for better monitoring, metering and serviceability.
- Open extendable platform with new age Platform and Service APIs.

Installation:
- Simplified installation.
- Red Hat OpenShift V4.3.

Service:
- Data processing and analytics are some of the key enhancements.
- Advanced integration with DataStage®, IBM DB2.
- Advanced predictive and analytical models using IBM SPSS® and Streams Watson suits.

Industry:
- Available industry examples to some solutions for various industries:
  [https://github.com/IBM/Industry-Accelerators/](https://github.com/IBM/Industry-Accelerators/)
Offering (IBM Cloud Pak for Data):
- DataStage Edition.
- Watson Studio Premium.
- DB2.

New services categories:
- AI.
- Integration with Spark, Hadoop.
- Developer tools like Jupyter Notebook with Python V3.6, RStudio V3.6.

IBM Cloud Pak for Data emerged as an extensible and highly scalable data and AI platform. It is built on a comprehensive foundation of unified code, data, and AI services, which is capable of taking advantage of multiple Cloud native services and flexible enough to adopt customization to address specific business needs through an extended service catalog.

Services in the catalog:
- AI: Consists of several Watson libraries, tools and studios.
- Analytics: Services include Trusted Predictive and analytical platforms like Cognos, SPSS, Dashboards, Python, and others.
- Data governance: Consists of IBM InfoSphere® and Watson libraries.
- Data store service: Industry solution, storage.

### 4.4.2 IBM Cloud Pak for Data requirements

The installation requirements can change without notice. It is recommended that you refer to the latest documentation when planning and performing your installation:


The minimum resource recommendations in this publication are for guidance only. Consult with your IBM Sales representative to generate recommendations based on your needs.

**Note:** The time on all of the nodes must be synchronized within 500 ms. Check you have the NTP correctly set. Use the machine configuration method described in Appendix A, “Configuring Red Hat CoreOS” on page 73 to configure /etc/chrony.conf to point to the correct NTP server.

To check your persistent volume provider comply with latency and throughput requirements for IBM Cloud Pak for Data, follow the storage test mentioned in the link:


IBM Cloud Pak for Data comes with many services to make sure you have a complete solution available to you. The services supported on the ppc64le platform and their requirements can be found in the following link:


### 4.4.3 Installing IBM Cloud Pak for Data on Power Systems servers

To install IBM Cloud Pak for Data on Red Hat OpenShift Container Platform V4.3, there are some prerequisites in the following website:
Before installing, you need a registry to hold IBM Cloud Pak for Data images. For simplicity, the internal registry of Red Hat OpenShift can be configured as shown in 3.2.6, “Configuring the internal Red Hat OpenShift registry” on page 49.

**Download the install and repository file**

Use the instructions on the following website to check you have the necessary files for the installation of IBM Cloud Pak for Data:


This case uses the `repo.yaml` file and the `cpd-ppc64le` client. Example 4-1 shows the `repo.yaml` file.

```
Example 4-1  repo.yaml file

registry:  
  - url: <URL>  
    username: <USERNAME>  
    name: base-registry
  fileservers:  
    - url: <URL>
```

You need the entitlement key to install IBM Cloud Pak for Data.

**Offline install method**

The name given to a package that contains a service and can be installed on Red Hat OpenShift is called assembly. The basic offline install method for all services consists on four steps:

1. Download the assembly on a server that has Internet access, and access to the internal registry.
2. Push downloaded images to the internal registry.
3. Apply the admin setup of the namespace for the assembly.
4. Installing the assembly into the namespace.

The base service is the lite assembly that contains the infrastructure and the core part of IBM Cloud Pak for Data. To download the assembly, configured your `repo.yaml` file and issue the command as shown in Example 4-2.

```
Example 4-2  Downloading the assembly

[root@client ~]# ./cpd-ppc64le preloadImages -a lite --action download -r ./repo.yaml --download-path=/repository/lite --arch ppc64le --verbose  
[v] [2020-06-02 06:37:35-0311] Running action mode: download
```

**Note:** Appendix A, “Configuring Red Hat CoreOS” on page 73 provides steps to configure these settings using machine config objects and tuned operator. Check you apply these settings before installing IBM Cloud Pak for Data.
After you downloaded it, push the images to your registry as shown in Example 4-3.

**Example 4-3  Pushing images to the internal registry**

```
[root@client ~]# oc login https://api.ocp4.ibm.lab:6443
Username: kubeadmin
Password: 
Login successful.
```
You have access to 82 projects, the list has been suppressed. You can list all projects with 'oc projects'

Using project "default".

```
[root@client ~]# ./cpd-ppc64le preloadImages --action push
--load-from=/repository/lite
--transfer-image-to=default-route-openshift-image-registry.apps.ocp4.ibm.lab/zen
--target-registry-username=kubeadmin --target-registry-password=$(oc whoami -t) -a
lite --arch ppc64le -v 3.0.1 --insecure-skip-tls-verify
```

```
[INFO] [2020-06-03 05:59:12-0351] --download-path not specified, creating workspace cpd-ppc64le-workspace
[INFO] [2020-06-03 05:59:12-0356] Enter credentials for target registry
default-route-openshift-image-registry.apps.ocp4.ibm.lab/zen
```

```
[INFO] [2020-06-03 05:59:12-0361] 0 file servers and 0 registries detected from current configuration.
[INFO] [2020-06-03 05:59:12-0365] Server configure files validated
[INFO] [2020-06-03 05:59:12-0370] Assembly version is validated
```

```
*** Parsing assembly data and generating a list of charts and images for download ***
```

```
[INFO] [2020-06-03 05:59:12-0380] Assembly data validated
[INFO] [2020-06-03 05:59:13-0062] The category field of module 0010-infra is not specified in its manifest file, assuming default type 'helm-chart'
[INFO] [2020-06-03 05:59:13-0063] The category field of module 0015-setup is not specified in its manifest file, assuming default type 'helm-chart'
[INFO] [2020-06-03 05:59:13-0064] The category field of module 0020-core is not specified in its manifest file, assuming default type 'helm-chart'
```

```
----------------------------------------------------------------------------------
List of charts required by assembly:

<table>
<thead>
<tr>
<th>MODULE</th>
<th>VERSION</th>
<th>ARCHITECTURE</th>
<th>CHART</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0010-infra</td>
<td>3.0.1</td>
<td>ppc64le</td>
<td>0010-infra-3.0.1.tgz</td>
<td>Downloaded</td>
</tr>
<tr>
<td>0015-setup</td>
<td>3.0.1</td>
<td>ppc64le</td>
<td>0015-setup-3.0.1.tgz</td>
<td>Downloaded</td>
</tr>
<tr>
<td>0020-core</td>
<td>3.0.1</td>
<td>ppc64le</td>
<td>0020-zen-base-3.0.1.tgz</td>
<td>Downloaded</td>
</tr>
</tbody>
</table>

```

```
[INFO] [2020-06-03 05:59:20-0316] Pushing image
zen-data-sorcerer:v3.0.1.0-ppc64le-42 from file
/repository/lite/images/zen-data-sorcerer----v3.0.1.0-ppc64le-42.tar (17/17)
Getting image source signatures
Copying blob
sha256:b042988a8be982f5747f4ebc010f18f0f6dca5a5ee4e4302904944a930665c
Copying blob
sha256:9ca4978d298bdf2b55077e7aa15c53e76f1e3d3fb083f59ead330f9c8b201b
Copying blob
sha256:3e17c9d66c285418e1bcefd2fb1278c7d9fcb5c6380bfeea3bd64ed0ba6bcaee2
Copying blob
sha256:b47ddf3a5117f29d7d0c7cf9c59b06d86bed0dfe4eb120af4d50c799043f3bb1
Copying blob
sha256:a39cb5f9504dec02f899b0ee1d778c290a7c0b725b6ddb2fb34a0debda415f
```
The next step is to apply the admin setup as shown in Example 4-4.

Example 4-4  Applying admin setup on the namespace

```
[root@client ~]# ./cpd-ppc64le adm --assembly lite --arch ppc64le --version 3.0.1 --namespace zen --load-from /repository/lite --apply
```

INFO [2020-06-03 05:59:37-0525] 0 file servers and 0 registries detected from current configuration.

INFO [2020-06-03 05:59:37-0530] Server configure files validated

INFO [2020-06-03 05:59:37-0540] Assembly version is validated

* Parsing assembly data and generating a list of charts and images for download *

INFO [2020-06-03 05:59:37-0548] Assembly data validated

INFO [2020-06-03 05:59:38-0046] The category field of module 0010-infra is not specified in its manifest file, assuming default type 'helm-chart'

INFO [2020-06-03 05:59:38-0048] The category field of module 0015-setup is not specified in its manifest file, assuming default type 'helm-chart'

INFO [2020-06-03 05:59:38-0050] The category field of module 0020-core is not specified in its manifest file, assuming default type 'helm-chart'

INFO [2020-06-03 05:59:46-0201]

securitycontextconstraints.security.openshift.io/cpd-zensys-scc added to:

"system:serviceaccount:zen:cpd-admin-sa"

INFO [2020-06-03 05:59:46-0203] Finished executing requests

*** Executing add Cluster Role to SA requests ***

INFO [2020-06-03 05:59:46-0210] Finished executing requests

INFO [2020-06-03 05:59:46-0213] Admin setup executed successfully

Finally you are ready to install the assembly for the lite service as shown in Example 4-5 on page 70.
Example 4-5  Installing the lite assembly

[root@client ~]# ./cpd-ppc64le -a lite --arch ppc64le -c managed-nfs-storage -n zen --cluster-pull-prefix=image-registry.openshift-image-registry.svc:5000/zen --cluster-pull-username=kubeadmin --cluster-pull-password=${(oc whoami -t) --insecure-skip-tls-verify --load-from=/repository/lite

[INFO] [2020-06-03 06:01:31-0974] --download-path not specified, creating workspace cpd-ppc64le-workspace
[INFO] [2020-06-03 06:01:33-0154] Detected root certificate in kube config. Ignoring --insecure-skip-tls-verify flag
[INFO] [2020-06-03 06:01:33-0179] Detected root certificate in kube config. Ignoring --insecure-skip-tls-verify flag
[INFO] [2020-06-03 06:01:33-0599] 0 file servers and 0 registries detected from current configuration.
[INFO] [2020-06-03 06:01:33-0601] Server configure files validated
[INFO] [2020-06-03 06:01:33-0603] Version for assembly is not specified, using the latest version '3.0.1' for assembly lite
[INFO] [2020-06-03 06:01:33-0604] Verifying the CR apiVersion is expected. This process could take up to 30 minutes
*** Parsing assembly data and generating a list of charts and images for download ***

[INFO] [2020-06-03 06:01:33-0778] Assembly data validated
[INFO] [2020-06-03 06:01:33-0949] The category field of module 0010-infra is not specified in its manifest file, assuming default type 'helm-chart'
[INFO] [2020-06-03 06:01:33-0949] The category field of module 0015-setup is not specified in its manifest file, assuming default type 'helm-chart'
[INFO] [2020-06-03 06:01:33-0950] The category field of module 0020-core is not specified in its manifest file, assuming default type 'helm-chart'

<table>
<thead>
<tr>
<th>Module</th>
<th>Arch</th>
<th>Version</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0010-infra</td>
<td>ppc64le</td>
<td>3.0.1</td>
<td>Ready</td>
</tr>
<tr>
<td>0015-setup</td>
<td>ppc64le</td>
<td>3.0.1</td>
<td>Ready</td>
</tr>
<tr>
<td>0020-core</td>
<td>ppc64le</td>
<td>3.0.1</td>
<td>Ready</td>
</tr>
</tbody>
</table>


*** Initializing version configmap for assembly lite ***

[INFO] [2020-06-03 06:10:02-0249] Assembly configmap update complete

[INFO] [2020-06-03 06:10:02-0251] *** Installation for assembly lite completed successfully ***

Your IBM Cloud Pak for Data is ready and accessible. Use your web browser to access the web console URL as the installer installation instruction shows in Example 4-5. You can now login to the console and see the window as shown in Figure 4-3 on page 71.

Note: The default login and password is: admin and password respectively.
Figure 4-3  IBM Cloud Pak for Data: Welcome window

You can repeat the steps from Example 4-2 on page 66 to Example 4-4 on page 69 to install other available assemblies, for example:

- aiopenscale
- cde
- db2oltp
- db2wh
- dods
- hadoop
- hadoop-addon
- lite
- rstudio
- spark
- spss
- spss-modeler
- wml
- wsl

Remember to change lite for the service you want to install.

4.4.4 IBM Cloud Pak for Data backup

Backups on most IBM Cloud Pak for Data services are done quiescing the applications by scaling the applications down to 0 as documented in the following link:


If you are using Spectrum Scale as storage for IBM Cloud Pak for Data persistent volumes using Cluster Export Services, you can use the snapshot feature to enable the backup.
**Note:** You must be aware of any exception as you might want some online backup features for specific applications. For example, IBM DB2 and DB2 Warehouse provides a method for online backup inside the container. Refer to the following websites:


https://www.ibm.com/support/knowledgecenter/SSQNUZ_3.0.1/cpd/svc/dbs/backup_db_db2w.html
Appendix A. Configuring Red Hat CoreOS

This appendix shows how to create machineconfig configuration beneficial to use with IBM Power Systems, and also helps with the requirements for IBM Cloud Pak for Data.

This section contains the following:
- CoreOS machine configuration management and machineconfig objects.
- Using CoreOS tuned operator to apply the sysctl parameters.

The changes provided in this appendix automatically set:
- Correct kernel value for slub_max_order parameter for IBM Cloud Pak for Data.
- Correct number of open files and pids on crio.conf for IBM Cloud Pak for Data
- Automatic setting for SMT with labels.
- Correct sysctl values for IBM Cloud Pak for Data.
CoreOS machine configuration management and machineconfig objects

When you login to the CoreOS by way of SSH, the message explicitly says to use the machineconfig objects as shown in Example A-1.

Example A-1  Message of the day on a regular CoreOS node

[root@client ~]# ssh core@worker1
Red Hat Enterprise Linux CoreOS 43.81.202004201335.0
Part of OpenShift 4.3, RHCOS is a Kubernetes native operating system
managed by the Machine Config Operator (`clusteroperator/machine-config`).

WARNING: Direct SSH access to machines is not recommended; instead, make configuration changes via `machineconfig` objects:
https://docs.openshift.com/container-platform/4.3/architecture/architecture-rhcos.html

---
Last login: Tue Jun  2 12:00:39 2020 from 10.17.201.99
[core@worker1 ~]$ 

Machine objects can be created using YAML files and can enable systemd services or change files on disk. The basic YAML structure is shown in Example A-2.

Example A-2  Basic structure of a machineconfig object

apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: master
  name: <OBJECT NAME>
spec:
  kernelArguments:
    - '<PARAMETER>=<VALUE>'
  config:
    ignition:
      version: 2.2.0
    storage:
      files:
        path: /path/to/file
        overwrite: true
        mode: 0700
        filesystem: root
        contents:
          source: data:text/plain;charset=utf-8;base64,<BASE64 CONTENT>
    systemd:
      units:
        - name: <SERVICENAME>.service
          enabled: true

We use the base64 content for text files because it eases the occurrence of any special characters and non-printable chars resulting in a single stream.
For example, for our case we configure the crio daemon, create a service that watches for a SMT label on the node, and apply the desired SMT configuration using that label. We also set the slub_max_order kernel parameter to 0 (zero).

The crio changes are necessary because some workloads have busy containers, and need more pids and open files than the allocated by default. We do not paste the full crio.conf because we changed only one parameter from the default file `pids_limit = 1024` and added a ulimit parameter as shown in Example A-3.

**Example A-3 Changes on the crio.conf default file**

```
pids_limit = 12288

# Adding nofiles for CP4D
default_ulimits = [
    "nofile=66560:66560"
]
```

We create two files to control the SMT across the cluster as shown in Example A-4. An explanation on how Kubernetes consumes CPU is in 2.4.1, “Red Hat OpenShift sizing guidelines” on page 22.

**Example A-4 Systemd file to run powersmt service**

```
[Unit]
Description=POWERSMT
After=network-online.target

[Service]
ExecStart="/usr/local/bin/powersmt"

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

Example A-4 shows the reference to the file `/usr/local/bin/powersmt`. We need to create that file to set the parameters correctly as shown in Example A-6 on page 76.

You need to transform the file to change any special chars, spaces (that are really meaningful in YAML files) to a simple string in a single line. The process is shown in Example A-5.

**Note:** During the project, we attempted different ways to create the YAML file and using base64 was the easiest way to overcome all situations on the YAML file definition. This is the reason we are using and encouraging the reader to use this format.

**Example A-5 Transforming a file in base64 string**

```
[root@client ~]# cat /etc/systemd/system/powersmt.service | base64 | xargs echo | sed 's/ //g'
W1VuaXRdCkR1c2NyaXBoaW9uPVBvOVVSU01UckFmdGVyPW51dHdvcmstb25saW5lLnRhcmdldAoKW1NTcnZpY2VdCkV4ZWNTdGFydhD0iL3Vzci9sb2Nhbc91aW4vcG93XJzbXQ1CgpbSW5zdGFsbF0KV2FUdGVkJnZXQK
```

**Note:** We do not show this process for other files, but if you want to use base64 for the source of the file contents, repeat the process for any file you have created.
Example A-6 shows a bash loop that uses the `lscpu` command output and the SMT label defined for the node where it is running to make a decision whether to get CPU offline or online. This script checks the value is always smaller than 2 for x86 architectures, 4 for OpenPower, and 8 for Power Systems servers with PowerVM capabilities.

### Example A-6   Service executable for powersmt service

```bash
#!/bin/bash
export PATH=/root/.local/bin:/root/bin:/sbin:/bin:/usr/local/sbin:/usr/local/bin:/$
usr/sbin:/usr/bin
export KUBECONFIG=/var/lib/kubelet/kubeconfig
COREPS=$(/bin/lscpu | /bin/awk -F: ' $1 ~ /^Core\(s\) per socket$/ {print $2}'|/bi
n/xargs)
SOCKETS=$(/bin/lscpu | /bin/awk -F: ' $1 ~ /^Socket\(s\)$/ {print $2}'|/bin/xargs)
let TOTALCORES=$COREPS*$SOCKETS
MAXTHREADS=$(/bin/lscpu | /bin/awk -F: ' $1 ~ /^CPU\(s\)$/ {print $2}'|/bin/xargs)
let MAXSMT=$MAXTHREADS/$TOTALCORES
CURRENTSMT=$(/bin/lscpu | /bin/awk -F: ' $1 ~ /^Thread\(s\) per core$/ {print $2}'
|/bin/xargs)

while :
    do
        ISNODEDEGRADED=$(/bin/oc get node $HOSTNAME -o yaml |/bin/grep machineconfigurat
ion.openshift.io/reason |/bin/grep "unexpected on-disk state validating")
        SMTLABEL=$(/bin/oc get node $HOSTNAME -L SMT --no-headers |/bin/awk '{print $6}'
 )
        if [[ -n $SMTLABEL ]]
            then
                case $SMTLABEL in
                    1) TARGETSMT=1
                        ;;
                    2) TARGETSMT=2
                        ;;
                    4) TARGETSMT=4
                        ;;
                    8) TARGETSMT=8
                        ;;
                        *) TARGETSMT=$CURRENTSMT ; echo "SMT value must be 1, 2, 4, or 8 and small
                        er than Maximum SMT."
                        ;;
                        esac
            else
                TARGETSMT=$MAXSMT
        fi

        if [[ -n $ISNODEDEGRADED ]]
            then
                touch /run/machine-config-daemon-force
        fi

        CURRENTSMT=$(/bin/lscpu | /bin/awk -F: ' $1 ~ /^Thread\(s\) per core$/ {print $2
}|/bin/xargs)

        if [[ $CURRENTSMT -ne $TARGETSMT ]]
            then
                INITONTHREAD=0
```
INITOFFTHREAD=$TARGETSMT
if [[ $MAXSMT -ge $TARGETSMT ]]
then
    while [[ $INITONTHREAD -lt $MAXTHREADS ]]
    do
        ONTHREAD=$INITONTHREAD
        OFFTHREAD=$INITOFFTHREAD
        while [[ $ONTHREAD -lt $OFFTHREAD ]]
        do
            /bin/echo 1 > /sys/devices/system/cpu/cpu$ONTHREAD/online
            let ONTHREAD=$ONTHREAD+1
        done
        let INITONTHREAD=$INITONTHREAD+$MAXSMT
    done
    while [[ $OFFTHREAD -lt $INITONTHREAD ]]
    do
        /bin/echo 0 > /sys/devices/system/cpu/cpu$OFFTHREAD/online
        let OFFTHREAD=$OFFTHREAD+1
    done
    let INITOFFTHREAD=$INITOFFTHREAD+$MAXSMT
else
    echo "Target SMT must be smaller or equal than Maximum SMT supported"
fi
fi
/bin/sleep 30
done

done

The machineconfig file used after including all these alterations is shown in Example A-7.

**Important:** The file shown in Example A-7 applies the configuration to all worker nodes. There is no need to apply this configuration to the master nodes. If you need to, the change `worker` to `master` on both metadata stanza occurrences, and create two different YAML configuration files. Both configuration files must be applied to on all masters and workers nodes. **Do not use this example for your installation, instead build your own file.** You must at minimum update the base64 strings from the ones in your crio.config. Remember that configuration entries can change over time, for example the `pause_image`. By default, the nodes automatically reboot in rolling fashion, one by one, after you apply the configuration.

---

**Example A-7 Machine configuration operator 99_openshift-machineconfig_99-worker-ibm.yaml file**

```yaml
apiVersion: machineconfiguration.openshift.io/v1
class: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 99-worker-ibm
spec:
  kernelArguments:
  - 'slub_max_order=0'
  config:
    ignition:
      version: 2.2.0
    storage:
```

---

Appendix A. Configuring Red Hat CoreOS 77
files:
- path: /usr/local/bin/powersmt
  overwrite: true
  filesystem: root
  contents:
  source: data:text/plain;charset=utf-8;base64,78

- path: /usr/local/bin/powersmt
  overwrite: true
  mode: 0700
  filesystem: root
  contents:
  source: data:text/plain;charset=utf-8;base64,78

- path: /etc/systemd/system/powersmt.service
  overwrite: true
  mode: 0644
  filesystem: root
  contents:
  source: data:text/plain;charset=utf-8;base64,78

- path: /etc/crio/crio.conf
  overwrite: true
  mode: 0644
  filesystem: root
  contents:
  source: data:text/plain;charset=utf-8;base64,78
systemd:
   units:
     - name: powersmt.service
enabled: true

Check the contents of the base64 hash with `base64 -d` to check the configuration you are applying is the one you intend to. You can always create a new hash with a different configuration when you have a different requirement.

After you created the YAML file apply the configuration as shown in Example A-8.

```
Example A-8   Applying 99_openshift-machineconfig_99-worker-ibm.yaml configuration

[root@client ~]# oc apply -f 99_openshift-machineconfig_99-worker-ibm.yaml
machineconfig.machineconfiguration.openshift.io/99-worker-ibm created
```

**Note:** By default, the nodes automatically reboot in rolling fashion, one by one, after you apply the configuration.

---

### Using CoreOS tuned operator to apply the sysctl parameters

To ensure that certain microservices run correctly, you must verify the following kernel parameters:

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

These settings are required for all deployments. Example A-9 assumes that you have worker nodes with 64 GB of RAM. If the worker nodes have 128 GB of RAM each, double the values for the `kernel.shm*` values.

```
Example A-9   64 GB worker node of a tuned.yaml file

apiVersion: tuned.openshift.io/v1
kind: Tuned
metadata:
  name: cp4d-wkc-ipc
  namespace: openshift-cluster-node-tuning-operator
spec:
  profile:
    - name: cp4d-wkc-ipc
      data: |
        [main]
        summary=Tune IPC Kernel parameters on OpenShift Worker Nodes running WKC Pods
        [sysctl]
        kernel.shmmax = 68719476736
        kernel.shmmax = 68719476736
        kernel.shmmax = 68719476736
        kernel.shmmax = 68719476736
        kernel.shmmni = 32768
```

---

By default, the nodes automatically reboot in rolling fashion, one by one, after you apply the configuration.
To apply the parameters follow Example A-10.

**Example A-10  Applying tuned configuration**

```
[root@client ~]# oc apply -f tuned.yaml
tuned.tuned.openshift.io/cp4d-wkc-ipc created
```
Booting from System Management Services

This appendix describes how to boot a partition from the System Management Services (SMS) menu by selecting the boot device for booting from network, and then defining the boot device order to boot the newly installed operating system (OS). The SMS services helps view information about your system or partition, and helps perform tasks such as changing the boot list and setting the network parameters. These SMS menus can be used for AIX or Linux logical partitions.

This appendix contains the following:

- Entering SMS mode
- Option 1: Boot directly to SMS from the HMC
- Option 2: Enter SMS from the startup console
- Configure booting from network
- Configure boot device order
Entering SMS mode

When booting an LPAR, it is possible to enter the SMS menu from the startup console (Figure B-1), or directly when booting from the Hardware Management Console (HMC).

Figure B-1  HMC GUI partitions view
Option 1: Boot directly to SMS from the HMC

To enter SMS from the HMC web GUI, choose the SMS option in the Advanced Settings Boot Mode window as shown in Figure B-2.

Alternatively you can run the following command from the HMC CLI:

```
user@hmc:~> chsysstate -r lpar -m <managed-system> -o on -f <profile> -b sms -n <lpar_name>
```
Option 2: Enter SMS from the startup console

From the startup console, it is possible to enter SMS by pressing 1 the first time this window appears as shown in Figure B-3.

| 1 = SMS Menu | 5 = Default Boot List |
| 2 = Boot From Cd | 6 = Stored Boot List |
| 8 = Open Firmware Prompt | |
| Memory | Keyboard | Network | Speaker |

Figure B-3   Startup console
Configure booting from network

After you are in the SMS menu, choose option number 2: *Setup Remote IPL (Initial Program Load)* as shown in Figure B-4.

![SMS: Main Menu selection pane](Image)
Figure B-5 shows the LPAR attached network cards. Select the network card to be used for network booting. In this case, there is only one, which is selected.

<table>
<thead>
<tr>
<th>Device</th>
<th>Location Code</th>
<th>Hardware Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interpartition Logical LAN</td>
<td>U8408.E8E.219283W-V5-C5-T1</td>
<td>4aed2bc15b05</td>
</tr>
</tbody>
</table>

Navigation keys:
M = return to Main Menu
ESC key = return to previous screen  X = exit System Management Services

Type menu item number and press Enter or select Navigation key:
Select the IP version to use for network booting as shown in Figure B-6. This case uses IPv4.

---

**PowerPC Firmware**
**Version FW840.40 (SV840_147)**
**SMS (c) Copyright IBM Corp. 2000,2016 All rights reserved.**

---

Select Internet Protocol Version.

1. IPv4 - Address Format 123.231.111.222
2. IPv6 - Address Format 1234:5678:90ab:cdef:1234:5678:90ab:cdef

---

Navigation keys:
M = return to Main Menu
ESC key = return to previous screen   X = eXit System Management Services

---

Type menu item number and press Enter or select Navigation key:

---

*Figure B-6  SMS: Select Internet Protocol Version pane*
Select BOOTP option as shown in Figure B-7.

<table>
<thead>
<tr>
<th>PowerPC Firmware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version FW840.40 (SV840_147)</td>
</tr>
<tr>
<td>SMS (c) Copyright IBM Corp. 2000,2016 All rights reserved.</td>
</tr>
</tbody>
</table>

---

Select Network Service.
1. BOOTP
2. iSCSI

---

Navigation keys:
M = return to Main Menu
ESC key = return to previous screen
X = eXit System Management Services

---

Type menu item number and press Enter or select Navigation key: [ ]

Figure B-7  SMS: Select Network Service selection pane
Select IP Parameters (Figure B-8), and set the LPAR address, the bootp server address and the network gateway and mask.

PowerPC Firmware
Version FW840.40 (SV840_147)
SMS (c) Copyright IBM Corp. 2000,2016 All rights reserved.

Network Parameters
Interpartition Logical LAN: U8408.E8E.219283W-V5-C2-T1
1. IP Parameters
2. Adapter Configuration
3. Ping Test
4. Advanced Setup: BOOTP

Navigation keys:
M = return to Main Menu
ESC key = return to previous screen X = eXit System Management Services

Type menu item number and press Enter or select Navigation key:
After you are done, press \textit{M} to go back to the main menu as shown in Figure B-9.

\begin{verbatim}
PowerPC Firmware 
Version FW840.40 (SV840_147) 
SMS (c) Copyright IBM Corp. 2000,2016 All rights reserved. 

---

IP Parameters 

Interpartition Logical LAN: U8408.E8E.219283W-V5-C2-T1 
1. Client IP Address \hspace{1cm} [000.000.000.000] 
2. Server IP Address \hspace{1cm} [000.000.000.000] 
3. Gateway IP Address \hspace{1cm} [000.000.000.000] 
4. Subnet Mask \hspace{1cm} [000.000.000.000] 

---

Navigation keys: 
\textbf{M} = return to Main Menu 
\textbf{ESC} key = return to previous screen \hspace{1cm} \textbf{X} = eXit System Management Services 

---

Type menu item number and press Enter or select Navigation key:
\end{verbatim}

\textit{Figure B-9} SMS: IP Parameters pane
In the main menu, select option number 5, *Select Boot Options* as shown in Figure B-10.

**Figure B-10  SMS: Main Menu pane**
In the boot options menu (Figure B-11), select option number 1, **Select Install/Boot Device**.

![Multiboot pane](image)

---

**Navigation keys:**
- M = return to Main Menu
- ESC key = return to previous screen
- X = eXit System Management Services

**Type menu item number and press Enter or select Navigation key:**

---

*Figure B-11  SMS: Multiboot pane*
Then select option number 4, *Network* as shown in Figure B-12.

![Figure B-12](image)

Select Device Type

1. Tape
2. CD/DVD
3. Hard Drive
4. Network
5. List all Devices

----

Navigation keys:

M = return to Main Menu
ESC key = return to previous screen   X = eXit System Management Services
----

Type menu item number and press Enter or select Navigation key:

[Diagram: SMS: Select Device Type pane]
Select option number 1, **BOOTP** as shown in Figure B-13.

---

**PowerPC Firmware**
Version FW840.40 (SV840_147)
SMS (c) Copyright IBM Corp. 2000, 2016 All rights reserved.

---

Select Network Service.
1. BOOTP
2. iSCSI

---

Navigation keys:
M = return to Main Menu
ESC key = return to previous screen  X = eXit System Management Services

---

Type menu item number and press Enter or select Navigation key:

*Figure B-13  SMS: Select Network Service pane*
Now select the network card you configured for network booting as shown in Figure B-14.

---

**PowerPC Firmware**  
Version FW840.40 (SV840_147)  
**SMS (c) Copyright IBM Corp. 2000,2016 All rights reserved.**

---

**Select Device**  
Device Current Device  
Number Position Name  
1. - Interpartition Logical LAN  
   (loc=U8408.E8E.219283W-V5-C5-T1)

---

**Navigation keys:**  
M = return to Main Menu  
ESC key = return to previous screen  
X = eXit System Management Services

---

Type menu item number and press Enter or select Navigation key: 

---

*Figure B-14  SMS: Select Device pane*
Select option number 2, *Normal Mode Boot* as shown in Figure B-15.

```
<table>
<thead>
<tr>
<th>PowerPC Firmware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version FW840.40 (SV840_147)</td>
</tr>
<tr>
<td>SMS (c) Copyright IBM Corp. 2000,2016 All rights reserved.</td>
</tr>
</tbody>
</table>

---

Select Task

Interpartition Logical LAN

( loc=U8408.E8E.219283W-V5-C5-T1 )

1. Information
2. Normal Mode Boot
3. Service Mode Boot

---

Navigation keys:
M = return to Main Menu
ESC key = return to previous screen       X = eXit System Management Services

---

Type menu item number and press Enter or select Navigation key:

---

*Figure B-15  SMS: Select Task pane*
Exit SMS. Confirm exiting SMS as shown in Figure B-16. This action boots the operating system from the network, installs the operating system and reboots.

PowerPC Firmware
Version FW840.40 (SV840_147)
SMS (c) Copyright IBM Corp. 2000,2016 All rights reserved.

Are you sure you want to exit System Management Services?
1. Yes
2. No

Navigation Keys:

X = eXit System Management Services

Type menu item number and press Enter or select Navigation key:

Figure B-16   SMS: Exit System Management Services pane
## Configure boot device order

After the reboot starts, you must enter again the SMS menu to change the primary boot device, or the LPAR boots from the network again as shown Figure B-17.

<table>
<thead>
<tr>
<th>Memory</th>
<th>Keyboard</th>
<th>Network</th>
<th>Speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
</tr>
<tr>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
</tr>
<tr>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
</tr>
<tr>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
</tr>
<tr>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
<td>IBM</td>
</tr>
</tbody>
</table>

1 = SMS Menu
8 = Open Firmware Prompt

5 = Default Boot List
6 = Stored Boot List

**Figure B-17**  SMS: Change the default boot option
In SMS mode again, select option number 5, *Select Boot Options* as shown in Figure B-18.

![Figure B-18 SMS: Main Menu pane](image-url)
In the boot options menu, select option number 2, *Configure Boot Device Order* as shown in Figure B-19.

```
PowerPC Firmware
Version FW840.40 (SV840_147)
SMS (c) Copyright IBM Corp. 2000,2016 All rights reserved.

-----------------------------------------------
Multiboot
1. Select Install/Boot Device
2. Configure Boot Device Order
3. Multiboot Startup <OFF>
4. SAN Zoning Support

-----------------------------------------------
Navigation keys:
M = return to Main Menu
ESC key = return to previous screen  X = eXit System Management Services

-----------------------------------------------
Type menu item number and press Enter or select Navigation key:
```

*Figure B-19*  SMS: Multiboot pane
Select option number 1, *Select 1st Boot Device*, to set the first boot device as shown in Figure B-20.

![PowerPC Firmware
Version FW840.40 (SV840_147)
SMS (c) Copyright IBM Corp. 2000,2016 All rights reserved.](image)

---

**Configure Boot Device Order**

1. Select 1st Boot Device
2. Select 2nd Boot Device
3. Select 3rd Boot Device
4. Select 4th Boot Device
5. Select 5th Boot Device
6. Display Current Setting
7. Restore Default Setting

---

**Navigation keys:**

*M* = return to Main Menu  
ESC key = return to previous screen  
*X* = exit System Management Services

---

Type menu item number and press Enter or select Navigation key:
Select option number 6, *List All Devices* as shown in Figure B-21.

![Table]

<table>
<thead>
<tr>
<th>Selection</th>
<th>Device Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Tape</td>
</tr>
<tr>
<td>2.</td>
<td>CD/DVD</td>
</tr>
<tr>
<td>3.</td>
<td>Hard Drive</td>
</tr>
<tr>
<td>4.</td>
<td>Network</td>
</tr>
<tr>
<td>5.</td>
<td>None</td>
</tr>
<tr>
<td>6.</td>
<td>List All Devices</td>
</tr>
</tbody>
</table>

**Navigation keys:**

- **M** = return to Main Menu
- **ESC** key = return to previous screen
- **X** = exit System Management Services

Type menu item number and press Enter or select Navigation key:
Select the device where you have installed the OS. Figure B-22 shows four hard drives which are the same due to the multipath configuration. Select one of the hard drives.

```
PowerPC Firmware
Version FW840.40 (SV840_147)
SMS (c) Copyright IBM Corp. 2000,2016 All rights reserved.

---
Select Device
Device Current Device
Number Position Name
1. - Interpartition Logical LAN
   ( loc=U8408.E8E.219283W-V5-C5-T1 )
2. 1 103 GB FC Harddisk
   ( loc=U8408.E8E.219283W-V5-C3-T1-W5001738033270140-L000100000000000 )
3. - 103 GB FC Harddisk
   ( loc=U8408.E8E.219283W-V5-C3-T1-W5001738033270141-L000100000000000 )
4. - 103 GB FC Harddisk
   ( loc=U8408.E8E.219283W-V5-C3-T1-W5001738033270150-L000100000000000 )
5. - 103 GB FC Harddisk
   ( loc=U8408.E8E.219283W-V5-C3-T1-W5001738033270161-L000100000000000 )

---
Navigation keys:
M = return to Main Menu       N = Next page of list
ESC key = return to previous screen       X = eXit System Management Services
---
Type menu item number and press Enter or select Navigation key:
```

Figure B-22  SMS: Select Device pane
Select option number 2, *Set Boot Sequence: Configure as 1st Boot Device* as shown in Figure B-23. This sets the device selected as the first boot device option.

---

**PowerPC Firmware**

**Version FW840.40 (SV840_147)**

SMS (c) Copyright IBM Corp. 2000,2016 All rights reserved.

---

Select Task

103 GB FC Harddisk

( loc=U8408.ESE.219283W-V5-C3-T1-W5001738033270140-L0001000000000000 )

1. **Information**
2. **Set Boot Sequence: Configure as 1st Boot Device**

---

**Navigation keys:**

M = return to Main Menu

ESC key = return to previous screen  X = eXit System Management Services

---

Type menu item number and press Enter or select Navigation key:

---

*Figure B-23  SMS: Select Task pane*
Figure B-24 shows the current boot sequence. Type x to exit SMS.

PowerPC Firmware  
Version FW840.40 (SV840_147)  
SMS (c) Copyright IBM Corp. 2000,2016 All rights reserved.

Current Boot Sequence  
1. 103 GB FC Harddisk  
   ( loc=U8408.E8E.219283W-V5-C3-T1-W5001738033270140-I000100000000000 )  
2. None  
3. None  
4. None

Navigation keys:  
M = return to Main Menu  
ESC key = return to previous screen  
X = eXit System Management Services

Type menu item number and press Enter or select Navigation key:

Confirm you want to exit SMS and boot normally.
Additional material

This paper refers to additional material that can be downloaded from the Internet as described in the following sections.

Locating the web material

The web material associated with this paper is available in softcopy on the Internet from the IBM Redbooks web server:

ftp://www.redbooks.ibm.com/redbooks/REDP5599

Alternatively, you can go to the IBM Redbooks website:

ibm.com/redbooks

Search for REDP5599, select the title, and then click Additional materials to open the directory that corresponds with the IBM Redpaper form number, REDP5599.

Using the web material

The additional web material that accompanies this paper includes the following files:

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REDP5599AdditionalMaterials.zip</td>
<td>Zipped YAML Files</td>
</tr>
</tbody>
</table>

System requirements for downloading the web material

The web material requires the following system configuration:

- **Hard disk space**: 100 MB minimum
- **Operating System**: Windows, Linux or macOS
- **Processor**: i3 or higher
- **Memory**: 1024 MB or higher
Downloading and extracting the web material

Create a subdirectory (folder) on your workstation, and extract the contents of the web material .zip file into this folder.
Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper.

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.

- *Red Hat OpenShift and IBM Cloud Paks on IBM Power Systems: Volume 1*, SG24-8459
- *NIM from A to Z in AIX 5L*, SG24-7296
- *IBM Power System E950: Technical Overview and Introduction*, REDP-5509
- *IBM Power System IC922: Technical Overview and Introduction*, REDP-5584

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:

- IBM PowerVC
- IBM PowerVC Standard Edition V1.4.4
- Network Installation Management
- IBM Redbooks highlighting POWER9 processor-based technology

Help from IBM

IBM Support and downloads

ibm.com/support